MICROCHIP

SAM9X60

Ultra-Low Power Arm® ARM926EJ-S™ Processor-Based MPU, 600 MHz, Camera, LCD, 2D Graphics, Dual 10/100 Ethernet, CAN, USB, QSPI, FLEXCOMs, AES, SHA

Introduction

The SAM9X60 is a high-performance, ultra-low power ARM926EJ-S CPU-based embedded microprocessor (MPU) running up to 600 MHz, with support for multiple memories such as SDRAM, LP-SDRAM, LPDDR, DDR2, and QSPI and e.MMC Flash. The device integrates powerful peripherals for connectivity and user interface applications, and offers security functions (tamper detection, secure boot program, secure key storage, etc.), TRNG, as well as high-performance crypto accelerators for AES and SHA.

Features

- CPU running up to 600 MHz
  - ARM926EJ-S Arm Thumb® processor
  - 32-Kbyte data cache, 32-Kbyte instruction cache, Memory Management Unit (MMU)
- Memories
  - One 160-Kbyte internal ROM
    - 64-Kbyte internal ROM embedding a secure bootloader program supporting boot on Nand Flash, SDCard, SPI or QSPI Flash. Bootloader features selectable by OTP bits
    - 96-Kbyte ROM for NAND Flash BCH ECC table
  - One 64-Kbyte internal SRAM (SRAM0), single-cycle access at system speed
  - High-bandwidth Multi-port DDR2/LPDDR Controller
  - 32/16-bit External Bus Interface (EBI) supporting 8/4-bank DDR2/LPDDR, 4/2-bank SDR/LPSDR, static memories, with scrambling
  - NAND Flash Controller, with up to 24-bit Programmable Multi-bit Error Correcting Code
  - One 11-Kbyte OTP memory for secure key storage with emulation mode (OTP bits are emulated by a 4-Kbyte SRAM (SRAM1))
- System Running up to 200 MHz
  - Power-on reset cells, Reset Controller, Shutdown Controller, Periodic Interval Timer, Watchdog Timer running on internal slow RC oscillator (32 kHz typical) and Real Time Clock running on slow crystal oscillator (32.768 kHz)
  - Two internal trimmed RC oscillators with typical values: 32 kHz (slow) and 12 MHz (fast)
  - Two crystal oscillators: 32.768 kHz (slow) and 12 to 48 MHz (fast)
  - One PLL for the system and one PLL optimized for USB high-speed operation (480 MHz)
  - One dual-port 16-channel DMA Controller
  - Advanced Interrupt Controller and Debug Unit
  - JTAG port with disable bit in OTP memory
  - Two programmable external clock signals
- Low-power Modes
  - Backup mode with RTC, eight 32-bit general purpose backup registers, and Shutdown Controller to control the external power supply
  - Clock Generator and Power Management Controller

Downloaded from Arrow.com.

  - Software-programmable ultra-low power modes: Very Slow Clock operating mode (ULP0), and No-clock operating Mode (ULP1) with fast wake-up capabilities
  - Software programmable power optimization capabilities
- Peripherals
  - LCD Controller with overlay, alpha-blending, rotation, scaling and color conversion. Up to 1024 x 768 resolution
  - 2D Graphics Controller supporting fill BLT, copy BLT, transparent BLT, blend/alpha BLT, ROP4 BLT (Raster Operations) and command ring buffer
  - ITU-R BT. 601/656, up to 12-bit Image Sensor Interface
  - One USB Device High Speed, three USB Host High Speed with dedicated On-Chip Transceivers
  - Two 10/100 Mbps Ethernet Mac Controller
  - Two 4-bit Secure Digital MultiMedia Card Controller
  - Two CAN Controllers
  - One Quad I/O SPI Controller
  - Two three-channel 32-bit Timers/Counters
  - One high resolution (64-bit) Periodic Interval Timer
  - One Synchronous Serial Controller
  - One Inter-IC Sound Multi-Channel Controller with TDM support
  - One Audio Class D Controller with single-ended or bridge-tied load connection to power stage
  - One four-channel 16-bit PWM Controller
  - Thirteen FLEXCOMs (USART, SPI and TWI)
  - One 12-channel 12-bit Analog-to-Digital Converter with 4/5 wires resistive touchscreen support
- Hardware Cryptography
  - SHA (SHA1, SHA224, SHA256, SHA384, SHA512) and HMAC: compliant with FIPS PUB 180-2
  - AES: 256-, 192-, 128-bit key algorithms, compliant with FIPS PUB 197
  - TDES: two-key or three-key algorithms, compliant with FIPS PUB 46-3
  - True Random Number Generator, compliant with NIST Special Publication 800-22 Test Suite and FIPS PUBs 140-2 and 140-3
- I/O Ports
  - Four 32-bit Parallel Input/Output Controllers
  - Up to 112 programmable I/O lines multiplexed with up to three peripheral I/Os
  - Input change interrupt capability on each I/O line, optional Schmitt trigger input
  - Individually programmable open-drain, pull-up and pull-down resistor, synchronous output
  - General-purpose analog and digital inputs tolerant to positive and negative current injection
- Package
  - 228-ball TFBGA 11x11 mm², 0.65 mm pitch, optimized for standard class PCB layout (down to four layers)
- Design for low ElectroMagnetic Interference (EMI)
  - Slewrate-controlled I/Os
  - DDR/SDR Phy with impedance-calibrated drivers
  - Spread spectrum PLLs
  - Careful BGA power/ground ball assignment to provide optimum decoupling capacitors placement
- Operating Conditions
  - Ambient temperature range (T_A): -40°C to +105°C
  - Junction temperature range (T_J): -40°C to +125°C

Downloaded from Arrow.com.

1. Configuration Summary

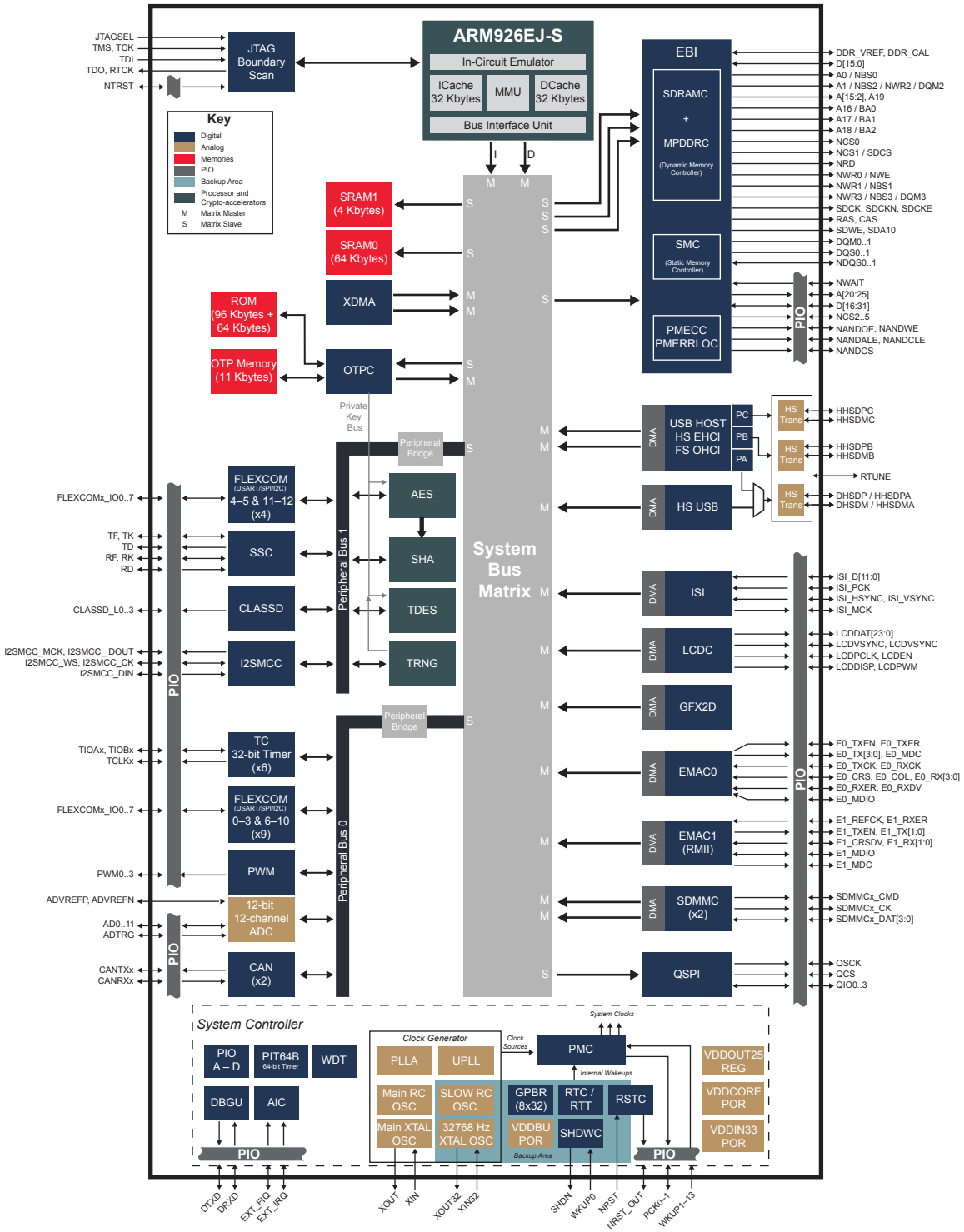
Table 1-1. Configuration Summary

Feature	SAM9X60
Package	BGA228, 11 x 11 mm², 0.65-mm pitch
Core	ARM926 @ 600MHz
SRAM0 + SRAM1	64 Kbytes + 4 Kbytes
L1 Cache (I + D)	32 Kbytes + 32 Kbytes
SDRAM Support	(LPSDR / SDR) 16/32-bit, (LPDDR / DDR2) 16-bit
External Bus I/F	Parallel Bus, NAND Flash
Camera I/F (ISI)	1x 12-bit
EMAC 10/100	1x MII / RMII + 1x RMII
USB	3x HS Tranceivers 2x Host + 1x (H or D)
CAN	2x
LCD / GFX2D	24-bit RGB Up to 1024 x 768 @ 60 fps
SDIO / SDCard / eMMC	2x (4-bit / up to 52 MHz)
ADC	1x 12-bit ADC
Serial I/F	13x FLEXCOMs
DDR QSPI	1x
Audio Peripherals SSC / I2S / CLASSD	1 / 1 / 1
Security	TDES / AES / SHA + Secure Bootloader

Downloaded from Arrow.com.

2. Block Diagram

Figure 2-1. SAM9X60 Block Diagram



Downloaded from Arrow.com.

3. Signal Description

The following table gives details on signal names classified by peripheral.

Table 3-1. Signal Description List

Signal Name	Function	Type	Comments	Active Level
Clocks, Oscillators and PLLs				
XIN	Main Crystal Oscillator Input	Input	–	–
XOUT	Main Crystal Oscillator Output	Output	–	–
XIN32	32.768 kHz Crystal Oscillator Input	Input	–	–
XOUT32	32.768 kHz Crystal Oscillator Output	Output	–	–
RTUNE	USB External Resistor	Analog	–	–
PCK0..1	Programmable Clock Output	Output	–	–
Shutdown, Wakeup Logic				
SHDN	Shutdown Control	Output	–	–
WKUP0..13	Wake-Up Inputs	Input	–	–
ICE and JTAG				
TCK	Test Clock	Input	–	–
TDI	Test Data In	Input	–	–
TDO	Test Data Out	Output	–	–
TMS	Test Mode Select	Input	–	–
JTAGSEL	JTAG Selection	Input	–	–
RTCK	Return Test Clock	Output		–
Reset/Test				
NRST	External nReset Input	Input	–	Low
NRST_OUT	Reset Controller Output	Output	–	Low
NTRST	Test Reset Signal	Input		–
Debug Unit - DBGU				
DRXD	Debug Receive Data	Input	–	–
DTXD	Debug Transmit Data	Output	–	–
Advanced Interrupt Controller - AIC				
EXT_IRQ	External Interrupt Input	Input	–	–
EXT_FIQ	Fast Interrupt Input	Input	–	–
PIO Controller - PIOA - PIOB - PIOC - PIOD				
PA0..31	Parallel IO Controller A	I/O	–	–
PB0..25	Parallel IO Controller B	I/O	–	–
PC0..31	Parallel IO Controller C	I/O	–	–
PD0..21	Parallel IO Controller D	I/O	–	–

Downloaded from Arrow.com.

.....continued

Signal Name	Function	Type	Comments	Active Level
External Bus Interface - EBI				
D[15:0]	Data Bus	I/O	–	–
D[31:16]	Data Bus	I/O	–	–
A[25:0]	Address Bus	Output	–	–
NWAIT	External Wait Signal	Input	–	Low
Static Memory Controller - SMC				
NCS0..5	Chip Select Lines	Output	–	Low
NWR0..3	Write Signal	Output	–	Low
NRD	Read Signal	Output	–	Low
NWE	Write Enable	Output	–	Low
NBS0..3	Byte Mask Signal	Output	–	Low
NAND Flash Controller				
NANDCS	NAND Flash Chip Select	Output	–	Low
NANDOE	NAND Flash Output Enable	Output	–	Low
NANDWE	NAND Flash Write Enable	Output	–	Low
NANDALE	NAND Flash Address Latch Enable	Output	–	–
NANDCLE	NAND Flash Command Latch Enable	Output	–	–
DDR2 / SDRAM / LPDDR / LPSDR Controller				
SDCK	DRAM Clock	Output	–	–
SDCKN	DRAM Clock bar (DDR2 / LPDDR only)		–	
SDCKE	DRAM Clock Enable	Output	–	High
SDCS	DRAM Chip Select	Output	–	Low
BA0..2	Bank Select	Output	–	Low
SDWE	DRAM Write Enable	Output	–	Low
DDR_VREF	DDR2 I/O Reference Voltage	Input	–	–
DDR_CAL	LPDDR / DDR2 Calibration Input	Input	–	–
RAS - CAS	Row and Column Signal	Output	–	Low
SDA10	SDRAM Address 10 Line	Output	–	–
DQS0..1	Positive Data Strobe	I/O	–	–
NDSQ0..1	Negative Data Strobe	I/O	–	–
DQM0..3	Write Data Mask	Output	–	–
Secure Data Memory Card - SDMMCx [0..1]				
SDMMCx_CMD	SDCard / eMMC Command line	I/O	–	–
SDMMCx_CK	SDCard / eMMC Clock Signal	Output	–	–
SDMMCx_DAT[3:0]	SDCard / eMMC Data Lines	I/O	–	–

Downloaded from Arrow.com.

.....continued

Signal Name	Function	Type	Comments	Active Level
Flexible Serial Communication Controller - FLEXCOMx [0..12]				
FLEXCOMx_IO0	TXD / MOSI / TWD	I/O	–	–
FLEXCOMx_IO1	RXD / MISO / TWCK	I/O	–	–
FLEXCOMx_IO2	SCK / SPCK / –	I/O	–	–
FLEXCOMx_IO3	CTS / NPCS0 or NSS / –	I/O	–	–
FLEXCOMx_IO4	RTS / NPCS1 / –	Output	–	–
FLEXCOMx_IO5	– / NPCS2 / –	Output	–	–
FLEXCOMx_IO6	– / NPCS3 / –	Output	–	–
FLEXCOMx_IO7	LONCOL / – / –	Input	–	–
Synchronous Serial Controller - SSC				
TD	SSC Transmit Data	Output	–	–
RD	SSC Receive Data	Input	–	–
TK	SSC Transmit Clock	I/O	–	–
RK	SSC Receive Clock	I/O	–	–
TF	SSC Transmit Frame Sync	I/O	–	–
RF	SSC Receive Frame Sync	I/O	–	–
Image Sensor Interface - ISI				
ISI_D[11:0]	Image Sensor Data	Input	–	–
ISI_MCK	Image sensor Reference Clock	output	–	–
ISI_HSYNC	Image Sensor Horizontal Synchro	input	–	–
ISI_VSYNC	Image Sensor Vertical Synchro	input	–	–
ISI_PCK	Image Sensor Data Clock	input	–	–
Timer / Counter - TCx [0..5]				
TCLKx	TC Channel x External Clock Input	Input	–	–
TIOAx	TC Channel x I/O Line A	I/O	–	–
TIOBx	TC Channel x I/O Line B	I/O	–	–
Pulse Width Modulation Controller- PWMC				
PWM0..3	Pulse Width Modulation Output	Output	–	–
USB Host High Speed Port - UHPHS				
HHSDMA	USB Host Port A High Speed Data -	Analog	–	–
HHSDPA	USB Host Port A High Speed Data +	Analog	–	–
HHSDMB	USB Host Port B High Speed Data -	Analog	–	–
HHSDPB	USB Host Port B High Speed Data +	Analog	–	–
HHSDMC	USB Host Port C High Speed Data -	Analog	–	–
HHSDPC	USB Host Port C High Speed Data +	Analog	–	–

Downloaded from Arrow.com.

.....continued

Signal Name	Function	Type	Comments	Active Level
USB Device High Speed Port - UDPHS				
DHSDM	USB Device High Speed Data -	Analog	–	–
DHSDP	USB Device High Speed Data +	Analog	–	–
Ethernet 10/100 - EMAC0				
E0_TXCK	Transmit Clock or Reference Clock	Input	–	–
E0_RXCK	Receive Clock	Input	–	–
E0_TXEN	Transmit Enable	Output	–	–
E0_TX[3:0]	Transmit Data	Output	–	–
E0_TXER	Transmit Coding Error	Output	–	–
E0_RXDV	Receive Data Valid	Input	–	–
E0_RX[3:0]	Receive Data	Input	–	–
E0_RXER	Receive Error	Input	–	–
E0_CRS	Carrier Sense and Data Valid	Input	–	–
E0_COL	Collision Detect	Input	–	–
E0_MDC	Management Data Clock	Output	–	–
E0_MDIO	Management Data Input/Output	I/O	–	–
RMII Ethernet 10/100 - EMAC1				
E1_REFCK	Transmit Clock or Reference Clock	Input	–	–
E1_TXEN	Transmit Enable	Output	–	–
E1_TX[1:0]	Transmit Data	Output	–	–
E1_CRSDV	Receive Data Valid	Input	–	–
E1_RX[1:0]	Receive Data	Input	–	–
E1_RXER	Receive Error	Input	–	–
E1_MDC	Management Data Clock	Output	–	–
E1_MDIO	Management Data Input/Output	I/O	–	–
LCD Controller - LCDC				
LCDDAT[23:0]	LCD Data Bus	Output	–	–
LCDVSYNC	LCD Vertical Synchronization	Output	–	–
LCDHSYNC	LCD Horizontal Synchronization	Output	–	–
LCDPCLK	LCD Pixel Clock	Output	–	–
LCDDEN	LCD Data Enable	Output	–	–
LCDPWM	LCD Contrast Control	Output	–	–
LCDDISP	LCD Display Enable	Output	–	–
12-bit Analog-to-Digital Converter with Resistive Touch - ADC				
$AD0_{XP_UL}$	Top/Upper Left Channel	Analog	–	–

Downloaded from Arrow.com.

.....continued

Signal Name	Function	Type	Comments	Active Level
$AD1_{XM_UR}$	Bottom/Upper Right Channel	Analog	–	–
$AD2_{YP_LL}$	Right/Lower Left Channel	Analog	–	–
$AD3_{YM_SENSE}$	Left/Sense Channel	Analog	–	–
$AD4_{LR}$	Lower Right Channel	Analog	–	–
AD5..11	7 Analog Inputs	Analog	–	–
ADTRG	ADC Trigger	Input	–	–
ADVREFN	ADC Negative Input Reference Voltage	Analog	–	–
ADVREFP	ADC Positive Input Reference Voltage	Analog	–	–
CAN Controller - CANx [0..1]				
CANRXx	CAN Receive	Input	–	–
CANTXx	CAN Transmit	Output	–	–
Class D Controller - CLASSD				
CLASSD_L0	Class D Controller Left Output 0	Output		
CLASSD_L1	Class D Controller Left Output 1	Output		
CLASSD_L2	Class D Controller Left Output 2	Output		
CLASSD_L3	Class D Controller Left Output 3	Output		
Quad I/O SPI - QSPI				
QSCK	Quad IO SPI Serial Clock	Output		
QCS	Quad IO SPI Chip Select	Output		
QIO3..0	Quad IO SPI I/O 0 to 3	I/O		
Inter IC Sound Multi Channel Controller - I2SMCC				
I2SMCC_MCK	Master Clock	Output		
I2SMCC_CK	Serial Clock	I/O		
I2SMCC_WS	I2S Word Select	I/O		
I2SMCC_DIN	Serial Data Input	Input		
I2SMCC_DOUT	Serial Data Output	Output		

Downloaded from Arrow.com.

4. Safety and Security Features

4.1 Design for Safety and IEC60730 Class B Certification

4.1.1 Background Information

The IEC 60730 standard encompasses all aspects of appliance design. Annex H of the standard covers the aspects most relevant to microcontrollers. It details the tests and diagnostics which are intended to ensure safe operation of embedded control hardware and software. IEC 60730 defines three classifications for electronic control functions:

- Class A - Control functions which are not intended to be relied upon for safety of the equipment
- Class B - Control functions intended to prevent unsafe operation of the controlled equipment
- Class C - Control functions intended to prevent special hazards such as explosions

Specific design techniques have been used in the SAM9X60 to ease compliance with the IEC 60730 Class B Certification and to resolve general-purpose safety concerns. This allows reduced software development and code size as well as savings on external hardware circuitry, since built-in self-tests are already embedded in the MPU. Table 4-1 gives the list of peripherals which incorporate these techniques, and details whether these features are applicable for the IEC 60730 Class B Certification or for general-purpose safety considerations.

4.2 Design for Security

The SAM9X60 embeds peripherals with security features to prevent counterfeiting, to secure external communication, and to authenticate the system.

Table 4-2 provides the list of peripherals and an overview of their security function. For more information, see the sections on each peripheral.

4.3 Safety and IEC 60730 Features

Table 4-1. Safety and IEC 60730 Features

Peripheral	Component	Fault/Error/Feature	Requirements for Class B IEC 60730(1)	General Safety
PMC	Clock	MCK frequency monitor - MCK out-of-range operation	–	X
		32.768 kHz crystal oscillator frequency monitor - Abnormal frequency deviation	X	X
		Main crystal oscillator failure detector - Crystal failure detection	X	X
System Controller	All	Safety critical peripherals and/or counters are fed by the always-on slow RC oscillator - WDT, RSTC, startup counters, timeout counters, etc.	–	X

Downloaded from Arrow.com.

.....continued

Peripheral	Component	Fault/Error/Feature	Requirements for Class B IEC 60730(1)	General Safety
PIOC	I/O lines	Digital I/O - Plausibility check	X	–
ADCC		Analog I/O and ADC converter - Plausibility check	X	–
NAND Flash Controller ECC	Memory	Non-volatile memory - Multiple error detection (2 to 24)	–	X
WDT, RSTC	Watchdog	Watchdog is driven by an internal always on clock - Program counter stuck at faults	X	X
		Watchdog configuration can be locked until the next reset - Errant writes (programming errors, errors introduced by system or hardware failures)	–	X
		Watchdog overflow generates a system reset	X	X
ARM926EJ-S MMU	Memory Management Unit	ARM926EJ-S Memory Management Unit	–	X
MATRIX, AIC, RTC, RTT, RSTC, SHDWC, SDRAM, PMC, PIOC, MPDDRC, SMC, CLASSD, SSC, FLEXCOM, QSPI, TC, I2SMCC, ADC	Peripherals	Configuration, Interrupt Enable/ Disable, Control registers can be independently write-protected - Errant writes (programming errors, errors introduced by system or hardware failures)	–	X
AES, TDES, SHA, PIT64B, TC, SDRAMC, MPDDRC	Peripherals	Embedded integrity checker with reports in status registers	–	X

Note:

1. Class B IEC 60730 Requirements. Annex H - Table H.1 (H.11.12.7 in Edition 3).

4.4 Security Features

Table 4-2. Security Features

Peripheral	Function	Description	Comments
ARM926EJ-S MMU	Memory Management Unit	Memory Management Unit	–
PIO	I/O Control/ Peripheral Access	When a peripheral is not selected (PIO-controlled), I/O lines have no access to the peripheral.	–

Downloaded from Arrow.com.

.....continued

Peripheral	Function	Description	Comments
AES	Cryptography Standards	Hardware-accelerated AES up to 256 bits	FIPS-compliant
SHA		SHA up to 512 and HMAC-SHA	
TDES		Hardware-accelerated Triple DES	
TRNG		True Random Number Generator	
AES, TDES	Cryptography Tamper	Immediate clear of keys in case of external tamper event detection (if enabled)	
AES, TDES, SHA	Cryptography Integrity Checks	AES/TDES/SHA embed integrity checks on configuration registers and algorithm circuitries and a specific flag in status register. If this specific flag is set, an integrity error has been detected. This can occur only on abnormal operating conditions (electromagnetic attacks, VDD glitches, etc.)	–
OTPC, AES, TDES, TRNG	Cryptography Private Key Bus	Capability to transfer a key to AES/TDES in a totally invisible manner from software	–
Secure Boot	Secure Boot	Code encrypted/decrypted, Trusted Code Authentication	Hardware SHA (HMAC) + Software RSA or AES Hardware (CMAC)
Memories	Scrambling	On-the-fly scrambling/ unscrambling for memories	All external memories such as QSPI, DDR, and all memories on SMC

Downloaded from Arrow.com.

.....continued

Peripheral	Function	Description	Comments
RTC	IO Tamper Pin	Eight tamper detection pins	VDDCORE WKUP1 to WKUP8 pins can be selected as a source of tamper, performing an immediate clear of AES/ TDES keys (if enabled), immediate clear of scrambling keys in SDR/DDR/QSPI/SMC, and immediate clear of General Purposes Backup Registers (if enabled)
	Timestamping	Timestamping of tamper events	All events are logged in the RTC. Timestamping gives the source of the reset/ erase memory/interruption
	Configuration	Protection against bad configuration (invalid entry for date and time are impossible)	–
	Glitch Robustness	Glitch on 32 KHz does not corrupt the downstream counters	Glitch on 32 KHz can only create a phase shift of the downstream counters
	Integrity Check	If RTC Status flag TDERR is set, counters integrity have been corrupted	–
Secure OTP	JTAG Access Control	Disable JTAG access by OTP bit	–
PIT64B, TC	Integrity Checks	PIT64B/TC embed integrity checks on configuration registers and algorithm circuitries and a specific flag in status register. If this specific flag is set, an integrity error has been detected. This can occur only on abnormal operating conditions (electromagnetic attacks, VDD glitches, etc.)	–
GPBR	Access Protection	GPBR can be write-protected and/or read-protected	–
	Tamper	GBPR can be immediately cleared on tamper detection (if enabled)	–

Downloaded from Arrow.com.

5. Package and Pinout

5.1 Packages

The SAM9X60 is available in the package indicated in the following table.

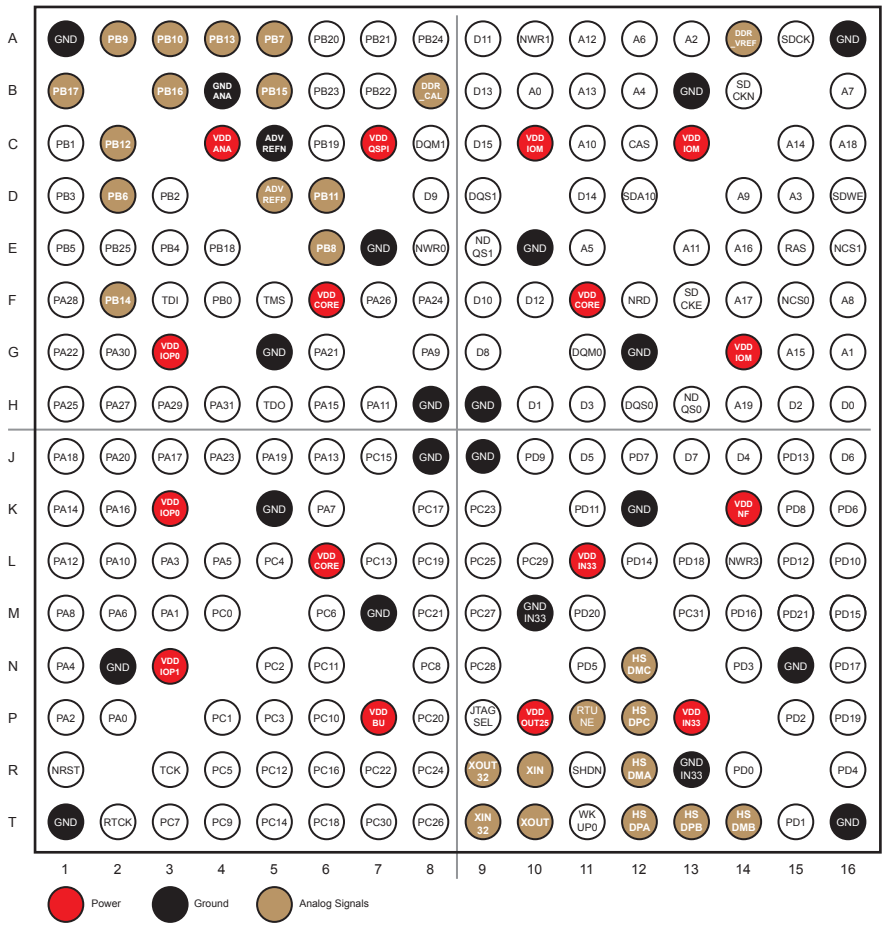
Table 5-1. SAM9X60 Package

Package Name	Pin Count	Ball Pitch
TFBGA228L	228	0.65 mm

For further details, refer to 58. Mechanical Characteristics.

5.2 Pinout

Figure 5-1. SAM9X60 BGA228 Pinout



Downloaded from Arrow.com.

Table 5-2. Pin Description

228-pin BGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral			Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
P2	VDDIOP0	GPIO	PA0	I/O	—	—	A	FLEXCOM0_IO0	I/O	PIO, I, PU, ST
							B	FLEXCOM5_IO4	O	
							C	FLEXCOM4_IO4	O	
M3	VDDIOP0	GPIO	PA1	I/O	—	—	A	FLEXCOM0_IO1	I/O	PIO, I, PU, ST
							B	FLEXCOM4_IO5	O	
P1	VDDIOP0	GPIO	PA2	I/O	WKUP1	—	A	FLEXCOM0_IO4	O	PIO, I, PU, ST
							B	SDMMC1_DAT1	I/O	
							C	E0_TX0	O	
L3	VDDIOP0	GPIO	PA3	I/O	—	—	A	FLEXCOM0_IO3	I/O	PIO, I, PU, ST
							B	SDMMC1_DAT2	I/O	
							C	E0_TX1	O	
N1	VDDIOP0	GPIO	PA4	I/O	—	—	A	FLEXCOM0_IO2	I/O	PIO, I, PU, ST
							B	SDMMC1_DAT3	I/O	
							C	E0_TXER	O	
L4	VDDIOP0	GPIO	PA5	I/O	—	—	A	FLEXCOM1_IO0	I/O	PIO, I, PU, ST
							B	CANTX1	O	
M2	VDDIOP0	GPIO	PA6	I/O	—	—	A	FLEXCOM1_IO1	I/O	PIO, I, PU, ST
							B	CANRX1	I	
K6	VDDIOP0	GPIO	PA7	I/O	—	—	A	FLEXCOM2_IO0	I/O	PIO, I, PU, ST
							B	FLEXCOM4_IO4	O	
							C	FLEXCOM5_IO4	O	
M1	VDDIOP0	GPIO	PA8	I/O	—	—	A	FLEXCOM2_IO1	I/O	PIO, I, PU, ST
							B	FLEXCOM5_IO3	I/O	
							C	FLEXCOM4_IO5	O	
G8	VDDIOP0	GPIO	PA9	I/O	WKUP2	—	A	DRXD	I	PIO, I, PU, ST
							B	CANRX0	I	
L2	VDDIOP0	GPIO	PA10	I/O	WKUP3	—	A	DTXD	O	PIO, I, PU, ST
							B	CANTX0	O	

© 2020 Microchip Technology Inc.

Downloaded from Arrow.com.

.....continued

228-pin BGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral			Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
H7	VDDIOP0	GPIO	PA11	I/O	—	—	A	FLEXCOM4_IO1	I/O	PIO, I, PU, ST
							B	SDMMC1_DAT0	I/O	
L1	VDDIOP0	GPIO	PA12	I/O	—	—	A	FLEXCOM4_IO0	I/O	PIO, I, PU, ST
							B	SDMMC1_CMD	I/O	
J6	VDDIOP0	GPIO	PA13	I/O	—	—	A	FLEXCOM4_IO2	I/O	PIO, I, PU, ST
							B	SDMMC1_CK	I/O	
K1	VDDIOP0	GPIO	PA14	I/O	—	—	A	FLEXCOM4_IO3	I/O	PIO, I, PU, ST
H6	VDDIOP0	GPIO	PA15	I/O	—	—	A	SDMMC0_DAT0	I/O	PIO, I, PU, ST
K2	VDDIOP0	GPIO	PA16	I/O	—	—	A	SDMMC0_CMD	I/O	PIO, I, PU, ST
J3	VDDIOP0	GPIO	PA17	I/O	—	—	A	SDMMC0_CK	I/O	PIO, I, PU, ST
J1	VDDIOP0	GPIO	PA18	I/O	—	—	A	SDMMC0_DAT1	I/O	PIO, I, PU, ST
J5	VDDIOP0	GPIO	PA19	I/O	—	—	A	SDMMC0_DAT2	I/O	PIO, I, PU, ST
J2	VDDIOP0	GPIO	PA20	I/O	—	—	A	SDMMC0_DAT3	I/O	PIO, I, PU, ST
G6	VDDIOP0	GPIO	PA21	I/O	—	—	A	TIOA0	I/O	PIO, I, PU, ST
							B	FLEXCOM5_IO1	I/O	
G1	VDDIOP0	GPIO	PA22	I/O	—	—	A	TIOA1	I/O	PIO, I, PU, ST
							B	FLEXCOM5_IO0	I/O	
J4	VDDIOP0	GPIO	PA23	I/O	—	—	A	TIOA2	I/O	PIO, I, PU, ST
							B	FLEXCOM5_IO2	I/O	
F8	VDDIOP0	GPIO	PA24	I/O	—	—	A	TCLK0	I	PIO, I, PU, ST
							B	TK	I/O	
							C	CLASSD_L0	O	
H1	VDDIOP0	GPIO	PA25	I/O	—	—	A	TCLK1	I	PIO, I, PU, ST
							B	TF	I/O	
							C	CLASSD_L1	O	
F7	VDDIOP0	GPIO	PA26	I/O	—	—	A	TCLK2	I	PIO, I, PU, ST
							B	TD	O	
							C	CLASSD_L2	O	

© 2020 Microchip Technology Inc.

Downloaded from Arrow.com.

.....continued

228-pin BGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral			Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
H2	VDDIOP0	GPIO	PA27	I/O	—	—	A	TIOB0	I/O	PIO, I, PU, ST
							B	RD	I	
							C	CLASSD_L3	O	
F1	VDDIOP0	GPIO	PA28	I/O	WKUP4	—	A	TIOB1	I/O	PIO, I, PU, ST
							B	RK	I/O	
H3	VDDIOP0	GPIO	PA29	I/O	—	—	A	TIOB2	I/O	PIO, I, PU, ST
							B	RF	I/O	
							C	FLEXCOM2_IO7	I	
G2	VDDIOP0	GPIO	PA30	I/O	—	—	A	FLEXCOM6_IO0	I/O	PIO, I, PU, ST
							B	FLEXCOM5_IO6	O	
							C	E0_MDC	O	
H4	VDDIOP0	GPIO	PA31	I/O	—	—	A	FLEXCOM6_IO1	I/O	PIO, I, PU, ST
							B	FLEXCOM5_IO5	O	
							C	E0_TXEN	O	
F4	VDDANA	GPIO	PB0	I/O	WKUP5	—	A	E0_RX0	I	PIO, I, PU, ST
							B	FLEXCOM2_IO4	O	
C1	VDDANA	GPIO	PB1	I/O	—	—	A	E0_RX1	I	PIO, I, PU, ST
							B	FLEXCOM2_IO3	I/O	
D3	VDDANA	GPIO	PB2	I/O	—	—	A	E0_RXER	I	PIO, I, PU, ST
							B	FLEXCOM2_IO2	I/O	
D1	VDDANA	GPIO	PB3	I/O	WKUP6	—	A	E0_RXDV	I	PIO, I, PU, ST
							B	FLEXCOM4_IO6	O	
E3	VDDANA	GPIO	PB4	I/O	—	—	A	E0_TXCK	I/O	PIO, I, PU, ST
							B	FLEXCOM8_IO0	I/O	
E1	VDDANA	GPIO	PB5	I/O	—	—	A	E0_MDIO	I/O	PIO, I, PU, ST
							B	FLEXCOM8_IO1	I/O	
D2	VDDANA	GPIO	PB6	I/O	AD7	—	A	E0_MDC	O	PIO, I, PU, ST
							B	FLEXCOM0_IO7	I	
A5	VDDANA	GPIO	PB7	I/O	AD8	—	A	E0_TXEN	O	PIO, I, PU, ST

© 2020 Microchip Technology Inc.

Downloaded from Arrow.com.

.....continued

228-pin BGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral			Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
E6	VDDANA	GPIO	PB8	I/O	AD9	—	A	E0_TXER	O	PIO, I, PU, ST
A2	VDDANA	GPIO	PB9	I/O	AD10	—	A	E0_TX0	O	PIO, I, PU, ST
							B	PCK1	O	
A3	VDDANA	GPIO	PB10	I/O	AD11	—	A	E0_TX1	O	PIO, I, PU, ST
							B	PCK0	O	
D6	VDDANA	GPIO	PB11	I/O	AD0	—	A	E0_TX2	O	PIO, I, PU, ST
							B	PWM0	O	
C2	VDDANA	GPIO	PB12	I/O	AD1	—	A	E0_TX3	O	PIO, I, PU, ST
							B	PWM1	O	
A4	VDDANA	GPIO	PB13	I/O	AD2	—	A	E0_RX2	I	PIO, I, PU, ST
							B	PWM2	O	
F2	VDDANA	GPIO	PB14	I/O	AD3	—	A	E0_RX3	I	PIO, I, PU, ST
							B	PWM3	O	
B5	VDDANA	GPIO	PB15	I/O	AD4	—	A	E0_RXCK	I	PIO, I, PU, ST
B3	VDDANA	GPIO	PB16	I/O	AD5	—	A	E0_CRS	I	PIO, I, PU, ST
B1	VDDANA	GPIO	PB17	I/O	AD6	—	A	E0_COL	I	PIO, I, PU, ST
E4	VDDANA	GPIO	PB18	I/O	WKUP7	—	A	IRQ	I	PIO, I, PU, ST
							B	ADTRG	I	
C6	VDDQSPI	GPIO	PB19	I/O	—	—	A	QSCK	O	PIO, I, PU, ST
							B	I2SMCC_CK	I/O	
							C	FLEXCOM11_IO0	I/O	
A6	VDDQSPI	GPIO	PB20	I/O	—	—	A	QCS	O	PIO, I, PU, ST
							B	I2SMCC_WS	I/O	
							C	FLEXCOM11_IO1	I/O	
A7	VDDQSPI	GPIO	PB21	I/O	—	—	A	QIO0	I/O	PIO, I, PU, ST
							B	I2SMCC_DIN0	I	
							C	FLEXCOM12_IO0	I/O	

Downloaded from Arrow.com.

.....continued

228-pin BGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral			Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
B7	VDDQSPI	GPIO	PB22	I/O	—	—	A	QIO1	I/O	PIO, I, PU, ST
							B	I2SMCC_DOUT0	O	
							C	FLEXCOM12_IO1	I/O	
B6	VDDQSPI	GPIO	PB23	I/O	—	—	A	QIO2	I/O	PIO, I, PU, ST
							B	I2SMCC_MCK	O	
A8	VDDQSPI	GPIO	PB24	I/O	—	—	A	QIO3	I/O	PIO, I, PU, ST
E2	VDDIOP0	GPIO	PB25	I/O	WKUP8	—	A	NRST_OUT	O	NRST_OUT, O, PD
							B	NTRST	I	
M4	VDDIOP1	GPIO	PC0	I/O	—	—	A	LCDDAT0	O	PIO, I, PU, ST
							B	ISI_D0	I	
							C	FLEXCOM7_IO0	I/O	
P4	VDDIOP1	GPIO	PC1	I/O	—	—	A	LCDDAT1	O	PIO, I, PU, ST
							B	ISI_D1	I	
							C	FLEXCOM7_IO1	I/O	
N5	VDDIOP1	GPIO	PC2	I/O	—	—	A	LCDDAT2	O	PIO, I, PU, ST
							B	ISI_D2	I	
							C	TIOA3	I/O	
P5	VDDIOP1	GPIO	PC3	I/O	—	—	A	LCDDAT3	O	PIO, I, PU, ST
							B	ISI_D3	I	
							C	TIOB3	I/O	
L5	VDDIOP1	GPIO	PC4	I/O	—	—	A	LCDDAT4	O	PIO, I, PU, ST
							B	ISI_D4	I	
							C	TCLK3	I	
R4	VDDIOP1	GPIO	PC5	I/O	—	—	A	LCDDAT5	O	PIO, I, PU, ST
							B	ISI_D5	I	
							C	TIOA4	I/O	
M6	VDDIOP1	GPIO	PC6	I/O	—	—	A	LCDDAT6	O	PIO, I, PU, ST
							B	ISI_D6	I	
							C	TIOB4	I/O	

© 2020 Microchip Technology Inc.

Downloaded from Arrow.com.

.....continued

228-pin BGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral			Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
T3	VDDIOP1	GPIO	PC7	I/O	—	—	A	LCDDAT7	O	PIO, I, PU, ST
							B	ISI_D7	I	
							C	TCLK4	I	
N8	VDDIOP1	GPIO	PC8	I/O	—	—	A	LCDDAT8	O	PIO, I, PU, ST
							B	ISI_D8	I	
							C	FLEXCOM9_IO0	I/O	
T4	VDDIOP1	GPIO	PC9	I/O	—	—	A	LCDDAT9	O	PIO, I, PU, ST
							B	ISI_D9	I	
							C	FLEXCOM9_IO1	I/O	
P6	VDDIOP1	GPIO	PC10	I/O	—	—	A	LCDDAT10	O	PIO, I, PU, ST
							B	ISI_D10	I	
							C	PWM0	O	
N6	VDDIOP1	GPIO	PC11	I/O	—	—	A	LCDDAT11	O	PIO, I, PU, ST
							B	ISI_D11	I	
							C	PWM1	O	
R5	VDDIOP1	GPIO	PC12	I/O	—	—	A	LCDDAT12	O	PIO, I, PU, ST
							B	ISI_PCK	I	
							C	TIOA5	I/O	
L7	VDDIOP1	GPIO	PC13	I/O	—	—	A	LCDDAT13	O	PIO, I, PU, ST
							B	ISI_VSYNC	I	
							C	TIOB5	I/O	
T5	VDDIOP1	GPIO	PC14	I/O	—	—	A	LCDDAT14	O	PIO, I, PU, ST
							B	ISI_HSYNC	I	
							C	TCLK5	I	
J7	VDDIOP1	GPIO	PC15	I/O	—	—	A	LCDDAT15	O	PIO, I, PU, ST
							B	ISI_MCK	O	
							C	PCK0	O	

Downloaded from Arrow.com.

.....continued

228-pin BGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral			Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
R6	VDDIOP1	GPIO	PC16	I/O	—	—	A	LCDDAT16	O	PIO, I, PU, ST
							B	E1_RXER	I	
							C	FLEXCOM10_IO0	I/O	
K8	VDDIOP1	GPIO	PC17	I/O	—	—	A	LCDDAT17	O	PIO, I, PU, ST
							B	FLEXCOM1_IO7	I	
							C	FLEXCOM10_IO1	I/O	
T6	VDDIOP1	GPIO	PC18	I/O	—	—	A	LCDDAT18	O	PIO, I, PU, ST
							B	E1_TX0	O	
							C	PWM0	O	
L8	VDDIOP1	GPIO	PC19	I/O	—	—	A	LCDDAT19	O	PIO, I, PU, ST
							B	E1_TX1	O	
							C	PWM1	O	
P8	VDDIOP1	GPIO	PC20	I/O	—	—	A	LCDDAT20	O	PIO, I, PU, ST
							B	E1_RX0	I	
							C	PWM2	O	
M8	VDDIOP1	GPIO	PC21	I/O	—	—	A	LCDDAT21	O	PIO, I, PU, ST
							B	E1_RX1	I	
							C	PWM3	O	
R7	VDDIOP1	GPIO	PC22	I/O	—	—	A	LCDDAT22	O	PIO, I, PU, ST
							B	FLEXCOM3_IO0	I/O	
K9	VDDIOP1	GPIO	PC23	I/O	—	—	A	LCDDAT23	O	PIO, I, PU, ST
							B	FLEXCOM3_IO1	I/O	
R8	VDDIOP1	GPIO	PC24	I/O	WKUP9	—	A	LCDDISP	O	PIO, I, PU, ST
							B	FLEXCOM3_IO4	O	
L9	VDDIOP1	GPIO	PC25	I/O	WKUP10	—	A	—	—	PIO, I, PU, ST
							B	FLEXCOM3_IO3	I/O	
T8	VDDIOP1	GPIO	PC26	I/O	—	—	A	LCDPWM	O	PIO, I, PU, ST
							B	FLEXCOM3_IO2	I/O	

© 2020 Microchip Technology Inc.

Downloaded from Arrow.com.

.....continued

228-pin BGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral			Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
M9	VDDIOP1	GPIO	PC27	I/O	—	—	A	LCDVSYNC	O	PIO, I, PU, ST
							B	E1_TXEN	O	
							C	FLEXCOM1_IO4	O	
N9	VDDIOP1	GPIO	PC28	I/O	—	—	A	LCDHSYNC	O	PIO, I, PU, ST
							B	E1_CRSDV	I	
							C	FLEXCOM1_IO3	I/O	
L10	VDDIOP1	GPIO	PC29	I/O	—	—	A	LCDDEN	O	PIO, I, PU, ST
							B	E1_TXCK	I/O	
							C	FLEXCOM1_IO2	I/O	
T7	VDDIOP1	GPIO	PC30	I/O	—	—	A	LCDPCK	O	PIO, I, PU, ST
							B	E1_MDC	O	
							C	FLEXCOM3_IO7	I	
M13	VDDIOP1	GPIO	PC31	I/O	WKUP11	—	A	FIQ	I	PIO, I, PU, ST
							B	E1_MDIO	I/O	
							C	PCK1	O	
R14	VDDNF	GPIO	PD0	I/O	—	—	A	NANDOE	O	PIO, I, PU, ST
T15	VDDNF	GPIO	PD1	I/O	—	—	A	NANDWE	O	PIO, I, PU, ST
P15	VDDNF	GPIO	PD2	I/O	—	—	A	A21/NANDALE	O	A21,O, PD, ST
N14	VDDNF	GPIO	PD3	I/O	—	—	A	A22/NANDCLE	O	A22,O, PD
R16	VDDNF	GPIO	PD4	I/O	—	—	A	NCS3/NANDCS	O	PIO, I, PU, ST
N11	VDDNF	GPIO	PD5	I/O	—	—	A	NWAIT	I	PIO, I, PU, ST
K16	VDDNF	GPIO	PD6	I/O	—	—	A	D16	I/O	PIO, I, PU, ST
J12	VDDNF	GPIO	PD7	I/O	—	—	A	D17	I/O	PIO, I, PU, ST
K15	VDDNF	GPIO	PD8	I/O	—	—	A	D18	I/O	PIO, I, PU, ST
J10	VDDNF	GPIO	PD9	I/O	—	—	A	D19	I/O	PIO, I, PU, ST
L16	VDDNF	GPIO	PD10	I/O	—	—	A	D20	I/O	PIO, I, PU, ST
K11	VDDNF	GPIO	PD11	I/O	—	—	A	D21	I/O	PIO, I, PU, ST
L15	VDDNF	GPIO	PD12	I/O	—	—	A	D22	I/O	PIO, I, PU, ST
J15	VDDNF	GPIO	PD13	I/O	—	—	A	D23	I/O	PIO, I, PU, ST

© 2020 Microchip Technology Inc.

Downloaded from Arrow.com.

.....continued

228-pin BGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral			Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
L12	VDDNF	GPIO	PD14	I/O	—	—	A	D24	I/O	PIO, I, PU, ST
M16	VDDNF	GPIO	PD15	I/O	—	—	A	D25	I/O	A20, O, PD
							B	A20	O	
M14	VDDNF	GPIO	PD16	I/O	—	—	A	D26	I/O	A23, O, PD
							B	A23	O	
N16	VDDNF	GPIO	PD17	I/O	WKUP12	—	A	D27	I/O	A24, O, PD
							B	A24	O	
L13	VDDNF	GPIO	PD18	I/O	WKUP13	—	A	D28	I/O	A25, O, PD
							B	A25	O	
P16	VDDNF	GPIO	PD19	I/O	—	—	A	D29	I/O	PIO, I, PU, ST
							B	NCS2	O	
M11	VDDNF	GPIO	PD20	I/O	—	—	A	D30	I/O	PIO, I, PU, ST
							B	NCS4	O	
M15	VDDNF	GPIO	PD21	I/O	—	—	A	D31	I/O	PIO, I, PU, ST
							B	NCS5	O	
H16	VDDIOM	DDRIO	D0	—	—	—	—	—	—	O, PD
H10	VDDIOM	DDRIO	D1	—	—	—	—	—	—	O, PD
H15	VDDIOM	DDRIO	D2	—	—	—	—	—	—	O, PD
H11	VDDIOM	DDRIO	D3	—	—	—	—	—	—	O, PD
J14	VDDIOM	DDRIO	D4	—	—	—	—	—	—	O, PD
J11	VDDIOM	DDRIO	D5	—	—	—	—	—	—	O, PD
J16	VDDIOM	DDRIO	D6	—	—	—	—	—	—	O, PD
J13	VDDIOM	DDRIO	D7	—	—	—	—	—	—	O, PD
G9	VDDIOM	DDRIO	D8	—	—	—	—	—	—	O, PD
D8	VDDIOM	DDRIO	D9	—	—	—	—	—	—	O, PD
F9	VDDIOM	DDRIO	D10	—	—	—	—	—	—	O, PD
A9	VDDIOM	DDRIO	D11	—	—	—	—	—	—	O, PD
F10	VDDIOM	DDRIO	D12	—	—	—	—	—	—	O, PD
B9	VDDIOM	DDRIO	D13	—	—	—	—	—	—	O, PD

Downloaded from Arrow.com.

.....continued

228-pin BGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral			Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
D11	VDDIOM	DDRIO	D14	—	—	—	—	—	—	O, PD
C9	VDDIOM	DDRIO	D15	—	—	—	—	—	—	O, PD
B10	VDDIOM	DDRIO	A0	—	NBS0	—	—	—	—	O, PD
G16	VDDIOM	DDRIO	A1	—	NBS2/DQM2/NWR2	—	—	—	—	O, PD
A13	VDDIOM	DDRIO	A2	—	—	—	—	—	—	O, PD
D15	VDDIOM	DDRIO	A3	—	—	—	—	—	—	O, PD
B12	VDDIOM	DDRIO	A4	—	—	—	—	—	—	O, PD
E11	VDDIOM	DDRIO	A5	—	—	—	—	—	—	O, PD
A12	VDDIOM	DDRIO	A6	—	—	—	—	—	—	O, PD
B16	VDDIOM	DDRIO	A7	—	—	—	—	—	—	O, PD
F16	VDDIOM	DDRIO	A8	—	—	—	—	—	—	O, PD
D14	VDDIOM	DDRIO	A9	—	—	—	—	—	—	O, PD
C11	VDDIOM	DDRIO	A10	—	—	—	—	—	—	O, PD
E13	VDDIOM	DDRIO	A11	—	—	—	—	—	—	O, PD
A11	VDDIOM	DDRIO	A12	—	—	—	—	—	—	O, PD
B11	VDDIOM	DDRIO	A13	—	—	—	—	—	—	O, PD
C15	VDDIOM	DDRIO	A14	—	—	—	—	—	—	O, PD
G15	VDDIOM	DDRIO	A15	—	—	—	—	—	—	O, PD
E14	VDDIOM	DDRIO	A16	—	BA0	—	—	—	—	O, PD
F14	VDDIOM	DDRIO	A17	—	BA1	—	—	—	—	O, PD
C16	VDDIOM	DDRIO	A18	—	BA2	—	—	—	—	O, PD
H14	VDDIOM	DDRIO	A19	—	—	—	—	—	—	O, PD
F15	VDDIOM	DDRIO	NCS0	—	—	—	—	—	—	O, PU
E16	VDDIOM	DDRIO	NCS1	—	SDCS	—	—	—	—	O, PU
F12	VDDIOM	DDRIO	NRD	—	—	—	—	—	—	O, PU
E8	VDDIOM	DDRIO	NWR0	—	NWE	—	—	—	—	O, PU
A10	VDDIOM	DDRIO	NWR1	—	NBS1	—	—	—	—	O, PU
L14	VDDIOM	GPIO	NWR3	—	NBS3/DQM3	—	—	—	—	O, PU
A15	VDDIOM	DDRIO	SDCK	—	—	—	—	—	—	O, PD

© 2020 Microchip Technology Inc.

Downloaded from Arrow.com.

.....continued

228-pin BGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral			Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
B14	VDDIOM	DDRIO	SDCKN	—	—	—	—	—	—	O, PU
F13	VDDIOM	DDRIO	SDCKE	—	—	—	—	—	—	O, PU
E15	VDDIOM	DDRIO	RAS	—	—	—	—	—	—	O, PU
C12	VDDIOM	DDRIO	CAS	—	—	—	—	—	—	O, PU
D16	VDDIOM	DDRIO	SDWE	—	—	—	—	—	—	O, PU
D12	VDDIOM	DDRIO	SDA10	—	—	—	—	—	—	O, PU
G11	VDDIOM	DDRIO	DQM0	—	—	—	—	—	—	O, PU
C8	VDDIOM	DDRIO	DQM1	—	—	—	—	—	—	O, PU
H12	VDDIOM	DDRIO	DQS0	—	—	—	—	—	—	O, PD
H13	VDDIOM	DDRIO	NDQS0	—	—	—	—	—	—	O, PU
D9	VDDIOM	DDRIO	DQS1	—	—	—	—	—	—	O, PD
E9	VDDIOM	DDRIO	NDQS1	—	—	—	—	—	—	O, PU
B8	VDDIOM	—	DDR_CAL	I/O	—	—	—	—	—	I
A14	VDDIOM	—	DDR_VREF	I/O	—	—	—	—	—	I
D5	VDDANA	—	ADVREFP	I	—	—	—	—	—	I
C5	VDDANA	—	ADVREFN	I	—	—	—	—	—	I
P11	VDDIN33	—	RTUNE	I/O	—	—	—	—	—	I
T12	VDDIN33	—	HHSDPA	I/O	DHSDP	—	—	—	—	O, PD
R12	VDDIN33	—	HHSDMA	I/O	DHSDM	—	—	—	—	O, PD
T13	VDDIN33	—	HHSDPB	I/O	—	—	—	—	—	O, PD
T14	VDDIN33	—	HHSDMB	I/O	—	—	—	—	—	O, PD
P12	VDDIN33	—	HHSDPC	I/O	—	—	—	—	—	O, PD
N12	VDDIN33	—	HHSDMC	I/O	—	—	—	—	—	O, PD
T11	VDDBU	—	WKUP0	I	—	—	—	—	—	I, ST
R11	VDDBU	—	SHDN	O	—	—	—	—	—	O, PD
P9	VDDBU	—	JTAGSEL	I	—	—	—	—	—	I, PD
R3	VDDIOP0	—	TCK	I	—	—	—	—	—	I, ST
F3	VDDIOP0	—	TDI	I	—	—	—	—	—	I, ST
H5	VDDIOP0	—	TDO	O	—	—	—	—	—	O

© 2020 Microchip Technology Inc.

Downloaded from Arrow.com.

.....continued

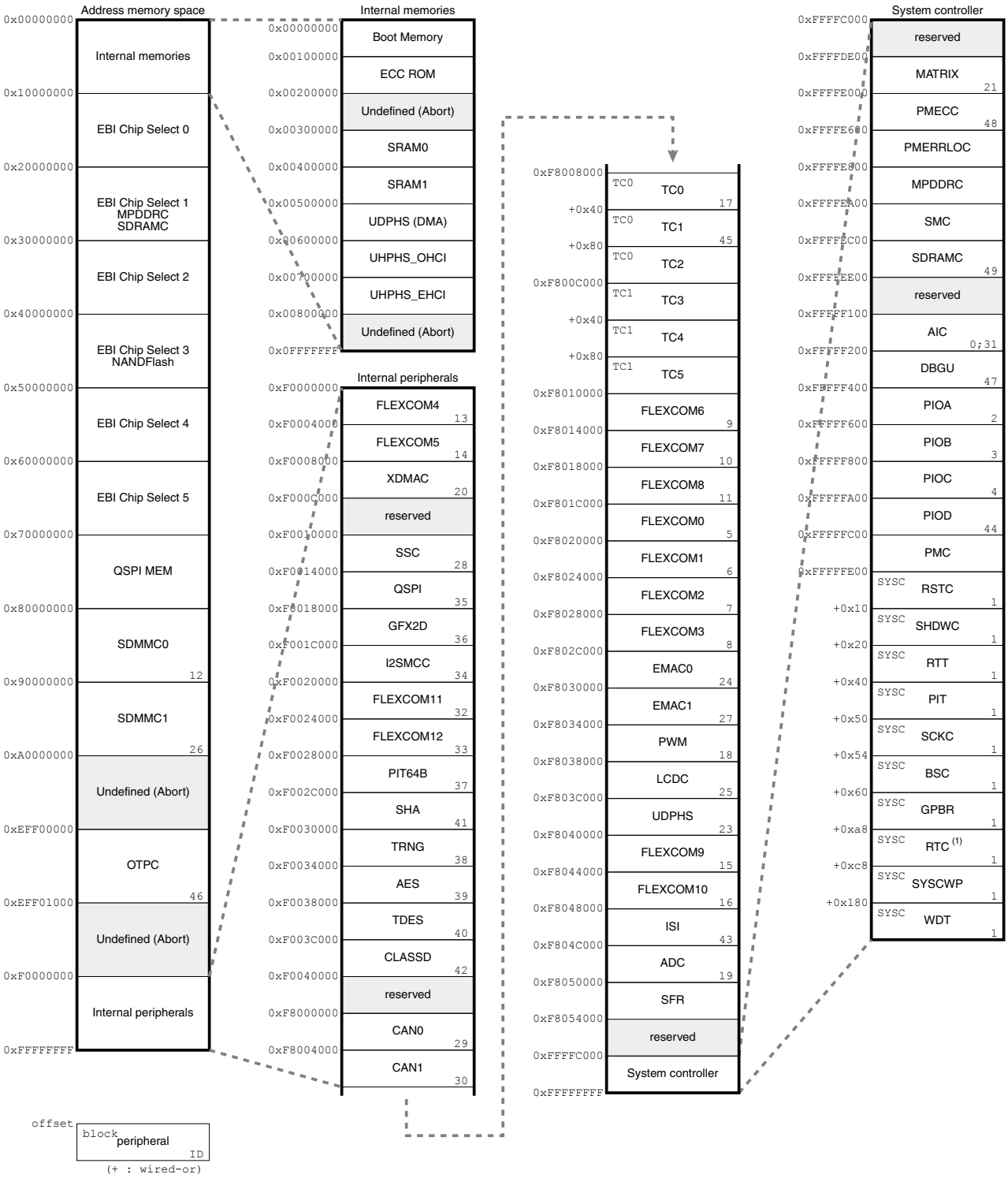
228-pin BGA	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral			Reset State
			Signal	Dir	Signal	Dir	Func	Signal	Dir	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
F5	VDDIOP0	—	TMS	I	—	—	—	—	—	I, ST
T2	VDDIOP0	—	RTCK	O	—	—	—	—	—	O
R1	VDDIOP0	—	NRST	I	—	—	—	—	—	I, PU, ST
T9	VDDBU	—	XIN32	I	—	—	—	—	—	I
R9	VDDBU	—	XOUT32	I/O	—	—	—	—	—	O
R10	VDDIN33	—	XIN	I	—	—	—	—	—	I
T10	VDDIN33	—	XOUT	I/O	—	—	—	—	—	O
C10, C13, G14	VDDIOM	power	—	—	—	—	—	—	—	—
A1, A16, B13, E7, E10, G5, G12, H8, H9, J8, J9, K5, K12, M7, N2, N15, T1, T16	GND	ground	—	—	—	—	—	—	—	—
K14	VDDNF	power	—	—	—	—	—	—	—	—
G3, K3	VDDIOP0	power	—	—	—	—	—	—	—	—
N3	VDDIOP1	power	—	—	—	—	—	—	—	—
P7	VDDBU	power	—	—	—	—	—	—	—	—
C4	VDDANA	power	—	—	—	—	—	—	—	—
B4	GNDANA	ground	—	—	—	—	—	—	—	—
P10	VDDOUT25	output	—	—	—	—	—	—	—	—
L11, P13	VDDIN33	power	—	—	—	—	—	—	—	—
M10, R13	GNDIN33	ground	—	—	—	—	—	—	—	—
F6, F11, L6	VDDCORE	power	—	—	—	—	—	—	—	—
C7	VDDQSPI	power	—	—	—	—	—	—	—	—

© 2020 Microchip Technology Inc.

Downloaded from Arrow.com.

6. Memories

Figure 6-1. Memory Mapping



(1) Refer to the table System Controller Peripheral Mapping in section 12. System Controller Write Protection (SYSCWP) for RTC detailed mapping.

Downloaded from Arrow.com.

6.1 Embedded Memories

6.1.1 Internal SRAM

The SAM9X60 embeds 68 Kbytes of high-speed SRAM, SRAM0. SRAM0 is always accessible at address 0x0030 0000. After remap, SRAM0 is also available at address 0x0000 0000. A 4-Kbyte SRAM memory, SRAM1, is used for OTP emulation and is always accessible at address 0x0040 0000.

6.1.2 Internal ROM

The ROM contains a bootloader program mapped at address 0 after reset and the BCH (Bose, Chaudhuri and Hocquenghem) code table mapped at address 0x0010_0000 for NAND Flash ECC correction.

6.1.3 Boot Strategies

For standard boot strategies, refer to 11. Boot Strategies.

For secure boot strategies, refer to the document “SAM9X60 Secure Boot Strategy”, document no. DS00003195 (Non-Disclosure Agreement required).

6.2 External Memory

The SAM9X60 offers connections to a wide range of external memories or to parallel peripherals.

6.2.1 External Bus Interface

The External Bus Interface (EBI) is an interface that features:

- Four external memory controllers:
  - a Static Memory Controller (SMC),
  - a NAND Flash Controller (NFC)
  - a Multi-Port DDR-SDRAM Controller (MPDDRC) to drive DDR2 and LPDDR
  - SDRAM controller to drive SDRAM and LPSDR devices
- 8-, 16-, or 32-bit data bus
- Up to 26-bit address bus
- Up to six chip selects with configurable assignment:
  - Static Memory Controller on NCS0, NCS1, NCS2, NCS3, NCS4, NCS5
  - MPDDRC/SDRAMC (SDCS) or Static Memory Controller on NCS1
  - Optional NAND Flash support on NCS3

The drive levels are configured in the EBI I/O Drive Configuration register (SFR_CCFG_EBICSA.EBI_DRIVE) in section 23. Special Function Registers (SFR). At reset, the selected drive is low. The user must make sure to program the correct drive. Refer to 57. Electrical Characteristics.

6.2.2 Supported Memories on MPDDRC/SDRAMC Interface

The MPDDRC and SDRAMC support the following memories:

- 4/2-bank SDR-SDRAM and LPSDR-SDRAM with 16 or 32-bit data path
- 8/4-bank DDR2-SDRAM and 4/2-bank LPDDR1-SDRAM with 16-bit data path
- 2K, 4K, 8K, 16K row address memory parts

6.2.3 Supported Memories on Static Memories and NAND Flash Interfaces

The SMC supports:

- Asynchronous SRAM-like memories and parallel peripherals
- 8-, 16- or 32-bit data bus

The NFC supports:

Downloaded from Arrow.com.

- 8-bit NAND Flash (SLC)
- Programmable Multi-bit Error Correcting Code (PMECC) based on BCH codes

6.2.4 DDR/SDR I/O Calibration and DDR Voltage Reference

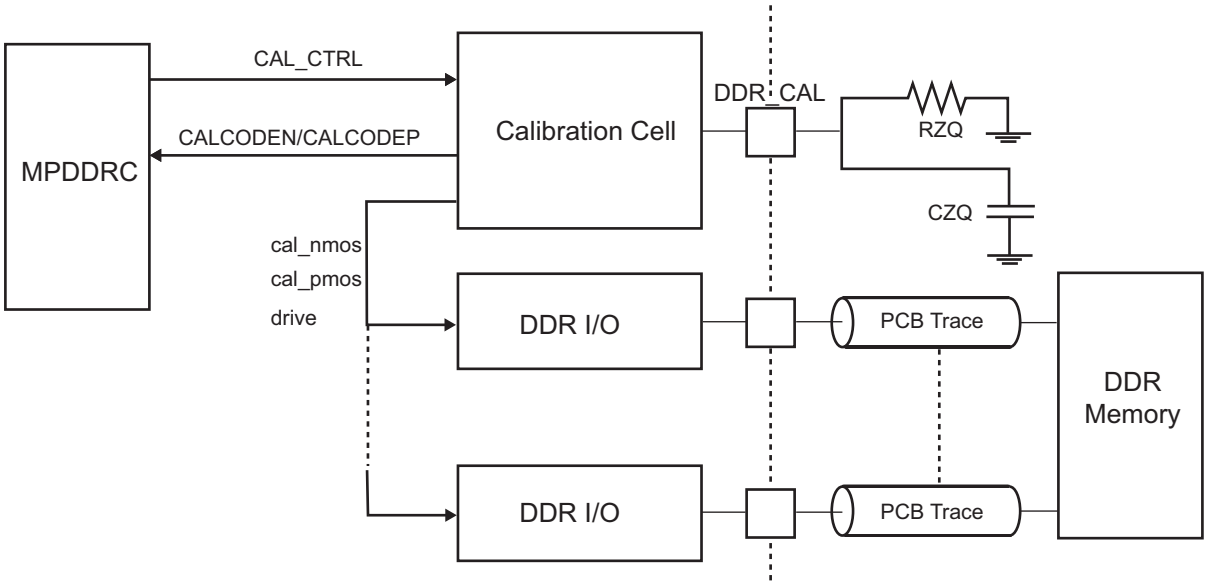
6.2.4.1 DDR/SDR I/O Calibration

The DDR/SDR I/Os embed an automatic impedance matching control to avoid overshoots and reach the best performance levels depending on the bus load and external memories. A serial termination connection scheme, where the driver has an output impedance matched to the characteristic impedance of the line, is used to improve signal quality and reduce EMI.

One specific analog input, DDR_CAL, is used to calibrate all DDR/SDR I/Os.

The MPDDRC and SDRAMC support the ZQ calibration procedure used to calibrate the SAM9X60 DDR/SDR I/O drive strength and the commands to set up the external memory device drive strength (refer to 31. AHB Multiport DDR-SDRAM Controller (MPDDRC)). The calibration cell supports all the supported memory types. Calibration is performed in the initialization phase only.

Figure 6-2. DDR Calibration Cell



The calibration cell provides an input pin, DDR_CAL, loaded with one of the following resistor RZQ values:

- 20 KΩ for LPDDR
- 20 KΩ for DDR2
- 16.9 KΩ for SDRAM
- 20 KΩ for LPSDRAM

The typical value for CZQ is 22 pF.

6.2.4.2 DDR_VREF Recommended Circuits

The DDR_VREF pin serves as a voltage reference input for the DDR I/Os when DDR2 or LPDDR external SDRAM memories are used. This pin is not used with SDR or LPSDR SDRAM memories, and must therefore be connected to ground.

The following figures give the recommended schematics for each case: DDR2, LPDDR and SDR/LPSDR.

Downloaded from Arrow.com.

Figure 6-3. DDR_VREF Recommended Schematic with DDR2-SDRAM

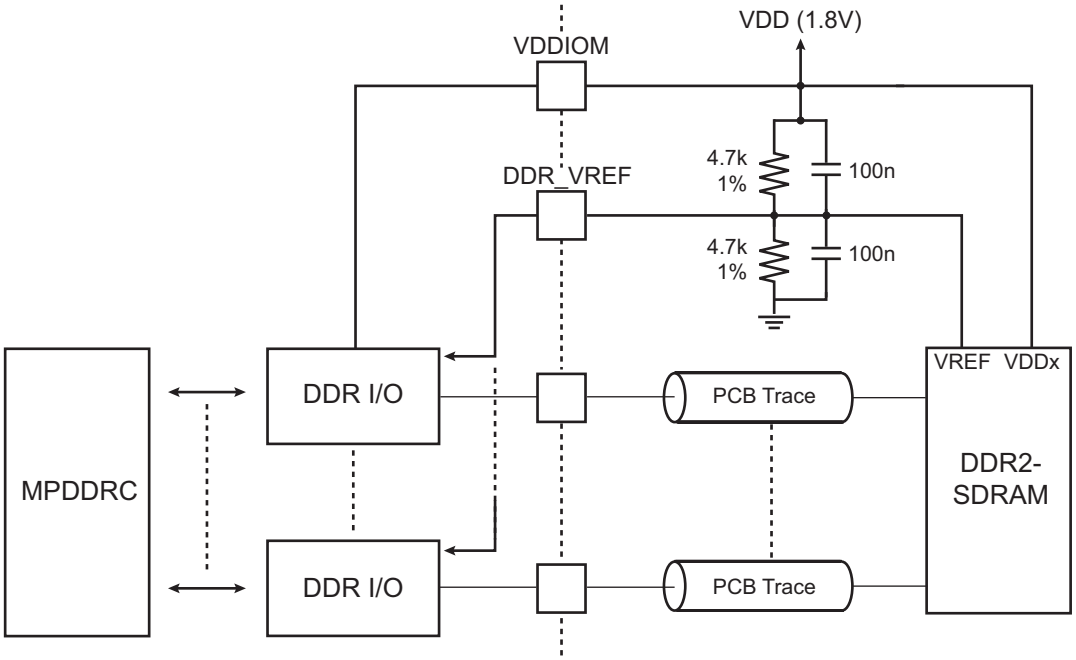
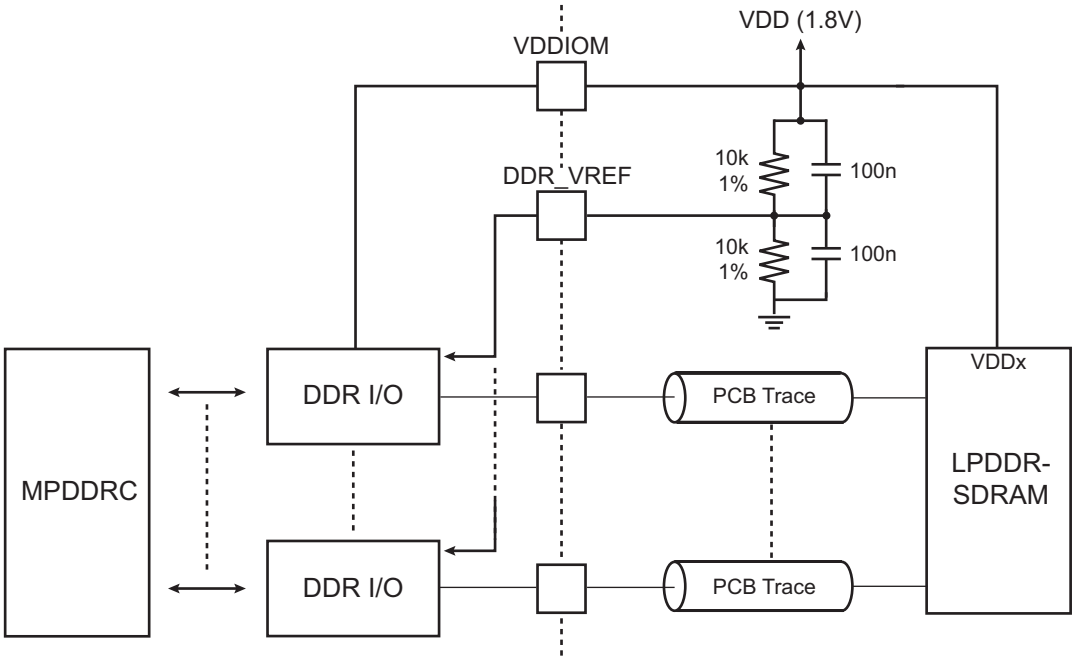
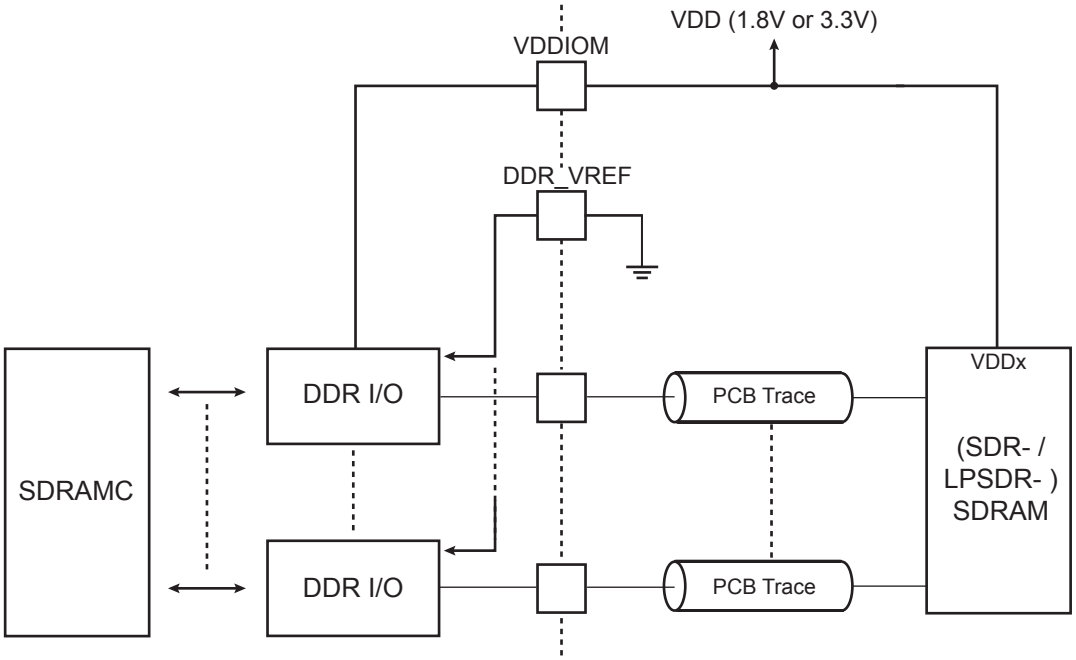


Figure 6-4. DDR_VREF Recommended Schematic with LPDDR-SDRAM



Downloaded from Arrow.com.

Figure 6-5. DDR_VREF Recommended Schematic with (LP)SDR-SDRAM



Downloaded from Arrow.com.

7. System Controller

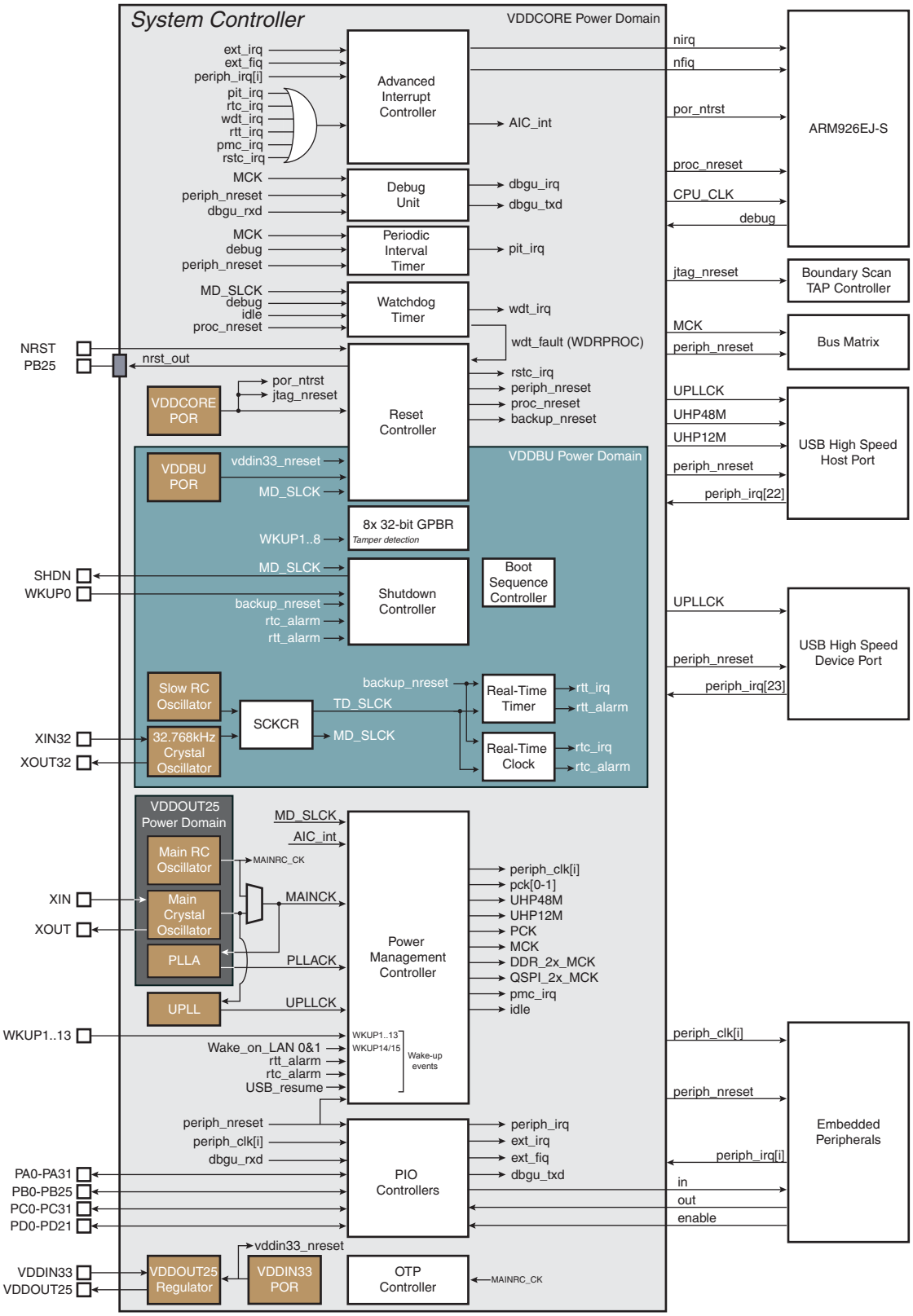
The System Controller is a set of peripherals that allows handling of key elements of the system, such as power, resets, clocks, time, interrupts, watchdog, etc.

The System Controller's peripherals are all mapped within the highest 16 Kbytes of address space, between addresses 0xFFFF_C000 and 0xFFFF_FFFF.

The following figure shows the System Controller block diagram.

Downloaded from Arrow.com.

Figure 7-1. System Controller Block Diagram



Downloaded from Arrow.com.

7.1 Power-On Reset

The SAM9X60 embeds three Power-On Resets (PORs) cells on VDDBU, VDDCORE and VDDIN33. Refer to 57. Electrical Characteristics for associated thresholds and timing information.

Downloaded from Arrow.com.

8. Peripherals

8.1 Peripheral Mapping

As shown in Figure 6-1, the peripherals are mapped in the upper 256 Mbytes of the address space, between addresses 0xF000 0000 and 0xFFFC 0000.

8.2 Peripheral Identifiers

Table 8-1. Peripheral Identifiers

Instance ID	Instance Name	Internal Interrupt	External Interrupt	PMC Clock Control	Generic Clock	f_{GCLK} (Max)	PLLACLK	UPLLCLK	Instance Description
0	AIC	–	EXT_FIQ	–	–	–	–	–	Advanced Interrupt Controller
1	SYSC	X	–	–	–	–	–	–	Logical-OR interrupt of SYSC, PMC, WDT, PIT, RSTC, RTT, RTC
2	PIOA	X	–	X	–	–	–	–	Parallel I/O Controller A
3	PIOB	X	–	X	–	–	–	–	Parallel I/O Controller B
4	PIOC	X	–	X	–	–	–	–	Parallel I/O Controller C
5	FLEXCOM0	X	–	X	X	f_{MCK}/ 3	X	X	FLEXCOM 0
6	FLEXCOM1	X	–	X	X	f_{MCK}/ 3	X	X	FLEXCOM 1
7	FLEXCOM2	X	–	X	X	f_{MCK}/ 3	X	X	FLEXCOM 2
8	FLEXCOM3	X	–	X	X	f_{MCK}/ 3	X	X	FLEXCOM 3
9	FLEXCOM6	X	–	X	X	f_{MCK}/ 3	X	X	FLEXCOM 6
10	FLEXCOM7	X	–	X	X	f_{MCK}/ 3	X	X	FLEXCOM 7
11	FLEXCOM8	X	–	X	X	f_{MCK}/ 3	X	X	FLEXCOM 8
12	SDMMC0	X	–	X	X	105 MHz	X	X	Secure Data Memory Card Controller 0
13	FLEXCOM4	X	–	X	X	f_{MCK}/ 3	X	X	FLEXCOM 4
14	FLEXCOM5	X	–	X	X	f_{MCK}/ 3	X	X	FLEXCOM 5
15	FLEXCOM9	X	–	X	X	f_{MCK}/ 3	X	X	FLEXCOM 9
16	FLEXCOM10	X	–	X	X	f_{MCK}/ 3	X	X	FLEXCOM 10
17	TC0	X	–	X	X	f_{MCK}/ 3	X	X	Timer Counters 0,1,2
18	PWM	X	–	X	–	–	–	–	Pulse Width Modulation Controller
19	ADC	X	–	X	X	f_{MCK}/ 3	X	X	ADC Controller
20	XDMAC	X	–	X	–	–	–	–	Extended DMA Controller
21	MATRIX	X	–	–	–	–	–	–	Matrix

Downloaded from Arrow.com.

.....continued

Instance ID	Instance Name	Internal Interrupt	External Interrupt	PMC Clock Control	Generic Clock	f_{GCLK} (Max)	PLLACLK	UPLLCLK	Instance Description
22	UHPHS	X	–	X	–	–	–	–	USB Host High Speed
23	UDPHS	X	–	X	–	–	–	–	USB Device High Speed
24	EMAC0	X	–	X	–	–	–	–	Ethernet MAC 0
25	LCDC	X	–	X	X	140 MHz	X	X	LCD Controller
26	SDMMC1	X	–	X	X	105 MHz	X	X	Secure Data Memory Card Controller 1
27	EMAC1	X	–	X	–	–	–	–	Ethernet MAC 1
28	SSC	X	–	X	–	–	–	–	Synchronous Serial Controller
29	CAN0	X	–	X	–	–	–	–	CAN Controller 0
30	CAN1	X	–	X	–	–	–	–	CAN Controller 1
31	AIC	–	EXT_IRQ		–	–	–	–	Advanced Interrupt Controller
32	FLEXCOM11	X	–	X	X	f_{MCK}/ 3	X	X	FLEXCOM 11
33	FLEXCOM12	X	–	X	X	f_{MCK}/ 3	X	X	FLEXCOM 12
34	I2SMCC	X	–	X	X	105 MHz	X	X	I2S Multi Channel Controller
35	QSPI	X	–	X	–	–	–	–	Quad I/O SPI Controller
36	GFX2D	X	–	X	–	–	–	–	2D Graphics Controller
37	PIT64B	X	–	X	X	f_{MCK}/ 3	X	X	64-bit Timer
38	TRNG	X	–	X	–	–	–	–	True Random Number Generator
39	AES	X	–	X	–	–	–	–	Advanced Encryption Standard
40	TDES	X	–	X	–	–	–	–	Triple Data Encryption Standard
41	SHA	X	–	X	–	–	–	–	Secure Hash Algorithm
42	CLASSD	X	–	X	X	100 MHz	X	X	CLASS D Controller
43	ISI	X	–	X	–	–	–	–	Image Sensor Interface
44	PIOD	X	–	X	–	–	–	–	Parallel I/O Controller D
45	TC1	X	–	X	X	f_{MCK}/ 3	X	X	Timer Counter 3, 4, 5
46	OTPC	X	–	–	–	–	–	–	OTP Controller
47	DBGU	X	–	X	X	f_{MCK}/ 3	X	X	Debug Unit
48	PMECC	X	–	–	–	–	–	–	logical-OR interrupt of PMECC and PMERRLOC

Downloaded from Arrow.com.

.....continued

Instance ID	Instance Name	Internal Interrupt	External Interrupt	PMC Clock Control	Generic Clock	f_{GCLK} (Max)	PLLACLK	UPLLCLK	Instance Description
49	SDRAMC/ MPDDRC	X	–	X	–	–	–	–	logical-OR interrupt of SDRAMC, MPDDRC and HSMC
50	UTMI	–	–	–	–	–	–	–	UTMI Controller

8.3 FLEXCOM Features

Table 8-2. FLEXCOM Features

Functions and Features	FLEXCOM Instance												
	0	1	2	3	4	5	6	7	8	9	10	11	12
TWI Function	X	X	X	X	X	X	X	X	X	X	X	X	X
Normal/fast (400 kbit/s) / FM+ (1 Mbit/s)	X	X	X	X	X	X	X	X	X	X	X	X	X
Alternate command	X	X	X	X	X	X	X	X	X	X	X	X	X
Three-slave ADDR	X	X	X	X	X	X	X	X	X	X	X	X	X
High speed (3.4 Mbit/s)	X	X	X	X	X	X	X	X	X	X	X	X	X
Sniffer	X	X	X	X	X	X	X	X	X	X	X	X	X
TWI FIFO size	16 bytes												
UART / USART Function	X	X	X	X	X	X	X	X	X	X	X	X	X
Two-wire UART	X	X	X	X	X	X	X	X	X	X	X	X	X
Five-wire USART	X	X	X	X	X	X	–	–	–	–	–	–	–
Hardware handshaking/RS485	X	X	X	X	X	X	–	–	–	–	–	–	–
ISO7816	X	X	X	X	X	X	–	–	–	–	–	–	–
IrDA	X	X	X	X	X	X	–	–	–	–	–	–	–
LIN	X	X	X	X	X	X	–	–	–	–	–	–	–
LON	X	X	X	X	–	–	–	–	–	–	–	–	–
Manchester	X	X	X	X	X	X	–	–	–	–	–	–	–
USART FIFO Size	16 bytes												
SPI Function	X	X	X	X	X	X	–	–	–	–	–	–	–
1CS	X	X	X	X	X	X	–	–	–	–	–	–	–
4CS	–	–	–	–	X	X	–	–	–	–	–	–	–
SPI FIFO size	16 bytes						–	–	–	–	–	–	–

8.4 Peripheral Signal Multiplexing on I/O Lines

The SAM9X60 features several PIO controllers that multiplex the I/O lines of the peripheral set.

Table 5-2 defines how the I/O lines are multiplexed on the different PIO Controllers.

Downloaded from Arrow.com.

The column “Reset State” shows whether the PIO line resets in I/O mode or in Peripheral mode. If I/O is shown, the PIO line resets in input with the pull-up enabled, so that the device is maintained in a static state as soon as the reset is released. As a result, the bit corresponding to the PIO line in register PIO_CFGR (PIO Configuration Register) resets low.

If a signal name is shown in the “Reset State” column, the PIO line is assigned to this function and the corresponding bit in PIO_CFGR resets high. That is the case for pins controlling memories, in particular address lines, which require the pin to be driven as soon as the reset is released.

Downloaded from Arrow.com.

9. ARM926EJ-S Processor

The ARM926EJ-S processor is a member of the ARM9™ family of general-purpose microprocessors. The ARM926EJ-S implements ARM architecture version 5TEJ and is targeted at multi-tasking applications where full memory management, high performance, low die size and low power are important features.

The ARM926EJ-S processor supports the 32-bit ARM and 16-bit THUMB instruction sets, enabling the user to trade off between high performance and high code density. It also supports the 8-bit Java instruction set and includes features for efficient execution of Java bytecode, providing a Java performance similar to JITs (Just-In-Time compilers), for the next generation of Java-powered wireless and embedded devices. It includes an enhanced multiplier design for improved DSP performance.

The ARM926EJ-S processor supports the ARM debug architecture and includes logic to assist in both hardware and software debug.

The ARM926EJ-S provides a complete high performance processor subsystem, including:

- an ARM9EJ-S™ integer core,
- a Memory Management Unit (MMU),
- separate instruction and data AMBA AHB bus interfaces,
- a 32-Kbyte L1 instruction cache and a 32-Kbyte data cache.

For information on the ARM926EJ-S processor, refer to *ARM926EJ-S™ Technical Reference Manual* on www.arm.com.

Downloaded from Arrow.com.

10. Debug and Test

10.1 Description

The product features a number of complementary debug and test capabilities. A common JTAG/ICE (In-Circuit Emulator) port is used for standard debugging functions, such as downloading code and single-stepping through programs. The Debug Unit provides a two-pin UART that can be used to upload an application into internal SRAM. It manages the interrupt handling of the internal COMMTX and COMMRX signals that trace the activity of the Debug Communication Channel.

A set of dedicated debug and test input/output pins gives direct access to these capabilities from a PC-based test environment.

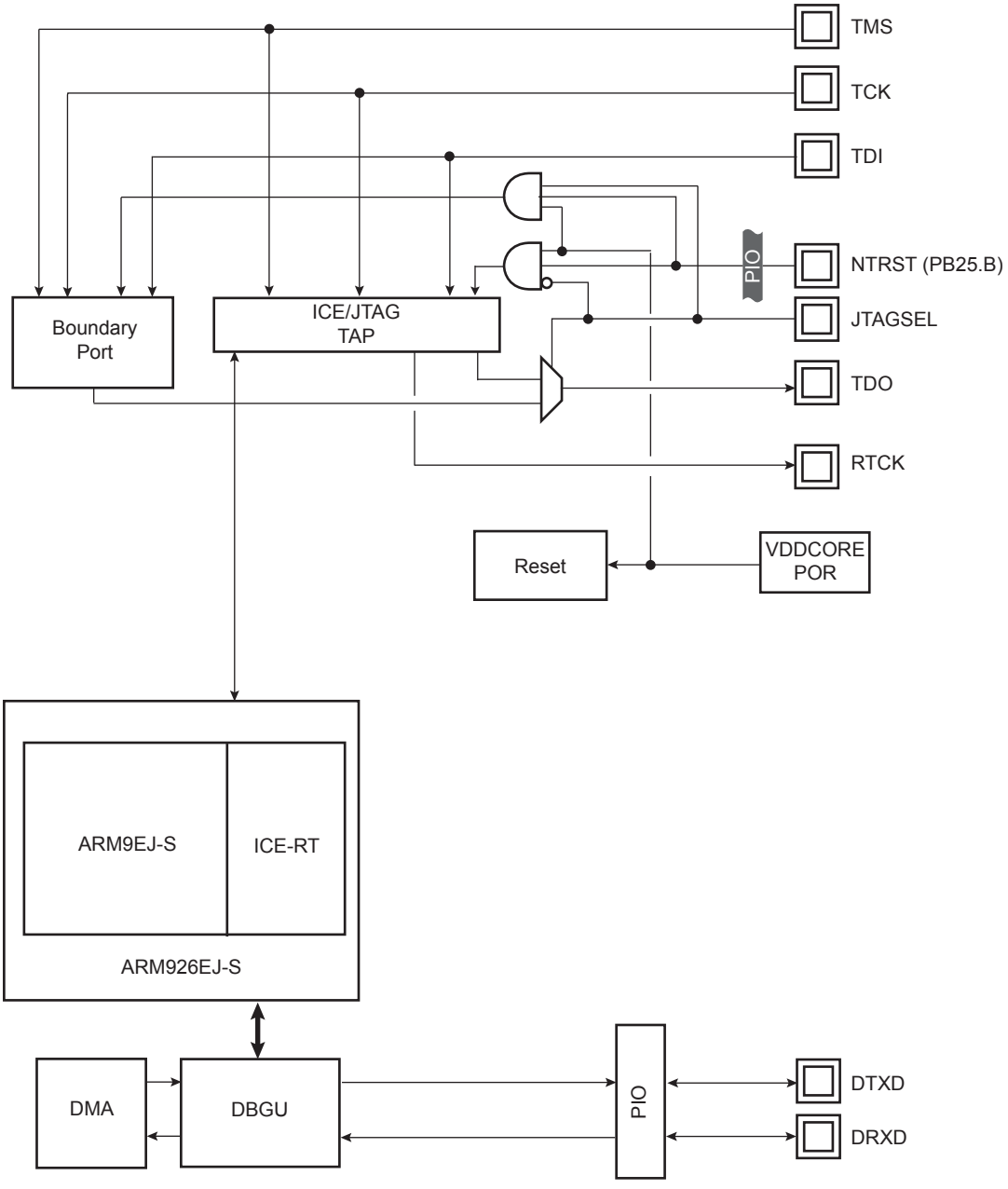
10.2 Embedded Characteristics

- ARM926 Real-time In-circuit Emulator
  - Two real-time watchpoint units
  - Two independent registers: Debug Control register and Debug Status register
  - Test access port accessible through JTAG protocol
  - Debug communications channel
- Debug Unit
  - Two-pin UART
  - Debug communication channel interrupt handling
  - Chip ID register
- IEEE® 1149.1 JTAG Boundary-scan on All Digital Pins

Downloaded from Arrow.com.

10.3 Block Diagram

Figure 10-1. Debug and Test Block Diagram



TAP: Test Access Port

10.4 Application Examples

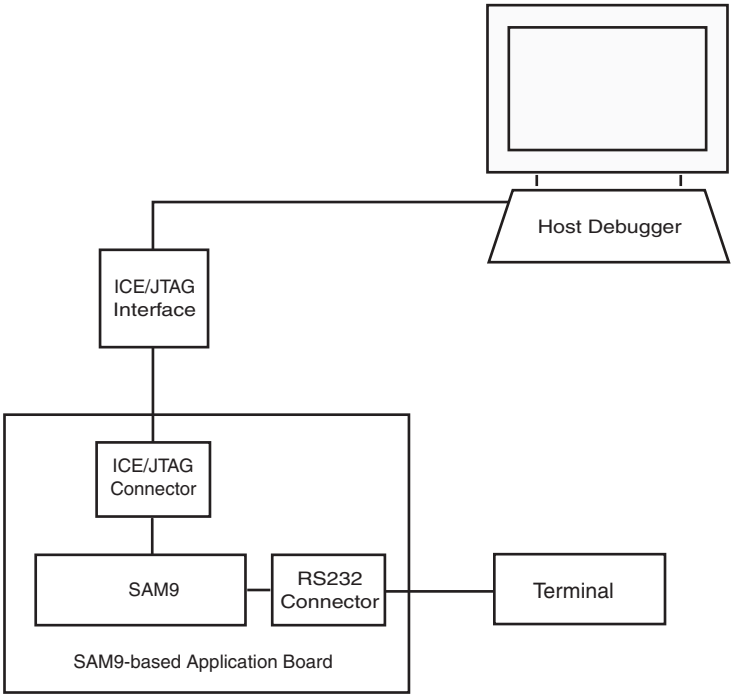
10.4.1 Debug Environment

The following figure shows a complete debug environment example. The ICE/JTAG interface is used for standard debugging functions, such as downloading code and single-stepping through the program. A software debugger

Downloaded from Arrow.com.

running on a personal computer provides the user interface for configuring a Trace Port interface utilizing the ICE/JTAG interface.

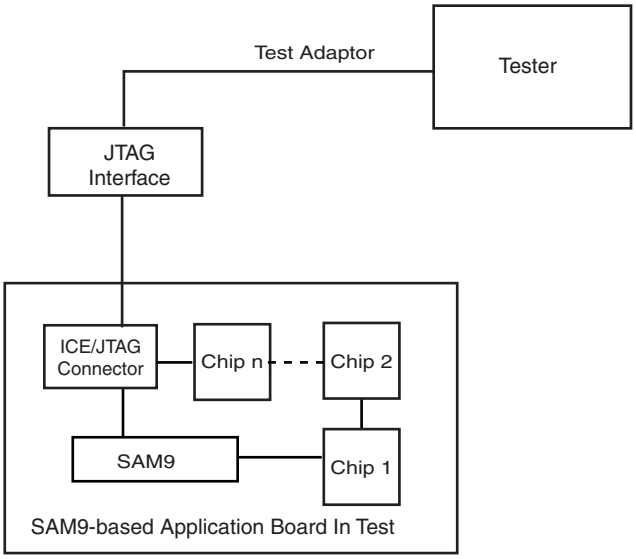
Figure 10-2. Application Debug and Trace Environment Example



10.4.2 Test Environment

The following figure shows a test environment example. Test vectors are sent and interpreted by the tester. In this example, the “board in test” is designed using a number of JTAG-compliant devices. These devices can be connected to form a single scan chain.

Figure 10-3. Application Test Environment Example



Downloaded from Arrow.com.

10.5 Debug and Test Pin Description

Table 10-1. Debug and Test Pin List

Pin Name	Function	Type	Active Level
Reset/Test			
NRST	Microcontroller Reset	Input	Low
ICE and JTAG			
NTRST	Test Reset Signal	Input	Low
TCK	Test Clock	Input	–
TDI	Test Data In	Input	–
TDO	Test Data Out	Output	–
TMS	Test Mode Select	Input	–
RTCK	Returned Test Clock	Output	–
JTAGSEL	JTAG Selection	Input	–
Debug Unit			
DRXD	Debug Receive Data	Input	–
DTXD	Debug Transmit Data	Output	–

10.6 Functional Description

10.6.1 EmbeddedICE™

The Arm 9EJ-S EmbeddedICE-RT™ is supported via the ICE/JTAG port. It is connected to a host computer via an ICE interface. Debug support is implemented using an ARM9EJ-S core embedded within the ARM926EJ-S. The internal state of the ARM926EJ-S is examined through an ICE/JTAG port which allows instructions to be serially inserted into the pipeline of the core without using the external data bus. Therefore, when in debug state, a store-multiple (STM) can be inserted into the instruction pipeline. This exports the contents of the ARM9EJ-S registers. This data can be serially shifted out without affecting the rest of the system.

There are two scan chains inside the ARM9EJ-S processor which support testing, debugging, and programming of the EmbeddedICE-RT. The scan chains are controlled by the ICE/JTAG port.

EmbeddedICE mode is selected when JTAGSEL is low. It is not possible to switch directly between ICE and JTAG operations. A chip reset must be performed after JTAGSEL is changed.

For further details on the EmbeddedICE-RT, refer to the Arm document ARM9EJ-S Technical Reference Manual (DDI 0222A).

10.6.2 JTAG Signal Description

TMS is the Test Mode Select input which controls the transitions of the test interface state machine.

TDI is the Test Data Input line which supplies the data to the JTAG registers (Boundary Scan Register, Instruction Register, or other data registers).

TDO is the Test Data Output line which is used to serially output the data from the JTAG registers to the equipment controlling the test. It carries the sampled values from the boundary scan chain (or other JTAG registers) and propagates them to the next chip in the serial test circuit.

NTRST (optional in IEEE Standard 1149.1) is a Test-ReSeT input which is mandatory in Arm cores and used to reset the debug logic. On Microchip ARM926EJ-S-based cores, NTRST is a Power On Reset output. It is asserted on

Downloaded from Arrow.com.

power on. If necessary, the user can also reset the debug logic with the NTRST pin assertion during 2.5 MCK periods.

TCK is the Test ClocK input which enables the test interface. TCK is pulsed by the equipment controlling the test and not by the tested device. It can be pulsed at any frequency. Note the maximum JTAG clock rate on ARM926EJ-S cores is 1/6th the clock of the CPU. This gives 5.45 kHz maximum initial JTAG clock rate for an ARM9E running from the 32.768 kHz slow clock.

RTCK is the Return Test Clock. Not an IEEE Standard 1149.1 signal added for a better clock handling by emulators. From some ICE Interface probes, this return signal can be used to synchronize the TCK clock and take not care about the given ratio between the ICE Interface clock and system clock equal to 1/6th. This signal is only available in JTAG ICE Mode and not in boundary scan mode.

10.6.3 Debug Unit

The Debug Unit manages the interrupt handling of the COMMTX and COMMRX signals that come from the ICE and that trace the activity of the Debug Communication Channel. The Debug Unit allows blockage of access to the system through the ICE interface.

For further details on the Debug Unit, refer to 21. Debug Unit (DBGU).

10.6.4 IEEE 1149.1 JTAG Boundary Scan

IEEE 1149.1 JTAG Boundary Scan allows pin-level access independent of the device packaging technology.

IEEE 1149.1 JTAG Boundary Scan is enabled when JTAGSEL is high. The SAMPLE, EXTEST and BYPASS functions are implemented. In ICE debug mode, the Arm processor responds with a non-JTAG chip ID that identifies the processor to the ICE system. This is not IEEE 1149.1 JTAG-compliant.

It is not possible to switch directly between JTAG and ICE operations. A chip reset must be performed after JTAGSEL is changed.

A Boundary-scan Descriptor Language (BSDL) file is provided to set up test.

Downloaded from Arrow.com.

10.6.5 JTAG ID Code Register

Access: Read-only

Bit	31	30	29	28	27	26	25	24
	VERSION				PART NUMBER			
Bit	23	22	21	20	19	18	17	16
	PART NUMBER							
Bit	15	14	13	12	11	10	9	8
	PART NUMBER				MANUFACTURER IDENTITY			
Bit	7	6	5	4	3	2	1	0
	MANUFACTURER IDENTITY							

- **VERSION[31:28]: Product Version Number**

Set to 0x0.

- **PART NUMBER[27:12]: Product Part Number**

Product part number is 0x5B2F

- **MANUFACTURER IDENTITY[11:1]**

Set to 0x01F.

Bit[0] required by IEEE Std. 1149.1.

Set to 0x1.

JTAG ID Code value is 0x05B2_F03F.

Downloaded from Arrow.com.

11. Boot Strategies

11.1 Description

The system always boots from the ROM memory at address 0x0.

The ROM code is a boot program contained in the embedded ROM. It is also called “Boot loader”.

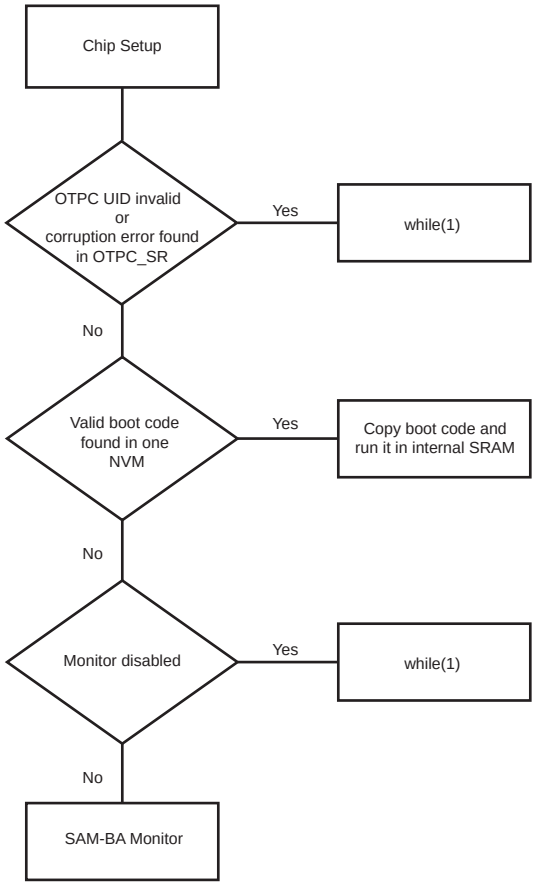
By default, the chip starts in a Standard Boot mode. To know how the Secure Boot mode can be enabled, refer to the document “SAM9X60 Secure Boot Strategy”, document no. DS00003195. To obtain this application note and additional information about the secure boot and related tools, contact a Microchip sales representative.

Note: JTAG access is disabled during execution of the ROM code sequence. It is re-enabled when jumping into SRAM when a valid code has been found on an external Non-Volatile Memory (NVM) at the same time the ROM memory is hidden. If no valid boot has been found on an external NVM, the ROM code enables the USB connection and one UART serial port, starts the standard SAM-BA® monitor, locks access to the ROM memory and re-enables the JTAG connection.

11.2 Flow Diagram

The following figure shows the ROM code global flow.

Figure 11-1. ROM Code Flow Diagram



Downloaded from Arrow.com.

11.3 Chip Setup

When the chip is powered on, the processor clock (CPU_CLK) and the master clock (MCK) source is the Main clock (MAINCK).

The ROM code performs a low-level initialization that follows the steps described below:

1. PLLA initialized at a frequency of 396 MHz.
2. Master clock selection: when the PLLA is stabilized, the master clock source is switched from the main clock to the PLLA clock. The PMC Status Register is polled to wait for MCK Ready. Now, the CPU_CLK frequency is the same as the PLLA clock, whereas the MCK frequency is the quarter of the PLLA clock.

For clock frequencies, see Table 11-4.

Note: No external crystal or clock is needed during the external boot memories sequence. An external clock source is checked before the launch of the SAM-BA monitor to get a more accurate clock signal for USB.

11.4 Boot Configuration

The boot sequence is controlled using Boot Configuration Packet, stored in the OTP area and configured through the OTP Controller (OTPC).

11.4.1 Default Boot Sequence (Without Boot Configuration Packet)

When no Boot Configuration Packet is available in the OTP area, the ROM code uses an internal default Boot Configuration Packet to configure the DBGU as a console and tries to boot from one of the following memories:

- SDMMC0 IOSET0
- SDMMC1 IOSET0
- QSPI0 IOSET0
- SPI0 IOSET0
- NAND0 IOSET0

Refer to Table 11-5 for further details.

If no bootable file is found in these memories, the ROM code goes to the monitor.

11.4.2 Using Boot Configuration Packet

The boot configuration data are stored in the Boot Configuration Packet in the OTP area. These data can be used for various boot sequence customizations:

- Set the IO pin configuration where the external memories used to boot are connected (see 11.4.8 Hardware and Software Constraints for a description of the IO).
- Enable the boot from selected memories.
- Configure the UART port used as a console.
- Enable/disable JTAG used for debug.

See 11.4.4 Boot Configuration User Interface for a detailed description of all the fields in these data.

By default, the values of the Boot Configuration Packet are 0x0. During prototyping phases, those values can be overridden by the content of the emulation memory through OTPC Emulation mode. To enable this feature, the bit BSCR_CR.EMUL_EN must be set.

The current running mode of the User area can be observed by reading BSCR_CR.EMUL_EN. If this bit is set, the Emulation mode is enabled, otherwise, the Emulation mode is disabled.

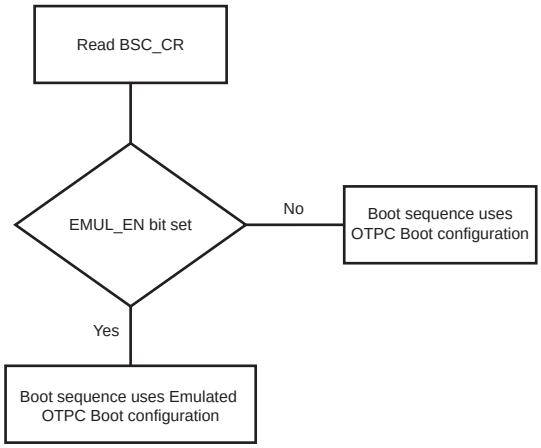
After a reset, the ROM code reads the Boot Configuration Packet from the SRAM dedicated to Emulation mode if the bit BSCR_CR.EMUL_EN is set to 1. Otherwise the packet is read from the OTP matrix.

Using Emulated OTP enables the user to test several boot configuration options, including Secure Boot mode, without programming the OTP.

Note: If Emulation mode is enabled, the emulation SRAM is not backed up. After a power off/on, the configuration and content are lost.

Downloaded from Arrow.com.

Figure 11-2. Boot Configuration Loading



Downloaded from Arrow.com.

11.4.3 Boot Sequence Controller Configuration Register

Name: BSC_CR

Address: 0xFFFFFE54

Bit	31	30	29	28	27	26	25	24
	WPKEY[15:8]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPKEY[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								EMUL_EN
Access								
Reset								

Bits 31:16 – WPKEY[15:0] Write Protect Key

Value	Name	Description
0x6683	PASSWD	Writing any other value in this field aborts the write operation of the BOOT field. Always reads as 0.

Bit 0 – EMUL_EN Emulation Enable

Value	Description
0	Emulation mode is disabled.
1	The OTP user area is emulated in internal SRAM1.

11.4.4 Boot Configuration User Interface

Table 11-1. Boot Configuration Packet Mapping

Offset	Name	Description
0x00	MON_DIS	Disables monitor
0x04	ZERO	Must be filled with zero
0x08	RESERVED	Reserved
0x0C	JTAG_DIS	Disables JTAG
0x10	CONSOLE_PIN	Console pin muxing
0x14-0x48	MEM_CFGx[2]	Two 32-bit words used to set memory configurations for x=0..6 MEM_CFGx[0]: this word contains mandatory options for all connected memories. MEM_CFGx[1]: this word contains mandatory options for specific memories.

Downloaded from Arrow.com.

11.4.4.1 Monitor Disable

Name: MON_DIS

Bit	31	30	29	28	27	26	25	24
	DISABLE_MONITOR[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DISABLE_MONITOR[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DISABLE_MONITOR[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DISABLE_MONITOR[7:0]							
Access								
Reset								

Bits 31:0 – DISABLE_MONITOR[31:0] SAM-BA Monitor Disable

Value	Description
0	If no boot file is found, launches the SAM-BA monitor.
Non-Zero	The SAM-BA monitor is never launched.

Downloaded from Arrow.com.

11.4.4.2 Boot Configuration Word Disable JTAG

Name: JTAG_DIS

Bit	31	30	29	28	27	26	25	24
	DISABLE_JTAG[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DISABLE_JTAG[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DISABLE_JTAG[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DISABLE_JTAG[7:0]							
Access								
Reset								

Bits 31:0 – DISABLE_JTAG[31:0] JTAG Disable

Value	Description
0	Enables JTAG.
Non-Zero	Disables JTAG.

Downloaded from Arrow.com.

11.4.4.3 Console Pin Muxing

Name: CONSOLE_PIN

WARNING To avoid any malfunctioning, the user must not write the "DO NOT USE (DNU)" bits.

Bit	31	30	29	28	27	26	25	24
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DNU	DNU	DNU	DNU	CONSOLE_IOSET[3:0]			
Access								
Reset								

Bits 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – DNU DO NOT USE

Bits 3:0 – CONSOLE_IOSET[3:0] Selects the pins and UART interface used as a console terminal

Value	Name	Description
0	DBGU	Uses DBGU
1	UART0	Uses FLEXCOM0 UART pins
2	UART1	Uses FLEXCOM1 UART pins
3	UART2	Uses FLEXCOM2 UART pins
4	UART3	Uses FLEXCOM3 UART pins
5	UART4	Uses FLEXCOM4 UART pins
6	UART5	Uses FLEXCOM5 UART pins
7	UART6	Uses FLEXCOM6 UART pins
8	UART7	Uses FLEXCOM7 UART pins
9	UART8	Uses FLEXCOM8 UART pins
10	UART9	Uses FLEXCOM9 UART pins
11	UART10	Uses FLEXCOM10 UART pins
12	UART11	Uses FLEXCOM11 UART pins
13	UART12	Uses FLEXCOM12 UART pins

Downloaded from Arrow.com.

11.4.4.4 QSPI Memory Configuration Data (First Word)

Name: MEM_CFGx[0]

WARNING To avoid any malfunctioning, the user must not write the "DO NOT USE (DNU)" bits.

Bit	31	30	29	28	27	26	25	24
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DNU	DNU	DNU	DNU	IFACE_TYPE[3:0]			
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	INSTANCE_ID[3:0]				IFACE_IOSET[3:0]			
Access								
Reset								

Bits 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – DNU DO NOT USE

Bits 11:8 – IFACE_TYPE[3:0] Interface Type

Value	Description
0	Interface is disabled
1	QSPI interface

Bits 7:4 – INSTANCE_ID[3:0] IP Instance ID

Value	Description
0	IP instance 0, QSPI

Bits 3:0 – IFACE_IOSET[3:0] Memory IOSET

Value	Description
0	PIO set 1, QSPI

Downloaded from Arrow.com.

11.4.4.5 QSPI Memory Configuration Data (Second Word)

Name: MEM_CFGx[1]

Bit	31	30	29	28	27	26	25	24
	XIP_ENABLE[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	XIP_ENABLE[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	XIP_ENABLE[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	XIP_ENABLE[7:0]							
Access								
Reset								

Bits 31:0 – XIP_ENABLE[31:0] XIP Mode Enable

Value	Description
0	XIP mode disabled
Non-Zero	XIP mode enabled

Downloaded from Arrow.com.

11.4.4.6 SDMMC Memory Configuration Data (First Word)

Name: MEM_CFGx[0]

WARNING To avoid any malfunctioning, the user must not write the "DO NOT USE (DNU)" bits.

Bit	31	30	29	28	27	26	25	24
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DNU	DNU	DNU	DNU	IFACE_TYPE[3:0]			
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	INSTANCE_ID[3:0]				IFACE_IOSET[3:0]			
Access								
Reset								

Bits 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – DNU DO NOT USE

Bits 11:8 – IFACE_TYPE[3:0] Interface Type

Value	Description
0	Interface is disabled
3	SDMMC interface

Bits 7:4 – INSTANCE_ID[3:0] IP Instance ID

Value	Description
0	IP instance 0, SDMMC
1	IP instance 1, SDMMC

Bits 3:0 – IFACE_IOSET[3:0] Memory IOSET

Value	Description
0	PIO set 1, SDMMC

Downloaded from Arrow.com.

11.4.4.7 SDMMC Memory Configuration Data (Second Word)

Name: MEM_CFGx[1]

WARNING To avoid any malfunctioning, the user must not write the "DO NOT USE (DNU)" bits.

Bit	31	30	29	28	27	26	25	24
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ENABLE	PID[1:0]		PIN[4:0]				
Access								
Reset								

Bits 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – DNU DO NOT USE

Bits 15:8 – WPKEY[7:0] Write Protect Key

Value	Name	Description
0x96	PASSWD	If any other value is written in this field, all the other bit values are ignored.

Bit 7 – ENABLE Card Detect Enable

Value	Description
0	Card detect disable, the ROM code does not use any card detect pin and directly tries to boot from the memory connected to the SDMMC controller.
1	Card detect enable, the ROM code checks the level of the card detect pin. If the level is 0, the ROM code tries to boot from the memory connected to the SDMMC controller. If the level is 1, the ROM code skips the SDMMC controller and jumps to the next interface in the boot sequence.

Bits 6:5 – PID[1:0] Peripheral ID
The peripheral ID of the PIO controller managing the card detect pin.

Value	Description
0	PIOA
1	PIOB
2	PIOC
3	PIOD

Bits 4:0 – PIN[4:0] Card Detect Pin Index
The index of the card detect pin inside the PIO controller.

Downloaded from Arrow.com.

11.4.4.8 FLEXCOM SPI Memory Configuration Data (First Word)

Name: MEM_CFGx[0]

WARNING To avoid any malfunctioning, the user must not write the "DO NOT USE (DNU)" bits.

Bit	31	30	29	28	27	26	25	24
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DNU	DNU	DNU	DNU	IFACE_TYPE[3:0]			
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	INSTANCE_ID[3:0]				IFACE_IOSET[3:0]			
Access								
Reset								

Bits 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – DNU DO NOT USE

Bits 11:8 – IFACE_TYPE[3:0] Interface Type

Value	Description
0	Interface is disabled
2	FLEXCOM SPI interface

Bits 7:4 – INSTANCE_ID[3:0] IP Instance ID

Value	Description
0	IP instance 0, FLEXCOM SPI
1	IP instance 1, FLEXCOM SPI
2	IP instance 2, FLEXCOM SPI
3	IP instance 3, FLEXCOM SPI
4	IP instance 4, FLEXCOM SPI
5	IP instance 5, FLEXCOM SPI

Bits 3:0 – IFACE_IOSET[3:0] Memory IOSET

Value	Description
0	PIO set 1, all FLEXCOM SPIs
1	PIO set 2, all FLEXCOM SPIs
2	PIO set 3, only for FLEXCOM4_SPI and FLEXCOM5_SPI
3	PIO set 4, only for FLEXCOM4_SPI and FLEXCOM5_SPI
4	PIO set 5, only for FLEXCOM4_SPI and FLEXCOM5_SPI
5	PIO set 6, only for FLEXCOM4_SPI

Downloaded from Arrow.com.

11.4.4.9 FLEXCOM SPI Memory Configuration Data (Second Word)

Name: MEM_CFGx[1]

WARNING To avoid any malfunctioning, the user must not write the "DO NOT USE (DNU)" bits.

Bit	31	30	29	28	27	26	25	24
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – DNU DO NOT USE

Downloaded from Arrow.com.

11.4.4.10 NAND Memory Configuration Data (First Word)

Name: MEM_CFGx[0]

WARNING To avoid any malfunctioning, the user must not write the "DO NOT USE (DNU)" bits.

Bit	31	30	29	28	27	26	25	24
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DNU	DNU	DNU	DNU	IFACE_TYPE[3:0]			
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	INSTANCE_ID[3:0]				IFACE_IOSET[3:0]			
Access								
Reset								

Bits 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – DNU DO NOT USE

Bits 11:8 – IFACE_TYPE[3:0] Interface Type

Value	Description
0	Interface is disabled
4	NAND interface

Bits 7:4 – INSTANCE_ID[3:0] IP Instance ID

Value	Description
0	IP instance 0, NAND

Bits 3:0 – IFACE_IOSET[3:0] Memory IOSET

Value	Description
0	PIO set 1, NAND
1	PIO set 2, NAND

Downloaded from Arrow.com.

11.4.4.11 NAND Memory Configuration Data (Second Word)

Name: MEM_CFGx[1]

WARNING To avoid any malfunctioning, the user must not write the "DO NOT USE (DNU)" bits.

Bit	31	30	29	28	27	26	25	24
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DNU	DNU	DNU	DNU	DNU	DNU	DNU	DNU
Access								
Reset								

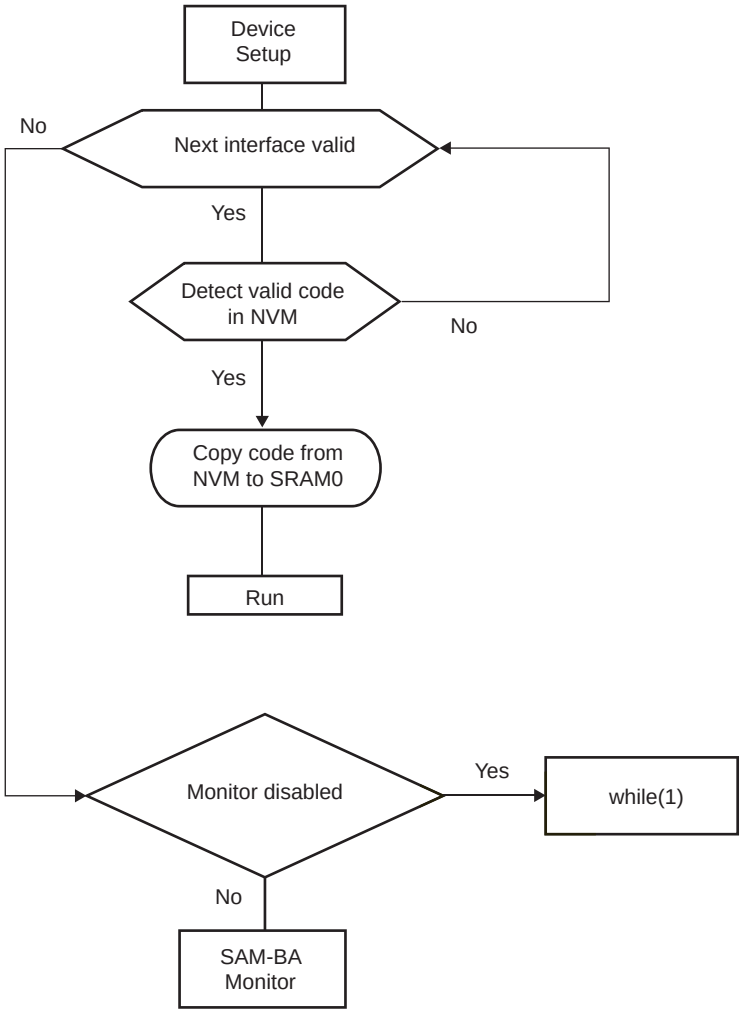
Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – DNU DO NOT USE

11.4.5 NVM Boot Sequence

The ROM code performs the initialization and valid code detection for external memories as described below when the memory interface boot is enabled in the Boot Configuration packet.

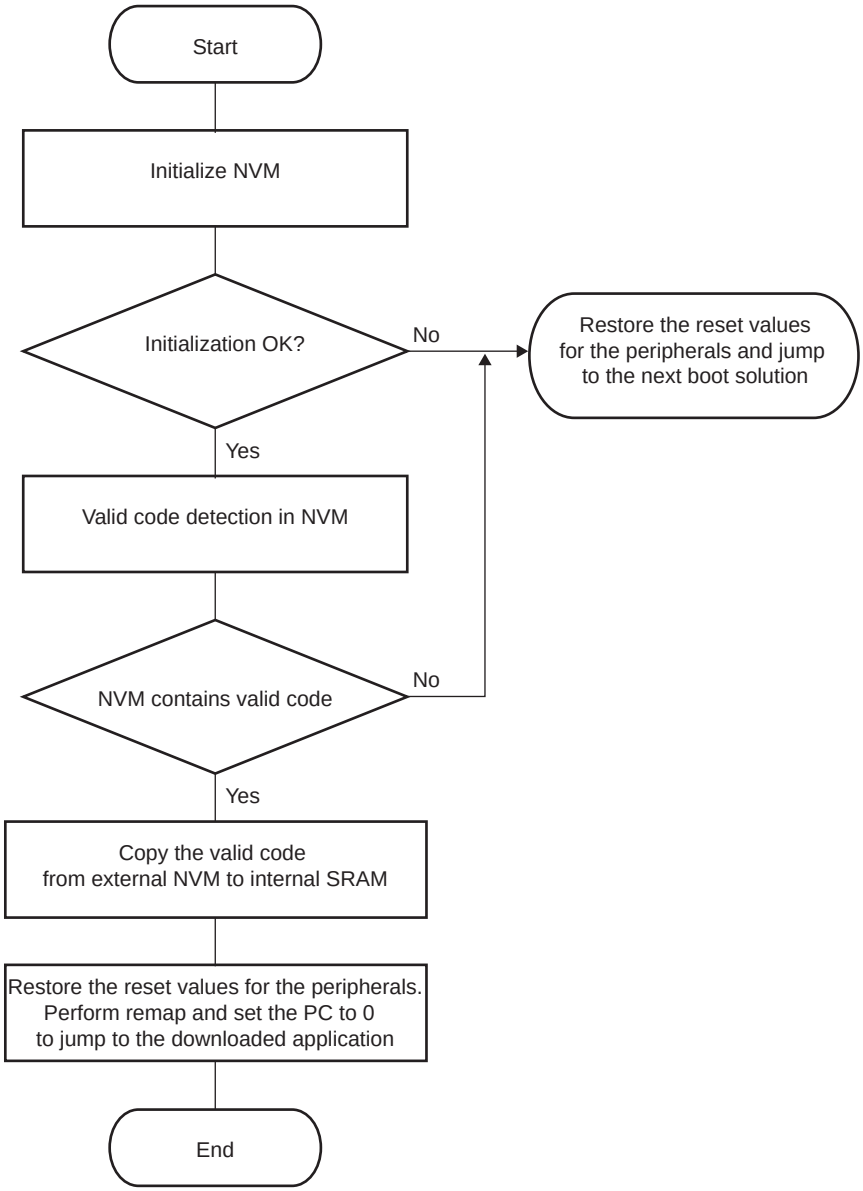
Downloaded from Arrow.com.

Figure 11-3. NVM Bootloader Program



Downloaded from Arrow.com.

Figure 11-4. NVM Boot Diagram



The NVM bootloader program first initializes the PIOs related to the NVM device. Then it configures the right peripheral depending on the NVM and tries to access this memory. If the initialization fails, it restores the reset values for the PIO and the peripheral, and then tries to perform the same operations on the next NVM of the sequence.

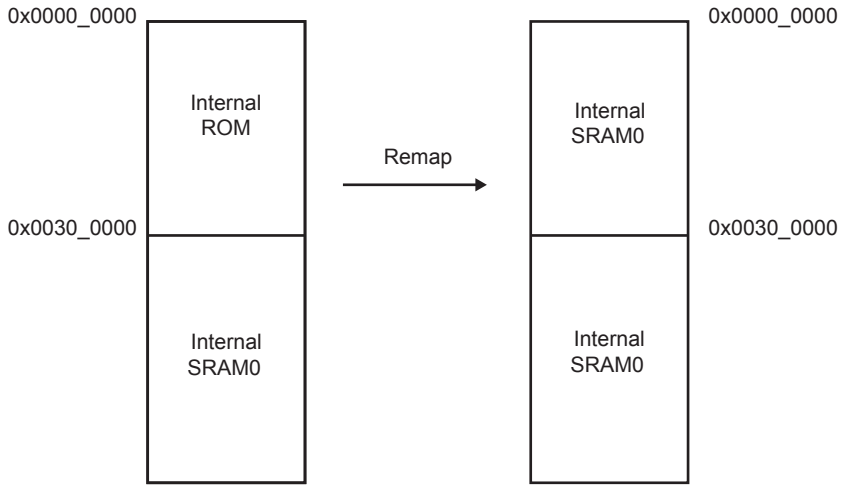
If the initialization is successful, the NVM bootloader program reads the beginning of the NVM and determines if the NVM contains a valid code.

If the NVM does not contain a valid code, the NVM bootloader program restores the reset value for the peripherals and then tries to perform the same operations on the next NVM of the sequence.

If a valid code is found, this code is loaded from the NVM into the internal SRAM and executed by branching at address 0x0000_0000 after remap. This code may be the application code or a second-level bootloader.

Downloaded from Arrow.com.

Figure 11-5. Remap Action after Download Completion



11.4.6 Valid Code Detection

Two types of valid code detection are available:

- Arm Exception Vectors Check
- boot.bin File Check

11.4.6.1 Arm Exception Vectors Check

The NVM bootloader program reads and analyzes the first 28 bytes corresponding to the first seven Arm exception vectors. Except for the sixth vector, these bytes must implement the Arm instructions for either branch or load PC with PC-relative addressing.

Figure 11-6. LDR Opcode

31 28	27 24	23 20	19 16	15 12	11 0
1 1 1 0	0 1 I P	U 1 W 0	Rn	Rd	Offset

Figure 11-7. B Opcode

31 28	27 24	23 0
1 1 1 0	1 0 1 0	Offset (24 bits)

Unconditional instruction: 0xE for bits 31 to 28.

Load PC with the PC-relative addressing instruction:

- Rn = Rd = PC = 0xF
- I==0 (12-bit immediate value)
- P==1 (pre-indexed)
- U offset added (U==1) or subtracted (U==0)
- W==1

The sixth vector, at offset 0x14, contains the size of the image to download. The user must replace this vector with the user's own vector. This procedure is described below.

Figure 11-8. Arm Vector 6 Structure

31 0
Size of the code to download in bytes

The value must be lower than 32 Kbytes.

Downloaded from Arrow.com.

The following is an example of valid vectors:

```
00            ea000006                  B 0x20
04            eafffffe                  B 0x04
08            eafffffe                  B 0x08
0c            eafffffe                  B 0x0c
10            eafffffe                  B 0x10
14            00001234                  ← Code size = 4660 bytes
18            eafffffe                  B 0x18
```

11.4.6.2 boot.bin File Check

This method is the one used on FAT-formatted SDCard and e.MMC. The boot program must be a file named `boot.bin` written in the file system root directory. Its size must not exceed the maximum size allowed: 32 Kbytes (0x8000). The Arm exception vectors at offset 0 in the "`boot.bin`" file must also pass the test described in 11.4.6.1 Arm Exception Vectors Check.

11.4.7 Detailed Memory Boot Procedures

11.4.7.1 NAND Flash Boot: NAND Flash Detection

After the NAND Flash interface configuration, a reset command is sent to the memory.

The reset time of the NAND memory, after this reset command, must not be higher than 100 µs.

Hardware ECC detection and correction are provided by the PMECC peripheral. See Section 37.18 "PMECC Controller Functional Description" for more details.

The Boot Program is able to retrieve NAND Flash parameters and ECC requirements using either one of the following methods:

- Method 1 (recommended): NAND Flash Specific Header Detection
- Method 2: ONFI 2.2 Parameters

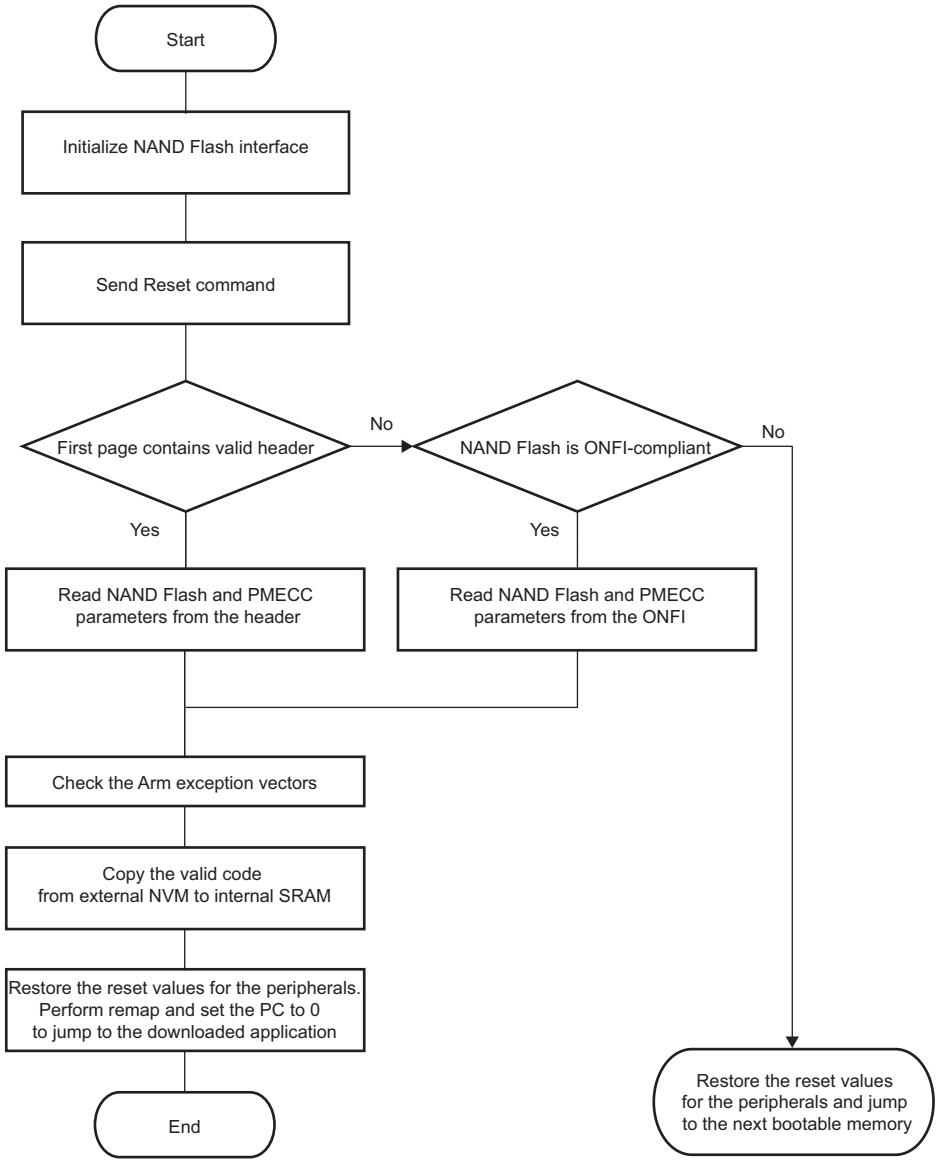
It is highly recommended to use Method 1, since it indicates exactly how the PMECC has been configured to read or write the bootable program in the NAND Flash, and does not rely only on the NAND Flash capabilities.

Once the Boot program retrieves the parameter, using one of the above two methods, it reads the first page again, with or without ECC, depending on the usePmecc parameter. Then it looks for a valid code programmed just after the header offset 0xD0. If the code is valid, the program is copied at the beginning of the internal SRAM.

Note: Booting on 16-bit NAND Flash is not possible; only 8-bit NAND Flash memories are supported.

Downloaded from Arrow.com.

Figure 11-9. Boot NAND Flash Download



11.4.7.1.1 Method 1 (recommended): NAND Flash Specific Header Detection

This is the first method used to determine NAND Flash parameters. After receiving the Initialization and Reset command, the Boot Program reads the first page without an ECC check, to determine whether the NAND parameter header is present. The header is made of 52 times the same 32-bit word (for redundancy reasons) which must contain NAND and PMECC parameters used to correctly perform the read of the rest of the data in the NAND. This 32-bit word is described below.

If the header is valid, the Boot program continues with the detection of a valid code.

Downloaded from Arrow.com.

[NAND Flash Specific Header Detection]

Name: NAND Flash Specific Header Detection

Bit	31	30	29	28	27	26	25	24
	key[3:0]					eccOffset[8:6]		
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	eccOffset[5:0]						sectorSize[1:0]	
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	eccBitReq[2:0]			spareSize[8:4]				
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	spareSize[3:0]				nbSectorPerPage[2:0]			usePmecc
Access								
Reset								

Bits 31:28 – key[3:0] Value 0xC Must be Written here to Validate the Content of the Whole Word

Bits 26:18 – eccOffset[8:0] Offset of the First ECC Byte in the Spare Zone
A value below 2 is not allowed and is considered as 2.

Bits 17:16 – sectorSize[1:0] Size of the ECC Sector

Value	Description
0	For 512 bytes
1	For 1024 bytes per sector
Other values	For future use

Bits 15:13 – eccBitReq[2:0] Number of ECC Bits Required

Value	Description
0	2-bit ECC
1	4-bit ECC
2	8-bit ECC
3	12-bit ECC
4	24-bit ECC

Bits 12:4 – spareSize[8:0] Size of the Spare Zone in Bytes

Bits 3:1 – nbSectorPerPage[2:0] Number of Sectors per Page

Value	Description
0	1 sector per page
1	2 sectors per page
2	4 sectors per page
3	8 sectors per page
4	16 sectors per page

Bit 0 – usePmecc Use PMECC

Value	Description
0	Do not use PMECC to detect and correct the data.

Downloaded from Arrow.com.

Value	Description
1	Use PMECC to detect and correct the data.

11.4.7.2 NAND Flash Boot: PMECC Error Detection and Correction

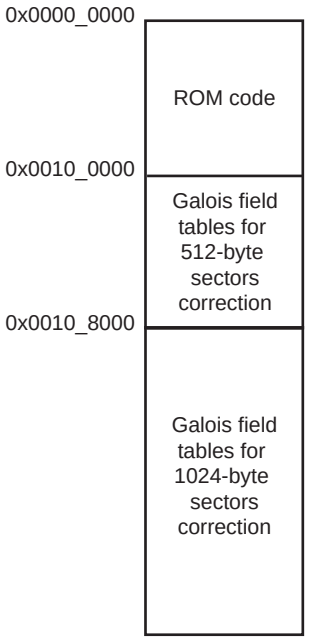
NAND Flash boot procedure uses PMECC to detect and correct errors during NAND Flash read operations in two cases:

- When the usePmecc flag is set in a specific NAND header. If the flag is not set, no ECC correction is performed during the NAND Flash page read.
- When the NAND Flash has been detected using ONFI parameters.

The ROM memory embeds the Galois field tables. The user does not need to embed them in his/her own software.

The Galois field tables are mapped in the ROM just after the ROM code, as shown in the following figure.

Figure 11-10. Galois Field Table Mapping



For a full description and an example of how to use the PMECC detection and correction feature, refer to the software package dedicated to this device on www.microchip.com.

11.4.7.3 SDCard/e.MMC Boot

In the case of activated Card Detect pin in the Boot Configuration packet, and if the level of the Card Detect pin is high, no communication with SDCard/e.MMC is performed (no IOs toggling). Otherwise, the SDCard/e.MMC access is initiated (IOs toggling).

Supported SDCard Devices

SDCard Boot supports all SDCard memories compliant with the SD Memory Card Specification V3.0. This includes SDMMC cards.

e.MMC with Boot Partition

The ROM code first checks if the e.MMC Boot Partition is enabled. If enabled, the ROM code reads the first 32 Kbytes of the boot partition, and copy them into the internal SRAM0.

FAT Filesystem Boot

If no boot partition is enabled on an e.MMC, the boot process continues with a standard SDCard/e.MMC detection, and the ROM code looks for a `boot.bin` file in the root directory of a FAT12/16/32 file system.

11.4.7.4 SPI Flash Boot

Two types of SPI Flash are supported: SPI Serial Flash and SPI DataFlash.

Downloaded from Arrow.com.

The SPI Flash bootloader tries to boot on SPI0, first looking for SPI Serial Flash, and then for SPI DataFlash.

It uses only one valid code detection: analysis of Arm exception vectors.

The SPI Flash read is done by means of a Continuous Read command from the address 0x0. This command is 0xE8 for DataFlash and 0x0B for Serial Flash devices.

Supported DataFlash Devices

The SPI Flash Boot program supports the DataFlash devices listed in the following table.

Table 11-2. DataFlash Devices

Device	Density	Page Size (bytes)	Number of Pages
AT45DB011	1 Mbit	264	512
AT45DB021	2 Mbits	264	1024
AT45DB041	4 Mbits	264	2048
AT45DB081	8 Mbits	264	4096
AT45DB161	16 Mbits	528	4096
AT45DB321	32 Mbits	528	8192
AT45DB642	64 Mbits	1056	8192
AT45DB641	64 Mbits	264	37768

Supported Serial Flash Devices

The SPI Flash Boot program supports all SPI Serial Flash devices responding correctly to both Get Status and Continuous Read commands.

11.4.7.5 QSPI NOR Flash Boot

Hardware Considerations

The ROM code configures the hardware so that:

- the QSPI controller uses SPI Mode 0 (CPOL = 0 and CPHA = 0),
- the QSPIx_SCK clock frequency is ≤ 50 MHz,
- QSPIx_SCK and QSPIx_CS do not use any internal pull-up/pull-down resistor,
- each QSPIx_IO{0,1,2,3} uses the PIO controller's internal pull-up resistor.

Software Considerations

Before reading any data, the ROM code sends a software reset to the QSPI NOR memory. Then the ROM code looks for the Serial Flash Discoverable Parameters (SFDP) of the QSPI NOR memory, if available, to learn the parameters (instruction op code, timing settings) required to read the user-programmed boot file.

If SFDP tables are not available, the ROM code uses hard-coded values as fallback settings to read the boot file.

The ROM code supports any QSPI NOR memory which can provide its Serial Flash Discoverable Parameters (SFDP) as defined in the JEDEC JESD216B standard.

The supported revisions of this JEDEC standard are:

- JESD216 (version 1.0)
- JESD216 rev. A (version 1.5)
- JESD216 rev. B (version 1.6)

Refer the QSPI NOR memory datasheet to check compliance with any of the above JEDEC JESD216 standard revisions/versions.

- QSPI NOR memories with SFDP (JEDEC JESD216x compliant)

The ROM code reads the memory SFDP tables to learn the factory settings (instruction op code, number of dummy cycles, etc.). The ROM code also reads bits[22:20] in DWORD15 from the Basic Flash Parameter table (refer to

Downloaded from Arrow.com.

JEDEC JESD216B specification) to select and then execute the relevant procedure, if any, to set the Quad Enable (QE) bit in some internal register of the QSPI NOR memory.

For most memory manufacturers, this QE bit is nonvolatile and must be set before performing any Quad SPI command. This is the only persistent setting that the ROM code may change in the internal registers of the QSPI NOR memory. All other settings are kept unchanged.

Refer to the QSPI NOR memory datasheet to find which value was chosen by the memory manufacturer and written into the SFDP tables.

Finally, the ROM code reads the boot file from the data area of the QSPI NOR memory, and then continues its boot procedure.

- QSPI NOR memories without SFDP

This section only applies when the ROM code fails to read the SFDP tables from the QSPI NOR memory.

The ROM code reads the JEDEC ID of the QSPI NOR memory, and then selects the read settings based on the manufacturer ID (first byte of the JEDEC ID) from the following hard-coded values:

	Cypress (01h)	Micron (20h)	Macronix (C2h)	Winbond (EFh)	Others
Fast Read protocol	SPI 1-4-4	SPI 1-4-4	SPI 1-4-4	SPI 1-4-4	SPI 1-1-1
Fast Read op code	EBh	EBh	EBh	EBh	0Bh
Address width	24 bits	24 bits	24 bits	24 bits	24 bits
Number of mode clock cycles	2	1	2	2	0
Number of wait states	4	9	4	4	8
Value of mode cycles to enter the 0-4-4 mode (XIP)	A0h	0h The ROM code first sets XIP bit[3] in the Volatile Configuration Register (VCR)	0Fh	A5h	N/A
Value of mode cycles to exit the 0-4-4 mode (normal read)	00h	1h	00h	FFh	N/A
XIP supported	yes	yes	yes	yes	no

Those hard-coded parameters give a last chance to the ROM code to boot from a QSPI NOR memory in either normal mode or XIP (Continuous Read) mode.

Table 11-3. QSPI NOR Memories Tested with and Supported by ROM Code (non-exhaustive)

Manufacturer	Memories
Microchip (SST)	SST26VF016B SST26VF032B SST26VF032BA SST26VF064B
Micron	N25Q128A N25Q128A13ESF N25Q256A13ESF N25Q512A13 MT25QL01G

Downloaded from Arrow.com.

.....continued	
Manufacturer	**Memories**
Macronix	MX25V4035FM2I MX25V8035FM2I MX25V1635FM2I MX25L3233FM2I-08G MX25L3273FM2I-08G MX25L6433FM2I-08G MX25L6473FM2I-08G MX25L12835FM2I-10G MX25L12845GMI-08G MX25L12873GM2I-08G MX25L25645G MX25L25673G MX25L51245GMI-10G MX66L1G45GMI-08G
Spansion	S25FL127 (normal boot only; XIP fails) S25FL164 S25FL512
Winbond	W25M512

Note: For an updated list of memories, refer to the "Booting from External Non-Volatile Memory (NVM) on SAM9X60" application note (available later).

11.4.8 Hardware and Software Constraints

The table below provides clock frequencies configured by the ROM code during boot.

Table 11-4. Clock Frequencies During External Memory Boot Sequence

Clock	Frequency
PLLA	396 MHz
CPU_CLK	396 MHz
MCK	99 MHz
SDMMC (init/operational)	400 kHz / 25 MHz
SPI	11 MHz
QSPI	33 MHz

The NVM drivers use several PIOs in Peripheral mode to communicate with external memory devices. Care must be taken when these PIOs are used by the application. The connected devices could be unintentionally driven at boot time, and thus electrical conflicts between the output pins used by the NVM drivers and the connected devices could occur.

The following table contains a list of pins that are driven during the boot program execution. These pins are driven during the boot sequence for a period of less than 1 second if no correct boot program is found. The drive strength of PULL-UP I/O pins is set to High while the pins are used in Peripheral mode by the ROM code.

Downloaded from Arrow.com.

Before performing the jump to the application in the internal SRAM, all the PIOs and peripherals used in the boot program are set to their reset state.

Table 11-5. PIO Driven during Boot Program Execution

NVM Bootloader	Peripheral	IO Set	Signal	PIO Line	Pull-up
SDCard/e.MMC	SDMMC_0	1	SDMMC0_DAT0	PIO_PA15A	X
			SDMMC0_CMD	PIO_PA16A	X
			SDMMC0_CK	PIO_PA17A	–
			SDMMC0_DAT1	PIO_PA18A	X
			SDMMC0_DAT2	PIO_PA19A	X
			SDMMC0_DAT3	PIO_PA20A	X
	SDMMC_1	1	SDMMC1_DAT1	PIO_PA2B	X
			SDMMC1_DAT2	PIO_PA3B	X
			SDMMC1_DAT3	PIO_PA4B	X
			SDMMC1_DAT0	PIO_PA11B	X
			SDMMC1_CMD	PIO_PA12B	X
			SDMMC1_CK	PIO_PA13B	–
NAND Flash	HSMC	1	D16–D23	PIO_PD6A-PIO_PD13A	–
			NANDOE	PIO_PD0A	–
			NANDWE	PIO_PD1A	–
			NAND ALE	PIO_PD2A	–
			NAND CLE	PIO_PD3A	–
			NANDCS3	PIO_PD4A	–
			NAND WAIT	PIO_PD5A	–
		2	NANDOE	PIO_PD0A	–
			NANDWE	PIO_PD1A	–
			A21-A22	PIO_PD2A-PIO_PD3A	–
			NANDCS3	PIO_PD4A	–
			NAND WAIT	PIO_PD5A	–
			D0-D7	–	–
			A20	PIO_PD15B	–
			A23-A25	PIO_PD16B-PIO_PD18B	–
			NANDCS2	PIO_PD19B	–
			NANDCS4 -NANDCS5	PIO_PD20B-PIO_PD21B	–

Downloaded from Arrow.com.

.....continued

NVM Bootloader	Peripheral	IO Set	Signal	PIO Line	Pull-up
SPI Flash	FLEXCOM0_SPI	1	MOSI	PIO_PA0A	–
			MISO	PIO_PA1A	X
			NPCS0	PIO_PA3A	–
			SPCK	PIO_PA4A	–
		2	MOSI	PIO_PA0A	–
			MISO	PIO_PA1A	X
			NPCS1	PIO_PA2A	–
			SPCK	PIO_PA4A	–
	FLEXCOM1_SPI	1	MOSI	PIO_PA5A	–
			MISO	PIO_PA6A	X
			NPCS0	PIO_PC28C	–
			SPCK	PIO_PC29C	–
		2	MOSI	PIO_PA5A	–
			MISO	PIO_PA6A	X
			NPCS1	PIO_PC27C	–
			SPCK	PIO_PC29C	–
	FLEXCOM2_SPI	1	MOSI	PIO_PA7A	–
			MISO	PIO_PA8A	X
			SPCK	PIO_PB1B	–
			NPCS0	PIO_PB2B	–
		2	MOSI	PIO_PA7A	–
			MISO	PIO_PA8A	X
			SPCK	PIO_PB2B	–
			NPCS1	PIO_PB0B	–
	FLEXCOM3_SPI	1	MOSI	PIO_PC22B	–
			MISO	PIO_PC23B	X
			NPCS0	PIO_PC25B	–
			SPCK	PIO_PC26B	X
		2	MOSI	PIO_PC22B	–
			MISO	PIO_PC23B	X
			NPCS1	PIO_PC24B	–
			SPCK	PIO_PC26B	–

Downloaded from Arrow.com.

.....continued

NVM Bootloader	Peripheral	IO Set	Signal	PIO Line	Pull-up
SPI Flash	FLEXCOM4_SPI	1	MISO	PIO_PA11A	X
			MOSI	PIO_PA12A	–
			SPCK	PIO_PA13A	–
			NPCS0	PIO_PA14A	–
		2	MISO	PIO_PA11A	X
			MOSI	PIO_PA12A	–
			SPCK	PIO_PA13A	–
			NPCS1	PIO_PA0C	–
		3	MISO	PIO_PA11A	X
			MOSI	PIO_PA12A	–
			SPCK	PIO_PA13A	–
			NPCS1	PIO_PA7B	–
		4	MISO	PIO_PA11A	X
			MOSI	PIO_PA12A	–
			SPCK	PIO_PA13A	–
			NPCS2	PIO_PA1B	–
		5	MISO	PIO_PA11A	X
			MOSI	PIO_PA12A	–
			SPCK	PIO_PA13A	–
			NPCS2	PIO_PA8C	–
		6	MISO	PIO_PA11A	X
			MOSI	PIO_PA12A	–
			SPCK	PIO_PA13A	–
			NPCS3	PIO_PB3B	–
SPI Flash	FLEXCOM5_SPI	1	NPCS0	PIO_PA8B	–
			MISO	PIO_PA21B	X
			MOSI	PIO_PA22B	–
			SPCK	PIO_PA23B	–
		2	NPCS1	PIO_PA0B	–
			MISO	PIO_PA21B	X
			MOSI	PIO_PA22B	–
			SPCK	PIO_PA23B	–
		3	MISO	PIO_PA21B	X
			MOSI	PIO_PA22B	–
			SPCK	PIO_PA23B	–
			NPCS1	PIO_PA7C	–
		4	MISO	PIO_PA21B	X
			MOSI	PIO_PA22B	–
			SPCK	PIO_PA23B	–
			NPCS2	PIO_PA31B	–
		5	MISO	PIO_PA21B	–
			MOSI	PIO_PA22B	X
			SPCK	PIO_PA23B	–
			NPCS3	PIO_PA30B	–

Downloaded from Arrow.com.

.....continued

NVM Bootloader	Peripheral	IO Set	Signal	PIO Line	Pull-up
QSPI Flash	QSPI_0	1	QSCK	PIO_PB19A	–
			QCS	PIO_PB20A	–
			QIO0	PIO_PB21A	X
			QIO1	PIO_PB22A	X
			QIO2	PIO_PB23A	X
			QIO3	PIO_PB24A	X
Console and SAM-BA Monitor	DBGU	1	DTXD	PIO_PA10A	–
			DRXD	PIO_PA9A	–
	FLEXCOM0_ UART	1	DTXD	PIO_PA0A	–
			DRXD	PIO_PA1A	–
	FLEXCOM1_UART	1	DTXD	PIO_PA5A	–
			DRXD	PIO_PA6A	–
	FLEXCOM2_UART	1	DTXD	PIO_PA7A	–
			DRXD	PIO_PA8A	–
	FLEXCOM3_UART	1	DTXD	PIO_PC22B	–
			DRXD	PIO_PC23B	–
	FLEXCOM4_UART	1	DTXD	PIO_PA12A	–
			DRXD	PIO_PA11A	–
	FLEXCOM5_UART	1	DTXD	PIO_PA22B	–
			DRXD	PIO_PA21B	–
	FLEXCOM6_UART	1	DTXD	PIO_PA30A	–
			DRXD	PIO_PA31A	–
	FLEXCOM7_UART	1	DTXD	PIO_PC0C	–
			DRXD	PIO_PC1C	–
	FLEXCOM8_UART	1	DTXD	PIO_PB4B	–
			DRXD	PIO_PB5B	–
	FLEXCOM9_UART	1	DTXD	PIO_PC8C	–
			DRXD	PIO_PC9C	–
	FLEXCOM10_UART	1	DTXD	PIO_PC16C	–
			DRXD	PIO_PC17C	–
	FLEXCOM11_UART	1	DTXD	PIO_PB19C	–
			DRXD	PIO_PB20C	–
	FLEXCOM12_UART	1	DTXD	PIO_PB21C	–
			DRXD	PIO_PB22C	–

11.5 SAM-BA Monitor

This part of the ROM code is executed when no valid code is found in any NVM during the NVM boot sequence, and if the MON_DIS field is not set to 1 in the Boot Configuration packet.

The main RC oscillator is used as the Main Clock. To configure the USB clock, the Main Oscillator is enabled by setting the CKGR_MOR.MOSCXTEN and CKGR_MOR.MOSCSEL bits. Then the external quartz detection is started. This detection is successful when the MOSCXTS bit rises, else the USB is not activated and only UART is used in SAM-BA Monitor.

If an external clock or crystal frequency is found, then the UPLL is configured to allow communication on the USB link for the SAM-BA Monitor.

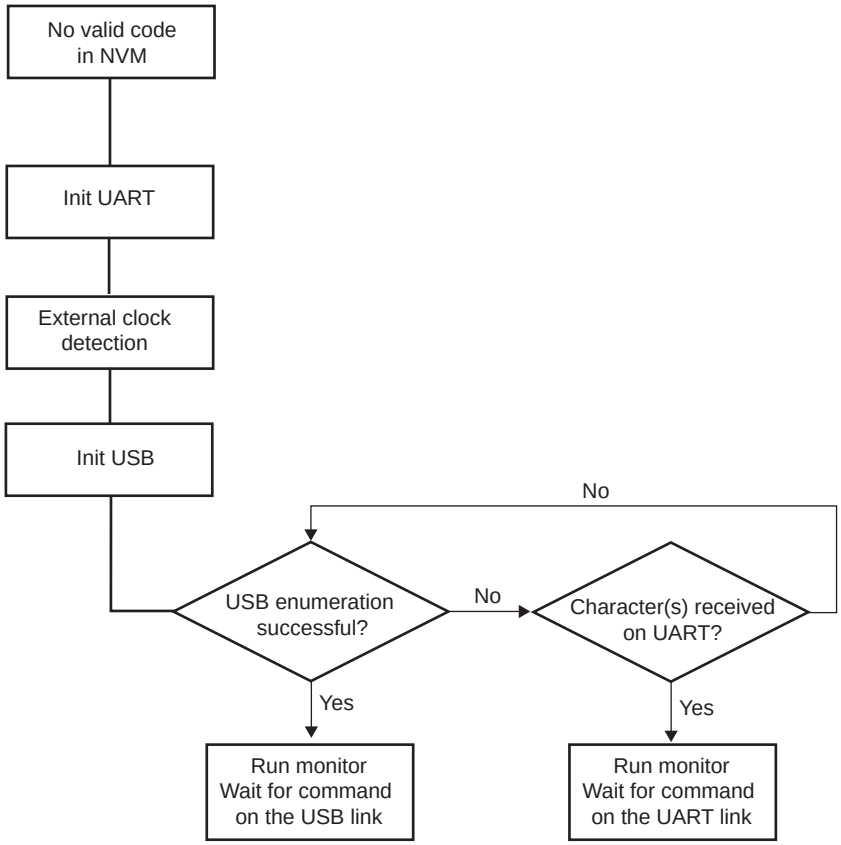
Downloaded from Arrow.com.

The SAM-BA Monitor steps are:

- Initialize UART and USB.
- Check if USB Device enumeration occurred.
- Check if characters are received on the UART.

Once the communication interface is identified, the application runs in an infinite loop waiting for different commands as listed in Table 11-6.

Figure 11-11. SAM-BA Monitor



11.5.1 Command List

Table 11-6. Commands Available Through SAM-BA Monitor

Command	Action	Argument(s)	Example
N	Set Normal Mode	No argument	**N**#
T	Set Terminal Mode	No argument	**T**#
O	Write a byte	Address, Value#	**O**200001,CA#
o	Read a byte	Address,#	**o**200001,#
H	Write a halfword	Address, Value#	**H**200002,CAFE#
h	Read a halfword	Address,#	**h**200002,#
W	Write a word	Address, Value#	**W**200000,CAFEDECA#
w	Read a word	Address,#	**w**200000,#
S	Send a file	Address,#	**S**200000,#

Downloaded from Arrow.com.

.....continued			
Command	**Action**	**Argument(s)**	**Example**
R	Receive a file	Address, NbOfBytes#	**R**200000,1234#
G	Go	Address#	**G**200200#
V	Display version	No argument	**V**#

- Mode commands:
  - Normal mode configures SAM-BA Monitor to send/receive data in binary format.
  - Terminal mode configures SAM-BA Monitor to send/receive data in ASCII format.
- Write commands: Writes a byte (**O**), a halfword (**H**) or a word (**W**) to the target.
  - Address: address in hexadecimal
  - Value: byte, halfword or word to write in hexadecimal
  - Output: '>'
- Read commands: Reads a byte (**o**), a halfword (**h**) or a word (**w**) from the target.
  - Address: address in hexadecimal
  - Output: the byte, halfword or word read in hexadecimal followed by '>'
- Send a file (**S**): Sends a file to a specified address.
  - Address: address in hexadecimal
  - Output: '>'
    Note: There is a timeout on this command which is reached when the prompt '>' appears before the end of the command execution.
- Receive a file (**R**): Receives data into a file from a specified address.
  - Address: address in hexadecimal
  - NbOfBytes: number of bytes in hexadecimal to receive
  - Output: '>'
- Go (**G**): Jumps to a specified address and executes the code.
  - Address: address to jump to in hexadecimal
  - Output: '>' once returned from the program execution. If the executed program does not handle the link register at its entry and does not return, the prompt is not displayed.
- Get Version (**V**): Returns the Boot Program version.
  - Output: version, date and time of ROM code followed by '>'

11.5.2 DBGU/UART Console Port

Communication is performed through the DBGU/UART port initialized to 115,200 bauds: 8 bits of data, no parity, 1-stop bit.

11.5.2.1 Xmodem Protocol

The Send and Receive File commands use the Xmodem protocol to communicate. Any terminal using this protocol can be used to send the application file to the target. The size of the binary file to send depends on the SRAM size embedded in the product. In all cases, the size of the binary file must be lower than the SRAM size because the Xmodem protocol requires some SRAM memory in order to work.

The Xmodem protocol supported is the 128-byte length block. This protocol uses a two-character CRC16 to guarantee detection of maximum bit errors.

Xmodem protocol with CRC is supported by successful transmission reports provided both by a sender and by a receiver. Each transfer block is as follows:

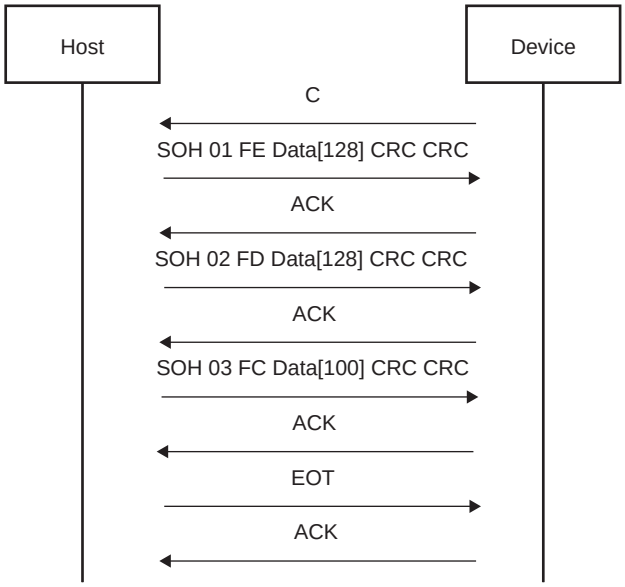
<SOH><blk #><255-blk #><--128 data bytes--><checksum> in which:

- <SOH> = 01 hex
- <blk #> = binary number, starts at 01, increments by 1, and wraps 0FFH to 00H (not to 01)
- <255-blk #> = 1's complement of the blk#.
- <checksum> = 2-byte CRC16

Downloaded from Arrow.com.

The following figure shows a transmission using this protocol.

Figure 11-12. Xmodem Transfer Example



11.5.3 USB Device Port

11.5.3.1 Supported External Crystal / External Clocks

The SAM-BA Monitor supports an external crystal or external clock frequency at 12 MHz, 16 MHz, 24 MHz or 48 MHz to allow USB communication.

11.5.3.2 USB Class

The device uses the USB Communication Device Class (CDC) drivers to take advantage of the installed PC Serial Communication software to talk over the USB. The CDC is implemented in all releases of Microsoft Windows®, starting from Windows 98SE®. The CDC document, available at www.usb.org, describes how to implement devices such as ISDN modems and virtual COM ports.

The vendor ID is 0x03EB. The product ID is 0x6124. These references are used by the host operating system to mount the correct driver. On Windows systems, INF files contain the correspondence between vendor ID and product ID.

11.5.3.3 Enumeration Process

The USB protocol is a master/slave protocol. The host starts the enumeration, sending requests to the device through the control endpoint. The device handles standard requests as defined in the USB Specification.

Table 11-7. Handled Standard Requests

Request	Definition
GET_DESCRIPTOR	Returns the current device configuration value
SET_ADDRESS	Sets the device address for all future device accesses
SET_CONFIGURATION	Sets the device configuration
GET_CONFIGURATION	Returns the current device configuration value
GET_STATUS	Returns status for the specified recipient
SET_FEATURE	Used to set or enable a specific feature
CLEAR_FEATURE	Used to clear or disable a specific feature

The device also handles some class requests defined in the CDC class.

Downloaded from Arrow.com.

Table 11-8. Handled Class Requests

Request	Definition
SET_LINE_CODING	Configures DTE rate, stop bits, parity and number of character bits
GET_LINE_CODING	Requests current DTE rate, stop bits, parity and number of character bits
SET_CONTROL_LINE_STATE	RS-232 signal used to indicate to the DCE device that the DTE device is now present

Unhandled requests are stalled.

11.5.3.4 Communication Endpoints

Endpoint 0 is used for the enumeration process.

Endpoint 1 (64-byte Bulk OUT) and endpoint 2 (64-byte Bulk IN) are used as communication endpoints.

SAM-BA Boot commands are sent by the host through Endpoint 1. If required, the message is split into several data payloads by the host driver.

If the command requires a response, the host sends IN transactions to pick up the response.

Downloaded from Arrow.com.

12. System Controller Write Protection (SYSCWP)

12.1 Functional Description

12.1.1 System Controller Peripheral Mapping

Table 12-1. System Controller Peripheral Mapping

Offset	System Controller Peripheral	Name
0x000-0x00C	Reset Controller	RSTC
0x010-0x01C	Shutdown Controller	SHDWC
0x020-0x03C	Real Time Timer	RTT
0x040-0x04C	Period Interval Timer	PIT
0x050-0x05C	Slow Clock Controller	SCKC
0x060-0x0A4	General Purpose Backup Registers	GPBR
0x0A8-0xD8	Real Time Clock	RTC
0x0DC	Write Protection Mode Register	SYSC_WPMR
0x0E0	Write Protection Status Register	SYSC_WPSR
0x100-0x16C	Real Time Clock (tamper control and timestamp)	RTC
0x180-0x1AC	Watchdog Timer	WDT

12.1.2 Register Write Protection

To prevent any single software error from modifying the configuration of the Reset Controller (RSTC), Shutdown Controller (SHDWC), Real-time Timer (RTT), Periodic Interval Timer (PIT), Slow Clock Controller (SCKC), General Purpose Backup Register (GPBR), Real-time Clock (RTC) and Watchdog Timer (WDT), some registers of these peripherals can be write-protected by setting the WPEN and/or WPITEN bits in the System Controller Write Protection Mode register (SYSC_WPMR).

Note: The WDT embeds additional write protection mechanisms.

When write protection is enabled, any attempt to write these registers is reported in the System Controller Write Protection Status register (SYSC_WPSR).

The following registers can be write-protected when SYSC_WPMR.WPEN=1:

- WDT Control Register
- WDT Mode Register
- RSTC Mode Register
- SHDWC Mode Register
- SHDWC Wakeup Inputs Register
- PIT Mode Register
- SCKC Configuration Register
- RTC Control Register
- RTC Mode Register
- RTC Time Alarm Register
- RTC Calendar Alarm Register
- RTT Mode Register
- RTT Alarm Register

Downloaded from Arrow.com.

- RTT Modulo Selection Register
- GPBR Full Clear Register
- GPBR Registers

The following registers can be write-protected when SYSC_WPMR.WPITEN=1:

- RTC Interrupt Enable Register
- RTC Interrupt Disable Register

Downloaded from Arrow.com.

12.2 Register Summary

Offset	Name	Bit Pos.								
0x00	SYSC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPITEN	WPEN
0x04	SYSC_WPSR	31:24								
		23:16								
		15:8	WVSRC[7:0]							
		7:0								WPVS

Downloaded from Arrow.com.

12.2.1 SYSC Write Protection Mode Register

Name: SYSC_WPMR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
							WPITEN	WPEN
Access							R/W	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x535943	PASSWD	Writing any other value in this field aborts the write operation of the WPEN and WPITEN bits. Always reads as 0.

Bit 1 – WPITEN Write Protection RTC Interrupt Enable

Value	Description
0	Disables the write protection of the RTC_IER/RTC_IDR configuration registers if WPKEY corresponds to 0x535943 (“SYC” in ASCII).
1	Enables the write protection of the RTC_IER/RTC_IDR configuration registers if WPKEY corresponds to 0x535943 (“SYC” in ASCII).

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables the write protection of the configuration registers if WPKEY corresponds to 0x535943 (“SYC” in ASCII).
1	Enables the write protection of the configuration registers if WPKEY corresponds to 0x535943 (“SYC” in ASCII).

Downloaded from Arrow.com.

12.2.2 SYSC Write Protection Status Register

Name: SYSC_WPSR
Offset: 0x04
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 15:8 – WVSRC[7:0] Write Violation Source
When bit WPVS is equal to 1, the field WVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Register Violation Status
WDT_CR, WDT_MR, RTT_MODR, RTC_IDR and RTC_IER can be write-protected but WPVS does not report any violation for these registers.

Value	Description
0	No write protection violation has occurred since the last read of SYSC_WPSR.
1	A write protection violation has occurred since the last read of SYSC_WPSR. The associated violation is reported into field WVSRC.

Downloaded from Arrow.com.

13. General Purpose Backup Registers (GPBR)

13.1 Description

The System Controller embeds 256 bits of General Purpose Backup registers organized as 8 32-bit registers.

It is possible to generate an immediate clear of the content of General Purpose Backup registers 0 to 3 if a tamper event is detected on WKUP1 to WKUP8 pins. These pins are internally routed through the VDDCORE area, thus tamper events can be generated only when the VDDCORE is powered. These pins are also used for fast wakeup in Power Management Controller (PMC). Thus, if some WKUP pins are not enabled for fast wakeup in PMC, they can be enabled for tamper event detection. The immediate clear of the GPBR is enabled if RSTC_MR.ENGCLR=1.

The immediate clear on tamper detection can be extended to all General Purpose Backup registers by writing to ‘1’ GPBR_FCLR.FCLR.

If an event has been detected on WKUP pins enabled for event detection in RTC, it is not possible to write to the General Purpose Backup registers (SYS_GPBRx) while the event has not been cleared.

SYS_GPBR0 to SYS_GPBR7 can be individually (each 32-bit part-select) read and write-protected by configuring the register GPBR_MR. This register is write-once, which means that once it has been configured, the read or write protection is available until the loss of VDDBU.

13.2 Embedded Characteristics

- 256 bits of General Purpose Backup Registers
- Immediate Clear on WKUP Event
- Read and Write Protection for SYS_GPBR0 to SYS_GPBR7

Downloaded from Arrow.com.

13.3 Register Summary

Offset	Name	Bit Pos.								
0x00	GPBR_MR	31:24	GPBRRP15	GPBRRP14	GPBRRP13	GPBRRP12	GPBRRP11	GPBRRP10	GPBRRP9	GPBRRP8
		23:16	GPBRRP7	GPBRRP6	GPBRRP5	GPBRRP4	GPBRRP3	GPBRRP2	GPBRRP1	GPBRRP0
		15:8	GPBRWP15	GPBRWP14	GPBRWP13	GPBRWP12	GPBRWP11	GPBRWP10	GPBRWP9	GPBRWP8
		7:0	GPBRWP7	GPBRWP6	GPBRWP5	GPBRWP4	GPBRWP3	GPBRWP2	GPBRWP1	GPBRWP0
0x04	GPBR_FCLR	31:24								
		23:16								
		15:8								
		7:0								FCLR
0x08	SYS_GPBR0	31:24	GPBR_VALUE[31:24]							
		23:16	GPBR_VALUE[23:16]							
		15:8	GPBR_VALUE[15:8]							
		7:0	GPBR_VALUE[7:0]							
0x0C	SYS_GPBR1	31:24	GPBR_VALUE[31:24]							
		23:16	GPBR_VALUE[23:16]							
		15:8	GPBR_VALUE[15:8]							
		7:0	GPBR_VALUE[7:0]							
0x10	SYS_GPBR2	31:24	GPBR_VALUE[31:24]							
		23:16	GPBR_VALUE[23:16]							
		15:8	GPBR_VALUE[15:8]							
		7:0	GPBR_VALUE[7:0]							
0x14	SYS_GPBR3	31:24	GPBR_VALUE[31:24]							
		23:16	GPBR_VALUE[23:16]							
		15:8	GPBR_VALUE[15:8]							
		7:0	GPBR_VALUE[7:0]							
0x18	SYS_GPBR4	31:24	GPBR_VALUE[31:24]							
		23:16	GPBR_VALUE[23:16]							
		15:8	GPBR_VALUE[15:8]							
		7:0	GPBR_VALUE[7:0]							
0x1C	SYS_GPBR5	31:24	GPBR_VALUE[31:24]							
		23:16	GPBR_VALUE[23:16]							
		15:8	GPBR_VALUE[15:8]							
		7:0	GPBR_VALUE[7:0]							
0x20	SYS_GPBR6	31:24	GPBR_VALUE[31:24]							
		23:16	GPBR_VALUE[23:16]							
		15:8	GPBR_VALUE[15:8]							
		7:0	GPBR_VALUE[7:0]							
0x24	SYS_GPBR7	31:24	GPBR_VALUE[31:24]							
		23:16	GPBR_VALUE[23:16]							
		15:8	GPBR_VALUE[15:8]							
		7:0	GPBR_VALUE[7:0]							

Downloaded from Arrow.com.

13.3.1 GPBR Mode Register

Name: GPBR_MR
Offset: 0x0
Reset: 0x00000000
Property: Read/Write-Once

This register is write-once. All bits are cleared at first power-up and on each loss of VDDBU.

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
	GPBRRP15	GPBRRP14	GPBRRP13	GPBRRP12	GPBRRP11	GPBRRP10	GPBRRP9	GPBRRP8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GPBRRP7	GPBRRP6	GPBRRP5	GPBRRP4	GPBRRP3	GPBRRP2	GPBRRP1	GPBRRP0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GPBRWP15	GPBRWP14	GPBRWP13	GPBRWP12	GPBRWP11	GPBRWP10	GPBRWP9	GPBRWP8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GPBRWP7	GPBRWP6	GPBRWP5	GPBRWP4	GPBRWP3	GPBRWP2	GPBRWP1	GPBRWP0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – GPBRRPx GPBRx Read Protection

Value	Description
0	The content of the corresponding GPBR register (32-bit part-select) can be read.
1	The corresponding GPBR register (32-bit part-select) always returns zero when read.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – GPBRWPx GPBRx Write Protection

Value	Description
0	The corresponding GPBR register (32-bit part-select) can be written.
1	The corresponding GPBR register (32-bit part-select) is write-protected.

Downloaded from Arrow.com.

13.3.2 GPBR Full Clear Register

Name: GPBR_FCLR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

All bits are cleared at first power-up and on each loss of VDDBU.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								FCLR
Access								R/W
Reset								0

Bit 0 – FCLR Full Clear Enable
GPBR full clear is only possible if the system is not in Backup mode. In Backup mode, FCLR has no effect.

Value	Description
0	SYS_GPBR0 to SYS_GPBR3 are immediately cleared in case of fast wakeup pin tamper event.
1	All SYS_GPBRx are immediately cleared in case of fast wakeup pin tamper event.

Downloaded from Arrow.com.

13.3.3 General Purpose Backup Register x [x=0..7]

Name: SYS_GPBRx
Offset: 0x08 + x*0x04 [x=0..7]
Reset: 0x00000000
Property: R/W

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

These registers are reset at first power-up and on each loss of VDDBU.

Bit	31	30	29	28	27	26	25	24
	GPBR_VALUE[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GPBR_VALUE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GPBR_VALUE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GPBR_VALUE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GPBR_VALUE[31:0] Value of SYS_GPBRx
Note: If an event has been detected on WKUP pins enabled for tamper event detection in RTC, it is not possible to write to the General Purpose Backup registers (SYS_GPBRx) while the event has not been cleared.

Downloaded from Arrow.com.

14. Watchdog Timer (WDT)

14.1 Description

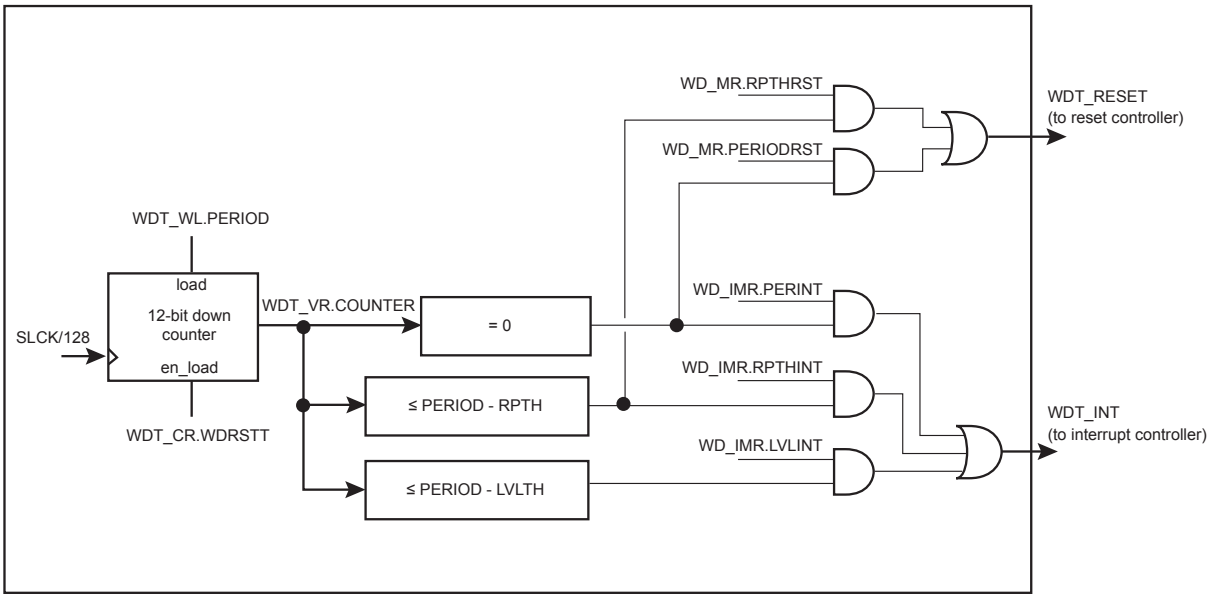
The Watchdog Timer (WDT) is used to prevent system lockup if the software becomes trapped in a deadlock. It features one 12-bit down counter that allows a watchdog period of up to 16 seconds (slow clock around 32 kHz). The WDT can generate a general reset or a processor reset only. In addition, it can be stopped while the processor is in Debug mode or Sleep mode (Idle mode).

14.2 Embedded Characteristics

- 12-bit Key-protected Programmable Counter
- Watchdog Clock is Independent from Processor Clock
- Provides Reset or Interrupt Signals to the System
- Counter May Be Stopped while the Processor is in Debug State or in Idle Mode

14.3 Block Diagram

Figure 14-1. WDT Block Diagram



14.4 Functional Description

The WDT is used to prevent system lockup if the software becomes trapped in a deadlock. It is supplied with VDDCORE. It restarts with initial values on processor reset.

The WDT is built around a 12-bit down counter loaded with the value defined in field PERIOD of the Window Level Register (WDT_WLR). WDT uses slow clock divided by 128 to establish the maximum watchdog period to 16 seconds (with a typical slow clock of 32.768 kHz).

The following parameters can be configured:

Downloaded from Arrow.com.

- Watchdog event period: a watchdog event occurs when the 12-bit down counter reaches 0, and leads to either an interrupt (if bit PERINT in the Interrupt Mask register (WDT_IMR) is high) or a reset (if bit PERIODRST in the Mode register (WDT_MR) is high).
- Minimum restart period: if the restart command is performed before this period, WDT creates a repeat violation. A repeat violation leads to either an interrupt (if WDT_IMR.RPTHINT = 1) or a reset (if WDT_MR.RPTHRST = 1).
- Maximum period before single interrupt event: if WDT_IMR.LVLINT = 1, a single interrupt is generated (no reset). The WDT_ILR.LVLTH value must be lower than WDT_WLR.PERIOD.

After a processor reset, the value of PERIOD is 0xFFF, corresponding to the maximum value of the counter with the external reset generation enabled (bit PERIODRST at 1 after a backup reset). This means that the WDT is running at reset, i.e., at powerup. The user can either disable the WDT by setting bit WDT_MR.WDDIS to '1' or reprogram the WDT to meet the maximum WDT period the application requires.

If the WDT is restarted by writing into the corresponding Control register (WDT_CR), the corresponding WDT_MR must not be programmed during a period of time of three slow clock periods following the WDT_CR write access. In any case, programming a new value in WDT_MR automatically initiates a restart instruction.

WDT_MR, WDT_WLR and WDT_ILR (Interrupt Level register) can be written until a WDT_CR.LOCKMR command is issued in the corresponding WDT_CR. Only a peripheral reset can configure the bit LOCKMR to 0.

When the bit WDT_CR.LOCKMR = 0, writing WDT_WLR reloads the corresponding WDT with the newly programmed mode parameters.

In normal operation, the user reloads the WDT at regular intervals before the timer underflow occurs, by setting bit WDT_CR.WDRSTT. The WDT counter is then immediately reloaded from PERIOD and restarted, and the slow clock 128 divider is reset and restarted.

Writing WDR_CR without the correct hard-coded key has no effect (see Watchdog Timer Control Register).

A repeat threshold can be defined for each watchdog in order to protect against dead-locks that would repeatedly restart the watchdog. WDT_WLR.RPTH defines the minimum number of cycles to wait after a watchdog restart before the WDT can be started again. If a watchdog restart occurs before this limit is reached, a repeat threshold failure is asserted and the RPTHINT bit in the Interrupt Status register (WDT_ISR) is set to one.

If WDT_IMR.RPTHINT is high and a repeat threshold violation occurs in the WDT, an interrupt is generated.

If WDT_MR.RPTHRST is high and a repeat threshold violation occurs in the WDT, a watchdog reset is generated.

WDT reload must occur while the WDT counter is within a window between 0 and (PERIOD–RPTH). PERIOD and RPTH are defined in WDT_WLR.

Note that this feature can be disabled by programming a null RPTH value. In such configuration, restarting the WDT is permitted in the whole range [0 up to PERIOD] and does not generate an error. This is the default configuration on reset (RPTH is null).

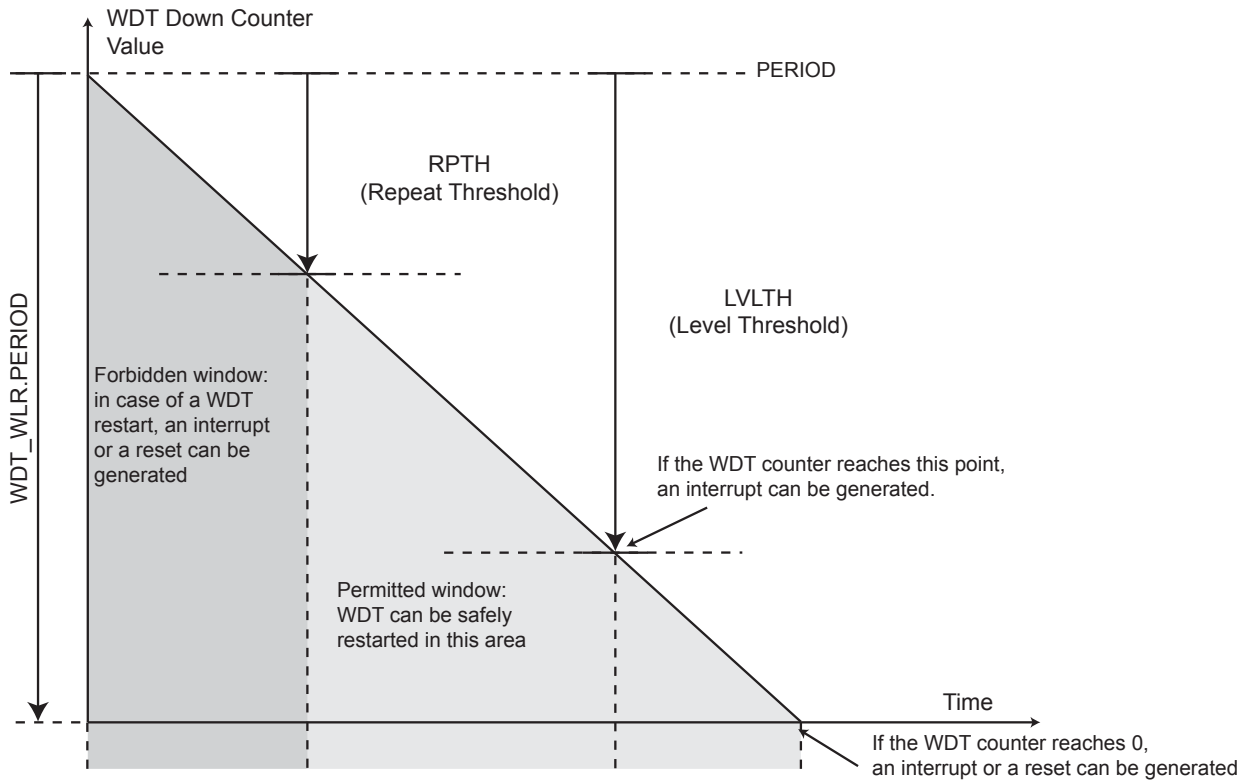
If a reset is generated or if WDT_SR is read, the status bits are reset and the interrupt is cleared.

Writing WDT_MR reloads and restarts the down counter.

While the processor is in debug state or in Sleep mode, the counter may be stopped depending on the value programmed for the bits WDIDLEHLT and WDDBGHLT in WDT_MR.

Downloaded from Arrow.com.

Figure 14-2. Watchdog Timing Diagram



Downloaded from Arrow.com.

14.5 Register Summary

Offset	Name	Bit Pos.								
0x00	WDT_CR	31:24	KEY[7:0]							
		23:16								
		15:8								
		7:0				LOCKMR				WDRSTT
0x04	WDT_MR	31:24			WDIDLEHLT	WDDBGHLT				
		23:16								
		15:8				WDDIS				
		7:0			RPTHRST	PERIODRST				
0x08	WDT_VR	31:24								
		23:16								
		15:8					COUNTER[11:8]			
		7:0	COUNTER[7:0]							
0x0C	WDT_WLR	31:24					RPTH[11:8]			
		23:16	RPTH[7:0]							
		15:8					PERIOD[11:8]			
		7:0	PERIOD[7:0]							
0x10	WDT_ILR	31:24								
		23:16								
		15:8					LVLTH[11:8]			
		7:0	LVLTH[7:0]							
0x14	WDT_IER	31:24								
		23:16								
		15:8								
		7:0						LVLINT	RPTHINT	PERINT
0x18	WDT_IDR	31:24								
		23:16								
		15:8								
		7:0						LVLINT	RPTHINT	PERINT
0x1C	WDT_ISR	31:24								
		23:16								
		15:8								
		7:0						LVLINT	RPTHINT	PERINT
0x20	WDT_IMR	31:24								
		23:16								
		15:8								
		7:0						LVLINT	RPTHINT	PERINT

Downloaded from Arrow.com.

14.5.1 Watchdog Timer Control Register

Name: WDT_CR
Offset: 0x00
Reset: –
Property: Write-only

The WDT_CR register values must not be modified within three slow clock periods following a restart of the WDT performed by a write access in WDT_CR. Any modification will cause the WDT to trigger an end of period earlier than expected.

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				LOCKMR				WDRSTT
Access				W				W
Reset				–				–

Bits 31:24 – KEY[7:0] Password

Value	Name	Description
0xA5	PASSWD	Writing any other value in this field aborts the write operation.

Bit 4 – LOCKMR Lock Mode Register Write Access

Value	Description
0	No effect.
1	Locks the configuration registers if KEY is written to 0xA5. Write accesses to WDT_MR, WDT_WLR and WDT_ILR have no effect.

Bit 0 – WDRSTT Watchdog Restart

Value	Description
0	No effect.
1	Restarts the WDT if KEY is written to 0xA5.

Downloaded from Arrow.com.

14.5.2 Watchdog Timer Mode Register

Name: WDT_MR
Offset: 0x04
Reset: 0x00000030
Property: Read/Write

Write access to this register has no effect if the LOCKMR command is issued in WDT_CR (unlocked on hardware reset).

The WDT_MR register values must not be modified within three slow clock periods following a restart of the WDT performed by a write access in WDT_CR. Any modification will cause the WDT to trigger an end of period earlier than expected.

Bit	31	30	29	28	27	26	25	24
			WDIDLEHLT	WDDBGHLT				
Access			R/W	R/W				
Reset			0	0				

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				WDDIS				
Access				R/W				
Reset				0				

Bit	7	6	5	4	3	2	1	0
			RPTHRST	PERIODRST				
Access			R/W	R/W				
Reset			1	1				

Bit 29 – WDIDLEHLT Watchdog Idle Halt

Value	Description
0	The WDT runs when the system is in idle state.
1	The WDT stops when the system is in idle state.

Bit 28 – WDDBGHLT Watchdog Debug Halt

Value	Description
0	The WDT runs when the processor is in debug state.
1	The WDT stops when the processor is in debug state.

Bit 12 – WDDIS Watchdog Disable

Value	Description
0	Enables the WDT.
1	Disables the WDT.

Bit 5 – RPTHRST Minimum Restart Period

Value	Description
0	No reset is generated if the WDT is restarted before the RPTH threshold.
1	A reset is generated if the WDT is restarted before the RPTH threshold.

Bit 4 – PERIODRST Period Reset

Value	Description
0	No reset is generated if the WDT down counter reaches 0.

Downloaded from Arrow.com.

Value	Description
1	A reset is generated once the WDT down counter reaches 0.

Downloaded from Arrow.com.

14.5.3 Watchdog Timer Value Register

Name: WDT_VR
Offset: 0x08
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					COUNTER[11:8]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNTER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – COUNTER[11:0] Watchdog Down Counter Value
Shows the current value of the WDT down counter for debug operation.
Due to the asynchronous operation of the WDT with respect to the rest of the chip, to be certain that the value read in this register is valid and stable, it is necessary to read the register twice. If the data is the same both times, then it is valid. Therefore, a minimum of two and a maximum of three accesses are required.

Downloaded from Arrow.com.

14.5.4 Watchdog Timer Window Level Register

Name: WDT_WLR
Offset: 0x0C
Reset: 0x00000FFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					RPTH[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RPTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					PERIOD[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PERIOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 27:16 – RPTH[11:0] Repeat Threshold
Defines the period before which a WDT restart generates an interrupt.

Bits 11:0 – PERIOD[11:0] Watchdog Period
Defines the period after which the WDT generates a reset.

Downloaded from Arrow.com.

14.5.5 Watchdog Timer Interrupt Level Register

Name: WDT_ILR
Offset: 0x10
Reset: 0x00000FFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					LVLTH[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	7	6	5	4	3	2	1	0
	LVLTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 11:0 – LVLTH[11:0] Level Threshold
Defines the period after which the WDT generates an interrupt.

Downloaded from Arrow.com.

14.5.6 Watchdog Interrupt Enable Register

Name: WDT_IER
Offset: 0x14
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						LVLINT	RPTHINT	PERINT
Access						W	W	W
Reset						–	–	–

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Enable

Bit 1 – RPTHINT Repeat Threshold Interrupt Enable

Bit 0 – PERINT Period Interrupt Enable

Downloaded from Arrow.com.

14.5.7 Watchdog Interrupt Disable Register

Name: WDT_IDR
Offset: 0x18
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						LVLINT	RPTHINT	PERINT
Access						W	W	W
Reset						–	–	–

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Disable

Bit 1 – RPTHINT Repeat Threshold Interrupt Disable

Bit 0 – PERINT Period Interrupt Disable

Downloaded from Arrow.com.

14.5.8 Watchdog Interrupt Status Register

Name: WDT_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						LVLINT	RPTHINT	PERINT
Access						R	R	R
Reset						0	0	0

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Status (cleared on read)

Value	Description
0	No level threshold failure has occurred in the WDT since the last read of WDT_ISR.
1	At least one level threshold failure has occurred in the WDT since the last read of WDT_ISR.

Bit 1 – RPTHINT Repeat Threshold Interrupt Status (cleared on read)

Value	Description
0	No repeat threshold failure has occurred in the WDT since the last read of WDT_ISR.
1	At least one repeat threshold failure has occurred in the WDT since the last read of WDT_ISR.

Bit 0 – PERINT Period Interrupt Status (cleared on read)

Value	Description
0	No period failure has occurred in the WDT since the last read of WDT_ISR.
1	At least one period failure has occurred in the WDT since the last read of WDT_ISR.

Downloaded from Arrow.com.

14.5.9 Watchdog Interrupt Mask Register

Name: WDT_IMR
Offset: 0x20
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						LVLINT	RPTHINT	PERINT
Access						R	R	R
Reset						0	0	0

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Mask

Bit 1 – RPTHINT Repeat Threshold Interrupt Mask

Bit 0 – PERINT Period Interrupt Mask

Downloaded from Arrow.com.

15. Reset Controller (RSTC)

15.1 Description

The Reset Controller (RSTC) handles all the resets of the system without any external components. It reports which reset occurred last.

The RSTC is driven by Power-on Reset (POR) cells, software, an external reset pin, and peripheral events.

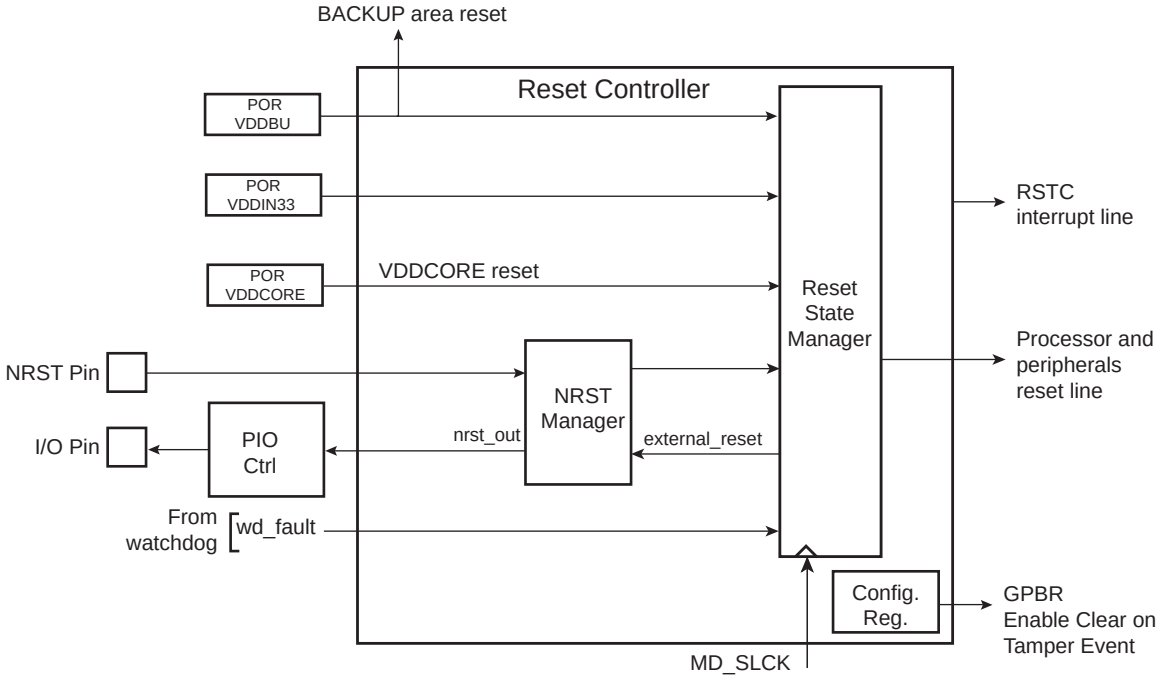
The RSTC drives simultaneously the External reset and the Peripheral and Processor resets.

15.2 Embedded Characteristics

- Driven by Embedded Power-on Reset, Software, External Reset Pin and Peripheral Events
- Management of All System Resets, Including
  - External devices through an I/O multiplexed output reset pin
  - Processor
  - Peripheral set
- Reset Source Status
  - Status of the last reset
  - Either VDDCORE, VDDIN33 and VDDBU POR reset, Software reset, User reset, Watchdog reset, 32.768 kHz Crystal Oscillator Failure Detection reset

15.3 Block Diagram

Figure 15-1. RSTC Block Diagram



Downloaded from Arrow.com.

15.4 Functional Description

15.4.1 Overview

The RSTC is made up of an NRST manager and a reset state manager. The RSTC clock is MD_SLCK (monitoring domain slow clock). The RSTC generates the following reset signals:

- Processor reset line (also resets the Watchdog Timer)
- Entire set of embedded peripherals reset line
- NRST pin

Note: Processor and peripheral reset lines are driven in the same way.

These internal reset signals are asserted by the RSTC, either on events generated by peripherals, events on NRST pin, or on software action. The reset state manager controls the generation of reset signals and drives the NRST pin when required.

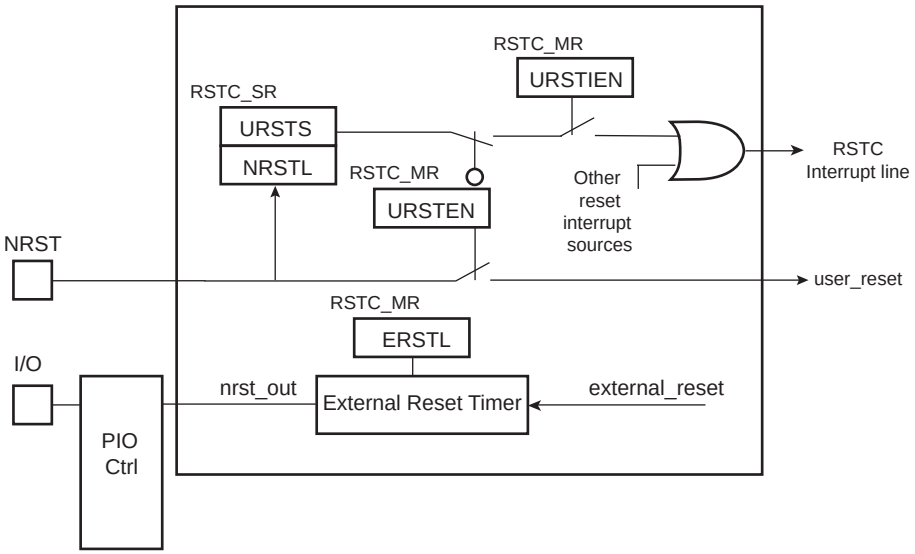
The NRST manager asserts the NRST pin during a programmable time, thus controlling external device resets.

The Mode register (RSTC_MR), used to configure the RSTC, is powered by VDDBU.

15.4.2 NRST Manager

The NRST manager samples the NRST pin and drives this pin low when required by the reset state manager. See the following figure.

Figure 15-2. NRST Pin Management



15.4.2.1 NRST Signal or Interrupt

The NRST manager handles the NRST input line asynchronously if RSTC_MR.URSTASYNC =1. When the NRST input is low, a user reset is immediately reported to the Reset State manager and the internal reset signals are asserted even if there is a clock failure on MD_SLCK (safe reset).

The NRST manager handles the NRST input line synchronously if RSTC_MR.URSTASYNC=0. When the line is low, it is first resynchronized on slow clock before it is reported to the Reset State manager. In both cases, when the NRST goes from low to high, the internal reset is synchronized with the monitoring slow clock to provide a safe internal de-assertion of reset (if enabled).

If RSTC_MR.URSTEN=0, the assertion of the NRST input pin does not trigger a VDDCORE domain reset.

The level of the pin NRST is reported in NRSTL of the Status register (RSTC_SR).

As soon as the pin NRST is asserted (low level), RSTC_SR.URSTS=1. This bit is cleared on read.

If RSTC_MR.URSTIEN=1, the assertion of NRST pin triggers an interrupt rather than a VDDCORE reset.

Downloaded from Arrow.com.

15.4.2.2 NRST External Reset Control

The RSTC can be configured to assert the external reset line (NRST). The NRST pin is driven low for a time programmed by RSTC_MR.ERSTL. This assertion duration lasts $2^{(ERSTL+1)}$ MD_SLCK cycles. This assertion duration time is in the range of 60 µs to 2 seconds. If ERSTL=0 , a two slow clock period duration is generated on the NRST pin.

This feature allows the NRST line to be compliant with any external devices connected on the system reset (i.e., when external devices require a longer start-up time than the processor system).

15.4.3 Reset States

The reset state manager handles the different reset sources and generates the internal reset signals. It reports the reset status in RSTC_SR.RSTTYP. RSTC_SR.RSTTYP is updated when the Processor reset is released.

If more than one reset event occurred since the last read of RST_SR, the field RSTTYP reports the first reset that occurred.

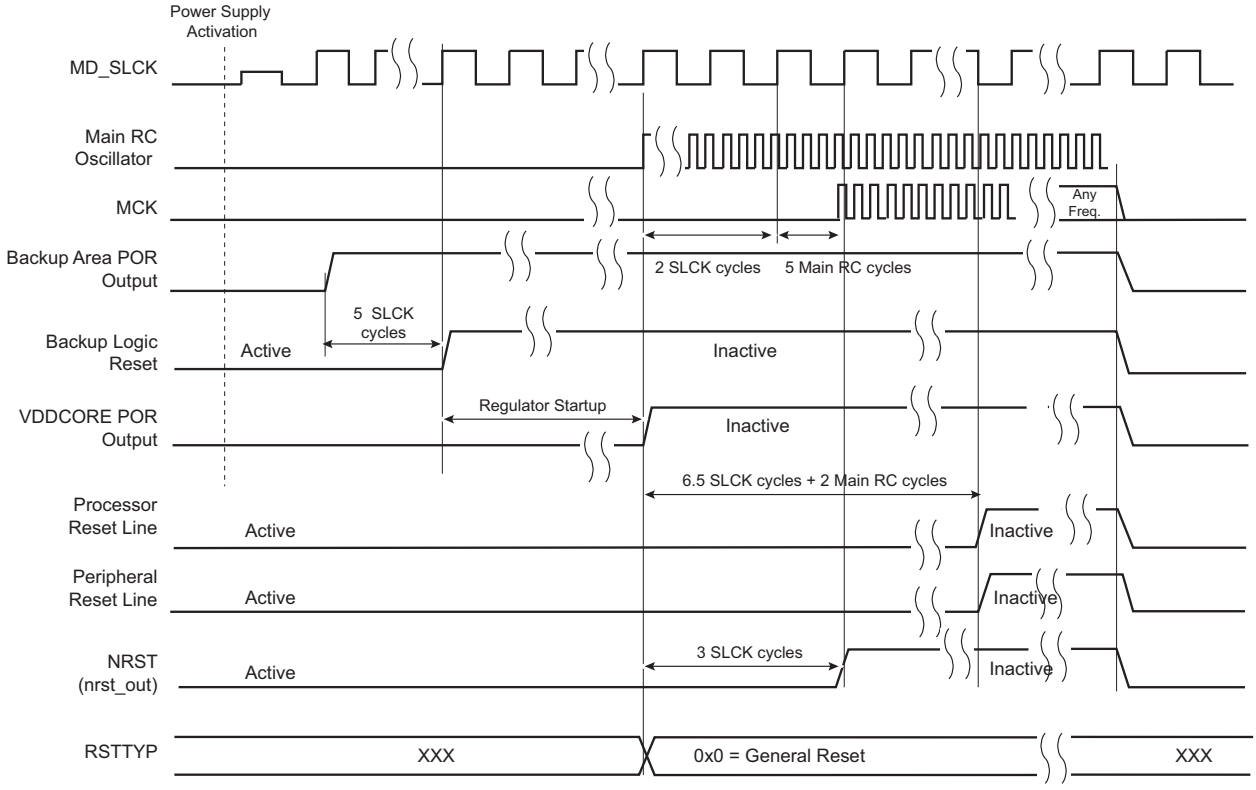
15.4.3.1 General Reset

A general reset occurs when a VDDBU Power-on reset is detected. The internal VDDCORE reset signal is asserted when a general reset occurs. All the reset signals are released and RSTC_SR.RSTTYP reports a general reset.

The NRST line rises two cycles after the VDDCORE reset line, as ERSTL defaults at value 0x0.

The following figure shows how the general reset affects the reset signals.

Figure 15-3. General Reset Timing Diagram



15.4.3.2 Backup Exit Reset

A Backup reset occurs when the chip exits from Backup mode. While exiting Backup mode, the VDDCORE reset signal is de-asserted.

RSTC_SR.RSTTYP is updated to report a Backup reset.

Downloaded from Arrow.com.

15.4.3.3 32.768 kHz Crystal Oscillator Failure Detection Reset

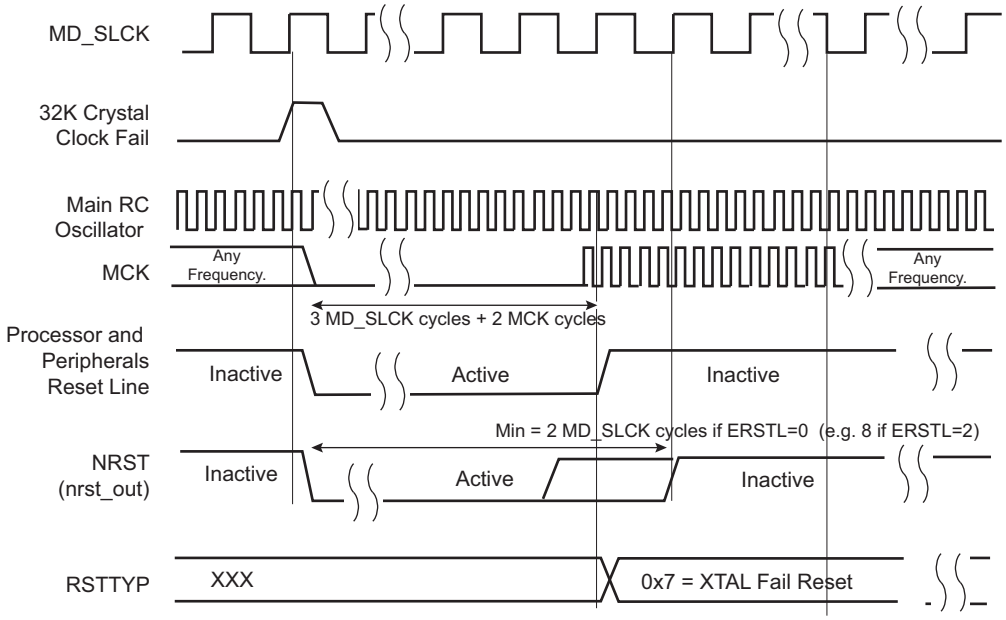
The 32.768 kHz Crystal Oscillator Failure Detection reset is done when the 32.768 kHz crystal oscillator frequency monitoring circuitry in the PMC detects a failure and RSTC_MR.SCKSW is written to '1'. This reset lasts three slow clock cycles.

When RSTC_MR.SCKSW is written to ‘0’, the 32.768 kHz crystal oscillator fault has no impact on the RSTC.

During the 32.768 kHz Crystal Oscillator Failure Detection reset, the Processor reset and the Peripheral reset are asserted. The NRST line is also asserted, depending on the value of RSTC_MR.ERSTL.

When the 32.768 kHz crystal oscillator failure generates a VDDCORE reset, PMC_SR.XT32KERR is automatically cleared by the Peripheral and Processor resets.

Figure 15-4. 32.768 kHz Crystal Oscillator Failure Detection Reset Timing Diagram



15.4.3.4 Watchdog Reset

The Watchdog reset is entered when a watchdog fault occurs. This reset lasts three MD_SLCK cycles.

When in Watchdog reset, the Processor reset and the Peripheral reset are asserted. The NRST line is also asserted, depending on the value of RSTC_MR.ERSTL. However, the resulting low level on NRST does not result in a User reset state.

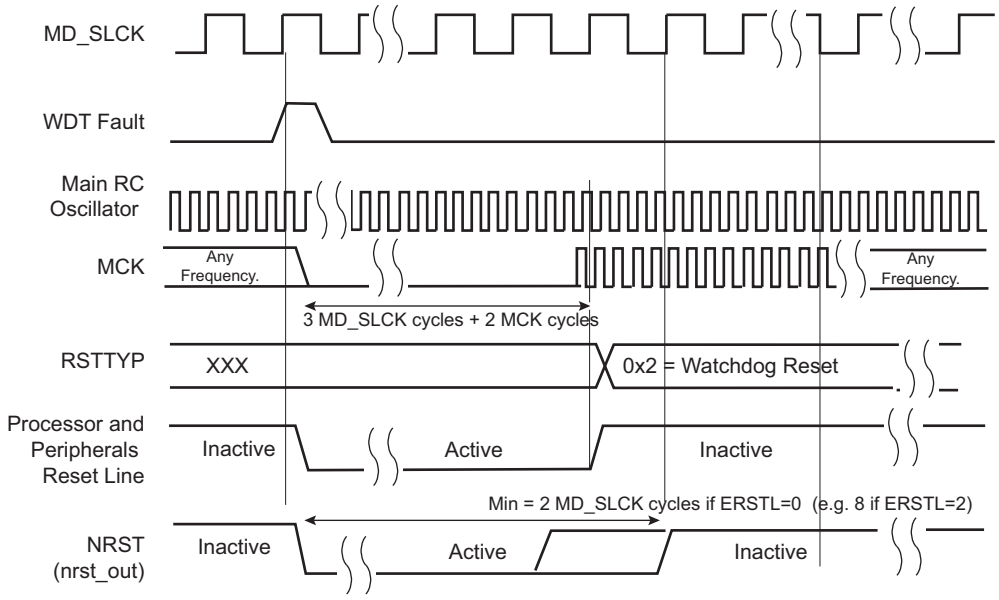
The WDT is reset by the Processor reset signal. As the watchdog fault always causes a Processor reset if WDT_MR.WDRSTEN is written to ‘1’, the WDT is always reset after a Watchdog reset, and the Watchdog is enabled by default and with a period set to a maximum.

When WDT_MR.WDRSTEN is written to ‘0’, the watchdog fault has no impact on the RSTC.

After a watchdog overflow occurs, the report on the RSTC_SR.RSTTYP may differ (either WDT_RST or USER_RST) depending on the external components driving the NRST pin. For example, if the NRST line is driven through a resistor and a capacitor (NRST pin debouncer), the reported value is USER_RST if the low-to-high transition is greater than one MD_SLCK cycle.

Downloaded from Arrow.com.

Figure 15-5. Watchdog Reset Timing Diagram



15.4.3.5 Software Reset

The RSTC offers commands to assert the different reset signals. These commands are performed by writing the Control register (RSTC_CR) with the following bits at ‘1’:

- RSTC_CR.PROCRST: Writing a ‘1’ to PROCRST resets the processor and all the embedded peripherals, including the memory system and, in particular, the Remap Command.
- RSTC_CR.EXTRST: Writing a ‘1’ to EXTRST asserts low the NRST pin during a time defined by the field RSTC_MR.ERSTL.

The Software reset is entered if at least one of these bits is written to ‘1’ by the software. All these commands can be performed independently or simultaneously. The Software reset lasts three MD_SLCK cycles.

The internal reset signals are asserted as soon as the register write is performed. This is detected on the Master Clock (MCK). They are released when the Software reset has ended, i.e., synchronously to MD_SLCK.

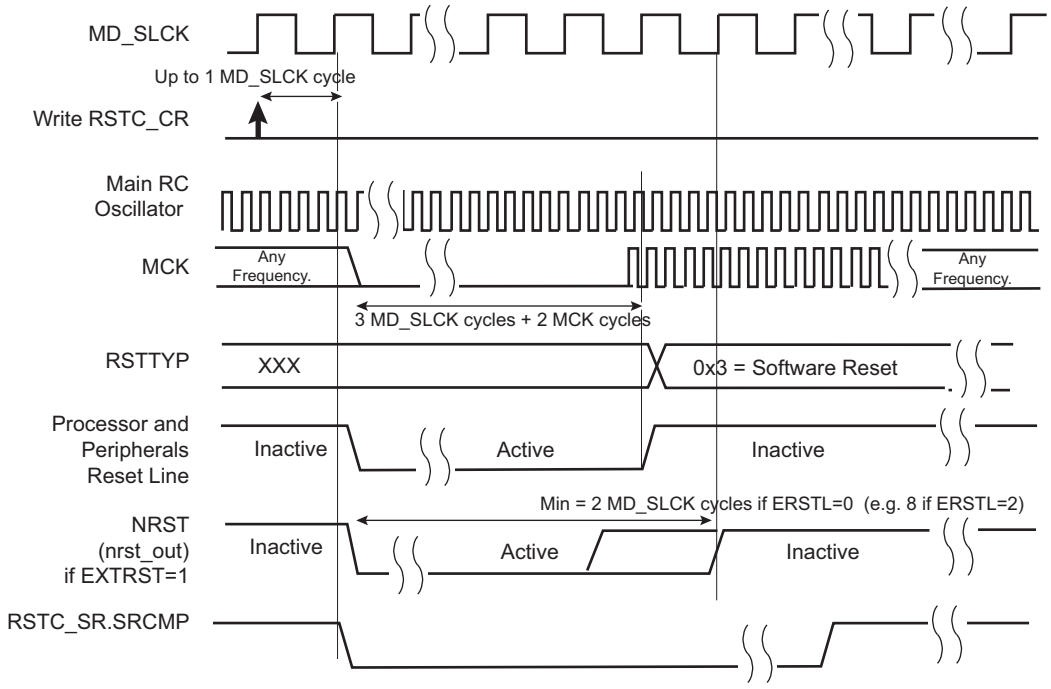
If EXTRST is written to ‘1’, the nrst_out signal is asserted depending on the configuration of RSTC_MR.ERSTL. However, the resulting falling edge on NRST does not lead to a User reset.

If and only if the RSTC_CR.PROCRST is written to ‘1’, the RSTC reports the software status in field RSTC_SR.RSTTYP. Other software resets are not reported in RSTTYP.

As soon as a software operation is detected, RSTC_SR.SRCMP is written to ‘1’. SRCMP is cleared at the end of the Software reset. No other Software reset can be performed while SRCMP=‘1’, and writing any value in the RSTC_CR has no effect.

Downloaded from Arrow.com.

Figure 15-6. Software Reset Timing Diagram



15.4.3.6 User Reset

The User reset is entered when a low level is detected on the NRST pin and RSTC_MR.URSTEN =1. If URSTASYNC=1, a falling edge of the NRST input signal immediately asserts internal reset lines. If URSTASYNC=0, the NRST input signal is resynchronized and internal reset lines are asserted once a falling edge has been detected on the resynchronized NRST input signal.

The Processor reset and the Peripheral reset are asserted.

The User reset is released when NRST rises, after a two-cycle resynchronization time and a 2-cycle processor start-up. The processor clock is re-enabled as soon as NRST is confirmed high.

When the Processor reset signal is released, RSTC_SR.RSTTYP is loaded with the value 0x4, indicating a User reset.

The NRST manager ensures that the NRST line is asserted as programmed in the field ERSTL. However, if NRST is driven low externally, the internal reset lines remain asserted until NRST rises.

Downloaded from Arrow.com.

Figure 15-7. User Reset State (URSTASYNC = '0')

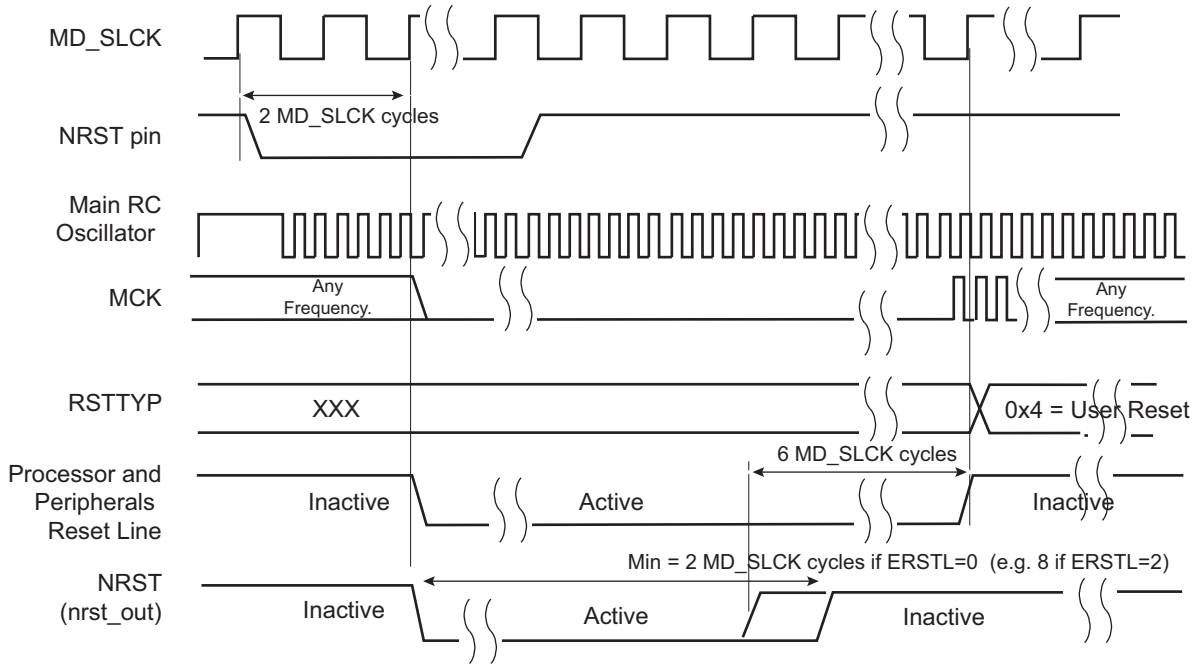
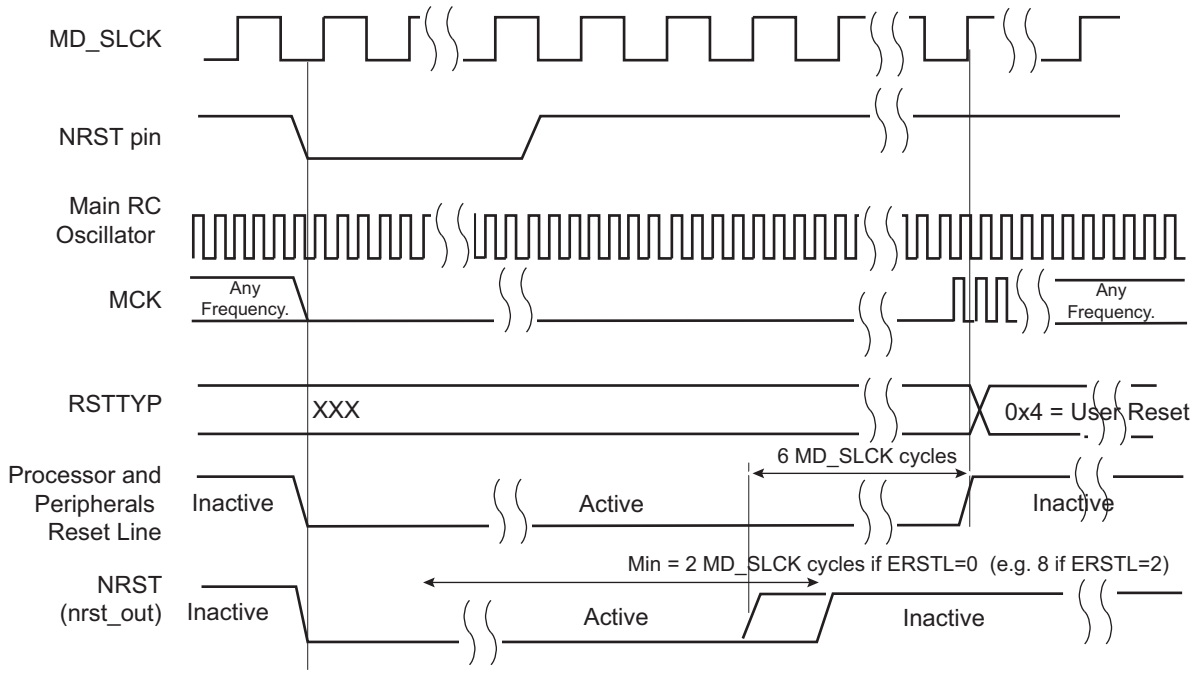


Figure 15-8. User Reset State (URSTASYNC = '1')



15.4.4 Reset State Priorities

The reset state manager manages the priorities among the different reset sources. The resets are listed in order of priority as follows:

1. General reset
2. Backup reset
3. 32.768 kHz Crystal Failure Detection reset
4. Watchdog reset

Downloaded from Arrow.com.

5. Software reset
6. User reset

Specific cases are listed below:

- When in User reset:
  - A watchdog event is impossible because the WDT is being reset by the Processor reset signal.
  - A Software reset is impossible, since the Processor reset is being activated.
- When in Software reset:
  - A watchdog event has priority over the current state.
  - The NRST has no effect.
- When in Watchdog reset:
  - The Processor reset is active and so a Software reset cannot be programmed.
  - A User reset cannot be entered.

Downloaded from Arrow.com.

15.5 Register Summary

Offset	Name	Bit Pos.								
0x00	RSTC_CR	31:24	KEY[7:0]							
		23:16								
		15:8								
		7:0					EXTRST			PROCRST
0x04	RSTC_SR	31:24								
		23:16							SRCMP	NRSTL
		15:8						RSTTYP[2:0]		
		7:0								URSTS
0x08	RSTC_MR	31:24	KEY[7:0]							
		23:16				ENGCLR				
		15:8					ERSTL[3:0]			
		7:0				URSTIEN		URSTASYNC	SCKSW	URSTEN

Downloaded from Arrow.com.

15.5.1 RSTC Control Register

Name: RSTC_CR
Offset: 0x00
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					EXTRST			PROCRST
Access					W			W
Reset					–			–

Bits 31:24 – KEY[7:0] System Reset Key

Value	Name	Description
0xA5	PASSWD	Writing any other value in this field aborts the write operation.

Bit 3 – EXTRST External Reset

Value	Description
0	No effect.
1	If KEY = 0xA5, asserts the NRST pin.

Bit 0 – PROCRST Processor Reset

Value	Description
0	No effect.
1	If KEY = 0xA5, resets the processor and all the embedded peripherals.

Downloaded from Arrow.com.

15.5.2 RSTC Status Register

Name: RSTC_SR
Offset: 0x04
Reset: 0x00000000
Property: Read-only

The reset value assumes that a general reset has been performed, subject to change if other types of reset are generated.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							SRCMP	NRSTL
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
						RSTTYP[2:0]		
Access						R	R	R
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
								URSTS
Access								R
Reset								0

Bit 17 – SRCMP Software Reset Command in Progress
When set, this bit indicates that a software reset command is in progress and that no further software reset should be performed until the end of the current one. This bit is automatically cleared at the end of the current software reset.

Value	Description
0	No software command is being performed by the RSTC. The RSTC is ready for a software command.
1	A software reset command is being performed by the RSTC. The RSTC is busy.

Bit 16 – NRSTL NRST Pin Level
Registers the NRST pin level sampled on each MCK rising edge.

Bits 10:8 – RSTTYP[2:0] Reset Type
This field reports the cause of the last processor reset. Reading RSTC_SR does not reset this field.

Value	Name	Description
0	GENERAL_RST	First power-up reset
1	BACKUP_RST	Return from Backup mode
2	WDT_RST	Watchdog fault occurred
3	SOFT_RST	Processor reset required by the software
4	USER_RST	NRST pin detected low
5	–	Reserved
6	–	Reserved
7	SLCK_XTAL_RST	32.768 kHz crystal failure detection fault occurred

Bit 0 – URSTS User Reset Status
A high-to-low transition of the NRST pin sets URSTS. This transition is also detected on the MCK rising edge. If the user reset is disabled (RSTC_MR.URSTEN = 0) and if the interrupt is enabled by RSTC_MR.URSTIEN, URSTS triggers an interrupt. Reading RSTC_SR resets URSTS and clears the interrupt.

Downloaded from Arrow.com.

Value	Description
0	No high-to-low edge on NRST happened since the last read of RSTC_SR.
1	At least one high-to-low transition of NRST has been detected since the last read of RSTC_SR.

Downloaded from Arrow.com.

15.5.3 RSTC Mode Register

Name: RSTC_MR
Offset: 0x08
Reset: 0x00000001
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	–

Bit	23	22	21	20	19	18	17	16
				ENGCLR				
Access				R/W				
Reset				0				

Bit	15	14	13	12	11	10	9	8
					ERSTL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
				URSTIEN		URSTASYNC	SCKSW	URSTEN
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	1

Bits 31:24 – KEY[7:0] Write Access Password

Value	Name	Description
0xA5	PASSWD	Writing any other value in this field aborts the write operation. Always reads as 0.

Bit 20 – ENGCLR Enable GPBR Clear on Tamper Event

Value	Description
0	Disables the GPBR immediate clear on tamper detection event.
1	Enables the GPBR immediate clear on tamper detection event

Bits 11:8 – ERSTL[3:0] External Reset Length
This field defines the external reset length. The external reset is asserted during a time of $2^{(ERSTL+1)}$ MD_SLCK cycles. This allows assertion duration to be programmed between 60 µs and 2 seconds. Note that synchronization cycles must also be considered when calculating the actual reset length as previously described.

Bit 4 – URSTIEN User Reset Interrupt Enable

Value	Description
0	RSTC_SR.USRTS at ‘1’ has no effect on the RSTC interrupt line.
1	RSTC_SR.USRTS at ‘1’ asserts the RSTC interrupt line if URSTEN = 0.

Bit 2 – URSTASYNC User Reset Asynchronous Control
See 15.4.2.1 NRST Signal or Interrupt for important information on the use of URSTASYNC.

Value	Description
0	The NRST input signal is managed synchronously.

Downloaded from Arrow.com.

Value	Description
1	The NRST input signal is managed asynchronously. **Note:** This mode cannot be selected if the external bus interface drives an SDR/DDR memory device and another memory on the same bus.

Bit 1 – SCKSW Slow Clock Switching

Value	Description
0	The detection of a 32.768 kHz crystal failure has no effect.
1	The detection of a 32.768 kHz crystal failure resets the logic supplied by VDDCORE.

Bit 0 – URSTEN User Reset Enable

Value	Description
0	The detection of a low level on the NRST pin does not generate a user reset.
1	The detection of a low level on the NRST pin triggers a user reset.

Downloaded from Arrow.com.

16. Real-time Timer (RTT)

16.1 Description

The Real-time Timer (RTT) is built around a 32-bit counter used to count roll-over events of the programmable 16-bit prescaler driven from the 32-kHz slow clock source. It generates a periodic interrupt and/or triggers an alarm on a programmed value.

The RTT can also be configured to be driven by the 1Hz RTC signal, thus taking advantage of a calibrated 1Hz clock.

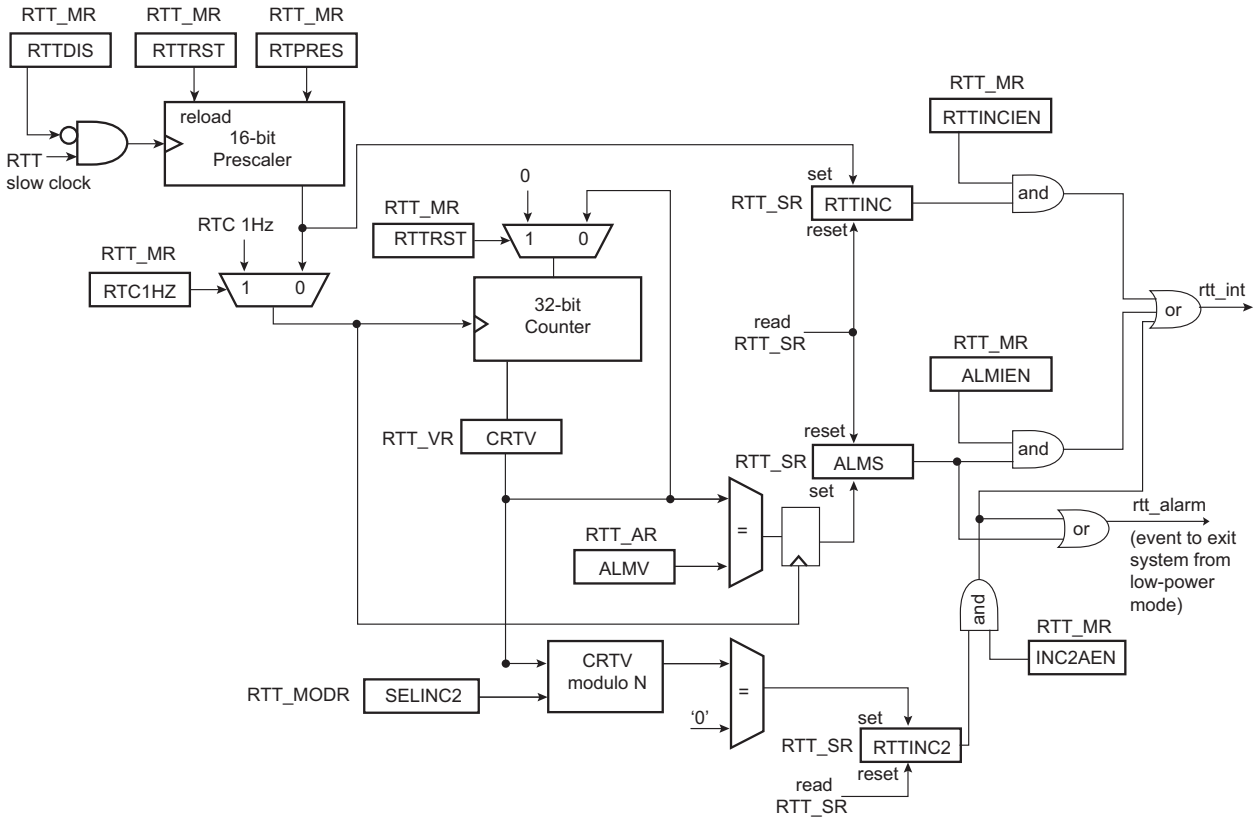
The slow clock source can be fully disabled to reduce power consumption when only an elapsed seconds count is required.

16.2 Embedded Characteristics

- 32-bit Free-running Counter on prescaled slow clock or RTC calibrated 1Hz clock
- 16-bit Configurable Prescaler
- Interrupt on Alarm or Counter Increment
- Programmable Event

16.3 Block Diagram

Figure 16-1. RTT Block Diagram



Downloaded from Arrow.com.

16.4 Functional Description

The programmable 16-bit prescaler value can be configured through the RTPRES field in the RTT Mode register (RTT_MR).

Configuring the RTPRES field value to 0x8000 (default value) corresponds to feeding the real-time counter with a 1Hz signal (if the slow clock is 32.768 kHz). The 32-bit counter can count up to 2^{32} seconds, corresponding to more than 136 years, then roll over to 0. Bit RTTINC in the RTT Status Register (RTT_SR) is set each time there is a prescaler roll-over (see the figure below).

The real-time 32-bit counter can also be supplied by the 1Hz RTC clock. This mode is interesting when the RTC 1Hz is calibrated (CORRECTION field ≠ 0 in RTC_MR) in order to guaranty the synchronism between RTC and RTT counters.

Setting the RTC1HZ bit in the RTT_MR drives the 32-bit RTT counter from the 1Hz RTC clock. In this mode, the RTPRES field has no effect on the 32-bit counter.

The prescaler roll-over generates an increment of the real-time timer counter if RTC1HZ = 0. Otherwise, if RTC1HZ = 1, the RTT counter is incremented every second. The RTTINC bit is set independently from the 32-bit counter increment.

The RTT can also be used as a free-running timer with a lower time-base. The best accuracy is achieved by writing RTPRES to 3 in RTT_MR.

Programming RTPRES to 1 or 2 is forbidden.

The CRTV field can be read at any time in the RTT Value register (RTT_VR). As this value can be updated asynchronously with the Master Clock, the CRTV field must be read twice at the same value to read a correct value.

The current value of the counter is compared with the value written in the RTT Alarm register (RTT_AR). If the counter value matches the alarm, the ALMS bit in the RTT_SR is set. The RTT_AR is set to its maximum value (0xFFFFFFFF) after a reset.

The ALMS flag is always a source of the RTT alarm signal that may be used to exit the system from low power modes (see the Real-time Timer Block Diagram above).

The alarm interrupt must be disabled (ALMIEN must be cleared in RTT_MR) when writing a new ALMV value in the RTT_AR.

The RTTINC bit can be used to start a periodic interrupt, the period being one second when the RTPRES field value = 0x8000 and the slow clock = 32.768 kHz.

The RTTINCIEN bit must be cleared prior to writing a new RTPRES value in the RTT_MR.

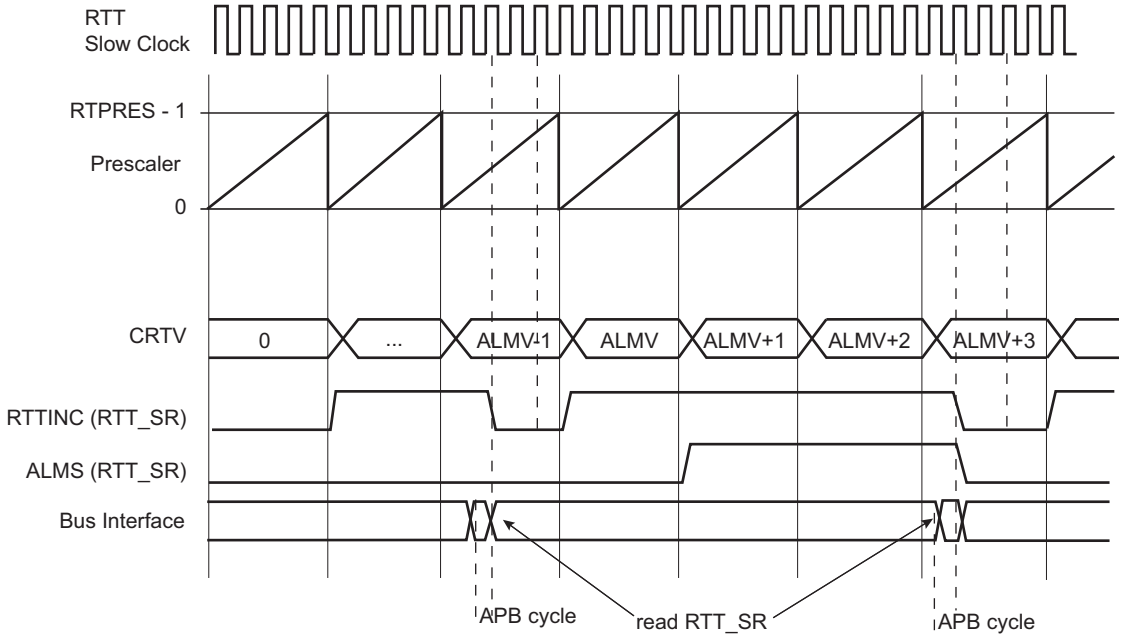
Reading the RTT_SR automatically clears the RTTINC and ALMS bits.

Writing the RTTRST bit in the RTT_MR immediately reloads and restarts the clock divider with the new programmed value. This also resets the 32-bit counter.

When not used, the RTT can be disabled in order to suppress dynamic power consumption in this module. This can be achieved by setting the RTTDIS bit in the RTT_MR.

Downloaded from Arrow.com.

Figure 16-2. RTT Counting



The RTTINC2 flag is set when the number of prescaler roll-overs programmed through the SELINC2 field in the RTT Modulo Selection register (RTT_MODR) has been reached since the last read of the RTT_SR.

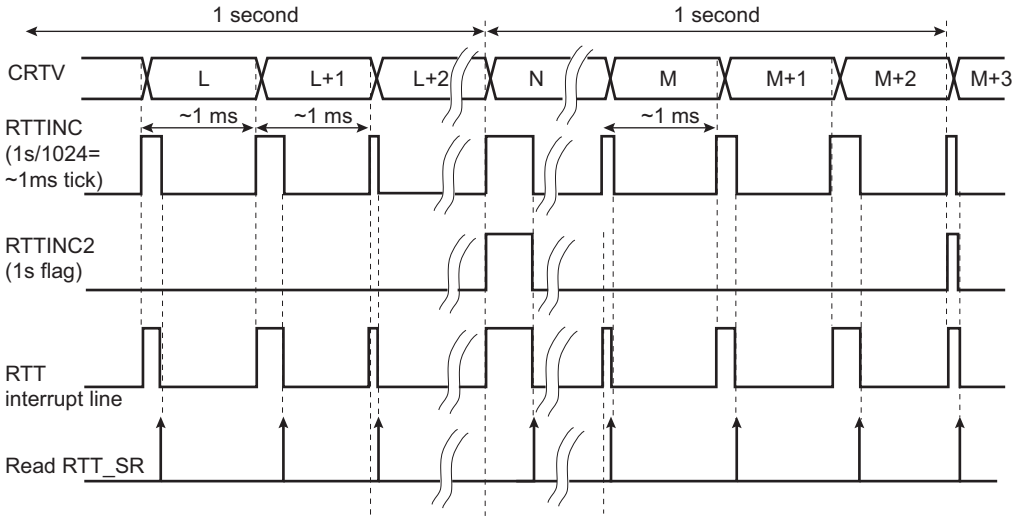
For example, it is possible to generate 2 sources of interrupt of different periods with flags RTTINC and RTTINC2. If the RTT slow clock frequency is 32.768 kHz and RTPRES=32, the RTTINC flag rises 1024 times per second (less than 1 ms period). If the field SELINC2=5, the RTTINC2 flag rises once per second.

If RTTINC is defined as the unique source of interrupt (RTTINCEN=1, ALMIEN=0 and INC2AEN=0 in RTT_MR), the value read in RTT_SR by the interrupt handler determines if the current interrupt event corresponds to a 1-second event (RTT_SR[2:1]=3) or to a 1-millisecond event (RTT_SR[2:1]=1). See the figure below.

If the bit INC2AEN=1, RTTINC2 flag is also a source for the RTT alarm signal. See Figure 16-1.

Figure 16-3. RTTINC2 Behavior

RTT slow clock=32.768KHz, RTPRES=32, SELINC2=5



Downloaded from Arrow.com.

16.5 Register Summary

Offset	Name	Bit Pos.								
0x00	RTT_MR	31:24								RTC1HZ
		23:16			INC2AEN	RTTDIS		RTTRST	RTTINCIEN	ALMIEN
		15:8	RTPRES[15:8]							
		7:0	RTPRES[7:0]							
0x04	RTT_AR	31:24	ALMV[31:24]							
		23:16	ALMV[23:16]							
		15:8	ALMV[15:8]							
		7:0	ALMV[7:0]							
0x08	RTT_VR	31:24	CRTV[31:24]							
		23:16	CRTV[23:16]							
		15:8	CRTV[15:8]							
		7:0	CRTV[7:0]							
0x0C	RTT_SR	31:24								
		23:16								
		15:8								
		7:0						RTTINC2	RTTINC	ALMS
0x10	RTT_MODR	31:24								
		23:16								
		15:8								
		7:0						SELINC2[2:0]		
0x14	RTT_TSR	31:24								
		23:16	TSTAMP[23:16]							
		15:8	TSTAMP[15:8]							
		7:0	TSTAMP[7:0]							

Downloaded from Arrow.com.

16.5.1 Real-time Timer Mode Register

Name: RTT_MR
Offset: 0x00
Reset: 0x00008000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
								RTC1HZ
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
			INC2AEN	RTTDIS		RTTRST	RTTINCIEN	ALMIEN
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
	RTPRES[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RTPRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 24 – RTC1HZ Real-Time Clock 1Hz Clock Selection

Value	Description
0	The RTT 32-bit counter is driven by the 16-bit prescaler roll-over events.
1	The RTT 32-bit counter is driven by the 1Hz RTC clock.

Bit 21 – INC2AEN RTTINC2 Alarm and Interrupt Enable

Value	Description
0	The RTTINC2 flag is not a source of the RTT alarm signal nor a source of interrupt.
1	The RTTINC2 flag is a source of the RTT alarm signal and a source of interrupt.

Bit 20 – RTTDIS Real-time Timer Disable

Value	Description
0	The RTT is enabled.
1	The RTT is disabled (no dynamic power consumption).

Bit 18 – RTTRST Real-time Timer Restart

Value	Description
0	No effect.
1	Reloads and restarts the clock divider with the new programmed value. This also resets the 32-bit counter.

Bit 17 – RTTINCIEN Real-time Timer Increment Interrupt Enable

Value	Description
0	The bit RTTINC in RTT_SR has no effect on interrupt.
1	The bit RTTINC in RTT_SR asserts interrupt.

Bit 16 – ALMIEN Alarm Interrupt Enable

Downloaded from Arrow.com.

Value	Description
0	The bit ALMS in RTT_SR has no effect on interrupt.
1	The bit ALMS in RTT_SR asserts interrupt.

Bits 15:0 – RTPRES[15:0] Real-time Timer Prescaler Value
Defines the number of RTT slow clock periods required to increment the Real-time timer. The RTTINCIEN bit must be cleared prior to writing a new RTPRES value.
RTPRES is defined as follows:

- RTPRES = 0: The prescaler period is equal to 2^{16} * slow clock periods.
- RTPRES = 1 or 2: forbidden.
- RTPRES ≠ 0, 1 or 2: The prescaler period is equal to RTPRES * slow clock periods.

Downloaded from Arrow.com.

16.5.2 Real-time Timer Alarm Register

Name: RTT_AR
Offset: 0x04
Reset: 0xFFFFFFFF
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
	ALMV[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	ALMV[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	ALMV[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	ALMV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:0 – ALMV[31:0] Alarm Value
When the CRTV value in RTT_VR equals the ALMV field, the ALMS flag is set in RTT_SR.
The alarm interrupt must be disabled (ALMIEN must be cleared in RTT_MR) when writing a new ALMV value.

Downloaded from Arrow.com.

16.5.3 Real-time Timer Value Register

Name: RTT_VR
Offset: 0x08
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CRTV[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CRTV[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CRTV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CRTV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRTV[31:0] Current Real-time Value
Returns the current value of the RTT.
As CRTV can be updated asynchronously, it must be read twice at the same value.

Downloaded from Arrow.com.

16.5.4 Real-time Timer Status Register

Name: RTT_SR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						RTTINC2	RTTINC	ALMS
Access						R	R	R
Reset						0	0	0

Bit 2 – RTTINC2 Predefined Number of Prescaler Roll-overs Status (cleared on read)

Value	Description
0	SELINC2 = 0 or the number of prescaler roll-overs programmed through the SELINC2 field in RTT_MODR has not been reached since the last read of the RTT_SR.
1	The number of prescaler roll-overs programmed through the SELINC2 field has been reached since the last read of the RTT_SR.

Bit 1 – RTTINC Prescaler Roll-over Status (cleared on read)

Value	Description
0	No prescaler roll-over occurred since the last read of the RTT_SR.
1	Prescaler roll-over occurred since the last read of the RTT_SR.

Bit 0 – ALMS Real-time Alarm Status (cleared on read)

Value	Description
0	The Real-time Alarm has not occurred since the last read of RTT_SR.
1	The Real-time Alarm occurred since the last read of RTT_SR.

Downloaded from Arrow.com.

16.5.5 Real-time Timer Modulo Selection Register

Name: RTT_MODR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						SELINC2[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – SELINC2[2:0] Selection of the 32-bit Counter Modulo to generate RTTINC2 Flag

Value	Name	Description
0	NO_RTTINC2	The RTTINC2 flag never rises.
1	MOD64	The RTTINC2 flag is set when CRTV modulo 64 equals 0.
2	MOD128	The RTTINC2 flag is set when CRTV modulo 128 equals 0.
3	MOD256	The RTTINC2 flag is set when CRTV modulo 256 equals 0.
4	MOD512	The RTTINC2 flag is set when CRTV modulo 512 equals 0.
5	MOD1024	The RTTINC2 flag is set when CRTV modulo 1024 equals 0. Example: If RTPRES=32 then RTTINC2 flag rises once per second if the slow clock is 32.768 kHz.
6	MOD2048	The RTTINC2 flag is set when CRTV modulo 2048 equals 0.
7	MOD4096	The RTTINC2 flag is set when CRTV modulo 4096 equals 0.

Downloaded from Arrow.com.

16.5.6 Real-time Timer Timestamp Register

Name: RTT_TSR
Offset: 0x14
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	TSTAMP[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TSTAMP[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TSTAMP[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – TSTAMP[23:0] Real-time Timer Value Timestamp
Each time an event triggers the flag RTT_SR.RTTINC2, RTT_VR.CRTV is copied into RTT_TSR.TSTAMP. The field TSTAMP remains stable until next event RTT_SR.RTTINC2 and can be used for event timestamping.

Downloaded from Arrow.com.

17. Real-time Clock (RTC)

17.1 Description

The Real-time Clock (RTC) peripheral is designed for very low power consumption. For optimal functionality, the RTC requires an accurate external 32.768 kHz clock, which can be provided by a crystal oscillator.

It combines a complete time-of-day clock with alarm and a Gregorian or Persian calendar, complemented by a programmable periodic interrupt. The alarm and calendar registers are accessed by a 32-bit data bus.

The RTC can also be configured for the UTC time format.

The time and calendar values are coded in binary-coded decimal (BCD) format. The time format can be 24-hour mode or 12-hour mode with an AM/PM indicator.

Updating time and calendar fields and configuring the alarm fields are performed by a parallel capture on the 32-bit data bus. An entry control is performed to avoid loading registers with incompatible BCD format data or with an incompatible date according to the current month/year/century.

A clock divider calibration circuitry can be used to compensate for crystal oscillator frequency variations.

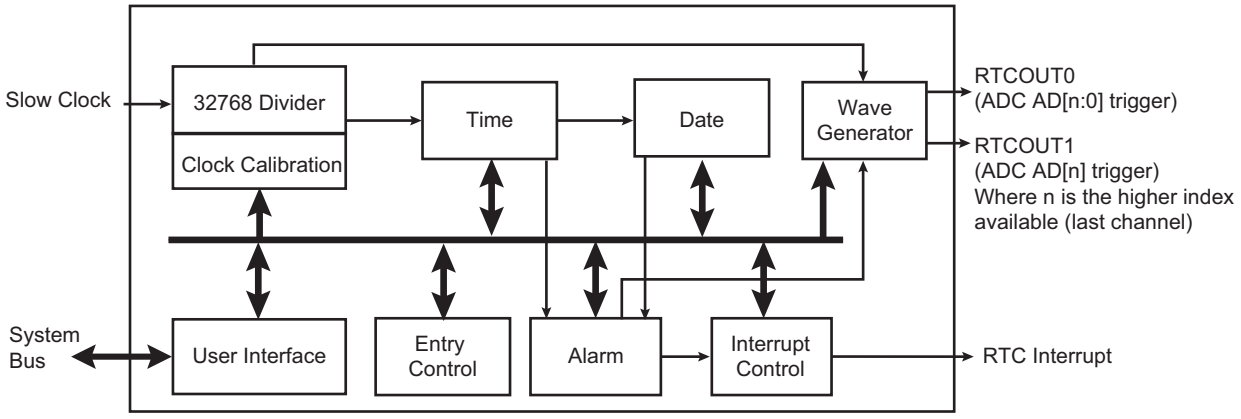
Timestamping capability reports the first and last occurrences of tamper events.

17.2 Embedded Characteristics

- Full Asynchronous Design for Ultra Low-Power Consumption
- Gregorian, UTC and Persian Modes Supported
- Programmable Periodic Interrupt
- Safety/Security Features:
  - Valid time and date programming check
  - On-the-fly time and date validity check
- Counters Calibration Circuitry to Compensate for Crystal Oscillator Variations
- Waveform Generation for Trigger Event
- Tamper Control Registers and Detection Logic
- Tamper Timestamping Registers
- Register Write Protection

17.3 Block Diagram

Figure 17-1. RTC Block Diagram



Downloaded from Arrow.com.

17.4 Product Dependencies

17.4.1 Power Management

The Real-time Clock is continuously clocked at 32.768 kHz. The Power Management Controller (PMC) has no effect on RTC behavior.

17.4.2 Interrupt

Within the System Controller, the RTC interrupt is OR-wired with all the other module interrupts.

Only one System Controller interrupt line is connected on one of the internal sources of the interrupt controller.

RTC interrupt requires the interrupt controller to be programmed first.

When a System Controller interrupt occurs, the service routine must first determine the cause of the interrupt. This is done by reading each status register of the System Controller peripherals successively.

17.5 Functional Description

The RTC provides a full binary-coded decimal (BCD) clock that includes century (19/20), year (with leap years), month, date, day, hours, minutes and seconds reported in RTC Time Register (RTC_TIMR) and RTC Calendar Register (RTC_CALR).

The RTC can operate in UTC mode, giving the number of seconds elapsed since a reference time defined by the user (the UTC standard—ISO 8601—reference time is the 30th of June 1972). In this mode, the timefield is 32 bits wide and coded in hexadecimal format.

The valid year range is up to 2099 in Gregorian mode (or 1300 to 1499 in Persian mode).

The RTC can operate in 24-hour mode or in 12-hour mode with an AM/PM indicator.

Corrections for leap years are included (all years divisible by 4 being leap years except 1900). This is correct up to the year 2099.

The RTC can generate events to trigger ADC measurements.

17.5.1 Reference Clock

The reference clock is the slow clock. It can be driven externally by a 32.768 kHz crystal, or internally.

17.5.2 Timing

In Gregorian and Persian modes, the RTC is updated in real time at one-second intervals in Normal mode for the counters of seconds, at one-minute intervals for the counter of minutes and so on.

In UTC mode, the RTC is updated in real-time at one-second intervals (32-bit UTC counter default configuration).

Due to the asynchronous operation of the RTC with respect to the rest of the chip, to ensure that the value read in the RTC registers (century, year, month, date, day, hours, minutes, seconds) are valid and stable, it is necessary to read these registers twice. If the data is the same both times, then it is valid. Therefore, a minimum of two and a maximum of three accesses are required.

17.5.3 Alarm

In Gregorian and Persian modes, the RTC has five programmable fields: month, date, hours, minutes and seconds.

Each of these fields can be enabled or disabled to match the alarm condition:

- If all the fields are enabled, an alarm flag is generated (the corresponding flag is asserted and an interrupt generated if enabled) at a given month, date, hour/minute/second.
- If only the “seconds” field is enabled, then an alarm is generated every minute.

Depending on the fields that are enabled in the RTC Calendar Alarm register (RTC_CALALR) and the RTC Time Alarm register(RTC_TIMALR), a large number of possibilities are available to the user ranging from minutes to 365/366 days.

Downloaded from Arrow.com.

Note: To change one of the RTC_TIMALR.SEC, MIN, HOUR and/or RTC_CALALR.DATE, MONTH fields, it is recommended to disable the field before changing the value and then re-enable it after the change has been made. This requires up to three accesses to the RTC_TIMALR or RTC_CALALR. The first access clears the enable corresponding to the field to change (RTC_TIMALR.SECEN, MINEN, HOUREN and/or RTC_CALALR.DATEEN, MTHEN). If the field is already cleared, this access is not required. The second access performs the change of the value (RTC_TIMALR.SEC, MIN, HOUR and/or RTC_CALALR.DATE, MONTH). The third access is required to re-enable the field by writing 1 in RTC_TIMALR.SECEN, MINEN, HOUREN and/or RTC_CALALR.DATEEN, MTHEN.

In UTC mode, RTC_TIMALR must be configured to set the UTC alarm value and bit 0 in RTC_CALALR must be used to enable or disable the UTC alarm. If the UTC alarm is enabled, the alarm is generated once the UTC time matches the programmed UTC_TIME alarm field.

To change the UTC_TIME alarm field, proceed as follows:

1. Disable the UTC alarm by clearing RTC_CALALR.UTCEN if it is not already cleared.
2. Change the RTC_TIMALR.UTC_TIME alarm value.
3. Re-enable the UTC alarm by setting RTC_CALALR.UTCEN.

17.5.4 Error Checking when Programming

Verification on user interface data is performed when accessing the century, year, month, date, day, hours, minutes, seconds and alarms. A check is performed on illegal BCD entries such as illegal date of the month with regard to the year and century configured.

If one of the time fields is not correct, the data is not loaded into the register/counter and a flag is set in the validity register. The user can not reset this flag. It is reset as soon as an acceptable value is programmed. This avoids any further side effects in the hardware. The same procedure is followed for the alarm.

The following checks are performed:

1. Century (check if it is in range 19–20 or 13–14 in Persian mode)
2. Year (BCD entry check)
3. Date (check range 01–31)
4. Month (check if it is in BCD range 01–12, check validity regarding “date”)
5. Day (check range 1–7)
6. Hour (BCD checks: in 24-hour mode, check range 00–23 and check that AM/PM flag is not set if RTC is set in 24-hour mode; in 12-hour mode check range 01–12)
7. Minute (check BCD and range 00–59)
8. Second (check BCD and range 00–59)

Note:

1. If the 12-hour mode is selected by means of the Mode register (RTC_MR), a 12-hour value can be programmed and the returned value on RTC_TIMR will be the corresponding 24-hour value. The entry control checks the value of the AM/PM indicator (bit 22 of RTC_TIMR) to determine the range to be checked.
2. In UTC mode, no check is performed on the entries. The RTC does not report any failure.

17.5.5 RTC Internal Free-Running Counter Error Checking

To improve the reliability and security of the RTC, a permanent check is performed on the internal free-running counters to report non-BCD or invalid date/time values.

An error is reported by RTC_SR.TDERR if an incorrect value has been detected. The flag can be cleared by setting the RTC_SCCR.TDERRCLR.

In all cases, RTC_SR.TDERR is set again if the source of the error has not been cleared before clearing RTC_SR.TDERR. The clearing of the source of such error can be done by reprogramming a correct value on RTC_CALR and/or RTC_TIMR.

The RTC internal free-running counters may automatically clear the source of RTC_SR.TDERR due to their roll-over (i.e., every 10 seconds for SECONDS[3:0] in RTC_TIMR). In this case, RTC_SR.TDERR is held high until a clear command is asserted by writing a 1 in RTC_SCCR.TDERRCLR.

Downloaded from Arrow.com.

17.5.6 Updating Time/Calendar

17.5.6.1 Gregorian and Persian Modes

To update time and date, the RTC must be stopped by setting the corresponding field in the Control register (RTC_CR). RTC_CR.UPDTIM must be set to update time fields (hour, minute, second) and RTC_CR.UPDCAL must be set to update calendar fields (century, year, month, date, day).

RTC_SR.ACKUPD must then be read to 1 by either polling RTC_SR or by enabling the acknowledge update interrupt by writing RTC_IER.ACKUPD to ‘1’. Once RTC_SR.ACKUPD is read to 1, it is mandatory to clear this flag by writing a 1 in RTC_SCCR.ACKCLR, after which the user can write to the Time register (RTC_TIMR), the Calendar register (RTC_CALR), or both.

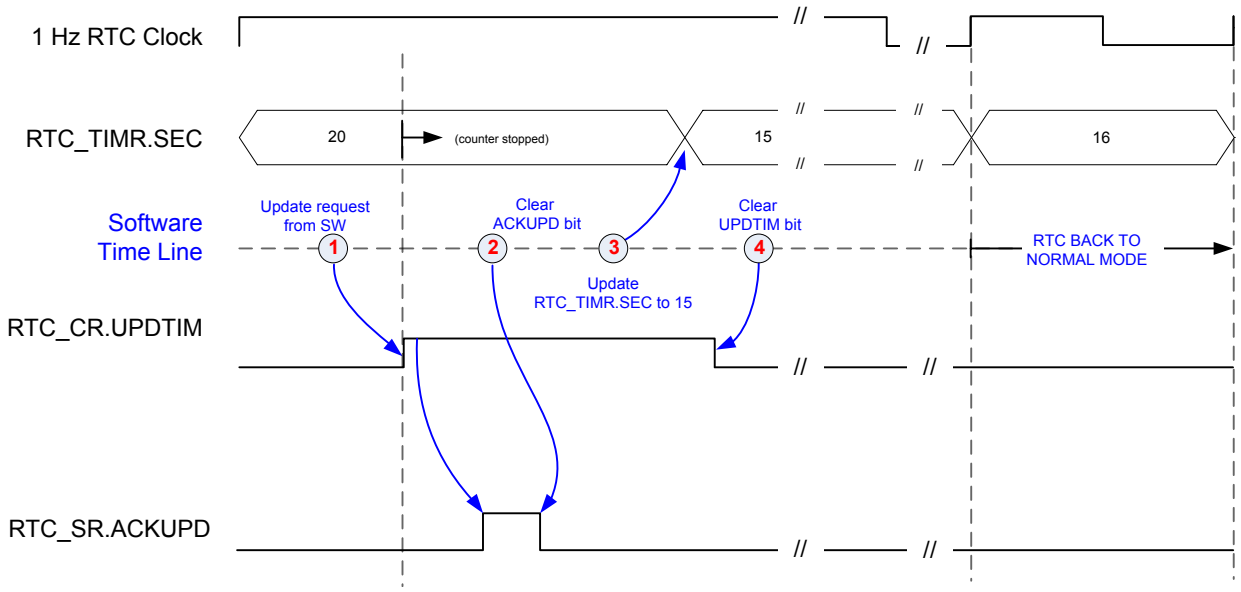
Once the update is finished, the user must write a ‘0’ in RTC_CR.UPDTIM and/or RTC_CR.UPDCAL.

The timing sequence of the time/calendar update is described in the figure below.

When entering the Programming mode of the calendar fields, the time fields remain enabled. When entering the Programming mode of the time fields, both the time and the calendar fields are stopped. This is due to the location of the calendar logical circuity (downstream for low-power considerations).

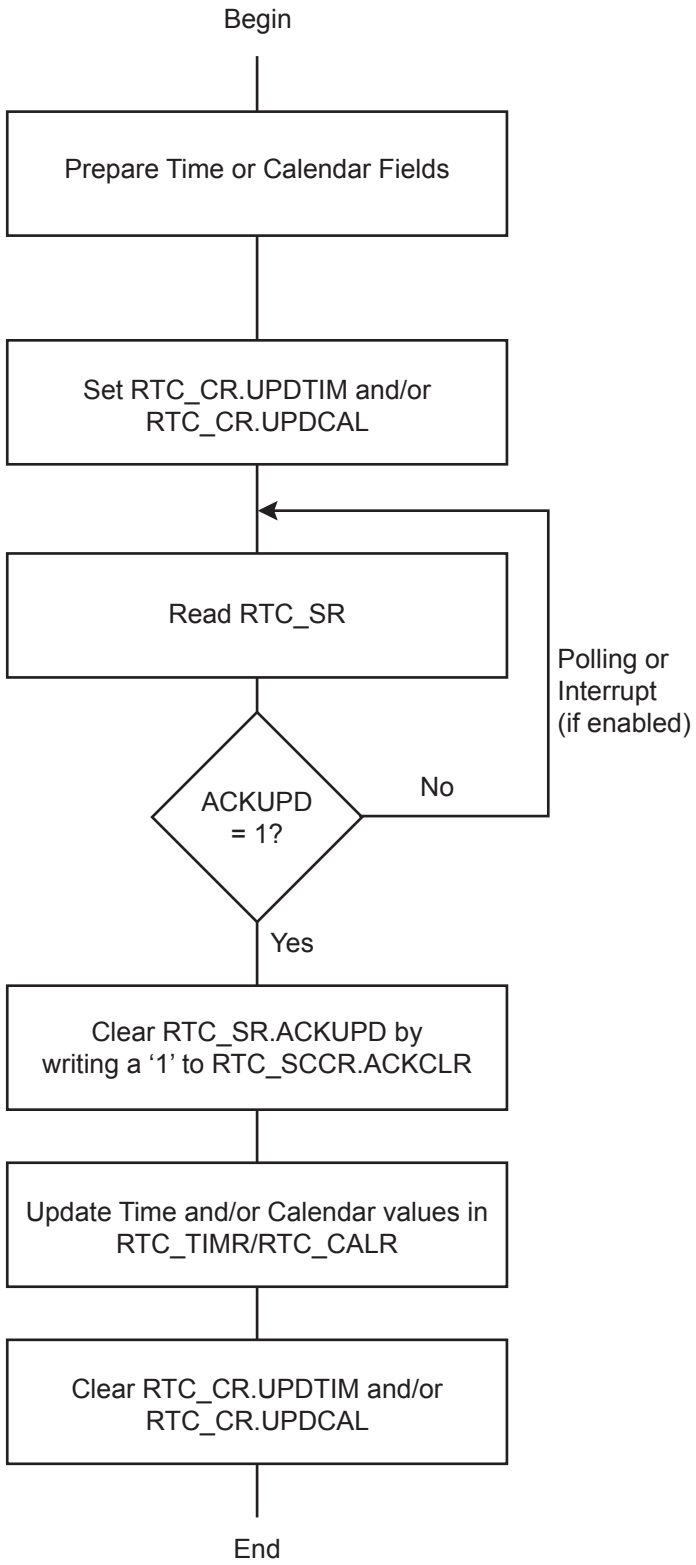
In successive update operations, the user must first check that RTC_CR.UPDTIM and RTC_CR.UPDCAL read 0 before writing these bits to ‘1’.

Figure 17-2. Time/Calendar Update Timing Diagram



Downloaded from Arrow.com.

Figure 17-3. Gregorian and Persian Modes Update Sequence



Downloaded from Arrow.com.

17.5.6.2 UTC Mode

To update the UTC time, the RTC must be stopped by writing a 1 in RTC_CR.UPDTIM and RTC_CR.UPDCAL.

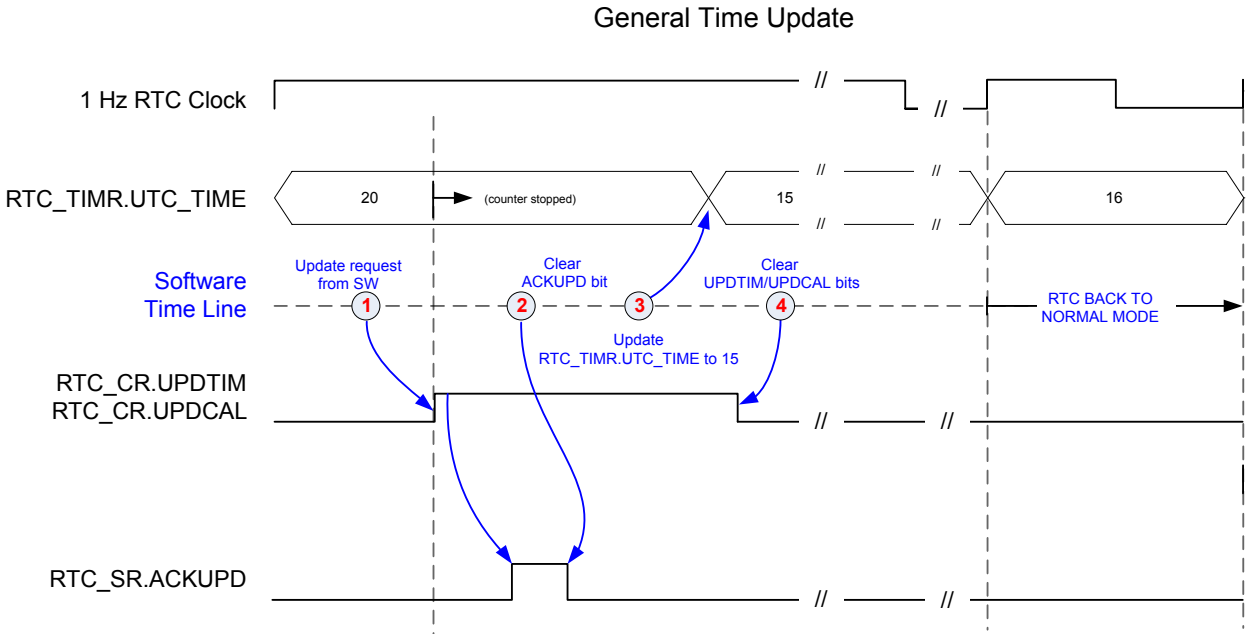
RTC_SR.ACKUPD must then be read to 1 by either polling RTC_SR or by enabling the acknowledge update interrupt by writing a 1 in RTC_IER.ACKUP. Once RTC_SR.ACKUPD is read to 1, it is mandatory to clear this flag by writing a 1 in RTC_SCCR.ACKCLR, after which the user can write to RTC_TIMR.

Once the update is finished, the user must write a 0 in RTC_CR.UPDTIM and a 0 in RTC_CR.UPCAL.

In successive update operations, the user must first check that RTC_CR.UPDTIM and RTC_CR.UPDCAL read 0 before writing a 1 in these bits.

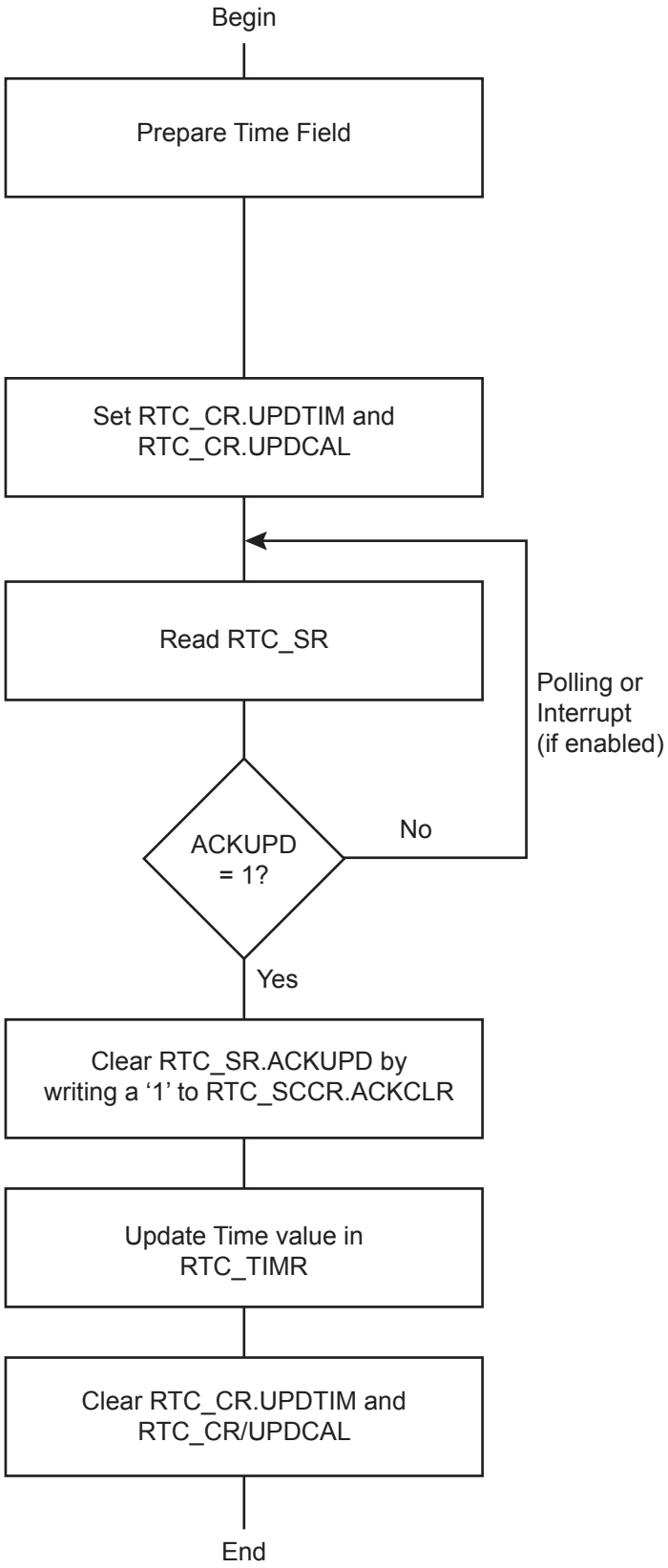
The timing sequence of the UTC time update is described in the figure below.

Figure 17-4. UTC Time Update Timing Diagram



Downloaded from Arrow.com.

Figure 17-5. UTC Mode Update Sequence



Downloaded from Arrow.com.

17.5.7 RTC Accurate Clock Calibration

The crystal oscillator that drives the RTC may not be as accurate as expected mainly due to temperature variation. The RTC is equipped with circuitry able to correct slow clock crystal drift.

To compensate for possible temperature variations over time, this accurate clock calibration circuitry can be programmed on-the-fly and also programmed during application manufacturing, in order to correct the crystal frequency accuracy at room temperature (20–25°C). The typical clock drift range at room temperature is ±20 ppm.

In the device operating temperature range, the 32.768 kHz crystal oscillator clock inaccuracy can be up to -200 ppm.

The RTC clock calibration circuitry allows positive or negative correction in a range of 1.5 ppm to 1950 ppm.

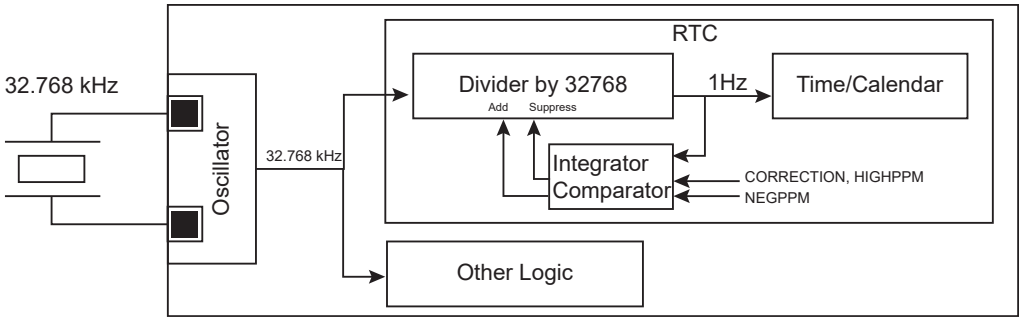
The calibration circuitry is fully digital. Thus, the configured correction is independent of temperature, voltage, process, etc., and no additional measurement is required to check that the correction is effective.

If the correction value configured in the calibration circuitry results from an accurate crystal frequency measure, the remaining accuracy is bounded by the values listed below:

- Below 1 ppm, for an initial crystal drift between 1.5 ppm up to 20 ppm, and from 30 ppm to 90 ppm
- Below 2 ppm, for an initial crystal drift between 20 ppm up to 30 ppm, and from 90 ppm to 130 ppm
- Below 5 ppm, for an initial crystal drift between 130 ppm up to 200 ppm

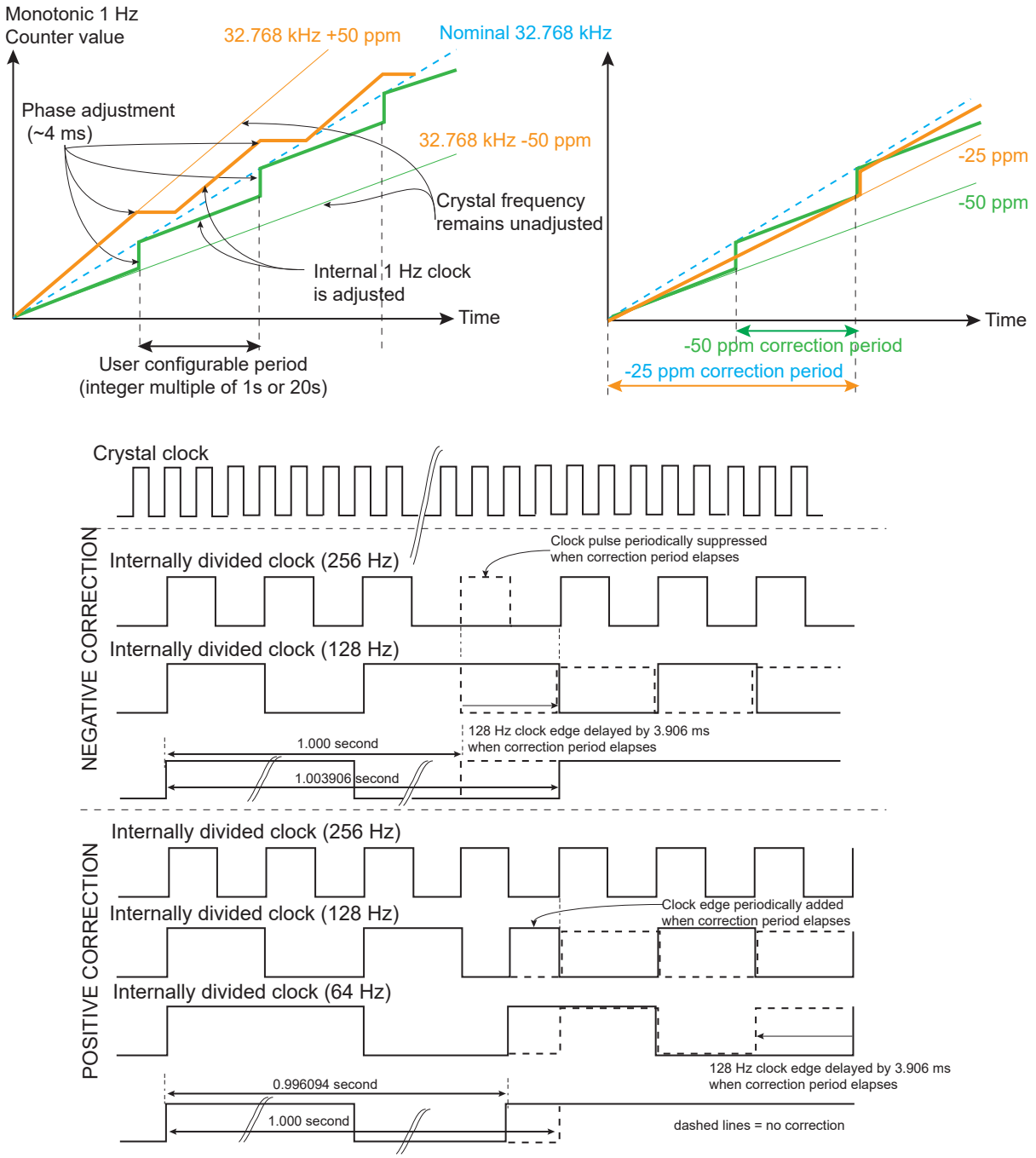
The calibration circuitry does not modify the 32.768 kHz crystal oscillator clock frequency but it acts by slightly modifying the 1 Hz clock period from time to time. The correction event occurs every 1 + [(20 - (19 x HIGHPPM)) x CORRECTION] seconds. When the period is modified, depending on the sign of the correction, the 1 Hz clock period increases or reduces by around 4 ms. Depending on the CORRECTION, NEGPPM and HIGHPPM values configured in RTC_MR, the period interval between two correction events differs.

Figure 17-6. Calibration Circuitry



Downloaded from Arrow.com.

Figure 17-7. Calibration Circuitry Waveforms



The inaccuracy of a crystal oscillator at typical room temperature (±20 ppm at 20–25 °C) can be compensated if a reference clock/signal is used to measure such inaccuracy. This kind of calibration operation can be set up during the final product manufacturing by means of measurement equipment embedding such a reference clock. The correction of value must be programmed into RTC_MR, and this value is kept as long as the circuitry is powered (backup area). Removing the backup power supply cancels this calibration. This room temperature calibration can be further processed by means of the networking capability of the target application.

Note that this adjustment does not take into account the temperature variation.

The frequency drift (up to -200 ppm) due to temperature variation can be compensated using a reference time if the application can access such a reference. If a reference time cannot be used, a temperature sensor can be placed close to the crystal oscillator in order to get the operating temperature of the crystal oscillator. Once obtained, the

Downloaded from Arrow.com.

temperature may be converted using a lookup table (describing the accuracy/temperature curve of the crystal oscillator used) and RTC_MR configured accordingly. The calibration can be performed on-the-fly. This adjustment method is not based on a measurement of the crystal frequency/drift and therefore can be improved by means of the networking capability of the target application.

If no crystal frequency adjustment has been done during manufacturing, it is still possible to make adjustments. In the case where a reference time of the day can be obtained through a LAN/WAN network, it is possible to calculate the drift of the application crystal oscillator by comparing the values read on RTC_TIMR and RTC_CALR and programming RTC_MR.HIGHPPM and RTC_MR.CORRECTION according to the difference measured between the reference time and those of RTC_TIMR and RTC_CALR.

17.5.8 Waveform Generation

Waveforms can be generated in order to take advantage of the RTC inherent prescalers while the RTC is the only powered circuitry (Low-power mode of operation, Backup mode) or in any active mode. Entering Backup or Low-power operating modes does not affect the waveform generation outputs.

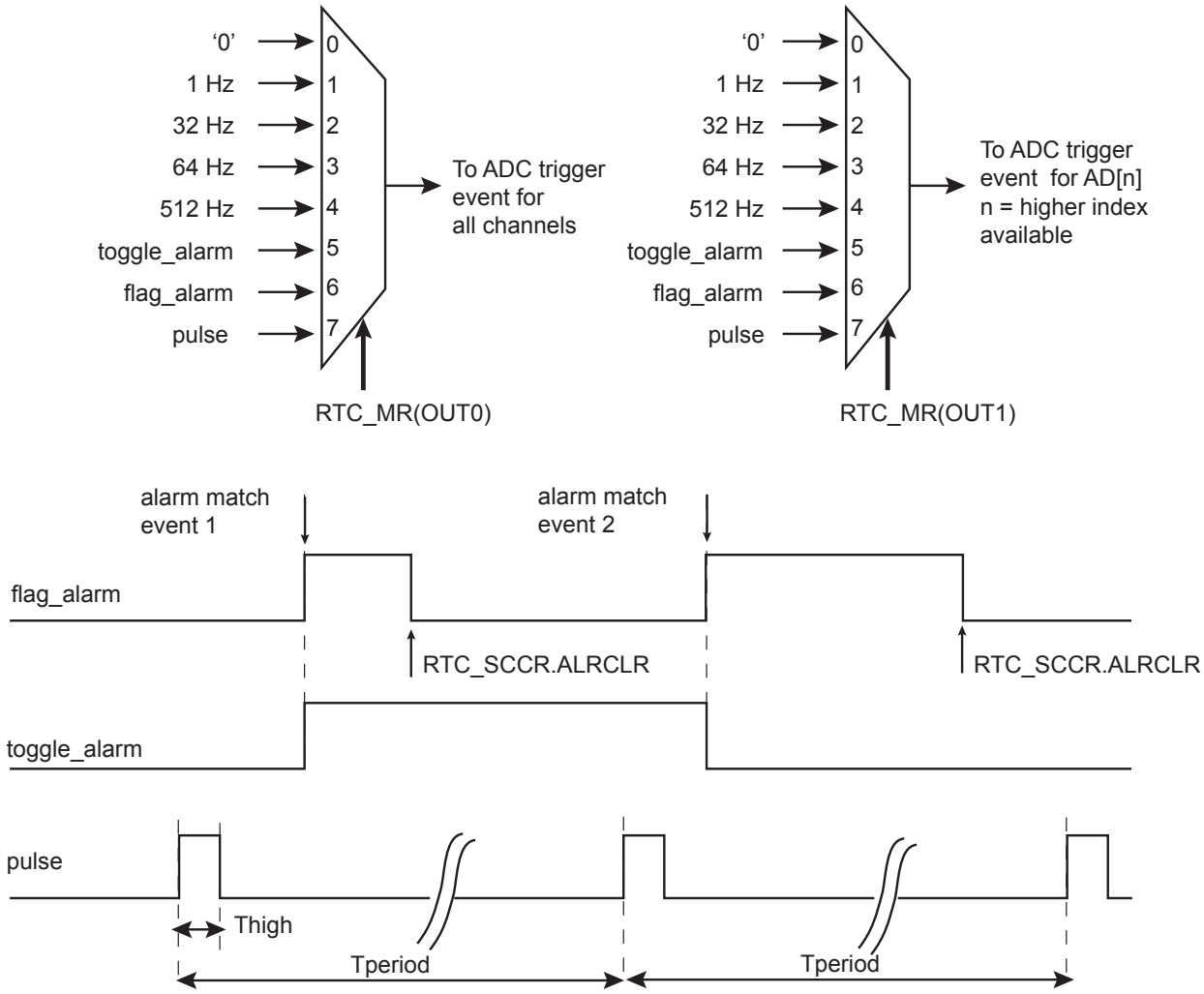
The RTC waveforms are internally routed to ADC trigger events. These events can be configured to provide several types of waveforms. The figure below illustrates the different signals available to generate the waveforms. Two different triggers can be generated at a time. The first is configured in RTC_MR.OUT0 while the second is configurable in RTC_MR.OUT1. OUT0 manages the trigger for channel AD[n:0] (where n is the higher index available (last channel)), while OUT1 manages the channel AD[n] only for specific modes. See the section "Analog to Digital Converter (ADC)" for selection of the measurement triggers and associated modes of operation.

The first selection choice sticks the associated output at 0. (This is the reset value and it can be used at any time to disable the waveform generation).

Selection choices 1 to 4 respectively select 1 Hz, 32 Hz, 64 Hz and 512 Hz.

Downloaded from Arrow.com.

Figure 17-8. Waveform Generation for ADC Trigger Event



17.5.9 Tamper Control Registers and Detection Logic

The WKUP pins used for fast wakeup in PMC are also routed to the tamper detection logic. Any WKUP pin which is not already configured as source of fast wakeup can be configured and selected as a source of a tamper event.

The tamper event can be used to immediately clear (no peripheral clock required) the content of the GPBR, clear the keys stored in AES/TDES, clear the scrambling keys of QSPI/SDR/DDR/SMC memory controllers. Each of these peripherals embeds a configuration bit to allow/disallow the clear on tamper event.

The polarity of each of the WKUP pins can be configured.

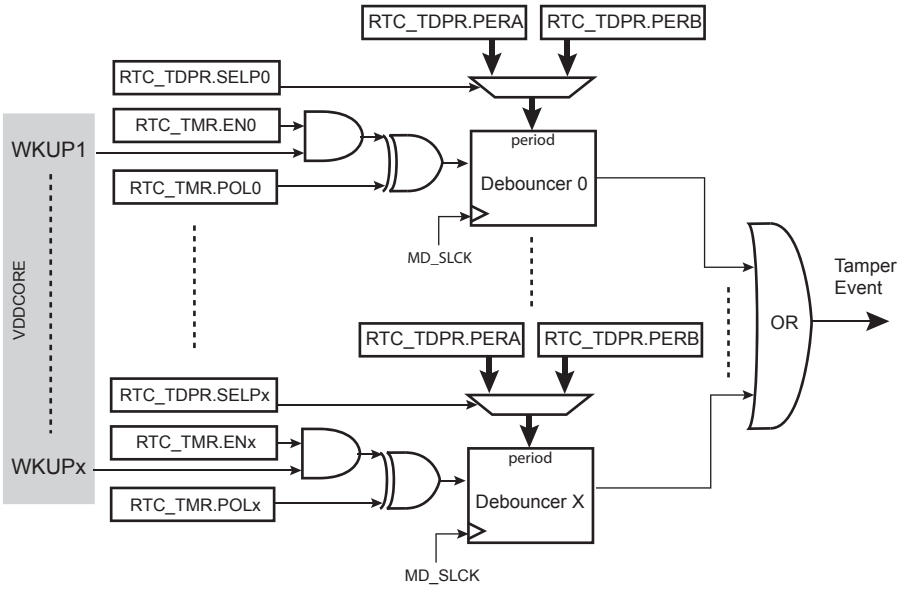
Each of the WKUP pins are debounced prior to create the tamper event.

The tamper event is asserted when one of the lines matches the configured polarity after the debouncing period (see the figure below).

The WKUP pins routed to tamper detection logic are all located on VDDCORE domain, thus there is no tamper detection event when the product is in Backup mode.

Downloaded from Arrow.com.

Figure 17-9. Tamper Detection Circuitry



To enable WKUPx pin to be a source of tamper event, not already configured for a fast wakeup (refer to PMC configuration), the bit RTC_TMR.ENx must be written to 1.

The polarity of the WKUPx pin is configured in RTC_TMR.POLx.

Two debounce periods can be defined by configuring the fields RTC_TDPR.PERA and RTC_TDPR.PERB.

For each WKPUPx pin, the debounce period can be selected from either RTC_TDPR.PERA or RTC_TDPR.PERB by configuring the bit RTC_TDPR.SELPx.

For safety/security reasons, it is possible to lock the tamper configuration registers by writing RTC_TMR.TLOCK=1. Once written to 1, the only way to clear this bit is to perform a VDDCORE reset.

17.5.10 Tamper Timestamping

As soon as a tamper is detected, the tamper counter is incremented and the RTC stores the time of the day, the date and the source of the tamper event in registers located in the backup area. Up to two tamper events can be stored.

In UTC mode, only the UTC time is stored. The date information is not relevant.

The tamper counter saturates at 15. Once this limit is reached, the exact number of tamper occurrences since the last read of stamping registers cannot be known.

The first set of timestamping registers (RTC_TSTR0, RTC_TSDR0, RTC_TSSR0) cannot be overwritten. Once they have been written, all data are stored until the registers are reset. Thus these registers store the first tamper occurrence after a read.

The second set of timestamping registers (RTC_TSTR1, RTC_TSDR1, RTC_TSSR1) are overwritten each time a tamper event is detected. Thus the date and the time data of the first and the second stamping registers may be equal. This occurs when the tamper counter value carried on RTC_TSTR0.TEVCNT equals 1. Thus this second set of registers stores the last occurrence of tamper before a read.

Reading a set of timestamping registers requires three accesses, one for the time of the day, one for the date and one for the tamper source.

Reading the third part (RTC_TSSR0/1) of a timestamping register set clears the whole content of the registers (time, date and tamper source) and makes the timestamping registers available to store a new event.

Downloaded from Arrow.com.

17.6 Register Summary

Offset	Name	Bit Pos.								
0x00	RTC_CR	31:24								
		23:16							CALEVSEL[1:0]	
		15:8							TIMEVSEL[1:0]	
		7:0							UPDCAL	UPDTIM
0x04	RTC_MR	31:24			TPERIOD[1:0]			THIGH[2:0]		
		23:16		OUT1[2:0]				OUT0[2:0]		
		15:8	HIGHPPM	CORRECTION[6:0]						
		7:0				NEGPPM		UTC	PERSIAN	HRMOD
0x08	RTC_TIMR	31:24								
		23:16		AMPM	HOUR[5:0]					
		15:8		MIN[6:0]						
		7:0		SEC[6:0]						
0x08	RTC_TIMR (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							
0x0C	RTC_CALR	31:24			DATE[5:0]					
		23:16	DAY[2:0]			MONTH[4:0]				
		15:8	YEAR[7:0]							
		7:0		CENT[6:0]						
0x10	RTC_TIMALR	31:24								
		23:16	HOUREN	AMPM	HOUR[5:0]					
		15:8	MINEN	MIN[6:0]						
		7:0	SECEN	SEC[6:0]						
0x10	RTC_TIMALR (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							
0x14	RTC_CALALR	31:24	DATEEN		DATE[5:0]					
		23:16	MTHEN			MONTH[4:0]				
		15:8								
		7:0								
0x14	RTC_CALALR (UTC_MODE)	31:24								
		23:16								
		15:8								
		7:0								UTCEN
0x18	RTC_SR	31:24								
		23:16								
		15:8								
		7:0			TDERR	CALEV	TIMEV	SEC	ALARM	ACKUPD
0x1C	RTC_SCCR	31:24								
		23:16								
		15:8								
		7:0			TDERRCLR	CALCLR	TIMCLR	SECCLR	ALRCLR	ACKCLR
0x20	RTC_IER	31:24								
		23:16								
		15:8								
		7:0			TDERREN	CALEN	TIMEN	SECEN	ALREN	ACKEN
0x24	RTC_IDR	31:24								
		23:16								
		15:8								
		7:0			TDERRDIS	CALDIS	TIMDIS	SECDIS	ALRDIS	ACKDIS
0x28	RTC_IMR	31:24								
		23:16								
		15:8								
		7:0			TDERR	CAL	TIM	SEC	ALR	ACK

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x2C	RTC_VER	31:24								
		23:16								
		15:8								
		7:0					NVCALALR	NVTIMALR	NVCAL	NVTIM
0x30 ... 0x57	Reserved									
0x58	RTC_TMR	31:24	TRLOCK							
		23:16	POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
		15:8								
		7:0	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
0x5C	RTC_TDPR	31:24								
		23:16	SELP7	SELP6	SELP5	SELP4	SELP3	SELP2	SELP1	SELP0
		15:8								
		7:0	PERB[3:0]				PERA[3:0]			
0x60 ... 0xAF	Reserved									
0xB0	RTC_TSTR0	31:24	BACKUP				TEVCNT[3:0]			
		23:16		AMPM	HOUR[5:0]					
		15:8		MIN[6:0]						
		7:0		SEC[6:0]						
0xB0	RTC_TSTR0 (UTC_MODE)	31:24	BACKUP				TEVCNT[3:0]			
		23:16								
		15:8								
		7:0								
0xB4	RTC_TSDR0	31:24			DATE[5:0]					
		23:16	DAY[2:0]			MONTH[4:0]				
		15:8	YEAR[7:0]							
		7:0		CENT[6:0]						
0xB4	RTC_TSDRx (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							
0xB8	RTC_TSSR0	31:24								
		23:16	DET7	DET6	DET5	DET4	DET3	DET2	DET1	DET0
		15:8								
		7:0								
0xBC	RTC_TSTR1	31:24	BACKUP							
		23:16		AMPM	HOUR[5:0]					
		15:8		MIN[6:0]						
		7:0		SEC[6:0]						
0xBC	RTC_TSTR1 (UTC_MODE)	31:24	BACKUP							
		23:16								
		15:8								
		7:0								
0xC0	RTC_TSDR1	31:24			DATE[5:0]					
		23:16	DAY[2:0]			MONTH[4:0]				
		15:8	YEAR[7:0]							
		7:0		CENT[6:0]						
0xC4	RTC_TSSR1	31:24								
		23:16	DET7	DET6	DET5	DET4	DET3	DET2	DET1	DET0
		15:8								
		7:0								

Downloaded from Arrow.com.

17.6.1 RTC Control Register

Name: RTC_CR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							CALEVSEL[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
							TIMEVSEL[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
							UPDCAL	UPDTIM
Access							R/W	R/W
Reset							0	0

Bits 17:16 – CALEVSEL[1:0] Calendar Event Selection
The event that generates the flag CALEV in RTC_SR depends on the value of CALEVSEL
In UTC mode, this field has no effect on RTC_SR.

Value	Name	Description
0	WEEK	Week change (every Monday at time 00:00:00)
1	MONTH	Month change (every 01 of each month at time 00:00:00)
2	YEAR	Year change (every January 1 at time 00:00:00)
3	YEAR	Reserved

Bits 9:8 – TIMEVSEL[1:0] Time Event Selection
The event that generates the flag TIMEV in RTC_SR depends on the value of TIMEVSEL.
In UTC mode, this field has no effect on RTC_SR.

Value	Name	Description
0	MINUTE	Minute change
1	HOUR	Hour change
2	MIDNIGHT	Every day at midnight
3	NOON	Every day at noon

Bit 1 – UPDCAL Update Request Calendar Register
Calendar counting consists of day, date, month, year and century counters. Calendar counters can be programmed once this bit is set and acknowledged by the RTC_SR.ACKUPD bit.
In UTC mode, both UPDTIM and UPDCAL must be set to '1' in order to update the UTC time value.

Value	Description
0	No effect or, if UPDCAL has been previously written to 1, stops the update procedure.
1	Stops the RTC calendar counting.

Downloaded from Arrow.com.

Bit 0 – UPDTIM Update Request Time Register
Time counting consists of second, minute and hour counters. Time counters can be programmed once this bit is set and acknowledged by the RTC_SR.ACKUPD bit.
In UTC mode, both UPDTIM and UPDCAL must be set to '1' in order to update the UTC time value.

Value	Description
0	No effect or, if UPDTIM has been previously written to 1, stops the update procedure.
1	Stops the RTC time counting.

Downloaded from Arrow.com.

17.6.2 RTC Mode Register

Name: RTC_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
			TPERIOD[1:0]			THIGH[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
		OUT1[2:0]				OUT0[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
	HIGHPPM	CORRECTION[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				NEGPPM		UTC	PERSIAN	HRMOD
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bits 29:28 – TPERIOD[1:0] Period of the Output Pulse

Value	Name	Description
0	P_1S	1 second
1	P_500MS	500 ms
2	P_250MS	250 ms
3	P_125MS	125 ms

Bits 26:24 – THIGH[2:0] High Duration of the Output Pulse

Value	Name	Description
0	H_31MS	31.2 ms
1	H_16MS	15.6 ms
2	H_4MS	3.91 ms
3	H_976US	976 µs
4	H_488US	488 µs
5	H_122US	122 µs
6	H_30US	30.5 µs
7	H_15US	15.2 µs

Bits 22:20 – OUT1[2:0] ADC Last Channel Trigger Event Source Selection

Value	Name	Description
0	NO_WAVE	No waveform, stuck at ‘0’
1	FREQ1HZ	1 Hz square wave
2	FREQ32HZ	32 Hz square wave
3	FREQ64HZ	64 Hz square wave
4	FREQ512HZ	512 Hz square wave
5	ALARM_TOGGLE	Output toggles when alarm flag rises

Downloaded from Arrow.com.

Value	Name	Description
6	ALARM_FLAG	Output is a copy of the alarm flag
7	PROG_PULSE	Duty cycle programmable pulse

Bits 18:16 – OUT0[2:0] All ADC Channel Trigger Event Source Selection

Value	Name	Description
0	NO_WAVE	No waveform, stuck at '0'
1	FREQ1HZ	1 Hz square wave
2	FREQ32HZ	32 Hz square wave
3	FREQ64HZ	64 Hz square wave
4	FREQ512HZ	512 Hz square wave
5	ALARM_TOGGLE	Output toggles when alarm flag rises
6	ALARM_FLAG	Output is a copy of the alarm flag
7	PROG_PULSE	Duty cycle programmable pulse

Bit 15 – HIGHPPM HIGH PPM Correction
If the absolute value of the correction to be applied is lower than 30 ppm, it is recommended to clear HIGHPPM. HIGHPPM set to 1 is recommended for 30 ppm correction and above.
Formula:
If HIGHPPM = 0, then the clock frequency correction range is from 1.5 ppm up to 98 ppm. The RTC accuracy is less than 1 ppm for a range correction from 1.5 ppm up to 30 ppm.
The correction field must be programmed according to the required correction in ppm; the formula is as follows:

$$\text{CORRECTION} = \frac{3906}{20 \times \text{ppm}} - 1$$

The value obtained must be rounded to the nearest integer prior to being programmed into CORRECTION field.
If HIGHPPM = 1, then the clock frequency correction range is from 30.5 ppm up to 1950 ppm. The RTC accuracy is less than 1 ppm for a range correction from 30.5 ppm up to 90 ppm.
The correction field must be programmed according to the required correction in ppm; the formula is as follows:

$$\text{CORRECTION} = \frac{3906}{\text{ppm}} - 1$$

The value obtained must be rounded to the nearest integer prior to be programmed into CORRECTION field.
If NEGPPM is set to 1, the ppm correction is negative (used to correct crystals that are faster than the nominal 32.768 kHz).

Value	Description
0	Lower range ppm correction with accurate correction.
1	Higher range ppm correction with accurate correction.

Bits 14:8 – CORRECTION[6:0] Slow Clock Correction

Value	Description
0	No correction
1-127	The slow clock will be corrected according to the formula given in HIGHPPM description.

Bit 4 – NEGPPM Negative PPM Correction
See CORRECTION and HIGHPPM field descriptions.
NEGPPM must be cleared to correct a crystal slower than 32.768 kHz.

Value	Description
0	Positive correction (the divider will be slightly higher than 32768).
1	Negative correction (the divider will be slightly lower than 32768).

Bit 2 – UTC UTC Time Format
It is forbidden to write a one to the UTC and PERSIAN bits at the same time.

Value	Description
0	Gregorian or Persian calendar.
1	UTC format.

Bit 1 – PERSIAN PERSIAN Calendar

Downloaded from Arrow.com.

Value	Description
0	Gregorian calendar.
1	Persian calendar.

Bit 0 – HRMOD 12-/24-hour Mode

Value	Description
0	24-hour mode is selected.
1	12-hour mode is selected.

Downloaded from Arrow.com.

17.6.3 RTC Time Register

Name: RTC_TIMR
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

In UTC mode, this register view is not relevant, see 17.6.7 RTC_TIMALR (UTC_MODE) .

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		AMPM	HOUR[5:0]					
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		MIN[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SEC[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 22 – AMPM Ante Meridiem Post Meridiem Indicator
This bit is the AM/PM indicator in 12-hour mode.

Value	Description
0	AM.
1	PM.

Bits 21:16 – HOUR[5:0] Current Hour
The range that can be set is 1–12 (BCD) in 12-hour mode or 0–23 (BCD) in 24-hour mode.

Bits 14:8 – MIN[6:0] Current Minute
The range that can be set is 0–59 (BCD).
The lowest four bits encode the units. The higher bits encode the tens.

Bits 6:0 – SEC[6:0] Current Second
The range that can be set is 0–59 (BCD).
The lowest four bits encode the units. The higher bits encode the tens.

Downloaded from Arrow.com.

17.6.4 RTC Time Register (UTC_MODE)

Name: RTC_TIMR (UTC_MODE)
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

This configuration is relevant only if UTC = 1 in RTC_MR.

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
	UTC_TIME[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	UTC_TIME[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UTC_TIME[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UTC_TIME[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – UTC_TIME[31:0] Current UTC Time
Any value can be set.

Downloaded from Arrow.com.

17.6.5 RTC Calendar Register

Name: RTC_CALR
Offset: 0x0C
Reset: 0x01411720
Property: Read/Write

In UTC mode, values read in this register are not relevant.

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
			DATE[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	1

Bit	23	22	21	20	19	18	17	16
	DAY[2:0]			MONTH[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
	YEAR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	0	1	1	1

Bit	7	6	5	4	3	2	1	0
		CENT[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	1	0	0	0	0	0

Bits 29:24 – DATE[5:0] Current Day in Current Month
The range that can be set is 01–31 (BCD).
The lowest four bits encode the units. The higher bits encode the tens.

Bits 23:21 – DAY[2:0] Current Day in Current Week
The range that can be set is 1–7 (BCD).
The coding of the number (which number represents which day) is user-defined as it has no effect on the date counter.

Bits 20:16 – MONTH[4:0] Current Month
The range that can be set is 01–12 (BCD).
The lowest four bits encode the units. The higher bits encode the tens.

Bits 15:8 – YEAR[7:0] Current Year
The range that can be set is 00–99 (BCD).
The lowest four bits encode the units. The higher bits encode the tens.

Bits 6:0 – CENT[6:0] Current Century
The range that can be set is 19–20 (Gregorian) or 13–14 (Persian) (BCD).
The lowest four bits encode the units. The higher bits encode the tens.

Downloaded from Arrow.com.

17.6.6 RTC Time Alarm Register

Name: RTC_TIMALR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

To change one of the SEC, MIN, HOUR fields, it is recommended to disable the field before changing the value and then re-enable it after the change has been made. This requires up to three accesses to RTC_TIMALR. The first access clears the enable corresponding to the field to change (SECEN, MINEN, HOUREN). If the field is already cleared, this access is not required. The second access performs the change of value (SEC, MIN, HOUR). The third access is required to re-enable the field by writing 1 in the SECEN, MINEN, HOUREN fields.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	HOUREN	AMPM	HOUR[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MINEN	MIN[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SECEN	SEC[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 23 – HOUREN Hour Alarm Enable

Value	Description
0	The hour-matching alarm is disabled.
1	The hour-matching alarm is enabled.

Bit 22 – AMPM AM/PM Indicator
This field is the alarm field corresponding to the BCD-coded hour counter.

Bits 21:16 – HOUR[5:0] Hour Alarm
This field is the alarm field corresponding to the BCD-coded hour counter.

Bit 15 – MINEN Minute Alarm Enable

Value	Description
0	The minute-matching alarm is disabled.
1	The minute-matching alarm is enabled.

Bits 14:8 – MIN[6:0] Minute Alarm
This field is the alarm field corresponding to the BCD-coded minute counter.

Bit 7 – SECEN Second Alarm Enable

Downloaded from Arrow.com.

Value	Description
0	The second-matching alarm is disabled.
1	The second-matching alarm is enabled.

Bits 6:0 – SEC[6:0] Second Alarm
This field is the alarm field corresponding to the BCD-coded second counter.

Downloaded from Arrow.com.

17.6.7 RTC Time Alarm Register (UTC_MODE)

Name: RTC_TIMALR (UTC_MODE)
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

This configuration is relevant only if UTC = 1 in RTC_MR.

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
	UTC_TIME[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	UTC_TIME[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UTC_TIME[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UTC_TIME[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – UTC_TIME[31:0] UTC_TIME Alarm
This field is the alarm field corresponding to the UTC time counter. To change it, proceed as follows:
1. Disable the UTC alarm by clearing RTC_CALALR.UTCEN if it is not already cleared.
2. Change the UTC_TIME alarm value.
3. Enable the UTC alarm by setting RTC_CALALR.UTCEN.

Downloaded from Arrow.com.

17.6.8 RTC Calendar Alarm Register

Name: RTC_CALALR
Offset: 0x14
Reset: 0x01010000
Property: Read/Write

In UTC mode, this register view is not relevant, see 17.6.9 RTC_CALALR (UTC_MODE).

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

To change one of the DATE, MONTH fields, it is recommended to disable the field before changing the value and then re-enable it after the change has been made. This requires up to three accesses to RTC_CALALR. The first access clears the enable corresponding to the field to change (DATEEN, MTHEN). If the field is already cleared, this access is not required. The second access performs the change of the value (DATE, MONTH). The third access is required to re-enable the field by writing 1 in DATEEN, MTHEN fields.

Bit	31	30	29	28	27	26	25	24
	DATEEN		DATE[5:0]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	1

Bit	23	22	21	20	19	18	17	16
	MTHEN			MONTH[4:0]				
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 31 – DATEEN Date Alarm Enable

Value	Description
0	The date-matching alarm is disabled.
1	The date-matching alarm is enabled.

Bits 29:24 – DATE[5:0] Date Alarm
This field is the alarm field corresponding to the BCD-coded date counter.

Bit 23 – MTHEN Month Alarm Enable

Value	Description
0	The month-matching alarm is disabled.
1	The month-matching alarm is enabled.

Bits 20:16 – MONTH[4:0] Month Alarm
This field is the alarm field corresponding to the BCD-coded month counter.

Downloaded from Arrow.com.

17.6.9 RTC Calendar Alarm Register (UTC_MODE)

Name: RTC_CALALR (UTC_MODE)
Offset: 0x14
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								UTCEN
Access								R/W
Reset								0

Bit 0 – UTCEN UTC Alarm Enable

Value	Description
0	The UTC-matching alarm is disabled.
1	The UTC-matching alarm is enabled.

Downloaded from Arrow.com.

17.6.10 RTC Status Register

Name: RTC_SR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TDERR	CALEV	TIMEV	SEC	ALARM	ACKUPD
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 5 – TDERR Time and/or Date Free Running Error
If the RTC is configured in UTC mode, the value returned by this field is not relevant.

Value	Name	Description
0	CORRECT	The internal free running counters are carrying valid values since the last read of the Status register (RTC_SR).
1	ERR_TIMEDATE	The internal free running counters have been corrupted (invalid date or time, non-BCD values) since the last read and/or they are still invalid.

Bit 4 – CALEV Calendar Event
The calendar event is selected in RTC_CR.TIMEVSEL and can be any one of the following events: week change, month change and year change. If the RTC is configured in UTC mode, the value returned by this field is not relevant.

Value	Name	Description
0	NO_CALEVENT	No calendar event has occurred since the last clear.
1	CALEVENT	At least one calendar event has occurred since the last clear.

Bit 3 – TIMEV Time Event
The time event is selected in RTC_CR.TIMEVSEL and can be any one of the following events: minute change, hour change, noon, midnight (day change). If the RTC is configured in UTC mode, the value returned by this field is not relevant.

Value	Name	Description
0	NO_TIMEVENT	No time event has occurred since the last clear.
1	TIMEVENT	At least one time event has occurred since the last clear.

Bit 2 – SEC Second Event

Value	Name	Description
0	NO_SECEVENT	No second event has occurred since the last clear.
1	SECEVENT	At least one second event has occurred since the last clear.

Downloaded from Arrow.com.

Bit 1 – ALARM Alarm Flag

Value	Name	Description
0	NO_ALARMEVENT	No alarm matching condition occurred.
1	ALARMEVENT	An alarm matching condition has occurred.

Bit 0 – ACKUPD Acknowledge for Update

Value	Name	Description
0	FREERUN	Time and calendar registers cannot be updated.
1	UPDATE	Time and calendar registers can be updated.

Downloaded from Arrow.com.

17.6.11 RTC Status Clear Command Register

Name: RTC_SCCR
Offset: 0x1C
Reset: –
Property: Write-only

To avoid missing clearing commands, wait for three slow clock cycles between two accesses to this register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Clears the corresponding status flag in the Status register (RTC_SR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TDERRCLR	CALCLR	TIMCLR	SECCLR	ALRCLR	ACKCLR
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 5 – TDERRCLR Time and/or Date Free Running Error Clear
If the RTC is configured in UTC mode, this bit has no effect.

Bit 4 – CALCLR Calendar Clear
If the RTC is configured in UTC mode, this bit has no effect.

Bit 3 – TIMCLR Time Clear
If the RTC is configured in UTC mode, this bit has no effect.

Bit 2 – SECCLR Second Clear

Bit 1 – ALRCLR Alarm Clear

Bit 0 – ACKCLR Acknowledge Clear

Downloaded from Arrow.com.

17.6.12 RTC Interrupt Enable Register

Name: RTC_IER
Offset: 0x20
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TDERREN	CALEN	TIMEN	SECEN	ALREN	ACKEN
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 5 – TDERREN Time and/or Date Error Interrupt Enable
If the RTC is configured in UTC mode, this bit has no effect.

Bit 4 – CALEN Calendar Event Interrupt Enable
If the RTC is configured in UTC mode, this bit has no effect.

Bit 3 – TIMEN Time Event Interrupt Enable
If the RTC is configured in UTC mode, this bit has no effect.

Bit 2 – SECEN Second Event Interrupt Enable

Bit 1 – ALREN Alarm Interrupt Enable

Bit 0 – ACKEN Acknowledge Update Interrupt Enable

Downloaded from Arrow.com.

17.6.13 RTC Interrupt Disable Register

Name: RTC_IDR
Offset: 0x24
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TDERRDIS	CALDIS	TIMDIS	SECDIS	ALRDIS	ACKDIS
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 5 – TDERRDIS Time and/or Date Error Interrupt Disable
If the RTC is configured in UTC mode, this bit has no effect.

Bit 4 – CALDIS Calendar Event Interrupt Disable
If the RTC is configured in UTC mode, this bit has no effect.

Bit 3 – TIMDIS Time Event Interrupt Disable
If the RTC is configured in UTC mode, this bit has no effect.

Bit 2 – SECDIS Second Event Interrupt Disable

Bit 1 – ALRDIS Alarm Interrupt Disable

Bit 0 – ACKDIS Acknowledge Update Interrupt Disable

Downloaded from Arrow.com.

17.6.14 RTC Interrupt Mask Register

Name: RTC_IMR
Offset: 0x28
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TDERR	CAL	TIM	SEC	ALR	ACK
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 5 – TDERR Time and/or Date Error Mask
If the RTC is configured in UTC mode, this bit has no effect.

Bit 4 – CAL Calendar Event Interrupt Mask
If the RTC is configured in UTC mode, this bit is not relevant.

Bit 3 – TIM Time Event Interrupt Mask
If the RTC is configured in UTC mode, this bit is not relevant.

Bit 2 – SEC Second Event Interrupt Mask

Bit 1 – ALR Alarm Interrupt Mask

Bit 0 – ACK Acknowledge Update Interrupt Mask

Downloaded from Arrow.com.

17.6.15 RTC Valid Entry Register

Name: RTC_VER
Offset: 0x2C
Reset: 0x00000000
Property: Read-only

If the RTC is configured in UTC mode, the values returned by this register are not relevant.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					NVCALALR	NVTIMALR	NVCAL	NVTIM
Access					R	R	R	R
Reset					0	0	0	0

Bit 3 – NVCALALR Non-valid Calendar Alarm

Value	Description
0	No invalid data has been detected in RTC_CALALR (Calendar Alarm register).
1	RTC_CALALR has contained invalid data since it was last programmed.

Bit 2 – NVTIMALR Non-valid Time Alarm

Value	Description
0	No invalid data has been detected in RTC_TIMALR (Time Alarm register).
1	RTC_TIMALR has contained invalid data since it was last programmed.

Bit 1 – NVCAL Non-valid Calendar

Value	Description
0	No invalid data has been detected in RTC_CALR (Calendar register).
1	RTC_CALR has contained invalid data since it was last programmed.

Bit 0 – NVTIM Non-valid Time

Value	Description
0	No invalid data has been detected in RTC_TIMR (Time register).
1	RTC_TIMR has contained invalid data since it was last programmed.

Downloaded from Arrow.com.

17.6.16 RTC TimeStamp Time Register 0

Name: RTC_TSTR0
Offset: 0xB0
Reset: 0x00000000
Property: Read-only

These fields are valid for non-UTC mode only.

RTC_TSTR0 reports the timestamp of the first tamper event after reading RTC_TSSR0.

Bit	31	30	29	28	27	26	25	24
	BACKUP				TEVCNT[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
		AMPM	HOUR[5:0]					
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		MIN[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SEC[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 31 – BACKUP System Mode of the Tamper (cleared by reading RTC_TSSR0)

Value	Description
0	The state of the system is different from Backup mode when the tamper event occurs.
1	The system is in Backup mode when the tamper event occurs.

Bits 27:24 – TEVCNT[3:0] Tamper Events Counter (cleared by reading RTC_TSSR0)
Each time a tamper event occurs, this counter is incremented. This counter saturates at 15. Once this value is reached, it is no more possible to know the exact number of tamper events.
If this field is not null, this implies that at least one tamper event occurs since last register reset and that the values stored in timestamping registers are valid.

Bit 22 – AMPM AM/PM Indicator of the Tamper (cleared by reading RTC_TSSR0)

Bits 21:16 – HOUR[5:0] Hours of the Tamper (cleared by reading RTC_TSSR0)

Bits 14:8 – MIN[6:0] Minutes of the Tamper (cleared by reading RTC_TSSR0)

Bits 6:0 – SEC[6:0] Seconds of the Tamper (cleared by reading RTC_TSSR0)

Downloaded from Arrow.com.

17.6.17 RTC TimeStamp Time Register 0 (UTC_MODE)

Name: RTC_TSTR0 (UTC_MODE)
Offset: 0xB0
Reset: 0x00000000
Property: Read-only

RTC_TSTR0 reports the timestamp of the first tamper event after reading RTC_TSSR0.

Bit	31	30	29	28	27	26	25	24
	BACKUP				TEVCNT[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 31 – BACKUP System Mode of the Tamper (cleared by reading RTC_TSSR0)

Value	Description
0	The state of the system is different from Backup mode when the tamper event occurs.
1	The system is in Backup mode when the tamper event occurs.

Bits 27:24 – TEVCNT[3:0] Tamper Events Counter (cleared by reading RTC_TSSR0)
Each time a tamper event occurs, this counter is incremented. This counter saturates at 15. Once this value is reached, it is no more possible to know the exact number of tamper events.
If this field is not null, this implies that at least one tamper event occurs since last register reset and that the values stored in timestamping registers are valid.

Downloaded from Arrow.com.

17.6.18 RTC TimeStamp Time Register 1

Name: RTC_TSTR1
Offset: 0xBC
Reset: 0x00000000
Property: Read-only

These fields are valid for non-UTC mode only.

RTC_TSTR1 reports the timestamp of the last tamper event after reading RTC_TSSR1.

Bit	31	30	29	28	27	26	25	24
	BACKUP							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
		AMPM	HOUR[5:0]					
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		MIN[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SEC[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 31 – BACKUP System Mode of the Tamper (cleared by reading RTC_TSSR1)

Value	Description
0	The state of the system is different from Backup mode when the tamper event occurs.
1	The system is in Backup mode when the tamper event occurs.

Bit 22 – AMPM AM/PM Indicator of the Tamper (cleared by reading RTC_TSSR1)

Bits 21:16 – HOUR[5:0] Hours of the Tamper (cleared by reading RTC_TSSR1)

Bits 14:8 – MIN[6:0] Minutes of the Tamper (cleared by reading RTC_TSSR1)

Bits 6:0 – SEC[6:0] Seconds of the Tamper (cleared by reading RTC_TSSR1)

Downloaded from Arrow.com.

17.6.19 RTC TimeStamp Time Register 1 (UTC_MODE)

Name: RTC_TSTR1 (UTC_MODE)
Offset: 0xBC
Reset: 0x00000000
Property: Read-only

RTC_TSTR1 reports the timestamp of the last tamper event after reading RTC_TSSR1.

Bit	31	30	29	28	27	26	25	24
	BACKUP							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 31 – BACKUP System Mode of the Tamper (cleared by reading RTC_TSSR1)

Value	Description
0	The state of the system is different from Backup mode when the tamper event occurs.
1	The system is in Backup mode when the tamper event occurs.

Downloaded from Arrow.com.

17.6.20 RTC TimeStamp Date Register

Name: RTC_TSDRx
Offset: 0xB4 + x*0x0C [x=0..1]
Reset: 0x00000000
Property: Read-only

These fields contain the date and the source of a tamper occurrence if RTC_TSTR0.TEVCNT field is not null.

These fields are relevant for non-UTC mode only.

RTC_TSDR0 reports the timestamp of the first tamper event after reading RTC_TSSR0, and RTC_TSDR1 reports the timestamp of the last tamper event.

Bit	31	30	29	28	27	26	25	24
			DATE[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DAY[2:0]			MONTH[4:0]				
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	YEAR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		CENT[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 29:24 – DATE[5:0] Date of the Tamper (cleared by reading RTC_TSSRx)

Bits 23:21 – DAY[2:0] Day of the Tamper (cleared by reading RTC_TSSRx)

Bits 20:16 – MONTH[4:0] Month of the Tamper (cleared by reading RTC_TSSRx)

Bits 15:8 – YEAR[7:0] Year of the Tamper (cleared by reading RTC_TSSRx)

Bits 6:0 – CENT[6:0] Century of the Tamper (cleared by reading RTC_TSSRx)

Downloaded from Arrow.com.

17.6.21 RTC TimeStamp Date Register (UTC_MODE)

Name: RTC_TSDRx (UTC_MODE)
Offset: 0xB4
Reset: 0x00000000
Property: Read-only

RTC_TSDR0 reports the timestamp of the first tamper event after reading RTC_TSSR0, and RTC_TSDR1 reports the timestamp of the last tamper event.
This register is cleared by reading RTC_TSSRx.

Bit	31	30	29	28	27	26	25	24
	UTC_TIME[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	UTC_TIME[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UTC_TIME[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UTC_TIME[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – UTC_TIME[31:0] Time of the Tamper (UTC format)
This configuration is relevant only if UTC = 1 in RTC_MR.

Downloaded from Arrow.com.

17.6.22 RTC TimeStamp Source Register

Name: RTC_TSSRx
Offset: 0xB8 + x*0x0C [x=0..1]
Reset: 0x00000000
Property: Read-only

This register is cleared after read and the read access also performs a clear on RTC_TSTRx and RTC_TSDRx. The following configuration values are valid for all listed bit names of this register:

0: No alarm generated since the last clear.

1: An alarm has been generated by the corresponding monitor since the last clear.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DET7	DET6	DET5	DET4	DET3	DET2	DET1	DET0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 16, 17, 18, 19, 20, 21, 22, 23 – DETx Tamper Detection on VDDCORE WKUP[8:1] (cleared on read)

Downloaded from Arrow.com.

17.6.23 RTC Tamper Mode Register

Name: RTC_TMR
Offset: 0x58
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	TRLOCK							
Access	W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
	POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – TRLOCK Tamper Registers Lock (Write-once, cleared by VDDCORE reset)

Value	Name	Description
0	UNLOCKED	RTC_TMR and RTC_TDPR can be written.
1	LOCKED	RTC_TMR and RTC_TDPR cannot be written until the next VDDCORE domain reset.

Bits 16, 17, 18, 19, 20, 21, 22, 23 – POLx WKUPx+1 Polarity

Value	Name	Description
0	LOW	If the source of tamper remains low for a debounce period, a tamper event is generated.
1	HIGH	If the source of tamper remains high for a debounce period, a tamper event is generated.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – ENx WKUPx+1 Tamper Source Enable

Value	Name	Description
0	DISABLE	WKUP pin index x+1 is not enabled as a source of tamper.
1	ENABLE	WKUP pin index x+1 is enabled as a source of tamper.

Downloaded from Arrow.com.

17.6.24 RTC Tamper Debounce Period Register

Name: RTC_TDPR
Offset: 0x5C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SELP7	SELP6	SELP5	SELP4	SELP3	SELP2	SELP1	SELP0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PERB[3:0]				PERA[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22, 23 – SELPx WKUPx+1 Debounce Period Selection

Value	Name	Description
0	SEL_PA	WKUP pin index x+1 is debounced with PERA period.
1	SEL_PB	WKUP pin index x+1 is debounced with PERB period.

Bits 7:4 – PERB[3:0] Debounce Period B

Value	Name	Description
0	MD_SLCK_2	The source of tamper must remain active for at least 2 monitoring domain slow clock cycles to generate a tamper event.
1	MD_SLCK_4	The source of tamper must remain active for at least 4 monitoring domain slow clock cycles to generate a tamper event.
2	MD_SLCK_8	The source of tamper must remain active for at least 8 monitoring domain slow clock cycles to generate a tamper event.
3	MD_SLCK_16	The source of tamper must remain active for at least 16 monitoring domain slow clock cycles to generate a tamper event.
4	MD_SLCK_32	The source of tamper must remain active for at least 32 monitoring domain slow clock cycles to generate a tamper event.
5	MD_SLCK_64	The source of tamper must remain active for at least 64 monitoring domain slow clock cycles to generate a tamper event.
6	MD_SLCK_128	The source of tamper must remain active for at least 128 monitoring domain slow clock cycles to generate a tamper event.
7	MD_SLCK_256	The source of tamper must remain active for at least 256 monitoring domain slow clock cycles to generate a tamper event.

Bits 3:0 – PERA[3:0] Debounce Period A

Value	Name	Description
0	MD_SLCK_2	The source of tamper must remain active for at least 2 monitoring domain slow clock cycles to generate a tamper event.

Downloaded from Arrow.com.

Value	Name	Description
1	MD_SLCK_4	The source of tamper must remain active for at least 4 monitoring domain slow clock cycles to generate a tamper event.
2	MD_SLCK_8	The source of tamper must remain active for at least 8 monitoring domain slow clock cycles to generate a tamper event.
3	MD_SLCK_16	The source of tamper must remain active for at least 16 monitoring domain slow clock cycles to generate a tamper event.
4	MD_SLCK_32	The source of tamper must remain active for at least 32 monitoring domain slow clock cycles to generate a tamper event.
5	MD_SLCK_64	The source of tamper must remain active for at least 64 monitoring domain slow clock cycles to generate a tamper event.
6	MD_SLCK_128	The source of tamper must remain active for at least 128 monitoring domain slow clock cycles to generate a tamper event.
7	MD_SLCK_256	The source of tamper must remain active for at least 256 monitoring domain slow clock cycles to generate a tamper event.

Downloaded from Arrow.com.

18. Shutdown Controller (SHDWC)

18.1 Description

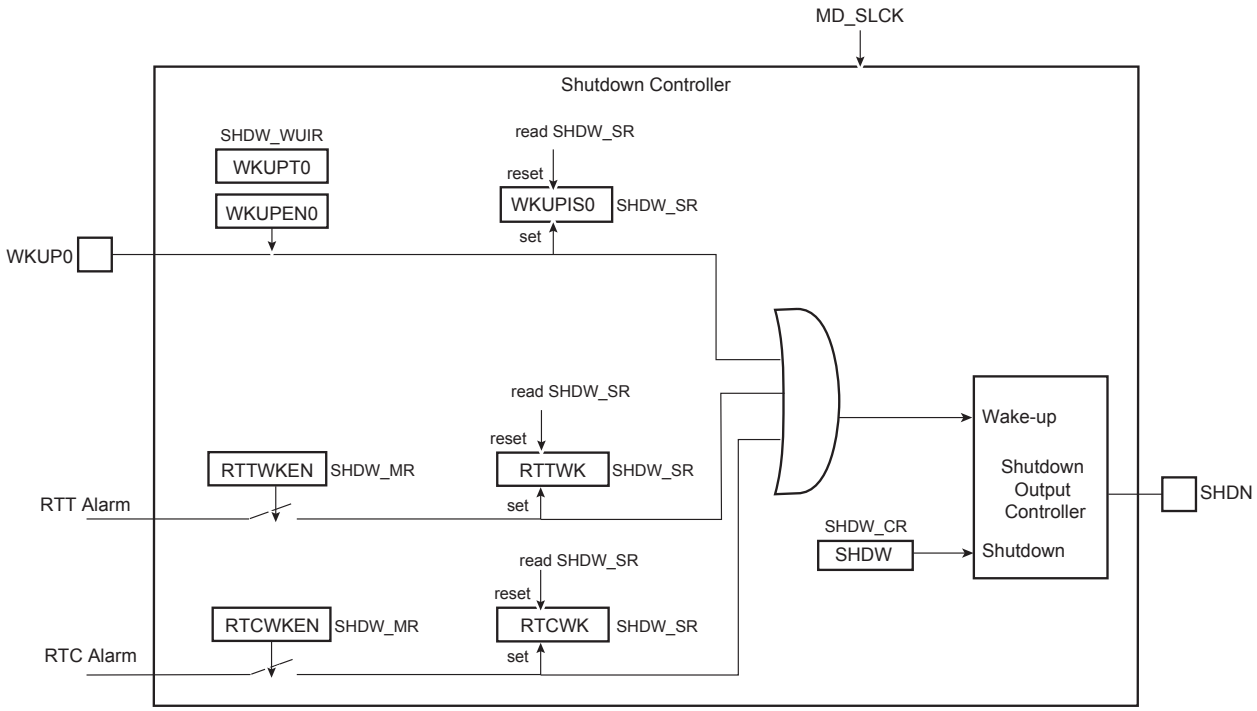
The Shutdown Controller (SHDWC) controls the SHDN output signal (to enable and disable an external power supply circuit), and manages the wake-up events detection.

18.2 Embedded Characteristics

- Shutdown Logic
  - Software assertion of the Shutdown Output Pin (SHDN)
  - Programmable de-assertion from the wake-up events
- Wake-Up Logic
  - Programmable wake-up event detection input pins, and internal wake-up event from RTC and RTT

18.3 Block Diagram

Figure 18-1. SHDWC Block Diagram



18.4 I/O Lines Description

Table 18-1. I/O Lines Description

Name	Description	Type
WKUP0	Wake-up inputs	Input
SHDN	Shutdown output	Output

Downloaded from Arrow.com.

18.5 Product Dependencies

18.5.1 Power Management

The SHDWC is continuously clocked by the Monitoring Domain Slow Clock (MD_SLCK). The Power Management Controller has no effect on the behavior of the SHDWC.

18.6 Functional Description

The SHDWC manages the main power supply. To do so, it is supplied with VDDBU and manages wake-up input pins and one output pin, SHDN.

A typical application connects the pin SHDN to the enable input of the device's power supply circuit. The wake-up inputs (WKUP0) connect to any push-buttons or signal that wake up the system.

The software is able to control the pin SHDN by writing the Shutdown Control Register (SHDW_CR) with the bit SHDW at 1. The shutdown is taken into account only two MD_SLCK cycles after the write of SHDW_CR. This register is password-protected and so the value written should contain the correct key for the command to be taken into account. As a result, the SHDN pin is driven low and the system should be powered down.

18.6.1 Wake-up Inputs

Any level change on the WKUP pin can trigger a wake-up. Wake-up is configured in theMode register (SHDW_MR) and Wakeup Inputs register (SHDW_WUIR). The transition detector can be programmed to detect either a positive or negative transition on the WKUP pin. The detection can also be disabled. Programming is performed by enabling the Wake-up Input (WKUPEN0 bit) and defining the Wake-up Input Type (WKUPT0 bit) in the SHDW_WUIR.

Moreover, a debouncing circuit can be programmed for WKUP. The debouncing circuit filters pulses on WKUP shorter than the programmed value in SHDW_MR.WKUPDBC. If the programmed level change is detected on a pin, a counter starts. When the counter reaches the value programmed in WKUPDBC, the SHDN pin is released. If a new input change is detected before the counter reaches the corresponding value, the counter is stopped and cleared. WKUPIS0 of the Status register (SHDW_SR) reports the detection of the programmed events on WKUP with a reset after the read of SHDW_SR.

The SHDWC can be programmed so as to activate the wake-up using the RTC and RTT alarms (detection of the rising edge event is synchronized with SLCK). This is done by writing the SHDW_MR using the RTCWKEN and RTTWKEN bits. When enabled, the detection of RTC and RTT alarms is reported in the RTCWK and RTTWK bits of SHDW_SR. They are cleared after reading SHDW_SR. When using the RTC and RTT alarms to wake up the system, the user must ensure that RTC and RTT alarm status flags are cleared before shutting down the system. Otherwise, no rising edge of the status flags may be detected and the wake-up fails.

Downloaded from Arrow.com.

18.7 Register Summary

Offset	Name	Bit Pos.								
0x00	SHDW_CR	31:24	KEY[7:0]							
		23:16								
		15:8								
		7:0								SHDW
0x04	SHDW_MR	31:24						WKUPDBC[2:0]		
		23:16							RTCWKEN	RTTWKEN
		15:8								
		7:0								
0x08	SHDW_SR	31:24								
		23:16								WKUPIS0
		15:8								
		7:0			RTCWK	RTTWK				WKUPS
0x0C	SHDW_WUIR	31:24								
		23:16								WKUPT0
		15:8								
		7:0								WKUPEN0

Downloaded from Arrow.com.

18.7.1 SHDWC Control Register

Name: SHDW_CR
Offset: 0x00
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SHDW
Access								W
Reset								–

Bits 31:24 – KEY[7:0] Password

Value	Name	Description
0xA5	PASSWD	Writing any other value in this field aborts the write operation.

Bit 0 – SHDW Shutdown Command

Value	Description
0	No effect.
1	If KEY value is correct, asserts the SHDN pin.

Downloaded from Arrow.com.

18.7.2 SHDWC Mode Register

Name: SHDW_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
						WKUPDBC[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
							RTCWKEN	RTTWKEN
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 26:24 – WKUPDBC[2:0] Wake-up Inputs Debouncer Period

Value	Name	Description
0	IMMEDIATE	Immediate, no debouncing, detected active at least on one MD_SLCK edge
1	3_SLCK	WKUP shall be in its active state for at least 3 MD_SLCK periods
2	32_SLCK	WKUP shall be in its active state for at least 32 MD_SLCK periods
3	512_SLCK	WKUP shall be in its active state for at least 512 MD_SLCK periods
4	4096_SLCK	WKUP shall be in its active state for at least 4,096 SLCK periods
5	32768_SLCK	WKUP shall be in its active state for at least 32,768 MD_SLCK periods

Bit 17 – RTCWKEN Real-time Clock Wake-up Enable

Value	Description
0	The RTC Alarm signal has no effect on the SHDWC.
1	The RTC Alarm signal forces the de-assertion of the SHDN pin.

Bit 16 – RTTWKEN Real-time Timer Wake-up Enable

Value	Description
0	The RTT Alarm signal has no effect on the SHDWC.
1	The RTT Alarm signal forces the de-assertion of the SHDN pin.

Downloaded from Arrow.com.

18.7.3 SHDWC Status Register

Name: SHDW_SR
Offset: 0x08
Reset: 0x00000000
Property: Read-only

Note: The events are detected only when the system is in Backup mode.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								WKUPIS0
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RTCWK	RTTWK				WKUPS
Access			R	R				R
Reset			0	0				0

Bit 16 – WKUPIS0 Wake-up 0 Input Status

Value	Name	Description
0	DISABLE	The wake-up 0 input is disabled, or was inactive at the time the debouncer triggered a wake-up event.
1	ENABLE	The wake-up 0 input was active at the time the debouncer triggered a wake-up event.

Bit 5 – RTCWK Real-time Clock Wake-up

Value	Description
0	No wake-up alarm from the RTC occurred since the last read of SHDW_SR.
1	At least one wake-up alarm from the RTC occurred since the last read of SHDW_SR.

Bit 4 – RTTWK Real-time Timer Wake-up

Value	Description
0	No wake-up alarm from the RTT occurred since the last read of SHDW_SR.
1	At least one wake-up alarm from the RTT occurred since the last read of SHDW_SR.

Bit 0 – WKUPS WKUP Wake-up Status

Value	Name	Description
0	NO	No wake-up due to the assertion of the WKUP pins has occurred since the last read of SHDW_SR.
1	PRESENT	At least one wake-up due to the assertion of the WKUP pins has occurred since the last read of SHDW_SR.

Downloaded from Arrow.com.

18.7.4 SHDWC Wake-up Inputs Register

Name: SHDW_WUIR
Offset: 0x0C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								R
Reset								0

Bit 16 – WKUPT0 Wake-up 0 Input Type

Value	Name	Description
0	LOW	A falling edge followed by a low level on the wake-up 0 input, for a period defined by WKUPDBC, forces wake-up of the core power supply.
1	HIGH	A rising edge followed by a high level on the wake-up 0 input, for a period defined by WKUPDBC, forces wake-up of the core power supply.

Bit 0 – WKUPEN0 Wake-up 0 Input Enable

Value	Name	Description
0	DISABLE	The wake-up 0 input has no wake-up effect.
1	ENABLE	The wake-up 0 input forces wake-up of the core power supply.

Downloaded from Arrow.com.

19. Periodic Interval Timer (PIT)

19.1 Description

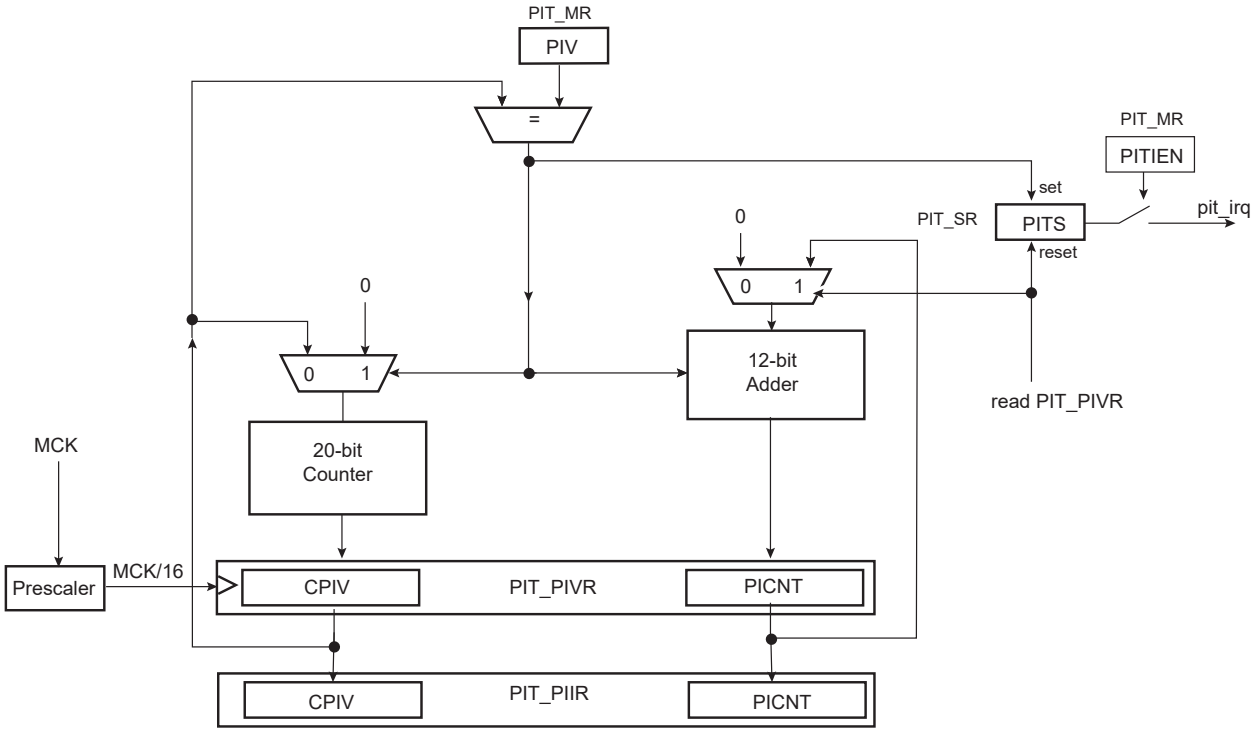
The Periodic Interval Timer (PIT) provides the operating system's scheduler interrupt. It is designed to offer maximum accuracy and efficient management, even for systems with long response time.

19.2 Embedded Characteristics

- 20-bit Programmable Counter plus 12-bit Interval Counter
- Reset-on-read Feature
- Both Counters Work on Master Clock/16

19.3 Block Diagram

Figure 19-1. Periodic Interval Timer



19.4 Functional Description

The Periodic Interval Timer provides periodic interrupts for use by operating systems.

The PIT provides a programmable overflow counter and a reset-on-read feature. It is built around two counters: a 20-bit CPIV counter and a 12-bit PICNT counter. Both counters work at Master Clock /16.

The first 20-bit CPIV counter increments from 0 up to a programmable overflow value set in the field PIV of the Mode Register (PIT_MR). When the counter CPIV reaches this value, it resets to 0 and increments the Periodic Interval Counter, PICNT. The status bit PITS in the Status Register (PIT_SR) rises and triggers an interrupt, provided the interrupt is enabled (PITIEN in PIT_MR).

Writing a new PIV value in PIT_MR does not reset/restart the counters.

Downloaded from Arrow.com.

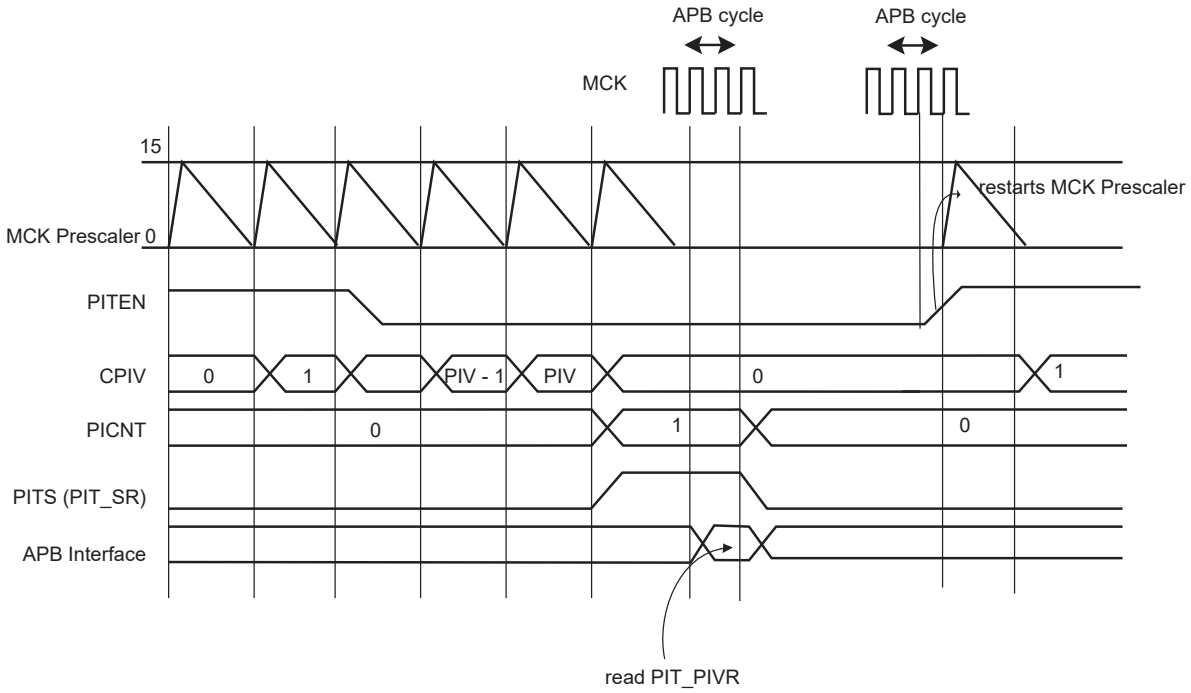
When CPIV and PICNT values are obtained by reading the Periodic Interval Value Register (PIT_PIVR), the overflow counter (PICNT) is reset and the PITS bit is cleared, thus acknowledging the interrupt. The value of PICNT gives the number of periodic intervals elapsed since the last read of PIT_PIVR.

When CPIV and PICNT values are obtained by reading the Periodic Interval Image Register (PIT_PIIR), there is no effect on the counters CPIV and PICNT, nor on the bit PITS. For example, a profiler can read PIT_PIIR without clearing any pending interrupt, whereas a timer interrupt clears the interrupt by reading PIT_PIVR.

The PIT may be enabled/disabled using the PITEN bit in the PIT_MR register (disabled on reset). The PITEN bit only becomes effective when the CPIV value is 0. The figure below illustrates the PIT counting. After the PIT Enable bit is reset (PITEN = 0), the CPIV goes on counting until the PIV value is reached, and is then reset. PIT restarts counting, only if the PITEN is set again.

The PIT is stopped when the core enters debug state.

Figure 19-2. Enabling/Disabling PIT with PITEN



Downloaded from Arrow.com.

19.5 Register Summary

Offset	Name	Bit Pos.								
0x00	PIT_MR	31:24							PITIEN	PITEN
		23:16					PIV[19:16]			
		15:8	PIV[15:8]							
		7:0	PIV[7:0]							
0x04	PIT_SR	31:24								
		23:16								
		15:8								
		7:0								PITS
0x08	PIT_PIVR	31:24	PICNT[11:4]							
		23:16	PICNT[3:0]				CPIV[19:16]			
		15:8	CPIV[15:8]							
		7:0	CPIV[7:0]							
0x0C	PIT_PIIR	31:24	PICNT[11:4]							
		23:16	PICNT[3:0]				CPIV[19:16]			
		15:8	CPIV[15:8]							
		7:0	CPIV[7:0]							

Downloaded from Arrow.com.

19.5.1 Periodic Interval Timer Mode Register

Name: PIT_MR
Offset: 0x00
Reset: 0x000FFFFF
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
							PITIEN	PITEN
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
					PIV[19:16]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	15	14	13	12	11	10	9	8
	PIV[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 25 – PITIEN Period Interval Timer Interrupt Enable

Value	Description
0	The bit PITS in PIT_SR has no effect on the interrupt.
1	The bit PITS in PIT_SR asserts an interrupt.

Bit 24 – PITEN Period Interval Timer Enabled

Value	Description
0	The Periodic Interval Timer is disabled when the PIV value is reached.
1	The Periodic Interval Timer is enabled.

Bits 19:0 – PIV[19:0] Periodic Interval Value
Defines the value compared with the primary 20-bit counter of the Periodic Interval Timer (CPIV). The period is equal to (PIV + 1).

Downloaded from Arrow.com.

19.5.2 Periodic Interval Timer Status Register

Name: PIT_SR
Offset: 0x04
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								PITS
Access								R
Reset								0

Bit 0 – PITS Periodic Interval Timer Status

Value	Description
0	The Periodic Interval timer has not reached PIV since the last read of PIT_PIVR.
1	The Periodic Interval timer has reached PIV since the last read of PIT_PIVR.

Downloaded from Arrow.com.

19.5.3 Periodic Interval Timer Value Register

Name: PIT_PIVR
Offset: 0x08
Reset: 0x00000000
Property: Read-only

Reading this register clears PITS in PIT_SR.

Bit	31	30	29	28	27	26	25	24
	PICNT[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PICNT[3:0]				CPIV[19:16]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CPIV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CPIV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:20 – PICNT[11:0] Periodic Interval Counter
Returns the number of occurrences of periodic intervals since the last read of PIT_PIVR.

Bits 19:0 – CPIV[19:0] Current Periodic Interval Value
Returns the current value of the periodic interval timer.

Downloaded from Arrow.com.

19.5.4 Periodic Interval Timer Image Register

Name: PIT_PIIR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	PICNT[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PICNT[3:0]				CPIV[19:16]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CPIV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CPIV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:20 – PICNT[11:0] Periodic Interval Counter
Returns the number of occurrences of periodic intervals since the last read of PIT_PIVR.

Bits 19:0 – CPIV[19:0] Current Periodic Interval Value
Returns the current value of the periodic interval timer.

Downloaded from Arrow.com.

20. 64-bit Periodic Interval Timer (PIT64B)

20.1 Description

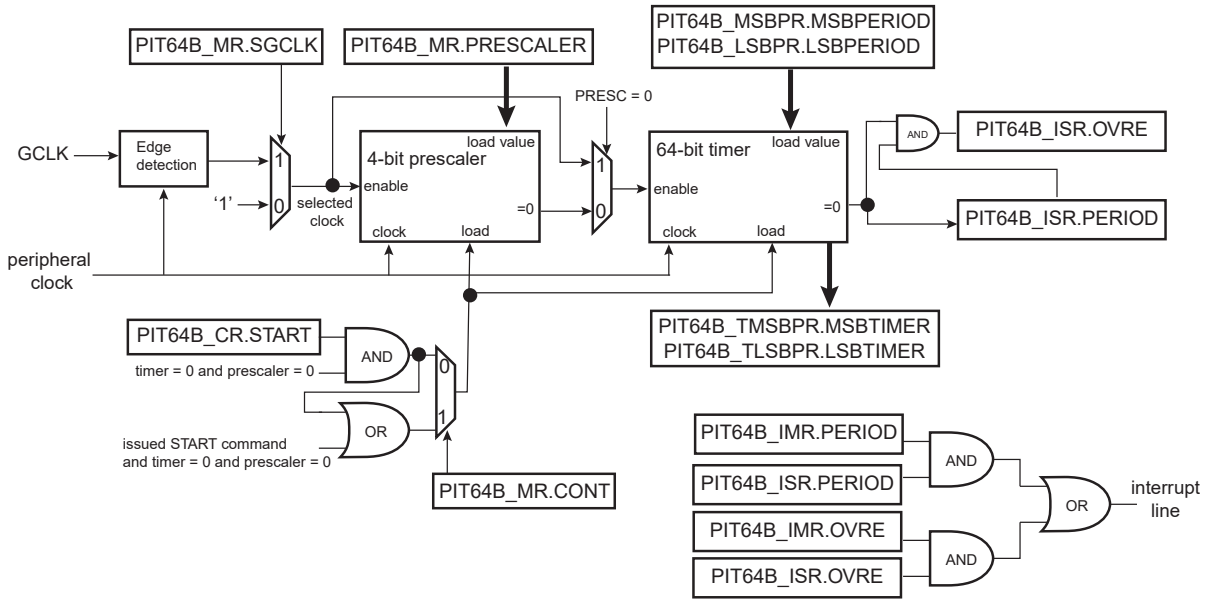
The 64-bit Periodic Interval Timer (PIT64B) provides the operating system scheduler interrupt, as well as any periodic source of interrupt to software. It is designed to offer maximum accuracy and efficient management, even for systems with long response times.

20.2 Embedded Characteristics

- 4-bit Prescaler
- 64-bit Timer
- Single Shot or Continuous Mode
- Safety/Security Access Reports
- Register Write protection

20.3 Block Diagram

Figure 20-1. Block Diagram



20.4 Product Dependencies

20.4.1 Power Management

The Power Management Controller (PMC) controls the PIT64B clock in order to save power. The programmer must first enable the PIT64B clock in the PMC before using the PIT64B.

After a hardware reset, the PIT64B clock is disabled by default.

20.4.2 Interrupt Generation

The PIT64B interface has an interrupt line connected to the Interrupt Controller.

Handling the PIT64B interrupt requires programming the Interrupt Controller before configuring the PIT64B.

Downloaded from Arrow.com.

20.5 Functional Description

20.5.1 Timer Clock Source

The two clock sources for the 64-bit timer are the peripheral clock and the generic clock (GCLK), which can be fully asynchronous to the peripheral clock. The selected clock can be prescaled before triggering the 64-bit timer.

The GCLK is selected as source clock for the prescaler when the SGCLK bit, in the Mode register (PIT64B_MR), is written to 1. The prescaler is active as soon as PIT64B_MR.PRESCALER>0.

If PIT64B_MR.PRESCALER=0, the timer is triggered either on each rising edge detection event of the GCLK if PIT64B_MR.SGCLK is written to 1, or on each rising edge of the peripheral clock.

If GCLK is selected, the frequency must be at least 3 times lower than the peripheral clock.

20.5.2 Single Period Mode

When the PIT64B_MR.CONT bit is written to 0, the PIT64B produces a single timer event. The timer period starts as soon as the START bit, in the Control register (PIT64B_CR), is written to 1. The period is defined by configuring the LSBPERIOD field in the LSB Period register (PIT64B_LSBPR) and the MSBPERIOD field in the MSB Period register (PIT64B_MSBPR). When the START command is issued, the 64-bit timer loads 0 and increments up to LSBPERIOD and MSBPERIOD field value minus 1, then automatically reloads 0 and stops.

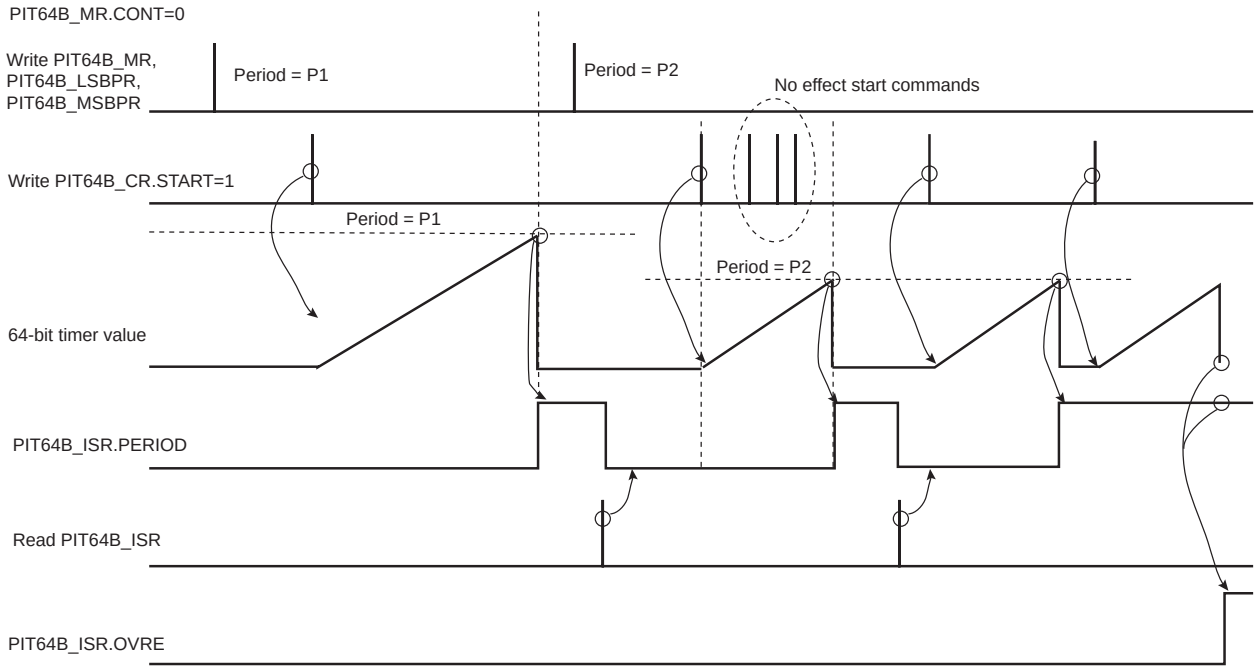
When time reaches the maximum value, the PERIOD flag, in the Interrupt Status register (PIT64B_ISR), is set. No other period will be started until a new START command is issued.

After a period is started and while it is not elapsed, any new values written in PIT64B_MR, PIT64B_LSBPR or PIT64B_MSBPR have no effect on the current period if bit PIT64B_MR.SMOD=0 (see Figure Single Waveform in Single Period Mode if bit PIT64B_MR.SMOD=0

If PIT64B_MR.SMOD=1 a start is also performed as soon as PIT64B_LSBPR is written, thus a modification of the period can be performed on-the-fly with a single write operation if the period requires no more than 32 bits. When writing a 64-bit value, the 32-bit MSB part must be configured first, followed by the 32-bit LSB part (see Figure Waveform in Single Period Mode if bit PIT64B_MR.SMOD=1). When configuring a value lower or equal to 32 bits after processing a period defined on 64 bits, first PIT64B_MSBPR must be written to 0, and then the 32-bit LSB must be written into PIT64B_LSBPR.

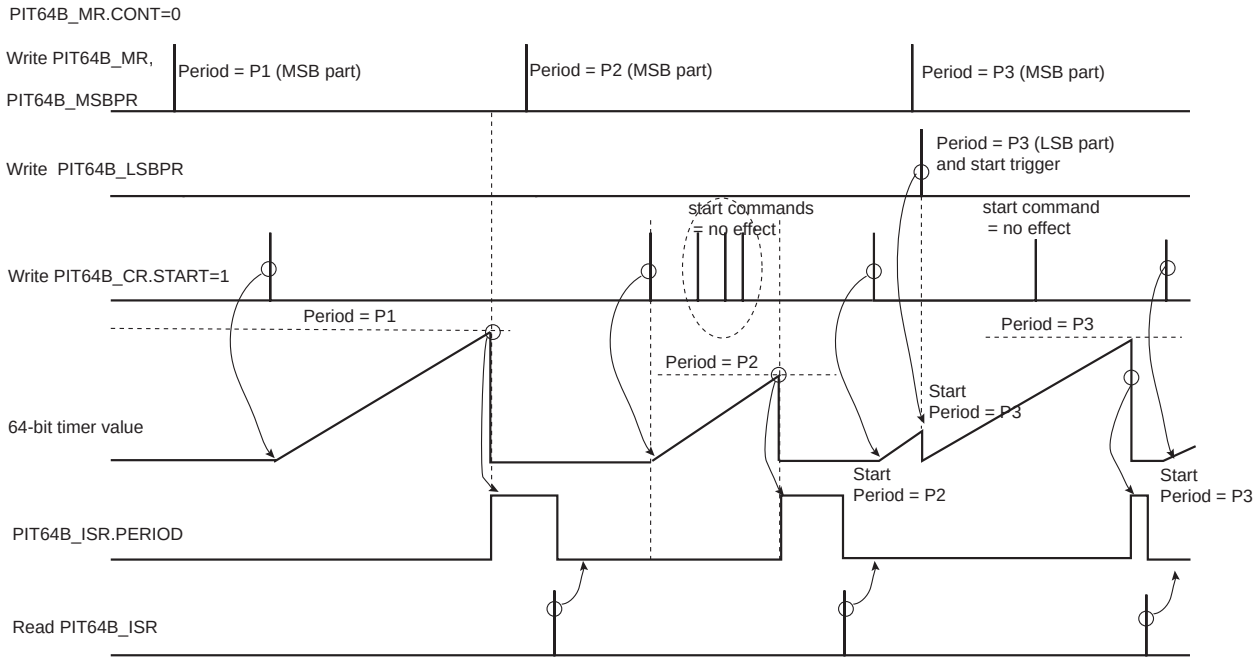
If PIT64B_CR.SWRST is written to 1, the current period is immediately stopped.

Figure 20-2. Single Waveform in Single Period Mode if bit PIT64B_MR.SMOD=0



Downloaded from Arrow.com.

Figure 20-3. Waveform in Single Period Mode if bit PIT64B_MR.SMOD=1



20.5.3 Continuous Period Mode

When the PIT64B_MR.CONT bit is written to 1, the PIT64B continuously produces timer events. The timer is started as soon as the PIT64B_CR.START bit is written to 1. The period is defined by configuring the PIT64B_LSBPR.LSBPERIOD field and PIT64B_MSBPR.MSBPERIOD field. When the START command is issued, the 64-bit timer loads 0 and increments up to LSBPERIOD and MSBPERIOD field value minus 1, then automatically reloads 0 anc restarts a new counting period until bit PIT64B_CR.SWRST is written to 1.

When the timer reaches its maximum value, the flag PIT64B_ISR.PERIOD is set. PIT64B_ISR.PERIOD is cleared when reading PIT64B_ISR. If a new period elapses and the PIT64B_ISR.PERIOD is 1, the PIT64B_ISR.OVRE flag is set to indicate a potential latency at system level

After the START command has been issued, any new values written in PIT64B_MR, PIT64B_LSBPR or PIT64B_MSBPR have no effect on the current period if bit PIT64B_MR.SMOD=0 (see Figure Waveform in Continuous Period Mode if the bit PIT64B_MR.SMOD=0). A software reset must be issued before configuring new values in PIT64B_LSBPR and PIT64B_MSBPR if bit PIT64B_MR.SMOD=0.

If PIT64B_MR.SMOD=1 a start can be also performed as soon as PIT64B_LSBPR is written, thus a modification of the period can be performed on-the-fly with a single write operation if the period requires no more than 32 bits. When writing a 64-bit value, the 32-bit MSB part must be configured first followed by a 32-bit LSB part (see Figure Waveform in Continuous Period Mode if the bit PIT64B_MR.SMOD=1). When configuring a value lower or equal to 32 bits after processing a period defined on 64 bits, first PIT64B_MSBPR must be written to 0, and then the 32-bit LSB must be written into PIT64B_LSBPR.

If PIT64B_CR.SWRST is written to 1, the current period is immediately stopped.

Downloaded from Arrow.com.

Figure 20-4. Waveform in Continuous Period Mode if bit PIT64B_MR.SMOD=0

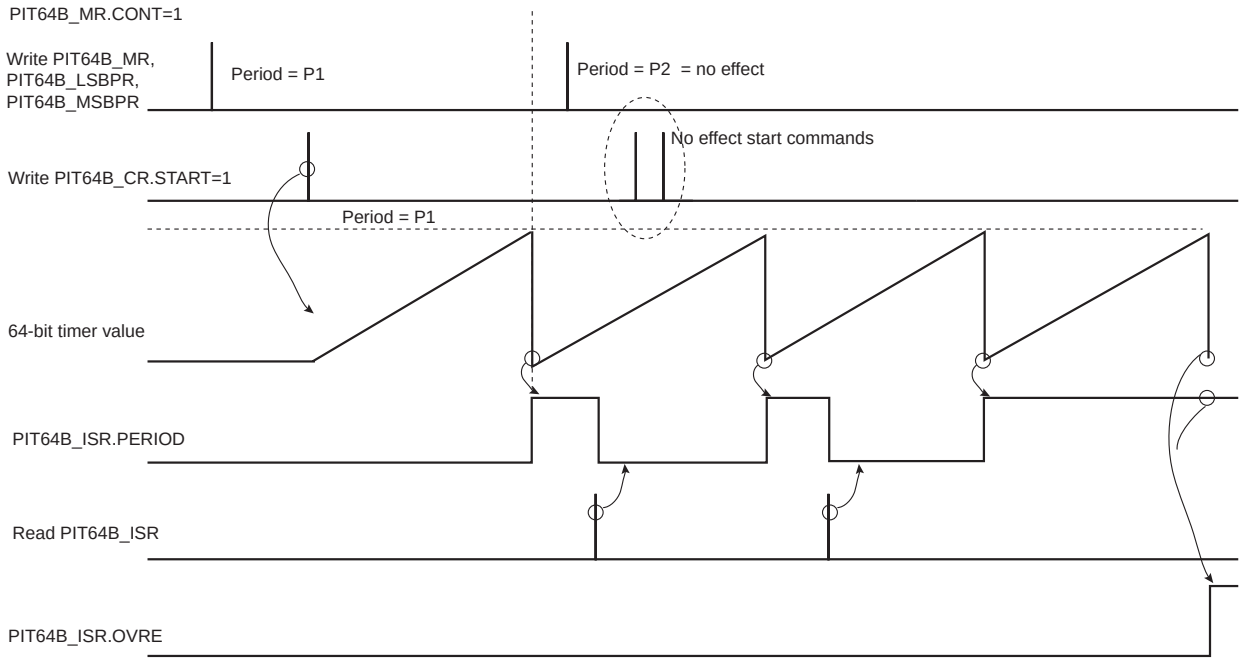
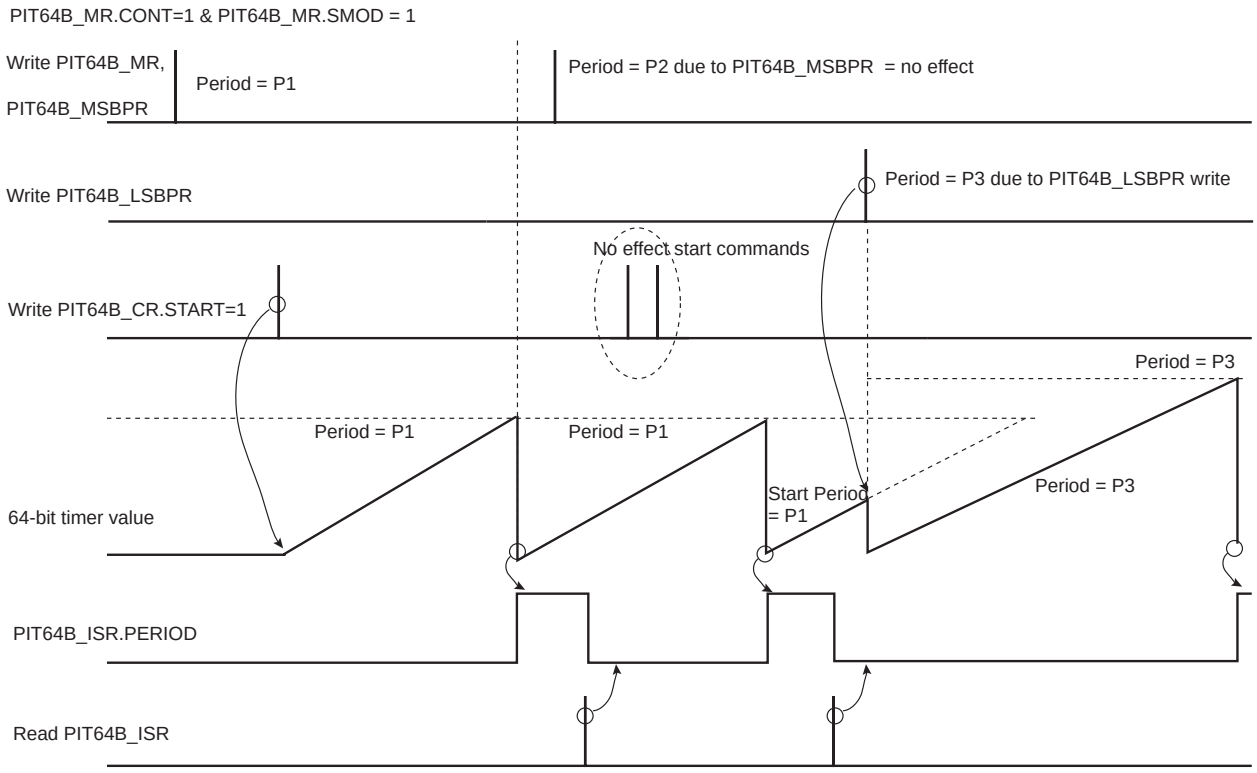


Figure 20-5. Waveform in Continuous Period Mode if bit PIT64B_MR.SMOD=1



20.5.4 Security and Safety Analysis and Reports

Several types of checks are performed when the PIT64B is running.

The peripheral clock of the PIT64B is monitored by a specific circuitry to detect abnormal waveforms on the internal clock that may affect the behavior of the PIT64B. Corruption on the triggering edge of the clock or a pulse with a

Downloaded from Arrow.com.

minimum duration may be identified. If the CGD flag is set in the Write Protection Status register (PIT64B_WPSR), an abnormal condition occurred on the peripheral clock. This flag is not set under normal operating conditions.

The internal timer sequence of the PIT64B is also monitored and if an abnormal state is detected, the flag PIT64B_WPSR.SEQE is set. This flag is not set under normal operating conditions.

The software accesses to the PIT64B are monitored and if an incorrect access is performed, the flag PIT64B_WPSR.SWE is set. The type of incorrect/abnormal software access is reported in the PIT64B_WPSR.SWETYP field (see PIT64B Write Protection Status Register for details), e.g., writing PIT64B_MR, PIT64B_LSBPR (if PIT64B_MR.SMOD=0), PIT64B_MSBPR (if PIT64B_MR.SMOD=0) when the timer is running (after a START command has been issued) is an error. PIT64B_WPSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The flags CGD, SEQE, SWE and WPVS are automatically cleared when PIT64B_WPSR is read.

If one of these flags is set, the PIT64B_ISR.SECE flag is set and can trigger an interrupt if the SECE bit, in the Interrupt Mask register (PIT64B_IMR), is ‘1’. SECE is cleared by reading PIT64B_ISR.

20.5.5 Register Write Protection

To prevent any single software error from corrupting PIT64B behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the PIT64B Write Protection Mode Register (PIT64B_WPMR).

If a write access to a write-protected register is detected, the WPVS (Write Protection Violation Status) flag in the PIT64B Write Protection Status Register (PIT64B_WPSR) is set and the WPVSRC (Write Protection Violation Source) field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading PIT64B_WPSR.

The following registers can be write-protected when WPEN is set in PIT64B_WPMR:

- PIT64B Mode Register
- PIT64B LSB Period Register
- PIT64B MSB Period Register

Note: PIT64B LSB Period Register and PIT64B MSB Period Register are not write-protected if PIT64B_MR.SMOD=1.

The following registers can be write-protected when WPITEN is set in PIT64B_WPMR:

- PIT64B Interrupt Enable Register
- PIT64B Interrupt Disable Register

The following register can be write-protected when WPCREN is set in PIT64B_WPMR:

- PIT64B Control Register

Downloaded from Arrow.com.

20.6 Register Summary

Offset	Name	Bit Pos.								
0x00	PIT64B_CR	31:24								
		23:16								
		15:8								SWRST
		7:0								START
0x04	PIT64B_MR	31:24								
		23:16								
		15:8					PRESCALER[3:0]			
		7:0				SMOD	SGCLK			CONT
0x08	PIT64B_LSBPR	31:24	LSBPERIOD[31:24]							
		23:16	LSBPERIOD[23:16]							
		15:8	LSBPERIOD[15:8]							
		7:0	LSBPERIOD[7:0]							
0x0C	PIT64B_MSBPR	31:24	MSBPERIOD[31:24]							
		23:16	MSBPERIOD[23:16]							
		15:8	MSBPERIOD[15:8]							
		7:0	MSBPERIOD[7:0]							
0x10	PIT64B_IER	31:24								
		23:16								
		15:8								
		7:0				SECE			OVRE	PERIOD
0x14	PIT64B_IDR	31:24								
		23:16								
		15:8								
		7:0				SECE			OVRE	PERIOD
0x18	PIT64B_IMR	31:24								
		23:16								
		15:8								
		7:0				SECE			OVRE	PERIOD
0x1C	PIT64B_ISR	31:24								
		23:16								
		15:8								
		7:0				SECE			OVRE	PERIOD
0x20	PIT64B_TLSBR	31:24	LSBTIMER[31:24]							
		23:16	LSBTIMER[23:16]							
		15:8	LSBTIMER[15:8]							
		7:0	LSBTIMER[7:0]							
0x24	PIT64B_TMSBR	31:24	MSBTIMER[31:24]							
		23:16	MSBTIMER[23:16]							
		15:8	MSBTIMER[15:8]							
		7:0	MSBTIMER[7:0]							
0x28 ... 0xE3	Reserved									
0xE4	PIT64B_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0				FIRSTE		WPCREN	WPITEN	WPEN
0xE8	PIT64B_WPSR	31:24	ECLASS						SWETYP[1:0]	
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS

Downloaded from Arrow.com.

20.6.1 PIT64B Control Register

Name: PIT64B_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the PIT64B Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
Access								W
Reset								–

Bit 8 – SWRST Software Reset

Value	Description
0	No effect.
1	Performs a software reset, clears the configuration and stops any timer period in progress.

Bit 0 – START Start Timer

Value	Description
0	No effect.
1	The timer counter is started for 1 or more periods. If the START command is applied during a non-elapsed timer period, there is no effect. Thus, in Continuous mode, the SWRST command is the only command to stop the PIT64B. If PIT64B_MR.SMOD=1 a start is also performed as soon as PIT64B_LSBPR is written (see 20.5.2 Single Period Mode and 20.5.3 Continuous Period Mode).

Downloaded from Arrow.com.

20.6.2 PIT64B Mode Register

Name: PIT64B_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PIT64B Write Protection Mode Register.

When the timer is running, writing a value to this register has no effect. The value written is this register is loaded anytime before a START command is issued.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					PRESCALER[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
				SMOD	SGCLK			CONT
Access				R/W	R/W			R/W
Reset				0	0			0

Bits 11:8 – PRESCALER[3:0] Prescaler Period

Value	Description
0	A prescaler divider of 1 is used.
1-15	The 64-bit timer is incremented at each (PRESCALER+1)x selected period (see SGCLK).

Bit 4 – SMOD Start Mode

Value	Description
0	Writing PIT64B_LSBPR does not start the timer period.
1	Writing PIT64B_LSBPR starts the timer period.

Bit 3 – SGCLK Generic Clock Selection Enable
If GCLK is asynchronous to the peripheral clock, a jitter of 1 peripheral clock period is created on the periodic interval event when Continuous mode is selected.

Value	Description
0	The prescaler is triggered at each rising edge of “Peripheral clock” and the timer is triggered.
1	GCLK clock is selected as clock source of the 8-bit prescaler.

Bit 0 – CONT Continuous Mode

Value	Description
0	A single period interrupt is generated from a START command.
1	Continuous periodic interrupts are generated after a single START command.

Downloaded from Arrow.com.

20.6.3 PIT64B LSB Period Register

Name: PIT64B_LSBPR
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PIT64B Write Protection Mode Register or if PIT64B_MR.SMOD=1.

When the timer is running, if PIT64B_MR.SMOD=0, writing a value to this register has no effect. The value written is this register must be loaded anytime before a START command is issued if PIT64B_MR.SMOD=0. If PIT64B_MR.SMOD=1, a write access to this register restarts a timer period.

Bit	31	30	29	28	27	26	25	24
	LSBPERIOD[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LSBPERIOD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LSBPERIOD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LSBPERIOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – LSBPERIOD[31:0] 32 LSB of the Timer Period
This field defines the 32 LSB of the timer period. The timer period is defined by selected clock x {MSBPERIOD,LSBPERIOD}.

Downloaded from Arrow.com.

20.6.4 PIT64B MSB Period Register

Name: PIT64B_MSBPR
Offset: 0x0C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PIT64B Write Protection Mode Register.

When the timer is running, if PIT64B_MR.SMOD=0, writing a value to this register has no effect. The value written is this register must be loaded anytime before a START command is issued if PIT64B_MR.SMOD=0.

Bit	31	30	29	28	27	26	25	24
	MSBPERIOD[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MSBPERIOD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MSBPERIOD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MSBPERIOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – MSBPERIOD[31:0] 32 MSB of the Timer Period
This field defines the 32 MSB of the timer period. The timer period is defined by selected clock x {MSBPERIOD,LSBPERIOD}.

Downloaded from Arrow.com.

20.6.5 PIT64B Interrupt Enable Register

Name: PIT64B_IER
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PIT64B Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Enables the corresponding interrupt

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				SECE			OVRE	PERIOD
Access				W			W	W
Reset				–			–	–

Bit 4 – SECE Safety and/or Security Report Interrupt Enable

Bit 1 – OVRE Overrun Error Interrupt Enable

Bit 0 – PERIOD Elapsed Timer Period Interrupt Enable

Downloaded from Arrow.com.

20.6.6 PIT64B Interrupt Disable Register

Name: PIT64B_IDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PIT64B Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Disables the corresponding interrupt

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				SECE			OVRE	PERIOD
Access				W			W	W
Reset				–			–	–

Bit 4 – SECE Safety and/or Security Report Interrupt Disable

Bit 1 – OVRE Overrun Error Interrupt Disable

Bit 0 – PERIOD Elapsed Timer Period Interrupt Disable

Downloaded from Arrow.com.

20.6.7 PIT64B Interrupt Mask Register

Name: PIT64B_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: Corresponding interrupt is not enabled.

1: Corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				SECE			OVRE	PERIOD
Access				R			R	R
Reset				0			0	0

Bit 4 – SECE Safety and/or Security Report Interrupt Mask

Bit 1 – OVRE Overrun Error Interrupt Mask

Bit 0 – PERIOD Elapsed Timer Period Interrupt Mask

Downloaded from Arrow.com.

20.6.8 PIT64B Interrupt Status Register

Name: PIT64B_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				SECE			OVRE	PERIOD
Access				R			R	R
Reset				0			0	0

Bit 4 – SECE Safety/Security Report (cleared on read)

Value	Description
0	There has been no security report in PIT64B_WPSR since the last read of PIT64B_ISR.
1	One security flag has been set in PIT64B_WPSR since the last read of PIT64B_ISR.

Bit 1 – OVRE Overrun Error (cleared on read)

Value	Description
0	No multiple rollovers occurred since the last read of PIT64B_ISR.
1	More than 1 rollover occurred since the last read of PIT64B_ISR.

Bit 0 – PERIOD Elapsed Timer Period Status Flag (cleared on read)

Value	Description
0	No timer rollover occurred since the last read of PIT64B_ISR.
1	A timer rollover occurred since the last read of PIT64B_ISR.

Downloaded from Arrow.com.

20.6.9 PIT64B Timer LSB Register

Name: PIT64B_TLSBR
Offset: 0x20
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	LSBTIMER[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LSBTIMER[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LSBTIMER[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LSBTIMER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – LSBTIMER[31:0] Current 32 LSB of the Timer
This field returns the 32 LSB of the current timer value.

Downloaded from Arrow.com.

20.6.10 PIT64B Timer MSB Register

Name: PIT64B_TMSBR
Offset: 0x24
Reset: 0x00000000
Property: Read-only

When operating with a timer value greater than 32 bits (PIT64B_MSBPR.MSBPERIOD > 0), the PIT64B_TLSBR must be read first, followed by the read of PIT64B_TMSBR. This sequence generates an atomic read of the 64-bit timer value whatever the lapse of time between the accesses. When operating with a timer value up to 32 bits (PIT64B_MSBPR.MSBPERIOD=0), reading PIT64B_TMSBR is not required.

Bit	31	30	29	28	27	26	25	24
	MSBTIMER[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MSBTIMER[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MSBTIMER[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MSBTIMER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – MSBTIMER[31:0] Current 32 MSB of the Timer
This field returns the 32 MSB of the current timer value.

Downloaded from Arrow.com.

20.6.11 PIT64B Write Protection Mode Register

Name: PIT64B_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				FIRSTE		WPCREN	WPITEN	WPEN
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x504954	PASSWD	Writing any other value in this field aborts the write operation of the WPCREN, WPITEN and WPEN bits. Always reads as 0.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in PIT64B_WPSR.WPVSRC and the last software control error type is reported in PIT64B_WPSR.SWETYP. The PIT64B_ISR.SECE flag is set at the first error occurring within a series.
1	Only the first write protection violation source is reported in PIT64B_WPSR.WPVSRC and only the first software control error type is reported in PIT64B_WPSR.SWETYP. The PIT64B_ISR.SECE flag is set at the first error occurring within a series.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x504954 (“PIT” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x504954 (“PIT” in ASCII).

Bit 1 – WPITEN Write Protection Interruption Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x504954 (“PIT” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x504954 (“PIT” in ASCII).

Bit 0 – WPEN Write Protection Enable
See Section 6.5 “Register Write Protection” for the list of registers that can be protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x504954 (“PIT” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x504954 (“PIT” in ASCII).

Downloaded from Arrow.com.

20.6.12 PIT64B Write Protection Status Register

Name: PIT64B_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS						SWETYP[1:0]	
Access	R						R	R
Reset	0						0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SWE	SEQE	CGD	WPVS
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – ECLASS Software Error Class (cleared on read)
0 (WARNING): An abnormal access that does not affect system functionality.
1 (ERROR): A write access is performed into PIT64B_MR, PIT64B_LSBR, PIT64B_MSBR while the PIT64B is running.

Bits 25:24 – SWETYP[1:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	A write-only register has been read (warning).
1	WRITE_RO	A write access has been performed on a read-only register (warning).
2	UNDEF_RW	Access to an undefined address (warning).
3	WEIRD_ACTION	A write access is performed into PIT64B_MR, PIT64B_LSBR, PIT64B_MSBR while the PIT64B is running (abnormal).

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS=1, WPVSRC indicates the register address offset at which a write access has been attempted.
When WPVS=0 and SWE=1, WPVSRC reports the address of the incorrect software access. As soon as WPVS=1, WPVSRC returns the address of the write-protected violation.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of PIT64B_WPSR.
1	A software error has occurred since the last read of PIT64B_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of PIT64B_WPSR.

Downloaded from Arrow.com.

Value	Description
1	A peripheral internal sequencer error has occurred since the last read of PIT64B_WPSR. This flag can only be set under abnormal operating conditions.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	The clock monitoring circuitry has not been corrupted since the last read of PIT64B_WPSR. Under normal operating conditions, this bit is always cleared.
1	The clock monitoring circuitry has been corrupted since the last read of PIT64B_WPSR. This flag can only be set in case of abnormal clock signal waveform (glitch).

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protection violation has occurred since the last read of the PIT64B_WPSR.
1	A write protection violation has occurred since the last read of the PIT64B_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

21. Debug Unit (DBGU)

21.1 Description

The Debug Unit (DBGU) provides a single entry point from the processor for access to all the debug capabilities of the product.

The DBGU features a two-pin UART that can be used for communication and trace purposes and offers an ideal medium for in-situ programming solutions.

Moreover, the association with a DMA controller permits packet handling for these tasks with processor time reduced to a minimum.

The DBGU also makes the Debug Communication Channel (DCC) signals provided by the In-circuit Emulator of the ARM processor visible to the software. These signals indicate the status of the DCC read and write registers and generate an interrupt to the ARM processor, making possible the handling of the DCC under interrupt control.

Chip identifier registers permit recognition of the device and its revision. These registers indicate the sizes and types of the on-chip memories, as well as the set of embedded peripherals.

A Force NTRST capability enables the software to decide whether to prevent access to the system via the In-circuit Emulator (ICE). This disables system access through the processor's ICE, thus protecting the code stored in ROM by asserting the NTRST line of the processor's ICE.

21.2 Embedded Characteristics

- ICE Access Prevention
- Debug Communication Channel (DCC) Support
- Chip ID Registers
- Two-pin UART
  - Independent receiver and transmitter with a common programmable baud rate generator
  - Baud rate can be driven by processor-independent generic source clock
  - Even, odd, mark or space parity generation
  - Parity, framing and overrun error detection
  - Automatic Echo, Local loopback and Remote Loopback Channel modes
  - Digital filter on receive line
  - Interrupt generation
  - Support for two DMA channels with connection to receiver and transmitter
  - Receiver timeout
  - Register write protection

Downloaded from Arrow.com.

21.3 Block Diagram

Figure 21-1. DBGU Block Diagram

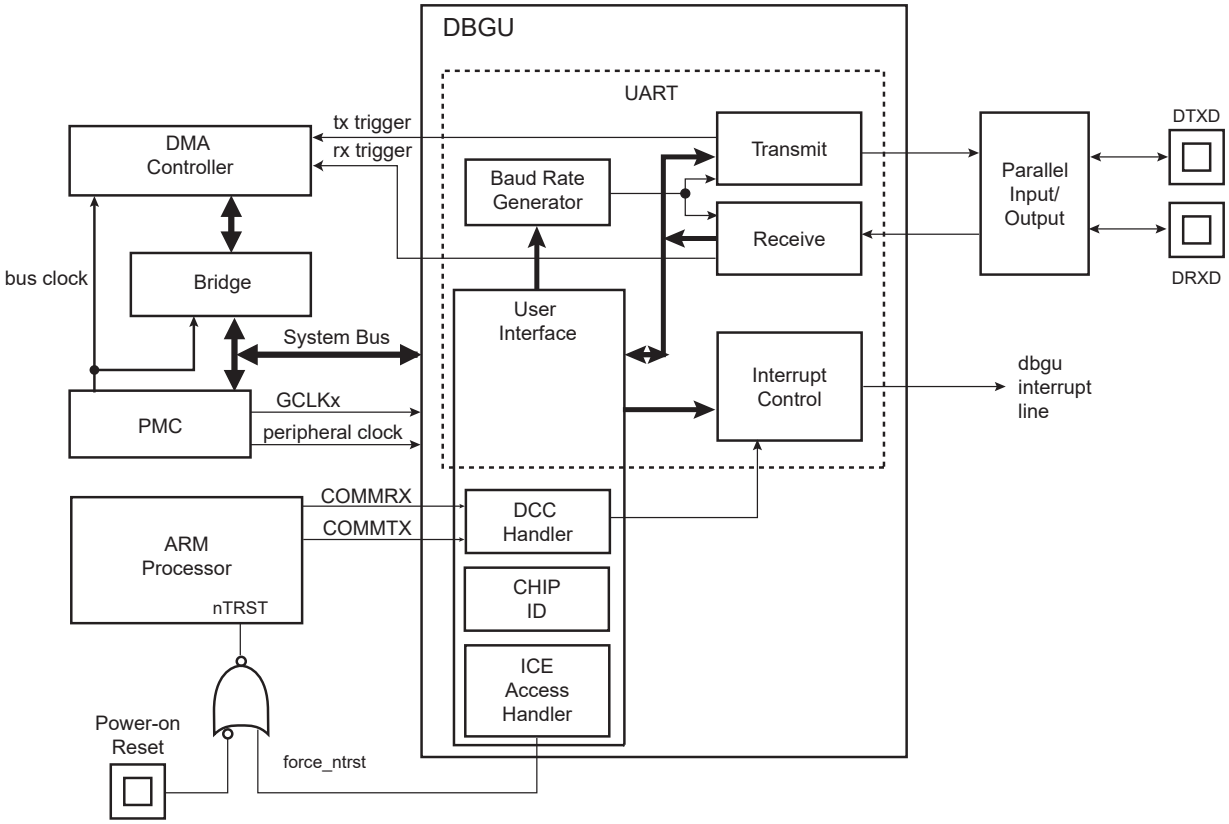


Table 21-1. DBGU Pin Description

Pin Name	Description	Type
DRXD	DBGU Receive Data	Input
DTXD	DBGU Transmit Data	Output

21.4 Product Dependencies

21.4.1 I/O Lines

The DBGU pins are multiplexed with PIO lines. The user must first configure the corresponding PIO Controller to enable I/O line operations of the DBGU.

21.4.2 Power Management

The DBGU clock can be controlled through the Power Management Controller (PMC). In this case, the user must first configure the PMC to enable the DBGU clock.

21.4.3 Interrupt Sources

The DBGU interrupt line is connected to one of the interrupt sources of the Interrupt Controller. Interrupt handling requires programming of the Interrupt Controller before configuring the DBGU.

Downloaded from Arrow.com.

21.5 Functional Description

The DBGU operates in Asynchronous mode only and supports only 8-bit character handling (with parity). It has no clock pin.

The DBGU is made up of a receiver and a transmitter that operate independently, and a common baud rate generator. Receiver timeout and transmitter time guard are not implemented. However, all the implemented features are compatible with those of a standard USART.

21.5.1 Baud Rate Generator

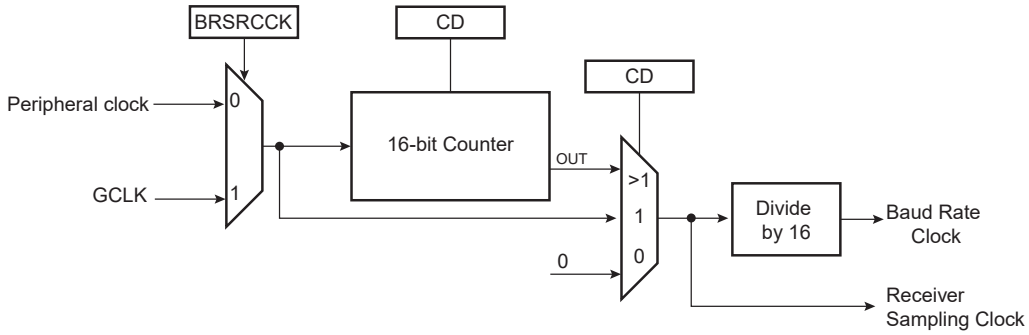
The baud rate generator provides the bit period clock named baud rate clock to both the receiver and the transmitter.

The baud rate clock is the peripheral clock divided by 16 times the clock divisor (CD) value written in the Baud Rate Generator register (DBGU_BRGR). If DBGU_BRGR is set to 0, the baud rate clock is disabled and the DBGU remains inactive. The maximum allowable baud rate is peripheral clock or GCLK divided by 16. The minimum allowable baud rate is peripheral clock divided by (16 x 65536). The clock source driving the baud rate generator (peripheral clock or GCLK) can be selected by writing the bit BRSRCCK in DBGU_MR.

If GCLK is selected, the baud rate is independent of the processor/bus clock. Thus the processor clock can be changed while DBGU is enabled. The processor clock frequency changes must be performed only by programming the field PRES in PMC_MCKR (see PMC section). Other methods to modify the processor/bus clock frequency (PLL multiplier, etc.) are forbidden when DBGU is enabled.

The peripheral clock frequency must be at least three times higher than GCLK.

Figure 21-2. Baud Rate Generator



21.5.2 Receiver

21.5.2.1 Receiver Reset, Enable and Disable

After device reset, the DBGU receiver is disabled and must be enabled before being used. The receiver can be enabled by writing the Control Register (DBGU_CR) with the bit RXEN at 1. At this command, the receiver starts looking for a start bit.

The programmer can disable the receiver by writing DBGU_CR with the bit RXDIS at 1. If the receiver is waiting for a start bit, it is immediately stopped. However, if the receiver has already detected a start bit and is receiving the data, it waits for the stop bit before actually stopping its operation.

The receiver can be put in reset state by writing DBGU_CR with the bit RSTRX at 1. In this case, the receiver immediately stops its current operations and is disabled, whatever its current state. If RSTRX is applied when data is being processed, this data is lost.

21.5.2.2 Start Detection and Data Sampling

The DBGU only supports asynchronous operations, and this affects only its receiver. The DBGU receiver detects the start of a received character by sampling the DRXD signal until it detects a valid start bit. A low level (space) on DRXD is interpreted as a valid start bit if it is detected for more than seven cycles of the sampling clock, which is 16 times the baud rate. Hence, a space that is longer than 7/16 of the bit period is detected as a valid start bit. A space which is 7/16 of a bit period or shorter is ignored and the receiver continues to wait for a valid start bit.

Downloaded from Arrow.com.

When a valid start bit has been detected, the receiver samples the DRXD at the theoretical midpoint of each bit. It is assumed that each bit lasts 16 cycles of the sampling clock (1-bit period) so the bit sampling point is eight cycles (0.5-bit period) after the start of the bit. The first sampling point is therefore 24 cycles (1.5-bit periods) after detecting the falling edge of the start bit.

Each subsequent bit is sampled 16 cycles (1-bit period) after the previous one.

Figure 21-3. Start Bit Detection

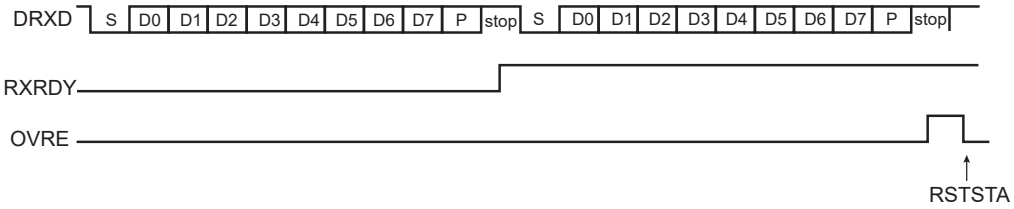
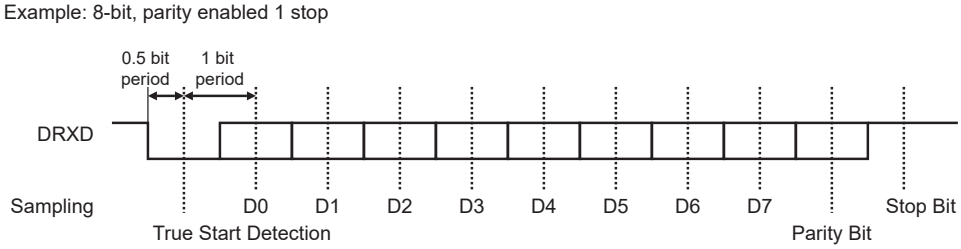


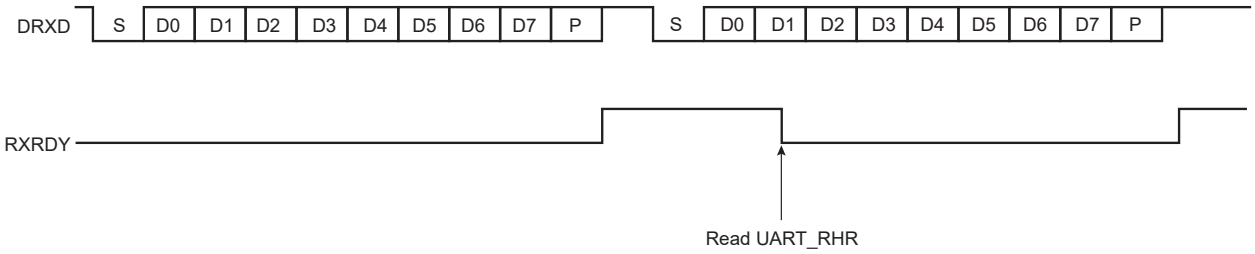
Figure 21-4. Character Reception



21.5.2.3 Receiver Ready

When a complete character is received, it is transferred to the Receive Holding Register (DBGU_RHR) and the RXRDY status bit in the Status Register (DBGU_SR) is set. The bit RXRDY is automatically cleared when DBGU_RHR is read.

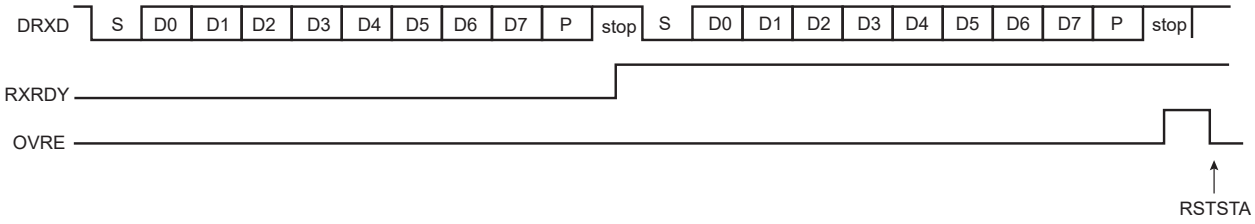
Figure 21-5. Receiver Ready



21.5.2.4 Receiver Overrun

The OVRE status bit in DBGU_SR is set if DBGU_RHR has not been read by the software (or the DMA Controller) since the last transfer, the RXRDY bit is still set and a new character is received. OVRE is cleared when the software writes a 1 to the bit RSTSTA (Reset Status) in DBGU_CR.

Figure 21-6. Receiver Overrun



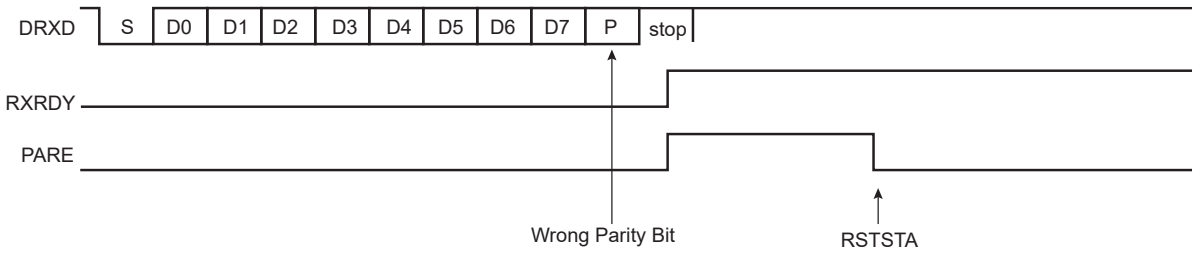
21.5.2.5 Parity Error

Each time a character is received, the receiver calculates the parity of the received data bits, in accordance with the field PAR in the Mode Register (DBGU_MR). It then compares the result with the received parity bit. If different, the parity error bit PARE in DBGU_SR is set at the same time RXRDY is set. The parity bit is cleared when DBGU_CR is

Downloaded from Arrow.com.

written with the bit RSTSTA (Reset Status) at 1. If a new character is received before the reset status command is written, the PARE bit remains at 1.

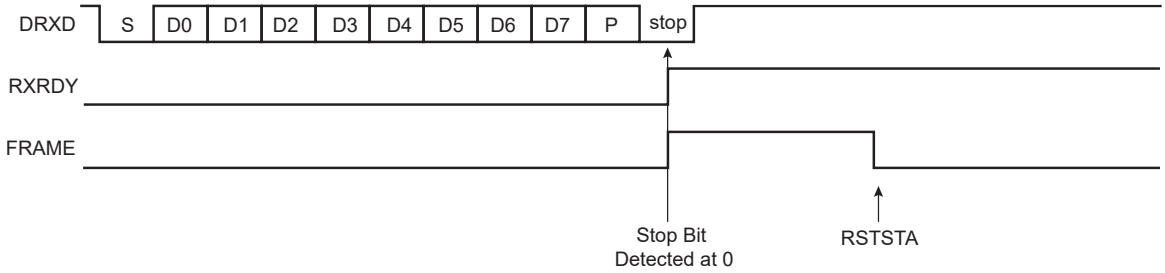
Figure 21-7. Parity Error



21.5.2.6 Receiver Framing Error

When a start bit is detected, it generates a character reception when all the data bits have been sampled. The stop bit is also sampled and when it is detected at 0, the FRAME (Framing Error) bit in DBGU_SR is set at the same time the RXRDY bit is set. The FRAME bit remains high until the Control Register (DBGU_CR) is written with the bit RSTSTA at 1.

Figure 21-8. Receiver Framing Error



21.5.2.7 Receiver Digital Filter

The DBGU embeds a digital filter on the receive line. It is disabled by default and can be enabled by writing a logical 1 in the FILTER bit of DBGU_MR. When enabled, the receive line is sampled using the 16x bit clock and a three-sample filter (majority 2 over 3) determines the value of the line.

21.5.2.8 Receiver Timeout

The Receiver Timeout provides support in handling variable-length frames. This feature detects an idle condition on the DRXD line. When a timeout is detected, the bit TIMEOUT in DBGU_SR rises and can generate an interrupt, thus indicating to the driver an end of frame.

The timeout delay period (during which the receiver waits for a new character) is programmed in the TO field of the Receiver Timeout register (DBGU_RTOR). If the TO field is written to 0, the Receiver Timeout is disabled and no timeout is detected. The TIMEOUT bit remains at 0. Otherwise, the receiver loads an 8-bit counter with the value programmed in TO. This counter is decremented at each bit period and reloaded each time a new character is received. If the counter reaches 0, the TIMEOUT bit rises. Then, the user can either:

- stop the counter clock until a new character is received. This is performed by writing a one to the STTTO (Start timeout) bit in DBGU_CR. In this case, the idle state on DRXD before a new character is received does not provide a timeout. This prevents having to handle an interrupt before a character is received and allows waiting for the next idle state on DRXD after a frame is received, or
- obtain an interrupt while no character is received. This is performed by writing a one to the RETTO (Reload and start timeout) bit in DBGU_CR. If RETTO is performed, the counter starts counting down immediately from the TO value. This enables generation of a periodic interrupt so that a user timeout can be handled, for example when no key is pressed on a keyboard.

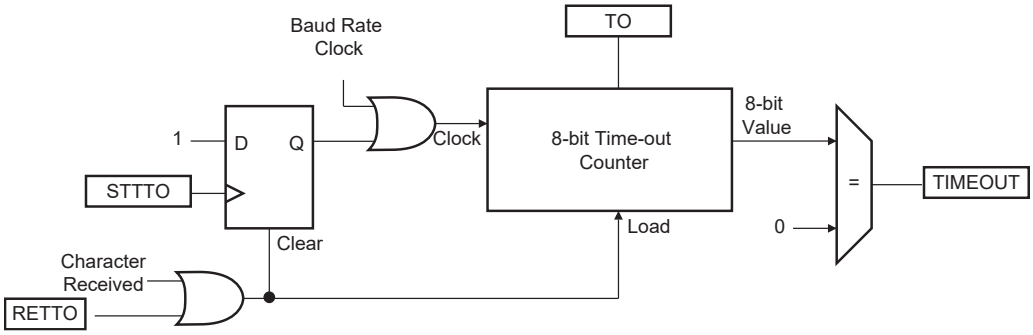
If STTTO is performed, the counter clock is stopped until a first character is received. The idle state on DRXD before the start of the frame does not provide a timeout. This prevents having to obtain a periodic interrupt and enables a wait of the end of frame when the idle state on DRXD is detected.

If RETTO is performed, the counter starts counting down immediately from the TO value. This enables generation of a periodic interrupt so that a user timeout can be handled, for example when no key is pressed on a keyboard.

Downloaded from Arrow.com.

The following figure shows the block diagram of the Receiver Timeout feature.

Figure 21-9. Receiver Timeout Block Diagram



The following table gives the maximum timeout period for some standard baud rates.

Table 21-2. Maximum Timeout Period

Baud Rate (bit/s)	Bit Time (µs)	Timeout (µs)
600	1,667	425,085
1,200	833	212,415
2,400	417	106,335
4,800	208	53,040
9,600	104	26,520
14,400	69	17,595
19,200	52	13,260
28,800	35	8,925
38,400	26	6,630
56,000	18	4,590
57,600	17	4,335
200,000	5	1,275

21.5.3 Transmitter

21.5.3.1 Transmitter Reset, Enable and Disable

After device reset, the DBGU transmitter is disabled and must be enabled before being used. The transmitter is enabled by writing DBGU_CR with the bit TXEN at 1. From this command, the transmitter waits for a character to be written in the Transmit Holding Register (DBGU_THR) before actually starting the transmission.

The programmer can disable the transmitter by writing DBGU_CR with the bit TXDIS at 1. If the transmitter is not operating, it is immediately stopped. However, if a character is being processed into the internal shift register and/or a character has been written in the DBGU_THR, the characters are completed before the transmitter is actually stopped.

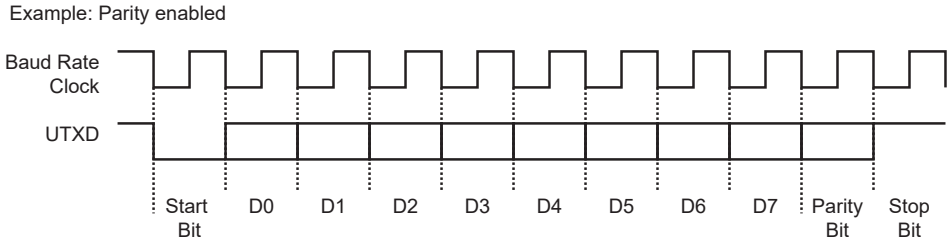
The programmer can also put the transmitter in its reset state by writing the DBGU_CR with the bit RSTTX at 1. This immediately stops the transmitter, whether or not it is processing characters.

21.5.3.2 Transmit Format

The DBGU transmitter drives the pin DTXD at the baud rate clock speed. The line is driven depending on the format defined in DBGU_MR and the data stored in the internal shift register. One start bit at level 0, then the 8 data bits, from the lowest to the highest bit, one optional parity bit and one stop bit at 1 are consecutively shifted out as shown in the following figure. The field PARE in DBGU_MR defines whether or not a parity bit is shifted out. When a parity bit is enabled, it can be selected between an odd parity, an even parity, or a fixed space or mark bit.

Downloaded from Arrow.com.

Figure 21-10. Character Transmission

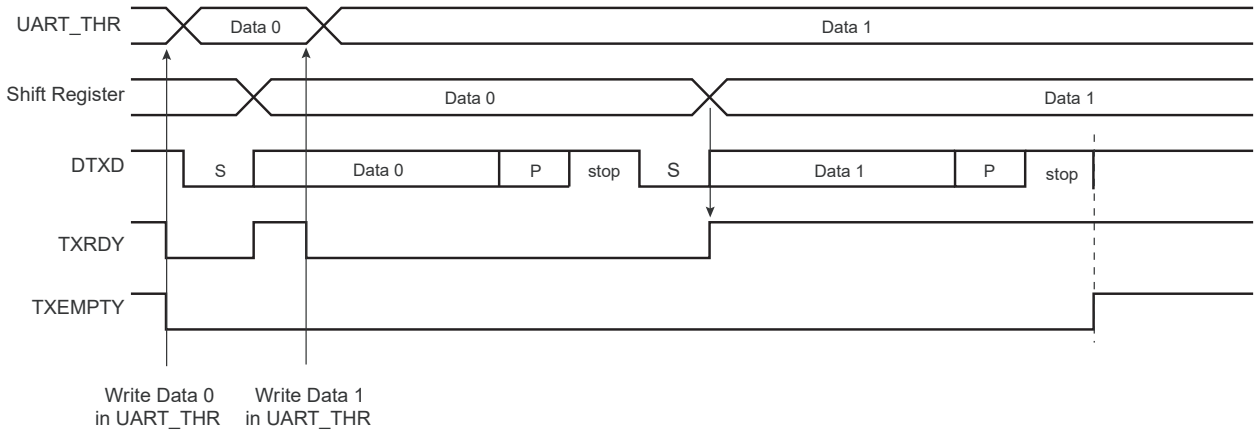


21.5.3.3 Transmitter Control

When the transmitter is enabled, the bit TXRDY (Transmitter Ready) is set in DBGU_SR. The transmission starts when the programmer writes in the DBGU_THR, and after the written character is transferred from DBGU_THR to the internal shift register. The TXRDY bit remains high until a second character is written in DBGU_THR. As soon as the first character is completed, the last character written in DBGU_THR is transferred into the internal shift register and TXRDY rises again, showing that the holding register is empty.

When both the internal shift register and DBGU_THR are empty, i.e., all the characters written in DBGU_THR have been processed, the TXEMPTY bit rises after the last stop bit has been completed.

Figure 21-11. Transmitter Control



21.5.4 DMA Support

Both the receiver and the transmitter of the DBGU are connected to a DMA Controller (DMAC) channel.

The DMA Controller channels are programmed via registers that are mapped within the DMAC user interface.

21.5.5 Register Write Protection

To prevent any single software error from corrupting DBGU behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the DBGU Write Protection Mode Register.

The following registers can be write-protected:

- DBGU Mode Register
- DBGU Baud Rate Generator Register
- DBGU Receiver Timeout Register
- Debug Unit Force NTRST Register

21.5.6 Test Modes

The DBGU supports three test modes. These modes of operation are programmed by using the CHMODE field in DBGU_MR.

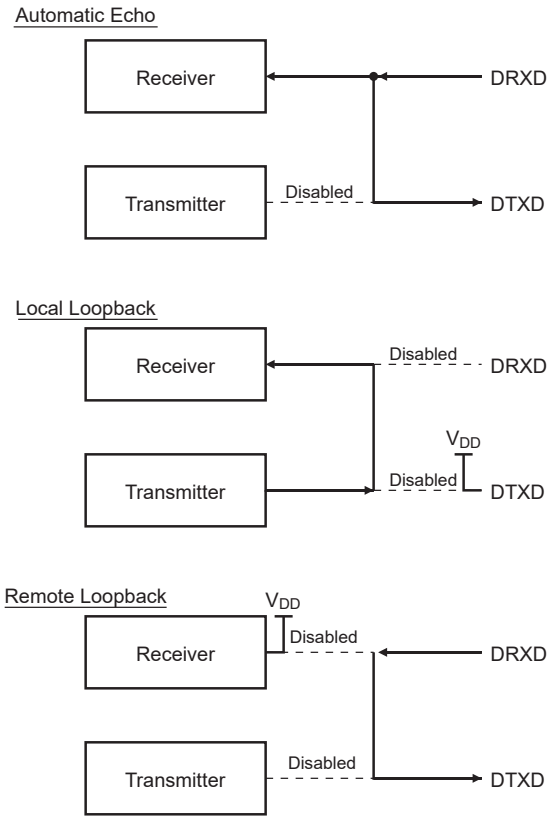
The Automatic Echo mode allows a bit-by-bit retransmission. When a bit is received on the DRXD line, it is sent to the DTXD line. The transmitter operates normally, but has no effect on the DTXD line.

Downloaded from Arrow.com.

The Local loopback mode allows the transmitted characters to be received. DTXD and DRXD pins are not used and the output of the transmitter is internally connected to the input of the receiver. The DRXD pin level has no effect and the DTXD line is held high, as in idle state.

The Remote loopback mode directly connects the DRXD pin to the DTXD line. The transmitter and the receiver are disabled and have no effect. This mode allows a bit-by-bit retransmission.

Figure 21-12. Test Modes



21.5.7 Debug Communication Channel Support

The DBGU handles the COMMRX and COMMTX signals that come from the Debug Communication Channel of the ARM processor and are driven by the In-circuit Emulator.

The Debug Communication Channel contains two registers that are accessible through the ICE Breaker on the JTAG side and through the coprocessor 0 on the ARM processor side.

As a reminder, the following instructions are used to read and write the Debug Communication Channel:

```
MRC p14, 0, Rd, c1, c0, 0
```

Returns the debug communication data read register into Rd

```
MCR p14, 0, Rd, c1, c0, 0
```

Writes the value in Rd to the debug communication data write register.

The COMMRX and COMMTX bits which indicate, respectively, that the read register has been written by the debugger but not yet read by the processor, and that the write register has been written by the processor and not yet read by the debugger, are wired on the two highest bits of DBGU_SR. These bits can generate an interrupt. This feature can be used to handle under interrupt a debug link between a debug monitor running on the target system and a debugger.

Downloaded from Arrow.com.

21.5.8 Chip Identifier

The DBGU features two chip identifier registers, DBGU Chip ID Register (DBGU_CIDR) and DBGU Extension ID Register (DBGU_EXID). Both registers contain a hard-wired value that is read-only.

21.5.9 ICE Access Prevention

The DBGU allows blockage of access to the system through the ARM processor's ICE interface. This feature is implemented via the Force NTRST Register (DBGU_FNR), that allows assertion of the NTRST signal of the ICE Interface. Writing the bit FNTRST (Force NTRST) to 1 in this register prevents any activity on the TAP controller.

On standard devices, the FNTRST bit resets to 0 and thus does not prevent ICE access.

This feature is especially useful on custom ROM devices for customers who do not want their on-chip code to be visible.

Downloaded from Arrow.com.

21.6 Register Summary

Offset	Name	Bit Pos.								
0x00	DBGU_CR	31:24								
		23:16								
		15:8					STTTO	RETTO		RSTSTA
		7:0	TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX		
0x04	DBGU_MR	31:24								
		23:16								
		15:8	CHMODE[1:0]			BRSRCCK	PAR[2:0]			
		7:0				FILTER				
0x08	DBGU_IER	31:24	COMMRX	COMMTX						
		23:16								
		15:8							TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE				TXRDY	RXRDY
0x0C	DBGU_IDR	31:24	COMMRX	COMMTX						
		23:16								
		15:8							TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE				TXRDY	RXRDY
0x10	DBGU_IMR	31:24	COMMRX	COMMTX						
		23:16								
		15:8							TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE				TXRDY	RXRDY
0x14	DBGU_SR	31:24	COMMRX	COMMTX						
		23:16								
		15:8							TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE				TXRDY	RXRDY
0x18	DBGU_RHR	31:24								
		23:16								
		15:8								
		7:0	RXCHR[7:0]							
0x1C	DBGU_THR	31:24								
		23:16								
		15:8								
		7:0	TXCHR[7:0]							
0x20	DBGU_BRGR	31:24								
		23:16								
		15:8	CD[15:8]							
		7:0	CD[7:0]							
0x24 ... 0x27	Reserved									
0x28	DBGU_RTOR	31:24								
		23:16								
		15:8								
		7:0	TO[7:0]							
0x2C ... 0x3F	Reserved									
0x40	DBGU_CIDR	31:24	EXT	CHID[30:24]						
		23:16	CHID[23:16]							
		15:8	CHID[15:8]							
		7:0	CHID[7:0]							
0x44	DBGU_EXID	31:24	EXID[31:24]							
		23:16	EXID[23:16]							
		15:8	EXID[15:8]							
		7:0	EXID[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x48	DBGU_FNR	31:24								
		23:16								
		15:8								
		7:0								FNTRST
0x4C ... 0xE3	Reserved									
0xE4	DBGU_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN

Downloaded from Arrow.com.

21.6.1 DBGU Control Register

Name: DBGU_CR
Offset: 0x0000
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					STTTO	RETTO		RSTSTA
Access					W	W		W
Reset					–	–		–

Bit	7	6	5	4	3	2	1	0
	TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX		
Access	W	W	W	W	W	W		
Reset	–	–	–	–	–	–		

Bit 11 – STTTO Start Timeout

Value	Description
0	No effect.
1	Starts waiting for a character before clocking the timeout counter. Resets status bit DBGU_SR.TIMEOUT.

Bit 10 – RETTO Rearm Timeout

Value	Description
0	No effect.
1	Restarts timeout.

Bit 8 – RSTSTA Reset Status

Value	Description
0	No effect.
1	Resets the status bits PARE, FRAME and OVRE in DBGU_SR.

Bit 7 – TXDIS Transmitter Disable

Value	Description
0	No effect.
1	The transmitter is disabled. If a character is being processed and a character has been written in DBGU_THR and RSTTX is not set, both characters are completed before the transmitter is stopped.

Bit 6 – TXEN Transmitter Enable

Value	Description
0	No effect.
1	The transmitter is enabled if TXDIS is 0.

Bit 5 – RXDIS Receiver Disable

Downloaded from Arrow.com.

Value	Description
0	No effect.
1	The receiver is disabled. If a character is being processed and RSTRX is not set, the character is completed before the receiver is stopped.

Bit 4 – RXEN Receiver Enable

Value	Description
0	No effect.
1	The receiver is enabled if RXDIS is 0.

Bit 3 – RSTTX Reset Transmitter

Value	Description
0	No effect.
1	The transmitter logic is reset and disabled. If a character is being transmitted, the transmission is aborted.

Bit 2 – RSTRX Reset Receiver

Value	Description
0	No effect.
1	The receiver logic is reset and disabled. If a character is being received, the reception is aborted.

Downloaded from Arrow.com.

21.6.2 DBGU Mode Register

Name: DBGU_MR
Offset: 0x0004
Reset: 0x0000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CHMODE[1:0]			BRSRCCK	PAR[2:0]			
Access	R/W	R/W		R/W	R/W	R/W	R/W	
Reset	0	0		0	0	0	0	

Bit	7	6	5	4	3	2	1	0
				FILTER				
Access				R/W				
Reset				0				

Bits 15:14 – CHMODE[1:0] Channel Mode

Value	Name	Description
0	NORMAL	Normal mode
1	AUTOMATIC	Automatic echo
2	LOCAL_LOOPBACK	Local loopback
3	REMOTE_LOOPBACK	Remote loopback

Bit 12 – BRSRCCK Baud Rate Source Clock
0 (PERIPH_CLK): The baud rate is driven by the peripheral clock
1 (GCLK): The baud rate is driven by a PMC-programmable clock GCLK (see section Power Management Controller (PMC)).

Bits 11:9 – PAR[2:0] Parity Type

Value	Name	Description
0	EVEN	Even Parity
1	ODD	Odd Parity
2	SPACE	Space: parity forced to 0
3	MARK	Mark: parity forced to 1
4	NO	No parity

Bit 4 – FILTER Receiver Digital Filter
0 (DISABLED): DBGU does not filter the receive line.
1 (ENABLED): DBGU filters the receive line using a three-sample filter (16x-bit clock) (2 over 3 majority).

Downloaded from Arrow.com.

21.6.3 DBGU Interrupt Enable Register

Name: DBGU_IER
Offset: 0x0008
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	COMMRX	COMMTX						
Access	W	W						
Reset	–	–						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							TXEMPTY	TIMEOUT
Access							W	W
Reset							–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE				TXRDY	RXRDY
Access	W	W	W				W	W
Reset	–	–	–				–	–

Bit 31 – COMMRX Enable COMMRX (from ARM) Interrupt

Bit 30 – COMMTX Enable COMMTX (from ARM) Interrupt

Bit 9 – TXEMPTY Enable TXEMPTY Interrupt

Bit 8 – TIMEOUT Enable Timeout Interrupt

Bit 7 – PARE Enable Parity Error Interrupt

Bit 6 – FRAME Enable Framing Error Interrupt

Bit 5 – OVRE Enable Overrun Error Interrupt

Bit 1 – TXRDY Enable TXRDY Interrupt

Bit 0 – RXRDY Enable RXRDY Interrupt

Downloaded from Arrow.com.

21.6.4 DBGU Interrupt Disable Register

Name: DBGU_IDR
Offset: 0x000C
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	COMMRX	COMMTX						
Access	W	W						
Reset	–	–						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							TXEMPTY	TIMEOUT
Access							W	W
Reset							–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE				TXRDY	RXRDY
Access	W	W	W				W	W
Reset	–	–	–				–	–

Bit 31 – COMMRX Disable COMMRX (from ARM) Interrupt

Bit 30 – COMMTX Disable COMMTX (from ARM) Interrupt

Bit 9 – TXEMPTY Disable TXEMPTY Interrupt

Bit 8 – TIMEOUT Disable Timeout Interrupt

Bit 7 – PARE Disable Parity Error Interrupt

Bit 6 – FRAME Disable Framing Error Interrupt

Bit 5 – OVRE Disable Overrun Error Interrupt

Bit 1 – TXRDY Disable TXRDY Interrupt

Bit 0 – RXRDY Disable RXRDY Interrupt

Downloaded from Arrow.com.

21.6.5 DBGU Interrupt Mask Register

Name: DBGU_IMR
Offset: 0x0010
Reset: 0x0000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
	COMMRX	COMMTX						
Access	R	R						
Reset	0	0						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							TXEMPTY	TIMEOUT
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE				TXRDY	RXRDY
Access	R	R	R				R	R
Reset	0	0	0				0	0

Bit 31 – COMMRX Mask COMMRX (from ARM) Interrupt

Bit 30 – COMMTX Mask COMMTX (from ARM) Interrupt

Bit 9 – TXEMPTY Mask TXEMPTY Interrupt

Bit 8 – TIMEOUT Mask Timeout Interrupt

Bit 7 – PARE Mask Parity Error Interrupt

Bit 6 – FRAME Mask Framing Error Interrupt

Bit 5 – OVRE Mask Overrun Error Interrupt

Bit 1 – TXRDY Disable TXRDY Interrupt

Bit 0 – RXRDY Mask RXRDY Interrupt

Downloaded from Arrow.com.

21.6.6 DBGU Status Register

Name: DBGU_SR
Offset: 0x0014
Reset: –
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	COMMRX	COMMTX						
Access	R	R						
Reset	–	–						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							TXEMPTY	TIMEOUT
Access							R	R
Reset							–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE				TXRDY	RXRDY
Access	R	R	R				R	R
Reset	–	–	–				–	–

Bit 31 – COMMRX Debug Communication Channel Read Status

Value	Description
0	COMMRX from the ARM processor is inactive.
1	COMMRX from the ARM processor is active.

Bit 30 – COMMTX Debug Communication Channel Write Status

Value	Description
0	COMMTX from the ARM processor is inactive.
1	COMMTX from the ARM processor is active.

Bit 9 – TXEMPTY Transmitter Empty

Value	Description
0	There are characters in DBGU_THR, or characters are being processed by the transmitter, or the transmitter is disabled.
1	There are no characters in DBGU_THR and there are no characters being processed by the transmitter.

Bit 8 – TIMEOUT Receiver Timeout

Value	Description
0	There has not been any timeout since the last Start timeout command (DBGU_CR.STTTO), or the Timeout register is 0.
1	There has been a timeout since the last Start timeout command (DBGU_CR.STTTO).

Bit 7 – PARE Parity Error

Value	Description
0	No parity error has occurred since the last RSTSTA.
1	At least one parity error has occurred since the last RSTSTA.

Bit 6 – FRAME Framing Error

Downloaded from Arrow.com.

Value	Description
0	No framing error has occurred since the last RSTSTA.
1	At least one framing error has occurred since the last RSTSTA.

Bit 5 – OVRE Overrun Error

Value	Description
0	No overrun error has occurred since the last RSTSTA.
1	At least one overrun error has occurred since the last RSTSTA.

Bit 1 – TXRDY Transmitter Ready

Value	Description
0	A character has been written to DBGU_THR and not yet transferred to the internal shift register, or the transmitter is disabled.
1	There is no character written to DBGU_THR that is not yet transferred to the internal shift register.

Bit 0 – RXRDY Receiver Ready

Value	Description
0	No character has been received since the last read of DBGU_RHR, or the receiver is disabled.
1	At least one complete character has been received, transferred to DBGU_RHR, and not yet read.

Downloaded from Arrow.com.

21.6.7 DBGU Receiver Holding Register

Name: DBGU_RHR
Offset: 0x0018
Reset: 0x0000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RXCHR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – RXCHR[7:0] Received Character
Last received character if RXRDY is set.

Downloaded from Arrow.com.

21.6.8 DBGU Transmit Holding Register

Name: DBGU_THR
Offset: 0x001C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	TXCHR[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 7:0 – TXCHR[7:0] Character to be Transmitted
Next character to be transmitted after the current character if TXRDY is not set.

Downloaded from Arrow.com.

21.6.9 DBGU Baud Rate Generator Register

Name: DBGU_BRGR
Offset: 0x0020
Reset: 0x0000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CD[15:0] Clock Divisor

Value	Description
0	Baud rate clock is disabled.
1	1 to 65,535: If BRSRCCK = 0: $CD = \frac{f_{peripheral\ clock}}{16 \times Baud\ Rate}$ If BRSRCCK = 1: $CD = \frac{f_{GCLKx}}{16 \times Baud\ Rate}$

Downloaded from Arrow.com.

21.6.10 DBGU Receiver Timeout Register

Name: DBGU_RTOR
Offset: 0x0028
Reset: 0x0000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	TO[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – TO[7:0] Timeout Value

Value	Description
0	The receiver timeout is disabled.
1–255	The receiver timeout is enabled and the timeout delay is TO x bit period.

Downloaded from Arrow.com.

21.6.11 DBGU Chip ID Register

Name: DBGU_CIDR
Offset: 0x0040
Reset: 0x819B35A1
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	EXT	CHID[30:24]						
Access	R	R	R	R	R	R	R	R
Reset								

Bit	23	22	21	20	19	18	17	16
	CHID[23:16]							
Access	R	R	R	R	R	R	R	R
Reset								

Bit	15	14	13	12	11	10	9	8
	CHID[15:8]							
Access	R	R	R	R	R	R	R	R
Reset								

Bit	7	6	5	4	3	2	1	0
	CHID[7:0]							
Access	R	R	R	R	R	R	R	R
Reset								

Bit 31 – EXT Extension Flag

Value	Description
0	Chip ID has a single register definition without extension.
1	An extended Chip ID exists.

Bits 30:0 – CHID[30:0] Chip ID Value

Downloaded from Arrow.com.

21.6.12 DBGU Chip ID Extension Register

Name: DBGU_EXID
Offset: 0x0044
Reset: –
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	EXID[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	EXID[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	EXID[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	EXID[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – EXID[31:0] Chip ID Extension
Read as 0 if the bit EXT in DBGU_CIDR is 0.

Downloaded from Arrow.com.

21.6.13 Debug Unit Force NTRST Register

Name: DBGU_FNR
Offset: 0x0048
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								FNTRST
Access								R/W
Reset								0

Bit 0 – FNTRST Force NTRST

Value	Description
0	NTRST of the ARM processor's TAP controller is driven by the power_on_reset signal.
1	NTRST of the ARM processor's TAP controller is held low.

Downloaded from Arrow.com.

21.6.14 DBGU Write Protection Mode Register

Name: DBGU_WPMR
Offset: 0x00E4
Reset: 0x0000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x554152	PASSWD	Writing any other value in this field aborts the write operation. Always reads as 0.

Bit 0 – WPEN Write Protection Enable

See Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x554152 (DBGU in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x554152 (DBGU in ASCII).

Downloaded from Arrow.com.

22. OTP Memory Controller (OTPC)

22.1 Description

The OTP Memory Controller (OTPC) is the secure interface between the system and the OTP memory.

The default value of a memory bit is logic ‘0’ (not programmed). A programmed memory bit is logic ‘1’.

An OTP matrix is a type of non-volatile memory. Each bit in the matrix can be programmed only once. The bits are used to store data such as:

- calibration bits for analog cells (e.g. RC oscillators, etc.),
- hardware configuration settings (e.g. JTAG disable, etc.),
- chip identifiers,
- key data invisible by software,
- user data.

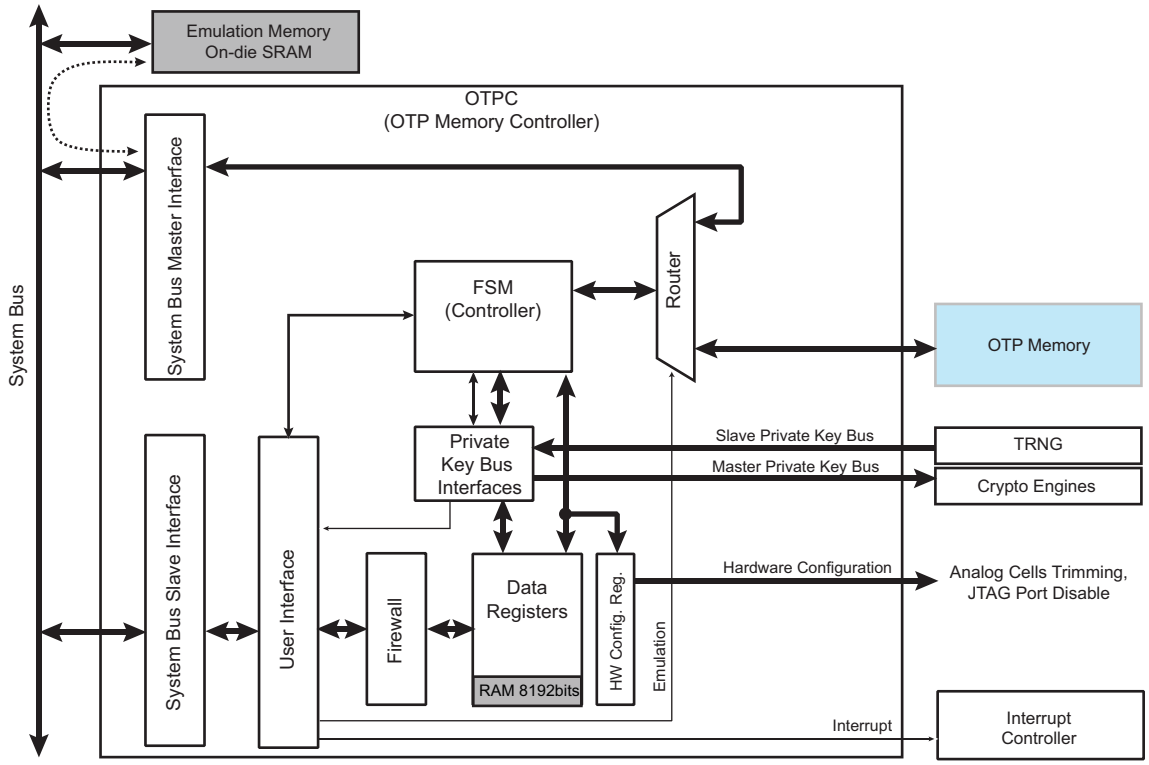
22.2 Embedded Characteristics

- Programs and Reads the Memory by Software
- Emulation Mode
- Automatic Check of Programmed Bits on Startup for Safe Operation
- User Area Organized by Packet for Flexibility on Size and Security:
  - Individual packet locking possibility (with checksum check)
  - Individual packet read access through only Private Key bus or System bus
  - Individual packet hiding (for packet with System bus access only)
  - Individual packet size of 32 bits to 8192 bits in 32-bit steps
  - Individual packet invalidation
- Firewall: Software/Hardware Protection Against Unexpected Read/Write

Downloaded from Arrow.com.

22.3 Block Diagram

Figure 22-1. OTPC Block Diagram



22.4 Functional Description

22.4.1 Bus Interfaces

The OTPC features four bus interfaces to access the OTP memory:

- Master System bus
- Slave System bus
- Master Key bus (Private Key bus)
- Slave Key bus (Private Key bus)

The Master System bus is used in Emulation mode to write and read data to/from the Emulation memory instead of the OTP memory. The Master System bus can only access the Emulation memory.

The Slave System bus is available to access to the user interface.

The Master Key bus is available to read keys stored in the User area of the OTP memory and transfer them to the slave crypto-engines connected to this bus (e.g. AES). No data accessible to the Master Key bus are accessible to the System bus.

The Slave Key bus is available to write some data to the User area of the OTP memory. No data coming from the Slave Key bus are accessible to the System bus.

22.4.2 OTP Memory Partitioning

The OTP memory is partitioned into different areas:

- Reserved area
- 11-Kbyte User area

Downloaded from Arrow.com.

The initial value of the OTP memory is ‘0’ but the memory may contain some “defective” bits already set to the value ‘1’.

The memory is organized into 32-bit words.

22.4.3 User Area

22.4.3.1 Area Configuration and Control

The User area is controlled and configured through the OTPC Control (OTPC_CR), OTPC Mode (OTPC_MR) and OTPC Data (OTPC_DR) registers.

22.4.3.2 Area Mapping

The entire User area space is mapped into 32-bit words. Each 32-bit word is part of one packet.

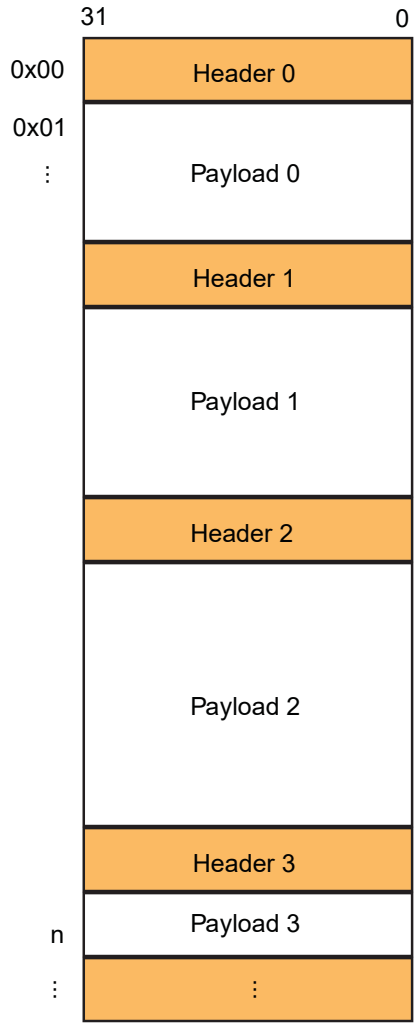
Each packet contains a 32-bit header and 32-bit words of payload (data). The number of payload words is defined in the header.

This mapping system allows three important functions for an OTP memory:

- Gives a flexible size for any type of data
- Improves the security by identifying and masking key areas to the System bus,
- Provides an easy way to invalidate a packet and create a replacement packet (key, data) while free space is available in OTP.

The packet organization into the User area is shown in the figure below.

Figure 22-2. User Area Memory Mapping



Downloaded from Arrow.com.

The number of packets depends on the size of the User area and the size of each packet.

22.4.3.3 Packet Definition

Each packet contains:

- One 32-bit header field
- One payload field containing at least 32 bits of data and up to 256 x 32 bits of data (8192 bits)

The payload field has no effect on the hardware except for “special” packets.

22.4.3.3.1 Header Field

The following table provides the definition of the header content.

31	30	29	28	27	26	25	24
CHECKSUM							
23	22	21	20	19	18	17	16
CHECKSUM							
15	14	13	12	11	10	9	8
SIZE							
7	6	5	4	3	2	1	0
ONE	–	INVLD		LOCK	PACKET		

- PACKET: Indicates the packet type. Six types are available:
  - REGULAR packet (value 1) accessible through the User Interface
  - KEY packet (value 2) accessible only through the Private Key buses
  - BOOT_CONFIGURATION “special” packet (value 3)
  - SECURE_BOOT_CONFIGURATION “special” packet (value 4)
  - HARDWARE_CONFIGURATION “special” packet (value 5)
  - CUSTOM “special” packet (value 6)
- LOCK: Written by the controller when the checksum is generated.
- INVLD: Written when an invalidation process is requested.
- ONE: Must be written to ‘1’.
- SIZE: Indicates the size in 32-bit words of the payload field. SIZE = 0 means payload is 32-bit size, SIZE = 255 means payload is 8192-bit size.
  The entire packet size (in bits) in OTP memory is 32 (header) + (32 * (SIZE + 1)).
- CHECKSUM: Represents the checksum of the packet excluding the CHECKSUM field of the header. The real value of the CHECKSUM field is not readable; it is generated by the OTPC.
  - When CHECKSUM is read as 0, the checksum has not been generated. It is possible to modify the packet content.
  - When CHECKSUM is read as 0x33CC, the checksum has not been generated but some bits are already at 1. Locking the packet may fail.
  - When CHECKSUM is read as 0xA5A5, the checksum has been generated and the last check was successful. It is impossible to modify the packet content.
  - When CHECKSUM is read as 0xCC33, the header of the packet is corrupted.
  - When CHECKSUM is read as 0xFFFF, the entire packet is no longer valid. It is possible to modify the packet content and it is up to the software to program the payload to a different value if needed.
  - For all other values of CHECKSUM, the checksum has been generated and the last check failed to match the checksum written in the OTP memory. It is impossible to modify the packet content.

22.4.3.3.2 “Special” Packets

The payload of “special” packets is interpreted by the OTPC and some actions can be triggered while the “special” packets are read. The address of the “special” packets inside the area does not matter.

Downloaded from Arrow.com.

If the checksum has been generated and does not match during the last read, the payload is not interpreted by the OTPC.

The table below provides the list of “special” packets.

Table 22-1. “Special” Packets

Name	SIZE	PACKET	Description
Boot Configuration	(see Note 1)	3	Boot configuration
Secure Boot Configuration	(see Note 2)	4	Secure Boot configuration
User Hardware Configuration	1	5	Hardware configuration
Custom	N/A	6	For user custom purposes. The size of this packet is user application dependent.

Note:

1. For details, refer to the section “Boot Strategies”.
2. For details, refer to “Secure Boot Strategy” document (Literature No. DS00003195), available under Non-Disclosure Agreement (NDA). Contact a Microchip Sales Representative for details.

The address of the Boot Configuration and Secure Boot Configuration special packets can be retrieved in the OTPC Boot Addresses (OTPC_BAR) register.

The address of the Custom special packets can be retrieved in the OTPC Custom Address (OTPC_CAR) register.

For each “special” packets, if there is more than one valid packet (of the same type), only the last packet will be considered (previous packets will be ignored). It is recommended to invalidate prior “special” packets to keep only one valid packet for each “special” packet type.

22.4.3.4 Init

After each reset, the OTPC parses the User area to check its content. The header of each packet will be read, depending on the header value some actions can be triggered:

- If the header is corrupted, the init sequence is interrupted and the OTPC_ISR.COERR bit is set.
- If the INVLD field of the header is 3, the OTPC ignores the packet and jumps to the next header.
- If the LOCK bit of the header is set, the payload is read and the checksum computed during the read is compared to the checksum saved in the header. If the checksums do not match, the OTPC_ISR.CKERR bit will be set.
- If the PACKET field of the header is BOOT_CONFIGURATION, the address of the packet will be stored in the OTPC_BAR.BCADDR field. Any previous value stored in the OTPC_BAR.BCADDR field is lost.
- If the PACKET field of the header is SECURE_BOOT_CONFIGURATION, the address of the packet will be stored in the OTPC_BAR.SBCADDR field. Any previous value stored in the OTPC_BAR.SBCADDR field is lost.
- If the PACKET field of the header is HARDWARE_CONFIGURATION, the payload is read and stored in the OTPC User Hardware Configuration (OTPC_UHCxR) registers (unless the checksums do not match if the packet is locked, in that case the reset value of the OTPC_UHCxR registers is stored). Any previous value stored in the OTPC_UHCxR registers is lost.
- If the PACKET field of the header is CUSTOM, the address of the packet will be stored in the OTPC_CAR.CADDR field. Any previous value stored in the OTPC_CAR.CADDR field is lost.

At the end of the init sequence, the value of the last HARDWARE_CONFIGURATION “special“ packet found is applied to the hardware.

22.4.3.5 Read Access

The User area can be read through the User interface at any time after a reset. Each packet is available through the OTPC_DR register. To trigger a packet read, follow the steps below:

1. The address of the header (or any address of the payload) must be written in OTPC_MR.ADDR.
2. OTPC_CR.READ must be set to ‘1’ (the OTPC_CR.KEY field value does not matter).
3. Wait for OTPC_ISR.EOR to be set to ‘1’ or for OTPC_SR.READ to be set to ‘0’, indicating that the whole packet has been transferred into temporary registers.

Downloaded from Arrow.com.

4. Read the header of the packet in the OTPC_HR register. To read each payload word, the address of the payload word must be written in OTPC_AR.DADDR. The payload word is then available in OTPC_DR. If OTPC_AR.INCRT is set to AFTER_READ, any read in the OTPC_DR increments the DADDR field.

   If INCRT is set to AFTER_WRITE, any write in the OTPC_DR increments the DADDR field.

The payload of a packet with PACKET set to KEY is read as '0'. The payload of a packet hidden since the last reset is read as '0'.

22.4.3.5.1 Transfer a Packet through the Master Key Bus

To transfer a packet from the OTP memory through the Master Key bus, follow the steps below:

1. The address of the header (or any address of the payload) must be written in OTPC_MR.ADDR. Only the packets with the packet type set to KEY are transferable on the Master Key bus.
2. The Key bus destination must be written in OTPC_MR.KBDST.
3. Write 0x7167in the OTPC_CR.KEY field and '1' to OTPC_CR.KBSTART.

The end of the transfer is indicated by OTPC_ISR.EOKT='1' and/or OTPC_SR.MKBB='0'.

If the type of the packet is not KEY, OTPC_ISR.KBERR is set.

To cancel a packet transfer, OTPC_CR.KEY must be set to 0x7167 and KBSTOP must be set.

22.4.3.5.2 Hiding a Packet

For security reasons, it is possible to hide a packet after having read it through the User Interface. Once hidden, any read to the payload of the packet returns '0'.

To unhide a packet, a reset is necessary.

Hiding a packet does not make it available through the Key Buses.

To hide a packet, follow the steps below:

1. Write the address value of the header of the packet to hide in OTPC_MR.ADDR.
2. Write 0x7167 in OTPC_CR.KEY and '1' to HIDE.

22.4.3.6 Write (Program) Considerations

Each word of the User area can be written only once. It is possible to write a packet payload partially and/or update a packet payload already written.

The packet to write (either the header or the payload) may contain some bits already at '1'. In this case, a dummy packet can be used to write the packet in a different location. The OTPC may also be able to fix the '1' already written if this bit also matches the packet to write. Thus before writing any new packet, it is necessary to proceed to a read at the last address.

22.4.3.6.1 '1' in the Header

After the read, the header may contain one or more '1's. If the '1's match the '1's to write in the header, the packet can be written.

If the '1's do not all match the '1's to write in the header, the packet cannot be written. It is mandatory to create (and then invalidate if necessary) a packet with a compatible header.

22.4.3.6.2 '1' in the Payload

If the payload is written and then updated later, the '1's already written must match the '1's to write.

If the payload is written only once (no update later), the '1's already written must all match either the '1' or the '0' to write (it cannot be a mix of '1's and '0's to write).

22.4.3.7 Write (Program) Access

The User area can be programmed at any time after a reset until OTPC_MR.WRDIS has been set.

22.4.3.7.1 Writing a New Packet from the User Interface

To write a new packet from the User Interface, follow the steps below:

1. Write OTPC_MR.NPCKT to '0' if it is set at '1'.
2. Write OTPC_MR.ADDR to its maximum value.

Downloaded from Arrow.com.

3. Write a ‘1’ to OTPC_CR.READ and wait for the read completion (OTPC_ISR.EOR=‘1’ when the read is completed).
4. Check there is no bit already set to ‘1’ in the OTPC_HR and OTPC_DR. The check for the data registers can be replaced by reading OTPC_SR.ONEF (if ONEF is set, at least one bit of the data registers is set to ‘1’). If the header or the payload of the packet already contains a 1, the new packet may need to be adapted.
5. Write OTPC_MR.ADDR to ‘0’ and set NPCKT.
   Depending on the contents of the temporary registers, an automatic flush can be triggered. If an automatic flush is started, OTPC_SR.FLUSH is set and OTPC_ISR.EOF is raised at the completion of the flush. It is mandatory to wait for the end of the flush.
6. Write the header value in OTPC_HR. The value of PACKET must not be the same as KEY.
7. Set DADDR to ‘0’.
8. Write the first data in the OTPC_DR register. To update the 32-bit data later (using the packet update), the OTPC_DR register must be set to 0.
9. Increment the DADDR field and write the next data in the OTPC_DR. Repeat this operation until all the data has been written.
   Skip the increment of DADDR if INCRT is set to AFTER_WRITE.
10. Write USER_KEY in the OTPC_CR.KEY field and ‘1’ to OTPC_CR.PGM.

Before the write operation in the OTP memory, the OTPC checks the consistency of the packet and that the packet does not overlap on any existing packet. In case of error, OTPC_ISR.WERR is set and the write operation is cancelled.

The end of the programming operation is indicated by OTPC_ISR.EOP=’1’ and/or OTPC_SR.PGM=’0’. At the end of the programming, the address of the header is available in OTPC_MR.ADDR and the OTPC_MR.NPCKT must be cleared.

The payload can be read back before programming. After read back, it is possible to update PACKET value to KEY before programming.

If the new written packet is the User Hardware Configuration special packet, its payload is ignored until the next reset or the next refresh. The OTPC_UHCxR registers will be updated after the reset or the refresh following programming.

22.4.3.7.2 Updating an Existing Packet from the User Interface

To update an existing packet from the User Interface, follow the steps below:

1. Write the address of the header of the packet to update in OTPC_MR.ADDR.
2. Start a read by setting OTPC_CR.READ and wait for the read completion indicated by OTPC_ISR.EOR.
3. Update the data using the OTPC_AR and OTPC_DR registers. Only the 32-bit data set to 0 can be updated, the non-zero 32-bit data must be left unchanged
4. Write 0x7167 in the OTPC_CR.KEY field and ‘1’ to OTPC_CR.PGM.

The end of the programming operation is indicated by OTPC_ISR.EOP='1' and/or OTPC_SR.PGM='0'.

If the updated packet is the User Hardware Configuration special packet, its new payload is ignored until the next reset. The OTPC_UHCxR registers will be updated after the reset or the refresh following programming.

22.4.3.7.3 Writing a Packet from the Slave Key Bus

To write a packet from the Slave Key bus interface (payload only), follow the steps below:

1. Write OTPC_MR.ADDR to ‘0’ and set NPCKT.
   Depending on the content of the temporary registers, an automatic flush can be triggered. If an automatic flush is started, OTPC_SR.FLUSH is set and OTPC_ISR.EOF is raised at the completion of the flush. It is mandatory to wait for the end of the flush.
2. Write the header value in the OTPC_HR register. The value of PACKET must be KEY.
3. Initiate a data transfer to the OTP memory through the TRNG Master Key bus.
4. Wait for the data transfer completion.
5. Check that no error happened during the key transfer (OTPC_ISR.KBERR must be cleared).
6. Write 0x7167 in the OTPC_CR.KEY field and ‘1’ to OTPC_CR.PGM.
7. Before the write operation in the OTP memory, the OTPC checks the consistency of the packet and that the packet does not overlap on any existing packet. In case of error, OTPC_ISR.WERR is set and the write

Downloaded from Arrow.com.

operation is cancelled. The end of the programming operation is indicated by OTPC_ISR.EOP=1 and/or OTPC_SR.PGM=0.

If the PACKET field is changed before programming, the payload is erased (and lost).

22.4.3.7.4 Locking a Packet

To lock a packet, follow the steps below:

1. Write the address value of the header of the packet to lock in OTPC_MR.ADDR.
2. Start a read by setting OTPC_CR.READ and waiting for the read completion indicated by OTPC_ISR.EOR.
3. Write 0x7167 in the OTPC_CR.KEY field and ‘1‘ in OTPC_CR.CKSGEN.

The end of the lock operation is indicated by OTPC_ISR.EOL=’1’ and/or OTPC_SR.LOCK=’0’.

Generating the checksum locks the packet and modification is no longer possible.

22.4.3.7.5 Invalidating a Packet

To invalidate a packet, follow the steps below:

1. Write the address value of the header of the packet to invalidate in OTPC_MR.ADDR.
2. Write 0x7167 in the OTPC_CR.KEY field and ‘1’ to OTPC_CR.INVLD.

The end of the invalidation operation is indicated by OTPC_ISR.EOI= ‘1’ and/or OTPC_SR.INVLD=’0’.

If the invalidated packet is the User Hardware Configuration special packet, its old payload remains active until the next reset or the next refresh. The OTPC_UHCxR registers will be updated after the reset or the refresh following the invalidation operation.

22.4.3.8 Fixing Corruption

During a programming sequence, packet header corruption may occur. This corruption can be caused by a partial programming of the header. It is mandatory to fix any corruption prior to any usage of the Engineering Area or User Area.

During the start sequence, the OTPC stops parsing the OTP memory at the first header corruption detected. When OTPC_ISR.COERR is set, a corruption has been detected. The corrupted header can be read in OTPC_HR and its location can be read in OTPC_MR.ADDR.

A header is corrupted if at least one of the following statements matches:

- The ONE bit is cleared (it must be set to fix the corruption).
- The INVLD field is 3 and the PACKET field is 0 (PACKET must be set to a non-0 value to fix the corruption).
- The SIZE and PACKET fields are not consistent (for PACKET set to PRODUCT_UID, HARDWARE_CONFIGURATION or SECURITY_CONFIGURATION, the packet must be invalidated to fix the corruption).

To fix a corruption, start a a read procedure at the location of the corrupted header. The OTPC reads the payload according to the size provided in the header and reads one extra word of payload, which should match the next header.

The corrupted header must be fixed by writing any missing ‘1’s or, if not possible, by extending its size if the supposed next header is 0, or by invalidating the packet.

A reset is required after fixing the corruption.

22.4.3.9 “Software” Protections

The User area can be protected against read accesses and/or modifications.

To enable read protection of the User data (OTPC_DR) and header (OTPC_HR) registers, OTPC_MR.RDDIS must be set. Clearing RDDIS allows read access again. When the OTPC_DR and OTPC_HR registers are read-protected, any read returns 0.

To enable write protection of the OTPC_DR registers, the WRDIS bit of OTPC_MR should be set. Clearing the WRDIS bit allows write access again.

To enable write protection of the User area, the write protection of the User data registers must be enabled.

The OTPC_MR can be locked until the next reset by setting the LOCK bit of OTPC_MR. Once locked, the current protection configuration of the OTPC_DR and OTPC_HR registers applies, it is then also impossible to update,

Downloaded from Arrow.com.

program, invalidate, hide or read a packet (the OTPC_MR.ADDR field is then locked too preventing to select a packet).

22.4.3.10 “Hardware” Protections

The User area can be protected against read accesses and/or modifications.

To enable the different protections of the User area, the User Configuration special packet must be programmed. The packet is described in the OTPC_UHCxR registers.

As an example, to disable the JTAG interface for an indefinite period, the JTAGDIS, UHCINVDIS and UHCPGDIS fields of the User Hardware Configuration special packet must all be programmed to a non-zero value. Thus, it will be impossible to update or invalidate the User Hardware Configuration special packet.

WARNING “Hardware” protections are in effect for an indefinite period and cannot be cancelled.

22.4.4 OTP Emulation Mode

The OTPC features an Emulation mode. This Emulation mode can be used to test all the operations allowed by the controller on a memory instead of the real OTP memory.

When the Emulation mode is enabled, the controller has the same behavior. The Emulation mode is enabled only when the OTP memory has not been previously programmed.

To enable/disable the Emulation mode on the User area, follow the steps below:

1. Set OTPC_MR.EMUL to ‘1’ (to enable) or to ‘0’ (to disable).
2. Refresh the User area by writing a ‘1’ to OTPC_CR.REFRESH and 0x7167 in OTPC_CR.KEY.
3. Wait for the refresh completion by polling OTPC_ISR.EORF.

The current running mode of the User area can be observed by reading OTPC_SR.EMUL. If EMUL is set to ‘1’, Emulation mode is enabled; if it is set to ‘0’, Emulation mode is disabled.

After a reset, Emulation mode is disabled.

22.4.5 Interrupts

An OTPC interrupt request can be triggered when one or several of the following bits are set in the OTPC Interrupt Status register (OTPC_ISR): End Of Programming (EOP), End Of Locking (EOL), End Of Invalidation (EOI), End Of Key Transfer (EOKT), Programming Error (PGERR), Locking Error (LKERR), Invalidation Error (IVERR), Write Error (WERR), End Of Read (EOR), End Of Flush (EOF), End Of Hide (EOH), End Of Refresh (EOF), Checksum Check Error (CKERR) or Key Invalid Error (KBERR).

The interrupt request is generated if the corresponding bit in the OTPC Interrupt Mask register (OTPC_IMR) is set. Bits in OTPC_IMR are set by writing a ‘1’ to the corresponding bit in the OTPC Interrupt Enable register (OTPC_IER) and cleared by writing a ‘1’ to the corresponding bit in the OTPC Interrupt Disable register (OTPC_IDR). The interrupt request remains active until the corresponding bit in OTPC_ISR is cleared.

Reading the OTPC_ISR clears all bits of the register.

22.4.6 Register Write Protection

To prevent any single software error from corrupting the OTPC behavior, certain registers in the address space can be write-protected by setting the Write Protection Configuration Enable (WPCFEN), Write Protection Interrupt Enable (WPITEN) and/or Write Protection Control Enable (WPCTEN) bit(s) in the Write Protection Mode Register (OTPC_WPMR).

If a write access to the protected registers is detected, the Write Protection Violation Status (WPVS) flag in the Write Protection Status Register (OTPC_WPSR) is set and the field Write Protection Violation Source (WPVSRC) indicates the register in which the write access has been attempted. An interrupt can be raised if the Security and/or Safety Event (SECE) interrupt is set in OTPC_IMR.

The WPVS flag is automatically reset by reading the OTPC_WPSR.

Downloaded from Arrow.com.

The following registers can be write-protected with the OTPC_WPMR.WPCFEN bit:

- OTPC Mode Register

The following registers can be write-protected with the OTPC_WPMR.WPITEN bit:

- OTPC Interrupt Enable Register
- OTPC Interrupt Disable Register

The following registers can be write-protected with the OTPC_WPMR.WPCTEN bit:

- OTPC Control Register

Downloaded from Arrow.com.

22.5 Register Summary

Offset	Name	Bit Pos.								
0x00	OTPC_CR	31:24	KEY[15:8]							
		23:16	KEY[7:0]							
		15:8	REFRESH						KBSTOP	KBSTART
		7:0	FLUSH	READ		HIDE		INVLD	CKSGEN	PGM
0x04	OTPC_MR	31:24	ADDR[15:8]							
		23:16	ADDR[7:0]							
		15:8	LOCK		KBDST[1:0]				WRDIS	RDDIS
		7:0	EMUL			NPCKT				UHCRRDIS
0x08	OTPC_AR	31:24								
		23:16								INCRT
		15:8								
		7:0	DADDR[7:0]							
0x0C	OTPC_SR	31:24								
		23:16								
		15:8							ONEF	HIDE
		7:0	FLUSH	READ	SKBB	MKBB	EMUL	INVLD	LOCK	PGM
0x10	OTPC_IER	31:24				SECE				
		23:16								KBERR
		15:8		HDERR	COERR	CKERR	EORF	EOH	EOF	EOR
		7:0	WERR	IVERR	LKERR	PGERR	EOKT	EOI	EOL	EOP
0x14	OTPC_IDR	31:24				SECE				
		23:16								KBERR
		15:8		HDERR	COERR	CKERR	EORF	EOH	EOF	EOR
		7:0	WERR	IVERR	LKERR	PGERR	EOKT	EOI	EOL	EOP
0x18	OTPC_IMR	31:24				SECE				
		23:16								KBERR
		15:8		HDERR	COERR	CKERR	EORF	EOH	EOF	EOR
		7:0	WERR	IVERR	LKERR	PGERR	EOKT	EOI	EOL	EOP
0x1C	OTPC_ISR	31:24				SECE				
		23:16								KBERR
		15:8		HDERR	COERR	CKERR	EORF	EOH	EOF	EOR
		7:0	WERR	IVERR	LKERR	PGERR	EOKT	EOI	EOL	EOP
0x20	OTPC_HR	31:24	CHECKSUM[15:8]							
		23:16	CHECKSUM[7:0]							
		15:8	SIZE[7:0]							
		7:0	ONE		INVLD[1:0]		LOCK	PACKET[2:0]		
0x24	OTPC_DR	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x28 ... 0x2F	Reserved									
0x30	OTPC_BAR	31:24	SBCADDR[15:8]							
		23:16	SBCADDR[7:0]							
		15:8	BCADDR[15:8]							
		7:0	BCADDR[7:0]							
0x34	OTPC_CAR	31:24								
		23:16								
		15:8	CADDR[15:8]							
		7:0	CADDR[7:0]							
0x38 ... 0x4F	Reserved									

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x50	OTPC_UHC0R	31:24								
		23:16								
		15:8								
		7:0	JTAGDIS[7:0]							
0x54	OTPC_UHC1R	31:24								
		23:16							URFDIS	CPGDIS
		15:8	CLKDIS	CINVDIS				SBCPGDIS	SBCLKDIS	SBCINVDIS
		7:0	BCPGDIS	BCLKDIS	BCINVDIS	UHCPGDIS	UHCLKDIS	UHCINVDIS	UPGDIS	URDDIS
0x58 ... 0x5F	Reserved									
0x60	OTPC_UID0R	31:24	UID[31:24]							
		23:16	UID[23:16]							
		15:8	UID[15:8]							
		7:0	UID[7:0]							
0x64	OTPC_UID1R	31:24	UID[31:24]							
		23:16	UID[23:16]							
		15:8	UID[15:8]							
		7:0	UID[7:0]							
0x68	OTPC_UID2R	31:24	UID[31:24]							
		23:16	UID[23:16]							
		15:8	UID[15:8]							
		7:0	UID[7:0]							
0x6C	OTPC_UID3R	31:24	UID[31:24]							
		23:16	UID[23:16]							
		15:8	UID[15:8]							
		7:0	UID[7:0]							
0x70 ... 0xE3	Reserved									
0xE4	OTPC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0				FIRSTE		WPCTEN	WPITEN	WPCFEN
0xE8	OTPC_WPSR	31:24	ECLASS				SWETYP[3:0]			
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS

Downloaded from Arrow.com.

22.5.1 OTPC Control Register

Name: OTPC_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCTEN bit is cleared in the OTPC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	KEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	REFRESH						KBSTOP	KBSTART
Access	W						W	W
Reset	–						–	–

Bit	7	6	5	4	3	2	1	0
	FLUSH	READ		HIDE		INVLD	CKSGEN	PGM
Access	W	W		W		W	W	W
Reset	–	–		–		–	–	–

Bits 31:16 – KEY[15:0] Programming Key
This field must be written with the correct key code (0x7167) to allow programming, checksum generation, packet invalidation or packet hiding.

Bit 15 – REFRESH Refresh the Area

Value	Description
0	No effect.
1	Starts a refresh of the area.

Bit 9 – KBSTOP Key Bus Transfer Stop

Value	Description
0	No effect.
1	Stops an on-going transfer on the Master Key bus.

Bit 8 – KBSTART Key Bus Transfer Start

Value	Description
0	No effect.
1	Starts a transfer through the Master Key bus.

Bit 7 – FLUSH Flush Temporary Registers

Value	Description
0	No effect.
1	Starts a flush of the temporary registers used to store the payload of the packet.

Bit 6 – READ Read Packet

Value	Description
0	No effect.
1	Starts a read sequence of the selected packet.

Downloaded from Arrow.com.

Bit 4 – HIDE Hide Packet

Value	Description
0	No effect.
1	The selected packet is not readable anymore until the next reset.

Bit 2 – INVLD Invalidate Packet

Value	Description
0	No effect.
1	Invalidates the selected packet.

Bit 1 – CKSGEN Generate Checksum

Value	Description
0	No effect.
1	Generates and programs the selected packet checksum. This action also locks the packet.

Bit 0 – PGM Program Packet

Value	Description
0	No effect.
1	The selected packet is written.

Downloaded from Arrow.com.

22.5.2 OTPC Mode Register

Name: OTPC_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPCFEN bit is cleared in the OTPC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LOCK		KBDST[1:0]				WRDIS	RDDIS
Access	R/W		R/W	R/W			R/W	R/W
Reset	0		0	0			0	0

Bit	7	6	5	4	3	2	1	0
	EMUL			NPCKT				UHCRRDIS
Access	R/W			R/W				R/W
Reset	0			0				0

Bits 31:16 – ADDR[15:0] Address
This field represents the address of the packet's header.

Bit 15 – LOCK Lock Register
The LOCK bit is set-only. Only a reset can disable lock.

Value	Description
0	The OTPC_MR register is unlocked; write access changes its value.
1	The OTPC_MR register is locked; write access does not change its value.

Bits 13:12 – KBDST[1:0] Key Bus Destination

Value	Name	Description
0	TDES	The TDES is the destination of the key transfer.
1	AES	The AES is the destination of the key transfer.

Bit 9 – WRDIS Write Disable

Value	Description
0	The write capability of the OTPC_DR register is enabled.
1	The write capability of the OTPC_DR register is disabled.

Bit 8 – RDDIS Read Disable

Value	Description
0	The read capability of the OTPC_HR and OTPC_DR registers are enabled.
1	The read capability of the OTPC_HR and OTPC_DR registers are disabled. In case of read, the returned value is 0.

Bit 7 – EMUL Emulation Enable

Value	Description
0	The Emulation mode of the User area is disabled, all accesses are computed in the OTP memory.

Downloaded from Arrow.com.

Value	Description
1	The Emulation mode of the User area is enabled, all accesses are computed in the Emulation memory.

Bit 4 – NPCKT New Packet

Value	Description
0	Updates the packet defined at the ADDR address.
1	Creates a new packet.

Bit 0 – UHCRRDIS User Hardware Configuration Register Read Disable

Value	Description
0	The User Hardware Configuration register can be read through the User Interface.
1	The User Hardware Configuration register cannot be read through the User Interface.

Downloaded from Arrow.com.

22.5.3 OTPC Address Register

Name: OTPC_AR
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								INCRT
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 16 – INCRT Increment Type

Value	Name	Description
0	AFTER_READ	Increment DADDR after a read of OTPC_DR.
1	AFTER_WRITE	Increment DADDR after a write of OTPC_DR.

Bits 7:0 – DADDR[7:0] Data Address
This field represents the word address of the payload to access through the OTPC_DR register.

Downloaded from Arrow.com.

22.5.4 OTPC Status Register

Name: OTPC_SR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							ONEF	HIDE
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	FLUSH	READ	SKBB	MKBB	EMUL	INVLD	LOCK	PGM
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 9 – ONEF One Found

Value	Description
0	No bit at ‘1’ found during the last packet read.
1	At least one ‘1’ has been found during the last packet read.

Bit 8 – HIDE Hiding On-Going

Value	Description
0	No packet hiding is on-going.
1	A packet hiding is on-going.

Bit 7 – FLUSH Flush On-Going

Value	Description
0	The temporary registers are not flushed.
1	The temporary registers are being flushed.

Bit 6 – READ Read On-Going

Value	Description
0	No packet read is on-going.
1	A packet read is running.

Bit 5 – SKBB Slave Key Bus Busy

Value	Description
0	The Slave Key bus is not busy.
1	The Slave Key bus is busy.

Bit 4 – MKBB Master Key Bus Busy

Value	Description
0	The Master Key bus is not busy.

Downloaded from Arrow.com.

Value	Description
1	The Master Key bus is busy.

Bit 3 – EMUL Emulation Enabled

Value	Description
0	The User area Emulation mode is disabled.
1	The User area Emulation mode is enabled.

Bit 2 – INVLD Invalidation On-Going

Value	Description
0	No packet invalidation is on-going.
1	A packet invalidation is running.

Bit 1 – LOCK Lock On-Going

Value	Description
0	No packet locking is on-going.
1	A packet locking is running.

Bit 0 – PGM Programming On-Going

Value	Description
0	No packet programming is on-going.
1	A packet programming is running.

Downloaded from Arrow.com.

22.5.5 OTPC Interrupt Enable Register

Name: OTPC_IER
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the OTPC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
				SECE				
Access				W				
Reset				–				

Bit	23	22	21	20	19	18	17	16
								KBERR
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
		HDERR	COERR	CKERR	EORF	EOH	EOF	EOR
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	WERR	IVERR	LKERR	PGERR	EOKT	EOI	EOL	EOP
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 28 – SECE Security and/or Safety Event Interrupt Enable

Bit 16 – KBERR Key Bus Error Interrupt Enable

Bit 14 – HDERR Hide Error Interrupt Enable

Bit 13 – COERR Corruption Error Interrupt Enable

Bit 12 – CKERR Checksum Check Error Interrupt Enable

Bit 11 – EORF End Of Refresh Interrupt Enable

Bit 10 – EOH End Of Hide Interrupt Enable

Bit 9 – EOF End Of Flush Interrupt Enable

Bit 8 – EOR End Of Read Interrupt Enable

Bit 7 – WERR Write Error Interrupt Enable

Bit 6 – IVERR Invalidation Error Interrupt Enable

Bit 5 – LKERR Locking Error Interrupt Enable

Downloaded from Arrow.com.

Bit 4 – PGERR Programming Error Interrupt Enable

Bit 3 – EOKT End Of Key Transfer Interrupt Enable

Bit 2 – EOI End Of Invalidation Interrupt Enable

Bit 1 – EOL End Of Locking Interrupt Enable

Bit 0 – EOP End Of Programming Interrupt Enable

Downloaded from Arrow.com.

22.5.6 OTPC Interrupt Disable Register

Name: OTPC_IDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the OTPC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
				SECE				
Access				W				
Reset				–				

Bit	23	22	21	20	19	18	17	16
								KBERR
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
		HDERR	COERR	CKERR	EORF	EOH	EOF	EOR
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	WERR	IVERR	LKERR	PGERR	EOKT	EOI	EOL	EOP
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 28 – SECE Security and/or Safety Event Interrupt Disable

Bit 16 – KBERR Key Bus Error Interrupt Disable

Bit 14 – HDERR Hide Error Interrupt Disable

Bit 13 – COERR Corruption Error Interrupt Disable

Bit 12 – CKERR Checksum Check Error Interrupt Disable

Bit 11 – EORF End Of Refresh Interrupt Disable

Bit 10 – EOH End Of Hide Interrupt Disable

Bit 9 – EOF End Of Flush Interrupt Disable

Bit 8 – EOR End Of Read Interrupt Disable

Bit 7 – WERR Write Error Interrupt Disable

Bit 6 – IVERR Invalidation Error Interrupt Disable

Bit 5 – LKERR Locking Error Interrupt Disable

Downloaded from Arrow.com.

Bit 4 – PGERR Programming Error Interrupt Disable

Bit 3 – EOKT End Of Key Transfer Interrupt Disable

Bit 2 – EOI End Of Invalidation Interrupt Disable

Bit 1 – EOL End Of Locking Interrupt Disable

Bit 0 – EOP End Of Programming Interrupt Disable

Downloaded from Arrow.com.

22.5.7 OTPC Interrupt Mask Register

Name: OTPC_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: Corresponding interrupt is not enabled.

1: Corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
				SECE				
Access				R				
Reset				0				

Bit	23	22	21	20	19	18	17	16
								KBERR
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
		HDERR	COERR	CKERR	EORF	EOH	EOF	EOR
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WERR	IVERR	LKERR	PGERR	EOKT	EOI	EOL	EOP
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 28 – SECE Security and/or Safety Event Interrupt Mask

Bit 16 – KBERR Key Bus Error Interrupt Mask

Bit 14 – HDERR Hide Error Interrupt Mask

Bit 13 – COERR Corruption Error Interrupt Mask

Bit 12 – CKERR Checksum Check Error Interrupt Mask

Bit 11 – EORF End Of Refresh Interrupt Mask

Bit 10 – EOH End Of Hide Interrupt Mask

Bit 9 – EOF End Of Flush Interrupt Mask

Bit 8 – EOR End Of Read Interrupt Mask

Bit 7 – WERR Write Error Interrupt Mask

Bit 6 – IVERR Invalidation Error Interrupt Mask

Bit 5 – LKERR Locking Error Interrupt Mask

Downloaded from Arrow.com.

Bit 4 – PGERR Programming Error Interrupt Mask

Bit 3 – EOKT End Of Key Transfer Interrupt Mask

Bit 2 – EOI End Of Invalidation Interrupt Mask

Bit 1 – EOL End Of Locking Interrupt Mask

Bit 0 – EOP End Of Programming Interrupt Mask

Downloaded from Arrow.com.

22.5.8 OTPC Interrupt Status Register

Name: OTPC_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
				SECE				
Access				R				
Reset				0				

Bit	23	22	21	20	19	18	17	16
								KBERR
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
		HDERR	COERR	CKERR	EORF	EOH	EOF	EOR
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WERR	IVERR	LKERR	PGERR	EOKT	EOI	EOL	EOP
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 28 – SECE Security and/or Safety Event (cleared on read)

Value	Description
0	No security or safety event occurred since the last read of OTPC_ISR.
1	One or more safety or security event occurred since the last read of OTPC_ISR. For details on the event, refer to OTPC_WPSR.

Bit 16 – KBERR Key Bus Error (cleared on read)

Value	Description
0	No error happened on the Key bus since the last read of OTPC_ISR.
1	An error happened on the Key bus since the last read of OTPC_ISR.

Bit 14 – HDERR Hide Error (cleared on read)

Value	Description
0	No hiding error occurred since the last read of OTPC_ISR.
1	A hiding error occurred since the last read of OTPC_ISR.

Bit 13 – COERR Corruption Error (cleared on read)

Value	Description
0	No corruption occurred during the last start-up since the last read of OTPC_ISR.
1	A corruption occurred since the last read of OTPC_ISR.

Bit 12 – CKERR Checksum Check Error (cleared on read)

Value	Description
0	No checksum check failure occurred during last reading sequence since the last read of OTPC_ISR.
1	A checksum check failure occurred since the last read of OTPC_ISR.

Bit 11 – EORF End Of Refresh (cleared on read)

Downloaded from Arrow.com.

Value	Description
0	No refresh sequence completion since the last read of OTPC_ISR.
1	At least one refresh sequence completion since the last read of OTPC_ISR.

Bit 10 – EOH End Of Hide (cleared on read)

Value	Description
0	No hiding sequence completion since the last read of OTPC_ISR.
1	At least one hiding sequence completion since the last read of OTPC_ISR.

Bit 9 – EOF End Of Flush (cleared on read)

Value	Description
0	No flush of the temporary registers since the last read of OTPC_ISR.
1	At least one flush hof the temporary registers has been completed since the last read of OTPC_ISR.

Bit 8 – EOR End Of Read (cleared on read)

Value	Description
0	No reading sequence completion since the last read of OTPC_ISR.
1	At least one reading sequence completion since the last read of OTPC_ISR.

Bit 7 – WERR Write Error (cleared on read)

Value	Description
0	No write error occurred since the last read of OTPC_ISR.
1	A write error occurred since the last read of OTPC_ISR.

Bit 6 – IVERR Invalidation Error (cleared on read)

Value	Description
0	No invalidation failure occurred during last invalidation sequence since the last read of OTPC_ISR.
1	A invalidation failure occurred since the last read of OTPC_ISR.

Bit 5 – LKERR Locking Error (cleared on read)

Value	Description
0	No locking failure occurred during last locking sequence since the last read of OTPC_ISR.
1	A locking failure occurred since the last read of OTPC_ISR.

Bit 4 – PGERR Programming Error (cleared on read)

Value	Description
0	No programming failure occurred during last programming sequence since the last read of OTPC_ISR.
1	A programming failure occurred since the last read of OTPC_ISR.

Bit 3 – EOKT End Of Key Transfer (cleared on read)

Value	Description
0	No key transfer completion since the last read of OTPC_ISR.
1	At least one key transfer has been completed on the Master Key bus since the last read of OTPC_ISR.

Bit 2 – EOI End Of Invalidation (cleared on read)

Value	Description
0	No invalidation sequence completion since the last read of OTPC_ISR.
1	At least one invalidation sequence completion since the last read of OTPC_ISR.

Bit 1 – EOL End Of Locking (cleared on read)

Value	Description
0	No locking sequence completion since the last read of OTPC_ISR.
1	At least one locking sequence completion since the last read of OTPC_ISR.

Bit 0 – EOP End Of Programming (cleared on read)

Downloaded from Arrow.com.

Value	Description
0	No programming sequence completion since the last read of OTPC_ISR.
1	At least one programming sequence completion since the last read of OTPC_ISR.

Downloaded from Arrow.com.

22.5.9 OTPC Header Register

Name: OTPC_HR
Offset: 0x20
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	CHECKSUM[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CHECKSUM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONE		INVLD[1:0]		LOCK	PACKET[2:0]		
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bits 31:16 – CHECKSUM[15:0] Packet Checksum
When CHECKSUM is read as 0, the checksum has not been generated. It is still possible to modify the packet content.
When CHECKSUM is read as 0x33CC, the checksum has not been generated but has already some bit at 1. Locking the packet may fail.
When CHECKSUM is read as 0xA5A5, the checksum has been generated and the last check was successful. It is impossible to modify the packet content.
When CHECKSUM is read as 0xCC33, the header of the packet is corrupted.
When CHECKSUM is read as 0xFFFF, the entire packet is no longer valid. It is possible to modify the packet content and it is up to the software to program the payload to a different value if needed.
For all other values of CHECKSUM, the checksum has been generated and the last check failed to match the checksum written in the OTP. It is impossible to modify the packet content.
This field is not writeable and is set by the OTPC during a lock request.

Bits 15:8 – SIZE[7:0] Packet Size
This field represents the size of the payload of the packet.
This field is writeable only for new packets.

Bit 7 – ONE One
This field is set to 1 by hardware and is not writeable.

Bits 5:4 – INVLD[1:0] Invalid Status
If set to value 3, this field indicates that the packet is not valid.
This field is not writeable and is set by the OTPC during an invalidation request.

Bit 3 – LOCK Lock Status
This field is not writeable and is set by the OTPC during a lock request.

Value	Description
0	The packet is not locked.

Downloaded from Arrow.com.

Value	Description
1	The packet is locked.

Bits 2:0 – PACKET[2:0] Packet Type
This field is writeable only for new packets.

Value	Name	Description
1	REGULAR	Regular packet accessible through the User Interface
2	KEY	Key packet accessible only through the Key Buses
3	BOOT_CONFIGURATION	Boot Configuration packet
4	SECURE_BOOT_CONFIGURATION	Secure Boot Configuration packet
5	HARDWARE_CONFIGURATION	Hardware Configuration packet
6	CUSTOM	Custom packet

Downloaded from Arrow.com.

22.5.10 OTPC Data Register

Name: OTPC_DR
Offset: 0x24
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Packet Data
This field represents the data of one of the packet. The data read or written is located at the address specified by the DADDR field of OTPC_AR register.

Downloaded from Arrow.com.

22.5.11 OTPC Boot Addresses Register

Name: OTPC_BAR
Offset: 0x30
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	SBCADDR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SBCADDR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BCADDR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BCADDR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – SBCADDR[15:0] Secure Boot Configuration Address
This field represents the address of the “Secure Boot Configuration” special packet.

Bits 15:0 – BCADDR[15:0] Boot Configuration Address
This field represents the address of the “Boot Configuration” special packet.

Downloaded from Arrow.com.

22.5.12 OTPC Custom Address Register

Name: OTPC_CAR
Offset: 0x34
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CADDR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CADDR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CADDR[15:0] Custom Address
This field represents the address of the “Custom” special packet.

Downloaded from Arrow.com.

22.5.13 OTPC User Hardware Configuration 0 Register

Name: OTPC_UHC0R
Offset: 0x50
Reset: 0x00000000
Property: Read-only

Note: The reset value depends on hardware configuration.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	JTAGDIS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – JTAGDIS[7:0] JTAG Disable

Value	Description
0	The JTAG is enabled.
Non-zero	The JTAG is disabled.

Downloaded from Arrow.com.

22.5.14 OTPC User Hardware Configuration 1 Register

Name: OTPC_UHC1R
Offset: 0x54
Reset: 0x00000000
Property: Read-only

Note: The reset value depends on hardware configuration.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							URFDIS	CPGDIS
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	CLKDIS	CINVDIS				SBCPGDIS	SBCLKDIS	SBCINVDIS
Access	R	R				R	R	R
Reset	0	0				0	0	0

Bit	7	6	5	4	3	2	1	0
	BCPGDIS	BCLKDIS	BCINVDIS	UHCPGDIS	UHCLKDIS	UHCINVDIS	UPGDIS	URDDIS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 17 – URFDIS User Refresh Disable

Value	Description
0	The OTPC_CR.REFRESH bit is fully functional.
1	The OTPC_CR.REFRESH bit is only functional in Emulation mode.

Bit 16 – CPGDIS Custom Packet Program Disable

Value	Description
0	The programming of Custom Special Packet is allowed.
1	The programming of Custom Special Packet is forbidden.

Bit 15 – CLKDIS Custom Packet Lock Disable

Value	Description
0	The generation of the checksum (lock) of the Custom Special Packet is allowed.
1	The generation of the checksum (lock) of the Custom Special Packet is forbidden.

Bit 14 – CINVDIS Custom Packet Invalidation Disable

Value	Description
0	The invalidation of the Custom Special Packet is allowed.
1	The invalidation of the Custom Special Packet is forbidden.

Bit 10 – SBCPGDIS Secure Boot Configuration Packet Program Disable

Value	Description
0	The programming of Secure Boot Configuration Special Packet is allowed.
1	The programming of Secure Boot Configuration Special Packet is forbidden.

Bit 9 – SBCLKDIS Secure Boot Configuration Packet Lock Disable

Value	Description
0	The generation of the checksum (lock) of the Secure Boot Configuration Special Packet is allowed.

Downloaded from Arrow.com.

Value	Description
1	The generation of the checksum (lock) of the Secure Boot Configuration Special Packet is forbidden.

Bit 8 – SBCINVDIS Secure Boot Configuration Packet Invalidation Disable

Value	Description
0	The invalidation of the Secure Boot Configuration Special Packet is allowed.
1	The invalidation of the Secure Boot Configuration Special Packet is forbidden.

Bit 7 – BCPGDIS Boot Configuration Packet Program Disable

Value	Description
0	The programming of Boot Configuration Special Packet is allowed.
1	The programming of Boot Configuration Special Packet is forbidden.

Bit 6 – BCLKDIS Boot Configuration Packet Lock Disable

Value	Description
0	The generation of the checksum (lock) of the Boot Configuration Special Packet is allowed.
1	The generation of the checksum (lock) of the Boot Configuration Special Packet is forbidden.

Bit 5 – BCINVDIS Boot Configuration Packet Invalidation Disable

Value	Description
0	The invalidation of the Boot Configuration Special Packet is allowed.
1	The invalidation of the Boot Configuration Special Packet is forbidden.

Bit 4 – UHCPGDIS User Hardware Configuration Packet Program Disable

Value	Description
0	The programming of User Hardware Configuration Special Packet is allowed.
1	The programming of User Hardware Configuration Special Packet is forbidden.

Bit 3 – UHCLKDIS User Hardware Configuration Packet Lock Disable

Value	Description
0	The generation of the checksum (lock) of the User Hardware Configuration Special Packet is allowed.
1	The generation of the checksum (lock) of the User Hardware Configuration Special Packet is forbidden.

Bit 2 – UHCINVDIS User Hardware Configuration Packet Invalidation Disable

Value	Description
0	The invalidation of the User Hardware Configuration Special Packet is allowed.
1	The invalidation of the User Hardware Configuration Special Packet is forbidden.

Bit 1 – UPGDIS User programming Disable

Value	Description
0	The OTPC_CR.PGM bit is fully functional.
1	The OTPC_CR.PGM bit is not functional.

Bit 0 – URDDIS User Read Disable

Value	Description
0	The OTPC_CR.READ bit is fully functional.
1	The OTPC_CR.READ bit is not functional.

Downloaded from Arrow.com.

22.5.15 OTPC Product UID x Register

Name: OTPC_UIDxR
Offset: 0x60 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read-only

Note: The reset value depends on hardware configuration.

Bit	31	30	29	28	27	26	25	24
	UID[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	UID[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UID[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UID[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – UID[31:0] Unique Product ID
This field represents the unique product ID.

Downloaded from Arrow.com.

22.5.16 OTPC Write Protection Mode Register

Name: OTPC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				FIRSTE		WPCTEN	WPITEN	WPCFEN
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x4F5450	PASSWD	Writing any other value in this field aborts the write operation. Always reads as 0.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in OTPC_WPSR.WPVSRC and the last software control error type is reported in OTPC_WPSR.SWETYP; The OTPC_ISR.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in OTPC_WPSR.WPVSRC and only the first software control error type is reported in OTPC_WPSR.SWETYP. The OTPC_ISR.SECE flag is set at the first error occurrence within a series.

Bit 2 – WPCTEN Write Protection Control Enable

Value	Description
0	Disables the write protection of the control if WPKEY matches to 0x4F5450 (OTP in ASCII).
1	Enables the write protection of the control if WPKEY matches to 0x4F5450 (OTP in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection of the interruption configuration if WPKEY matches to 0x4F5450 (OTP in ASCII).
1	Enables the write protection of the interruption configuration if WPKEY matches to 0x4F5450 (OTP in ASCII).

Bit 0 – WPCFEN Write Protection Configuration Enable

Downloaded from Arrow.com.

Value	Description
0	Disables the write protection of the configuration if WPKEY matches to 0x4F5450 (OTP in ASCII).
1	Enables the write protection of the configuration if WPKEY matches to 0x4F5450 (OTP in ASCII).

Downloaded from Arrow.com.

22.5.17 OTPC Write Protection Status Register

Name: OTPC_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS				SWETYP[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
					SWE	SEQE	CGD	WPVS
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – ECLASS Software Error Class

Value	Name	Description
0	WARNING	An abnormal access that does not have any impact.
1	ERROR	An abnormal access that may have an impact.

Bits 27:24 – SWETYP[3:0] Software Error Type

Value	Name	Description
0	READ_WO	A write-only register has been read (warning).
1	WRITE_RO	A write access has been performed on a read-only register (warning).
2	CONF_CHG	A change has been made into the configuration (error).
3	KEY_ERROR	A write has been computed in OTPC_CR or OTPC_WPMR register with a wrong value in the related KEY field (error).
4	DATA_ACC	The non-secure world application tried to read a packet from the secure world (error).

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of OTPC_WPSR.
1	A software error has occurred since the last read of OTPC_WPSR. The field SWETYP details the type of software error encountered.

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of OTPC_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of OTPC_WPSR. This flag can be set under abnormal operating conditions.

Downloaded from Arrow.com.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	No clock glitch has occurred since the last read of OTPC_WPSR.
1	A clock glitch has occurred since the last read of OTPC_WPSR. This flag can be set under abnormal operating conditions.

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protection violation has occurred since the last read of OTPC_WPSR.
1	A write protection violation has occurred since the last read of OTPC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into WPVSRC.

Downloaded from Arrow.com.

23. Special Function Registers (SFR)

23.1 Description

Special Function Registers (SFR) manage specific aspects of the integrated memory, bridge implementations, processor and other functionality not controlled elsewhere.

23.2 Embedded Characteristics

- 32-bit Special Function Registers Control Specific Behavior of the Product

Downloaded from Arrow.com.

23.3 Register Summary

Offset	Name	Bit Pos.								
0x00 ... 0x03	Reserved									
0x04	SFR_CCFG_EBICSA	31:24							DDR_MP_EN	NFD0_ON_D16
		23:16				DQIEN_F				EBI_DRIVE
		15:8							EBI_DBPDC	EBI_DBPUC
		7:0			EBI_CS5A	EBI_CS4A	EBI_CS3A		EBI_CS1A	
0x08 ... 0x0F	Reserved									
0x10	SFR_OHCIICR	31:24								
		23:16	UDPPUDIS							
		15:8						SUSP2	SUSP1	SUSP0
		7:0			APPSTART	ARIE		RES2	RES1	RES0
0x14	SFR_OHCIISR	31:24								
		23:16								
		15:8								
		7:0						RIS2	RIS1	RIS0
0x18 ... 0x33	Reserved									
0x34	SFR_UTMIHSTRIM	31:24								
		23:16						SLOPE2[2:0]		
		15:8		SLOPE1[2:0]				SLOPE0[2:0]		
		7:0								
0x38	SFR_UTMIFSTRIM	31:24		ZP_CAL[2:0]				ZN_CAL[2:0]		
		23:16		ZP[2:0]				ZN[2:0]		
		15:8								
		7:0								
0x3C	SFR_UTMISWAP	31:24								
		23:16								
		15:8								
		7:0						PORT2	PORT1	PORT0
0x40 ... 0x7B	Reserved									
0x7C	SFR_LS	31:24								
		23:16								MEM_POWER_GATING_ULP1_EN
		15:8							LS9	LS8
		7:0	LS7	LS6	LS5	LS4	LS3	LS2	LS1	LS0
0x80 ... 0xE3	Reserved									
0xE4	SFR_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN

Downloaded from Arrow.com.

23.3.1 EBI Chip Select Register

Name: SFR_CCFG_EBICSA
Offset: 0x04
Reset: 0x00000300
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							DDR_MP_EN	NFD0_ON_D16
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
				DQIEN_F				EBI_DRIVE
Access				R/W				R/W
Reset				0				0

Bit	15	14	13	12	11	10	9	8
							EBI_DBPDC	EBI_DBPUC
Access							R/W	R/W
Reset							1	1

Bit	7	6	5	4	3	2	1	0
			EBI_CS5A	EBI_CS4A	EBI_CS3A		EBI_CS1A	
Access			R/W	R/W	R/W		R/W	
Reset			0	0	0		0	

Bit 25 – DDR_MP_EN DDR Multi-port Enable

Value	Description
0	DDR Multi-port is disabled (default).
1	DDR Multi-port is enabled, performance is increased.

Bit 24 – NFD0_ON_D16 NAND Flash Databus Selection

Value	Description
0	NAND Flash I/Os are connected to D0–D7 (default).
1	NAND Flash I/Os are connected to D16–D23.

Bit 20 – DQIEN_F Force Analog Input Comparator Configuration

Value	Description
0	No effect
1	Enables the input comparator in the VDDIOM I/O data lines. This bit must be set to one in an initialization phase whenever an MPDDRC external component (DDR2 or LPDDR) and an SMC external component (e.g., NAND Flash) are multiplexed on the D0-D15 bus.

Bit 16 – EBI_DRIVE EBI I/O Drive Configuration

Value	Description
0	EBI D0–D15 Low Drive
1	EBI D0–D15 High Drive

Bit 9 – EBI_DBPDC EBI Data Bus Pulldown Configuration

Value	Description
0	EBI D0–D15 Data Bus bits are not internally pulled down.
1	EBI D0–D15 Data Bus bits are internally pulled down to the ground.

Bit 8 – EBI_DBPUC EBI Data Bus Pullup Configuration

Downloaded from Arrow.com.

Value	Description
0	EBI D0–D15 Data Bus bits are internally pulled up to the VDDIOM power supply.
1	EBI D0–D15 Data Bus bits are not internally pulled up.

Bit 5 – EBI_CS5A EBI Chip Select 5 Assignment

Value	Description
0	EBI Chip Select 5 is only assigned to the Static Memory Controller and EBI_NCS5 behaves as defined by the SMC.
1	EBI Chip Select 5 is assigned to the Static Memory Controller.

Bit 4 – EBI_CS4A EBI Chip Select 4 Assignment

Value	Description
0	EBI Chip Select 4 is only assigned to the Static Memory Controller and EBI_NCS4 behaves as defined by the SMC.
1	EBI Chip Select 4 is assigned to the Static Memory Controller.

Bit 3 – EBI_CS3A EBI Chip Select 3 Assignment

Value	Description
0	EBI Chip Select 3 is only assigned to the Static Memory Controller and EBI_NCS3 behaves as defined by the SMC.
1	EBI Chip Select 3 is assigned to the Static Memory Controller and the NAND Flash Logic is activated.

Bit 1 – EBI_CS1A EBI Chip Select 1 Assignment

Value	Description
0	EBI Chip Select 1 is assigned to the Static Memory Controller (SMC).
1	EBI Chip Select 1 is assigned to the MPDDRC or the SDRAMC controller.

Downloaded from Arrow.com.

23.3.2 OHCI Interrupt Configuration Register

Name: SFR_OHCIICR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	UDPPUDIS							
Access	R/W							
Reset	0							

Bit	15	14	13	12	11	10	9	8
						SUSP2	SUSP1	SUSP0
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
			APPSTART	ARIE		RES2	RES1	RES0
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit 23 – UDPPUDIS Reserved

Value	Description
0	Must write 0.

Bits 8, 9, 10 – SUSP USB PORTx

Value	Description
0	Does not suspend USB PORTx.
1	Forces PORTx suspend.

Bit 5 – APPSTART Reserved

Value	Description
0	Must write 0.

Bit 4 – ARIE OHCI Asynchronous Resume Interrupt Enable

Value	Description
0	Disables interrupt.
1	Enables interrupt.

Bits 0, 1, 2 – RESx USB PORTx Reset

Value	Description
0	No effect (USB PORTx reset released, default value)
1	Resets USB PORTx.

Downloaded from Arrow.com.

23.3.3 OHCI Interrupt Status Register

Name: SFR_OHCIISR
Offset: 0x14
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						RIS2	RIS1	RIS0
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 0, 1, 2 – RIS OHCI Resume Interrupt Status Port x

Value	Description
0	OHCI port resume is not detected.
1	OHCI port resume is detected.

Downloaded from Arrow.com.

23.3.4 UTMI High-Speed Trimming Register

Name: SFR_UTMIHSTRIM
Offset: 0x34
Reset: 0x00044433
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						SLOPE2[2:0]		
Access						R/W	R/W	R/W
Reset						1	0	0

Bit	15	14	13	12	11	10	9	8
		SLOPE1[2:0]				SLOPE0[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		1	0	0		1	0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 8:10, 12:14, 16:18 – SLOPE UTMI HS PORTx Transceiver Slope Trimming
Adjusts HS transceiver output slope for PORTx.
These bits are duplicated for each port because the HS slope depends on the PCB line length. Short lines tend to be seen as lumped capacitances and exhibit a slower rise/fall time, while longer lines appear as transmission lines with sharper edges.

Value	Name
111	Slower
110	–
101	–
100	Default
011	–
010	–
001	–
000	Faster

Downloaded from Arrow.com.

23.3.5 UTMI Full-Speed Trimming Register

Name: SFR_UTMIFSTRIM
Offset: 0x38
Reset: 0x00430211
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
		ZP_CAL[2:0]				ZN_CAL[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
		ZP[2:0]				ZN[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		1	0	0		0	1	1

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 30:28 – ZP_CAL[2:0] FS Transceiver PMOS Impedance Calibration
Adjusts the FS transceiver PMOS output impedance calibration.

Value	Name
011	Higher
010	–
001	–
000	Default
111	–
110	–
101	–
100	Lower

Bits 26:24 – ZN_CAL[2:0] FS Transceiver NMOS Impedance Calibration
Adjusts the FS transceiver NMOS output impedance calibration.

Value	Name
011	Higher
010	–
001	–
000	Default
111	–
110	–
101	–
100	Lower

Bits 22:20 – ZP[2:0] FS Transceiver PMOS Impedance Trimming
Adjusts the FS transceiver PMOS output impedance.

Value	Name
111	Higher

Downloaded from Arrow.com.

Value	Name
110	–
101	–
100	Default
011	–
010	–
001	–
000	Lower

Bits 18:16 – ZN[2:0] FS Transceiver NMOS Impedance Trimming
Adjusts the FS transceiver NMOS output impedance.

Value	Name
111	Lower
110	–
101	–
100	–
011	Default
010	–
001	–
000	Higher

Downloaded from Arrow.com.

23.3.6 UTMI DP/DM Pin Swapping Register

Name: SFR_UTMISWAP
Offset: 0x3C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						PORT2	PORT1	PORT0
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 0, 1, 2 – PORT PORT x DP/DM Pin Swapping
0 (NORMAL): DP/DM normal pinout.
1 (SWAPPED): DP/DM swapped pinout.

Downloaded from Arrow.com.

23.3.7 SFR Light Sleep Register

Name: SFR_LS
Offset: 0x7C
Reset: 0x00000000
Property: Read/Write

The following configuration values are valid for all listed LSx bit names of this register:

0: Disables Light Sleep mode.

1: Enables Light Sleep mode.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								MEM_POWER_GATING_ULP1_EN
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
							LS9	LS8
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	LS7	LS6	LS5	LS4	LS3	LS2	LS1	LS0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 16 – MEM_POWER_GATING_ULP1_EN Light Sleep Value for ULP1 Power-Gated Memories
The memory power gating can be automatically enabled when entering ULP1 Low-power mode. Refer to section "Electrical Characteristics".

Value	Description
0	Light Sleep mode is not activated by the MEM_POWER_GATING_ULP1 output signal from PMC.
1	Light Sleep mode is activated when the MEM_POWER_GATING_ULP1 output signal from PMC is activated.

Bit 9 – LS9 Light Sleep Value (ARM926)

Bit 8 – LS8 Light Sleep Value (ROM + OTPC)

Bit 7 – LS7 Light Sleep Value (SRAM1 (OTPC))

Bit 6 – LS6 Light Sleep Value (SRAM0)

Bit 5 – LS5 Light Sleep Value (EHCI/OHCI)

Bit 4 – LS4 Light Sleep Value (HXDMA)

Bit 3 – LS3 Light Sleep Value (HUSB)

Bit 2 – LS2 Light Sleep Value (SDMMC)

Downloaded from Arrow.com.

Bit 1 – LS1 Light Sleep Value (HLCDC5)

Bit 0 – LS0 Light Sleep Value (GFX2D)

Downloaded from Arrow.com.

23.3.8 SFR Write Protection Mode Register

Name: SFR_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

All registers are write-protected.

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x534652	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables write protection if WPKEY corresponds to 0x534652 (“SFR” in ASCII).
1	Enables write protection if WPKEY corresponds to 0x534652 (“SFR” in ASCII).

Downloaded from Arrow.com.

24. Bus Matrix (MATRIX)

24.1 Description

The Bus Matrix (MATRIX) implements a multi-layer AHB, based on the AHB-Lite protocol, that enables parallel access paths between multiple masters and slaves in a system, thus increasing the overall bandwidth.

The MATRIX interconnects 15 masters to 13 slaves. The normal latency to connect a master to a slave is one cycle, except for the default master of the accessed slave which is connected directly (zero cycle latency).

The MATRIX user interface is compliant with the Arm Advanced Peripheral Bus.

24.1.1 MATRIX Masters

The MATRIX manages 15 masters listed in the table below. Each master can perform an access, concurrently with others, to an available slave. The MATRIX operates at the master clock (MCK) frequency. Each master has its own decoder, which is defined specifically for each master. In order to simplify the addressing, all the masters have the same decodings.

Table 24-1. List of MATRIX Masters

Master No.	Description
0	ARM926 instruction
1	ARM926 data
2, 3	XDMA controller with QoS support
4	SDMMC0 DMA
5	SDMMC1 DMA
6	USB high-speed device port (UDPHS) DMA
7	USB high-speed host port (UHPHS) EHCI DMA
8	USB high-speed host port (UHPHS) OHCI DMA
9	ISI DMA
10	EMAC0 DMA
11	EMAC1 DMA
12	OTP controller master interface
13	GFX2D DMA
14	LCDC DMA with QoS support

24.1.2 MATRIX Slaves

The MATRIX manages the 13 slaves listed in the table below. Each slave has its own arbiter providing a dedicated arbitration per slave.

Table 24-2. List of MATRIX Slaves

Slave No.	Description
0	SRAM0
1	OTPC slave interface (ROM and OTP memory)

Downloaded from Arrow.com.

.....continued	
Slave No.	**Description**
2	UDPHS dual port RAM
	UHPHS OHCI configuration registers
	UHPHS EHCI configuration registers
3	External Bus Interface / MPDDRC / SDRAMC port 0 with QoS support
4	MPDDRC / SDRAMC port 1(1) with QoS support
5	MPDDRC / SDRAMC port 2(1) with QoS support
6	MPDDRC / SDRAMC port 3(1) with QoS support
7	Peripheral bridge 0
8	Peripheral bridge 1
9	QSPI
10	SDMMC0 configuration registers
11	SDMMC1 configuration registers
12	SRAM1

Note:

1. The multiport is available for the SDRAMC when the Multiplexed Address/Data Lines or the Address/Data/Command Lines mode is used. In standard SDRAM Connection mode, only port 0 is used. Refer to “Interface with Multiplexed Data/Address Lines and Data/Address/Command Lines”, in section SDRAM Controller (SDRAMC).

24.1.3 Master to Slave Access

The table below describes how masters and slaves can be interconnected. Writing in a register or field not dedicated to a master or a slave has no effect.

Table 24-3. Master to Slave Access

Masters		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Slaves		ARM926 Instr.	ARM926 Data	XDMAC0	XDMAC1	SDMMC0 DMA	SDMMC1 DMA	UDPHS DMA	UHPHS EHCI	UHPHS OHCI	ISI DMA	EMAC0 DMA	EMAC1 DMA	OTPC Master I/F	GFX2D	LCDC DMA
0	SRAM0	X	X	X	X	X	X	X	X	X	X	X	X	–	X	X
1	OTPC Slave I/F	X	X	–	–	–	–	–	–	–	–	–	–	–	–	–
2	UDPHS DPRAM UHPHS EHCI config. reg. UHPHS OHCI config. reg.	X	X	–	–	–	–	–	–	–	–	–	–	–	–	–
3	EBI / MPDDRC / SDRAMC port 0	X	X	X	X	X	X	X	X	X	X	X	X	–	X	X
4	MPDDRC / SDRAMC port 1	X	–	X	–	X	–	X	–	–	–	X	–	–	–	–
5	MPDDRC / SDRAMC port 2	–	X	–	X	–	X	–	X	X	–	–	X	–	X	–

Downloaded from Arrow.com.

.....continued

Masters		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Slaves		ARM926 Instr.	ARM926 Data	XDMAC0	XDMAC1	SDMMC0 DMA	SDMMC1 DMA	UDPHS DMA	UHPHS EHCI	UHPHS OHCI	ISI DMA	EMAC0 DMA	EMAC1 DMA	OTPC Master I/F	GFX2D	LCDC DMA
6	MPDDRC / SDRAMC port 3	–	–	–	–	–	–	–	–	–	X	–	–	–	–	X
7	Peripheral Bridge 0	–	X	X	X	–	–	–	–	–	–	–	–	–	–	–
8	Peripheral Bridge 1	–	X	X	X	–	–	–	–	–	–	–	–	–	–	–
9	QSPI	X	X	X	X	–	–	–	–	–	–	–	–	–	–	X
10	SDMMC0 config. reg.	–	X	–	–	–	–	–	–	–	–	–	–	–	–	–
11	SDMMC1 config. reg.	–	X	–	–	–	–	–	–	–	–	–	–	–	–	–
12	SRAM1	X	X	–	–	–	–	–	–	–	–	–	–	X	–	–

24.2 Embedded Characteristics

- 15 Configurable Master Ports
- 13 Configurable Slave Ports
- One Decoder for Each Master
- One Remap Function for Each Master
- Support for Long Bursts of Length 32, 64, 128 and Up to the Limit of 256-bit Burst Beats of Words
- Enhanced Programmable Mixed Arbitration for Each Slave
  - Round-robin
  - Fixed priority
  - Latency Quality of Service
- Programmable Default Master for Each Slave
  - No default master
  - Last accessed default master
  - Fixed default master
- Deterministic Maximum Access Latency for Masters
- Zero or One Cycle Arbitration Latency for the First Access of a Burst
- Bus Lock Forwarding to Slaves
- Master Number Forwarding to Slaves
- Register Write Protection of User Interface Registers

24.3 Memory Mapping

The MATRIX provides one decoder for every master interface. The decoder offers each master several memory mappings. Each memory area can be assigned to several slaves. Booting at the same address while using different slaves (i.e., external RAM, internal ROM or internal Flash, etc.) becomes possible.

24.4 Special Bus Granting Techniques

The MATRIX provides some speculative bus granting techniques in order to anticipate access requests from masters. Hence, latency is reduced at first access of a burst or, for a single transfer, as long as the slave is free from any other master access. It does not provide any benefit if the slave is continuously accessed by more than one master, since arbitration is pipelined and has no negative effect on the slave bandwidth or access latency.

This bus granting technique sets a different default master for every slave.

Downloaded from Arrow.com.

At the end of the current access, if no other request is pending, the slave remains connected to its associated default master. A slave can be associated with three kinds of default masters:

- no default master
- last access master
- fixed default master

To change from one type of default master to another, the user interface provides Slave Configuration registers (MATRIX_SCFGx), one for every slave, which set a default master for each slave. MATRIX_SCFGx contain two fields to manage master selection: DEFMSTR_TYPE and FIXED_DEFMSTR. The 2-bit DEFMSTR_TYPE field selects the default master type (no default, last access master, fixed default master), whereas the 4-bit FIXED_DEFMSTR field selects a fixed default master provided that DEFMSTR_TYPE is set to fixed default master. Refer to section 24.10.2 MATRIX_SCFGx.

24.5 No Default Master

After the end of the current access, if no other request is pending, the slave is disconnected from all masters.

This configuration incurs one latency clock cycle for the first access of a burst after bus Idle. Arbitration without default master may be used for masters that perform significant bursts or several transfers with no Idle cycle in-between, or if the slave bus bandwidth is widely used by one or more masters.

This configuration provides no benefit on access latency or bandwidth when reaching maximum slave bus throughput whatever the number of requesting masters.

24.6 Last Access Master

After the end of the current access, if no other request is pending, the slave remains connected to the last master that performed an access request.

This enables the MATRIX to remove the one latency cycle for the last master that accessed the slave. Other non-privileged masters still get one latency clock cycle if they need to access the same slave. This technique is useful for masters that mainly perform single accesses or short bursts with some Idle cycles in-between.

This configuration provides no benefit on access latency or bandwidth when reaching maximum slave bus throughput whatever is the number of requesting masters.

24.7 Fixed Default Master

After the end of the current access, if no other request is pending, the slave connects to its fixed default master. Unlike the last access master, the fixed default master does not change unless the user modifies it by software (FIXED_DEFMSTR field of the related MATRIX_SCFG).

This allows the MATRIX arbiters to remove the one latency clock cycle for the fixed default master of the slave. All requests attempted by the fixed default master do not cause any arbitration latency, whereas other non-privileged masters get one latency cycle. This technique is useful for a master that mainly performs single accesses or short bursts with Idle cycles in-between.

This configuration provides no benefit on access latency or bandwidth when reaching maximum slave bus throughput, regardless of the number of requesting masters.

24.8 Arbitration

The MATRIX provides an arbitration technique that reduces latency when conflicts occur, i.e., when two or more masters try to access the same slave at the same time. One arbiter per slave is provided, thus arbitrating each slave specifically.

The user can either choose one of the following arbitration types, or mix them for each slave:

1. Round-robin Arbitration (default)

Downloaded from Arrow.com.

2. Fixed Priority Arbitration

The resulting algorithm may be complemented by selecting a default master configuration for each slave.

When re-arbitration must be done, specific conditions apply. See section 24.8.1 Arbitration Scheduling .

24.8.1 Arbitration Scheduling

Each arbiter has the ability to arbitrate between two or more different master requests. In order to avoid burst breaking as well as to provide the maximum throughput for slave interfaces, arbitration may only take place during the following cycles:

- Idle Cycles: When a slave is not connected to any master or is connected to a master which is not currently accessing it.
- Single Cycles: When a slave is currently doing a single access.
- End of Burst Cycles: When the current cycle is the last cycle of a burst transfer. For defined length burst, predicted end of burst matches the size of the transfer but is managed differently for undefined length burst. See section 24.8.1.1 Undefined Length Burst Arbitration.
- Slot Cycle Limit: When the slot cycle counter has reached the limit value indicating that the current master access is too long and must be broken. See section 24.8.1.2 Slot Cycle Limit Arbitration.

24.8.1.1 Undefined Length Burst Arbitration

In order to prevent long burst lengths that can lock the access to the slave for an excessive period of time, the user can trigger the re-arbitration before the end of the incremental bursts. The re-arbitration period can be selected from the following Undefined Length Burst Type (ULBT) possibilities:

- Unlimited: no predetermined end of burst is generated. This value enables 1 Kbyte burst lengths.
- 1-beat bursts: predetermined end of burst is generated at each single transfer during the INCR transfer.
- 4-beat bursts: predetermined end of burst is generated at the end of each 4-beat boundary during INCR transfer.
- 8-beat bursts: predetermined end of burst is generated at the end of each 8-beat boundary during INCR transfer.
- 16-beat bursts: predetermined end of burst is generated at the end of each 16-beat boundary during INCR transfer.
- 32-beat bursts: predetermined end of burst is generated at the end of each 32-beat boundary during INCR transfer.
- 64-beat bursts: predetermined end of burst is generated at the end of each 64-beat boundary during INCR transfer.
- 128-beat bursts: predetermined end of burst is generated at the end of each 128-beat boundary during INCR transfer.

The use of undefined length 8-beat bursts, or less, is discouraged since this may decrease the overall bus bandwidth due to arbitration and slave latencies at each first access of a burst.

Undefined-length bursts lower than 8 beats should not be used since this may decrease the overall bus bandwidth due to arbitration and slave latencies at each first access of a burst.

However, if the length of undefined-length bursts is known for a master, it is recommended to configure MATRIX_MCFG.ULBT accordingly.

24.8.1.2 Slot Cycle Limit Arbitration

The MATRIX contains specific logic to break long accesses, such as very long bursts on a very slow slave (e.g., an external low speed memory). At each arbitration time, a counter is loaded with the value previously written in MATRIX_SCFGx.SLOT_CYCLE and decreased at each clock cycle. When the counter elapses, the arbiter has the ability to re-arbitrate at the end of the current system bus access cycle.

Unless a master has a very tight access latency constraint, which could lead to data overflow or underflow due to a badly undersized internal FIFO with respect to its throughput, the Slot Cycle Limit should be disabled (SLOT_CYCLE = 0) or set to its default maximum value in order not to inefficiently break long bursts performed by some masters.

In most cases, this feature is not needed and should be disabled for power saving.

Downloaded from Arrow.com.

WARNING This feature cannot prevent any slave from locking its access indefinitely.

24.8.2 Arbitration Priority Scheme

The MATRIX arbitration scheme is organized in priority pools, each corresponding to an access criticality class as shown in the “Latency Quality of Service” column in the table below. When the Latency Quality of Service is enabled for a master-slave pair through the MATRIX, the priority pool number to use for arbitration at the slave port is determined from the master. When the Latency Quality of Service is disabled, it is determined through the MATRIX user interface. See 24.10.3 MATRIX_PRASx.

After reset, the Latency Quality of Service is enabled by default on all of the master ports that are connected to a master driving the Latency Quality of Service signals, as shown in the bit LQOSEN of 24.10.3 MATRIX_PRASx and 24.10.4 MATRIX_PRBSx.

Table 24-4. Arbitration Priority Pools

Priority Pool	Latency Quality of Service
3	Latency Critical
2	Latency Sensitive
1	Bandwidth Sensitive
0	Background Transfers

Round-robin priority is used in the highest and lowest priority pools 3 and 0, whereas fixed level priority is used between priority pools and in the intermediate priority pools 2 and 1.

For each slave, each master is assigned to one of the slave priority pools based on the Latency Quality of Service inputs or to the priority registers for slaves (MxPR fields of MATRIX_PRAS and MATRIX_PRBS). When evaluating master requests, this priority pool level always takes precedence.

After reset, most of the masters belong to the lowest priority pool (MxPR = 0, Background Transfer) and are therefore granted bus access in a true round-robin order.

The highest priority pool must be specifically reserved for masters requiring very low access latency. If more than one master belongs to this pool, those masters are granted bus access in a biased round-robin manner which enables tight and deterministic maximum access latency from system bus requests. In the worst case, any currently occurring high-priority master request is granted after the current bus master access has ended and any other high priority pool master requests have been granted once each.

The lowest priority pool shares the remaining bus bandwidth between masters.

Intermediate priority pools enable fine priority tuning. Typically, a latency-sensitive master or a bandwidth-sensitive master use such a priority level. The higher the priority level (MxPR value), the higher the master priority.

For good CPU performance, it is recommended to let the CPU priority configured with the default reset value 2 (Latency Sensitive).

All combinations of MxPR values are allowed for all masters and slaves. For example, some masters might be assigned the highest priority pool (round-robin), and remaining masters the lowest priority pool (round-robin), with no master for intermediate fixed priority levels.

24.8.2.1 Fixed Priority Arbitration

The fixed priority arbitration algorithm is the first and only arbitration algorithm applied between masters from distinct priority pools. It is also used in priority pools other than the highest and lowest priority pools (intermediate priority pools).

Fixed priority arbitration enables the MATRIX arbiters to dispatch the requests from different masters to the same slave by using the fixed priority defined by the user in the MxPR field for each master in the registers, MATRIX_PRAS and MATRIX_PRBS. If two or more master requests are active at the same time, the master with the highest priority MxPR number is serviced first.

Downloaded from Arrow.com.

In intermediate priority pools, if two or more master requests with the same priority are active at the same time, the master with the highest number is serviced first.

24.8.2.2 Round-Robin Arbitration

This algorithm is only used in the highest and lowest priority pools. It allows the MATRIX arbiters to properly dispatch requests from different masters to the same slave. If two or more master requests are active at the same time in the priority pool, they are serviced in a round-robin increasing master number order.

24.9 Register Write Protection

To prevent any single software error from corrupting MATRIX behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the Write Protection Mode register (MATRIX_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the Write Protection Status register (MATRIX_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS flag is reset by writing the MATRIX_WPMR with the appropriate access key WPKEY.

The registers listed below can be write-protected.

Downloaded from Arrow.com.

24.10 Register Summary

Offset	Name	Bit Pos.								
0x00	MATRIX_MCFG0	31:24								
		23:16								
		15:8								
		7:0						ULBT[2:0]		
...										
0x38	MATRIX_MCFG14	31:24								
		23:16								
		15:8								
		7:0						ULBT[2:0]		
0x3C ... 0x3F	Reserved									
0x40	MATRIX_SCFG0	31:24								
		23:16			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
		15:8								SLOT_CYCLE[8]
		7:0	SLOT_CYCLE[7:0]							
...										
0x70	MATRIX_SCFG12	31:24								
		23:16			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
		15:8								SLOT_CYCLE[8]
		7:0	SLOT_CYCLE[7:0]							
0x74 ... 0x7F	Reserved									
0x80	MATRIX_PRAS0	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0x84	MATRIX_PRBS0	31:24						LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0x88	MATRIX_PRAS1	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0x8C	MATRIX_PRBS1	31:24						LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0x90	MATRIX_PRAS2	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0x94	MATRIX_PRBS2	31:24						LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0x98	MATRIX_PRAS3	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.							
0x9C	MATRIX_PRBS3	31:24						LQOSEN14	M14PR[1:0]
		23:16	LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8	LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0	LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xA0	MATRIX_PRAS4	31:24	LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16	LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8	LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0	LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xA4	MATRIX_PRBS4	31:24					LQOSEN14	M14PR[1:0]	
		23:16	LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8	LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0	LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xA8	MATRIX_PRAS5	31:24	LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16	LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8	LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0	LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xAC	MATRIX_PRBS5	31:24					LQOSEN14	M14PR[1:0]	
		23:16	LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8	LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0	LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xB0	MATRIX_PRAS6	31:24	LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16	LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8	LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0	LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xB4	MATRIX_PRBS6	31:24					LQOSEN14	M14PR[1:0]	
		23:16	LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8	LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0	LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xB8	MATRIX_PRAS7	31:24	LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16	LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8	LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0	LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xBC	MATRIX_PRBS7	31:24					LQOSEN14	M14PR[1:0]	
		23:16	LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8	LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0	LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xC0	MATRIX_PRAS8	31:24	LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16	LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8	LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0	LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xC4	MATRIX_PRBS8	31:24					LQOSEN14	M14PR[1:0]	
		23:16	LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8	LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0	LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xC8	MATRIX_PRAS9	31:24	LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16	LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8	LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0	LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xCC	MATRIX_PRBS9	31:24					LQOSEN14	M14PR[1:0]	
		23:16	LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8	LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0	LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xD0	MATRIX_PRAS10	31:24	LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16	LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8	LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0	LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0xD4	MATRIX_PRBS10	31:24						LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xD8	MATRIX_PRAS11	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xDC	MATRIX_PRBS11	31:24						LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xE0	MATRIX_PRAS12	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xE4	MATRIX_PRBS12	31:24						LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xE8 ... 0xFF	Reserved									
0x0100	MATRIX_MRCR	31:24								
		23:16								
		15:8		RCB14	RCB13	RCB12	RCB11	RCB10	RCB9	RCB8
		7:0	RCB7	RCB6	RCB5	RCB4	RCB3	RCB2	RCB1	RCB0
0x0104 ... 0x014F	Reserved									
0x0150	MATRIX_MEIER	31:24								
		23:16								
		15:8		MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
		7:0	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
0x0154	MATRIX_MEIDR	31:24								
		23:16								
		15:8		MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
		7:0	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
0x0158	MATRIX_MEIMR	31:24								
		23:16								
		15:8		MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
		7:0	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
0x015C	MATRIX_MESR	31:24								
		23:16								
		15:8		MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
		7:0	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
0x0160	MATRIX_MEAR0	31:24	ERRADD[31:24]							
		23:16	ERRADD[23:16]							
		15:8	ERRADD[15:8]							
		7:0	ERRADD[7:0]							
...										
0x0198	MATRIX_MEAR14	31:24	ERRADD[31:24]							
		23:16	ERRADD[23:16]							
		15:8	ERRADD[15:8]							
		7:0	ERRADD[7:0]							
0x019C ... 0x01E3	Reserved									

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x01E4	MATRIX_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0	CFGFRZ							WPEN
0x01E8	MATRIX_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

Downloaded from Arrow.com.

24.10.1 MATRIX Master Configuration Register x

Name: MATRIX_MCFGx
Offset: 0x00 + x*0x04 [x=0..14]
Reset: 0x00000004
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						ULBT[2:0]		
Access						R/W	R/W	R/W
Reset						1	0	0

Bits 2:0 – ULBT[2:0] Undefined Length Burst Type

Value	Name	Description
0	UNLIMITED	Unlimited Length Burst—No predicted end of burst is generated, therefore INCR bursts coming from this master can only be broken if the Slave Slot Cycle Limit is reached. If the Slot Cycle Limit is not reached, the burst is normally completed by the master, at the latest, on the next system bus 1 Kbyte address boundary, allowing up to 256-beat word bursts or 128-beat double-word bursts. This value should not be used in the very particular case of a master capable of performing back-to-back undefined length bursts on a single slave, since this could indefinitely freeze the slave arbitration and thus prevent another master from accessing this slave.
1	SINGLE	Single Access—The undefined length burst is treated as a succession of single accesses, allowing re-arbitration at each beat of the INCR burst or bursts sequence.
2	4_BEAT	4-beat Burst—The undefined length burst or bursts sequence is split into 4-beat bursts or less, allowing re-arbitration every 4 beats.
3	8_BEAT	8-beat Burst—The undefined length burst or bursts sequence is split into 8-beat bursts or less, allowing re-arbitration every 8 beats.
4	16_BEAT	16-beat Burst—The undefined length burst or bursts sequence is split into 16-beat bursts or less, allowing re-arbitration every 16 beats.
5	32_BEAT	32-beat Burst—The undefined length burst or bursts sequence is split into 32-beat bursts or less, allowing re-arbitration every 32 beats.
6	64_BEAT	64-beat Burst—The undefined length burst or bursts sequence is split into 64-beat bursts or less, allowing re-arbitration every 64 beats.
7	128_BEAT	128-beat Burst—The undefined length burst or bursts sequence is split into 128-beat bursts or less, allowing re-arbitration every 128 beats. Unless duly needed, the ULBT should be left at its default 0 value for power saving.

Downloaded from Arrow.com.

24.10.2 MATRIX Slave Configuration Register x

Name: MATRIX_SCFGx
Offset: 0x40 + x*0x04 [x=0..12]
Reset: 0x000001FF
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
								SLOT_CYCLE[8]
Access								R/W
Reset								1

Bit	7	6	5	4	3	2	1	0
	SLOT_CYCLE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 21:18 – FIXED_DEFMSTR[3:0] Fixed Default Master
This is the number of the Default Master for this slave. Only used if DEFMSTR_TYPE is 2. Specifying the number of a master which is not connected to the selected slave is equivalent to setting DEFMSTR_TYPE to 0.

Bits 17:16 – DEFMSTR_TYPE[1:0] Default Master Type

Value	Name	Description
0	NONE	No Default Master—At the end of the current slave access, if no other master request is pending, the slave is disconnected from all masters. This results in a one clock cycle latency for the first access of a burst transfer or for a single access.
1	LAST	Last Default Master—At the end of the current slave access, if no other master request is pending, the slave stays connected to the last master having accessed it. This results in not having one clock cycle latency when the last master tries to access the slave again.
2	FIXED	Fixed Default Master—At the end of the current slave access, if no other master request is pending, the slave connects to the fixed master the number that has been written in the FIXED_DEFMSTR field. This results in not having one clock cycle latency when the fixed master tries to access the slave again.

Bits 8:0 – SLOT_CYCLE[8:0] Maximum Bus Grant Duration for Masters
When SLOT_CYCLE system bus clock cycles have elapsed since the last arbitration, a new arbitration takes place to let another master access this slave. If another master is requesting the slave bus, then the current master burst is broken.
If SLOT_CYCLE = 0, the Slot Cycle Limit feature is disabled and bursts always complete unless broken according to the ULBT.

Downloaded from Arrow.com.

This limit has been placed in order to enforce arbitration so as to meet potential latency constraints of masters waiting for slave access.
This limit must not be too small. Unreasonably small values break every burst and the MATRIX arbitrates without performing any data transfer. The default maximum value is usually an optimal conservative choice.
In most cases, this feature is not needed and should be disabled for power saving.
See section Slot Cycle Limit Arbitration for details.

Downloaded from Arrow.com.

24.10.3 MATRIX Priority Register A For Slaves x

Name: MATRIX_PRASx
Offset: 0x80 + x*0x08 [x=0..12]
Reset: –
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Table 24-5. MATRIX_PRASx Register Reset Values

Registers	Reset Values
PRAS0, PRAS3, PRAS7, PRAS8, PRAS9	0x00007722
PRAS1, PRAS2, PRAS10, PRAS11, PRAS12	0x00000022
PRAS4	0x00000702
PRAS5	0x00007020
PRAS6	0x00000000

Bit	31	30	29	28	27	26	25	24
		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		–	–	–		–	–	–

Bit	23	22	21	20	19	18	17	16
		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		–	–	–		–	–	–

Bit	15	14	13	12	11	10	9	8
		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		–	–	–		–	–	–

Bit	7	6	5	4	3	2	1	0
		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		–	–	–		–	–	–

Bits 2, 6, 10, 14, 18, 22, 26, 30 – LQOSENx Latency Quality of Service Enable for Master x

Value	Description
0	Disables propagation of Latency Quality of Service from the Master x to the Slave and apply MxPR priority for all access from Master x to the Slave.
1	Enables the propagation of Latency Quality of Service from the Master x to the Slave if supported by the Master x.

Bits 0:1, 4:5, 8:9, 12:13, 16:17, 20:21, 24:25, 28:29 – MxPR Master x Priority
Fixed priority of Master x for accessing the selected slave. The higher the number, the higher the priority.
All the masters programmed with the same MxPR value for the slave make up a priority pool.
Round-robin arbitration is used in the lowest (MxPR = 0) and highest (MxPR = 3) priority pools.
Fixed priority is used in intermediate priority pools (MxPR = 1) and (MxPR = 2).
See section Arbitration Priority Scheme for details.
If LQOSENx bit is cleared, then this priority value is used as it for arbitration and downward propagation to the slave.
If LQOSENx bit is set, then this priority acts as the upper limit for the Latency Quality of Service from Master x.
For masters other than the CPU, the usual value of this field should be 0x0 if LQOSENx bit is cleared, and 0x1 if LQOSENx bit is set. For the CPU master, the usual value of this field should be 0x2.

Downloaded from Arrow.com.

24.10.4 MATRIX Priority Register B For Slaves x

Name: MATRIX_PRBSx
Offset: 0x84 + x*0x08 [x=0..12]
Reset: –
Property: Read/Write

Table 24-6. MATRIX_PRBSx Register Reset Values

Registers	Reset Values
PRBS0, PRBS3, PRBS6, PRBS9	0x07000000
PRBS1, PRBS2, PRBS4, PRBS5, PRBS7, PRBS8, PRBS10, PRBS11, PRBS12	0x00000000

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
						LQOSEN14	M14PR[1:0]	
Access						R/W	R/W	R/W
Reset						–	–	–

Bit	23	22	21	20	19	18	17	16
		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		–	–	–		–	–	–

Bit	15	14	13	12	11	10	9	8
		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		–	–	–		–	–	–

Bit	7	6	5	4	3	2	1	0
		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		–	–	–		–	–	–

Bits 2, 6, 10, 14, 18, 22, 26 – LQOSENx Latency Quality of Service Enable for Master x

Value	Description
0	Disables propagation of Latency Quality of Service from the Master x to the Slave and apply MxPR priority for all access from Master x to the Slave.
1	Enables the propagation of Latency Quality of Service from the Master x to the Slave if supported by the Master x.

Bits 0:1, 4:5, 8:9, 12:13, 16:17, 20:21, 24:25 – MxPR Master x Priority
Fixed priority of Master x for accessing the selected slave. The higher the number, the higher the priority.
All the masters programmed with the same MxPR value for the slave make up a priority pool.
Round-robin arbitration is used in the lowest (MxPR = 0) and highest (MxPR = 3) priority pools.
Fixed priority is used in intermediate priority pools (MxPR = 1) and (MxPR = 2).
See section Arbitration Priority Scheme for details.
If LQOSENx bit is cleared, then this priority value is used as it for arbitration and downward propagation to the slave.
If LQOSENx bit is set, then this priority acts as the upper limit for the Latency Quality of Service from Master x.
For masters other than the CPU, the usual value of this field should be 0x0 if LQOSENx bit is cleared, and 0x1 if LQOSENx bit is set. For the CPU master, the usual value of this field should be 0x2.

Downloaded from Arrow.com.

24.10.5 MATRIX Master Remap Control Register

Name: MATRIX_MRCR
Offset: 0x0100
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		RCB14	RCB13	RCB12	RCB11	RCB10	RCB9	RCB8
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RCB7	RCB6	RCB5	RCB4	RCB3	RCB2	RCB1	RCB0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 – RCBx Remap Command Bit for Master x

Value	Description
0	Disables remapped address decoding for the selected master.
1	Enables remapped address decoding for the selected master.

Downloaded from Arrow.com.

24.10.6 MATRIX Master Error Interrupt Enable Register

Name: MATRIX_MEIER
Offset: 0x0150
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 – MERRx Master x Access Error

Value	Description
0	No effect.
1	Enables Master x Access Error interrupt source.

Downloaded from Arrow.com.

24.10.7 MATRIX Master Error Interrupt Disable Register

Name: MATRIX_MEIDR
Offset: 0x0154
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 – MERRx Master x Access Error

Value	Description
0	No effect.
1	Disables Master x Access Error interrupt source.

Downloaded from Arrow.com.

24.10.8 MATRIX Master Error Interrupt Mask Register

Name: MATRIX_MEIMR
Offset: 0x0158
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 – MERRx Master x Access Error

Value	Description
0	Master x Access Error does not trigger any interrupt.
1	Master x Access Error triggers the MATRIX interrupt line.

Downloaded from Arrow.com.

24.10.9 MATRIX Master Error Status Register

Name: MATRIX_MESR
Offset: 0x015C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 – MERRx Master x Access Error

Value	Description
0	No Master Access Error has occurred since the last read of the MATRIX_MESR.
1	At least one Master Access Error has occurred since the last read of the MATRIX_MESR.

Downloaded from Arrow.com.

24.10.10 MATRIX Master Error Address Register x

Name: MATRIX_MEARx
Offset: 0x0160 + x*0x04 [x=0..14]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ERRADD[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ERRADD[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ERRADD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ERRADD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ERRADD[31:0] Master Error Address
32 most significant bits of the last access error address

Downloaded from Arrow.com.

24.10.11 MATRIX Write Protection Mode Register

Name: MATRIX_WPMR
Offset: 0x01E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CFGFRZ							WPEN
Access	R/W							R/W
Reset	0							0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x4D4154	PASSWD	Writing any other value in this field aborts the write operation of the WPEN and CFGFRZ bits. Always reads as 0.

Bit 7 – CFGFRZ Configuration Freeze

Value	Description
0	The MATRIX configuration is not frozen.
1	Freezes the MATRIX configuration until hardware reset. The registers that can be protected by the WPEN bit and the Write Protection Mode Register are no longer modifiable.

Bit 0 – WPEN Write Protection Enable

See section Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x4D4154 (“MAT” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x4D4154 (“MAT” in ASCII).

Downloaded from Arrow.com.

24.10.12 MATRIX Write Protection Status Register

Name: MATRIX_WPSR
Offset: 0x01E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last write of the MATRIX_WPMR.
1	A write protection violation has occurred since the last write of the MATRIX_WPMR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

25. Advanced Interrupt Controller (AIC)

25.1 Description

The Advanced Interrupt Controller (AIC) is an 8-level priority, individually maskable, vectored interrupt controller providing handling of up to one hundred and twenty-eight interrupt sources. It is designed to substantially reduce the software and real-time overhead in handling internal and external interrupts.

The AIC drives the nFIQ (fast interrupt request) and the nIRQ (standard interrupt request) inputs of an ARM processor. Inputs of the AIC are either internal peripheral interrupts or external interrupts coming from the product's pins.

The 8-level Priority Controller allows the user to define the priority for each interrupt source, thus permitting higher priority interrupts to be serviced even if a lower priority interrupt is being processed.

Internal interrupt sources can be programmed to be level-sensitive or edge-triggered. External interrupt sources can be programmed to be rising-edge or falling-edge triggered or high-level or low-level sensitive.

The fast-forcing feature redirects any internal or external interrupt source to provide a fast interrupt rather than a normal interrupt.

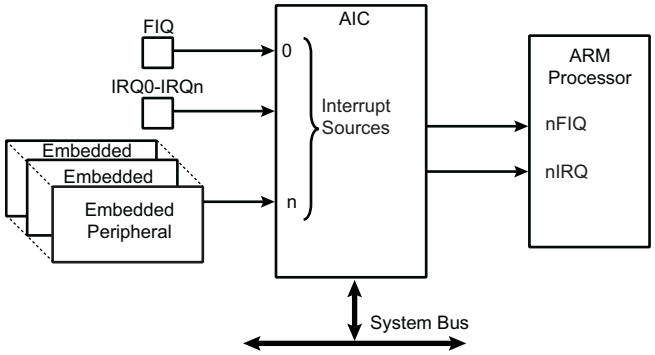
25.2 Embedded Characteristics

- Controls the Interrupt Lines (nIRQ and nFIQ) of an ARM Processor
- 50 Individually Maskable and Vectored Interrupt Sources
  - Source 0 is reserved for the fast interrupt input (FIQ)
  - Source 1 is reserved for system peripheral interrupts
  - Source 2 to Source 49 control up to 126 embedded peripheral interrupts or external interrupts
  - Programmable edge-triggered or level-sensitive internal sources
  - Programmable rising/falling edge-triggered or high/low level-sensitive external sources
- 8-level Priority Controller
  - Drives the normal interrupt of the processor
  - Handles priority of the interrupt sources 1 to 49
  - Higher priority interrupts can be served during service of lower priority interrupt
- Vectoring
  - Optimizes interrupt service routine branch and execution
  - One 32-bit vector register for all interrupt sources
  - Interrupt vector register reads the corresponding current interrupt vector or the current interrupt number
- Protect Mode
  - Easy debugging by preventing automatic operations when protect models are enabled
- Fast Forcing
  - Permits redirecting any normal interrupt source to the fast interrupt of the processor
- General Interrupt Mask
  - Provides processor synchronization on events without triggering an interrupt
- Register Write Protection

Downloaded from Arrow.com.

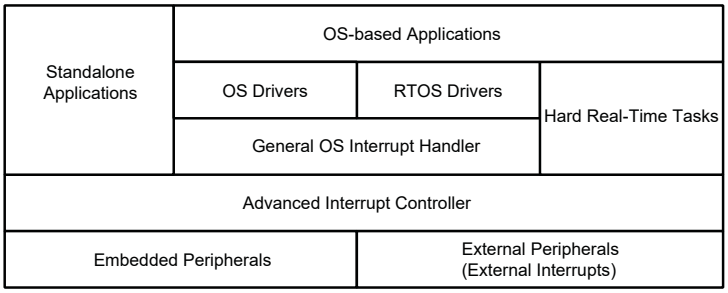
25.3 Block Diagram

Figure 25-1. Block Diagram



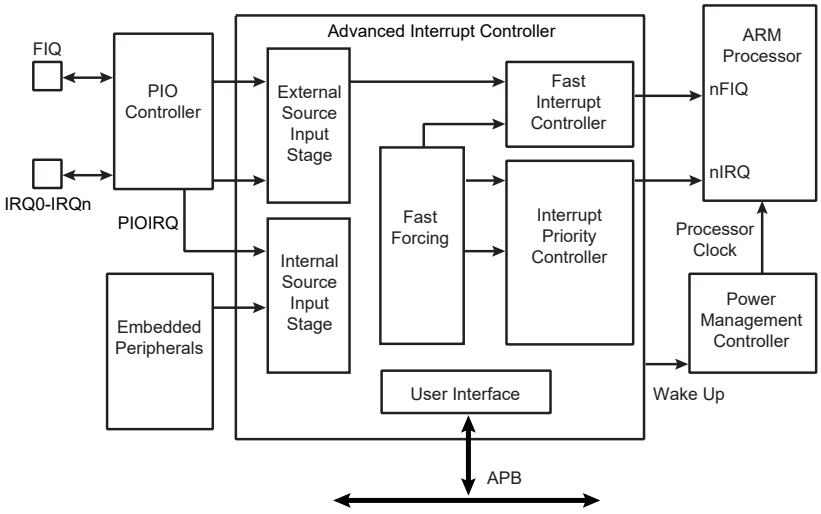
25.4 Application Block Diagram

Figure 25-2. Description of the Application Block



25.5 AIC Detailed Block Diagram

Figure 25-3. AIC Detailed Block Diagram



Downloaded from Arrow.com.

25.6 I/O Line Description

Table 25-1. I/O Line Description

Pin Name	Pin Description	Type
FIQ	Fast Interrupt	Input
IRQ0–IRQn	Interrupt 0–Interrupt n	Input

25.7 Product Dependencies

25.7.1 I/O Lines

The interrupt signals FIQ and IRQ0 to IRQn are normally multiplexed through the PIO controllers. Depending on the features of the PIO controller used in the product, the pins must be programmed in accordance with their assigned interrupt functions. This is not applicable when the PIO controller used in the product is transparent on the input path.

25.7.2 Power Management

The AIC is continuously clocked. The Power Management Controller has no effect on the AIC behavior.

The assertion of the AIC outputs, either nIRQ or nFIQ, wakes up the ARM processor while it is in Idle mode. The General Interrupt Mask feature enables the AIC to wake up the processor without asserting the interrupt line of the processor, thus providing synchronization of the processor on an event.

25.7.3 Interrupt Sources

FIQ always drives Interrupt Source 0.

The System Controller interrupt drives Interrupt Source 1.

The System Controller interrupt is the result of the OR-wiring of the System Controller interrupt lines. When a System Controller interrupt occurs, the service routine must first distinguish the cause of the interrupt. This is performed by reading successively the status registers of the System Controller peripherals.

Interrupt sources 2 to 49 can either be connected to the interrupt outputs of an embedded user peripheral, or to external interrupt lines. The external interrupt lines can be connected either directly or through the PIO Controller.

PIO controllers are considered as user peripherals in the scope of interrupt handling. Accordingly, the PIO controller interrupt lines are connected to interrupt sources 2 to 49.

The peripheral identification defined at the product level corresponds to the interrupt source number (as well as the bit number controlling the clock of the peripheral). Consequently, to simplify the description of the functional operations and the user interface, the interrupt sources are named FIQ, SYS, and PID2 to PID49.

25.8 Functional Description

25.8.1 Interrupt Source Control

25.8.1.1 Interrupt Source Mode

The AIC independently programs each interrupt source. The SRCTYPE field of the Source Mode register (AIC_SMR) selects the interrupt condition of the interrupt source selected by the INTSEL field of the Source Select register (AIC_SSR).

Note: Configuration registers such as AIC_SMR and AIC_SSR return the values corresponding to the interrupt source selected by INTSEL.

The internal interrupt sources wired on the interrupt outputs of the embedded peripherals can be programmed either in Level-Sensitive mode or in Edge-Triggered mode. The active level of the internal interrupts is not important for the user.

Downloaded from Arrow.com.

The external interrupt sources can be programmed either in High Level-Sensitive or Low Level-Sensitive modes, or in Rising Edge-Triggered or Negative Edge-Triggered modes.

25.8.1.2 Interrupt Source Enabling

Each interrupt source, including the FIQ in source 0, can be enabled or disabled by using the command registers Interrupt Enable Command register (AIC_IECR) and Interrupt Disable Command register (AIC_IDCR). The interrupt mask of the selected interrupt source can be read in the Interrupt Mask register (AIC_IMR). A disabled interrupt does not affect servicing of other interrupts.

25.8.1.3 Interrupt Clearing and Setting

All interrupt sources programmed to be edge-triggered (including the FIQ in source 0) can be individually set or cleared by writing respectively the Interrupt Set Command register (AIC_ISCR) and Interrupt Clear Command register (AIC_ICCR). Clearing or setting interrupt sources programmed in Level-Sensitive mode has no effect.

The clear operation is perfunctory, as the software must perform an action to reset the “memorization” circuitry activated when the source is programmed in Edge-Triggered mode. However, the set operation is available for auto-test or software debug purposes. It can also be used to execute an AIC-implementation of a software interrupt.

The AIC features an automatic clear of the current interrupt when AIC_IVR (Interrupt Vector register) is read. Only the interrupt source being detected by the AIC as the current interrupt is affected by this operation. (See the section “Priority Controller”.) The automatic clear reduces the operations required by the interrupt service routine entry code to read AIC_IVR. Note that the automatic interrupt clear is disabled if the interrupt source has the Fast Forcing feature enabled, as it is considered uniquely as an FIQ source. (For further details, see the section ”Fast Forcing”).

The automatic clear of interrupt source 0 is performed when the FIQ Vector register (AIC_FVR) is read.

25.8.1.4 Interrupt Status

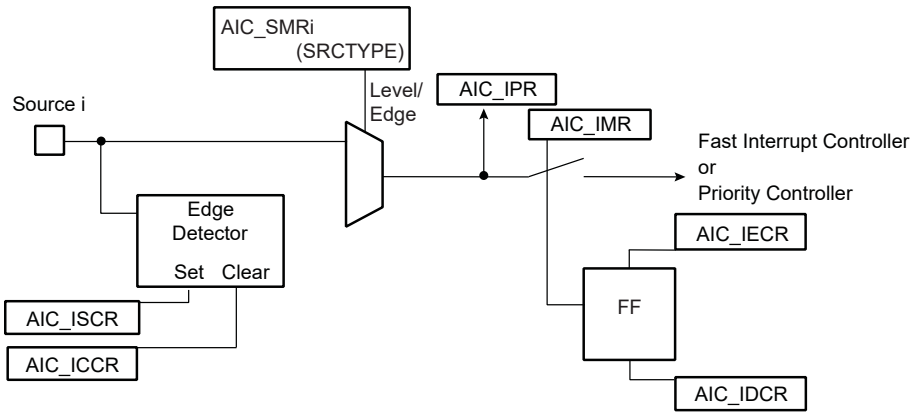
Interrupt Pending registers (AIC_IPR) represent the state of the interrupt lines, whether they are masked or not. AIC_IMR can be used to define the mask of the interrupt lines.

The Interrupt Status register (AIC_ISR) reads the number of the current interrupt (see the section ”Priority Controller”) and the Core Interrupt Status register (AIC_CISR) gives an image of the nIRQ and nFIQ signals driven on the processor.

Each status referred to above can be used to optimize the interrupt handling of the systems.

25.8.1.5 Internal Interrupt Source Input Stage

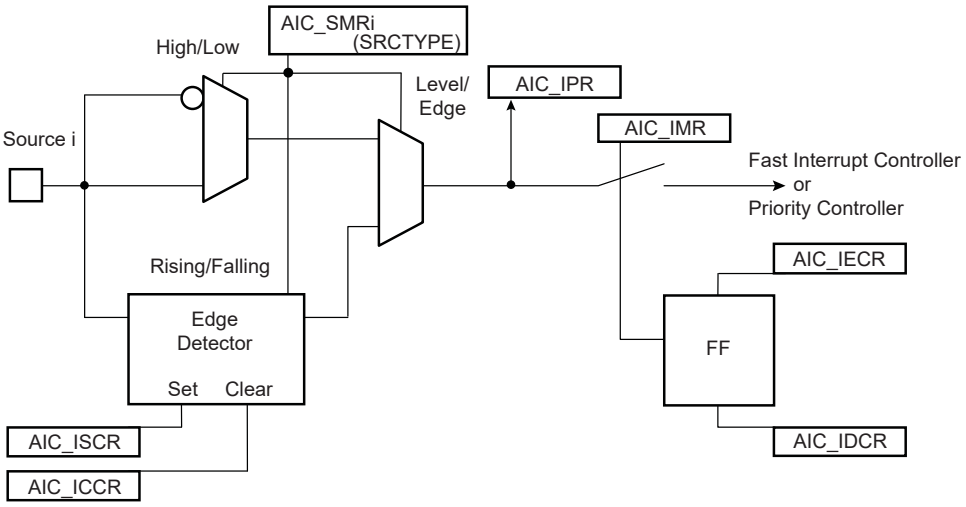
Figure 25-4. Internal Interrupt Source Input Stage



Downloaded from Arrow.com.

25.8.1.6 External Interrupt Source Input Stage

Figure 25-5. External Interrupt Source Input Stage



25.8.2 Interrupt Latencies

Global interrupt latencies depend on several parameters, including:

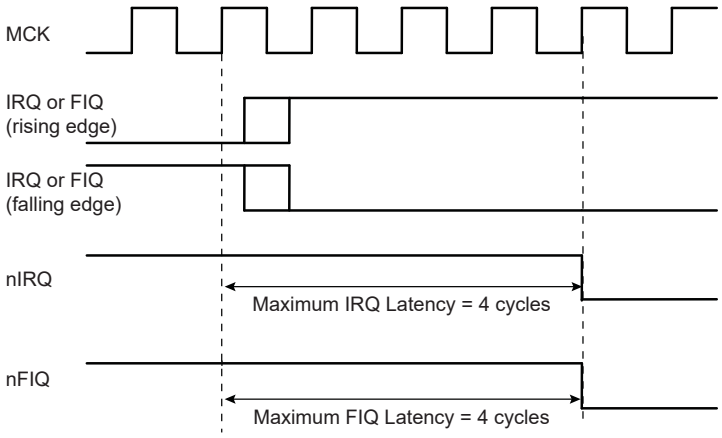
- The time the software masks the interrupts
- Occurrence, either at the processor level or at the AIC level
- The execution time of the instruction in progress when the interrupt occurs
- The treatment of higher priority interrupts and the resynchronization of the hardware signals

This section addresses hardware resynchronizations only. It gives details about the latency times between the events on an external interrupt leading to a valid interrupt (edge or level) or the assertion of an internal interrupt source and the assertion of the nIRQ or nFIQ line on the processor. The resynchronization time depends on the programming of the interrupt source and on its type (internal or external). For the standard interrupt, resynchronization times are given assuming there is no higher priority in progress.

The PIO Controller multiplexing has no effect on the interrupt latencies of the external interrupt sources.

25.8.2.1 External Interrupt Edge Triggered Source

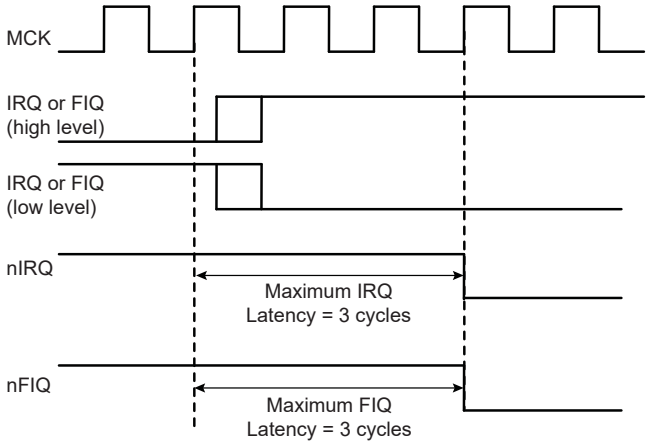
Figure 25-6. External Interrupt Edge Triggered Source



Downloaded from Arrow.com.

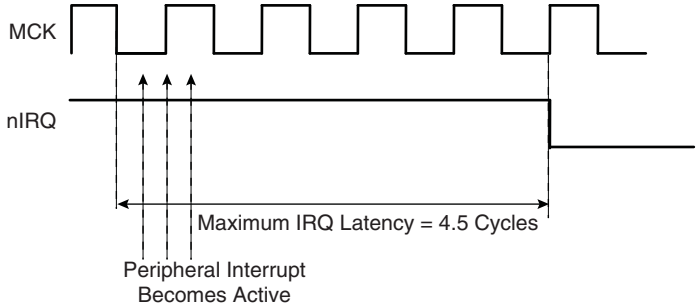
25.8.2.2 External Interrupt Level Sensitive Source

Figure 25-7. External Interrupt Level Sensitive Source



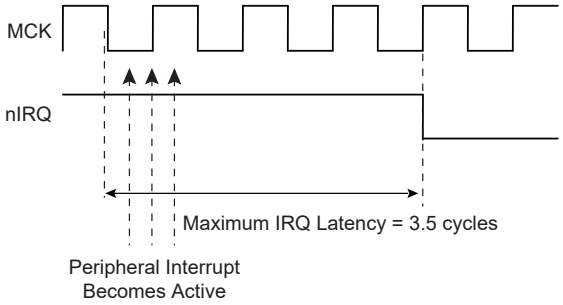
25.8.2.3 Internal Interrupt Edge Triggered Source

Figure 25-8. Internal Interrupt Edge Triggered Source



25.8.2.4 Internal Interrupt Level Sensitive Source

Figure 25-9. Internal Interrupt Level Sensitive Source



25.8.3 Normal Interrupt

25.8.3.1 Priority Controller

An 8-level priority controller drives the nIRQ line of the processor, depending on the interrupt conditions occurring on the interrupt sources 1 to 49 (except for those programmed in Fast Forcing).

Each interrupt source has a programmable priority level of 7 to 0, which is user-definable by writing AIC_SMR.PRIOR. Level 7 is the highest priority and level 0 the lowest.

As soon as an interrupt condition occurs, as defined by AIC_SMR.SRCTYPE, the nIRQ line is asserted. As a new interrupt condition might have happened on other interrupt sources since the nIRQ has been asserted, the priority controller determines the current interrupt at the time AIC_IVR is read. The read of AIC_IVR is the entry point of the interrupt handling which allows the AIC to consider that the interrupt has been taken into account by the software.

Downloaded from Arrow.com.

The current priority level is defined as the priority level of the current interrupt.

If several interrupt sources of equal priority are pending and enabled when AIC_IVR is read, the interrupt with the lowest interrupt source number is serviced first.

The nIRQ line can be asserted only if an interrupt condition occurs on an interrupt source with a higher priority. If an interrupt condition happens (or is pending) during the interrupt treatment in progress, it is delayed until the software indicates to the AIC the end of the current service by writing AIC_EOICR (End of Interrupt Command register). The write of AIC_EOICR is the exit point of the interrupt handling.

25.8.3.2 Interrupt Nesting

The priority controller utilizes interrupt nesting in order for the high priority interrupt to be handled during the service of lower priority interrupts. This requires the interrupt service routines of the lower interrupts to re-enable the interrupt at the processor level.

When an interrupt of a higher priority happens during an already occurring interrupt service routine, the nIRQ line is re-asserted. If the interrupt is enabled at the core level, the current execution is interrupted and the new interrupt service routine should read AIC_IVR. At this time, the current interrupt number and its priority level are pushed into an embedded hardware stack, so that they are saved and restored when the higher priority interrupt servicing is finished and AIC_EOICR is written.

The AIC is equipped with an 8-level wide hardware stack in order to support up to eight interrupt nestings to match the eight priority levels.

25.8.3.3 Interrupt Vectoring

The interrupt handler address corresponding to the interrupt source selected by the INTSEL field can be stored in AIC_SVR (Source Vector register). When the processor reads AIC_IVR, the value written into AIC_SVR corresponding to the current interrupt is returned. Optionally, the AIC_IVR register can return the current interrupt number instead. This can be defined separately for each interrupt source using the AIC SVR Return Enable Register (AIC_SVRRER) and AIC SVR Return Disable Register (AIC_SVRRDR).

This feature offers a way to branch in one single instruction to the handler corresponding to the current interrupt, as AIC_IVR is mapped at the absolute address 0xFFFFF100 and thus accessible from the ARM interrupt vector at address 0x00000018 through the following instruction:

```
LDR PC,[PC,# -&F20]
```

When the processor executes this instruction, it loads the read value in AIC_IVR in its program counter, thus branching the execution on the correct interrupt handler.

25.8.3.4 Interrupt Handlers

This section gives an overview of the fast interrupt handling sequence when using the AIC. It is assumed that the programmer understands the architecture of the ARM processor, and especially the Processor Interrupt modes and the associated status bits.

It is assumed that:

1. The AIC has been programmed, AIC_SVR registers are loaded with corresponding interrupt service routine addresses and interrupts are enabled.
2. The instruction at the ARM interrupt exception vector address is required to work with the vectoring

```
LDR PC, [PC, # -&F20]
```

When nIRQ is asserted, if the bit “I” of CPSR is 0, the sequence is as follows:

1. The CPSR is stored in SPSR_irq, the current value of the Program Counter is loaded in the Interrupt link register (R14_irq) and the Program Counter (R15) is loaded with 0x18. In the following cycle during fetch at address 0x1C, the ARM core adjusts R14_irq, decrementing it by four.
2. The ARM core enters Interrupt mode, if it has not already done so.
3. When the instruction loaded at address 0x18 is executed, the program counter is loaded with the value read in AIC_IVR. Reading AIC_IVR has the following effects:
   – Sets the current interrupt to be the pending and enabled interrupt with the highest priority. The current level is the priority level of the current interrupt.

Downloaded from Arrow.com.

- De-asserts the nIRQ line on the processor. Even if vectoring is not used, AIC_IVR must be read in order to de-assert nIRQ.
- Automatically clears the interrupt, if it has been programmed to be edge-triggered.
- Pushes the current level and the current interrupt number on to the stack.
- Returns the value written in AIC_SVR corresponding to the current interrupt.

4. The previous step has the effect of branching to the corresponding interrupt service routine. This should start by saving the link register (R14_irq) and SPSR_IRQ. The link register must be decremented by four when it is saved if it is to be restored directly into the program counter at the end of the interrupt. For example, the instruction `SUB PC, LR, #4` may be used.
5. Further interrupts can then be unmasked by clearing the "I" bit in CPSR, allowing re-assertion of the nIRQ to be taken into account by the core. This can happen if an interrupt with a higher priority than the current interrupt occurs.
6. The interrupt handler can then proceed as required, saving the registers that will be used and restoring them at the end. During this phase, an interrupt of higher priority than the current level will restart the sequence from step 1.
   Note: If the interrupt is programmed to be level-sensitive, the source of the interrupt must be cleared during this phase.
7. The "I" bit in CPSR must be set in order to mask interrupts before exiting to ensure that the interrupt is completed in an orderly manner.
8. AIC_EOICR must be written in order to indicate to the AIC that the current interrupt is finished. This causes the current level to be popped from the stack, restoring the previous current level if one exists on the stack. If another interrupt is pending, with lower or equal priority than the old current level but with higher priority than the new current level, the nIRQ line is re-asserted, but the interrupt sequence does not immediately start because the "I" bit is set in the core. SPSR_irq is restored. Finally, the saved value of the link register is restored directly into the PC. This has the effect of returning from the interrupt to whatever was being executed before, and of loading the CPSR with the stored SPSR, masking or unmasking the interrupts depending on the state saved in SPSR_irq.
   Note: The "I" bit in SPSR is significant. If it is set, it indicates that the ARM core was on the verge of masking an interrupt when the mask instruction was interrupted. Hence, when SPSR is restored, the mask instruction is completed (interrupt is masked).

25.8.4 Fast Interrupt

25.8.4.1 Fast Interrupt Source

Interrupt source 0 is the only source which can raise a fast interrupt request to the processor except if fast forcing is used. Interrupt source 0 is generally connected to a FIQ pin of the product, either directly or through a PIO Controller.

25.8.4.2 Fast Interrupt Control

The fast interrupt logic of the AIC has no priority controller. The mode of interrupt source 0 is programmed with AIC_SMR and INTSEL = 0; the PRIOR field of this register is not used even if it reads what has been written. AIC_SMR.SRCTYPE enables programming the fast interrupt source to be rising-edge triggered or falling-edge triggered or high-level sensitive or low-level sensitive.

Writing 0x1 in AIC_IECR and AIC_IDCR respectively enables and disables the fast interrupt when INTSEL = 0. Bit 0 of AIC_IMR indicates whether the fast interrupt is enabled or disabled.

25.8.4.3 Fast Interrupt Vectoring

The fast interrupt handler address can be stored through AIC_SVR. The value written into this register when INTSEL = 0 is returned when the processor reads AIC_FVR (FIQ Vector register). This offers a way to branch in one single instruction to the interrupt handler, as AIC_FVR is mapped at the absolute address 0xFFFF F104 and thus accessible from the ARM fast interrupt vector at address 0x0000 001C through the following instruction:

```
LDR PC,[PC,# -&F20]
```

When the processor executes this instruction, it loads the value read in AIC_FVR in its program counter, thus branching the execution on the fast interrupt handler. It also automatically performs the clear of the fast interrupt source if it is programmed in Edge-Triggered mode.

Downloaded from Arrow.com.

25.8.4.4 Fast Interrupt Handlers

This section gives an overview of the fast interrupt handling sequence when using the AIC. It is assumed that the programmer understands the architecture of the ARM processor, and especially the Processor Interrupt modes and associated status bits.

Assuming that:

1. The AIC has been programmed, AIC_SVR is loaded with the fast interrupt service routine address, and interrupt source 0 is enabled.
2. The Instruction at address 0x1C (FIQ exception vector address) is required to vector the fast interrupt: `LDR PC,[PC, # -&F20]`
3. The user does not need nested fast interrupts.

When nFIQ is asserted, if bit “F” of CPSR is 0, the sequence is:

1. The CPSR is stored in SPSR_fiq, the current value of the program counter is loaded in the FIQ link register (R14_FIQ) and the program counter (R15) is loaded with 0x1C. In the following cycle, during fetch at address 0x20, the ARM core adjusts R14_fiq, decrementing it by four.
2. The ARM core enters FIQ mode.
3. When the instruction loaded at address 0x1C is executed, the program counter is loaded with the value read in AIC_FVR. Reading AIC_FVR has the effect of automatically clearing the fast interrupt, if it has been programmed to be edge-triggered. In this case only, it de-asserts the nFIQ line on the processor.

```
FIQ_Handler_Branch
   mov r14, pc
   bx r0
```

4. The previous step enables branching to the corresponding interrupt service routine. It is not necessary to save the link register R14_fiq and SPSR_fiq if nested fast interrupts are not needed.
5. The Interrupt Handler can then proceed as required. It is not necessary to save registers R8 to R13 because the FIQ mode has its own dedicated registers and registers R8 to R13 are banked. The other registers, R0 to R7, must be saved before being used, and restored at the end (before the next step).
   Note: If the fast interrupt is programmed to be level-sensitive, the source of the interrupt must be cleared during this phase in order to de-assert interrupt source 0.
6. Finally, Link register R14_fiq is restored into the PC after decrementing it by four (with instruction `SUB PC, LR, #4` for example). This has the effect of returning from the interrupt to whatever was being executed before, loading the CPSR with the SPSR and masking or unmasking the fast interrupt depending on the state saved in the SPSR.
   Note: The “F” bit in SPSR is significant. If it is set, it indicates that the ARM core was just about to mask FIQ interrupts when the mask instruction was interrupted. Hence, when the SPSR is restored, the interrupted instruction is completed (FIQ is masked).

Another way to handle the fast interrupt is to map the interrupt service routine at the address of the ARM vector 0x1C. This method does not use vectoring, so that reading AIC_FVR must be performed at the very beginning of the handler operation. However, this method saves the execution of a branch instruction.

25.8.4.5 Fast Forcing

The Fast Forcing feature of the advanced interrupt controller provides redirection of any normal Interrupt source on the fast interrupt controller.

Fast Forcing is enabled or disabled by writing to the Fast Forcing Enable register (AIC_FFER) and the Fast Forcing Disable register (AIC_FFDR). Writing to these registers results in an update of the Fast Forcing Status register (AIC_FFSR) that controls the feature for each internal or external interrupt source.

When Fast Forcing is disabled, the interrupt sources are handled as described in the previous sections.

When Fast Forcing is enabled, the edge/level programming and, in certain cases, edge detection of the interrupt source is still active but the source cannot trigger a normal interrupt to the processor and is not seen by the priority handler.

If the interrupt source is programmed in Level-Sensitive mode and an active level is sampled, Fast Forcing results in the assertion of the nFIQ line to the core.

Downloaded from Arrow.com.

If the interrupt source is programmed in Edge-Triggered mode and an active edge is detected, Fast Forcing results in the assertion of the nFIQ line to the core.

The Fast Forcing feature does not affect the Source 0 pending bit in AIC_IPR.

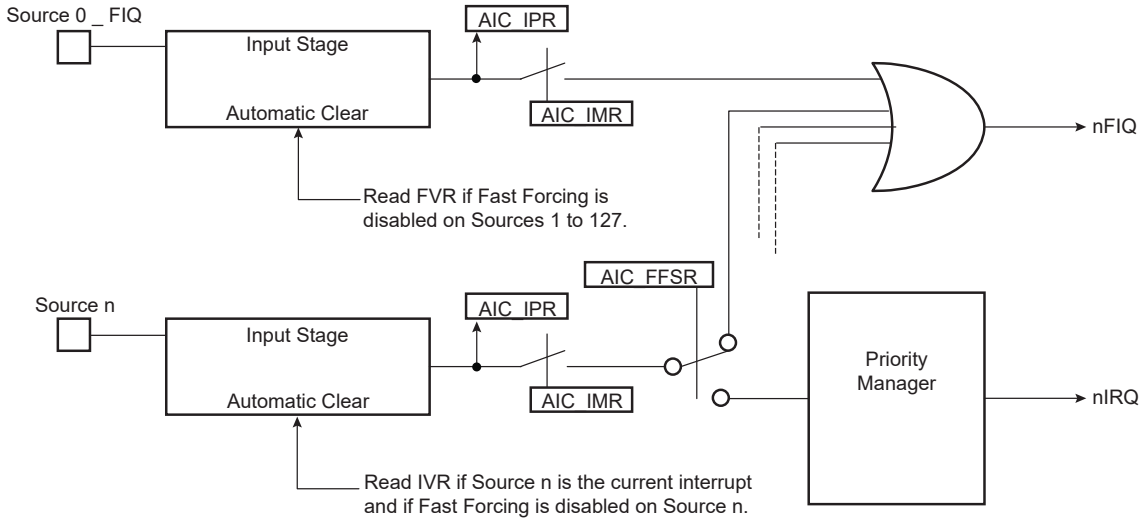
The FIQ Vector register (AIC_FVR) reads the contents of the Source Vector register (AIC_SVR), whatever the source of the fast interrupt may be. The read of the FVR does not clear Source 0 when the fast forcing feature is used and the interrupt source should be cleared by writing to AIC_ICCR.

All enabled and pending interrupt sources that have the fast forcing feature enabled and that are programmed in Edge-Triggered mode must be cleared by writing to the Interrupt Clear Command register. In doing so, they are cleared independently and thus lost interrupts are prevented.

The read of AIC_IVR does not clear the source that has the fast forcing feature enabled.

Source 0, reserved to the fast interrupt, continues operating normally and becomes one of the Fast Interrupt sources.

Figure 25-10. Fast Forcing



25.8.5 Protect Mode

The Protect mode is used to read the Interrupt Vector register without performing the associated automatic operations. This is necessary when working with a debug system. When a debugger, working either with a Debug Monitor or the ARM processor's ICE, stops the applications and updates the opened windows, it might read the AIC User Interface and thus the IVR. This has adverse consequences:

- If an enabled interrupt with a higher priority than the current one is pending, it is stacked.
- If there is no enabled pending interrupt, the spurious vector is returned.

In either case, an End of Interrupt command is necessary to acknowledge and restore the context of the AIC. This operation is generally not performed by the debug system, as the debug system would become strongly intrusive and cause the application to enter an undesired state.

This is avoided by using the Protect mode. Writing PROT in the Debug Control register (AIC_DCR) at 0x1 enables the Protect mode.

When the Protect mode is enabled, the AIC performs interrupt stacking only when a write access is performed on AIC_IVR. Therefore, the Interrupt Service Routines must write (arbitrary data) to AIC_IVR just after reading it. The new context of the AIC, including the value of AIC_ISR, is updated with the current interrupt only when AIC_IVR is written.

An AIC_IVR read on its own (e.g., by a debugger) modifies neither the AIC context nor AIC_ISR. Extra AIC_IVR reads perform the same operations. However, it is recommended to not stop the processor between the read and the write of AIC_IVR of the interrupt service routine to make sure the debugger does not modify the AIC context.

To summarize, in normal operating mode, the read of AIC_IVR performs the following operations within the AIC:

Downloaded from Arrow.com.

1. Calculates active interrupt (higher than current or spurious).
2. Determines and returns the vector of the active interrupt.
3. Memorizes the interrupt.
4. Pushes the current priority level onto the internal stack.
5. Acknowledges the interrupt.

However, while the Protect mode is activated, only operations 1 to 3 are performed when AIC_IVR is read. Operations 4 and 5 are only performed by the AIC when AIC_IVR is written.

Software that has been written and debugged using the Protect mode runs correctly in normal mode without modification. However, in normal mode, the AIC_IVR write has no effect and can be removed to optimize the code.

25.8.6 Spurious Interrupt

The AIC features a protection against spurious interrupts. A spurious interrupt is defined as being the assertion of an interrupt source long enough for the AIC to assert the nIRQ, but no longer present when AIC_IVR is read. This is most prone to occur when:

- An external interrupt source is programmed in Level-Sensitive mode and an active level occurs for only a short time.
- An internal interrupt source is programmed in level-sensitive and the output signal of the corresponding embedded peripheral is activated for a short time (as is the case for the watchdog).
- An interrupt occurs just a few cycles before the software begins to mask it, thus resulting in a pulse on the interrupt source.

The AIC detects a spurious interrupt at the time AIC_IVR is read while no enabled interrupt source is pending. When this happens, the AIC returns the value stored by the programmer in the Spurious Vector register (AIC_SPU). The programmer must store the address of a spurious interrupt handler in AIC_SPU as part of the application, to enable an as fast as possible return to the normal execution flow. This handler writes in AIC_EOICR and performs a return from interrupt.

25.8.7 General Interrupt Mask

The AIC features a General Interrupt Mask bit (AIC_DCR.GMSK) to prevent interrupts from reaching the processor. Both the nIRQ and the nFIQ lines are driven to their inactive state if AIC_DCR.GMSK is set. However, this mask does not prevent waking up the processor if it has entered Idle mode. This function facilitates synchronizing the processor on a next event and, as soon as the event occurs, performs subsequent operations without having to handle an interrupt. It is strongly recommended to use this mask with caution.

25.8.8 Register Write Protection

To prevent any single software error from corrupting AIC behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the AIC Write Protection Mode Register (AIC_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the AIC Write Protection Status Register (AIC_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading AIC_WPSR.

The following registers can be write-protected:

- AIC Source Mode Register
- AIC Source Vector Register
- AIC Spurious Interrupt Vector Register
- AIC Debug Control Register

Downloaded from Arrow.com.

25.9 Register Summary

Offset	Name	Bit Pos.								
0x00	AIC_SSR	31:24								
		23:16								
		15:8								
		7:0		INTSEL[6:0]						
0x04	AIC_SMR	31:24								
		23:16								
		15:8								
		7:0		SRCTYPE[1:0]				PRIOR[2:0]		
0x08	AIC_SVR	31:24	VECTOR[31:24]							
		23:16	VECTOR[23:16]							
		15:8	VECTOR[15:8]							
		7:0	VECTOR[7:0]							
0x0C ... 0x0F	Reserved									
0x10	AIC_IVR	31:24	IRQV[31:24]							
		23:16	IRQV[23:16]							
		15:8	IRQV[15:8]							
		7:0	IRQV[7:0]							
0x14	AIC_FVR	31:24	FIQV[31:24]							
		23:16	FIQV[23:16]							
		15:8	FIQV[15:8]							
		7:0	FIQV[7:0]							
0x18	AIC_ISR	31:24								
		23:16								
		15:8								
		7:0		IRQID[6:0]						
0x1C ... 0x1F	Reserved									
0x20	AIC_IPR0	31:24	PID31	PID30	PID29	PID28	PID27	PID26	PID25	PID24
		23:16	PID23	PID22	PID21	PID20	PID19	PID18	PID17	PID16
		15:8	PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8
		7:0	PID7	PID6	PID5	PID4	PID3	PID2	SYS	FIQ
0x24	AIC_IPR1	31:24	PID63	PID62	PID61	PID60	PID59	PID58	PID57	PID56
		23:16	PID55	PID54	PID53	PID52	PID51	PID50	PID49	PID48
		15:8	PID47	PID46	PID45	PID44	PID43	PID42	PID41	PID40
		7:0	PID39	PID38	PID37	PID36	PID35	PID34	PID33	PID32
0x28	AIC_IPR2	31:24	PID95	PID94	PID93	PID92	PID91	PID90	PID89	PID88
		23:16	PID87	PID86	PID85	PID84	PID83	PID82	PID81	PID80
		15:8	PID79	PID78	PID77	PID76	PID75		PID73	PID72
		7:0	PID71	PID70	PID69	PID68	PID67	PID66	PID65	PID64
0x2C	AIC_IPR3	31:24	PID127	PID126	PID125	PID124	PID123	PID122	PID121	PID120
		23:16	PID119	PID118	PID117	PID116	PID115	PID114	PID113	PID112
		15:8	PID111	PID110	PID109	PID108	PID107	PID106	PID105	PID104
		7:0	PID103	PID102	PID101	PID100	PID99	PID98	PID97	PID96
0x30	AIC_IMR	31:24								
		23:16								
		15:8								
		7:0								INTM
0x34	AIC_CISR	31:24								
		23:16								
		15:8								
		7:0							NIRQ	NFIQ

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x38	AIC_EOICR	31:24								
		23:16								
		15:8								
		7:0								ENDIT
0x3C	AIC_SPU	31:24	SIVR[31:24]							
		23:16	SIVR[23:16]							
		15:8	SIVR[15:8]							
		7:0	SIVR[7:0]							
0x40	AIC_IECR	31:24								
		23:16								
		15:8								
		7:0								INTEN
0x44	AIC_IDCR	31:24								
		23:16								
		15:8								
		7:0								INTD
0x48	AIC_ICCR	31:24								
		23:16								
		15:8								
		7:0								INTCLR
0x4C	AIC_ISCR	31:24								
		23:16								
		15:8								
		7:0								INTSET
0x50	AIC_FFER	31:24								
		23:16								
		15:8								
		7:0								FFEN
0x54	AIC_FFDR	31:24								
		23:16								
		15:8								
		7:0								FFDIS
0x58	AIC_FFSR	31:24								
		23:16								
		15:8								
		7:0								FFS
0x5C ... 0x5F	Reserved									
0x60	AIC_SVRRER	31:24								
		23:16								
		15:8								
		7:0								SVRREN
0x64	AIC_SVRRDR	31:24								
		23:16								
		15:8								
		7:0								SVRRDIS
0x68	AIC_SVRRSR	31:24								
		23:16								
		15:8								
		7:0								SVRRS
0x6C	AIC_DCR	31:24								
		23:16								
		15:8								
		7:0							GMSK	PROT
0x70 ... 0xE3	Reserved									

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0xE4	AIC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0xE8	AIC_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

Downloaded from Arrow.com.

25.9.1 AIC Source Select Register

Name: AIC_SSR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		INTSEL[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 6:0 – INTSEL[6:0] Interrupt Line Selection
0–49 = Selects the interrupt line to handle.
See the section ”Interrupt Source Mode”.

Downloaded from Arrow.com.

25.9.2 AIC Source Mode Register

Name: AIC_SMR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AIC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		SRCTYPE[1:0]				PRIOR[2:0]		
Access		R/W	R/W			R/W	R/W	R/W
Reset		0	0			0	0	0

Bits 6:5 – SRCTYPE[1:0] Interrupt Source Type
The active level or edge is not programmable for the internal interrupt source selected by INTSEL.

Value	Name	Description
0	INT_LEVEL_SENSITIVE	High-level sensitive for internal source. Low-level sensitive for external source.
1	EXT_NEGATIVE_EDGE	Negative-edge triggered for external source.
2	EXT_HIGH_LEVEL	High-level sensitive for internal source. High-level sensitive for external source.
3	EXT_POSITIVE_EDGE	Positive-edge triggered for external source.

Bits 2:0 – PRIOR[2:0] Priority Level
Programs the priority level of the source selected by INTSEL except FIQ source (source 0).
The priority level can be between 0 (lowest) and 7 (highest).
The priority level is not used for the FIQ.

Downloaded from Arrow.com.

25.9.3 AIC Source Vector Register

Name: AIC_SVR
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AIC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	VECTOR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	VECTOR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	VECTOR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VECTOR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – VECTOR[31:0] Source Vector
The user may store in this register the address of the corresponding handler for the interrupt source selected by INTSEL.

Downloaded from Arrow.com.

25.9.4 AIC Interrupt Vector Register

Name: AIC_IVR
Offset: 0x10
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	IRQV[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	IRQV[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IRQV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IRQV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – IRQV[31:0] Interrupt Vector Register
The Interrupt Vector Register contains the vector programmed by the user in the Source Vector Register or the interrupt index corresponding to the current interrupt. (See the sections ”AIC SVR Return Enable Register”, ”AIC SVR Return Disable Register” and ”AIC SVR Return Status Register”.)
The Source Vector Register is indexed using the current interrupt number when the Interrupt Vector Register is read.
When there is no current interrupt, the Interrupt Vector Register reads the value stored in AIC_SPU.

Downloaded from Arrow.com.

25.9.5 AIC FIQ Vector Register

Name: AIC_FVR
Offset: 0x14
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	FIQV[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FIQV[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FIQV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FIQV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – FIQV[31:0] FIQ Vector Register
The FIQ Vector Register contains the vector programmed by the user in the Source Vector Register when INTSEL = 0. When there is no fast interrupt, the FIQ Vector Register reads the value stored in AIC_SPU.

Downloaded from Arrow.com.

25.9.6 AIC Interrupt Status Register

Name: AIC_ISR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		IRQID[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 6:0 – IRQID[6:0] Current Interrupt Identifier
The Interrupt Status Register returns the current interrupt source number.

Downloaded from Arrow.com.

25.9.7 AIC Interrupt Pending Register 0

Name: AIC_IPR0
Offset: 0x20
Reset: 0x00000000
Property: Read-only

The reset value of this register depends on the level of the external interrupt source. All other sources are cleared at reset, thus not pending.

Bit	31	30	29	28	27	26	25	24
	PID31	PID30	PID29	PID28	PID27	PID26	PID25	PID24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PID23	PID22	PID21	PID20	PID19	PID18	PID17	PID16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PID7	PID6	PID5	PID4	PID3	PID2	SYS	FIQ
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32 – PIDx Interrupt Pending
PID2...PID31 refer to the identifiers as defined in the Peripheral Identifiers section.

Value	Description
0	The corresponding interrupt is not pending.
1	The corresponding interrupt is pending.

Bit 1 – SYS Interrupt Pending

Value	Description
0	The corresponding interrupt is not pending.
1	The corresponding interrupt is pending.

Bit 0 – FIQ Interrupt Pending

Value	Description
0	The corresponding interrupt is not pending.
1	The corresponding interrupt is pending.

Downloaded from Arrow.com.

25.9.8 AIC Interrupt Pending Register 1

Name: AIC_IPR1
Offset: 0x24
Reset: 0x00000000
Property: Read-only

The reset value of this register depends on the level of the external interrupt source. All other sources are cleared at reset, thus not pending.
PID32...PID63 refer to the identifiers as defined in the Peripheral Identifiers section.

Bit	31	30	29	28	27	26	25	24
	PID63	PID62	PID61	PID60	PID59	PID58	PID57	PID56
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PID55	PID54	PID53	PID52	PID51	PID50	PID49	PID48
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PID47	PID46	PID45	PID44	PID43	PID42	PID41	PID40
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PID39	PID38	PID37	PID36	PID35	PID34	PID33	PID32
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – PIDx Interrupt Pending

Value	Description
0	The corresponding interrupt is not pending.
1	The corresponding interrupt is pending.

Downloaded from Arrow.com.

25.9.9 AIC Interrupt Pending Register 2

Name: AIC_IPR2
Offset: 0x28
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	PID95	PID94	PID93	PID92	PID91	PID90	PID89	PID88
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PID87	PID86	PID85	PID84	PID83	PID82	PID81	PID80
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PID79	PID78	PID77	PID76	PID75		PID73	PID72
Access	R	R	R	R	R		R	R
Reset	0	0	0	0	0		0	0

Bit	7	6	5	4	3	2	1	0
	PID71	PID70	PID69	PID68	PID67	PID66	PID65	PID64
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – PIDx Interrupt Pending

Value	Description
0	The corresponding interrupt is not pending.
1	The corresponding interrupt is pending.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 – PIDx Interrupt Pending

Value	Description
0	The corresponding interrupt is not pending.
1	The corresponding interrupt is pending.

Downloaded from Arrow.com.

25.9.10 AIC Interrupt Pending Register 3

Name: AIC_IPR3
Offset: 0x2C
Reset: 0x00000000
Property: Read-only

The reset value of this register depends on the level of the external interrupt source. All other sources are cleared at reset, thus not pending.
PID96...PID127 bit fields refer to the identifiers as defined in the Peripheral Identifiers section.

Bit	31	30	29	28	27	26	25	24
	PID127	PID126	PID125	PID124	PID123	PID122	PID121	PID120
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PID119	PID118	PID117	PID116	PID115	PID114	PID113	PID112
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PID111	PID110	PID109	PID108	PID107	PID106	PID105	PID104
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PID103	PID102	PID101	PID100	PID99	PID98	PID97	PID96
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – PIDx Interrupt Pending

Value	Description
0	The corresponding interrupt is not pending.
1	The corresponding interrupt is pending.

Downloaded from Arrow.com.

25.9.11 AIC Interrupt Mask Register

Name: AIC_IMR
Offset: 0x30
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								INTM
Access								R
Reset								0

Bit 0 – INTM Interrupt Mask

Value	Description
0	The interrupt source selected by AIC_SSR.INTSEL is disabled.
1	The interrupt source selected by AIC_SSR.INTSEL is enabled.

Downloaded from Arrow.com.

25.9.12 AIC Core Interrupt Status Register

Name: AIC_CISR
Offset: 0x34
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							NIRQ	NFIQ
Access							R	R
Reset							0	0

Bit 1 – NIRQ NIRQ Status

Value	Description
0	nIRQ line is deactivated.
1	nIRQ line is active.

Bit 0 – NFIQ NFIQ Status

Value	Description
0	nFIQ line is deactivated.
1	nFIQ line is active.

Downloaded from Arrow.com.

25.9.13 AIC End of Interrupt Command Register

Name: AIC_EOICR
Offset: 0x38
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ENDIT
Access								W
Reset								–

Bit 0 – ENDIT Interrupt Processing Complete Command
The End of Interrupt Command Register is used by the interrupt routine to indicate that the interrupt treatment is complete. Any value can be written because it is only necessary to make a write to this register location to signal the end of interrupt treatment.

Downloaded from Arrow.com.

25.9.14 AIC Spurious Interrupt Vector Register

Name: AIC_SPU
Offset: 0x3C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AIC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	SIVR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SIVR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SIVR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SIVR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SIVR[31:0] Spurious Interrupt Vector Register
The user may store the address of a spurious interrupt handler in this register. The written value is returned in AIC_IVR in case of a spurious interrupt, or in AIC_FVR in case of a spurious fast interrupt.

Downloaded from Arrow.com.

25.9.15 AIC Interrupt Enable Command Register

Name: AIC_IECR
Offset: 0x40
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								INTEN
Access								W
Reset								–

Bit 0 – INTEN Interrupt Enable

Value	Description
0	No effect.
1	Enables the interrupt source selected by AIC_SSR.INTSEL.

Downloaded from Arrow.com.

25.9.16 AIC Interrupt Disable Command Register

Name: AIC_IDCR
Offset: 0x44
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								INTD
Access								W
Reset								–

Bit 0 – INTD Interrupt Disable

Value	Description
0	No effect.
1	Disables the interrupt source selected by AIC_SSR.INTSEL.

Downloaded from Arrow.com.

25.9.17 AIC Interrupt Clear Command Register

Name: AIC_ICCR
Offset: 0x48
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								INTCLR
Access								W
Reset								–

Bit 0 – INTCLR Interrupt Clear
Clears one the following depending on the setting of AIC_SSR.INTSEL: FIQ, SYS, PID2-PID49

Value	Description
0	No effect.
1	Clears the interrupt source selected by AIC_SSR.INTSEL.

Downloaded from Arrow.com.

25.9.18 AIC Interrupt Set Command Register

Name: AIC_ISCR
Offset: 0x4C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								INTSET
Access								W
Reset								–

Bit 0 – INTSET Interrupt Set

Value	Description
0	No effect.
1	Sets the interrupt source selected by INTSEL.

Downloaded from Arrow.com.

25.9.19 AIC Fast Forcing Enable Register

Name: AIC_FFER
Offset: 0x50
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								FFEN
Access								W
Reset								–

Bit 0 – FFEN Fast Forcing Enable

Value	Description
0	No effect.
1	Enables the fast forcing feature on the interrupt source selected by INTSEL.

Downloaded from Arrow.com.

25.9.20 AIC Fast Forcing Disable Register

Name: AIC_FFDR
Offset: 0x54
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								FFDIS
Access								W
Reset								–

Bit 0 – FFDIS Fast Forcing Disable

Value	Description
0	No effect.
1	Disables the Fast Forcing feature on the interrupt source selected by INTSEL.

Downloaded from Arrow.com.

25.9.21 AIC Fast Forcing Status Register

Name: AIC_FFSR
Offset: 0x58
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								FFS
Access								R
Reset								0

Bit 0 – FFS Fast Forcing Status

Value	Description
0	The Fast Forcing feature is disabled on the interrupt source selected by INTSEL.
1	The Fast Forcing feature is enabled on the interrupt source selected by INTSEL.

Downloaded from Arrow.com.

25.9.22 AIC SVR Return Enable Register

Name: AIC_SVRRER
Offset: 0x60
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								W
Reset								–

Bit 0 – SVRREN SVR Return Enable

Value	Description
0	No effect.
1	IVR register returns the interrupt index for the interrupt source selected by INTSEL.

Downloaded from Arrow.com.

25.9.23 AIC SVR Return Disable Register

Name: AIC_SVRRDR
Offset: 0x64
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								W
Reset								–

Bit 0 – SVRRDIS SVR Return Disable

Value	Description
0	No effect.
1	IVR register returns the corresponding vector programmed in AIC_SVR for the interrupt source selected by INTSEL.

Downloaded from Arrow.com.

25.9.24 AIC SVR Return Status Register

Name: AIC_SVRRSR
Offset: 0x68
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SVRRS
Access								R
Reset								0

Bit 0 – SVRRS SVR Return Status

Value	Description
0	IVR register returns the corresponding vector programmed in AIC_SVR for the interrupt source selected by INTSEL.
1	IVR register returns the interrupt index for the interrupt source selected by INTSEL.

Downloaded from Arrow.com.

25.9.25 AIC Debug Control Register

Name: AIC_DCR
Offset: 0x6C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AIC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							GMSK	PROT
Access							R/W	R/W
Reset							0	0

Bit 1 – GMSK General Interrupt Mask

Value	Description
0	The nIRQ and nFIQ lines are normally controlled by the AIC.
1	The nIRQ and nFIQ lines are tied to their inactive state.

Bit 0 – PROT Protection Mode

Value	Description
0	The Protection mode is disabled.
1	The Protection mode is enabled.

Downloaded from Arrow.com.

25.9.26 AIC Write Protection Mode Register

Name: AIC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x414943	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 0 – WPEN Write Protection Enable
See section "Register Write Protection" for the list of registers that can be protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x414943 ("AIC" in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x414943 ("AIC" in ASCII).

Downloaded from Arrow.com.

25.9.27 AIC Write Protection Status Register

Name: AIC_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of AIC_WPSR.
1	A write protection violation has occurred since the last read of AIC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

26. Slow Clock Controller (SCKC)

26.1 Description

The System Controller embeds a Slow Clock Controller (SCKC). The SCKC selects the slow clock for the RTT and the RTC from one of two sources:

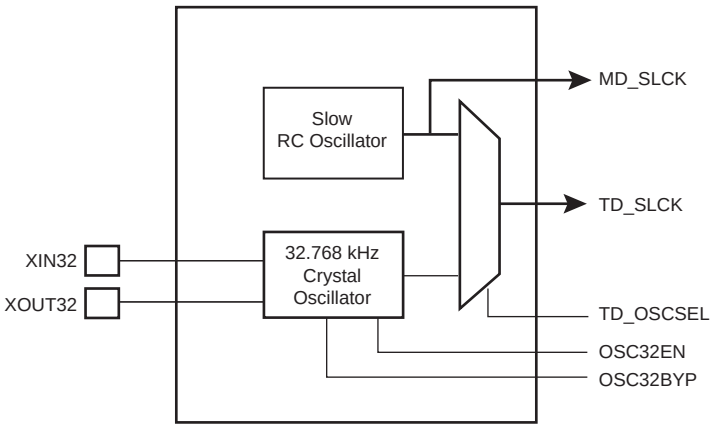
- External 32.768 kHz crystal oscillator
- Embedded 32 kHz (typical) RC oscillator

26.2 Embedded Characteristics

- 32 kHz (typical) RC Oscillator or 32.768 kHz Crystal Oscillator Selector
- VDDBU Powered

26.3 Block Diagram

Figure 26-1. Block Diagram



26.4 Functional Description

The TD_OSCSEL bit located in the Slow Clock Controller Configuration register (SCKC_CR) is in the backup domain and its value is kept while VDDBU is present.

The embedded 32 kHz (typical) Slow RC oscillator is always enabled as soon as VDDBU is established. The Slow Clock Selector command TD_OSCSEL bit selects the slow clock source of the RTT and the RTC.

After the VDDBU power-on reset, the default configuration is TD_OSCSEL = 0.

The programmer controls the slow clock switching by software, so precautions must be taken during the switching phase.

26.4.1 Switching from Embedded 32 kHz RC Oscillator to 32.768 kHz Crystal Oscillator

The sequence to switch from the embedded 32 kHz (typical) RC oscillator to the 32.768 kHz crystal oscillator is the following:

1. Switch the master clock to a source different from slow clock (PLL or Main Oscillator) through the Power Management Controller.
2. Switch from the embedded 32 kHz (typical) RC oscillator to the 32.768 kHz crystal oscillator by writing a 1 to the TD_OSCSEL bit.

Downloaded from Arrow.com.

3. Wait 5 slow clock cycles for internal resynchronization.

26.4.2 Bypassing the 32.768 kHz Crystal Oscillator

The sequence to bypass the 32.768 kHz crystal oscillator is the following:

1. An external clock must be connected on XIN32.
2. Enable the bypass path by writing a 1 to the OSC32BYP bit.
3. Disable the 32.768 kHz crystal oscillator by writing a 0 to the OSC32EN bit.

26.4.3 Switching from 32.768 kHz Crystal Oscillator to Embedded 32 kHz RC Oscillator

The sequence to switch from the 32.768 kHz crystal oscillator to the embedded 32 kHz (typical) RC oscillator is the following:

1. Switch the master clock to a source different from slow clock (PLL or Main Oscillator).
2. Switch from the 32.768 kHz crystal oscillator to the embedded RC oscillator by writing a 0 to the TD_OSCSEL bit.
3. Wait 5 slow clock cycles for internal resynchronization.
4. Disable the 32.768 kHz crystal oscillator by writing a 0 to the OSC32EN bit.

Downloaded from Arrow.com.

26.5 Register Summary

Offset	Name	Bit Pos.								
0x00	SCKC_CR	31:24								TD_OSCSEL
		23:16								
		15:8								
		7:0						OSC32BYP	OSC32EN	

Downloaded from Arrow.com.

26.5.1 Slow Clock Controller Configuration Register

Name: SCKC_CR
Offset: 0x0
Reset: 0x00000001
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
								TD_OSCSEL
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						OSC32BYP	OSC32EN	
Access						R/W	R/W	
Reset						0	0	

Bit 24 – TD_OSCSEL Timing Domain Slow Clock Selector

Value	Description
0 (RC)	Slow clock of the timing domain is driven by the embedded RCFREQ kHz (typical) RC oscillator.
1 (XTAL)	Slow clock of the timing domain is driven by the 32.768 kHz crystal oscillator.

Bit 2 – OSC32BYP 32.768 kHz Crystal Oscillator Bypass

Value	Description
0	32.768 kHz crystal oscillator is not bypassed.
1	32.768 kHz crystal oscillator is bypassed and accepts an external slow clock on XIN32.

Bit 1 – OSC32EN 32.768 kHz Crystal Oscillator

Value	Description
0	32.768 kHz crystal oscillator is disabled.
1	32.768 kHz crystal oscillator is enabled.

Downloaded from Arrow.com.

27. Clock Generator

27.1 Description

The Clock Generator user interface is embedded within the Power Management Controller and is described in the 28.16 Register Summary. However, the Clock Generator registers are named CKGR_.

27.2 Embedded Characteristics

The Clock Generator is made up of:

- Oscillators
  - A low-power 32.768 kHz oscillator supporting crystals, resonators and Bypass mode (referred to as “32.768 KHz crystal oscillator” throughout the document)
  - An embedded always-on, slow RC oscillator generating a typical 32 kHz clock
  - A 12 to 48 MHz oscillator supporting crystals, resonators and Bypass mode (referred to as “main crystal oscillator” throughout the document)
  - A main RC oscillator generating a typical 12 MHz clock
- Two fractional-N PLLs with an input range of 12 to 48 MHz and an internal frequency range of 600 to 1200 MHz

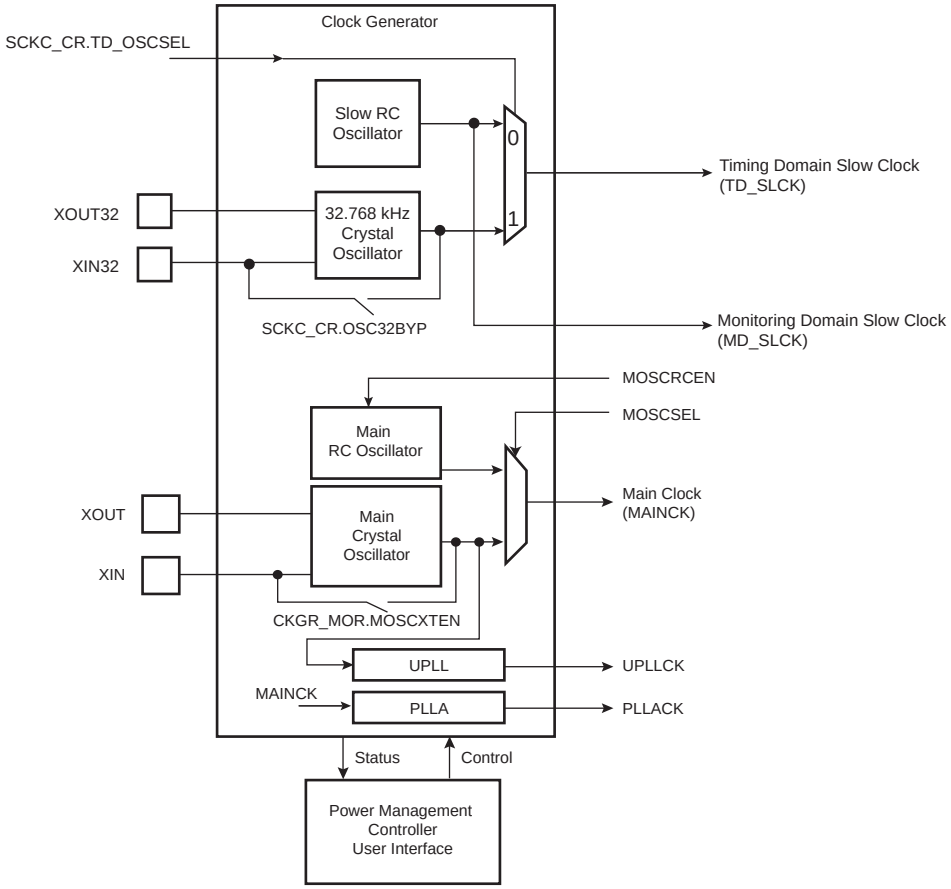
The Clock Generator provides the following clocks:

- MD_SLCK—Monitoring domain slow clock. This clock, sourced from the always-on slow RC oscillator only, is the only permanent clock of the system and feeds safety-critical functions of the device (WDT, RSTC, SCKC, frequency monitors and detectors, PMC startup time counters).
- TD_SLCK—Timing domain slow clock. This clock, sourced from the 32.768 kHz crystal oscillator or the always-on slow RC oscillator, is routed to the RTC and RTT peripherals.
- MAINCK—Output of the main clock oscillator selection. This clock is either the main crystal oscillator or the main RC oscillator.
- PLL Clocks—Outputs of embedded PLLs

Downloaded from Arrow.com.

27.3 Block Diagram

Figure 27-1. Clock Generator Block Diagram



27.4 Slow Clock

The PMC does not control the slow clock generation. The control of the slow clock is performed by the Slow Clock Controller (SCKC) which embeds a slow clock generator that is supplied with the VDDBU power supply. As soon as VDDBU is supplied, both the 32.768 kHz crystal oscillator and the slow RC oscillator are powered, but only the slow RC oscillator is enabled.

MD_SLCK is always generated by the slow RC oscillator.

TD_SLCK is generated either by the 32.768 kHz crystal oscillator or by the slow RC oscillator.

The TD_SLCK source clock selection is made via the TD_OSCSEL bit in the Slow Clock Controller Configuration register (SCKC_CR).

27.4.1 Slow RC Oscillator (32 kHz typical)

The slow RC oscillator is a permanent clock that is the source clock of MD_SLCK and the default source clock of TD_SLCK.

Compared to the 32.768 kHz crystal oscillator, this oscillator offers a faster startup time and is less exposed to the external environment, as it is fully integrated. However, its output frequency is subject to larger variations with supply voltage, temperature and manufacturing process. Therefore, the user must take these variations into account when this oscillator is used as a time base (startup counter, frequency monitor, etc.). Refer to the section “Electrical Characteristics”.

Downloaded from Arrow.com.

27.4.2 32.768 kHz Crystal Oscillator

By default, the 32.768 kHz oscillator is disabled. To use this oscillator, the XIN32 and XOUT32 pins must be connected to a 32.768 kHz crystal or to a ceramic resonator. Refer to the section “Electrical Characteristics” for appropriate loading capacitors selection on XIN32 and XOUT32.

To select the 32.768 kHz crystal oscillator as the source of TD_SLCK, SCKC_CR.TD_OSCSEL must be set. The switch of TD_SLCK source is glitch-free.

Reverting to the slow RC oscillator is only possible by shutting down the VDDBU power supply.

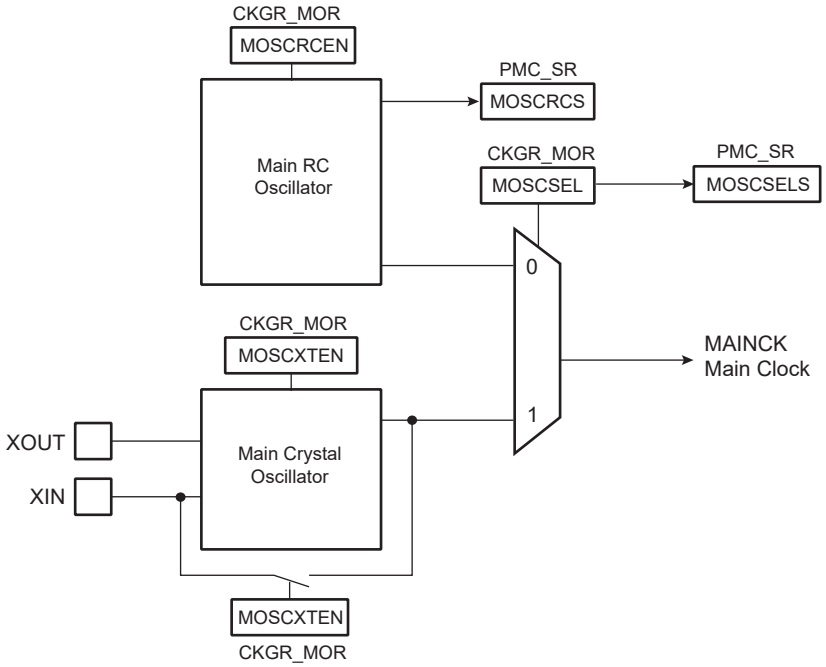
The user can also set the 32.768 kHz crystal oscillator in Bypass mode instead of connecting a crystal. In this case, the user must provide the external clock signal on XIN32. For input characteristics of the XIN32 pin, refer to the section “Electrical Characteristics”. To enter Bypass mode, the OSC32BYP bit of the Slow Clock Controller Configuration register (SCKC_CR) must be set prior to setting SCKC_CR.TD_OSCSEL.

27.5 Main Clock

The main clock (MAINCK) has two sources:

- A main RC oscillator with a fast startup time and that is selected by default to start the system
- A main crystal oscillator with Bypass mode

Figure 27-2. Main Clock (MAINCK) Block Diagram



27.5.1 Main RC Oscillator

After reset, the main RC oscillator is enabled. This oscillator is selected as the source of MAINCK. MAINCK is the default clock selected to start the system.

The main RC oscillator is calibrated in production. For output frequency specifications, refer to the section “Electrical Characteristics”.

The software can disable or enable the main RC oscillator with the MOSCRCEN bit in the Clock Generator Main Oscillator register (CKGR_MOR).

When disabling the main RC oscillator by clearing the CKGR_MOR.MOSCRCEN bit, the PMC_SR.MOSCRCS bit is automatically cleared, indicating that the oscillator is off.

Setting the MOSCRCS bit in the Power Management Controller Interrupt Enable Register (PMC_IER) triggers an interrupt to the processor.

Downloaded from Arrow.com.

27.5.2 Main RC Oscillator Frequency Adjustment

The user can adjust the value of the main RC oscillator frequency by modifying the trimming values set in production by Microchip. This may be used to compensate frequency drifts due to temperature or voltage.

By default, PMC_OCR.SEL12 is cleared, so the main RC oscillator is driven with the factory-defined calibration bits which are programmed during chip production.

In order to adjust the oscillator frequency, PMC_OCR.SEL12 must be set to '1' and a valid value must be configured in PMC_OCR.CAL12.

It is possible to adjust the oscillator frequency while operating from this oscillator.

When reading the PMC_OCR register, the CAL12 field contains the value of the trimming that is currently sent to the main RC oscillator. This means that the read value is either the factory-defined value (PMC_OCR.SEL12='0') or the value written in the register by the user (PMC_OCR.SEL12='1').

At any time, the user can measure the main RC oscillator output frequency by means of the main frequency counter (see Main Frequency Counter). Once the frequency measurement is done, the main RC oscillator calibration field CAL12 can be adjusted accordingly to correct this oscillator output frequency.

27.5.3 Main Crystal Oscillator

After reset, the main crystal oscillator is disabled and is not selected as the source of MAINCK.

The software enables or disables this oscillator in order to reduce power consumption via CKGR_MOR.MOSCXTEN.

When disabling this oscillator by clearing CKGR_MOR.MOSCXTEN, the PMC_SR.MOSCXTS status bit is automatically cleared, indicating the oscillator is off. To activate the Main Crystal Oscillator Bypass mode, see Bypassing the Main Crystal Oscillator.

When enabling this oscillator, the user must initiate the startup time counter. The startup time depends on the characteristics of the external device connected to this oscillator.

When CKGR_MOR.MOSCXTEN and CKGR_MOR.MOSCXTST are written to enable this oscillator, XIN and XOUT are driven by the main crystal oscillator. PMC_SR.MOSCXTS is cleared and the counter starts counting down on MD_SLCK divided by 8 from the CKGR_MOR.MOSCXTST value. Since the CKGR_MOR.MOSCXTST value is coded with 8 bits, the startup time can be programmed up to 2048 MD_SLCK periods, corresponding to about 62 ms when running at 32.768 kHz.

When the startup time counter reaches '0', PMC_SR.MOSCXTS is set, indicating that the oscillator is stabilized. Setting the MOSCXTS bit in the Interrupt Mask Register (PMC_IMR) can trigger an interrupt to the processor.

27.5.4 Main Clock Source Selection

The source of MAINCK can be selected from the following:

- the main RC oscillator
- the main crystal oscillator
- an external clock signal provided on the XIN input (Bypass mode of the main crystal oscillator)

The advantage of the main RC oscillator is its fast startup time. By default, this oscillator is selected to start the system and it must be selected prior to entering ULP mode 1.

The advantage of the main crystal oscillator is its high level of accuracy.

The selection of the oscillator is made by configuring CKGR_MOR.MOSCSEL. The switchover of the MAINCK source is glitch-free, thus the switchover can be performed even if MCK is fed by MAINCK. PMC_SR.MOSCSELS indicates when the switch sequence is done.

Setting PMC_IMR.MOSCSELS triggers an interrupt to the processor.

27.5.5 Bypassing the Main Crystal Oscillator

Prior to bypassing the main crystal oscillator, the XOUT pin must be grounded and the external clock frequency provided on the XIN pin must be stable and within the values specified in the XIN Clock characteristics in the section "Electrical Characteristics". Then the main crystal oscillator must be enabled by setting the CKGR_MOR.MOSCXTEN bit to 1.

Downloaded from Arrow.com.

27.5.6 Main Frequency Counter

The main frequency counter measures the main RC oscillator or the main crystal oscillator against the MD_SLCK and is managed by CKGR_MCFR.

During the measurement period, the main frequency counter increments at the speed of the clock defined by the bit CKGR_MCFR.CCSS.

A measurement is started in the following cases:

- When CKGR_MCFR.RCMEAS is written to ‘1’.
- When the main RC oscillator is selected as the source of MAINCK and when this oscillator is stable (i.e., when the MOSCRCS bit is set)
- When the main crystal oscillator is selected as the source of MAINCK and when this oscillator is stable (i.e., when the MOSCXTS bit is set)
- When MAINCK source selection is modified

The measurement period ends at the 16th falling edge of MD_SLCK, the MAINFRDY bit in CKGR_MCFR is set and the counter stops counting. Its value can be read in the MAINF field of CKGR_MCFR and gives the number of clock cycles during 16 periods of MD_SLCK, so that the frequency of the main RC oscillator or main crystal oscillator can be determined.

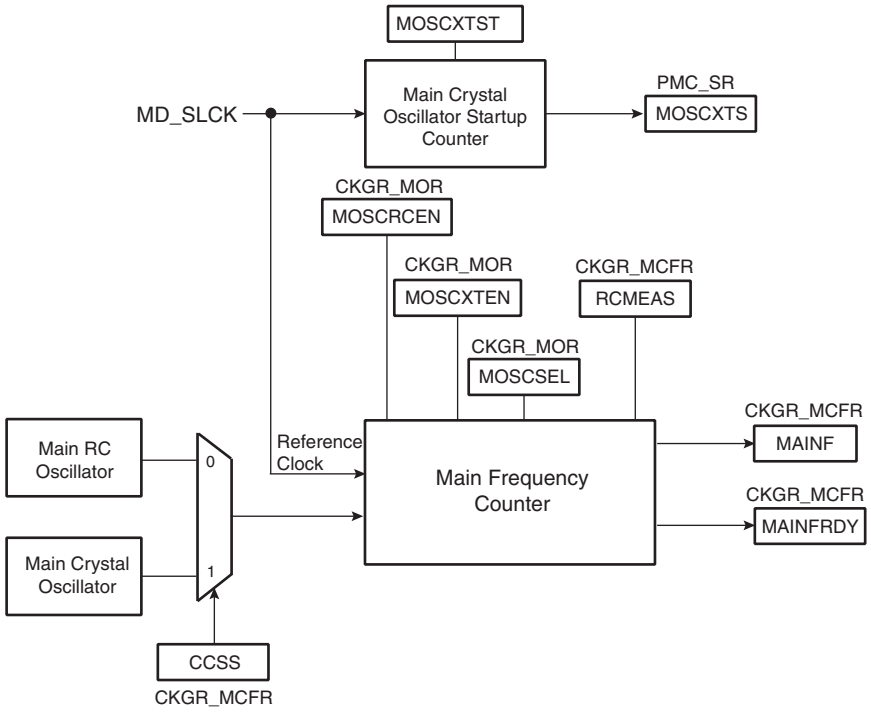
When switching the source of MAINCK from the main RC oscillator to the main crystal oscillator, follow the programming sequence below to ensure that the oscillator is present and that its frequency is valid:

1. Enable the main crystal oscillator by setting CKGR_MOR.MOSCXTEN. Configure the CKGR_MOR. MOSCXTST field with the main crystal oscillator startup time as defined in the section “Electrical Characteristics”.
2. Wait for PMC_SR.MOSCXTS flag to rise, indicating the end of a startup period of the main crystal oscillator.
3. Select the main crystal oscillator as the source clock of the main frequency counter by setting CKGR_MCFR.CCSS.
4. Initiate a frequency measurement by setting CKGR_MCFR.RCMEAS.
5. Read CKGR_MCFR.MAINFRDY until its value equals 1.
6. Read CKGR_MCFR.MAINF and compute the value of the main crystal frequency.

If the MAINF value is valid, software can switch MAINCK to the main crystal oscillator. See Main Clock Source Selection.

Downloaded from Arrow.com.

Figure 27-3. Main Frequency Counter Block Diagram



27.6 PLL Controls

The PMC embeds 2 PLLs (PLLA and UPLL) that are controlled by the PMC_PLL_CTRL0, PMC_PLL_CTRL1, PMC_PLL_SSR, PMC_PLL_ACR and PMC_PLL_UPDATE registers. Each PLL is accessed in read or write through its index as defined in the table below, corresponding to the register field PMC_PLL_UPDT.ID. At any time, PLL_CTRL0, PLL_CTRL1 and PLL_ACR reflect the controls for the PLL with index PMC_PLL_UPDT.ID. When the UPDATE bit is set in PMC_PLL_UPDT, the PLL of index PMC_PLL_UPDT.ID is updated with the content of registers PLL_CTRL0, PLL_CTRL1 and PLL_ACR.

PLLA is fed by MAINCK while UPLL is fed by the main crystal oscillator. Each PLL has a constraint on the frequency it can generate on its clock output. Refer to the section “Electrical Characteristics”.

The table below describes all PLLs with their names and source clocks. For maximum frequency, refer to the section “Electrical Characteristics”.

Table 27-1. PLL IDs

Index	Name	Clock Source
0	PLLA	MAINCK
1	UPLL	MAINXTAL

27.6.1 Divider and Phase Lock Loop Programming

Each PLL is controlled the same way. The internal clock frequency is configured by setting PMC_PLL_CTRL1.MUL and PMC_PLL_CTRL1.FRACR.

PLLA can apply a division ratio on this internal clock to generate the clock for the PMC (PLLACK). UPLL always divides the internal clock by two to generate the clock for the PMC (UPLLCK) and the UTMI USB.

The COREPLLCK operating frequency is defined as:

$$f_{\mathrm{COREPLLCK}} = f_{\mathrm{ref}}\left(\mathrm{MUL} + 1 + \frac{\mathrm{FRACR}}{2^{22}}\right)$$

Downloaded from Arrow.com.

The PLLA clock frequency is defined by the following formula:

$$f_{\text{PLL Clock}} = \frac{f_{\text{COREPLLCK}}}{(\text{DIVPMC} + 1)}$$

The UPLL clock frequency is defined by the following formula:

$$f_{\text{PLL Clock}} = \frac{f_{\text{COREPLLCK}}}{2}$$

Each PLL sends a lock signal to the PMC to indicate its lock status. Once the lock signal has risen, the clock generated by the PLL is stable and can be sent to the PMC.

This signal reports the lock status of the PLL by setting the corresponding PMC_PLL_CTRL0.ENLOCK to ‘1’.

If the lock status is disabled, a startup time can be used instead in the PMC_PLL_UPDT register. The startup time is expressed as a number of MD_SLCK cycles. Once the counter has reached the specified value, a flag rises. The startup time field can only be written while all PLLs are disabled (i.e., their PLLEN fields are null).

If both a startup time and the lock are enabled, the lock sent by the PLL is read once the startup time has elapsed.

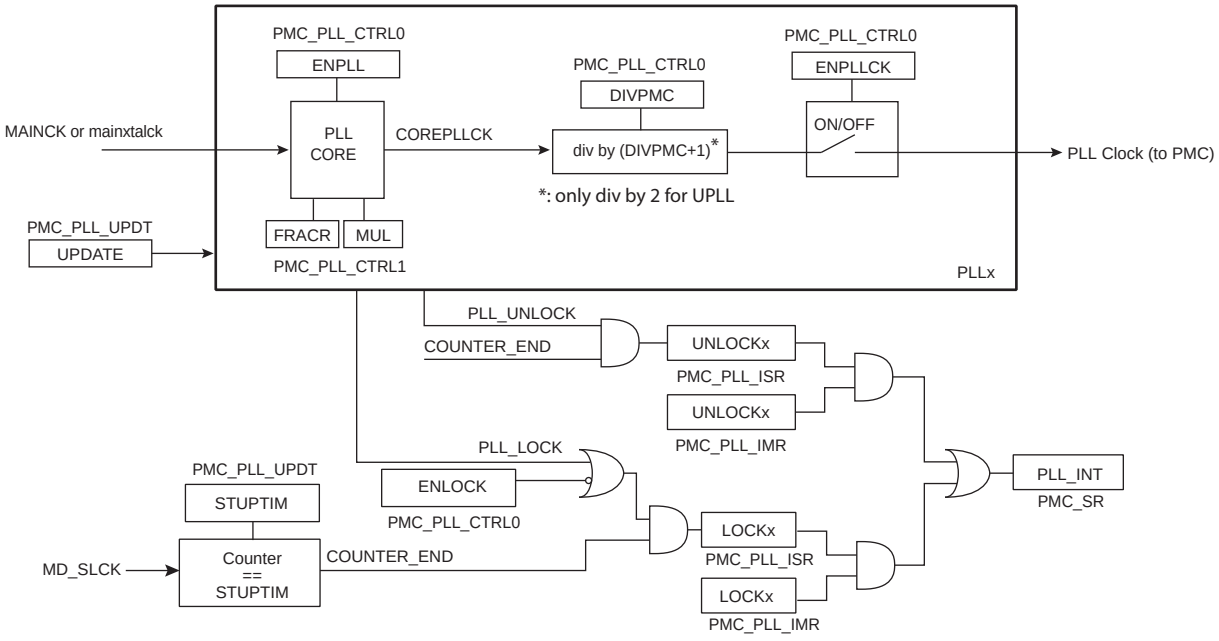
If neither the startup time nor the lock are enabled, there is no way to know the lock status of the PLL.

The PLL also embeds an unlock status that informs when the PLL lock is lost. When enabled, this status is read once the startup time (if defined) has elapsed.

The lock and unlock status can be used as interrupts.

See the following figure.

Figure 27-4. PLL Controls



Follow the steps below to enable a PLL:

1. Define the ID (ID=n) and startup time by configuring the fields PMC_PLL_UPDT.ID and PMC_PLL_UPDT.STUPTIM. Set PMC_PLL_UPDT.UPDATE to ‘0’.
2. Configure PMC_PLL_ACR.LOOP_FILTER.
3. Define the MUL and FRACR to be applied to PLL(n) in PMC_PLL_CTRL1.
   In case UPLL is being configured, follow Step 4. to Step 7., otherwise jump to Step 8.
4. Write PMC_PLL_ACR.UTMIBG to ‘1’ to enable the UTMI internal bandgap.
5. Wait 10 µs.
6. Write PMC_PLL_ACR.UTMIVR to ‘1’ to enable the UTMI internal regulator.
7. Wait 10 µs.

Downloaded from Arrow.com.

8. Set PMC_PLL_UPDT.UPDATE to ‘1’. PMC_PLL_UPDT.ID must equal the one written during Step 1., otherwise the update is cancelled.
9. In PMC_PLL_CTRL0, write a ‘1’ to ENLOCK and to ENPLL and configure DIVPMC (for PLLA only, as UPLL has a fixed divider value) and ENPLLCK.
10. Set PMC_PLL_UPDT.UPDATE to ‘1’. PMC_PLL_UPDT.ID must equal the one written during Step 1. otherwise the update is cancelled.
11. Wait for the lock bit to rise by polling the PMC_PLL_ISR0 or by enabling the corresponding interrupt in PMC_PLL_IER.
12. Disable the interrupt (if enabled).
13. Enable the unlock interrupt to quickly detect a failure on the generation of the PLL clock.

Once enabled (PMC_PLL_CTRL0.ENPLL=1), the PLL core generates its core clock (COREPLLCK).

Once the PLL has been enabled and has locked, the PLL configuration can be modified without switching off the cell.

The clock generated by the PLL is sent to the PMC if ENPLLCK has been set to ‘1’ and the PMC_PLL_UPDT.UPDATE bit has then been written to ‘1’.

To disable a PLL, the following sequence must be applied:

1. If the PLL drives a section of the system that is active, modify the source clock of the system.
2. Define the ID (ID=n) of the PLL to be switched off in PMC_UPDT. The bit UPDATE in this register must be set at 0 in this step.
3. In PMC_PLL_CTRL0, set ENPLLCK to 0 and leave ENPLL at ‘1’.
4. Set PMC_PLL_UPDT.UPDATE to ‘1’. PMC_PLL_UPDT.ID must equal the one written during step 2, otherwise the update is cancelled.
5. Write a ‘0’ to PMC_PLL_CTRL0.ENPLL.
6. In case a UPLL is being powered down, write a ‘0’ to PMC_PLL_ACR.UTMIBG and PMC_PLL_ACR.UTMIVR.
7. Set PMC_PLL_UPDT.UPDATE to ‘1’. PMC_PLL_UPDT.ID must equal the one written during Step 2., otherwise the update is cancelled.

27.6.2 PLL Unlock

Each PLL has an unlock flag. It is recommended to set the UNLOCK interrupt by setting the corresponding PMC_PLL_IER.UNLOCK bit to quickly detect a failure on PLL clock generation.

The rise of a PLL unlock signal implies a failure in the normal operation of the PLL (e.g., input clock loss). In this case, the PLL keeps operating but stops trying to lock the input clock. A manual clock switching to a stable clock should be performed to ensure CPU_CLK integrity

27.6.3 Spread Spectrum

Spread spectrum is obtained by slightly modifying the PLL target frequency. Two parameters are used to configure the spread spectrum:

- STEP—the frequency step
- NSTEP—the number of times the STEP will be applied

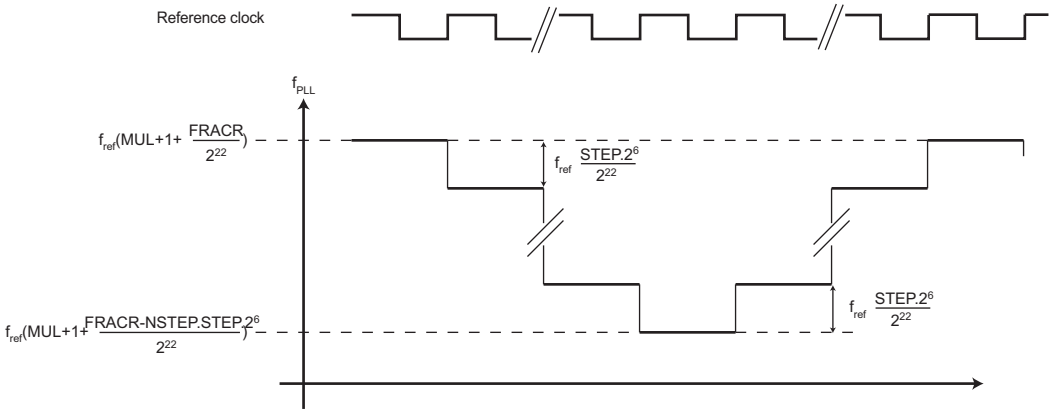
The spread spectrum can be applied only if the PLL is already enabled and locked. Once the spread spectrum has been enabled, it is no longer possible to modify the target frequency of the PLL. Prior to change the PLL frequency, the spread spectrum must be disabled and a period of 2 x NSTEP cycles of the PLL source clock must elapse.

When enabled, the spread spectrum logic modifies the fractional part of the PLL. The fractional factor applied to the PLL is in the following range: FRACR - (64 x STEP x NSTEP) up to FRACR.

Starting from the base frequency of the PLL configured in PMC_PLL_CTRL1 (MUL, FRACR), the spread spectrum mechanism decreases the PLL frequency, and when the minimum is reached, the PLL frequency is increased up to the value configured through the PMC_PLL_CTRL1 register (the PLL frequency never overpasses that value).

Downloaded from Arrow.com.

Figure 27-5. Spread Spectrum Mechanism



Downloaded from Arrow.com.

28. Power Management Controller (PMC)

28.1 Description

The Power Management Controller (PMC) optimizes power consumption by controlling all system and user peripheral clocks. The PMC enables/disables the clock inputs to many of the peripherals and to the processor.

The Slow Clock Controller (SCKC) selects the source of TD_SLCK (drives the real-time part (RTT/RTC)). The source of MD_SLCK (drives the rest of the system controller: wakeup logic, watchdog, PMC, etc.) is always the slow RC oscillator.

By default, at startup, the chip runs out of MCK using the main RC oscillator.

28.2 Embedded Characteristics

The Power Management Controller provides the following clocks:

- Master Clock (MCK)—programmable from a few hundred Hz to the maximum operating frequency of the device. It is available to the modules running permanently.
- Processor Clock (CPU_CLK)—can be tuned through a frequency scaler module and automatically switched off when entering the processor in Sleep mode.
- Free-running Processor Clock (FCLK)—the source clock of CPU_CLK. Is not affected when Sleep mode is activated.
- UHDP Clocks (UHP48M and UHP12M)—required by USB Host Device Port operations.
- Peripheral Clocks with independent on/off control, provided to the peripherals. Each peripheral clock is inherited from MCK.
- Programmable Clock Outputs (PCKx), selected from the clock generator outputs to drive the device PCKx pins.
- Generic Clock (GCLK) with controllable division and on/off control, independent of MCK and CPU_CLK. Provided to selected peripherals. Refer to the table “Peripheral Identifiers” for more details on GCLK availability per peripheral.

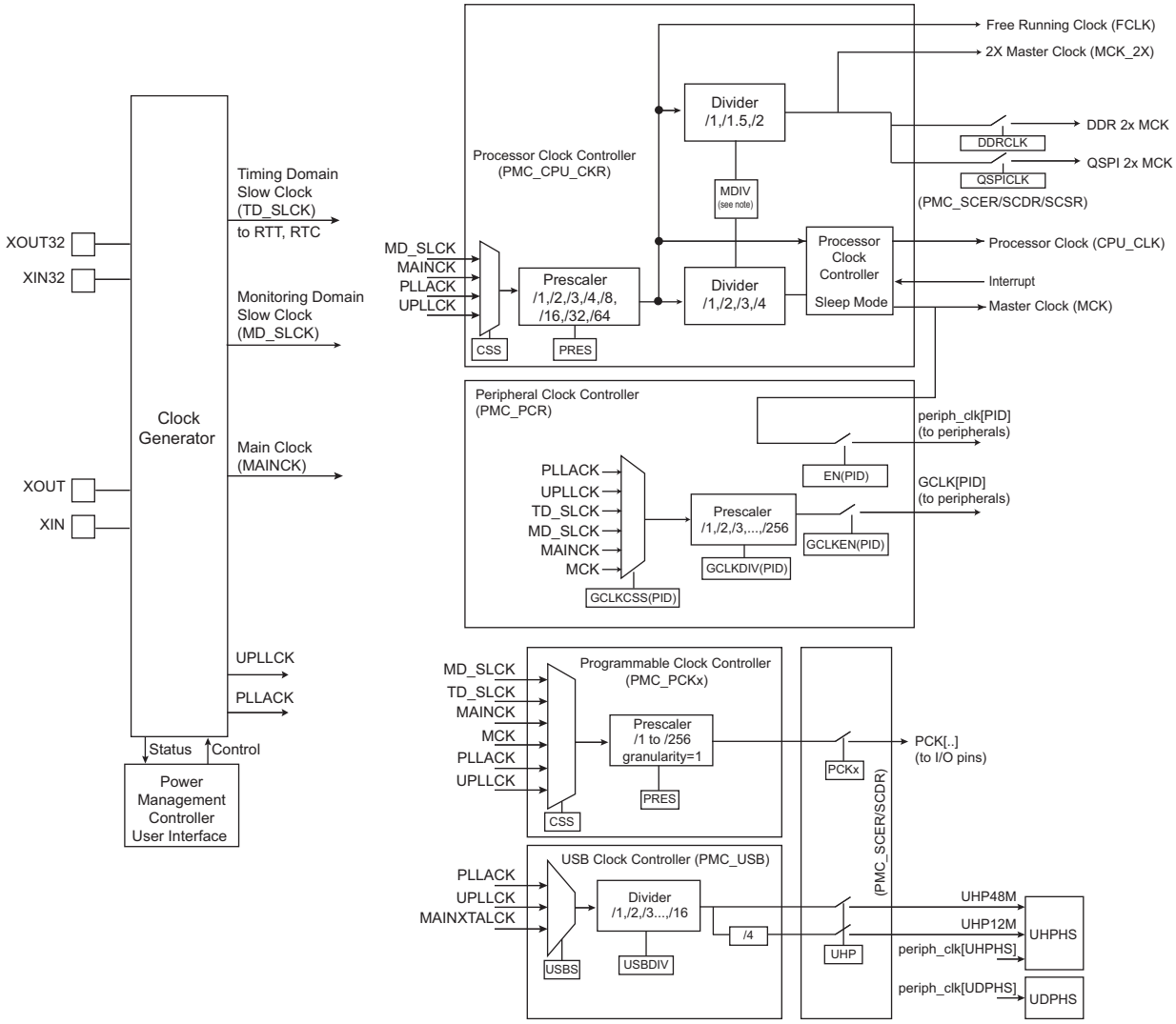
The Power Management Controller also provides the following features on clocks:

- A main crystal oscillator failure detector
- A 32.768 kHz crystal oscillator frequency monitor
- A frequency counter on main crystal oscillator or main RC oscillator
- An MCK failure detector

Downloaded from Arrow.com.

28.3 Block Diagram

Figure 28-1. General Clock Distribution Block Diagram



Note: MDIV should always be different from 0 when using DDR memories. If MDIV must be set to 0, first switch the DDR memories to Self-refresh mode and disable DDRCLK.

28.4 Processor Clock Controller

The PMC features a Processor Clock (CPU_CLK) controller that implements the processor Sleep mode. CPU_CLK can be disabled by executing the WFI (WaitForInterrupt) processor instruction.

CPU_CLK is enabled after a reset and is automatically re-enabled by any enabled interrupt. The processor Sleep mode is entered by disabling CPU_CLK, which is automatically re-enabled by any enabled interrupt, or by the reset of the product.

When processor Sleep mode is entered, the current instruction is finished before the CPU_CLK is stopped, but this does not prevent data transfers from other masters of the system bus.

The clock selection is done in PMC_CPU_CKR.CSS.

Downloaded from Arrow.com.

The prescaler is configured in PMC_CPU_CKR.PRES.

The Processor Clock Controller also generates a master clock, MCK, which is a subdivision of the CPU_CLK.

Only one of CSS, PRES and MDIV fields can be modified at a time. When one of these parameters is modified, no other modification can be performed on these fields as long as the MCKRDY status flag is low.

Any modification in CSS, PRES or MDIV fields must never lead to generate a MCK frequency that is greater than the maximum allowed system frequency. When changing the source clock of the system to a faster clock, the fields must be modified using the following order: MDIV, PRES and then CSS. When changing the source clock of the system to a slower clock, the fields must be modified using the following order: CSS, PRES and then MDIV.

If the destination clock does not exist, the switching is not performed. The CPU_CLK keeps running with the previous clock and the system must be reset to run correctly again.

28.5 USB Clock Controller

The user can select the PLLA, the UPLL or the main Crystal Oscillator output as the USB source clock by writing the PMC_USB.USBS field. If using the USB, the user must program the PLL to generate an appropriate frequency depending on the PMC_USB.USBDIV field.

When the PLL output is stable, i.e., the LOCK bit is set, the USB device and host clocks can be enabled by setting the UHP bits in the System Clock Enable register (PMC_SCER). To save power on this peripheral when it is not used, the user can set the UHP bits in the System Clock Disable register (PMC_SCDR). The UHP bits in the System Clock Status register (PMC_SCSR) gives the activity of this clock. The USB device and host ports requires both the 48 MHz signal and the peripheral clock. The USB peripheral clock may be controlled by means of the Peripheral Clock Controller.

28.6 Free-running Processor Clock

The free-running Processor clock (FCLK) used for sampling interrupts and clocking debug blocks ensures that interrupts can be sampled, and sleep events can be traced, while the processor is sleeping.

28.7 Peripheral and Generic Clock Controller

The PMC controls the clocks of the embedded peripherals by means of the Peripheral Control register (PMC_PCR). With this register, the user can enable and disable the different clocks used by the peripherals:

- Peripheral clocks (periph_clk[PID]), routed to every peripheral and derived from MCK. It is mandatory to enable this clock before using a peripheral.
- Generic clocks (GCLK[PID]), routed to selected peripherals only (refer to the Peripheral Identifiers table in section Peripherals). These clocks are independent of the core and bus clocks (CPU_CLK, MCK and periph_clk[PID]). They are generated by selection and division of available sources. The list of available source clocks depends on the peripheral. Refer to the description of each peripheral to know available sources and limitations to be applied to GCLK[PID] compared to periph_clk[PID].

To configure a peripheral's clocks, PMC_PCR.CMD must be written to '1' and PMC_PCR.PID must be written with the index of the corresponding peripheral. All other configuration fields must be correctly set.

To read the current clock configuration of a peripheral, PMC_PCR must be first accessed with PMC_PCR.CMD bit written to '0' and PMC_PCR.PID field written with the index of the corresponding peripheral. This write does not modify the configuration of the peripheral. The PMC_PCR can then be read to know the configuration status of the corresponding PID.

The status of the peripheral clock activity can be read in the Peripheral Clock Status registers (PMC_CSRx).

The status of the peripheral generic clock activity can be read in the Generic Clock Status registers (PMC_GCSRx).

When a peripheral or a generic clock is disabled, it is immediately stopped. These clocks are disabled after a reset. The source and the division ratio of generic clocks must not be modified while the peripheral is enabled. The generic clock configuration must be set before the peripheral is enabled.

Downloaded from Arrow.com.

To stop a peripheral clock, it is recommended that the system software wait until the peripheral has executed its last programmed operation before disabling the clock. This is to avoid data corruption or erroneous behavior of the system.

28.8 Programmable Clock Output Controller

The PMC controls two signals to be output on the external pins PCKx. Each signal can be independently programmed via the Programmable Clock registers (PMC_PCKx).

PCKx can be independently selected between MD_SLCK, TD_SLCK, MAINCK, MCK and any PLLCK by configuring PMC_PCKx.CSS. Each output signal can also be divided by 1 to 256 by configuring PMC_PCKx.PRES.

Each output signal can be enabled and disabled by writing a '1' to the corresponding bits PMC_SCER.PCKx and PMC_SCDR.PCKx, respectively. The status of the active programmable output clocks is given in PMC_SCSR.PCKx.

The status flag PMC_SR.PCKRDYx indicates that the clock configured through the PMC_PCKx register is correctly established.

As the Programmable Clock Controller does not manage with glitch prevention when switching clocks, it is strongly recommended to disable PCKx before any configuration change and to re-enable it once the change is performed.

28.9 Ultra-Low Power Mode and Fast Startup

The following sections give a brief description of the Ultra-Low Power mode features of the device as seen from the PMC. A more detailed description, including power consumption and wake-up time figures, can be found in the section "Electrical Characteristics".

28.9.1 ULP Mode 1

When the system is in Ultra-Low Power (ULP) mode 1, all clocks of the system except MD_SLCK are stopped. The source clock of all MCKx must be set to the main clock, and the source of the main clock must be set to main RC oscillator.

Prior to instructing the device to enter ULP mode 1:

1. Select main RC as the source of MAINCK by configuring CKGR_MOR.MOSCSEL to '0'.
2. Select MAINCK as the source of MCK by configuring PMC_CPU_CKR.CSS to '1'.
3. Disable the PLL if enabled and disable the main crystal oscillator by setting CKGR_MOR.MOSCXTEN to '0'.
4. Wait for two SLCK clock cycles.
5. Clear the internal wakeup sources.
6. Verify that none of the enabled external wakeup inputs (WKUP) hold an active polarity.

The system enters ULP mode 1 by setting CKGR_MOR.ULP1. The PMC registers must not be accessed immediately after this access.

28.9.2 Fast Startup

At exit from ULP mode 1, the device allows the processor to restart in several microseconds only if the C-code function that manages the ULP mode 1 entry and exit is linked to and executed from on-chip SRAM.

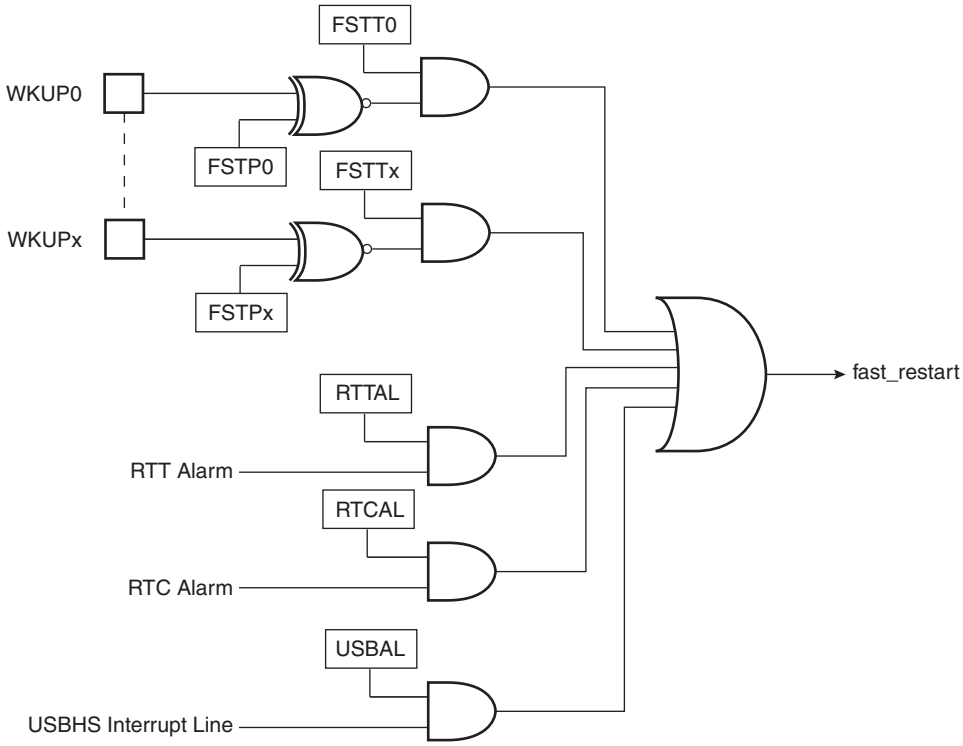
A fast startup occurs upon the detection of a programmed level on one of the wakeup inputs (WKUP) or upon an active alarm from the RTC, RTT and USB Controller. The polarity of each of the wakeup inputs is programmable in the PMC Wakeup Control Register (PMC_WCR).

WARNING The duration of the WKUPx pins active level must be greater than four MAINCK cycles.

The fast startup circuitry, as shown in the following figure, is fully asynchronous and provides a fast startup signal to the PMC. As soon as the fast startup signal is asserted, the main RC oscillator restarts automatically.

Downloaded from Arrow.com.

Figure 28-2. Fast Startup Circuitry



Each wakeup input pin can be configured to generate a fast startup event by setting the corresponding bits in PMC_WCR.

To configure a wakeup pin, a write access must be performed in PMC_WCR (CMD='1'). Field PID must be written with the ID of the wakeup pin, FSTP set to the polarity of the wakeup pin and EN set to enable/disable the wakeup pin.

To read the configuration status of a wakeup pin, PMC_WCR.PID must be written with the ID of the wakeup pin and the CMD bit set to '0'. Then the next read access to PMC_WCR sends the configuration status of the wakeup pin specified in PID.

Each alarm can be enabled to generate a fast startup event by setting the corresponding bit in PMC_FSMR.

The user interface does not provide any status for fast startup. The status can be read in the PIO Controller and the status registers of the RTC, RTT and USB Controller.

28.10 Main Crystal Oscillator Failure Detection

The main crystal oscillator failure detector monitors the main crystal oscillator against the slow RC oscillator and provides an automatic switchover of the MAINCK source to the main RC oscillator in case of failure detection.

The failure detector can be enabled or disabled by configuring CKGR_MOR.CFDEN. It cannot be enabled if the main crystal oscillator is disabled. It must be disabled before disabling the main crystal oscillator.

It is also disabled in either of the following cases:

- after a VDDCORE reset
- when the main crystal oscillator is disabled (MOSCXTEN = 0)

A failure is detected by means of a counter incrementing on the main crystal oscillator output and detection logic is triggered by the slow RC oscillator which is automatically enabled when CFDEN = 1.

The counter is cleared when the slow RC oscillator clock signal is low and enabled when the signal is high. Thus, the failure detection time is one slow RC oscillator period. If, during the high level period of the slow RC oscillator clock signal, less than eight main crystal oscillator clock periods have been counted, then a failure is reported. Note that

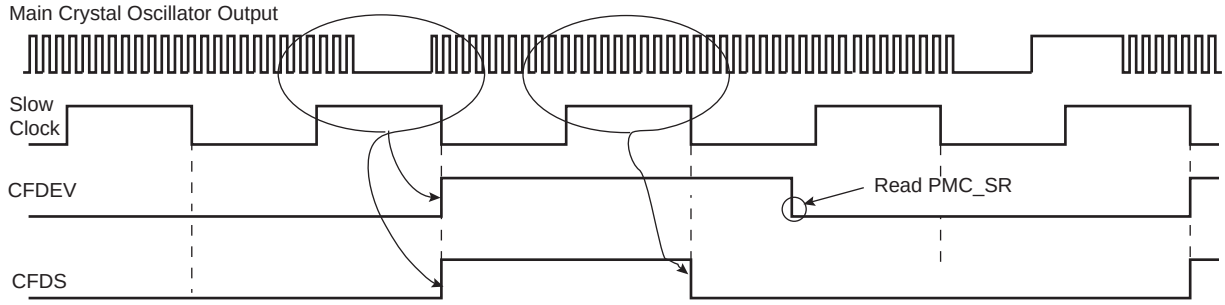
Downloaded from Arrow.com.

when enabling the failure detector, up to two cycles of the slow RC oscillator are needed to detect a failure of the main crystal oscillator.

If a main crystal oscillator failure is detected, PMC_SR.CFDEV and PMC_SR.FOS both indicate a failure event. PMC_SR.CFDEV is cleared on read of PMC_SR, and PMC_SR.FOS is cleared by writing a ‘1’ to the FOCLR bit in the PMC Fault Output Clear register (PMC_FOCR).

Only PMC_SR.CFDEV can generate an interrupt if the corresponding interrupt source is enabled in PMC_IER. The current status of the clock failure detection can be read at any time from PMC_SR.CFDS.

Figure 28-3. Clock Failure Detection Example



Note: Ratio of clock periods is for illustration purposes only.

If the CKGR_MOR.AUTOMAINSW bit is set to'1', the source of MAINCK automatically switches to the MAIN RC oscillator. If the main RC oscillator was previously powered off, it is first powered on before switching. If the CKGR_MOR.AUTOCPUSW bit is set to'1', the source of MCK automatically switches to MAINCK.

If the main crystal oscillator is selected as the source of MAINCK, the PMC can be configured to automatically select the main RC oscillator as the source of MAINCK in case of a main crystal oscillator failure detection by setting the CKGR_MOR.AUTOMAINSW to '1'. Additionally, if the source of CPU_CLK is a PLL driven by the main crystal oscillator, the PMC can be configured to automatically select the MAINCK as the source of CPU_CLK in case of a main crystal oscillator failure detection by setting the CKGR_MOR.AUTOCPUSW to '1'. CKGR_MOR.AUTOMAINSW must be set to '1' prior to setting CKGR_MOR.AUTOCPUSW to '1'.

Two slow RC oscillator clock cycles are necessary to detect and switch from the main crystal oscillator to the main RC oscillator if the source of MCK is MAINCK, or three slow RC oscillator clock cycles if the source of MCK is a PLL.

28.11 32.768 kHz Crystal Oscillator Frequency Monitor

The frequency of the 32.768 kHz crystal oscillator can be monitored by configuring CKGR_MOR.XT32KFME. Prior to enabling the monitoring, the 32.768 kHz crystal oscillator must be started and its startup time be elapsed. Refer to the section “Slow Clock Controller (SCKC)” for details on the slow clock generator.

An error flag (PMC_SR.XT32KERR) is asserted when the 32.768 kHz crystal oscillator frequency is out of its nominal frequency value (i.e., 32.768 kHz). The error flag can be cleared only if the monitoring is disabled.

The frequency drift is computed with the main RC oscillator. The permitted drift of the crystal is 10000 ppm (1%), which allows any standard crystal to be used.

The monitored clock frequency is declared invalid if at least four consecutive 32.768 kHz crystal oscillator clock period measurement results are over the nominal period. Note that modifying the trimming values of the main RC oscillator (PMC_OCR) may impact the monitor accuracy and lead to inappropriate failure detection.

The error flag can be defined as an interrupt source of the PMC by setting PMC_IER.XT32KERR. This flag is also routed to the Reset Controller (RSTC) and may generate a reset of the device.

28.12 MCK Frequency Monitor

The frequency of MCK can be monitored with the main RC oscillator. This monitoring can only be performed if the MCK frequency is at least three times faster than the embedded main RC oscillator. This function is enabled by writing a ‘1’ to PMC_IER.MCKMON.

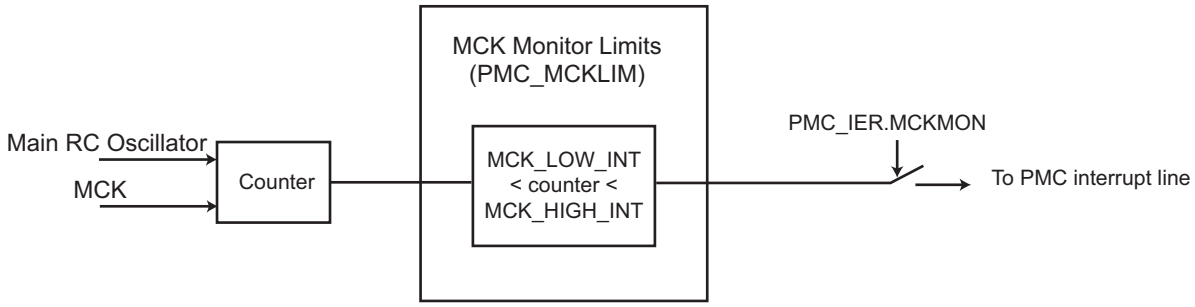
Downloaded from Arrow.com.

An error on the MCK frequency can lead to a PMC interrupt.

When the corresponding PMC interrupt is enabled, the status of the MCK monitoring can be read on PMC_SR.MCKMON. This status is cleared on read.

Once enabled, the monitor continuously counts the number of MCK cycles within 15 cycles of the embedded main RC oscillator. The result is then compared to threshold values defined in the PMC_MCKLIM register. Two levels of threshold can be defined to generate a reset of the system.

Figure 28-4. MCK Frequency Monitor



28.13 Recommended Programming Sequence

Follow the steps below to program the PMC:

1. If the main crystal oscillator is not required, the PLL can be directly configured (step 5) else this oscillator must be started (step 2).
2. Verify the existence and frequency value of the main crystal oscillator following the sequence defined in 27.5.6 Main Frequency Counter
3. If the main crystal oscillator is enabled and valid, the source of MAINCK can be switched to the main crystal oscillator by writing CKGR_MOR.MOSCSEL to 1 else the PLL can be directly configured.
4. Wait for the end of the MAINCK source switching by either polling the MOSCSELS or setting the corresponding interrupt
5. Configure the PLLs by following the setup defined in 27.6.1 Divider and Phase Lock Loop Programming (if not required, proceed to step 6):
6. Configure the MCK division ratio by setting PMC_CPU_CKR.MDIV. Available values are 0, 1, 2, 3. MCK output is the CPU_CLK frequency divided by 1, 2, 3 or 4, depending on the value programmed in MDIV.
   By default, MDIV is cleared, which indicates that the CPU_CLK is equal to MCK.
7. Wait for the end of the MCK ratio switching by either polling the MCKRDY or setting the corresponding interrupt.
8. Select the division ratio of CPU_CLK by setting PMC_CPU_CKR.PRES.
   PRES is used to define the CPU_CLK and MCK prescaler. The user can choose between different values (1, 2, 3, 4, 8, 16, 32, 64). Prescaler output is the selected clock source frequency divided by the PRES value.
9. Wait for the end of the CPU_CLK ratio switching by either polling the MCKRDY or setting the corresponding interrupt.
10. Select the source clock of CPU_CLK by setting PMC_CPU_CKR.CSS.
    CSS is used to select the clock source of MCK and CPU_CLK. By default, the selected clock source is MAINCK.
11. Wait for the end of the CPU_CLK source switching by either polling the MCKRDY or setting the corresponding interrupt.
    PMC_CPU_CKR must not be programmed in a single write operation.

    Reconfiguring MDIV, PRES and CSS fields must always be done by following the right order of operation described above (steps 6 to 11).
12. Configure the programmable clocks (PCKx):
    PCKx are controlled via registers PMC_SCER, PMC_SCDR and PMC_SCSR.

Downloaded from Arrow.com.

PCKx can be enabled and/or disabled via PMC_SCER and PMC_SCDR. Two PCKx can be used. PMC_SCSR indicates which PCKx is enabled. By default all PCKx are disabled.

PMC_PCKx registers are used to configure PCKx as described in 28.8 Programmable Clock Output Controller.

13. Enable the peripheral and generic clocks.
Once all of the previous steps have been completed, the peripheral and generic clocks can be configured via register PMC_PCR as described in 28.7 Peripheral and Generic Clock Controller.

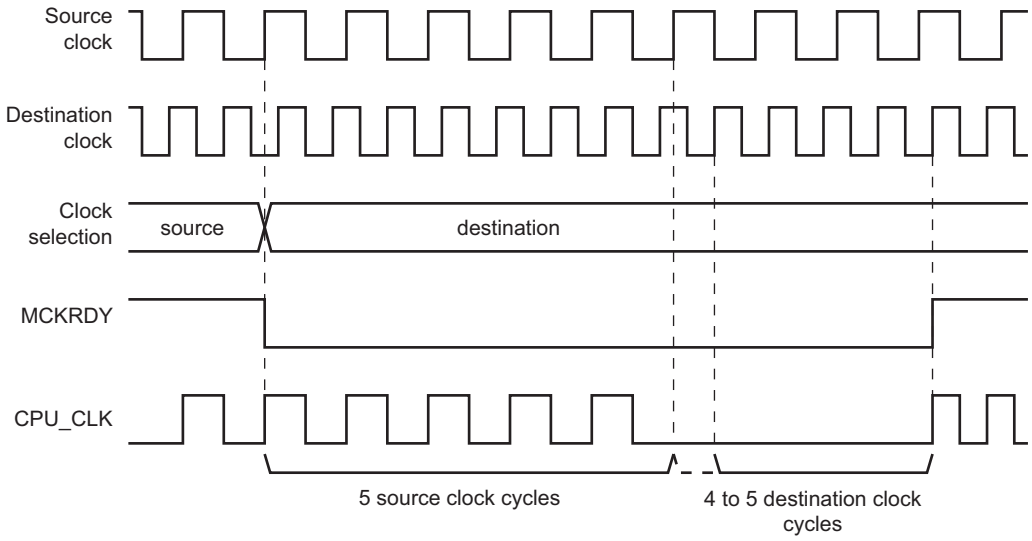
28.14 Clock Switching Details

28.14.1 CPU Clock Switching Timings

The glitch-free clock switcher implemented to control the sources of CPU_CLK and MCK performs clock switching in 5 clock cycles of the currently used clock plus 5 cycles of the target clock.

The clock switching is effective once MCKRDY rises. See the following figure.

Figure 28-5. Switch CPU Clock (CPU_CLK) from Source Clock to Destination Clock



28.15 Register Write Protection

To prevent any single software error from corrupting PMC behavior, certain registers in the address space can be write-protected by setting the WPEN bit or the WPITEN bit in the PMC Write Protection Mode Register (PMC_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the PMC Write Protection Status Register (PMC_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the PMC_WPSR.

The following registers are write-protected when the WPEN bit is set in PMC_WPMR:

- PMC System Clock Enable Register
- PMC System Clock Disable Register
- PMC PLL Control Register 0
- PMC PLL Control Register 1
- PMC PLL Spread Spectrum Register
- PMC PLL Analog Control Register
- PMC PLL Update Register

Downloaded from Arrow.com.

- PMC Clock Generator Main Oscillator Register
- PMC Clock Generator Main Clock Frequency Register
- PMC CPU Clock Register
- PMC_USB
- PMC Programmable Clock Register
- PMC Fast Startup Mode Register
- PMC Wakeup Control Register
- PMC Peripheral Control Register
- PMC Oscillator Calibration Register
- PMC MCK0 Monitor Limits Register

The following interrupt registers are write-protected when the WPITEN bit is set in PMC_WPMR:

- PMC Interrupt Enable Register
- PMC Interrupt Disable Register
- PMC PLL Interrupt Enable Register
- PMC PLL Interrupt Disable Register

Downloaded from Arrow.com.

28.16 Register Summary

Offset	Name	Bit Pos.								
0x00	PMC_SCER	31:24								
		23:16					QSPICLK			
		15:8							PCK1	PCK0
		7:0		UHP				DDRCK		
0x04	PMC_SCDR	31:24								
		23:16					QSPICLK			
		15:8							PCK1	PCK0
		7:0		UHP				DDRCK		
0x08	PMC_SCSR	31:24								
		23:16					QSPICLK			
		15:8							PCK1	PCK0
		7:0		UHP				DDRCK		
0x0C	PMC_PLL_CTRL0	31:24	ENLOCK		ENPLLCK	ENPLL				
		23:16								
		15:8								
		7:0	DIVPMC[7:0]							
0x10	PMC_PLL_CTRL1	31:24	MUL[7:0]							
		23:16			FRACR[21:16]					
		15:8	FRACR[15:8]							
		7:0	FRACR[7:0]							
0x14	PMC_PLL_SSR	31:24				ENSPREAD				
		23:16	NSTEP[7:0]							
		15:8	STEP[15:8]							
		7:0	STEP[7:0]							
0x18	PMC_PLL_ACR	31:24			LOOP_FILTER[5:0]					
		23:16						LOCK_THR[2:0]		
		15:8			UTMIBG	UTMIVR	CONTROL[11:8]			
		7:0	CONTROL[7:0]							
0x1C	PMC_PLL_UPDT	31:24								
		23:16	STUPTIM[7:0]							
		15:8								UPDATE
		7:0								ID
0x20	CKGR_MOR	31:24		AUTOCPUSW	AUTOMAINSW			XT32KFME	CFDEN	MOSCSEL
		23:16	KEY[7:0]							
		15:8	MOSCXTST[7:0]							
		7:0					MOSCRCEN	ULP1		MOSCXTEN
0x24	CKGR_MCFR	31:24								CCSS
		23:16				RCMEAS				MAINFRDY
		15:8	MAINF[15:8]							
		7:0	MAINF[7:0]							
0x28	PMC_CPU_CKR	31:24								
		23:16								
		15:8						MDIV[2:0]		
		7:0		PRES[2:0]					CSS[1:0]	
0x2C ... 0x37	Reserved									
0x38	PMC_USB	31:24								
		23:16								
		15:8					USBDIV[3:0]			
		7:0							USBS[1:0]	
0x3C ... 0x3F	Reserved									

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x40	PMC_PCK0	31:24								
		23:16								
		15:8	PRES[7:0]							
		7:0				CSS[4:0]				
0x44	PMC_PCK1	31:24								
		23:16								
		15:8	PRES[7:0]							
		7:0				CSS[4:0]				
0x48 ... 0x5F	Reserved									
0x60	PMC_IER	31:24							PLL_INT	
		23:16	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
		15:8							PCKRDY1	PCKRDY0
		7:0					MCKRDY			MOSCXTS
0x64	PMC_IDR	31:24							PLL_INT	
		23:16	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
		15:8							PCKRDY1	PCKRDY0
		7:0					MCKRDY			MOSCXTS
0x68	PMC_SR	31:24							PLL_INT	GCLKRDY
		23:16	MCKMON		XT32KERR	FOS	CFDS	CFDEV	MOSCRCS	MOSCSELS
		15:8							PCKRDY1	PCKRDY0
		7:0					MCKRDY			MOSCXTS
0x6C	PMC_IMR	31:24							PLL_INT	
		23:16	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
		15:8							PCKRDY1	PCKRDY0
		7:0					MCKRDY			MOSCXTS
0x70	PMC_FSMR	31:24							WLAN1	WLAN0
		23:16						USBAL	RTCAL	RTTAL
		15:8								
		7:0								
0x74	PMC_WCR	31:24								CMD
		23:16							WIP	EN
		15:8								
		7:0					WKPIONB[3:0]			
0x78	PMC_FOCR	31:24								
		23:16								
		15:8								
		7:0								FOCLR
0x7C ... 0x7F	Reserved									
0x80	PMC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPITEN	WPEN
0x84	PMC_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS
0x88	PMC_PCR	31:24	CMD		GCLKEN	EN	GCLKDIV[7:4]			
		23:16	GCLKDIV[3:0]							
		15:8				GCLKCSS[4:0]				
		7:0		PID[6:0]						
0x8C	PMC_OCR	31:24								
		23:16	SEL12	CAL12[6:0]						
		15:8								
		7:0								

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x90 ... 0x9B	Reserved									
0x9C	PMC_MCKLIM	31:24								
		23:16								
		15:8	MCK_HIGH_IT[7:0]							
		7:0	MCK_LOW_IT[7:0]							
0xA0	PMC_CSR0	31:24		PID30	PID29	PID28	PID27	PID26	PID25	PID24
		23:16	PID23	PID22		PID20	PID19	PID18	PID17	PID16
		15:8	PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8
		7:0	PID7	PID6	PID5	PID4	PID3	PID2		
0xA4	PMC_CSR1	31:24								
		23:16							PID49	
		15:8	PID47		PID45	PID44	PID43	PID42	PID41	PID40
		7:0	PID39	PID38	PID37	PID36	PID35	PID34	PID33	PID32
0xA8 ... 0xBF	Reserved									
0xC0	PMC_GCSR0	31:24						GPID26	GPID25	
		23:16					GPID19		GPID17	GPID16
		15:8	GPID15	GPID14	GPID13	GPID12	GPID11	GPID10	GPID9	GPID8
		7:0	GPID7	GPID6	GPID5					
0xC4	PMC_GCSR1	31:24								
		23:16								
		15:8	GPID47		GPID45			GPID42		
		7:0			GPID37			GPID34	GPID33	GPID32
0xC8 ... 0xDF	Reserved									
0xE0	PMC_PLL_IER	31:24								
		23:16							UNLOCKU	UNLOCKA
		15:8								
		7:0							LOCKU	LOCKA
0xE4	PMC_PLL_IDR	31:24								
		23:16							UNLOCKU	UNLOCKA
		15:8								
		7:0							LOCKU	LOCKA
0xE8	PMC_PLL_IMR	31:24								
		23:16							UNLOCKU	UNLOCKA
		15:8								
		7:0							LOCKU	LOCKA
0xEC	PMC_PLL_ISR0	31:24								
		23:16							UNLOCKU	UNLOCKA
		15:8								
		7:0							LOCKU	LOCKA
0xF0	PMC_PLL_ISR1	31:24								
		23:16							OVRU	OVRA
		15:8								
		7:0							UDRU	UDRA

Downloaded from Arrow.com.

28.16.1 PMC System Clock Enable Register

Name: PMC_SCER
Offset: 0x0000
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					QSPICLK			
Access					W			
Reset					–			

Bit	15	14	13	12	11	10	9	8
							PCK1	PCK0
Access							W	W
Reset							–	–

Bit	7	6	5	4	3	2	1	0
		UHP				DDRCK		
Access		W				W		
Reset		–				–		

Bit 19 – QSPICLK QSPI 2x Clock Enable

Value	Description
0	No effect.
1	Enables the QSPI 2x clock.

Bits 8, 9 – PCKx Programmable Clock x Output Enable

Value	Description
0	No effect.
1	Enables the corresponding Programmable Clock output.

Bit 6 – UHP USB Host OHCI Clocks Enable

Value	Description
0	No effect.
1	Enables the UHP48M and UHP12M OHCI clocks.

Bit 2 – DDRCK MPDDRC/SDRAMC Clock Enable

Value	Description
0	No effect.
1	Enables the MPDDRC or SDRAMC clock.

Downloaded from Arrow.com.

28.16.2 PMC System Clock Disable Register

Name: PMC_SCDR
Offset: 0x0004
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					QSPICLK			
Access					W			
Reset					–			

Bit	15	14	13	12	11	10	9	8
							PCK1	PCK0
Access							W	W
Reset							–	–

Bit	7	6	5	4	3	2	1	0
		UHP				DDRCK		
Access		W				W		
Reset		–				–		

Bit 19 – QSPICLK QSPI 2x Clock Disable

Value	Description
0	No effect.
1	Disables the QSPI 2x clock.

Bits 8, 9 – PCKx Programmable Clock x Output Disable

Value	Description
0	No effect.
1	Disables the corresponding Programmable Clock output.

Bit 6 – UHP USB Host OHCI Clocks Disable

Value	Description
0	No effect.
1	Disables the UHP48M and UHP12M OHCI clocks.

Bit 2 – DDRCK MPDDRC/SDRAMC Clock Disable

Value	Description
0	No effect.
1	Disables the MPDDRC or SDRAMC clock.

Downloaded from Arrow.com.

28.16.3 PMC System Clock Status Register

Name: PMC_SCSR
Offset: 0x0008
Reset: 0x00000001
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					QSPICLK			
Access					R			
Reset					0			

Bit	15	14	13	12	11	10	9	8
							PCK1	PCK0
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
		UHP				DDRCK		
Access		R				R		
Reset		0				0		

Bit 19 – QSPICLK QSPI 2x Clock Status

Value	Description
0	The QSPI 2x clock is disabled.
1	The QSPI 2x clock is enabled.

Bits 8, 9 – PCKx Programmable Clock x Output Status

Value	Description
0	The corresponding Programmable Clock output is disabled.
1	The corresponding Programmable Clock output is enabled.

Bit 6 – UHP USB Host OHCI Clocks Status

Value	Description
0	The UHP48M and UHP12M OHCI clocks are disabled.
1	The UHP48M and UHP12M OHCI clocks are enabled.

Bit 2 – DDRCK MPDDRC/SDRAMC Clock Status

Value	Description
0	The MPDDRC or SDRAMC clock is disabled.
1	The MPDDRC or SDRAMC clock is enabled.

Downloaded from Arrow.com.

28.16.4 PMC PLL Control Register 0

Name: PMC_PLL_CTRL0
Offset: 0x000C
Reset: 0x00000000
Property: Read/Write

All fields defined here are applied to the PLL defined by the last ID field written in the PMC_PLL_UPDT register.

Bit	31	30	29	28	27	26	25	24
	ENLOCK		ENPLLCK	ENPLL				
Access	R/W		R/W	R/W				
Reset	0		0	0				

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DIVPMC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – ENLOCK Enable PLL Lock

Value	Description
0	The lock signal sent by the PLL is ignored. The PLL is considered as locked once the startup time defined by PMC_PLL_UPDT.STUPTIM has elapsed.
1	The PLL is considered as locked once the startup time defined by PMC_PLL_UPDT.STUPTIM has elapsed and the lock signal sent by the PLL has risen.

Bit 29 – ENPLLCK Enable PLL Clock for PMC

Value	Description
0	The clock generated by the PLL is not send to the PMC.
1	The clock generated by the PLL is sent to the PMC.

Bit 28 – ENPLL Enable PLL

Value	Description
0	The PLL is off.
1	The PLL is on.

Bits 7:0 – DIVPMC[7:0] Divider for PMC
Specifies the division ratio applied to the internal PLL clock before being sent to the PMC. The frequency is defined by the following formula:

$$f_{\text{PLL Clock}} = \frac{f_{\text{COREPLLCK}}}{(\text{DIVPMC} + 1)}$$

Downloaded from Arrow.com.

28.16.5 PMC PLL Control Register 1

Name: PMC_PLL_CTRL1
Offset: 0x0010
Reset: 0x00000000
Property: Read/Write

All fields defined here are applied to the PLL defined by the last ID field written in the PMC_PLL_UPDT register.

Bit	31	30	29	28	27	26	25	24
	MUL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			FRACR[21:16]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FRACR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FRACR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – MUL[7:0] Multiplier Factor Value
Configures the internal clock frequency. See Divider and Phase Lock Loop Programming.

Bits 21:0 – FRACR[21:0] Fractional Loop Divider Setting

Downloaded from Arrow.com.

28.16.6 PMC PLL Spread Spectrum Register

Name: PMC_PLL_SSR
Offset: 0x0014
Reset: 0x00000000
Property: Read/Write

All fields defined here are applied to the PLL defined by the last ID field written in the PMC_PLL_UPDT register.

Bit	31	30	29	28	27	26	25	24
				ENSPREAD				
Access				R/W				
Reset				0				

Bit	23	22	21	20	19	18	17	16
	NSTEP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	STEP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	STEP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 28 – ENSPREAD Spread Spectrum Enable

Value	Description
0	The spread spectrum is not applied to the PLL.
1	The spread spectrum is applied to the PLL.

Bits 23:16 – NSTEP[7:0] Spread Spectrum Number of Steps
Specifies how many times STEP is applied to the PLL ratio. The value of NSTEP must be equal to or greater than 1.

Bits 15:0 – STEP[15:0] Spread Spectrum Step Size
When the spread spectrum is active, this field defines the step size that will be applied the PMC_PLL_CTRL1.FRACR factor. The step is applied on the LSB of PMC_PLL_CTRL1.FRACR.

Downloaded from Arrow.com.

28.16.7 PMC PLL Analog Control Register

Name: PMC_PLL_ACR
Offset: 0x0018
Reset: 0x00020033
Property: Read/Write

All fields defined here are applied to the PLL defined by the last ID field written in the PMC_PLL_UPDT register.

Bit	31	30	29	28	27	26	25	24
			LOOP_FILTER[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
						LOCK_THR[2:0]		
Access						R/W	R/W	R/W
Reset						0	1	0

Bit	15	14	13	12	11	10	9	8
			UTMIBG	UTMIVR	CONTROL[11:8]			
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CONTROL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	0	0	1	1

Bits 29:24 – LOOP_FILTER[5:0] LOOP Filter Selection
Recommended value for this field = 0x1B.

Bits 18:16 – LOCK_THR[2:0] PLL Lock Threshold Value Selection
Recommended value for this field = 0x4

Bit 13 – UTMIBG UPLL Bandgap Control
This bit has no effect when applied to PLLA.

Value	Description
0	The UPLL bandgap is switched off.
1	The UPLL bandgap is switched on.

Bit 12 – UTMIVR UPLL Voltage Regulator Control
This bit has no effect when applied to PLLA.

Value	Description
0	The UPLL voltage regulator is switched off.
1	The UPLL voltage regulator is switched on.

Bits 11:0 – CONTROL[11:0] PLL CONTROL Value Selection
Recommended value for this field = 0x010.
On PLLA, this field controls the DCO analog filters:

Field	Description
CONTROL[1:0]	Analog VCO Filter Selection
CONTROL[4:2]	Process Configuration
CONTROL[6:5]	VCO Gain Configuration

Downloaded from Arrow.com.

.....continued	
Field	**Description**
CONTROL[7]	Offset Frequency Adjustment
CONTROL[8]	External Pad Connection
CONTROL[9]	Test Mode dedicated
CONTROL[10]	DAC Mode
CONTROL[11]	Enable Output Phases

On UPLL, this field controls the following PLL ports:

Field	Description
CONTROL[1:0]	Not used
CONTROL[4:2]	Process Configuration
CONTROL[6:5]	VCO Gain Configuration
CONTROL[7]	Offset Frequency Adjustment
CONTROL[11:8]	Not used

Downloaded from Arrow.com.

28.16.8 PMC PLL Update Register

Name: PMC_PLL_UPDT
Offset: 0x001C
Reset: 0x00030000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	STUPTIM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	1

Bit	15	14	13	12	11	10	9	8
								UPDATE
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
								ID
Access								R/W
Reset								0

Bits 23:16 – STUPTIM[7:0] Startup Time
0: Only the lock of the PLL is considered to know the lock status of the PLL. If the lock of the PLL is not enabled, the lock never rises.
Other values: If PMC_PLL_CTRL0.ENLOCK is low, specifies the startup time of the PLL. If PMC_PLL_CTRL0.ENLOCK is high, specifies how long the LOCK signal of the PLL is masked before being read. The startup time is defined as a number of MD_SLCK cycles and is the same for all PLLs.

Bit 8 – UPDATE PLL Setting Update (write-only)

Value	Description
0	No effect.
1	The PLL configuration written in PMC_PLL_CTRL0 and PMC_PLL_CTRL1 are applied to the PLL defined by the last ID written in the PMC_PLL_CTRL0 register.

Bit 0 – ID PLL ID
When writing a PLL control register (PMC_PLL_CTRLx), this ID specifies which PLL is impacted by written fields.
When reading a PLL control register (PMC_PLL_CTRLx), this ID specifies which PLL fields are read.

Downloaded from Arrow.com.

28.16.9 PMC Clock Generator Main Oscillator Register

Name: CKGR_MOR
Offset: 0x0020
Reset: 0x00000008
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.
Note: Bit 5 is always read at 1.

Bit	31	30	29	28	27	26	25	24
		AUTOCPUSW	AUTOMAINSW			XT32KFME	CFDEN	MOSCSEL
Access		R/W	R/W			R/W	R/W	R/W
Reset		0	0			0	0	0

Bit	23	22	21	20	19	18	17	16
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MOSCXTST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					MOSCRCEN	ULP1		MOSCXTEN
Access					R/W	W		R/W
Reset					1	0		0

Bit 30 – AUTOCPUSW Automatic Processor Clock Source Switching

Value	Description
0	A main crystal oscillator failure detection has no effect on the processor clock source selection.
1	If a main crystal oscillator failure is detected, the processor clock source selection automatically switches to the main clock.

Bit 29 – AUTOMAINSW Automatic Main Clock Source Switching

Value	Description
0	A main crystal oscillator failure detection has no effect on the main clock source selection.
1	If a main crystal oscillator failure is detected, the main clock source selection automatically switches to the main RC.

Bit 26 – XT32KFME 32.768 kHz Crystal Oscillator Frequency Monitoring Enable

Value	Description
0	The 32.768 kHz crystal oscillator frequency monitoring is disabled.
1	The 32.768 kHz crystal oscillator frequency monitoring is enabled.

Bit 25 – CFDEN Clock Failure Detector Enable

Value	Description
0	The clock failure detector is disabled.
1	The clock failure detector is enabled.

Bit 24 – MOSCSEL Main Clock Oscillator Selection

Value	Description
0	The main RC oscillator is selected.
1	The main crystal oscillator is selected.

Downloaded from Arrow.com.

Bits 23:16 – KEY[7:0] Write Access Password

Value	Name	Description
0x37	PASSWD	Writing any other value in this field aborts the write operation. Always reads as 0.

Bits 15:8 – MOSCXTST[7:0] Main Crystal Oscillator Startup Time
Specifies the number of MD_SLCK cycles multiplied by 8 for the main crystal oscillator startup time.

Bit 3 – MOSCRCEN Main RC Oscillator Enable
When MOSCRCEN is set, the MOSCRCS flag is set once the main RC oscillator startup time is achieved.

Value	Description
0	The main RC oscillator is disabled.
1	The main RC oscillator is enabled.

Bit 2 – ULP1 ULP Mode 1 Command

Value	Description
0	No effect.
1	Puts the device in ULP mode 1.

Bit 0 – MOSCXTEN Main Crystal Oscillator Enable
A crystal must be connected between XIN and XOUT or a clock signal must be provided on XIN with XOUT grounded.
When MOSCXTEN is set, the MOSCXTS flag is set once the main crystal oscillator startup time is achieved.

Value	Description
0	The main crystal oscillator is disabled.
1	The main crystal oscillator is enabled or in bypass.

Downloaded from Arrow.com.

28.16.10 PMC Clock Generator Main Clock Frequency Register

Name: CKGR_MCFR
Offset: 0x0024
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								CCSS
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
				RCMEAS				MAINFRDY
Access				R/W				R/W
Reset				0				0

Bit	15	14	13	12	11	10	9	8
	MAINF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MAINF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 24 – CCSS Counter Clock Source Selection

Value	Description
0	The measured clock of the MAINF counter is the main RC oscillator.
1	The measured clock of the MAINF counter is the main crystal oscillator.

Bit 20 – RCMEAS RC Oscillator Frequency Measure (write-only)
The measurement is performed on the main frequency (i.e., not limited to the main RC oscillator only). If the source of MAINCK is the main crystal oscillator, the restart of measurement may not be required because of the stability of crystal oscillators.

Value	Description
0	No effect.
1	Restarts measuring of the frequency of MAINCK. MAINF carries the new frequency as soon as a low-to-high transition occurs on the MAINFRDY flag.

Bit 16 – MAINFRDY Main Clock Frequency Measure Ready
To ensure that a correct value is read on the MAINF field, the MAINFRDY flag must be read at '1' then another read access must be performed on the register to get a stable value on the MAINF field.

Value	Description
0	MAINF value is not valid or the measured oscillator is disabled or a measure has just been started by means of RCMEAS.
1	The measured oscillator has been enabled previously and MAINF value is available.

Bits 15:0 – MAINF[15:0] Main Clock Frequency
Gives the number of cycles of the clock selected by the bit CCSS within 16 MD_SLCK periods. To calculate the frequency of the measured clock:
$f_{SELCLK} = (MAINF \times f_{MD_SLCK})/16$
where frequency is in MHz.

Downloaded from Arrow.com.

28.16.11 PMC CPU Clock Register

Name: PMC_CPU_CKR
Offset: 0x0028
Reset: 0x00000001
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

The CSS, PRES and MDIV fields cannot be modified simultaneously. If more than one field modification is required, proceed sequentially: modify the first field and wait for PMC_SR.MCKRDY low, then modify the second field and wait for PMC_SR.MCKRDY low, etc.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						MDIV[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
		PRES[2:0]					CSS[1:0]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		0	0	0			0	1

Bits 10:8 – MDIV[2:0] MCK Division

Value	Name	Description
0	EQ_PCK	MCK is FCLK divided by 1. MCK_2X is FCLK divided by 1.
1	PCK_DIV2	MCK is FCLK divided by 2. MCK_2X is FCLK divided by 1.
2	PCK_DIV4	MCK is FCLK divided by 4. MCK_2X is FCLK divided by 2.
3	PCK_DIV3	MCK is FCLK divided by 3. MCK_2X is FCLK divided by 1.5.

Bits 6:4 – PRES[2:0] Processor Clock Prescaler

Value	Name	Description
0	CLK_1	Selected clock
1	CLK_2	Selected clock divided by 2
2	CLK_4	Selected clock divided by 4
3	CLK_8	Selected clock divided by 8
4	CLK_16	Selected clock divided by 16
5	CLK_32	Selected clock divided by 32
6	CLK_64	Selected clock divided by 64
7	CLK_3	Selected clock divided by 3

Bits 1:0 – CSS[1:0] MCK Source Selection

Value	Name	Description
0	SLOW_CLK	MD_SLCK is selected.
1	MAIN_CLK	MAINCK is selected.
2	PLLACK	PLLACK is selected.
3	UPLLCK	UPLL is selected.

Downloaded from Arrow.com.

28.16.12 PMC USB Clock Register

Name: PMC_USB
Offset: 0x0038
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					USBDIV[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
							USBS[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 11:8 – USBDIV[3:0] Divider for USB OHCI Clock
USB Clock is Input clock divided by USBDIV + 1.

Bits 1:0 – USBS[1:0] USB OHCI/EHCI Input Clock Selection

Value	Name	Description
0	PLLA	USB Clock Input is PLLACK.
1	UPLL	USB Clock Input is UPLLCK.
2	MAINXTAL	USB Clock Input is MAINXTALCK.
3	Reserved	–

Downloaded from Arrow.com.

28.16.13 PMC Programmable Clock Register

Name: PMC_PCKx
Offset: 0x40 + x*0x04 [x=0..1]
Reset: 0
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				CSS[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 15:8 – PRES[7:0] Programmable Clock Prescaler

Value	Description
0–255	Selected clock is divided by PRES+1.

Bits 4:0 – CSS[4:0] Programmable Clock Source Selection
Values not listed are considered “reserved”.

Value	Name	Description
0	MD_SLOW_CLK	MD_SLCK is selected
1	TD_SLOW_CLOCK	TD_SLCK is selected
2	MAINCK	MAINCK is selected
3	MCK	MCK is selected
4	PLLA	PLLA is selected.
5	UPLL	UPLL is selected.

Downloaded from Arrow.com.

28.16.14 PMC Interrupt Enable Register

Name: PMC_IER
Offset: 0x0060
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PMC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
							PLL_INT	
Access							W	
Reset							–	

Bit	23	22	21	20	19	18	17	16
	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
Access	W		W			W	W	W
Reset	–		–			–	–	–

Bit	15	14	13	12	11	10	9	8
							PCKRDY1	PCKRDY0
Access							W	W
Reset							–	–

Bit	7	6	5	4	3	2	1	0
					MCKRDY			MOSCXTS
Access					W			W
Reset					–			–

Bit 25 – PLL_INT PLL Interrupt Enable

Bit 23 – MCKMON Master Clock Clock Monitor Interrupt Enable

Bit 21 – XT32KERR 32.768 kHz Crystal Oscillator Error Interrupt Enable

Bit 18 – CFDEV Clock Failure Detector Event Interrupt Enable

Bit 17 – MOSCRCS Main RC Oscillator Status Interrupt Enable

Bit 16 – MOSCSELS Main Clock Source Oscillator Selection Status Interrupt Enable

Bits 8, 9 – PCKRDYx Programmable Clock Ready x Interrupt Enable

Bit 3 – MCKRDY Master Clock Ready Interrupt Enable

Bit 0 – MOSCXTS Main Crystal Oscillator Status Interrupt Enable

Downloaded from Arrow.com.

28.16.15 PMC Interrupt Disable Register

Name: PMC_IDR
Offset: 0x0064
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PMC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
							PLL_INT	
Access							W	
Reset							–	

Bit	23	22	21	20	19	18	17	16
	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
Access	W		W			W	W	W
Reset	–		–			–	–	–

Bit	15	14	13	12	11	10	9	8
							PCKRDY1	PCKRDY0
Access							W	W
Reset							–	–

Bit	7	6	5	4	3	2	1	0
					MCKRDY			MOSCXTS
Access					W			W
Reset					–			–

Bit 25 – PLL_INT PLL Interrupt Disable

Bit 23 – MCKMON Master Clock Clock Monitor Interrupt Disable

Bit 21 – XT32KERR 32.768 kHz Crystal Oscillator Error Interrupt Disable

Bit 18 – CFDEV Clock Failure Detector Event Interrupt Disable

Bit 17 – MOSCRCS Main RC Status Interrupt Disable

Bit 16 – MOSCSELS Main Clock Source Oscillator Selection Status Interrupt Disable

Bits 8, 9 – PCKRDYx Programmable Clock Ready x Interrupt Disable

Bit 3 – MCKRDY Master Clock Ready Interrupt Disable

Bit 0 – MOSCXTS Main Crystal Oscillator Status Interrupt Disable

Downloaded from Arrow.com.

28.16.16 PMC Status Register

Name: PMC_SR
Offset: 0x0068
Reset: 0x00030008
Property: Read-only

Bit	31	30	29	28	27	26	25	24
							PLL_INT	GCLKRDY
Access							R	R
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	MCKMON		XT32KERR	FOS	CFDS	CFDEV	MOSCRCS	MOSCSELS
Access	R		R	R	R	R	R	R
Reset	0		0	0	0	0	1	1

Bit	15	14	13	12	11	10	9	8
							PCKRDY1	PCKRDY0
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
					MCKRDY			MOSCXTS
Access					R			R
Reset					1			0

Bit 25 – PLL_INT PLL Interrupt Status

Value	Description
0	No PLL interrupt has occurred.
1	A PLL interrupt has occurred. PLL interrupt is defined by the configuration of the PMC_IMR register.

Bit 24 – GCLKRDY GCLK Ready

Value	Description
0	A GCLK is not ready to use (clock switching in progress).
1	All GCLKs are switched to their selected source clock and ready to use.

Bit 23 – MCKMON Master Clock Clock Monitor Error
This status is cleared on read.

Value	Description
0	The Master Clock is correct or the CPU clock monitor is disabled.
1	The Master Clock is incorrect or has been incorrect for an elapsed period of time since the monitoring has been enabled.

Bit 21 – XT32KERR Slow Crystal Oscillator Error

Value	Description
0	The frequency of the 32.768 kHz crystal oscillator is correct (32.768 kHz ±1%) or the monitoring is disabled.
1	The frequency of the 32.768 kHz crystal oscillator is incorrect or has been incorrect for an elapsed period of time since the monitoring has been enabled.

Bit 20 – FOS Clock Failure Detector Fault Output Status

Value	Description
0	The fault output of the clock failure detector is inactive.
1	The fault output of the clock failure detector is active. This status is cleared by writing a ‘1’ to FOCLR in PMC_FOCR.

Downloaded from Arrow.com.

Bit 19 – CFDS Clock Failure Detector Status

Value	Description
0	A clock failure of the main crystal oscillator clock is not detected.
1	A clock failure of the main crystal oscillator clock is detected.

Bit 18 – CFDEV Clock Failure Detector Event

Value	Description
0	No clock failure detection of the main crystal oscillator clock has occurred since the last read of PMC_SR.
1	At least one clock failure detection of the main crystal oscillator clock has occurred since the last read of PMC_SR.

Bit 17 – MOSCRCS Main RC Oscillator Status

Value	Description
0	Main RC oscillator is not stabilized.
1	Main RC oscillator is stabilized.

Bit 16 – MOSCSELS Main Clock Source Oscillator Selection Status

Value	Description
0	Selection is in progress.
1	Selection is done.

Bits 8, 9 – PCKRDYx Programmable Clock Ready Status

Value	Description
0	Programmable Clock x is not ready.
1	Programmable Clock x is ready.

Bit 3 – MCKRDY Master Clock Status

Value	Description
0	Master Clock is not ready.
1	Master Clock is ready.

Bit 0 – MOSCXTS Main Crystal Oscillator Status

Value	Description
0	Main crystal oscillator is not stabilized.
1	Main crystal oscillator is stabilized.

Downloaded from Arrow.com.

28.16.17 PMC Interrupt Mask Register

Name: PMC_IMR
Offset: 0x006C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
							PLL_INT	
Access							W	
Reset							0	

Bit	23	22	21	20	19	18	17	16
	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
Access	R		R			R	R	R
Reset	0		0			0	0	0

Bit	15	14	13	12	11	10	9	8
							PCKRDY1	PCKRDY0
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
					MCKRDY			MOSCXTS
Access					R			R
Reset					0			0

Bit 25 – PLL_INT PLL Interrupt Mask

Bit 23 – MCKMON Master Clock Monitor Error Interrupt Mask

Bit 21 – XT32KERR 32.768 kHz Crystal Oscillator Error Interrupt Mask

Bit 18 – CFDEV Clock Failure Detector Event Interrupt Mask

Bit 17 – MOSCRCS Main RC Status Interrupt Mask

Bit 16 – MOSCSELS Main Clock Source Oscillator Selection Status Interrupt Mask

Bits 8, 9 – PCKRDYx Programmable Clock Ready x Interrupt Mask

Bit 3 – MCKRDY Master Clock Ready Interrupt Mask

Bit 0 – MOSCXTS Main Crystal Oscillator Status Interrupt Mask

Downloaded from Arrow.com.

28.16.18 PMC Fast Startup Mode Register

Name: PMC_FSMR
Offset: 0x0070
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
							WLAN1	WLAN0
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
						USBAL	RTCAL	RTTAL
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 24, 25 – WLANx Wakeup on LAN[x]

Value	Description
0	The Wakeup on LAN[x] alarm has no effect on the PMC.
1	The Wakeup on LAN[x] alarm enables a fast restart signal to the PMC.

Bit 18 – USBAL USB Alarm Enable

Value	Description
0	The USB alarm has no effect on the PMC.
1	The USB alarm enables a fast restart signal to the PMC.

Bit 17 – RTCAL RTC Alarm Enable

Value	Description
0	The RTC alarm has no effect on the PMC.
1	The RTC alarm enables a fast restart signal to the PMC.

Bit 16 – RTTAL RTT Alarm Enable

Value	Description
0	The RTT alarm has no effect on the PMC.
1	The RTT alarm enables a fast restart signal to the PMC.

Downloaded from Arrow.com.

28.16.19 PMC Wakeup Control Register

Name: PMC_WCR
Offset: 0x0074
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
								CMD
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
							WIP	EN
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					WKPIONB[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 24 – CMD Command

Value	Description
0	Read mode.
1	Write mode.

Bit 17 – WIP Wakeup Input Polarity
Defines the active polarity of the selected wakeup input. If the corresponding wakeup input is enabled at the FSTP level, it enables a fast restart signal.

Bit 16 – EN Wakeup Input Enable

Value	Description
0	The selected wakeup input has no effect on the PMC.
1	The selected wakeup input enables a fast restart signal to the PMC.

Bits 3:0 – WKPIONB[3:0] Wakeup Input Number
Defines which wakeup source is to be modified during a write access (CMD is set to ‘1’) or which wakeup source status is read on the next read access to this register (CMD is set to ‘0’).

Primary Signal Name	WKPIONB
PA2	0
PA9	1
PA10	2
PA28	3
PB0	4
PB3	5

Downloaded from Arrow.com.

.....continued	
Primary Signal Name	**WKPIONB**
PB18	6
PB25	7
PC24	8
PC25	9
PC31	10
PD17	11
PD18	12

Downloaded from Arrow.com.

28.16.20 PMC Fault Output Clear Register

Name: PMC_FOCR
Offset: 0x0078
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								W
Reset								–

Bit 0 – FOCLR Fault Output Clear
Clears the clock failure detector fault output.

Downloaded from Arrow.com.

28.16.21 PMC Write Protection Mode Register

Name: PMC_WPMR
Offset: 0x0080
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
							WPITEN	WPEN
Access							R/W	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x504D43	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 1 – WPITEN Write Protection Interrupt Enable
See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x504D43 (“PMC” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x504D43 (“PMC” in ASCII).

Bit 0 – WPEN Write Protection Enable
See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x504D43 (“PMC” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x504D43 (“PMC” in ASCII).

Downloaded from Arrow.com.

28.16.22 PMC Write Protection Status Register

Name: PMC_WPSR
Offset: 0x0084
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the PMC_WPSR.
1	A write protection violation has occurred since the last read of the PMC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

28.16.23 PMC Peripheral Control Register

Name: PMC_PCR
Offset: 0x0088
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	CMD		GCLKEN	EN	GCLKDIV[7:4]			
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GCLKDIV[3:0]							
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	15	14	13	12	11	10	9	8
				GCLKCSS[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		PID[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 31 – CMD Command

Value	Description
0	Read mode.
1	Write mode.

Bit 29 – GCLKEN Generic Clock Enable

Value	Description
0	The selected generic clock is disabled.
1	The selected generic clock is enabled.

Bit 28 – EN Enable

Value	Description
0	Selected Peripheral clock is disabled.
1	Selected Peripheral clock is enabled.

Bits 27:20 – GCLKDIV[7:0] Generic Clock Division Ratio
Generic clock is the selected clock period divided by GCLKDIV + 1.
GCLKDIV must not be changed while the peripheral selects GCLKx (e.g., bit rate, etc.).

Bits 12:8 – GCLKCSS[4:0] Generic Clock Source Selection

Value	Name	Description
0	MD_SLOW_CLK	MD_SLCK is selected
1	TD_SLOW_CLOCK	TD_SLCK is selected
2	MAINCK	MAINCK is selected
3	MCK	MCK is selected
4	PLLA	PLLA is selected.
5	UPLL	UPLL is selected.

Downloaded from Arrow.com.

Bits 6:0 – PID[6:0] Peripheral ID
Peripheral ID selection.
Refer to the identifiers as defined in the section “Peripheral Identifiers”.

Downloaded from Arrow.com.

28.16.24 PMC Oscillator Calibration Register

Name: PMC_OCR
Offset: 0x008C
Reset: 0x00404040
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SEL12	CAL12[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 23 – SEL12 Selection of Main RC Oscillator Calibration Bits

Value	Description
0	Factory-defined value.
1	Value written by user in CAL12 field of this register.

Bits 22:16 – CAL12[6:0] Main RC Oscillator Calibration Bits
Calibration bits applied to the RC Oscillator. Refer to the section “Electrical Characteristics”.

Downloaded from Arrow.com.

28.16.25 PMC MCK Monitor Limits Register

Name: PMC_MCKLIM
Offset: 0x009C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	MCK_HIGH_IT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MCK_LOW_IT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – MCK_HIGH_IT[7:0] MCK Monitoring High IT Limit
Beyond this limit, the MCK frequency monitor generates an interrupt.

Bits 7:0 – MCK_LOW_IT[7:0] MCK Monitoring Low IT Limit
Below this limit, the MCK frequency monitor generates an interrupt.

Downloaded from Arrow.com.

28.16.26 PMC Peripheral Clock Status Register 0

Name: PMC_CSR0
Offset: 0x00A0
Reset: 0x00000000
Property: Read-only

"PIDx" refers to identifiers as defined in the section "Peripheral Identifiers".

The following configuration values are valid for all listed bit names of this register:

0: The corresponding peripheral clock is disabled.

1: The corresponding peripheral clock is enabled.

Bit	31	30	29	28	27	26	25	24
		PID30	PID29	PID28	PID27	PID26	PID25	PID24
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PID23	PID22		PID20	PID19	PID18	PID17	PID16
Access	R	R		R	R	R	R	R
Reset	0	0		0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PID7	PID6	PID5	PID4	PID3	PID2		
Access	R	R	R	R	R	R		
Reset	0	0	0	0	0	0		

Bits 22, 23, 24, 25, 26, 27, 28, 29, 30 – PIDx Peripheral Clock x Status

Bits 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 – PIDx Peripheral Clock x Status

Downloaded from Arrow.com.

28.16.27 PMC Peripheral Clock Status Register 1

Name: PMC_CSR1
Offset: 0x00A4
Reset: 0x00000000
Property: Read-only

"PIDx" refers to identifiers as defined in the section "Peripheral Identifiers".

The following configuration values are valid for all listed bit names of this register:

0: The corresponding peripheral clock is disabled.

1: The corresponding peripheral clock is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							PID49	
Access							R	
Reset							0	

Bit	15	14	13	12	11	10	9	8
	PID47		PID45	PID44	PID43	PID42	PID41	PID40
Access	R		R	R	R	R	R	R
Reset	0		0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PID39	PID38	PID37	PID36	PID35	PID34	PID33	PID32
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 17 – PIDx Peripheral Clock x Status

Bit 15 – PIDx Peripheral Clock x Status

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 – PIDx Peripheral Clock x Status

Downloaded from Arrow.com.

28.16.28 PMC Generic Clock Status Register 0

Name: PMC_GCSR0
Offset: 0x00C0
Reset: 0x00000000
Property: Read-only

“PIDx” refers to identifiers as defined in the section “Peripheral Identifiers".

The following configuration values are valid for all listed bit names of this register:

0: The corresponding generic clock is disabled.

1: The corresponding generic clock is enabled.

Bit	31	30	29	28	27	26	25	24
						GPID26	GPID25	
Access						R	R	
Reset						0	0	

Bit	23	22	21	20	19	18	17	16
					GPID19		GPID17	GPID16
Access					R		R	R
Reset					0		0	0

Bit	15	14	13	12	11	10	9	8
	GPID15	GPID14	GPID13	GPID12	GPID11	GPID10	GPID9	GPID8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GPID7	GPID6	GPID5					
Access	R	R	R					
Reset	0	0	0					

Bits 25, 26 – GPIDx Generic Clock x Status

Bit 19 – GPIDx Generic Clock x Status

Bits 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 – GPIDx Generic Clock x Status

Downloaded from Arrow.com.

28.16.29 PMC Generic Clock Status Register 1

Name: PMC_GCSR1
Offset: 0x00C4
Reset: 0x00000000
Property: Read-only

“PIDx” refers to identifiers as defined in the section “Peripheral Identifiers".

The following configuration values are valid for all listed bit names of this register:

0: The corresponding generic clock is disabled.

1: The corresponding generic clock is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	GPID47		GPID45			GPID42		
Access	R		R			R		
Reset	0		0			0		

Bit	7	6	5	4	3	2	1	0
			GPID37			GPID34	GPID33	GPID32
Access			R			R	R	R
Reset			0			0	0	0

Bit 15 – GPIDx Generic Clock x Status

Bit 13 – GPIDx Generic Clock x Status

Bit 10 – GPIDx Generic Clock x Status

Bit 5 – GPIDx Generic Clock x Status

Bits 0, 1, 2 – GPIDx Generic Clock x Status

Downloaded from Arrow.com.

28.16.30 PMC PLL Interrupt Enable Register

Name: PMC_PLL_IER
Offset: 0x00E0
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							UNLOCKU	UNLOCKA
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							LOCKU	LOCKA
Access							W	W
Reset							–	–

Bit 17 – UNLOCKU UPLL Unlock Interrupt Enable

Bit 16 – UNLOCKA PLLA Unlock Interrupt Enable

Bit 1 – LOCKU UPLL Lock Interrupt Enable

Bit 0 – LOCKA PLLA Lock Interrupt Enable

Downloaded from Arrow.com.

28.16.31 PMC PLL Interrupt Disable Register

Name: PMC_PLL_IDR
Offset: 0x00E4
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							UNLOCKU	UNLOCKA
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							LOCKU	LOCKA
Access							W	W
Reset							–	–

Bit 17 – UNLOCKU UPLL Unlock Interrupt Disable

Bit 16 – UNLOCKA PLLA Unlock Interrupt Disable

Bit 1 – LOCKU UPLL Lock Interrupt Disable

Bit 0 – LOCKA PLLA Lock Interrupt Disable

Downloaded from Arrow.com.

28.16.32 PMC PLL Interrupt Mask Register

Name: PMC_PLL_IMR
Offset: 0x00E8
Reset: 0x00000000
Property: Read-only

This register can only be written if the WPITEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							UNLOCKU	UNLOCKA
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							LOCKU	LOCKA
Access							R	R
Reset							0	0

Bit 17 – UNLOCKU UPLL Unlock Interrupt Mask

Bit 16 – UNLOCKA PLLA Unlock Interrupt Mask

Bit 1 – LOCKU UPLL Lock Interrupt Mask

Bit 0 – LOCKA PLLA Lock Interrupt Mask

Downloaded from Arrow.com.

28.16.33 PMC PLL Interrupt Status Register 0

Name: PMC_PLL_ISR0
Offset: 0x00EC
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							UNLOCKU	UNLOCKA
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							LOCKU	LOCKA
Access							R	R
Reset							0	0

Bit 17 – UNLOCKU UPLL Unlock Interrupt Status

Value	Description
0	UPLL is not unlocked.
1	UPLL is unlocked. To know the unlock type, the PMC_PISR1 register can be read.

Bit 16 – UNLOCKA PLLA Unlock Interrupt Status

Value	Description
0	PLLA is not unlocked.
1	PLLA is unlocked. To know the unlock type, the PMC_PISR1 register can be read.

Bit 1 – LOCKU UPLL Lock Interrupt Status

Value	Description
0	UPLL is not locked.
1	UPLL is locked.

Bit 0 – LOCKA PLLA Lock Interrupt Status

Value	Description
0	PLLA is not locked.
1	PLLA is locked.

Downloaded from Arrow.com.

28.16.34 PMC PLL Interrupt Status Register 1

Name: PMC_PLL_ISR1
Offset: 0x00F0
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							OVRU	OVRA
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							UDRU	UDRA
Access							R	R
Reset							0	0

Bit 17 – OVRU UPLL Overflow

Value	Description
0	UPLL is not in overflow state.
1	UPLL has encountered an overflow.

Bit 16 – OVRA PLLA Overflow

Value	Description
0	PLLA is not in overflow state.
1	PLLA has encountered an overflow.

Bit 1 – UDRU UPLL Underflow

Value	Description
0	UPLL is not in underflow state.
1	UPLL has encountered an underflow.

Bit 0 – UDRA PLLA Underflow

Value	Description
0	PLLA is not in underflow state.
1	PLLA has encountered an underflow.

Downloaded from Arrow.com.

29. Parallel Input/Output Controller (PIO)

29.1 Description

The Parallel Input/Output Controller (PIO) manages up to 32 fully programmable input/output lines. Each I/O line may be dedicated as a general-purpose I/O or be assigned to a function of an embedded peripheral. This ensures effective optimization of the pins of the product.

Each I/O line is associated with a bit number in all of the 32-bit registers of the 32-bit wide user interface.

Each I/O line of the PIO Controller features the following:

- An input change interrupt enabling level change detection on any I/O line
- Additional Interrupt modes enabling rising edge, falling edge, low-level or high-level detection on any I/O line
- A glitch filter providing rejection of glitches lower than one-half of peripheral clock cycle
- A debouncing filter providing rejection of unwanted pulses from key or push button operations
- Multi-drive capability similar to an open drain I/O line
- Control of the I/O line pullup and pulldown
- Input visibility and output control

The PIO Controller also features a synchronous output providing up to 32 bits of data output in a single write operation.

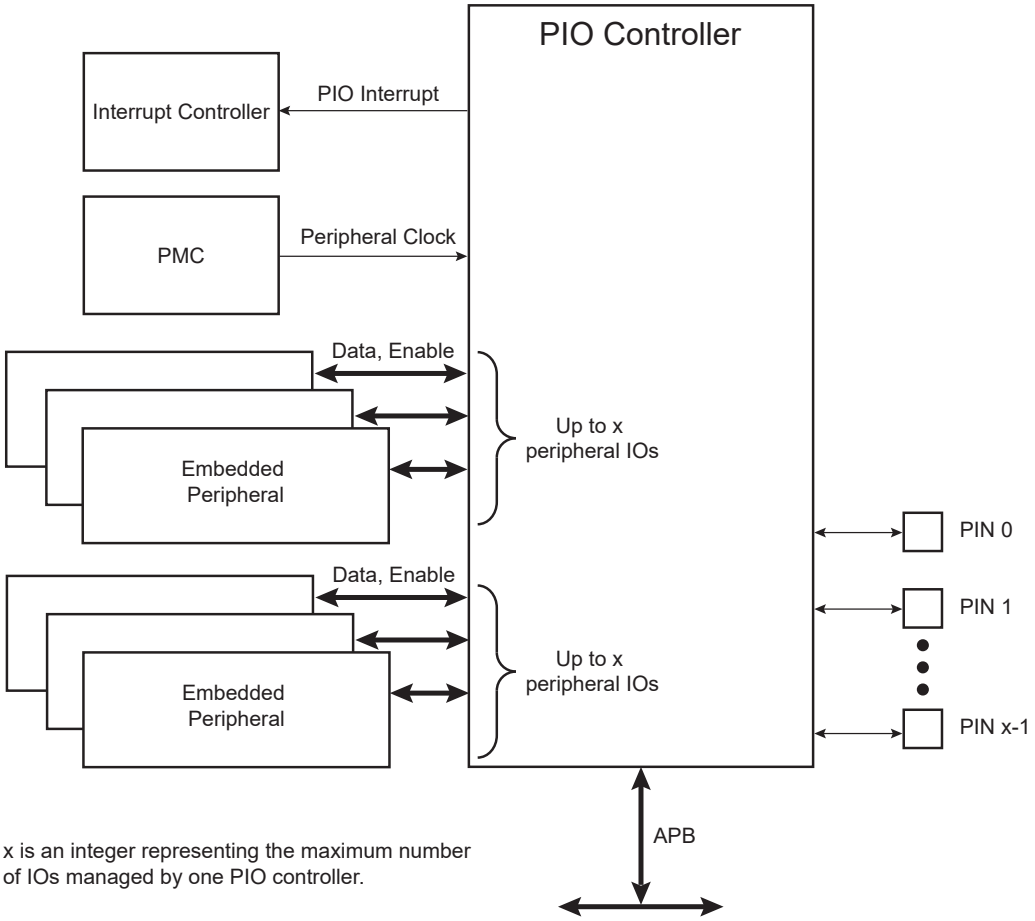
29.2 Embedded Characteristics

- Up to 32 Programmable I/O Lines
- Fully Programmable through Set/Clear Registers
- Multiplexing of Four Peripheral Functions per I/O Line
- For each I/O Line (Whether Assigned to a Peripheral or Used as General Purpose I/O)
  - Input Change Interrupt
  - Programmable Glitch Filter
  - Programmable Debouncing Filter
  - Multi-drive Option Enables Driving in Open Drain
  - Programmable Pullup on Each I/O Line
  - Pin Data Status Register, Supplies Visibility of the Level on the Pin at Any Time
  - Additional Interrupt Modes on a Programmable Event: Rising Edge, Falling Edge, Low-Level or High-Level
- Synchronous Output, Provides Set and Clear of Several I/O Lines in a Single Write
- Register Write Protection
- Programmable Schmitt Trigger Inputs
- Programmable Slewrate per I/O Line
- Programmable I/O Drive

Downloaded from Arrow.com.

29.3 Block Diagram

Figure 29-1. Block Diagram



29.4 Product Dependencies

29.4.1 Pin Multiplexing

Each pin is configurable, depending on the product, as either a general-purpose I/O line only, or as an I/O line multiplexed with one or two peripheral I/Os. As the multiplexing is hardware defined and thus product-dependent, the hardware designer and programmer must carefully determine the configuration of the PIO Controllers required by their application. When an I/O line is general-purpose only, i.e., not multiplexed with any peripheral I/O, programming of the PIO Controller regarding the assignment to a peripheral has no effect and only the PIO Controller can control how the pin is driven by the product.

29.4.2 External Interrupt Lines

The interrupt signals FIQ and IRQ0 to IRQn are generally multiplexed through the PIO Controllers. However, it is not necessary to assign the I/O line to the interrupt function as the PIO Controller has no effect on inputs and the external interrupt lines are used only as inputs.

When the WKUPx input pins must be used as external interrupt lines, the PIO Controller must be configured to disable the peripheral control on these IOs, and the corresponding IO lines must be set to Input mode.

Downloaded from Arrow.com.

29.4.3 Power Management

The Power Management Controller controls the peripheral clock in order to save power. Writing any of the registers of the user interface does not require the peripheral clock to be enabled. This means that the configuration of the I/O lines does not require the peripheral clock to be enabled.

However, when the clock is disabled, not all of the features of the PIO Controller are available, including glitch filtering. Note that the input change interrupt, the interrupt modes on a programmable event and the read of the pin level require the clock to be validated.

After a hardware reset, the peripheral clock is disabled by default.

The user must configure the Power Management Controller before any access to the input line information.

29.4.4 Interrupt Sources

For interrupt handling, the PIO Controllers are considered as user peripherals. This means that the PIO Controller interrupt lines are connected among the interrupt sources. Refer to the PIO Controller peripheral identifier in the Peripheral Identifiers table to identify the interrupt sources dedicated to the PIO Controllers. Using the PIO Controller requires the Interrupt Controller to be programmed first.

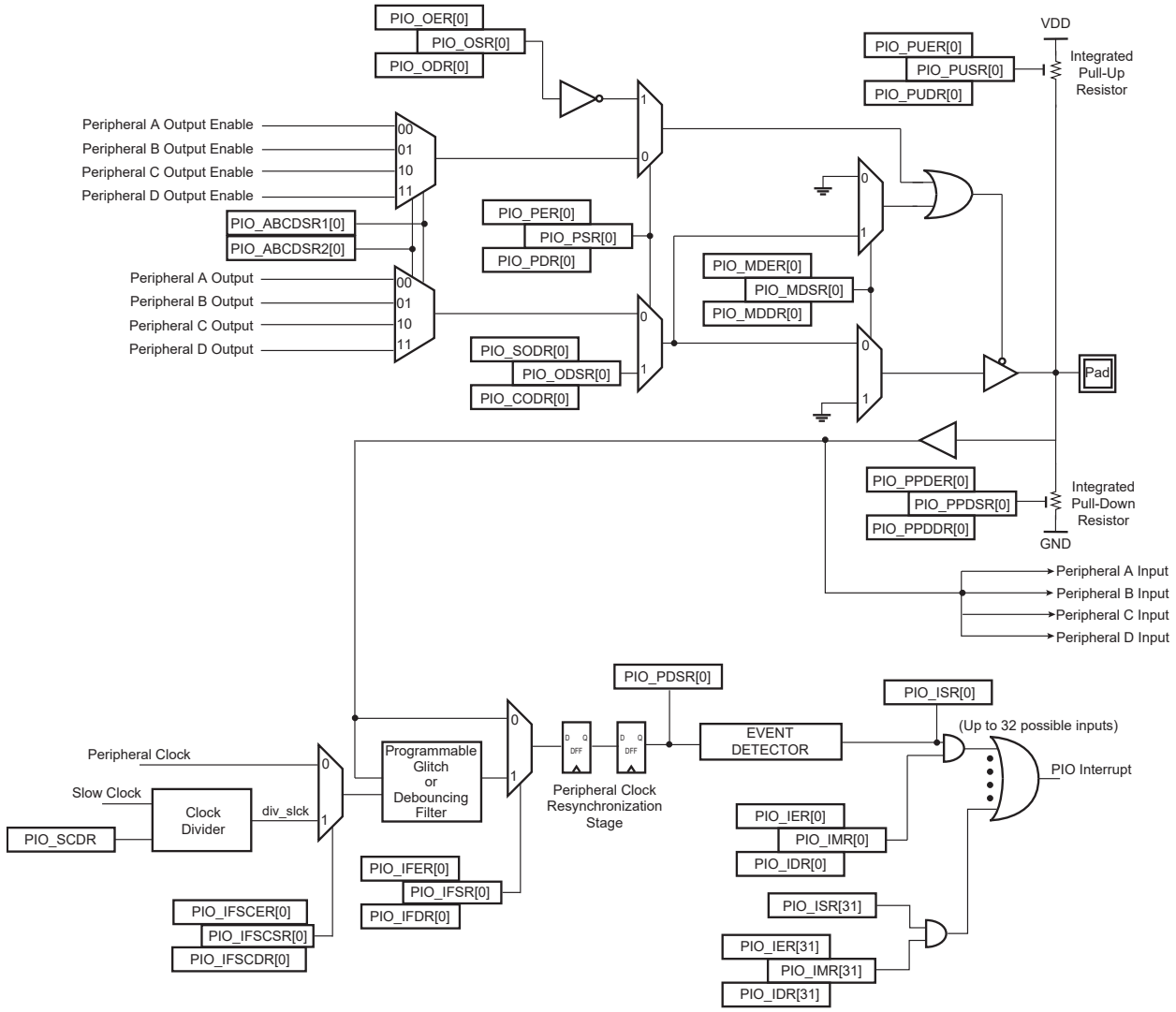
The PIO Controller interrupt can be generated only if the peripheral clock is enabled.

29.5 Functional Description

The PIO Controller features up to 32 fully-programmable I/O lines. Most of the control logic associated to each I/O is represented in the following figure. In this description each signal shown represents one of up to 32 possible indexes.

Downloaded from Arrow.com.

Figure 29-2. I/O Line Control Logic



29.5.1 Pullup and Pulldown Resistor Control

Each I/O line is designed with an embedded pullup resistor and an embedded pulldown resistor. The pullup resistor can be enabled or disabled by writing to the Pull-Up Enable Register (PIO_PUER) or Pull-Up Disable Register (PIO_PUDR), respectively. Writing to these registers results in setting or clearing the corresponding bit in the Pull-Up Status Register (PIO_PUSR). Reading a one in PIO_PUSR means the pullup is disabled and reading a zero means the pullup is enabled. The pulldown resistor can be enabled or disabled by writing the Pull-Down Enable Register (PIO_PPDER) or the Pull-Down Disable Register (PIO_PPDDR), respectively. Writing in these registers results in setting or clearing the corresponding bit in the Pull-Down Status Register (PIO_PPDSR). Reading a one in PIO_PPDSR means the pullup is disabled and reading a zero means the pulldown is enabled.

Enabling the pulldown resistor while the pullup resistor is still enabled is not possible. In this case, the write of PIO_PPDER for the relevant I/O line is discarded. Likewise, enabling the pullup resistor while the pulldown resistor is still enabled is not possible. In this case, the write of PIO_PUER for the relevant I/O line is discarded.

Control of the pullup resistor is possible regardless of the configuration of the I/O line.

After reset, depending on the I/O, pullup or pulldown can be set.

Downloaded from Arrow.com.

29.5.2 I/O Line or Peripheral Function Selection

When a pin is multiplexed with one or two peripheral functions, the selection is controlled with the Enable Register (PIO_PER) and the Disable Register (PIO_PDR). The Status Register (PIO_PSR) is the result of the set and clear registers and indicates whether the pin is controlled by the corresponding peripheral or by the PIO Controller. A value of zero indicates that the pin is controlled by the corresponding on-chip peripheral selected in the Peripheral ABCD Select registers (PIO_ABCDSR1 and PIO_ABCDSR2). A value of one indicates the pin is controlled by the PIO Controller.

If a pin is used as a general-purpose I/O line (not multiplexed with an on-chip peripheral), PIO_PER and PIO_PDR have no effect and PIO_PSR returns a one for the corresponding bit.

After reset, the I/O lines are controlled by the PIO Controller, i.e., PIO_PSR resets at one. However, in some events, it is important that PIO lines are controlled by the peripheral (as in the case of memory chip select lines that must be driven inactive after reset, or for address lines that must be driven low for booting out of an external memory). Thus, the reset value of PIO_PSR is defined at the product level and depends on the multiplexing of the device.

29.5.3 Peripheral A or B or C or D Selection

The PIO Controller provides multiplexing of up to four peripheral functions on a single pin. The selection is performed by writing PIO_ABCDSR1 and PIO_ABCDSR2.

For each pin:

- The corresponding bit at level zero in PIO_ABCDSR1 and the corresponding bit at level zero in PIO_ABCDSR2 means peripheral A is selected.
- The corresponding bit at level one in PIO_ABCDSR1 and the corresponding bit at level zero in PIO_ABCDSR2 means peripheral B is selected.
- The corresponding bit at level zero in PIO_ABCDSR1 and the corresponding bit at level one in PIO_ABCDSR2 means peripheral C is selected.
- The corresponding bit at level one in PIO_ABCDSR1 and the corresponding bit at level one in PIO_ABCDSR2 means peripheral D is selected.

Note that multiplexing of peripheral lines A, B, C and D only affects the output line. The peripheral input lines are always connected to the pin input (see Figure 29-2).

Writing in PIO_ABCDSR1 and PIO_ABCDSR2 manages the multiplexing regardless of the configuration of the pin. However, assignment of a pin to a peripheral function requires a write in PIO_ABCDSR1 and PIO_ABCDSR2 in addition to a write in PIO_PDR.

After reset, PIO_ABCDSR1 and PIO_ABCDSR2 are zero, thus indicating that all the PIO lines are configured on peripheral A. However, peripheral A generally does not drive the pin as the PIO Controller resets in I/O Line mode.

If the software selects a peripheral A, B, C or D which does not exist for a pin, no alternate functions are enabled for this pin and the selection is taken into account. The PIO Controller does not carry out checks to prevent selection of a peripheral which does not exist.

29.5.4 Output Control

When the I/O line is assigned to a peripheral function, i.e., the corresponding bit in PIO_PSR is at zero, the drive of the I/O line is controlled by the peripheral. Peripheral A or B or C or D depending on the value in PIO_ABCDSR1 and PIO_ABCDSR2 determines whether the pin is driven or not.

When the I/O line is controlled by the PIO Controller, the pin can be configured to be driven. This is done by writing the Output Enable Register (PIO_OER) and Output Disable Register (PIO_ODR). The results of these write operations are detected in the Output Status Register (PIO_OSR). When a bit in this register is at zero, the corresponding I/O line is used as an input only. When the bit is at one, the corresponding I/O line is driven by the PIO Controller.

The level driven on an I/O line can be determined by writing in the Set Output Data Register (PIO_SODR) and the Clear Output Data Register (PIO_CODR). These write operations, respectively, set and clear the Output Data Status Register (PIO_ODSR), which represents the data driven on the I/O lines. Writing in PIO_OER and PIO_ODR manages PIO_OSR whether the pin is configured to be controlled by the PIO Controller or assigned to a peripheral function. This enables configuration of the I/O line prior to setting it to be managed by the PIO Controller.

Downloaded from Arrow.com.

Similarly, writing in PIO_SODR and PIO_CODR affects PIO_ODSR. This is important as it defines the first level driven on the I/O line.

29.5.5 Synchronous Data Output

Clearing one or more PIO line(s) and setting another one or more PIO line(s) synchronously cannot be done by using PIO_SODR and PIO_CODR. It requires two successive write operations into two different registers. To overcome this, the PIO Controller offers a direct control of PIO outputs by single write access to PIO_ODSR. Only bits unmasked by the Output Write Status Register (PIO_OWSR) are written. The mask bits in PIO_OWSR are set by writing to the Output Write Enable Register (PIO_OWER) and cleared by writing to the Output Write Disable Register (PIO_OWDR).

After reset, the synchronous data output is disabled on all the I/O lines as PIO_OWSR resets at 0x0.

29.5.6 Multi-Drive Control (Open Drain)

Each I/O can be independently programmed in open drain by using the multi-drive feature. This feature permits several drivers to be connected on the I/O line which is driven low only by each device. An external pullup resistor (or enabling of the internal one) is generally required to guarantee a high level on the line.

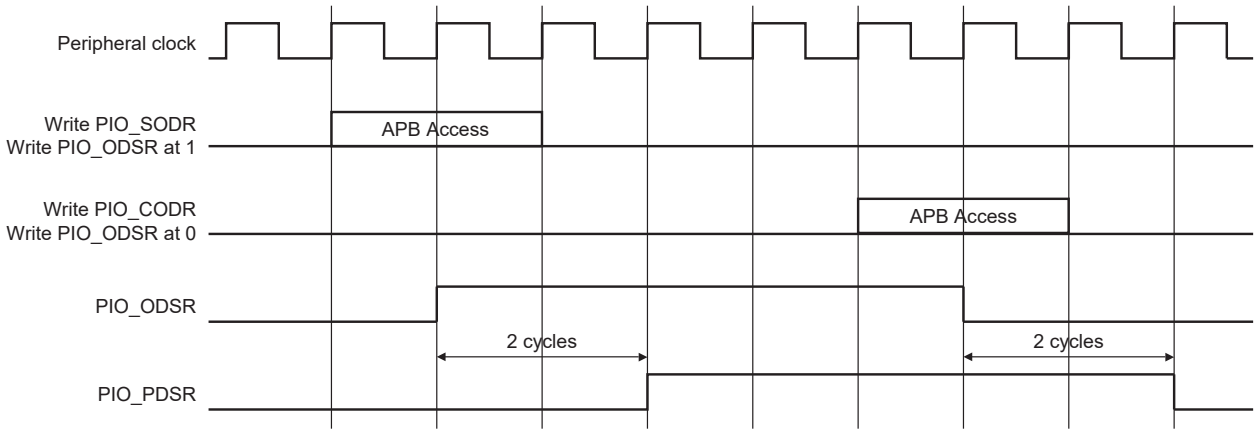
The multi-drive feature is controlled by the Multi-driver Enable Register (PIO_MDER) and the Multi-driver Disable Register (PIO_MDDR). The multi-drive can be selected whether the I/O line is controlled by the PIO Controller or assigned to a peripheral function. The Multi-driver Status Register (PIO_MDSR) indicates the pins that are configured to support external drivers.

After reset, the multi-drive feature is disabled on all pins, i.e., PIO_MDSR resets at value 0x0.

29.5.7 Output Line Timings

The following figure shows how the outputs are driven either by writing PIO_SODR or PIO_CODR, or by directly writing PIO_ODSR. This last case is valid only if the corresponding bit in PIO_OWSR is set. The Output Line Timings figure also shows when the feedback in the Pin Data Status Register (PIO_PDSR) is available.

Figure 29-3. Output Line Timings



29.5.8 Inputs

The level on each I/O line can be read through PIO_PDSR. This register indicates the level of the I/O lines regardless of their configuration, whether uniquely as an input, or driven by the PIO Controller, or driven by a peripheral.

Reading the I/O line levels requires the clock of the PIO Controller to be enabled, otherwise PIO_PDSR reads the levels present on the I/O line at the time the clock was disabled.

29.5.9 Input Glitch and Debouncing Filters

Optional input glitch and debouncing filters are independently programmable on each I/O line.

The glitch filter can filter a glitch with a duration of less than 1/2 peripheral clock and the debouncing filter can filter a pulse of less than 1/2 period of a programmable divided slow clock.

Downloaded from Arrow.com.

The selection between glitch filtering or debounce filtering is done by writing in the PIO Input Filter Slow Clock Disable Register (PIO_IFSCDR) and the PIO Input Filter Slow Clock Enable Register (PIO_IFSCER). Writing PIO_IFSCDR and PIO_IFSCER, respectively, sets and clears bits in the Input Filter Slow Clock Status Register (PIO_IFSCSR).

The current selection status can be checked by reading the PIO_IFSCSR.

- If PIO_IFSCSR[i] = 0: The glitch filter can filter a glitch with a duration of less than 1/2 master clock period.
- If PIO_IFSCSR[i] = 1: The debouncing filter can filter a pulse with a duration of less than 1/2 programmable divided slow clock period.

For the debouncing filter, the period of the divided slow clock is defined by writing in the DIV field of the Slow Clock Divider Debouncing Register (PIO_SCDR):

$t_{div_slck} = ((DIV + 1) \times 2) \times t_{slck}$

When the glitch or debouncing filter is enabled, a glitch or pulse with a duration of less than 1/2 selected clock cycle (selected clock represents peripheral clock or divided slow clock depending on PIO_IFSCDR and PIO_IFSCER programming) is automatically rejected, while a pulse with a duration of one selected clock (peripheral clock or divided slow clock) cycle or more is accepted. For pulse durations between 1/2 selected clock cycle and one selected clock cycle, the pulse may or may not be taken into account, depending on the precise timing of its occurrence. Thus for a pulse to be visible, it must exceed one selected clock cycle, whereas for a glitch to be reliably filtered out, its duration must not exceed 1/2 selected clock cycle.

The filters also introduce some latencies, illustrated in the following two figures.

The glitch filters are controlled by the Input Filter Enable Register (PIO_IFER), the Input Filter Disable Register (PIO_IFDR) and the Input Filter Status Register (PIO_IFSR). Writing PIO_IFER and PIO_IFDR respectively sets and clears bits in PIO_IFSR. This last register enables the glitch filter on the I/O lines.

When the glitch and/or debouncing filter is enabled, it does not modify the behavior of the inputs on the peripherals. It acts only on the value read in PIO_PDSR and on the input change interrupt detection. The glitch and debouncing filters require that the peripheral clock is enabled.

Figure 29-4. Input Glitch Filter Timing

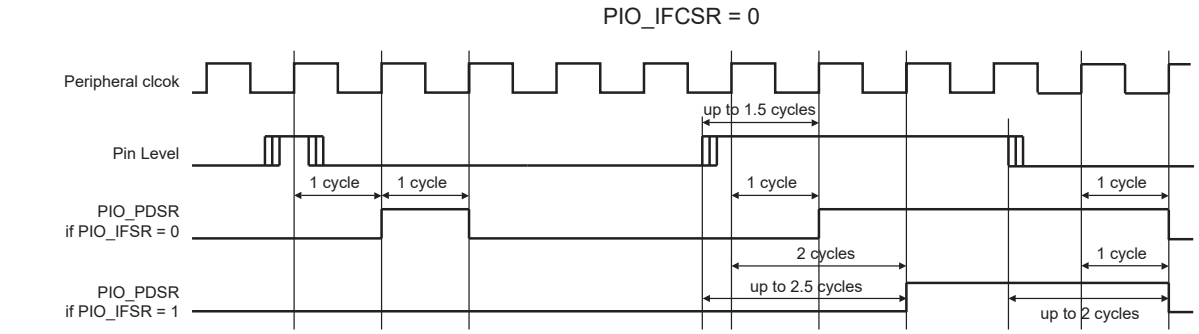
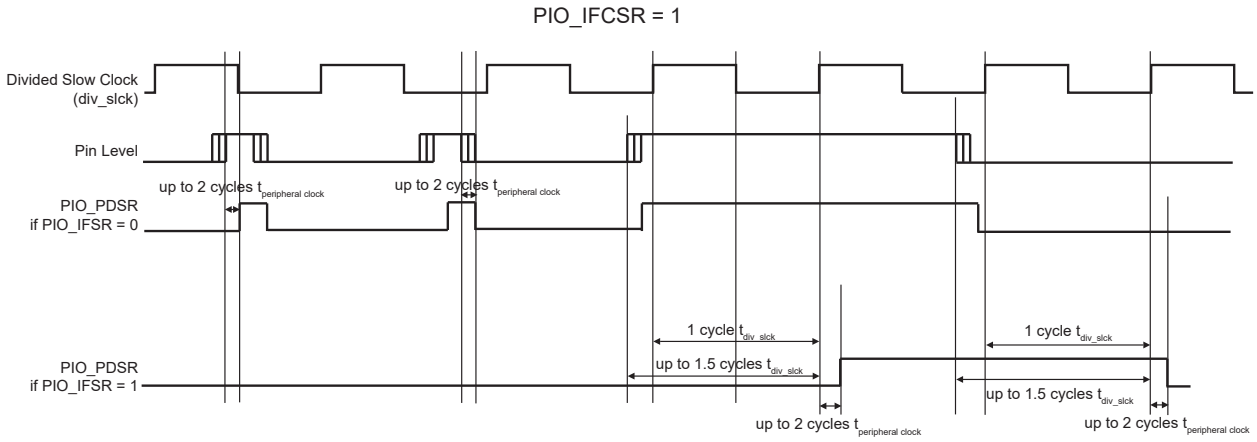


Figure 29-5. Input Debouncing Filter Timing



Downloaded from Arrow.com.

29.5.10 Input Edge/Level Interrupt

The PIO Controller can be programmed to generate an interrupt when it detects an edge or a level on an I/O line. The Input Edge/Level interrupt is controlled by writing the Interrupt Enable Register (PIO_IER) and the Interrupt Disable Register (PIO_IDR), which enable and disable the input change interrupt respectively by setting and clearing the corresponding bit in the Interrupt Mask Register (PIO_IMR). As input change detection is possible only by comparing two successive samplings of the input of the I/O line, the peripheral clock must be enabled. The Input Change interrupt is available regardless of the configuration of the I/O line, i.e., configured as an input only, controlled by the PIO Controller or assigned to a peripheral function.

By default, the interrupt can be generated at any time an edge is detected on the input.

Some additional interrupt modes can be enabled/disabled by writing in the Additional Interrupt Modes Enable Register (PIO_AIMER) and Additional Interrupt Modes Disable Register (PIO_AIMDR). The current state of this selection can be read through the Additional Interrupt Modes Mask Register (PIO_AIMMR).

These additional modes are:

- Rising edge detection
- Falling edge detection
- Low-level detection
- High-level detection

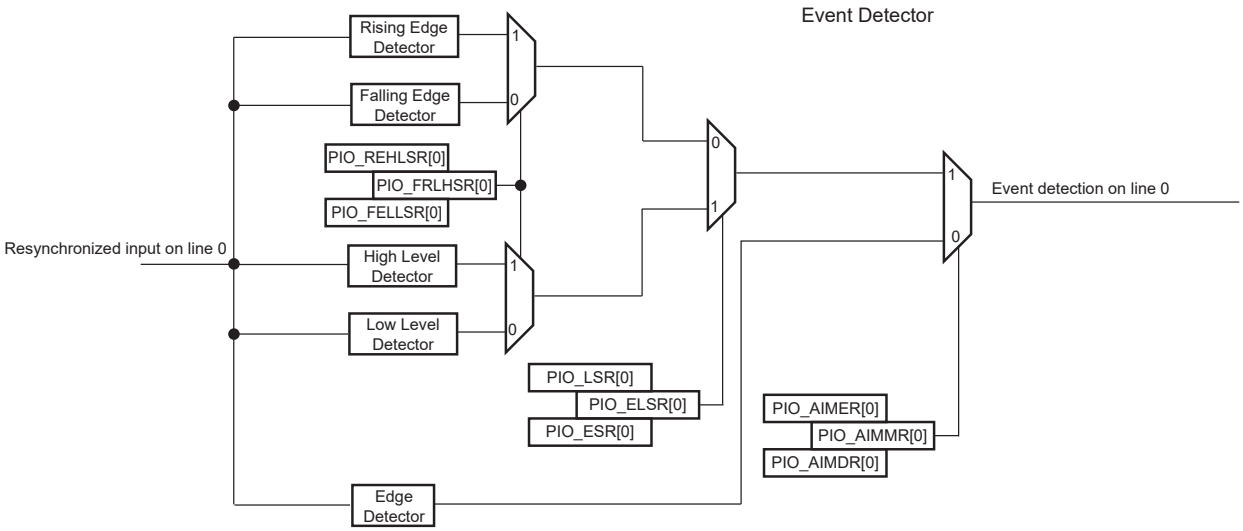
In order to select an additional interrupt mode:

- The type of event detection (edge or level) must be selected by writing in the Edge Select Register (PIO_ESR) and Level Select Register (PIO_LSR) which select, respectively, the edge and level detection. The current status of this selection is accessible through the Edge/Level Status Register (PIO_ELSR).
- The polarity of the event detection (rising/falling edge or high/low-level) must be selected by writing in the Falling Edge/Low-Level Select Register (PIO_FELLSR) and Rising Edge/High-Level Select Register (PIO_REHLSR) which allow to select falling or rising edge (if edge is selected in PIO_ELSR) edge or high- or low-level detection (if level is selected in PIO_ELSR). The current status of this selection is accessible through the Fall/Rise - Low/High Status Register (PIO_FRLHSR).

When an input edge or level is detected on an I/O line, the corresponding bit in the Interrupt Status Register (PIO_ISR) is set. If the corresponding bit in PIO_IMR is set, the PIO Controller interrupt line is asserted.The interrupt signals of the 32 channels are ORed-wired together to generate a single interrupt signal to the interrupt controller.

When the software reads PIO_ISR, all the interrupts are automatically cleared. This signifies that all the interrupts that are pending when PIO_ISR is read must be handled. When an Interrupt is enabled on a “level”, the interrupt is generated as long as the interrupt source is not cleared, even if some read accesses in PIO_ISR are performed.

Figure 29-6. Event Detector on Input Lines (Figure Represents Line 0)



Example of interrupt generation on following lines:

Downloaded from Arrow.com.

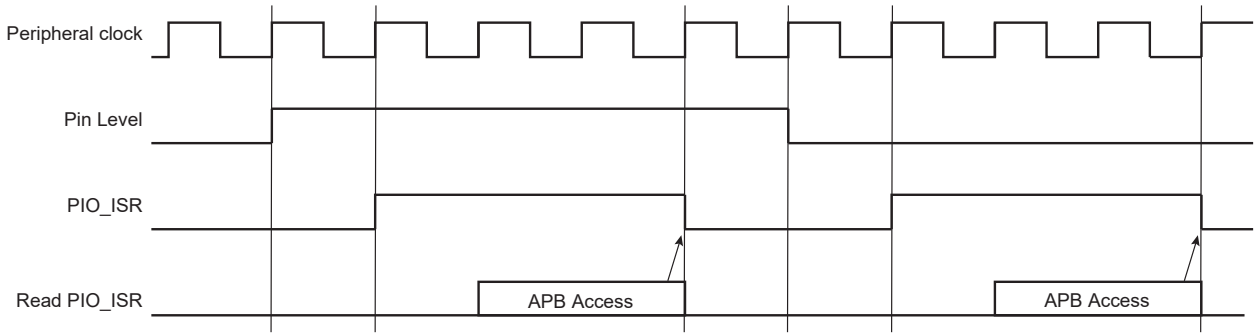
- Rising edge on PIO line 0
- Falling edge on PIO line 1
- Rising edge on PIO line 2
- Low-level on PIO line 3
- High-level on PIO line 4
- High-level on PIO line 5
- Falling edge on PIO line 6
- Rising edge on PIO line 7
- Any edge on the other lines

The following table provides the required configuration for this example.

Table 29-1. Configuration for Example Interrupt Generation

Configuration	Description
Interrupt Mode	All the interrupt sources are enabled by writing 32'hFFFF_FFFF in PIO_IER. Then the additional Interrupt mode is enabled for lines 0 to 7 by writing 32'h0000_00FF in PIO_AIMER.
Edge or Level Detection	Lines 3, 4 and 5 are configured in level detection by writing 32'h0000_0038 in PIO_LSR. The other lines are configured in edge detection by default, if they have not been previously configured. Otherwise, lines 0, 1, 2, 6 and 7 must be configured in edge detection by writing 32'h0000_00C7 in PIO_ESR.
Falling/Rising Edge or Low/High-Level Detection	Lines 0, 2, 4, 5 and 7 are configured in rising edge or high-level detection by writing 32'h0000_00B5 in PIO_REHLSR. The other lines are configured in falling edge or low-level detection by default if they have not been previously configured. Otherwise, lines 1, 3 and 6 must be configured in falling edge/low-level detection by writing 32'h0000_004A in PIO_FELLSR.

Figure 29-7. Input Change Interrupt Timings When No Additional Interrupt Modes



29.5.11 Programmable Schmitt Trigger

It is possible to configure each input for the Schmitt trigger. By default the Schmitt trigger is active. Disabling the Schmitt trigger is requested when using the QTouch® Library.

29.5.12 I/O Lines Programming Example

The programming example shown in the following table is used to obtain the following configuration:

- 4-bit output port on I/O lines 0 to 3 (should be written in a single write operation), open-drain, with pullup resistor
- Four output signals on I/O lines 4 to 7 (to drive LEDs for example), driven high and low, no pullup resistor, no pulldown resistor
- Four input signals on I/O lines 8 to 11 (to read push-button states for example), with pullup resistors, glitch filters and input change interrupts

Downloaded from Arrow.com.

- Four input signals on I/O line 12 to 15 to read an external device status (polled, thus no input change interrupt), no pullup resistor, no glitch filter
- I/O lines 16 to 19 assigned to peripheral A functions with pullup resistor
- I/O lines 20 to 23 assigned to peripheral B functions with pulldown resistor
- I/O lines 24 to 27 assigned to peripheral C with input change interrupt, no pullup resistor and no pulldown resistor
- I/O lines 28 to 31 assigned to peripheral D, no pullup resistor and no pulldown resistor

Table 29-2. Programming Example

Register	Value to be Written
PIO_PER	0x0000_FFFF
PIO_PDR	0xFFFF_0000
PIO_OER	0x0000_00FF
PIO_ODR	0xFFFF_FF00
PIO_IFER	0x0000_0F00
PIO_IFDR	0xFFFF_F0FF
PIO_SODR	0x0000_0000
PIO_CODR	0x0FFF_FFFF
PIO_IER	0x0F00_0F00
PIO_IDR	0xF0FF_F0FF
PIO_MDER	0x0000_000F
PIO_MDDR	0xFFFF_FFF0
PIO_PUDR	0xFFF0_00F0
PIO_PUER	0x000F_FF0F
PIO_PPDDR	0xFF0F_FFFF
PIO_PPDER	0x00F0_0000
PIO_ABCDSR1	0xF0F0_0000
PIO_ABCDSR2	0xFF00_0000
PIO_OWER	0x0000_000F
PIO_OWDR	0x0FFF_FFF0

29.5.13 Register Write Protection

To prevent any single software error from corrupting PIO behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the PIO Write Protection Mode Register (PIO_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the PIO Write Protection Status Register (PIO_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the PIO_WPSR.

The following registers can be write-protected:

- PIO Enable Register
- PIO Disable Register
- PIO Output Enable Register
- PIO Output Disable Register
- PIO Input Filter Enable Register

Downloaded from Arrow.com.

- PIO Input Filter Disable Register
- PIO Multi-driver Enable Register
- PIO Multi-driver Disable Register
- PIO Pull-Up Disable Register
- PIO Pull-Up Enable Register
- PIO Peripheral ABCD Select Register 1
- PIO Peripheral ABCD Select Register 2
- PIO Output Write Enable Register
- PIO Output Write Disable Register
- PIO Pad Pull-Down Disable Register
- PIO Pad Pull-Down Enable Register

Downloaded from Arrow.com.

29.6 Register Summary

Each I/O line controlled by the PIO Controller is associated with a bit in each of the PIO Controller User Interface registers. Each register is 32 bits wide. If a parallel I/O line is not defined, writing to the corresponding bits has no effect. Undefined bits read zero. If the I/O line is not multiplexed with any peripheral, the I/O line is controlled by the PIO Controller and PIO_PSR returns one systematically.

Offset	Name	Bit Pos.								
0x00	PIO_PER	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x04	PIO_PDR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x08	PIO_PSR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x0C ... 0x0F	Reserved									
0x10	PIO_OER	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x14	PIO_ODR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x18	PIO_OSR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1C ... 0x1F	Reserved									
0x20	PIO_IFER	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x24	PIO_IFDR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x28	PIO_IFSR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x2C ... 0x2F	Reserved									
0x30	PIO_SODR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x34	PIO_CODR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x38	PIO_ODSR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x3C	PIO_PDSR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x40	PIO_IER	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x44	PIO_IDR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x48	PIO_IMR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x4C	PIO_ISR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x50	PIO_MDER	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x54	PIO_MDDR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x58	PIO_MDSR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x5C ... 0x5F	Reserved									
0x60	PIO_PUDR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x64	PIO_PUER	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x68	PIO_PUSR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x6C ... 0x6F	Reserved									

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x70	PIO_ABCDSR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x74	PIO_ABCDSR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x78 ... 0x7F	Reserved									
0x80	PIO_IFSCDR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x84	PIO_IFSCER	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x88	PIO_IFSCSR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x8C	PIO_SCDR	31:24								
		23:16								
		15:8			DIV[13:8]					
		7:0	DIV[7:0]							
0x90	PIO_PPDDR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x94	PIO_PPDER	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x98	PIO_PPDSR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x9C ... 0x9F	Reserved									
0xA0	PIO_OWER	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xA4	PIO_OWDR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xA8	PIO_OWSR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xAC ... 0xAF	Reserved									

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0xB0	PIO_AIMER	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xB4	PIO_AIMDR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xB8	PIO_AIMMR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xBC ... 0xBF	Reserved									
0xC0	PIO_ESR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xC4	PIO_LSR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xC8	PIO_ELSR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xCC ... 0xCF	Reserved									
0xD0	PIO_FELLSR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xD4	PIO_REHLSR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xD8	PIO_FRLHSR	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xDC ... 0xE3	Reserved									
0xE4	PIO_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0xE8	PIO_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS
0xEC ... 0xFF	Reserved									

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x0100	PIO_SCHMITT	31:24	SCHMITT31	SCHMITT30	SCHMITT29	SCHMITT28	SCHMITT27	SCHMITT26	SCHMITT25	SCHMITT24
		23:16	SCHMITT23	SCHMITT22	SCHMITT21	SCHMITT20	SCHMITT19	SCHMITT18	SCHMITT17	SCHMITT16
		15:8	SCHMITT15	SCHMITT14	SCHMITT13	SCHMITT12	SCHMITT11	SCHMITT10	SCHMITT9	SCHMITT8
		7:0	SCHMITT7	SCHMITT6	SCHMITT5	SCHMITT4	SCHMITT3	SCHMITT2	SCHMITT1	SCHMITT0
0x0104 ... 0x010F	Reserved									
0x0110	PIO_SLEWR	31:24	SR31	SR30	SR29	SR28	SR27	SR26	SR25	SR24
		23:16	SR23	SR22	SR21	SR20	SR19	SR18	SR17	SR16
		15:8	SR15	SR14	SR13	SR12	SR11	SR10	SR9	SR8
		7:0	SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
0x0114 ... 0x0117	Reserved									
0x0118	PIO_DRIVER1	31:24	DR31	DR30	DR29	DR28	DR27	DR26	DR25	DR24
		23:16	DR23	DR22	DR21	DR20	DR19	DR18	DR17	DR16
		15:8	DR15	DR14	DR13	DR12	DR11	DR10	DR9	DR8
		7:0	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0

Downloaded from Arrow.com.

29.6.1 PIO Enable Register

Name: PIO_PER
Offset: 0x0000
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Enable

Value	Description
0	No effect.
1	Enables the PIO to control the corresponding pin (disables peripheral control of the pin).

Downloaded from Arrow.com.

29.6.2 PIO Disable Register

Name: PIO_PDR
Offset: 0x0004
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Disable

Value	Description
0	No effect.
1	Disables the PIO from controlling the corresponding pin (enables peripheral control of the pin).

Downloaded from Arrow.com.

29.6.3 PIO Status Register

Name: PIO_PSR
Offset: 0x0008
Property: Read-only

Reset values depend on the product implementation.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset								

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset								

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset								

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset								

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Status

Value	Description
0	PIO is inactive on the corresponding I/O line (peripheral is active).
1	PIO is active on the corresponding I/O line (peripheral is inactive).

Downloaded from Arrow.com.

29.6.4 PIO Output Enable Register

Name: PIO_OER
Offset: 0x0010
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Output Enable

Value	Description
0	No effect.
1	Enables the output on the I/O line.

Downloaded from Arrow.com.

29.6.5 PIO Output Disable Register

Name: PIO_ODR
Offset: 0x0014
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Output Disable

Value	Description
0	No effect.
1	Disables the output on the I/O line.

Downloaded from Arrow.com.

29.6.6 PIO Output Status Register

Name: PIO_OSR
Offset: 0x0018
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Output Status

Value	Description
0	The I/O line is a pure input.
1	The I/O line is enabled in output.

Downloaded from Arrow.com.

29.6.7 PIO Input Filter Enable Register

Name: PIO_IFER
Offset: 0x0020
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Input Filter Enable

Value	Description
0	No effect.
1	Enables the input glitch filter on the I/O line.

Downloaded from Arrow.com.

29.6.8 PIO Input Filter Disable Register

Name: PIO_IFDR
Offset: 0x0024
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Input Filter Disable

Value	Description
0	No effect.
1	Disables the input glitch filter on the I/O line.

Downloaded from Arrow.com.

29.6.9 PIO Input Filter Status Register

Name: PIO_IFSR
Offset: 0x0028
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Input Filter Status

Value	Description
0	The input glitch filter is disabled on the I/O line.
1	The input glitch filter is enabled on the I/O line.

Downloaded from Arrow.com.

29.6.10 PIO Set Output Data Register

Name: PIO_SODR
Offset: 0x0030
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Set Output Data

Value	Description
0	No effect.
1	Sets the data to be driven on the I/O line.

Downloaded from Arrow.com.

29.6.11 PIO Clear Output Data Register

Name: PIO_CODR
Offset: 0x0034
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Clear Output Data

Value	Description
0	No effect.
1	Clears the data to be driven on the I/O line.

Downloaded from Arrow.com.

29.6.12 PIO Output Data Status Register

Name: PIO_ODSR
Offset: 0x0038
Reset: –
Property: Read-only
or Read/Write

PIO_ODSR is Read-only or Read/Write depending on PIO_OWSR I/O lines.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Output Data Status

Value	Description
0	The data to be driven on the I/O line is 0.
1	The data to be driven on the I/O line is 1.

Downloaded from Arrow.com.

29.6.13 PIO Pin Data Status Register

Name: PIO_PDSR
Offset: 0x003C
Property: Read-only

Reset values depend on the level of the I/O lines.

Reading the I/O line levels requires the clock of the PIO Controller to be enabled, otherwise PIO_PDSR reads the levels present on the I/O line at the time the clock was disabled.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset								

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset								

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset								

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset								

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Output Data Status

Value	Description
0	The I/O line is at level 0.
1	The I/O line is at level 1.

Downloaded from Arrow.com.

29.6.14 PIO Interrupt Enable Register

Name: PIO_IER
Offset: 0x0040
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Input Change Interrupt Enable

Value	Description
0	No effect.
1	Enables the input change interrupt on the I/O line.

Downloaded from Arrow.com.

29.6.15 PIO Interrupt Disable Register

Name: PIO_IDR
Offset: 0x0044
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Input Change Interrupt Disable

Value	Description
0	No effect.
1	Disables the input change interrupt on the I/O line.

Downloaded from Arrow.com.

29.6.16 PIO Interrupt Mask Register

Name: PIO_IMR
Offset: 0x0048
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Input Change Interrupt Mask

Value	Description
0	Input change interrupt is disabled on the I/O line.
1	Input change interrupt is enabled on the I/O line.

Downloaded from Arrow.com.

29.6.17 PIO Interrupt Status Register

Name: PIO_ISR
Offset: 0x004C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Input Change Interrupt Status

Value	Description
0	No input change has been detected on the I/O line since PIO_ISR was last read or since reset.
1	At least one input change has been detected on the I/O line since PIO_ISR was last read or since reset.

Downloaded from Arrow.com.

29.6.18 PIO Multi-driver Enable Register

Name: PIO_MDER
Offset: 0x0050
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Multi-drive Enable

Value	Description
0	No effect.
1	Enables multi-drive on the I/O line.

Downloaded from Arrow.com.

29.6.19 PIO Multi-driver Disable Register

Name: PIO_MDDR
Offset: 0x0054
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Multi-drive Disable

Value	Description
0	No effect.
1	Disables multi-drive on the I/O line.

Downloaded from Arrow.com.

29.6.20 PIO Multi-driver Status Register

Name: PIO_MDSR
Offset: 0x0058
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Multi-drive Status

Value	Description
0	The multi-drive is disabled on the I/O line. The pin is driven at high- and low-level.
1	The multi-drive is enabled on the I/O line. The pin is driven at low-level only.

Downloaded from Arrow.com.

29.6.21 PIO Pull-Up Disable Register

Name: PIO_PUDR
Offset: 0x0060
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Pull-Up Disable

Value	Description
0	No effect.
1	Disables the pullup resistor on the I/O line.

Downloaded from Arrow.com.

29.6.22 PIO Pull-Up Enable Register

Name: PIO_PUER
Offset: 0x0064
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Pull-Up Enable

Value	Description
0	No effect.
1	Enables the pullup resistor on the I/O line.

Downloaded from Arrow.com.

29.6.23 PIO Pull-Up Status Register

Name: PIO_PUSR
Offset: 0x0068
Property: Read-only

Reset values depend on the product implementation.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset								

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset								

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset								

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset								

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Pull-Up Status

Value	Description
0	Pullup resistor is enabled on the I/O line.
1	Pullup resistor is disabled on the I/O line.

Downloaded from Arrow.com.

29.6.24 PIO Peripheral ABCD Select Register 1

Name: PIO_ABCDSR1
Offset: 0x0070
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Peripheral Select
If the same bit is set to '0' in PIO_ABCDSR2:
0: Assigns the I/O line to the Peripheral A function.
1: Assigns the I/O line to the Peripheral B function.
If the same bit is set to '1' in PIO_ABCDSR2:
0: Assigns the I/O line to the Peripheral C function.
1: Assigns the I/O line to the Peripheral D function.

Downloaded from Arrow.com.

29.6.25 PIO Peripheral ABCD Select Register 2

Name: PIO_ABCDSR2
Offset: 0x0074
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Peripheral Select
If the same bit is set to '0' in PIO_ABCDSR1:
0: Assigns the I/O line to the Peripheral A function.
1: Assigns the I/O line to the Peripheral C function.
If the same bit is set to '1' in PIO_ABCDSR1:
0: Assigns the I/O line to the Peripheral B function.
1: Assigns the I/O line to the Peripheral D function.

Downloaded from Arrow.com.

29.6.26 PIO Input Filter Slow Clock Disable Register

Name: PIO_IFSCDR
Offset: 0x0080
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Peripheral Clock Glitch Filtering Select

Value	Description
0	No effect.
1	The glitch filter is able to filter glitches with a duration < $t_{peripheral\ clock}/2$.

Downloaded from Arrow.com.

29.6.27 PIO Input Filter Slow Clock Enable Register

Name: PIO_IFSCER
Offset: 0x0084
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Slow Clock Debouncing Filtering Select

Value	Description
0	No effect.
1	The debouncing filter is able to filter pulses with a duration < $t_{div_slck}/2$.

Downloaded from Arrow.com.

29.6.28 PIO Input Filter Slow Clock Status Register

Name: PIO_IFSCSR
Offset: 0x0088
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Glitch or Debouncing Filter Selection Status

Value	Description
0	The glitch filter is able to filter glitches with a duration < $t_{peripheral\ clock}/2$.
1	The debouncing filter is able to filter pulses with a duration < $t_{div_slck}/2$.

Downloaded from Arrow.com.

29.6.29 PIO Slow Clock Divider Debouncing Register

Name: PIO_SCDR
Offset: 0x008C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			DIV[13:8]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 13:0 – DIV[13:0] Slow Clock Divider Selection for Debouncing
$t_{div_slck} = ((DIV + 1) \times 2) \times t_{slck}$

Downloaded from Arrow.com.

29.6.30 PIO Pad Pull-Down Disable Register

Name: PIO_PPDDR
Offset: 0x0090
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Pull-Down Disable

Value	Description
0	No effect.
1	Disables the pull-down resistor on the I/O line.

Downloaded from Arrow.com.

29.6.31 PIO Pad Pull-Down Enable Register

Name: PIO_PPDER
Offset: 0x0094
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Pull-Down Enable

Value	Description
0	No effect.
1	Enables the pull-down resistor on the I/O line.

Downloaded from Arrow.com.

29.6.32 PIO Pad Pull-Down Status Register

Name: PIO_PPDSR
Offset: 0x0098
Property: Read-only

Reset values depend on the product implementation.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset								

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset								

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset								

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset								

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Pull-Down Status

Value	Description
0	Pull-down resistor is enabled on the I/O line.
1	Pull-down resistor is disabled on the I/O line.

Downloaded from Arrow.com.

29.6.33 PIO Output Write Enable Register

Name: PIO_OWER
Offset: 0x00A0
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Output Write Enable

Value	Description
0	No effect.
1	Enables writing PIO_ODSR for the I/O line.

Downloaded from Arrow.com.

29.6.34 PIO Output Write Disable Register

Name: PIO_OWDR
Offset: 0x00A4
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Output Write Disable

Value	Description
0	No effect.
1	Disables writing PIO_ODSR for the I/O line.

Downloaded from Arrow.com.

29.6.35 PIO Output Write Status Register

Name: PIO_OWSR
Offset: 0x00A8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Output Write Status

Value	Description
0	Writing PIO_ODSR does not affect the I/O line.
1	Writing PIO_ODSR affects the I/O line.

Downloaded from Arrow.com.

29.6.36 PIO Additional Interrupt Modes Enable Register

Name: PIO_AIMER
Offset: 0x00B0
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Additional Interrupt Modes Enable

Value	Description
0	No effect.
1	The interrupt source is the event described in PIO_ELSR and PIO_FRLHSR.

Downloaded from Arrow.com.

29.6.37 PIO Additional Interrupt Modes Disable Register

Name: PIO_AIMDR
Offset: 0x00B4
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Additional Interrupt Modes Disable

Value	Description
0	No effect.
1	The Interrupt mode is set to the default Interrupt mode (Both-edge Detection).

Downloaded from Arrow.com.

29.6.38 PIO Additional Interrupt Modes Mask Register

Name: PIO_AIMMR
Offset: 0x00B8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO I/O Line Index
Selects the I/O event type triggering an interrupt.

Value	Description
0	The interrupt source is a both-edge detection event.
1	The interrupt source is described by the registers PIO_ELSR and PIO_FRLHSR.

Downloaded from Arrow.com.

29.6.39 PIO Edge Select Register

Name: PIO_ESR
Offset: 0x00C0
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Edge Interrupt Selection

Value	Description
0	No effect.
1	The interrupt source is an edge-detection event.

Downloaded from Arrow.com.

29.6.40 PIO Level Select Register

Name: PIO_LSR
Offset: 0x00C4
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Level Interrupt Selection

Value	Description
0	No effect.
1	The interrupt source is a level-detection event.

Downloaded from Arrow.com.

29.6.41 PIO Edge/Level Status Register

Name: PIO_ELSR
Offset: 0x00C8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Edge/Level Interrupt Source Selection

Value	Description
0	The interrupt source is an edge-detection event.
1	The interrupt source is a level-detection event.

Downloaded from Arrow.com.

29.6.42 PIO Falling Edge/Low-Level Select Register

Name: PIO_FELLSR
Offset: 0x00D0
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Falling Edge/Low-Level Interrupt Selection

Value	Description
0	No effect.
1	The interrupt source is set to a falling edge detection or low-level detection event, depending on PIO_ELSR.

Downloaded from Arrow.com.

29.6.43 PIO Rising Edge/High-Level Select Register

Name: PIO_REHLSR
Offset: 0x00D4
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Rising Edge/High-Level Interrupt Selection

Value	Description
0	No effect.
1	The interrupt source is set to a rising edge detection or high-level detection event, depending on PIO_ELSR.

Downloaded from Arrow.com.

29.6.44 PIO Fall/Rise - Low/High Status Register

Name: PIO_FRLHSR
Offset: 0x00D8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P
PIO Edge/Level Interrupt Source Selection

Value	Description
0	The interrupt source is a falling edge detection (if PIO_ELSR = 0) or low-level detection event (if PIO_ELSR = 1).
1	The interrupt source is a rising edge detection (if PIO_ELSR = 0) or high-level detection event (if PIO_ELSR = 1).

Downloaded from Arrow.com.

29.6.45 PIO Write Protection Mode Register

Name: PIO_WPMR
Offset: 0x00E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x50494F	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 0 – WPEN Write Protection Enable
See "Register Write Protection" for the list of registers that can be protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x50494F ("PIO" in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x50494F ("PIO" in ASCII).

Downloaded from Arrow.com.

29.6.46 PIO Write Protection Status Register

Name: PIO_WPSR
Offset: 0x00E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the PIO_WPSR.
1	A write protection violation has occurred since the last read of the PIO_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

29.6.47 PIO Schmitt Trigger Register

Name: PIO_SCHMITT
Offset: 0x0100
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	SCHMITT31	SCHMITT30	SCHMITT29	SCHMITT28	SCHMITT27	SCHMITT26	SCHMITT25	SCHMITT24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SCHMITT23	SCHMITT22	SCHMITT21	SCHMITT20	SCHMITT19	SCHMITT18	SCHMITT17	SCHMITT16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SCHMITT15	SCHMITT14	SCHMITT13	SCHMITT12	SCHMITT11	SCHMITT10	SCHMITT9	SCHMITT8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SCHMITT7	SCHMITT6	SCHMITT5	SCHMITT4	SCHMITT3	SCHMITT2	SCHMITT1	SCHMITT0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – SCHMITTx PIO Schmitt Trigger Control

Value	Description
0	Schmitt trigger is enabled.
1	Schmitt trigger is disabled.

Downloaded from Arrow.com.

29.6.48 PIO I/O Slewrate Control Register

Name: PIO_SLEWR
Offset: 0x0110
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	SR31	SR30	SR29	SR28	SR27	SR26	SR25	SR24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SR23	SR22	SR21	SR20	SR19	SR18	SR17	SR16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SR15	SR14	SR13	SR12	SR11	SR10	SR9	SR8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – SRx Slewrate Control for IO line x
Refer to section “Electrical characteristics” for recommended usage of this bit.

Value	Name	Description
0	DISABLED	No slewrate control
1	ENABLED	Slewrate controlled

Downloaded from Arrow.com.

29.6.49 PIO I/O Drive Register 1

Name: PIO_DRIVER1
Offset: 0x0118
Reset: 0x00000000
Property: Read/Write

Refer to section “Electrical characteristics” for recommended usage of the following bits.

Bit	31	30	29	28	27	26	25	24
	DR31	DR30	DR29	DR28	DR27	DR26	DR25	DR24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DR23	DR22	DR21	DR20	DR19	DR18	DR17	DR16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DR15	DR14	DR13	DR12	DR11	DR10	DR9	DR8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – DRx Drive of PIO Line

Value	Name	Description
0	LOW_DRIVE	Lowest drive
1	HIGH_DRIVE	Highest drive

Downloaded from Arrow.com.

30. External Bus Interface (EBI)

30.1 Description

The External Bus Interface (EBI) is designed to ensure the successful data transfer between several external devices and the embedded Memory Controller of the SAM9X60.

The Static Memory, MPDDR, SDRAM and ECC Controllers are all featured external Memory Controllers on the EBI. These external Memory Controllers are capable of handling several types of external memory and peripheral devices, such as SRAM, PROM, EPROM, EEPROM, Flash and (DDR2-/LPDDR-/SDR-/LPSDR-) SDRAM. The EBI operates with 1.8V or 3.3V Power Supplies (VDDIOM and VDDNF).

The EBI also supports the NAND Flash protocols via integrated circuitry that greatly reduces the requirements for external components. Furthermore, the EBI handles data transfers with up to six external devices, each assigned to six address spaces defined by the embedded Memory Controller. Data transfers are performed through a 16-bit or 32-bit data bus, an address bus of up to 26 bits, up to six chip select lines (NCS[5:0]) and several control pins that are generally multiplexed between the different external Memory Controllers.

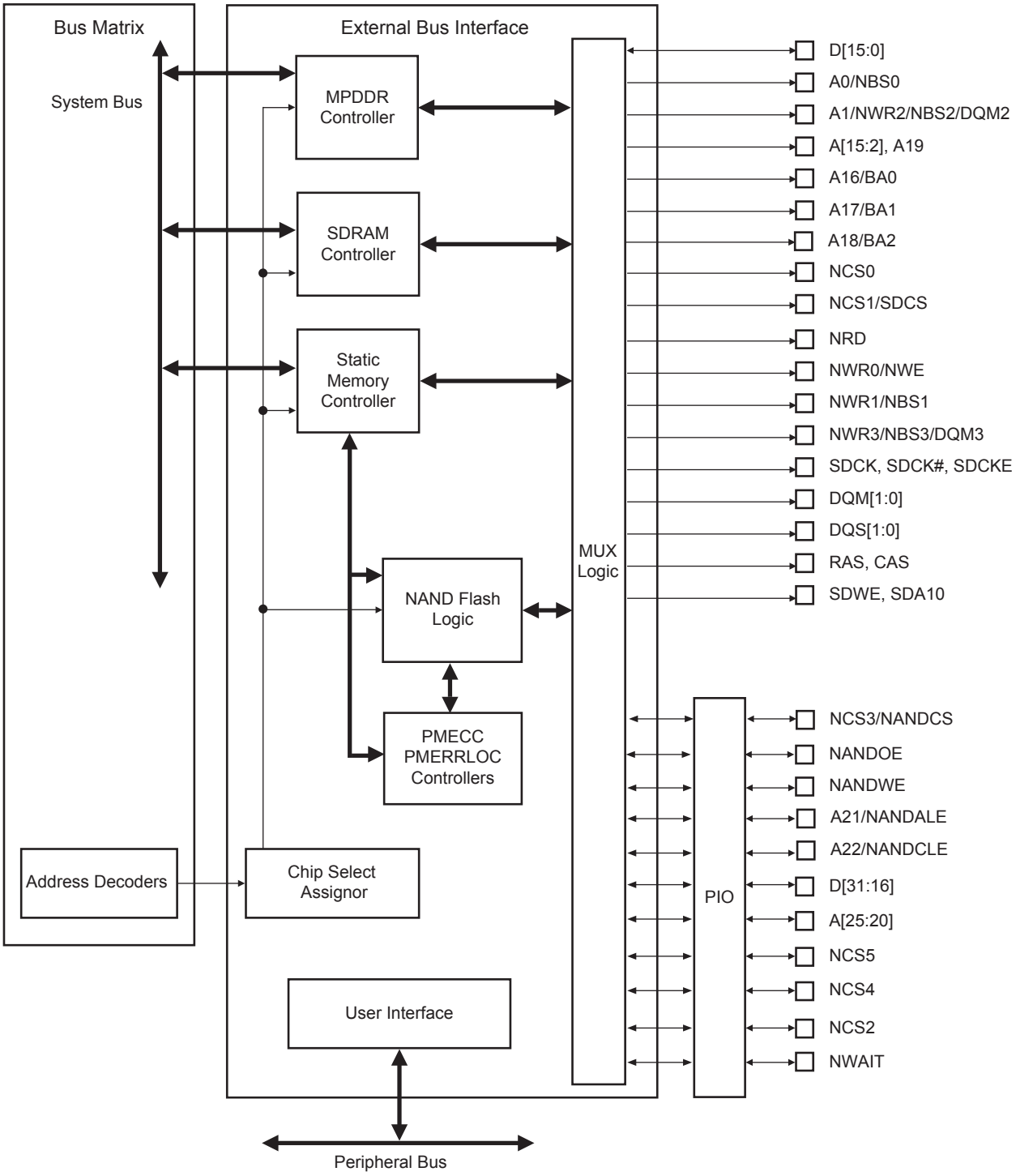
30.2 Embedded Characteristics

- Integrates Four External Memory Controllers:
  - Static memory controller
  - MPDDR controller
  - SDRAM controller
  - 8-bit NAND Flash ECC controller
- Up to 26-bit Address Bus (up to 64 Mbytes linear per chip select)
- Up to Six Chip Selects with Configurable Assignment:
  - Static memory controller on NCS0, NCS1, NCS2, NCS3, NCS4, NCS5
  - MPDDR / SDRAM controller (SDCS) or static memory controller on NCS1
  - NAND Flash support on NCS3

Downloaded from Arrow.com.

30.3 EBI Block Diagram

Figure 30-1. External Bus Interface Organization



Downloaded from Arrow.com.

30.4 I/O Lines Description

Table 30-1. EBI I/O Lines Description

Name	Function	Type	Active Level
EBI			
EBI_D0–EBI_D31	Data Bus	I/O	–
EBI_A0–EBI_A25	Address Bus	Output	–
EBI_NWAIT	External Wait Signal	Input	Low
SMC			
EBI_NCS0–EBI_NCS5	Chip Select Lines	Output	Low
EBI_NWR0–EBI_NWR3	Write Signals	Output	Low
EBI_NRD	Read Signal	Output	Low
EBI_NWE	Write Enable	Output	Low
EBI_NBS0–EBI_NBS3	Byte Mask Signals	Output	Low
EBI for NAND Flash Support			
EBI_NANDCS	NAND Flash Chip Select Line	Output	Low
EBI_NANDOE	NAND Flash Output Enable	Output	Low
EBI_NANDWE	NAND Flash Write Enable	Output	Low
MPDDR / SDRAM Controllers			
EBI_SDCK, EBI_SDCK#	MPDDR Differential Clock	Output	–
EBI_SDCK	SDRAM Clock	Output	–
EBI_SDCKE	MPDDR/SDRAM Clock Enable	Output	High
EBI_SDCS	MPDDR/SDRAM Chip Select Line	Output	Low
EBI_BA0–2	Bank Select	Output	–
EBI_SDWE	MPDDR/SDRAM Write Enable	Output	Low
EBI_RAS - EBI_CAS	Row and Column Signal	Output	Low
EBI_SDA10	SDRAM Address 10 Line	Output	–

The connection of some signals through the MUX logic is not direct and depends on the Memory Controller currently in use.

The following table details the connections between the two Memory Controllers and the EBI pins.

Table 30-2. EBI Pins and Memory Controllers I/O Lines Connections

EBIx Pins	SDRAM I/O Lines	SMC I/O Lines
EBI_NWR1/NBS1/CFIOR	NBS1	NWR1
EBI_A0/NBS0	Not Supported	SMC_A0
EBI_A1/NBS2/NWR2	Not Supported	SMC_A1
EBI_A[11:2]	SDRAM_A[9:0]	SMC_A[11:2]
EBI_SDA10	SDRAM_A10	Not Supported

Downloaded from Arrow.com.

.....continued		
EBIx Pins	**SDRAM I/O Lines**	**SMC I/O Lines**
EBI_A12	Not Supported	SMC_A12
EBI_A[15:13]	SDRAM_A[13:11]	SMC_A[15:13]
EBI_A[25:16]	Not Supported	SMC_A[25:16]
EBI_D[31:0]	D[31:0]	D[31:0]

30.5 Application Examples

30.5.1 Hardware Interface

The following table details the connections to be applied between the EBI pins and the external devices for each memory controller.

Table 30-3. EBI Pins and External Static Device Connections

Signals: EBI_	Pins of the Interfaced Device					
	8-bit Static Device	2 x 8-bit Static Devices	16-bit Static Device	4 x 8-bit Static Devices	2 x 16-bit Static Devices	32-bit Static Device
Controller	**SMC**					
D0–D7	D0–D7	D0–D7	D0–D7	D0–D7	D0–D7	D0–D7
D8–D15	–	D8–D15	D8–D15	D8–D15	D8–15	D8–15
D16–D23	–	–	–	D16–D23	D16–D23	D16–D23
D24–D31(1)	–	–	–	D24–D31	D24–D31	D24–D31
A0/NBS0	A0	–	NLB	–	NLB(2)	BE0
A1/NWR2/NBS2/ DQM2	A1	A0	A0	WE(3)	NLB(4)	BE2
A2–A22(1)	A[2:22]	A[1:21]	A[1:21]	A[0:20]	A[0:20]	A[0:20]
A23–A25(1)	A[23:25]	A[22:24]	A[22:24]	A[21:23]	A[21:23]	A[21:23]
NCS0	CS	CS	CS	CS	CS	CS
NCS1/DDRSDCS	CS	CS	CS	CS	CS	CS
NCS2(1)	CS	CS	CS	CS	CS	CS
NCS3/NANDCS	CS	CS	CS	CS	CS	CS
NCS4(1)	CS	CS	CS	CS	CS	CS
NCS5(1)	CS	CS	CS	CS	CS	CS
NRD	OE	OE	OE	OE	OE	OE
NWR0/NWE	WE	WE(5)	WE	WE(3)	WE	WE
NWR1/NBS1	–	WE(5)	NUB	WE(3)	NUB(2)	BE1
NWR3/NBS3/ DQM3	–	–	–	WE(3)	NUB(4)	BE3

Downloaded from Arrow.com.

Note:

1. D24–31 and A20, A23–A25, NCS2, NCS4, NCS5 are multiplexed on PD15–PD21.
2. NBS0 and NBS1 enable respectively lower and upper bytes of the lower 16-bit word.
3. NWRx enables corresponding byte x writes (x = 0, 1, 2 or 3).
4. NBS2 and NBS3 enable respectively lower and upper bytes of the upper 16-bit word.
5. NWR1 enables upper byte writes. NWR0 enables lower byte writes.

Table 30-4. EBI Pins and External Device Connections

Signals: EBI_	Power supply	Pins of the Interfaced Device		
		DDR2/LPDDR	SDR/LPSDR	NAND Flash
Controller		MPDDRC	SDRAMC	NFC
D0–D7	VDDIOM	D0–D7	D0–D7	NFD0–NFD7(1)
D8–D15	VDDIOM	D8–D15	D8–D15	–
D16–D23	VDDNF	–	D16–D23	NFD0–NFD7(1)
D24–D31	VDDNF	–	D24–D31	–
A0/NBS0	VDDIOM	–	–	–
A1/NWR2/NBS2/DQM2	VDDIOM	–	DQM2	–
DQM0–DQM1	VDDIOM	DQM0–DQM1	DQM0–DQM1	–
DQS0–DQS1	VDDIOM	DQS0–DQS1	–	–
A2–A10	VDDIOM	A[0:8]	A[0:8]	–
A11	VDDIOM	A9	A9	–
SDA10	VDDIOM	A10	A10	–
A12	VDDIOM	–	–	–
A13–A14	VDDIOM	A[11:12]	A[11:12]	–
A15	VDDIOM	A13	A13	–
A16/BA0	VDDIOM	BA0	BA0	–
A17/BA1	VDDIOM	BA1	BA1	–
A18/BA2	VDDIOM	BA2	–	–
A19	VDDIOM	–	–	–
A20	VDDNF	–	–	–
A21/NANDALE	VDDNF	–	–	ALE
A22/NANDCLE	VDDNF	–	–	CLE
A23–A24	VDDNF	–	–	–
A25	VDDNF	–	–	–
NCS0	VDDIOM	–	–	–
NCS1/DDRSDCS	VDDIOM	DDRCS	SDCS	–
NCS2	VDDNF	–	–	–
NCS3/NANDCS	VDDNF	–	–	CE
NCS4	VDDNF	–	–	–

Downloaded from Arrow.com.

.....continued				
Signals: EBI_	Power supply	Pins of the Interfaced Device		
		DDR2/LPDDR	SDR/LPSDR	NAND Flash
Controller		MPDDRC	SDRAMC	NFC
NCS5	VDDNF	–	–	–
NANDOE	VDDNF	–	–	OE
NANDWE	VDDNF	–	–	WE
NRD	VDDIOM	–	–	–
NWR0/NWE	VDDIOM	–	–	–
NWR1/NBS1	VDDIOM	–	–	–
NWR3/NBS3/DQM3	VDDNF	–	DQM3	–
SDCK	VDDIOM	CK	CK	–
SDCK#	VDDIOM	CK#	–	–
SDCKE	VDDIOM	CKE	CKE	–
RAS	VDDIOM	RAS	RAS	–
CAS	VDDIOM	CAS	CAS	–
SDWE	VDDIOM	WE	WE	–
Pxx	VDDNF	–	–	CE
NWAIT	VDDNF	–	–	RDY

Note: 1. The switch NFD0_ON_D16 is used to select NAND Flash path on D0–D7 or D16–D23 depending on memory power supplies. This switch is located in the SFR_CCFG_EBICSA register in the Special Function Register.

30.5.2 Product Dependencies

30.5.2.1 I/O Lines

The pins used for interfacing the External Bus Interface may be multiplexed with the PIO lines. The programmer must first program the PIO controller to assign the External Bus Interface pins to their peripheral function. If I/O lines of the External Bus Interface are not used by the application, they can be used for other purposes by the PIO Controller.

30.5.3 Functional Description

The EBI transfers data between the internal AHB Bus (handled by the Bus Matrix) and the external devices (memories, FPGA, etc.). It controls the waveforms and the parameters of the external address, data and control buses and is composed of the following elements:

- Static Memory Controller (SMC)
- MPDDR and SDRAM Controllers (MPDDRC and SDRAMC)
- Programmable Multibit ECC Controller (PMECC)
- A chip select assignment feature that assigns an AHB address space to the external devices
- A multiplex controller circuit that shares the pins between the different Memory Controllers
- Programmable NAND Flash support logic

30.5.3.1 Bus Multiplexing

The EBI offers a complete set of control signals that share the 32-bit data lines, the address lines of up to 26 bits and the control signals through a multiplex logic operating in function of the memory area requests.

Multiplexing is specifically organized in order to guarantee the maintenance of the address and output control lines at a stable state while no external access is being performed. Multiplexing is also designed to respect the data float

Downloaded from Arrow.com.

times defined in the Memory Controllers. Furthermore, refresh cycles of the DDR2, LP-DDR and SDRAM are executed independently by the MPDDRC or SDRAMC without delaying the other external Memory Controller accesses.

30.5.3.2 Pull-up and Pull-down Control

The SFR_CCFG_EBICSA register in the Special Function Register User Interface enable on-chip pull-up and pull-down resistors on data bus lines not multiplexed with the PIO Controller lines. The pull-down resistors are enabled after reset. The bits, EBIx_DBPUC and EBI_DBPDC, control the pull-up and pull-down resistors on the D0–D15 lines. Pull-up or pull-down resistors on the D16–D31 lines can be performed by programming the appropriate PIO controller.

30.5.3.3 Voltage Level Control

The EBI I/Os accept two voltage level ranges: 1.7V to 1.9V range or 3.0V to 3.6V range. The EBI I/O circuits must be programmed to accommodate the voltage level in each application:

- 1.7V to 1.9V range: the SFR_CCFG_EBICSA.EBI_DRIVE must be programmed to HIGH_DRIVE (1).
- 3.0V to 3.6V range: the SFR_CCFG_EBICSA.EBI_DRIVE must be programmed to LOW_DRIVE (0).

At reset output, the drive selection defaults to LOW_DRIVE.

This setting is applied to all the device I/Os with “DDRIO” type as defined in the Pin Description table (refer to 5. Package and Pinout). Other EBI I/Os with “GPIO” type must be programmed according to the recommendations on the GPIO DRIVE and SLEWRATE controls in the PIO user interface.

30.5.3.4 Power Supplies

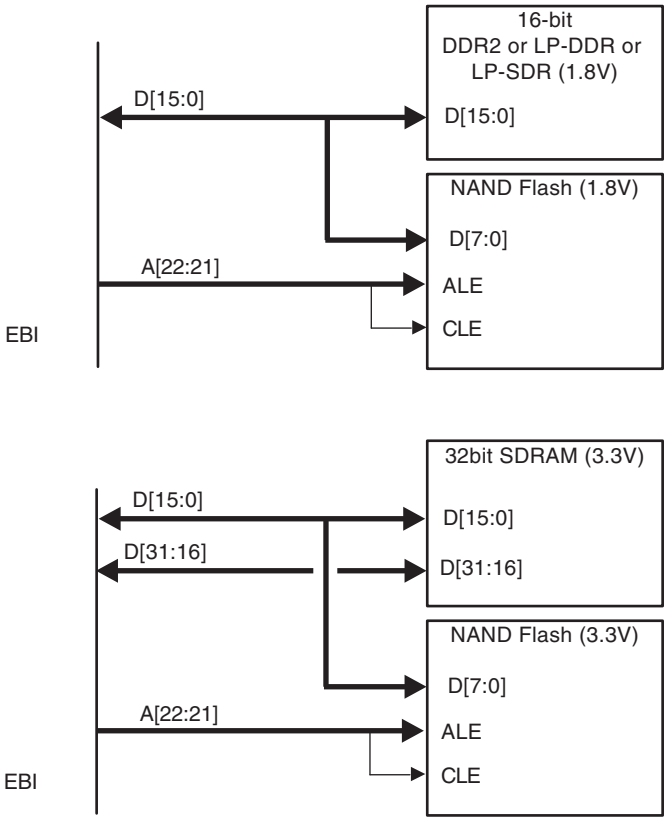
The product embeds a dual power supply for the EBI: VDDNF for NAND Flash signals, and VDDIOM for other signals. This makes it possible to use a 1.8V or 3.3V NAND Flash independently of the SDRAM power supply.

The switch NFD0_ON_D16 is used to select the NAND Flash path on D0–D7 or D16–D23 depending on memory power supplies. This switch is located in the SFR_CCFG_EBICSA register (refer to 23. Special Function Registers (SFR)).

The following figure illustrates an example of the NAND Flash and the external RAM (DDR2 or LP-DDR or 16-bit LP-SDR) in the same power supply range (NFD0_ON_D16 = default).

Downloaded from Arrow.com.

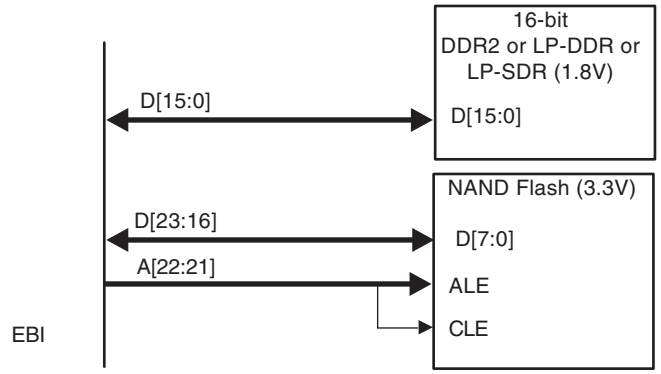
Figure 30-2. NAND Flash and External RAM in Same Power Supply Range (NFD0_ON_D16 = default)



The following figure illustrates an example of the NAND Flash and the external RAM (DDR2 or LP-DDR or 16-bit LP-SDR) NOT in the same power supply range (NFD0_ON_D16 = 1).

This can be used if the SMC connects to the NAND Flash only. Using this function with another device on the SMC will lead to an unpredictable behavior of that device. In that case, the default value must be selected.

Figure 30-3. NAND Flash and External RAM Not in Same Power Supply Range (NFD0_ON_D16 = 1)



At reset NFD0_ON_D16 = 0 and the NAND Flash bus is connected to D0–D7.

30.5.3.5 Static Memory Controller

For information on the Static Memory Controller, refer to the section “Static Memory Controller (SMC)”.

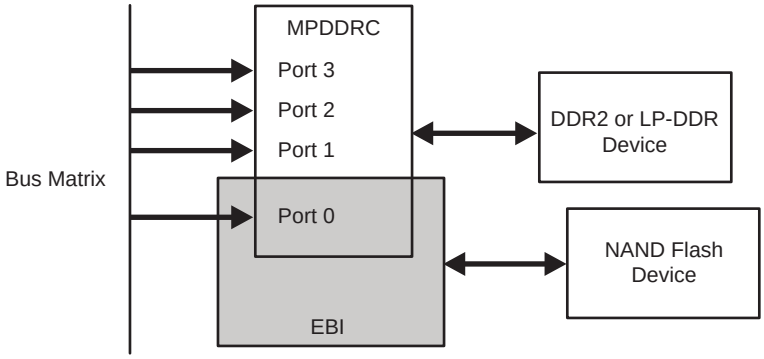
30.5.3.6 Multi-Port DDR and SDRAM Controllers

The product embeds a multi-port DDR Controller. This allows to use three additional ports on the MPDDRC to lessen the EBI load from a part of SDRAM accesses. This increases the bandwidth when DDR2 and NAND Flash devices are used. This feature is used with DDR2, LPDDR1 and SDR-SDRAM devices in Address/data or Address/data/command multiplexed mode.

Downloaded from Arrow.com.

It is controlled by the DDR_MP_EN bit in EBI Chip Select Assignment Register.

Figure 30-4. Multi-Port Enabled MPDDRC (DDR_MP_EN = 1)

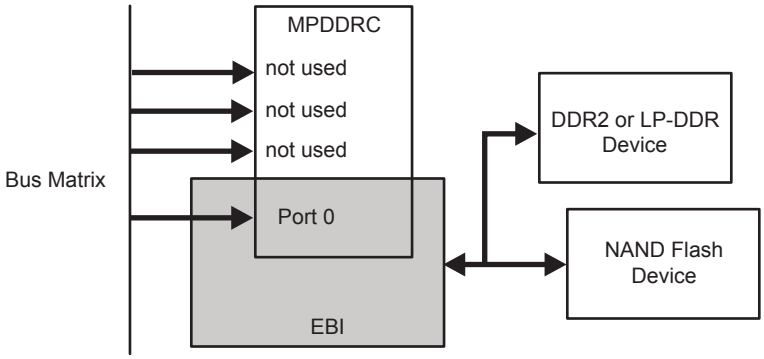


When:

- a NAND Flash memory is connected to D16-D23 and
- a DDR2-SDRAM or LPDDR-SDRAM is connected to D0-D15,

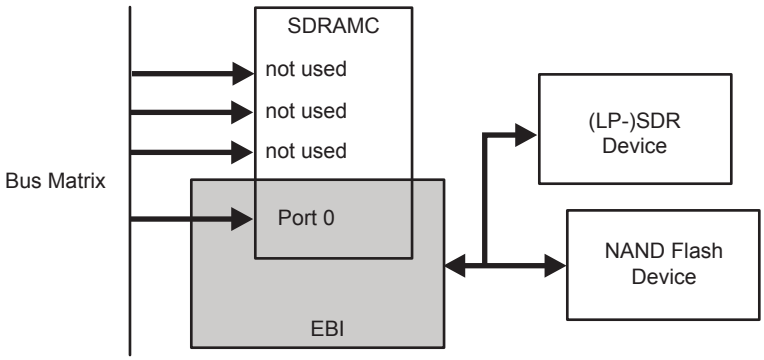
the bits SFR_CCFG_EBICSA.DDR_MP_EN and SFR_CCFG_EBICSA.NFD0_ON_D16 must both be set before performing the SDRAM initialization.

Figure 30-5. Multi-Port Disabled MPDDRC (DDR_MP_EN = 0)



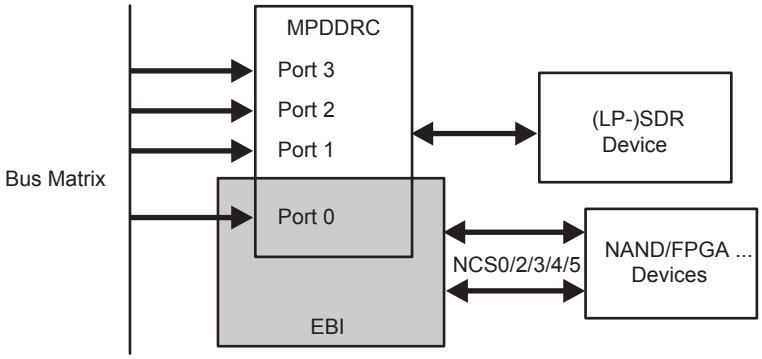
Set DQIEN_F to 1: force EBI D0-D15 data pads in Input mode. Mandatory when EBI D0-D15 is shared between DDR-DRAM (LPDDR/DDR2) and static memory (via MPDDRC and SMC).

Figure 30-6. Multi-Port Disabled SDRAMC (DDR_MP_EN = 0)



Downloaded from Arrow.com.

Figure 30-7. Multiplexed Mode Multi-Port Enabled SDRAMC (DDR_MP_EN = 1) (With Addr/Data/Cmd Multiplexed Mode)



The product embeds a multi-port SDR Controller in Address/Data or Address/Data/Command multiplexed mode. This allows to use three additional ports on the SDRAMC to remove SDRAM accesses from the EBI load. This increases the bandwidth when SDR-SDRAM and a full SMC are used. This configuration allows to support up to five chip selects on SMC.

30.5.3.7 Programmable Multibit ECC Controller

For information on the PMECC Controller, refer to 34. Programmable Multibit Error Correction Code Controller (PMECC) and 35. Programmable Multibit ECC Error Location Controller (PMERRLOC). Also refer to 11.4.7.2 NAND Flash Boot: PMECC Error Detection and Correction.

30.5.3.8 NAND Flash Support

External Bus Interfaces integrate circuitry that interfaces to NAND Flash devices.

External Bus Interface

The NAND Flash logic is driven by the Static Memory Controller on the NCS3 address space. Programming the EBI_CS3A field in the SFR_CCFG_EBICSA Register in the SFR User Interface to the appropriate value enables the NAND Flash logic. For details on this register, refer to 23. Special Function Registers (SFR). Access to an external NAND Flash device is then made by accessing the address space reserved to NCS3 (i.e., between 0x4000 0000 and 0x4FFF FFFF).

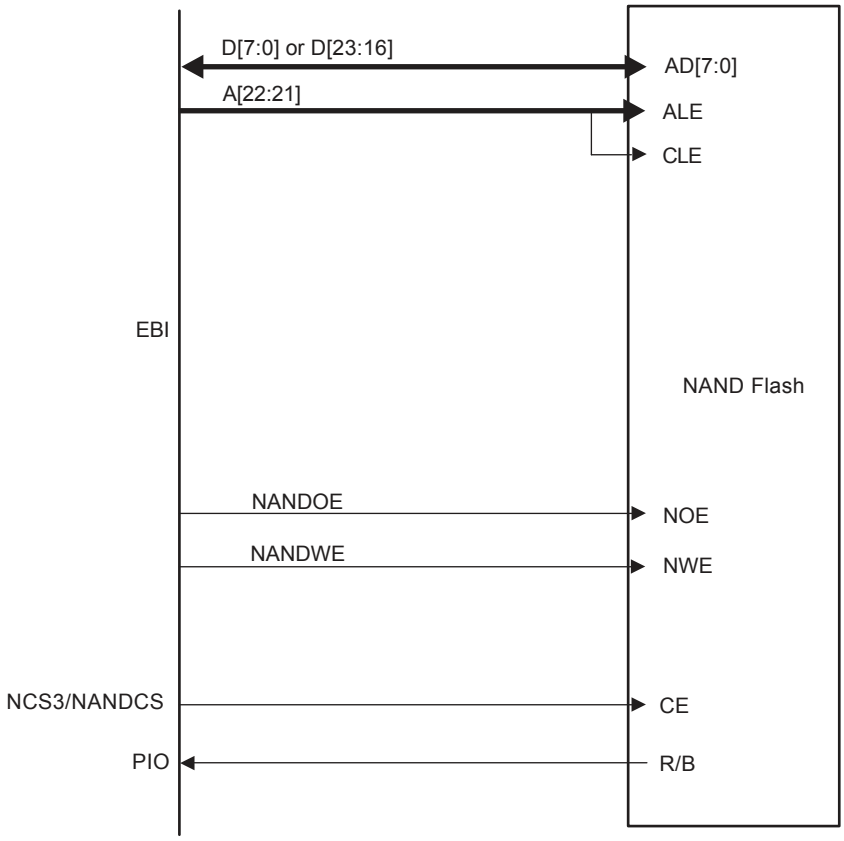
The NAND Flash Logic drives the read and write command signals of the SMC on the NANDOE and NANDWE signals when the NCS3 signal is active. NANDOE and NANDWE are invalidated as soon as the transfer address fails to lie in the NCS3 address space. See the figure below for more information. For details on these waveforms, refer to 33. Static Memory Controller (SMC).

NAND Flash Signals

The address latch enable and command latch enable signals on the NAND Flash device are driven by address bits A22 and A21 of the EBI address bus. The command, address or data words on the data bus of the NAND Flash device are distinguished by using their addresses within the NCSx address space. The chip enable (CE) signal of the device and the ready/busy (R/B) signals are connected to PIO lines. The CE signal then remains asserted even when NCSx is not selected, preventing the device from returning to Standby mode.

Downloaded from Arrow.com.

Figure 30-8. NAND Flash Application Example



Note: The CE signal of the NAND Flash must be connected to PIOD4 (NCS3/NANDCS) if the user's system boots out of NAND Flash.

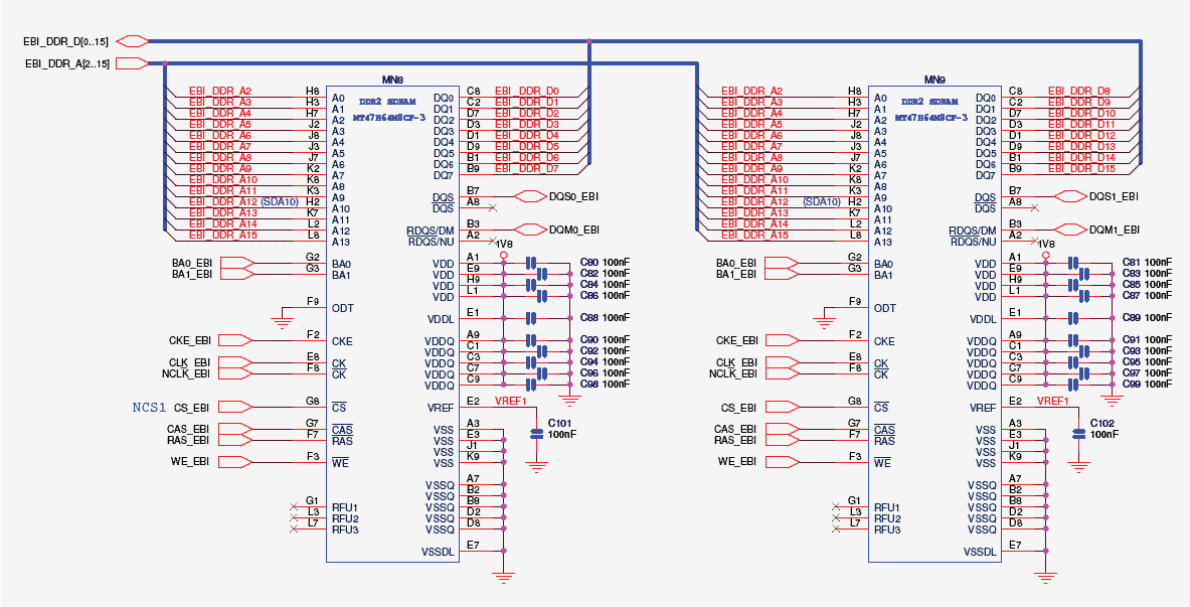
30.5.4 Implementation Examples

The following hardware configurations are given for illustration only. The user should refer to the memory manufacturer web site to check current device availability.

Downloaded from Arrow.com.

30.5.4.1 2x8-bit DDR2 on EBI

30.5.4.1.1 Hardware Configuration



30.5.4.1.2 Software Configuration

- Assign EBI_CS1 to the MPDDRC controller by setting the EBI_CS1A bit in the SFR_CCFG_EBICSA register.
- Initialize the MPDDR Controller depending on the DDR2 device and system bus frequency.

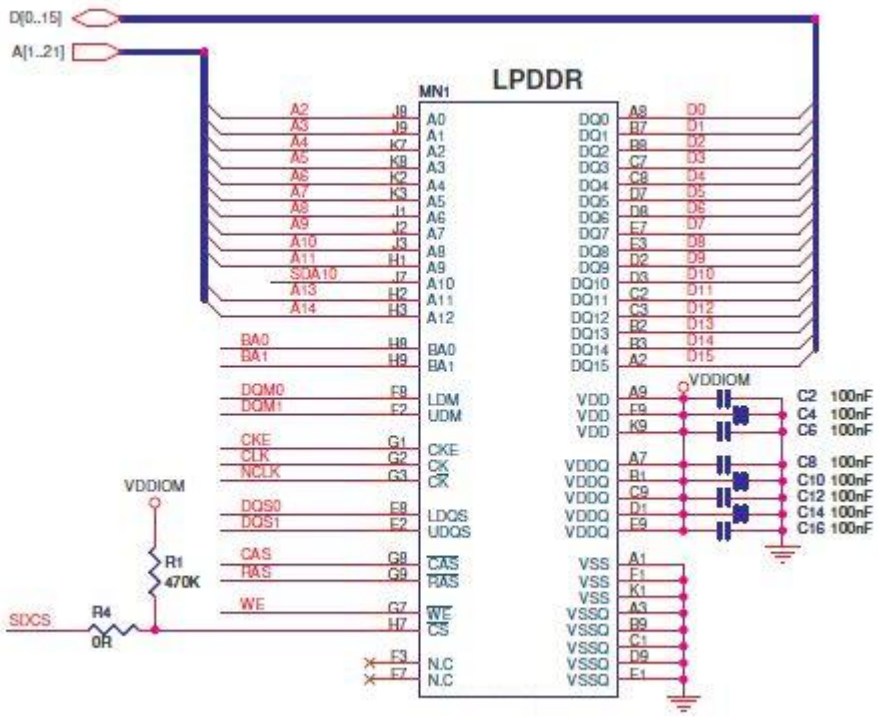
The DDR2 initialization sequence is described in the subsection “DDR2 Device Initialization” of the MPDDRC section.

In this case, VDDNF can be different from VDDIOM. The NAND Flash device can be 3.3V or 1.8V and wired on the D16–D23 data bus. NFD0_ON_D16 is to be set to 1.

Downloaded from Arrow.com.

30.5.4.2 16-bit LPDDR on EBI

30.5.4.2.1 Hardware Configuration



30.5.4.2.2 Software Configuration

The following configuration must be performed:

- Assign EBI_CS1 to the MPDDR controller by setting the bit EBI_CS1A bit in the SFR_CCFG_EBICSA register.
- Initialize the MPDDR Controller depending on the LP-DDR device and system bus frequency.

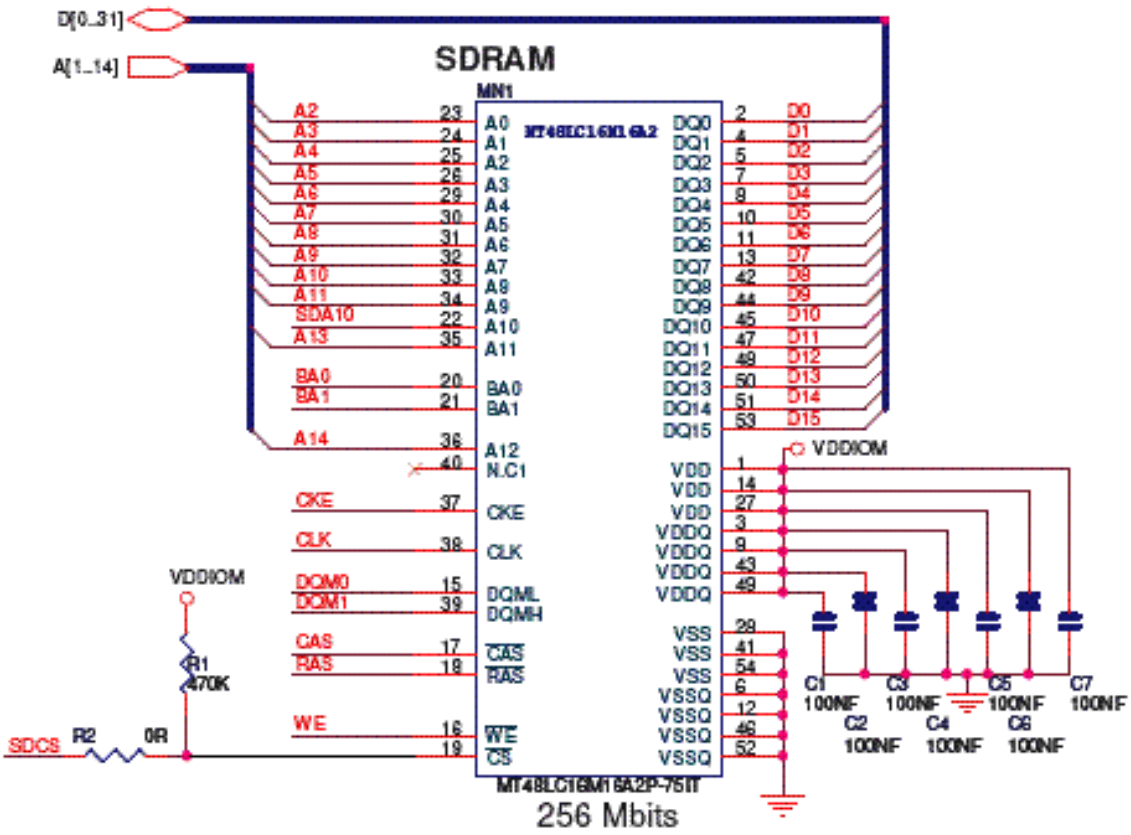
The LP-DDR initialization sequence is described in the section “Low-power DDR1-SDRAM Initialization” in the MPDDRC section.

In this case, VDDNF can be different from VDDIOM. The NAND Flash device can be 3.3V or 1.8V and wired on the D16–D23 data bus. NFD0_ON_D16 is to be set to 1.

Downloaded from Arrow.com.

30.5.4.3 16-bit SDRAM on EBI

30.5.4.3.1 Hardware Configuration



30.5.4.3.2 Software Configuration

The following configuration must be performed:

- Assign the EBI CS1 to the SDRAM controller by setting the EBI_CS1A bit in the SFR_CCFG_EBICSA register.
- Initialize the SDRAM Controller depending on the SDRAM device and system bus frequency.

Program the Data Bus Width to 16 bits.

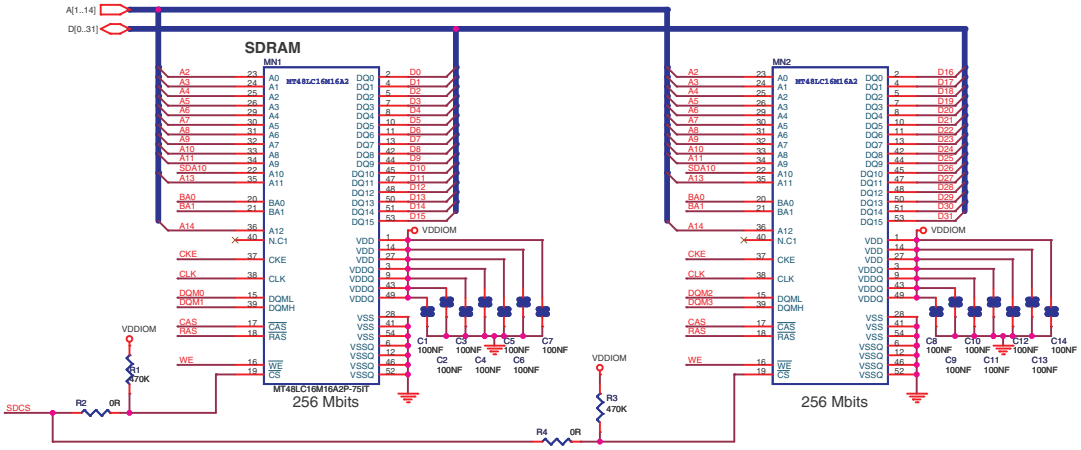
The SDRAM initialization sequence is described in the section “SDRAM Device Initialization” in “SDRAM Controller (SDRAMC)”.

In this case, VDDNF can be different from VDDIOM. The NAND Flash device can be 3.3V or 1.8V and wired on the D16–D23 data bus. NFD0_ON_D16 is to be set to 1.

Downloaded from Arrow.com.

30.5.4.4 2x16-bit SDRAM on EBI

30.5.4.4.1 Hardware Configuration



30.5.4.4.2 Software Configuration

The following configuration must be performed:

- Assign the EBI CS1 to the SDRAM controller by setting the EBI_CS1A bit in the SFR_CCFG_EBICSA register.
- Initialize the SDRAM Controller depending on the SDRAM device and system bus frequency.

Program the Data Bus Width to 32 bits. The data lines D[16..31] are multiplexed with PIO lines and thus the dedicated PIOs must be programmed in Peripheral mode in the PIO controller.

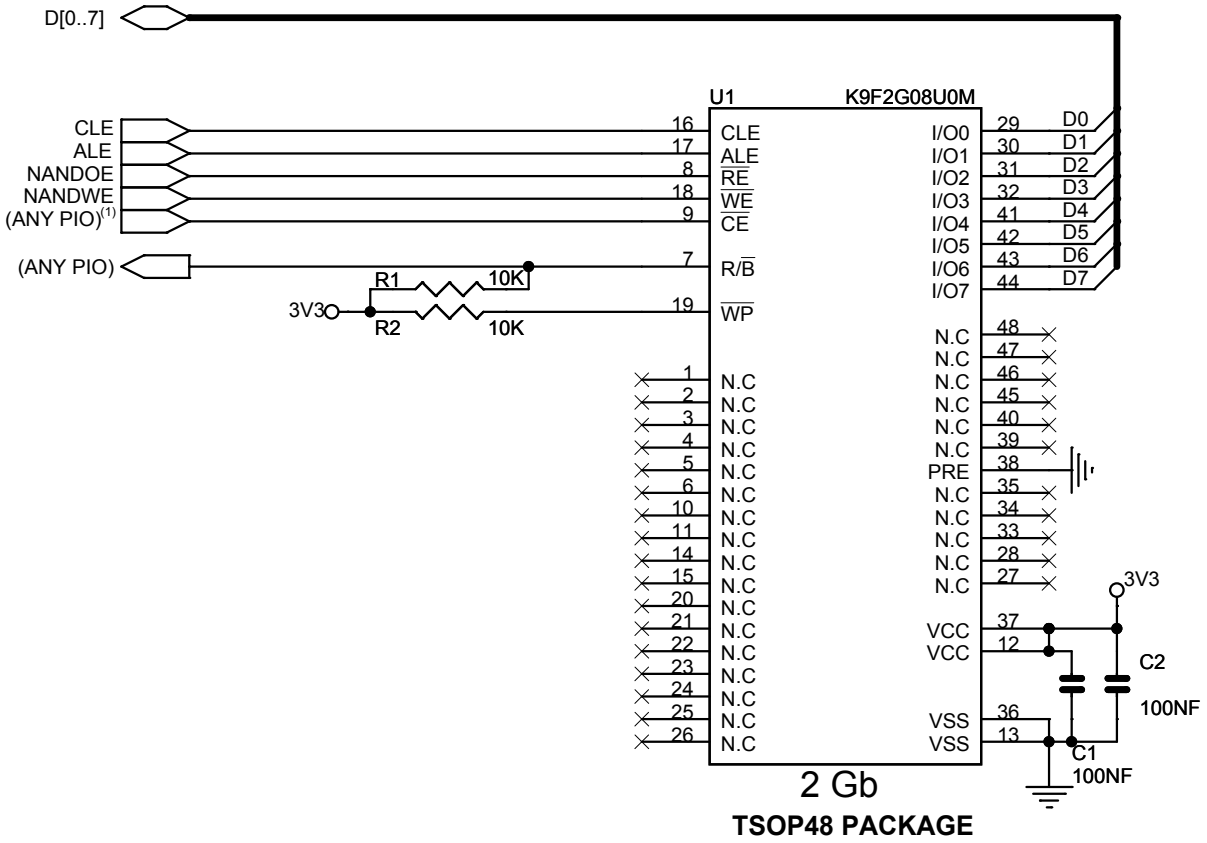
The SDRAM initialization sequence is described in the section “SDRAM Device Initialization” in “SDRAM Controller (SDRAMC)”.

In this case, VDDNF must be equal to VDDIOM. The NAND Flash device must be 3.3V and wired on the D0–D7 data bus. NFD0_ON_D16 is to be set to 0.

Downloaded from Arrow.com.

30.5.4.5 8-bit NAND Flash with NFD0_ON_D16 = 0

30.5.4.5.1 Hardware Configuration



(1) The CE must be connected to NCS3 PIOD4 if the NAND Flash is used by the ROM code.

Note: The CE signal of the NAND Flash must be connected to PIOD4 (NCS3/NANDCS) if the user's system boots out of NAND Flash.

30.5.4.5.2 Software Configuration

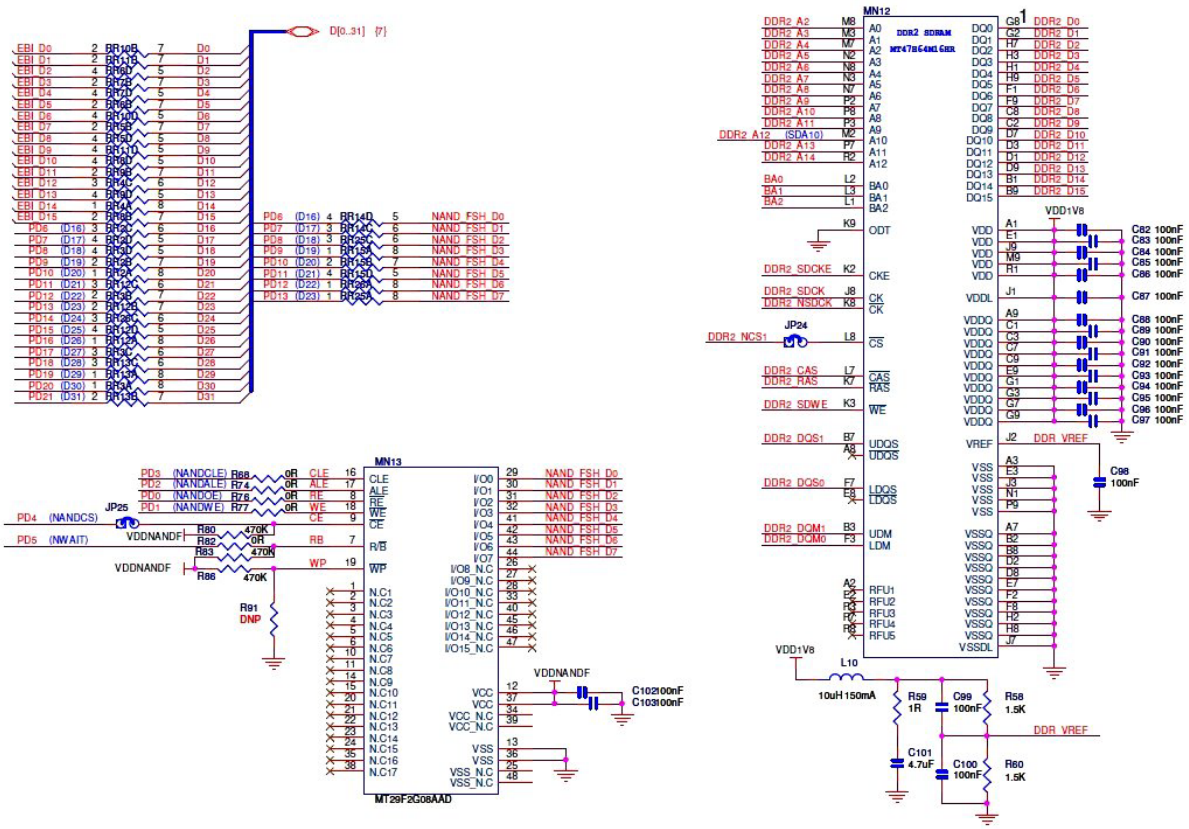
The following configuration has to be performed:

- Set NFD0_ON_D16 = 0 in the SFR_CCFG_EBICSA register
- Assign the EBI CS3 to the NAND Flash by setting the EBI_CS3A bit in the SFR_CCFG_EBICSA register
- Reserve A21/A22 for ALE/CLE functions. Address and Command Latches are controlled respectively by setting to 1 the address bits A21 and A22 during accesses.
- Configure a PIO line as an input to manage the Ready/Busy signal.
- Configure Static Memory Controller CS3 Setup, Pulse, Cycle and Mode according to NAND Flash timings, data bus width and system bus frequency.

Downloaded from Arrow.com.

30.5.4.6 8-bit NAND Flash with NFD0_ON_D16 = 1

30.5.4.6.1 Hardware Configuration



30.5.4.6.2 Software Configuration

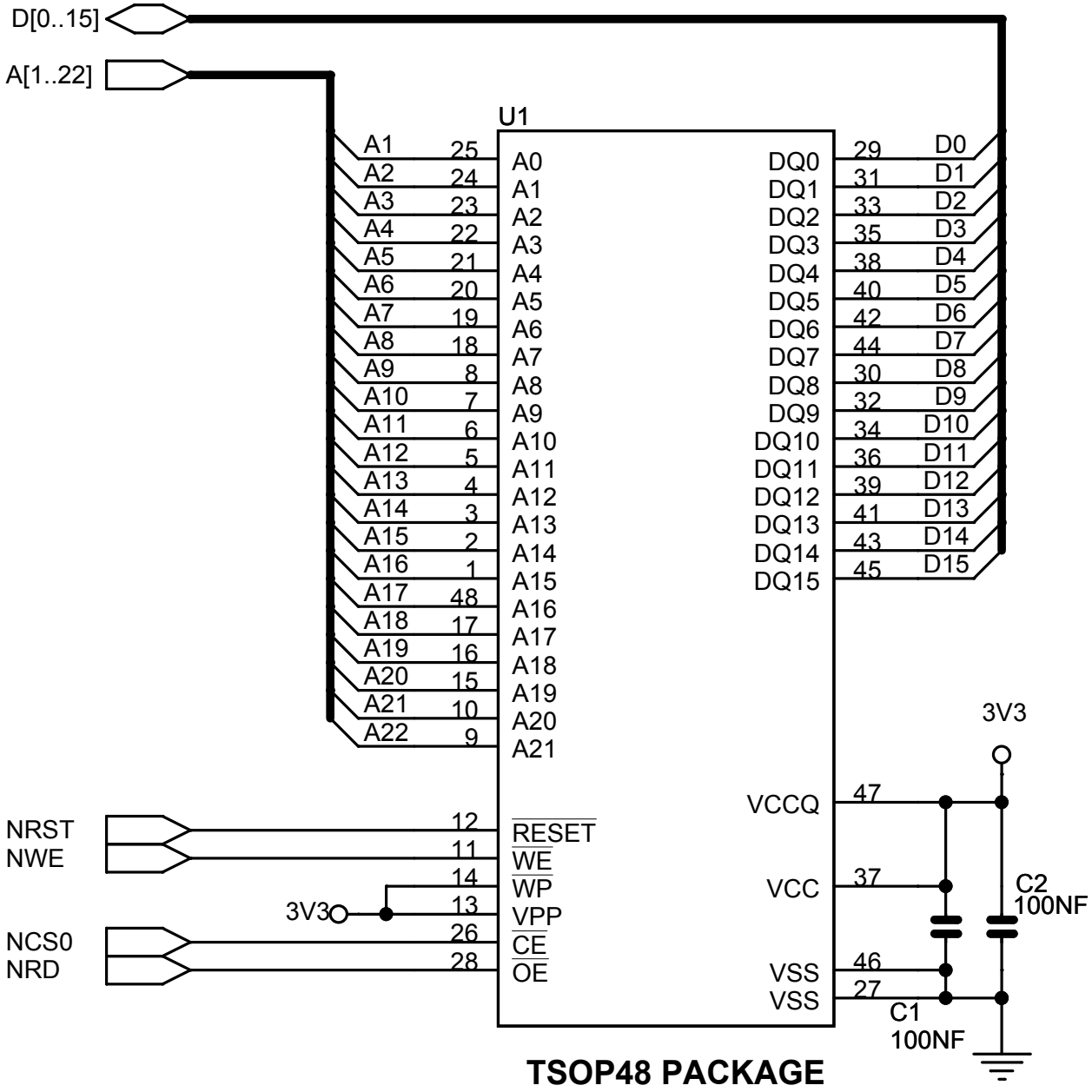
The following configuration must be performed:

- Set NFD0_ON_D16 = 1 in the SFR_CCFG_EBICSA register.
- Assign the EBI CS3 to the NAND Flash by setting the EBI_CS3A bit in the SFR_CCFG_EBICSA register.
- Reserve A21 / A22 for ALE / CLE functions. Address and Command Latches are controlled respectively by setting to 1 the address bits A21 and A22 during accesses.
- Configure a PIO line as an input to manage the Ready/Busy signal.
- Configure Static Memory Controller CS3 Setup, Pulse, Cycle and Mode according to NAND Flash timings, data bus width and system bus frequency.

Downloaded from Arrow.com.

30.5.4.7 NOR Flash on NCS0

30.5.4.7.1 Hardware Configuration



30.5.4.7.2 Software Configuration

The default configuration for the Static Memory Controller, Byte Select mode, 16-bit data bus, Read/Write controlled by Chip Select, allows boot on 16-bit non-volatile memory at slow clock.

For another configuration, configure the Static Memory Controller CS0 Setup, Pulse, Cycle and Mode depending on Flash timings and system bus frequency.

Downloaded from Arrow.com.

31. AHB Multiport DDR-SDRAM Controller (MPDDRC)

31.1 Description

The Multiport DDR-SDRAM Controller (MPDDRC) is a multiport memory controller. It comprises four slave AHB interfaces. All simultaneous accesses (four independent AHB ports) are interleaved to maximize memory bandwidth and minimize transaction latency due to DDR-SDRAM protocol.

The MPDDRC extends the memory capabilities of a chip by providing the interface to the external 16-bit DDR-SDRAM device. The page size supports ranges from 2048 to 16384 rows and from 256 to 4096 columns. It supports word (32-bit), half-word (16-bit), and byte (8-bit) accesses.

The MPDDRC supports a read or write burst length of eight locations. This enables the command and address bus to anticipate the next command, thus reducing latency imposed by the DDR-SDRAM protocol and improving the DDR-SDRAM bandwidth. Moreover, MPDDRC keeps track of the active row in each bank, thus maximizing DDR-SDRAM performance, e.g., the application may be placed in one bank and data in other banks. To optimize performance, avoid accessing different rows in the same bank. The MPDDRC supports a CAS latency of 2, 3 and optimizes the read access depending on the frequency.

Self-refresh, Powerdown and Deep Powerdown modes minimize the consumption of the DDR-SDRAM device.

OCD (Off-chip Driver) and ODT (On-die Termination) modes are not supported.

31.2 Embedded Characteristics

- Numerous Memory Devices Supported
  - Low-power DDR1-SDRAM (LPDDR1)
  - Low-cost LPDDR1 with 2 internal banks
  - DDR2-SDRAM
- Arbitration Policies: Round-Robin, On Request, Bandwidth, Quality of Service
- Four Advanced High-Performance Bus (AHB) Interfaces, Management of all Accesses Maximizes Memory Bandwidth and Minimizes Transaction Latency
- Bus Transfer: word, half word, byte Access
- Numerous Configurations Supported
  - 2K, 4K, 8K, 16K row address memory parts
  - DDR-SDRAM with two or four internal banks (low-power DDR1-SDRAM)
  - DDR-SDRAM with four or eight internal banks (DDR2-SDRAM)
  - DDR-SDRAM with 16-bit data path for system-oriented word access
  - One chip select for SDRAM device (256-Mbyte address space)
- Programming Facilities
  - Multibank ping-pong access (up to four or eight banks opened at the same time = reduced average latency of transactions)
  - Timing parameters specified by software
  - Automatic refresh operation, refresh rate is programmable
  - Automatic update of DS, TCR and PASR parameters (low-power DDR-SDRAM devices)
- Energy-Saving Capabilities
  - Self-refresh, Powerdown, Active Powerdown and Deep Powerdown modes supported
- DDR-SDRAM Powerup Initialization by Software
- CAS Latency of 2, 3 Supported
- Reset Function Supported (DDR2-SDRAM)
- Clock Frequency Change in Self-Refresh Mode Supported (Low-power DDR-SDRAM)
- Autoprecharge Command Not Used

Downloaded from Arrow.com.

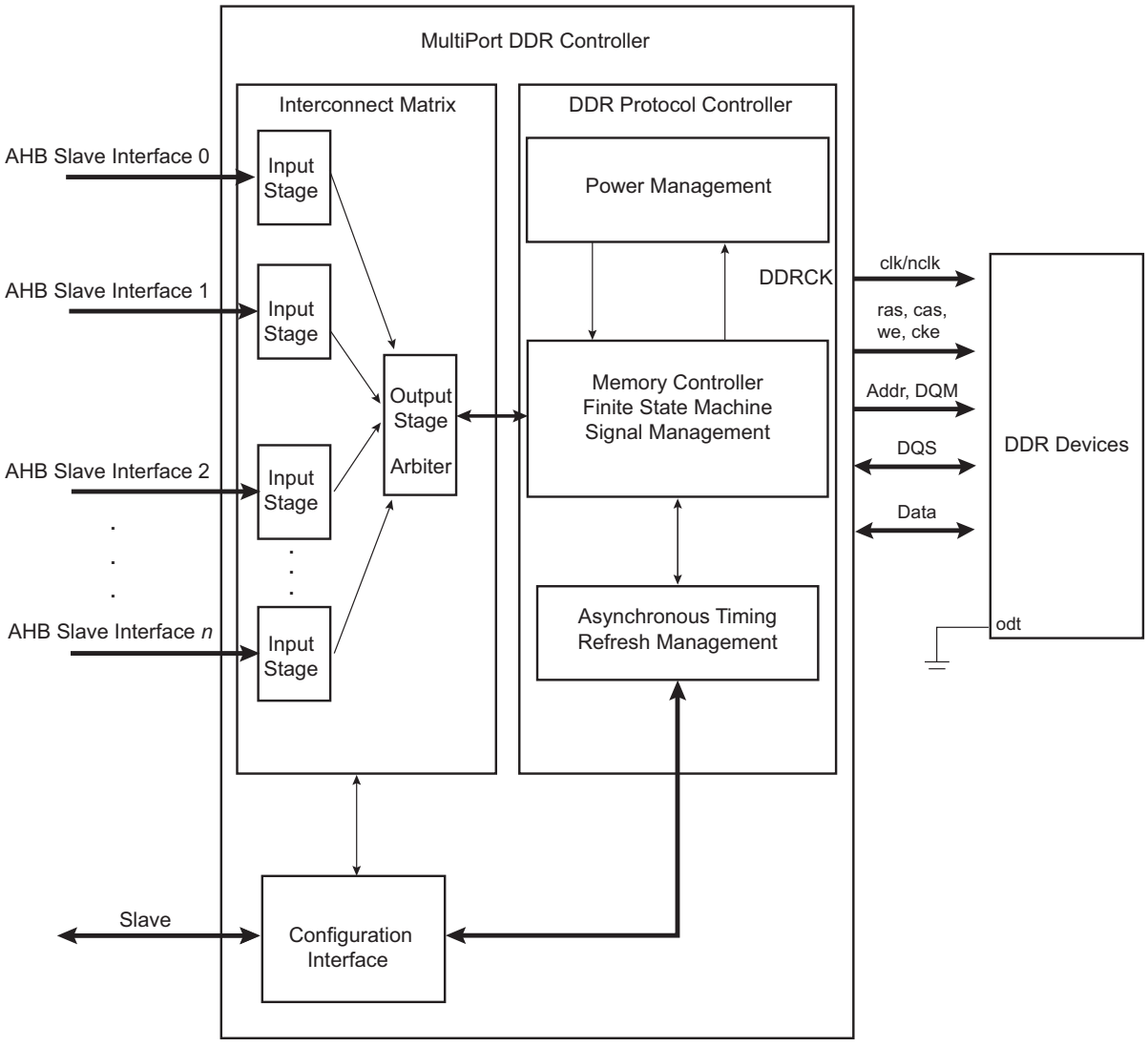
- OCD (Off-chip Driver) Mode, ODT (On-die Termination), Are Not Supported
- Abnormal Software Access and Sequencer Integrity Error Reports
- Dynamic Scrambling with User Key (No Impact on Bandwidth)
- Bus Monitor

31.3 Block Diagram

The MPDDRC is partitioned in two blocks (see figure below):

- An Interconnect Matrix block that manages concurrent accesses on the AHB bus between four AHB masters and integrates an arbiter
- A DDR Controller that translates AHB requests (read/write) in the DDR-SDRAM protocol

Figure 31-1. Block Diagram



31.4 Product Dependencies, Initialization Sequence

31.4.1 Low-power DDR1-SDRAM Initialization

The initialization sequence is generated by software.

Downloaded from Arrow.com.

The low-power DDR1-SDRAM devices are initialized by the following sequence:

1. Program the memory device type in the Memory Device register (MPDDRC_MD).
2. To comply with the LPDDR1 standard, DQS must be used in Single-ended mode. NDQS must be disabled in the MPDDRC Configuration register (MPDDRC_CR).
3. Program the shift sampling value in the Read Data Path register (MPDDRC_RD_DATA_PATH).
4. Program the features of the low-power DDR1-SDRAM device in the MPDDRC Configuration register (MPDDRC_CR) (number of columns, rows, banks, CAS latency and output drive strength) and in the Timing Parameter 0 register/Timing Parameter 1 register (MPDDRC_TPR0/1) (asynchronous timing (TRC, TRAS, etc.)).
5. Program Temperature Compensated Self-refresh (TCR), Partial Array Self-refresh (PASR) and Drive Strength (DS) parameters in the Low-power register (MPDDRC_LPR).
6. A NOP command is issued to the low-power DDR1-SDRAM. Program the NOP command in the Mode register (MPDDRC_MR). The application must configure the MODE field to 1 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any low-power DDR1-SDRAM address to acknowledge this command. The clocks which drive the low-power DDR1-SDRAM device are now enabled.
7. A pause of at least 200 µs must be observed before a signal toggle.
8. A NOP command is issued to the low-power DDR1-SDRAM. Program the NOP command in the MPDDRC_MR. The application must configure the MODE field to 1 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any low-power DDR1-SDRAM address to acknowledge this command. A calibration request is now made to the I/O pad.
9. An All Banks Precharge command is issued to the low-power DDR1-SDRAM. Program All Banks Precharge command in the MPDDRC_MR. The application must configure the MODE field to 2 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any low-power DDR1-SDRAM address to acknowledge this command.
10. Two autorefresh (CBR) cycles are provided. Program the Autorefresh command (CBR) in the MPDDRC_MR. The application must configure the MODE field to 4 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any low-power DDR1-SDRAM location twice to acknowledge these commands.
11. An Extended Mode Register Set (EMRS) cycle is issued to program the low-power DDR1-SDRAM parameters (TCSR, PASR, DS). The application must configure the MODE field to 5 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the SDRAM to acknowledge this command. The write address must be chosen so that signal BA[1] is set to 1 and BA[0] is set to 0. For example: with a 16-bit, 128-Mbit, low-power DDR1-SDRAM (12 rows, 9 columns, 4 banks), the SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x00800000 In the case of low-cost and low-density low-power DDR1-SDRAM (2 internal banks), the write address must be chosen so that signal BA[0] is set to 1. BA[1] is not used.
    Note: This address is given as an example only. The real address depends on implementation in the product.
12. A Mode Register Set (MRS) cycle is issued to program parameters of the low-power DDR1-SDRAM devices, in particular CAS latency. The application must configure the MODE field to 3 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the low-power DDR1-SDRAM to acknowledge this command. The write address must be chosen so that signals BA[1:0] are set to 0. For example, the SDRAM write access should be done at the address: BASE_ADDRESS_DDR.
13. The application must enter Normal mode, write a zero to the MODE field in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access at any location in the low-power DDR1-SDRAM to acknowledge this command.
14. Write the refresh rate into the COUNT field in the Refresh Timer register (MPDDRC_RTR). To compute the value, see MPDDRC Refresh Timer Register.

After initialization, the low-power DDR1-SDRAM device is fully functional.

31.4.2 DDR2-SDRAM Initialization

The initialization sequence is generated by software. The DDR2-SDRAM devices are initialized by the following sequence:

Downloaded from Arrow.com.

1. Program the memory device type in the Memory Device register (MPDDRC_MD).
2. Program the shift sampling value in the Read Data Path register (MPDDRC_RD_DATA_PATH).
3. Program features of the DDR2-SDRAM device in the Configuration register (MPDDRC_CR) (number of columns, rows, banks, CAS latency and output driver impedance control) and in the Timing Parameter 0 register/Timing Parameter 1 register (MPDDRC_TPR0/1) (asynchronous timing: TRC, TRAS, etc.).
4. A NOP command is issued to the DDR2-SDRAM. Program the NOP command in the Mode register (MPDDRC_MR). The application must configure the MODE field to 1 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any DDR2-SDRAM address to acknowledge this command. The clocks which drive the DDR2-SDRAM device are now enabled.
5. A pause of at least 200 µs must be observed before a signal toggle.
6. A NOP command is issued to the DDR2-SDRAM. Program the NOP command in the MPDDRC_MR. The application must configure the MODE field to 1 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any DDR2-SDRAM address to acknowledge this command. CKE is now driven high.
7. An All Banks Precharge command is issued to the DDR2-SDRAM. Program All Banks Precharge command in the MPDDRC_MR. The application must configure the MODE field to 2 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any DDR2-SDRAM address to acknowledge this command.
8. An Extended Mode Register Set (EMRS2) cycle is issued to choose between commercial or high temperature operations. The application must configure the MODE field to 5 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signal BA[1] is set to 1 and signal BA[0] is set to 0. For example: with a 16-bit, 128-Mbit, DDR2-SDRAM (12 rows, 9 columns, 4 banks), the DDR2-SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x00800000.
   Note: This address is given as an example only. The real address depends on implementation in the product.
9. An Extended Mode Register Set (EMRS3) cycle is issued to set the Extended Mode register to 0. The application must configure the MODE field to 5 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signal BA[1] is set to 1 and signal BA[0] is set to 1. For example: with a 16-bit, 128-Mbit, DDR2-SDRAM (12 rows, 9 columns, 4 banks), the DDR2-SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x00C00000
10. An Extended Mode Register Set (EMRS1) cycle is issued to enable DLL and to program D.I.C. (Output Driver Impedance Control). The application must configure the MODE field to 5 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signal BA[1] is set to 0 and signal BA[0] is set to 1. For example: with a 16-bit, 128-Mbit, DDR2-SDRAM (12 rows, 9 columns, 4 banks), the DDR2-SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x00400000.
11. An additional 200 cycles of clock are required for locking DLL.
12. Write a ‘1’ to the DLL bit (enable DLL reset) in the Configuration register (MPDDRC_CR).
13. A Mode Register Set (MRS) cycle is issued to reset DLL. The application must configure the MODE field to 3 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signals BA[1:0] are set to 0. For example, the SDRAM write access should be done at the address: BASE_ADDRESS_DDR.
14. An All Banks Precharge command is issued to the DDR2-SDRAM. Program the All Banks Precharge command in the MPDDRC_MR. The application must configure the MODE field to 2 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any DDR2-SDRAM address to acknowledge this command.
15. Two autorefresh (CBR) cycles are provided. Program the Autorefresh command (CBR) in the MPDDRC_MR. The application must configure the MODE field to 4 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any DDR2-SDRAM location twice to acknowledge these commands. TRFC must be checked between two autorefreshes (see MPDDRC_TPR1).
16. Write a ‘0’ to the DLL bit (disable DLL reset) in the MPDDRC_CR.

Downloaded from Arrow.com.

17. A Mode Register Set (MRS) cycle is issued to program parameters of the DDR2-SDRAM device, in particular CAS latency and to disable DLL reset. The application must configure the MODE field to 3 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signals BA[1:0] are set to 0. For example: with a 16-bit, 128-Mbit, DDR2-SDRAM (12 rows, 9 columns, 4 banks) bank address, the SDRAM write access should be done at the address: BASE_ADDRESS_DDR.
18. Configure the OCD field (default OCD calibration) to 7 in the MPDDRC_CR.
19. An Extended Mode Register Set (EMRS1) cycle is issued to the default OCD value. The application must configure the MODE field to 5 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signal BA[1] is set to 0 and signal BA[0] is set to 1. For example: with a 16-bit, 128-Mbit, DDR2-SDRAM (12 rows, 9 columns, 4 banks), the DDR2-SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x00400000.
20. Configure the OCD field (exit OCD calibration mode) to 0 in the MPDDRC_CR.
21. An Extended Mode Register Set (EMRS1) cycle is issued to enable OCD exit. The application must configure the MODE field to 5 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signal BA[1] is set to 0 and signal BA[0] is set to 1. For example: with a 16-bit, 128-Mbit, DDR2-SDRAM (12 rows, 9 columns, 4 banks) bank address, the DDR2-SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x00400000.
22. A Normal Mode command is provided. Program the Normal mode in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any DDR2-SDRAM address to acknowledge this command.
23. Write the refresh rate into the COUNT field in the Refresh Timer register (MPDDRC_RTR). To compute the value, see MPDDRC Refresh Timer Register.

After initialization, the DDR2-SDRAM devices are fully functional.

31.5 Functional Description

31.5.1 DDR-SDRAM Controller Write Cycle

The MPDDRC provides burst access or single access in Normal mode (MPDDRC_MR.MODE = 0). Whatever the access type, the MPDDRC keeps track of the active row in each bank, thus maximizing performance.

The DDR-SDRAM device is programmed with a burst length (bl) equal to 8. This determines the length of a sequential data input by the write command that is set to 8. The latency from write command to data input depends on the memory type, as shown in the following table.

Table 31-1. CAS Write Latency

Memory Devices	CAS Write Latency (CWL)
Low-power DDR1-SDRAM	1
DDR2-SDRAM	2

To initiate a single access, the MPDDRC checks if the page access is already open. If row/bank addresses match with the previous row/bank addresses, the controller generates a write command. If the bank addresses are not identical or if bank addresses are identical but the row addresses are not identical, the controller generates a precharge command, activates the new row and initiates a write command. To comply with DDR-SDRAM timing parameters, additional clock cycles are inserted between precharge/active (t_{RP}) commands and active/write (t_{RCD}) commands. As the burst length is set to 8, in case of single access, it has to stop the burst, otherwise seven invalid values may be written. In case of the DDR-SDRAM device, the burst stop command is not supported for the burst write operation. Thus, in order to interrupt the write operation, the DM (data mask) input signal must be set to 1 to mask invalid data (see Figures Single Write Access, Row Closed, DDR-SDRAM Devices and Burst Write Access, Row Closed, DDR-SDRAM Devices), and DQS must continue to toggle.

Downloaded from Arrow.com.

To initiate a burst access, the MPDDRC uses the transfer type signal provided by the master requesting the access. If the next access is a sequential write access, writing to the DDR-SDRAM device is carried out. If the next access is a write non-sequential access, then an automatic access break is inserted, the MPDDRC generates a precharge command, activates the new row and initiates a write command. To comply with DDR-SDRAM timing parameters, additional clock cycles are inserted between precharge/active (t_{RP}) commands and active/write (t_{RCD}) commands.

For the definition of timing parameters, see MPDDRC Timing Parameter 0 Register.

Write accesses to the DDR-SDRAM device are burst oriented and the burst length is programmed to 8. It determines the maximum number of column locations that can be accessed for a given write command. When the write command is issued, eight columns are selected. All accesses for that burst take place within these eight columns, thus the burst wraps within these eight columns if a boundary is reached. These eight columns are selected by addr[13:3]. addr[2:0] is used to select the starting location within the block.

In case of incrementing burst (INCR/INCR4/INCR8/INCR16), the addresses can cross the 16-byte boundary of the DDR-SDRAM device. For example, when a transfer (INCR4) starts at address 0x0C, the next access is 0x10, but since the burst length is programmed to 8, the next access is at 0x00. Since the boundary is reached, the burst is wrapped. The MPDDRC takes this feature of the DDR-SDRAM device into account. In case of a transfer starting at address 0x04/0x08/0x0C or starting at address 0x10/0x14/0x18/0x1C, two write commands are issued to avoid wrapping when the boundary is reached. The last write command is subject to DM input logic level. If DM is registered high, the corresponding data input is ignored and the write access is not done. This avoids additional writing.

Figure 31-2. Single Write Access, Row Closed, DDR-SDRAM Devices

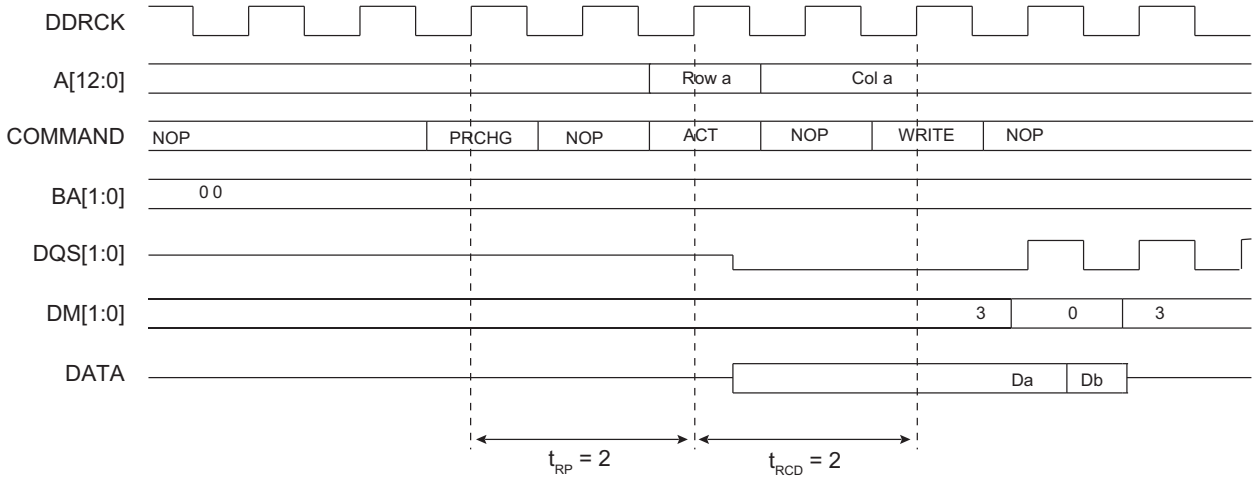
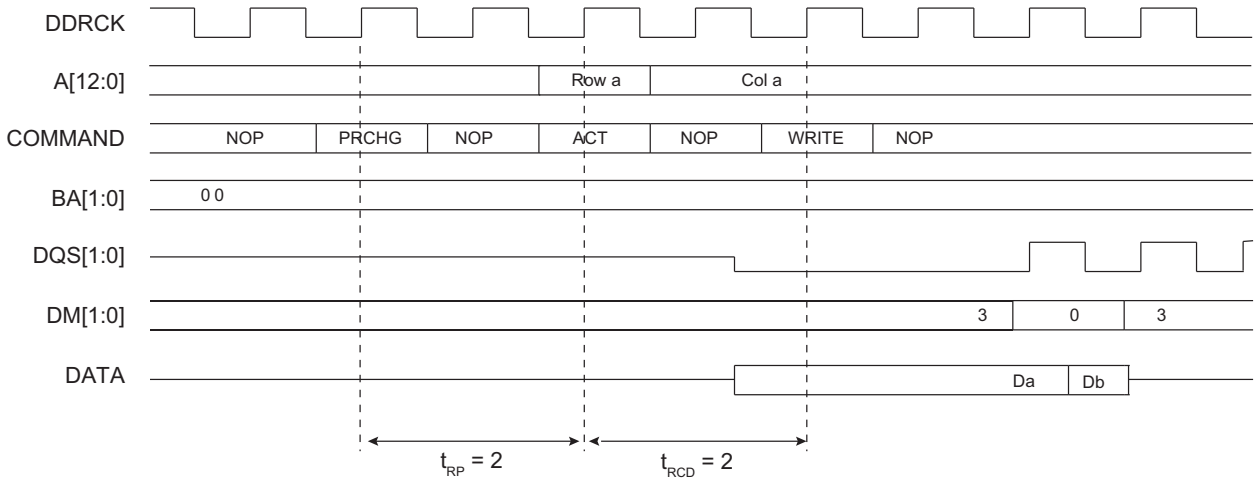


Figure 31-3. Single Write Access, Row Closed, DDR2-SDRAM Devices



Downloaded from Arrow.com.

Figure 31-4. Burst Write Access, Row Closed, DDR-SDRAM Devices

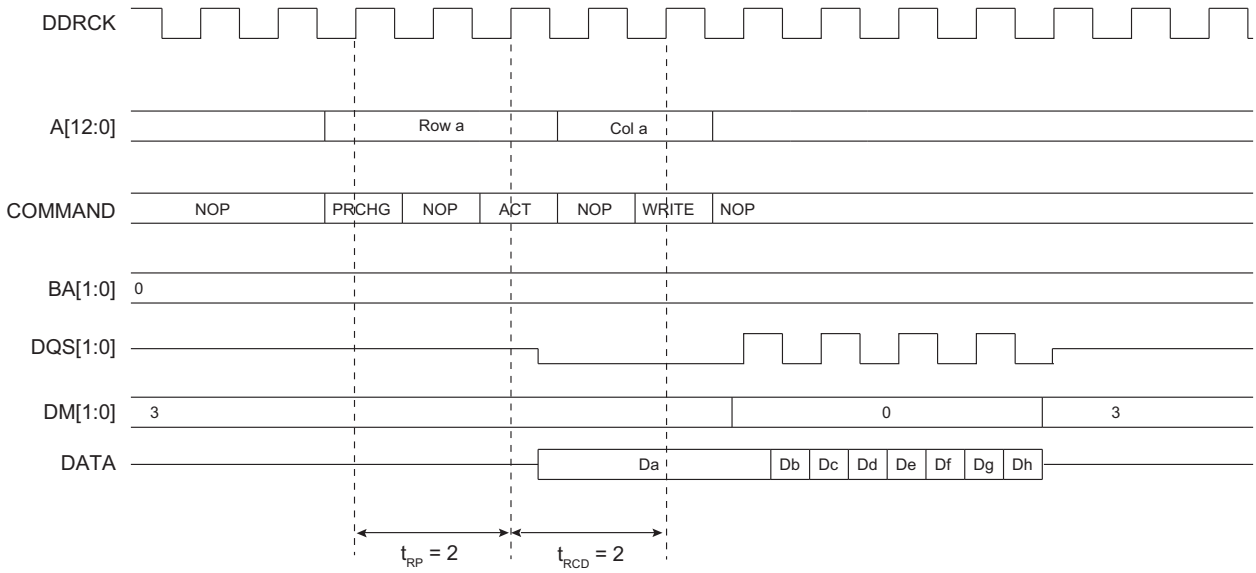
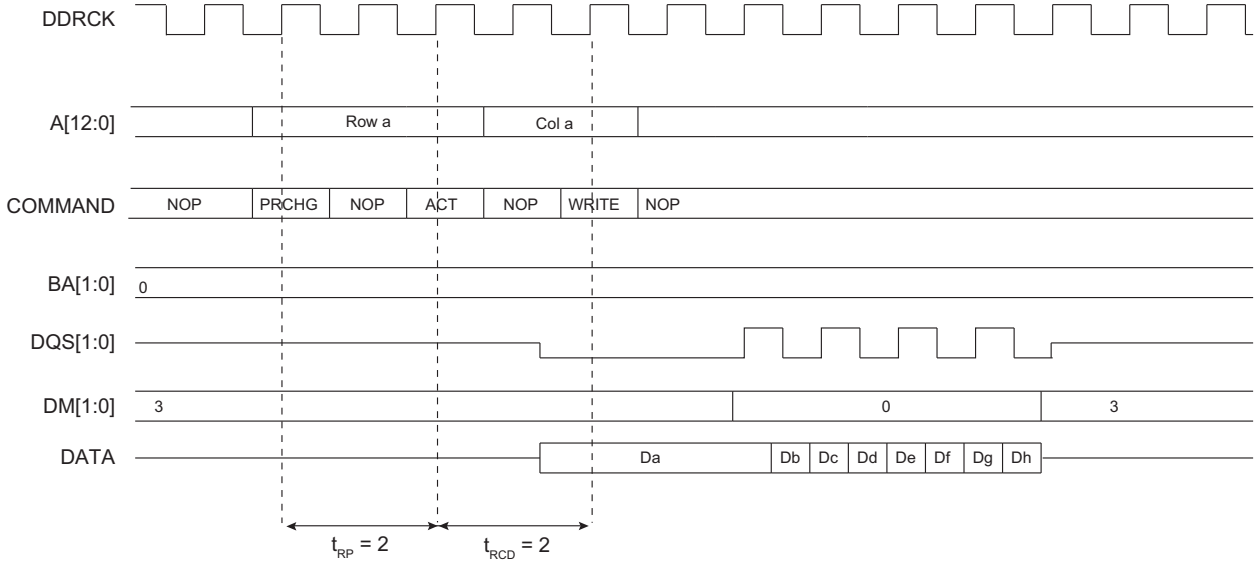


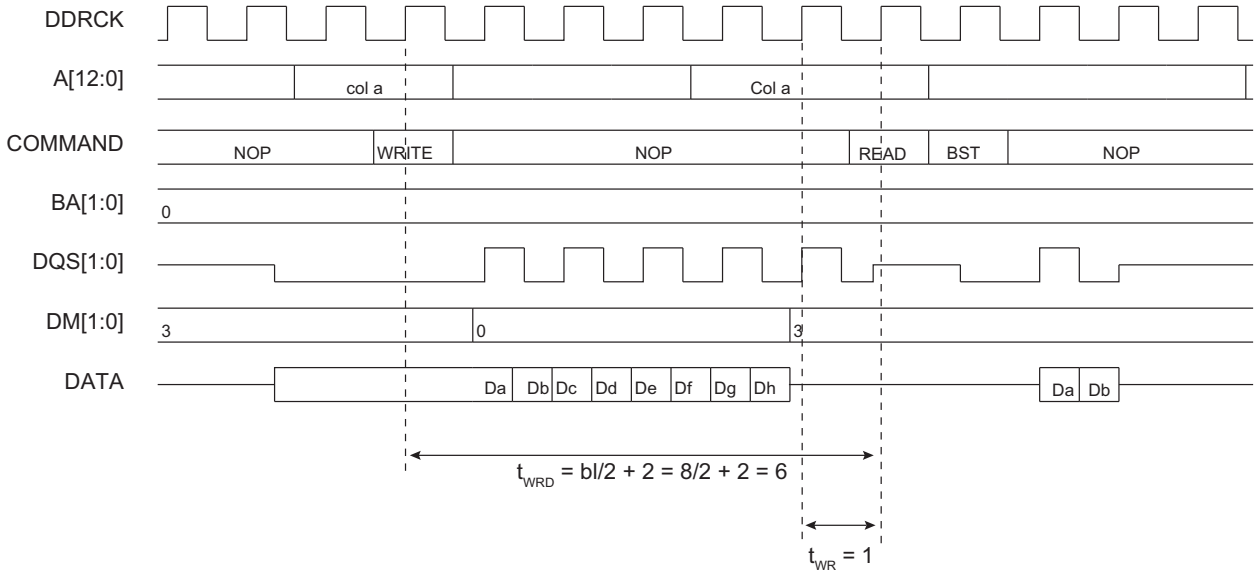
Figure 31-5. Burst Write Access, Row Closed, DDR2-SDRAM Devices



A write command can be followed by a read command. To avoid breaking the current write burst, t_{WTR}/t_{WRD} (bl/2 + 2 = 6 cycles) should be met. See the figure below.

Downloaded from Arrow.com.

Figure 31-6. Write Command Followed by a Read Command without Burst Write Interrupt, DDR-SDRAM Devices



In case of a single write access, write operation should be interrupted by a read access but DM must be input 1 cycle prior to the read command to avoid writing invalid data. See the figure below.

Figure 31-7. Single Write Access Followed by a Read Access, DDR-SDRAM Devices

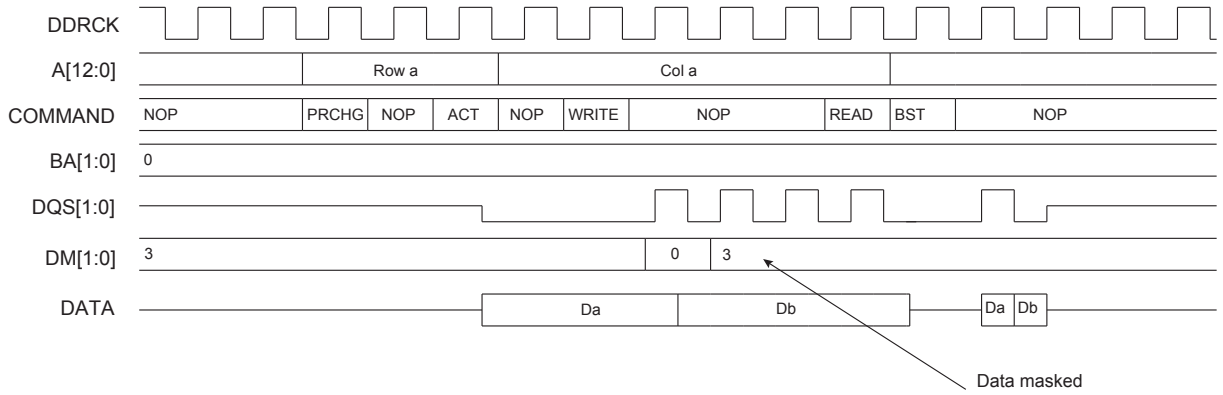
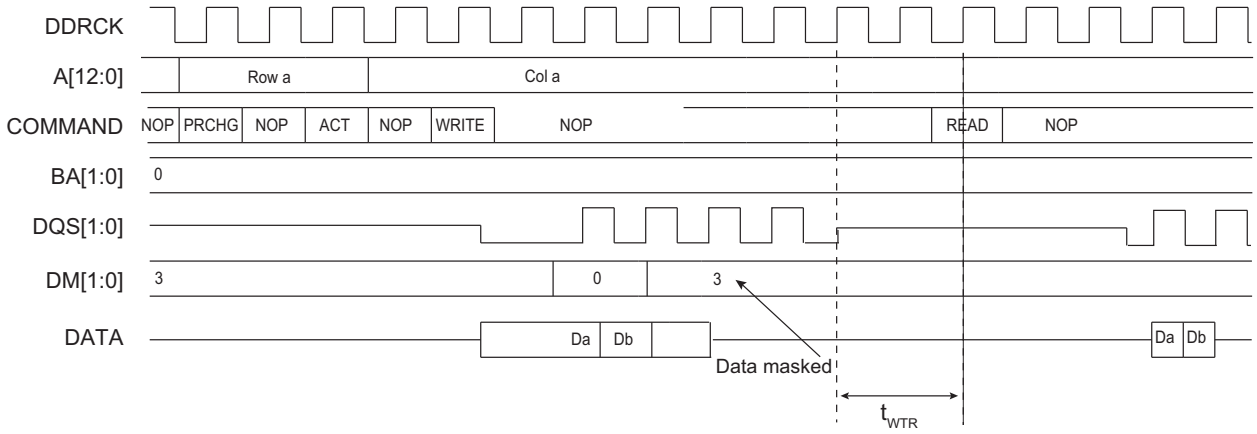


Figure 31-8. Single Write Access Followed by a Read Access, DDR2-SDRAM Devices



Downloaded from Arrow.com.

31.5.2 DDR-SDRAM Controller Read Cycle

The MPDDRC provides burst access or single access in Normal mode (MPDDRC_MR.MODE = 0). Whatever the access type, the MPDDRC keeps track of the active row in each bank, thus maximizing performance of the MPDDRC.

The DDR-SDRAM devices are programmed with a burst length equal to 8 which determines the length of a sequential data output by the read command that is set to 8. The latency from read command to data output depends on the memory type, as shown in the following table. This value is programmed during the initialization phase (see Product Dependencies, Initialization Sequence).

Table 31-2. CAS Read Latency

Memory Devices	CAS Read Latency
Low-power DDR1-SDRAM	2/3
DDR2-SDRAM	3

To initiate a single access, the MPDDRC checks if the page access is already open. If row/bank addresses match with the previous row/bank addresses, the controller generates a read command. If the bank addresses are not identical or if bank addresses are identical but the row addresses are not identical, the controller generates a precharge command, activates the new row and initiates a read command. To comply with DDR-SDRAM timing parameters, additional clock cycles are inserted between precharge/active (t_{RP}) commands and active/read (t_{RCD}) commands. After a read command, additional wait states are generated to comply with CAS latency. The MPDDRC supports a CAS latency delay of 2, 3 clock cycles. As the burst length is set to 8, in case of a single access or a burst access inferior to 8 data requests, it has to stop the burst, otherwise an additional seven or X values could be read. The Burst Stop command (BST) is used to stop output during a burst read. If the DDR2-SDRAM Burst Stop command is not supported by the JEDEC standard, in a single read access, an additional seven unwanted data will be read.

To initiate a burst access, the MPDDRC checks the transfer type signal. If the next accesses are sequential read accesses, reading to the SDRAM device is carried out. If the next access is a read non-sequential access, then an automatic page break can be inserted. If the bank addresses are not identical or if bank addresses are identical but the row addresses are not identical, the controller generates a precharge command, activates the new row and initiates a read command. If page access is already open, a read command is generated.

To comply with DDR-SDRAM timing parameters, additional clock cycles are inserted between precharge/active (t_{RP}) commands and active/read (t_{RCD}) commands. The MPDDRC supports a CAS latency delay of 2, 3 clock cycles. During this delay, the controller uses internal signals to anticipate the next access and improve the performance of the controller. Depending on the latency, the MPDDRC anticipates 2, 3 read accesses. In case of burst of specified length, accesses are not anticipated, but if the burst is broken (border, Busy mode, etc.), the next access is treated as an incrementing burst of unspecified length, and depending on the latency, the MPDDRC anticipates 2, 3 read accesses.

For the definition of timing parameters, see MPDDRC Configuration Register.

Read accesses to the DDR-SDRAM are burst oriented and the burst length is programmed to 8. The burst length determines the maximum number of column locations that can be accessed for a given read command. When the read command is issued, eight columns are selected. All accesses for that burst take place within these eight columns, meaning that the burst wraps within these eight columns if the boundary is reached. These eight columns are selected by addr[13:3]; addr[2:0] is used to select the starting location within the block.

In case of incrementing burst (INCR/INCR4/INCR8/INCR16), the addresses can cross the 16-byte boundary of the DDR-SDRAM device. For example, when a transfer (INCR4) starts at address 0x0C, the next access is 0x10, but since the burst length is programmed to 8, the next access is 0x00. Since the boundary is reached, the burst wraps. The MPDDRC takes into account this feature of the SDRAM device. In case of the DDR-SDRAM device, transfers start at address 0x04/0x08/0x0C. Two read commands are issued to avoid wrapping when the boundary is reached. The last read command may generate additional reading (1 read cmd = 4 DDR words).

To avoid additional reading, it is possible to use the burst stop command to truncate the read burst and to decrease power consumption. The DDR2-SDRAM devices do not support the burst stop command.

Downloaded from Arrow.com.

Figure 31-9. Single Read Access, Row Closed, Latency = 2, DDR-SDRAM Devices

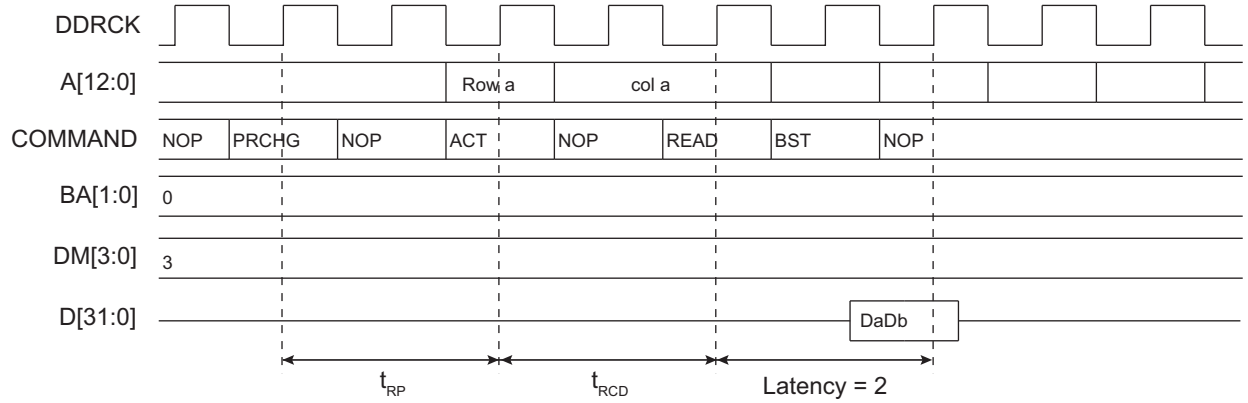


Figure 31-10. Single Read Access, Row Closed, Latency = 3, DDR2-SDRAM Devices

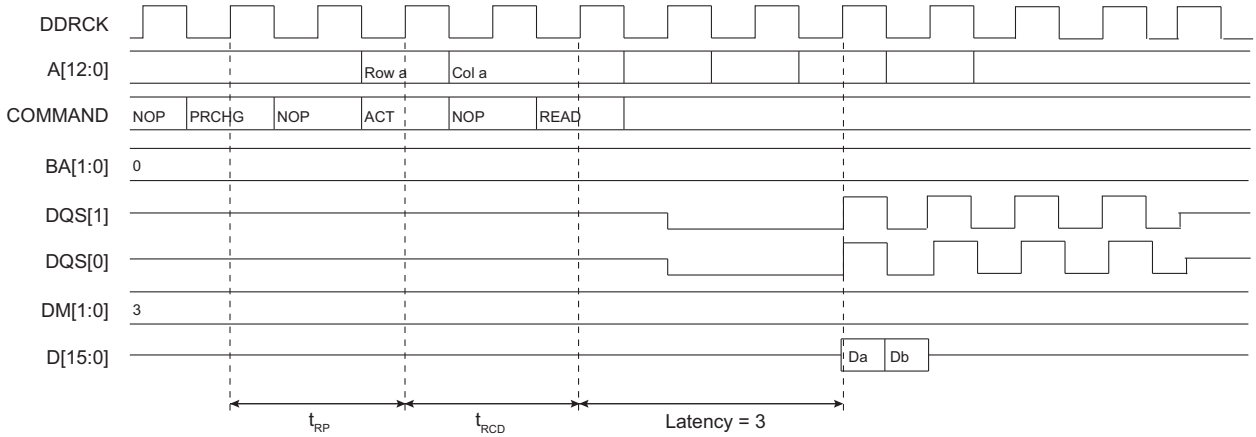
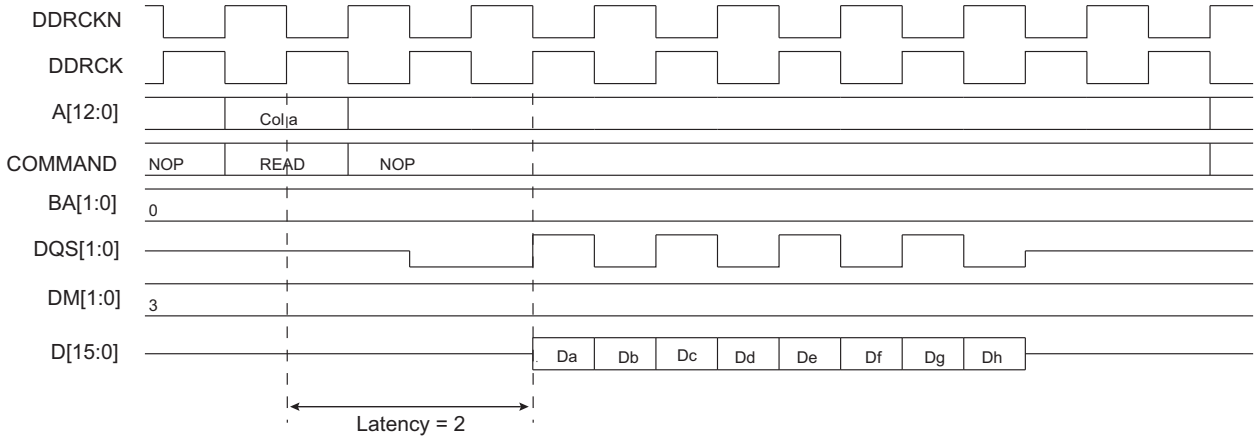
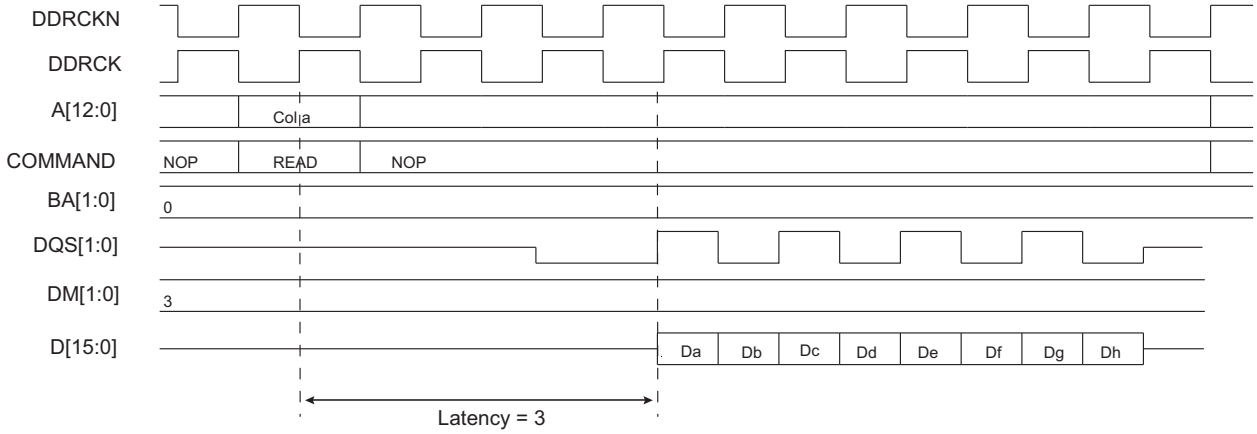


Figure 31-11. Burst Read Access, Latency = 2, DDR-SDRAM Devices



Downloaded from Arrow.com.

Figure 31-12. Burst Read Access, Latency = 3, DDR2-SDRAM Devices



31.5.3 Refresh (Autorefresh Command)

An Autorefresh command is used to refresh the external SDRAM devices. Refresh addresses are generated internally by the DDR-SDRAM device and incremented automatically after each autorefresh. The MPDDRC generates these autorefresh commands periodically. A timer is loaded in the MPDDRC_RTR with the value which indicates the number of clock cycles between refresh cycles (see MPDDRC Refresh Timer Register). When the MPDDRC initiates a refresh of the DDR-SDRAM device, internal memory accesses are not delayed. However, if the CPU tries to access the DDR-SDRAM device, the slave indicates that the device is busy. A refresh request does not interrupt a burst transfer in progress.

31.5.4 Power Management

31.5.4.1 Self-refresh Mode

This mode is activated by configuring the Low-power Command bit (LPCB) to 1 in the MPDDRC Low-Power Register (MPDDRC_LPR).

Self-refresh mode is used in Powerdown mode, i.e., when no access to the DDR-SDRAM device is possible. In this case, power consumption is very low. In Self-refresh mode, the DDR-SDRAM device retains data without external clocking and provides its own internal clocking, thus performing its own autorefresh cycles. During the self-refresh period, CKE is driven low. As soon as the DDR-SDRAM device is selected, the MPDDRC provides a sequence of commands and exits Self-refresh mode.

The MPDDRC reenables Self-refresh mode as soon as the DDR-SDRAM device is not selected. It is possible to define when Self-refresh mode is to be enabled by configuring the TIMEOUT field in the MPDDRC_LPR:

0: Self-refresh mode is enabled as soon as the DDR-SDRAM device is not selected.

1: Self-refresh mode is enabled 64 clock cycles after completion of the last access.

2: Self-refresh mode is enabled 128 clock cycles after completion of the last access.

This controller also interfaces the low-power DDR-SDRAM. To optimize power consumption, the Low Power DDR SDRAM provides programmable self-refresh options comprised of Partial Array Self Refresh (full, half, quarter and 1/8 and 1/16 array).

Disabled banks are not refreshed in Self-refresh mode. This feature permits to reduce the self-refresh current. In case of low-power DDR1-SDRAM, the Extended Mode register controls this feature. It includes Temperature Compensated Self-refresh (TSCR) and Partial Array Self-refresh (PASR) parameters and the drive strength (DS) (see MPDDRC Low-Power Register). These parameters are set during the initialization phase. After initialization, as soon as the PASR/DS/TCSR fields are modified, the memory device Extended Mode register are automatically accessed. Thus if MPDDRC does not share an external bus with another controller, PASR/DS/TCSR bits are updated before entering Self-refresh mode or during a refresh command. If MPDDRC does share an external bus with another controller, PASR/DS/TCSR bits are also updated during a pending read or write access. This type of update depends on the UPD_MR bit (see MPDDRC Low-Power Register).

Downloaded from Arrow.com.

The low-power DDR1-SDRAM must remain in Self-refresh mode during the minimum of TRFC periods (see MPDDRC Timing Parameter 1 Register), and may remain in Self-refresh mode for an indefinite period.

The DDR2-SDRAM must remain in Self-refresh mode during the minimum of t_{CKE} periods (see the memory device datasheet), and may remain in Self-refresh mode for an indefinite period.

Figure 31-13. Self-refresh Mode Entry, TIMEOUT = 0

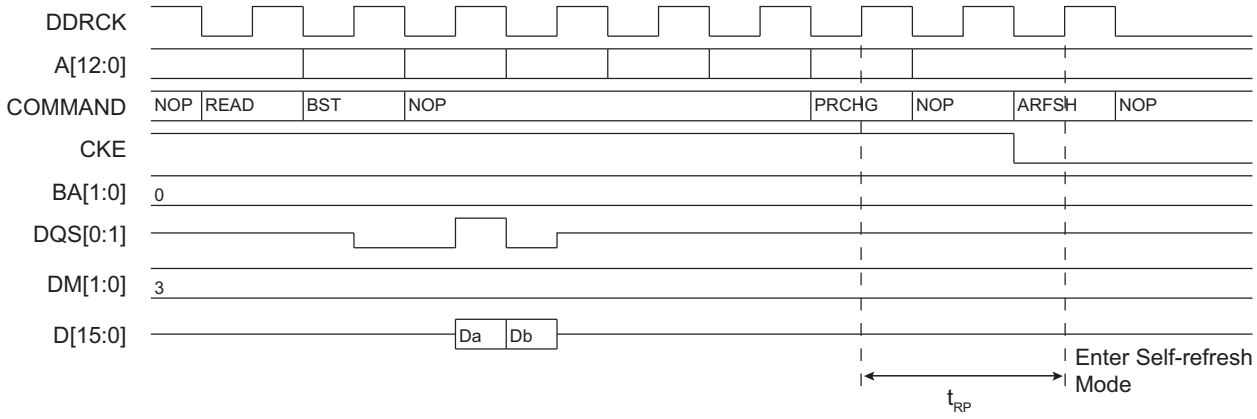


Figure 31-14. Self-refresh Mode Entry, TIMEOUT = 1 or 2

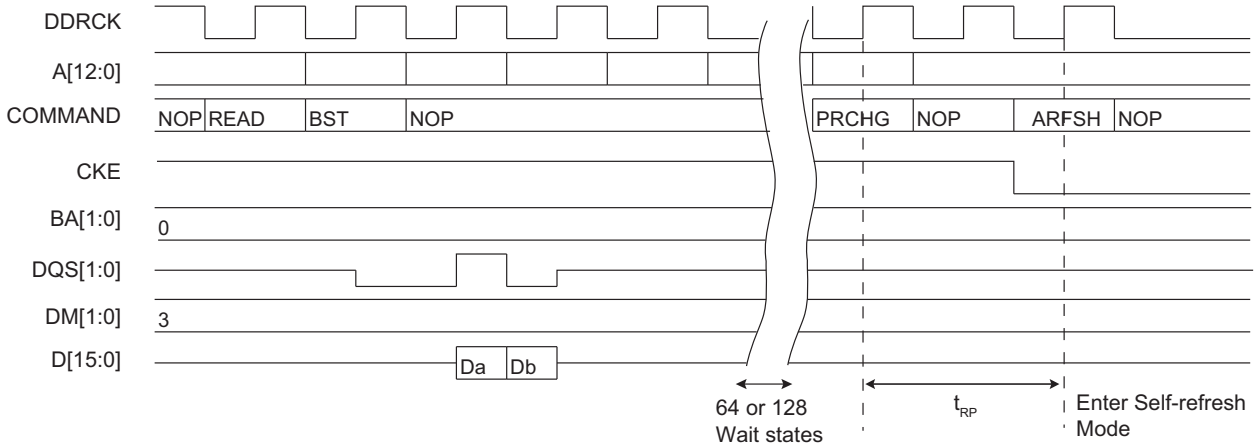
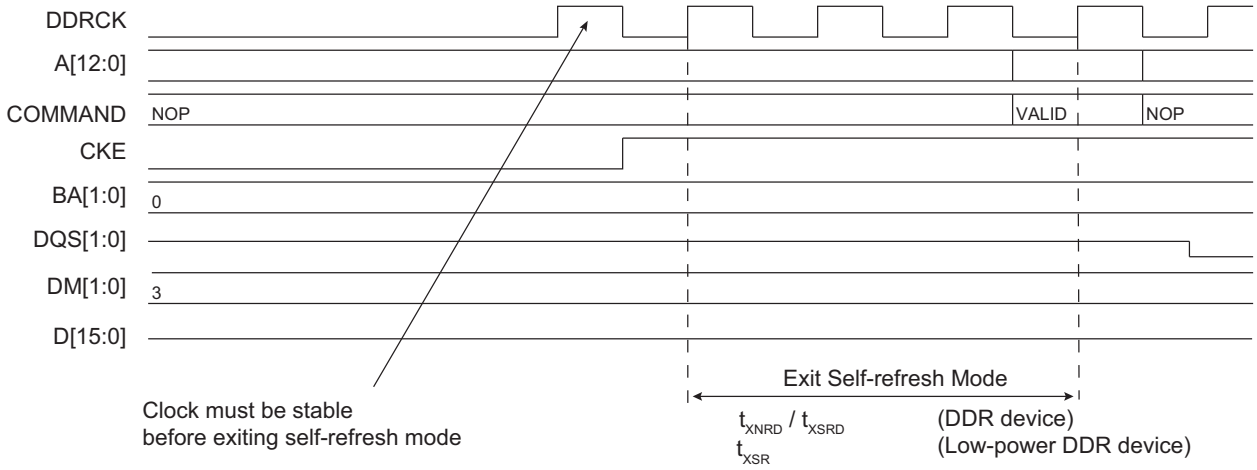


Figure 31-15. Self-refresh Mode Exit



Downloaded from Arrow.com.

31.5.4.2 Powerdown Mode

This mode is activated by configuring the Low-power Command bit (LPCB) to 2 in the MPDDRC Low-Power Register (MPDDRC_LPR).

Powerdown mode is used when no access to the DDR-SDRAM device is possible. In this mode, power consumption is greater than in Self-refresh mode. This state is similar to Normal mode (no Low-power mode/no Self-refresh mode), but the CKE pin is low and the input and output buffers are deactivated as soon the DDR-SDRAM device is no longer accessible. In contrast to Self-refresh mode, the DDR-SDRAM device cannot remain in Low-power mode longer than one refresh period (64 ms/32 ms). As no autorefresh operations are performed in this mode, the MPDDRC carries out the refresh operation. For the low-power DDR-SDRAM devices, a NOP command must be generated for a minimum period defined in the TXP field of the Timing Parameter 1 register (MPDDRC_TPR1). For DDR-SDRAM devices, a NOP command must be generated for a minimum period defined in the TXP field of MPDDRC_TPR1 (see MPDDRC Timing Parameter 1 Register) and in the TXARD and TXARDS fields of MPDDRC_TPR2 (see MPDDRC Timing Parameter 2 Register) for DDR2_SDRAM devices. In addition, low-power DDR-SDRAM and DDR-SDRAM must remain in Powerdown mode for a minimum period corresponding to t_{CKE}, t_{PD}, etc. (refer to the memory device datasheet).

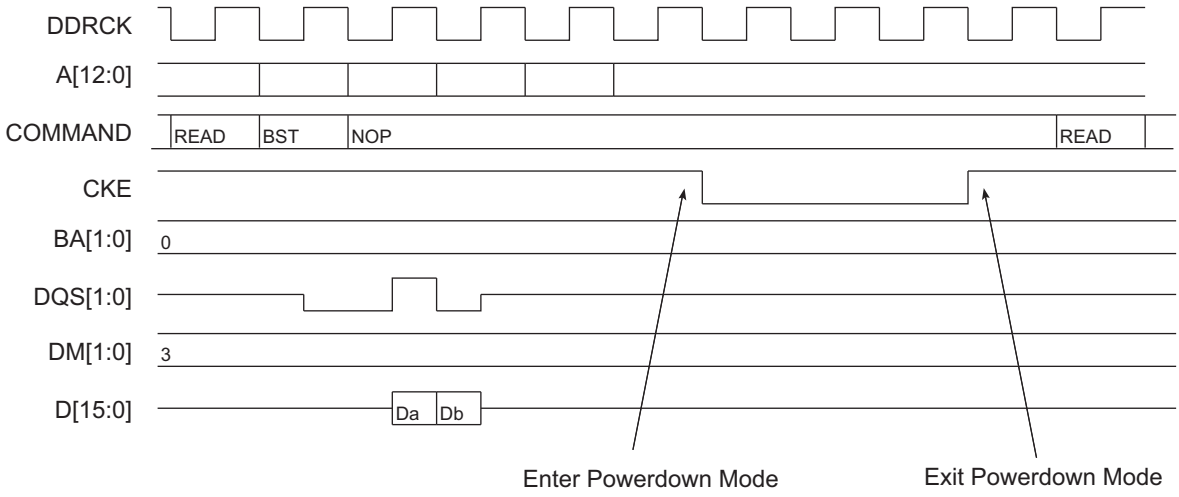
The exit procedure is faster than in Self-refresh mode. See the following figure. The MPDDRC returns to Powerdown mode as soon as the DDR-SDRAM device is not selected. It is possible to define when Powerdown mode is enabled by configuring the TIMEOUT field in the MPDDRC_LPR:

0: Powerdown mode is enabled as soon as the DDR-SDRAM device is not selected.

1: Powerdown mode is enabled 64 clock cycles after completion of the last access.

2: Powerdown mode is enabled 128 clock cycles after completion of the last access.

Figure 31-16. Powerdown Entry/Exit, TIMEOUT = 0



31.5.4.3 Deep Powerdown Mode

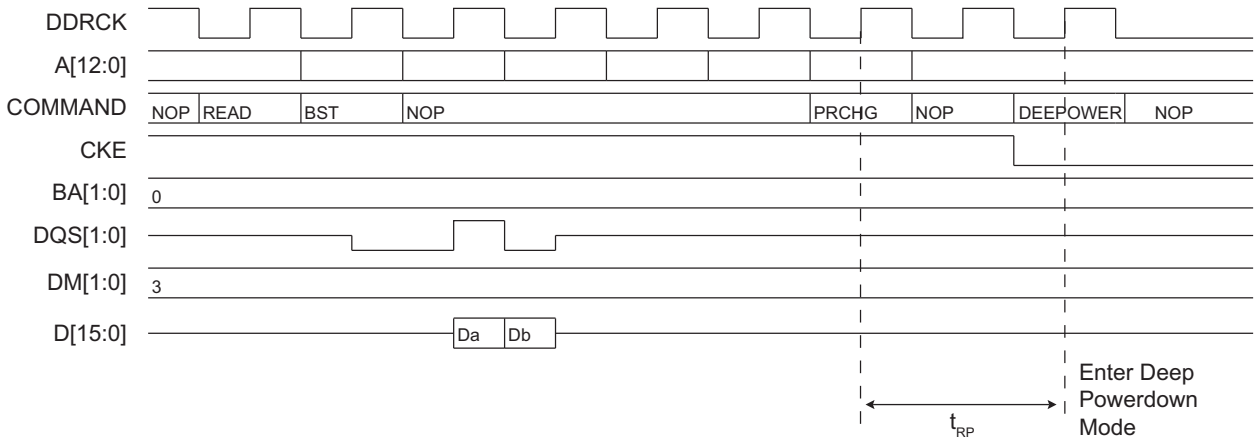
The Deep Powerdown mode is a feature of low-power DDR-SDRAM. When this mode is activated, all internal voltage generators inside the device are stopped and all data is lost.

Deep Powerdown mode is activated by configuring the Low-power Command bit (LPCB) to 3 in the MPDDRC Low-Power Register (MPDDRC_LPR). When this mode is enabled, the MPDDRC leaves Normal mode (MPDDRC_MR.MODE = 0) and the controller is frozen. The clock can be stopped during Deep Powerdown mode by setting the CLK_FR field to 1.

Before enabling this mode, the user must make sure there is no access in progress. To exit Deep Powerdown mode, the Low-power Command bit (LPCB) and Clock Frozen bit (CLK_FR) must be 0 and the initialization sequence must be generated by software. See Low-power DDR1-SDRAM Initialization.

Downloaded from Arrow.com.

Figure 31-17. Deep Powerdown Mode Entry



31.5.4.4 Change Frequency During Self-Refresh Mode with Low-power DDR-SDRAM Devices

To change frequency, Self-refresh mode must be activated. This is done by configuring the Low-power Command bit (LPCB) to 1 and writing a '1' to the Change Frequency Command bit (CHG_FR) in the Low-power register (MPDDRC_LPR).

Once the low-power DDR-SDRAM is in Self-refresh mode, the user must make sure there is no access in progress. Then, the user can change the clock frequency. The device input clock frequency changes only within minimum and maximum operating frequencies as specified by the low-power DDR-SDRAM providers. Once the input clock frequency is changed, new stable clocks must be provided to the device before exiting from Self-refresh mode.

To exit from Self-refresh mode, the DDR-SDRAM device must be selected. The MPDDRC provides a sequence of commands and exits Self-refresh mode.

During a change frequency procedure, the Change Frequency Command bit (CHG_FR) is set to 0 automatically.

It is not possible to change the frequency with DDR2-SDRAM devices.

Before changing frequency, make sure the processor clock (PCK) value is twice the system bus clock (MCK) value.

31.5.4.5 Reset Mode

The Reset mode is a feature of DDR2-SDRAM. This mode is activated by configuring the Low-power Command bit (LPCB) to 3 and writing a '1' to the Clock Frozen Command bit (CLK_FR) in the Low-power register (MPDDRC_LPR).

When this mode is enabled, the MPDDRC leaves Normal mode (MPDDRC_MR.MODE = 0) and the controller is frozen. Before enabling this mode, the user must make sure there is no access in progress.

To exit Reset mode, the Low-power Command bit (LPCB) must be configured to 0, the Clock Frozen Command bit (CLK_FR) must be written to '0' and the initialization sequence must be generated by software (see DDR2-SDRAM Initialization).

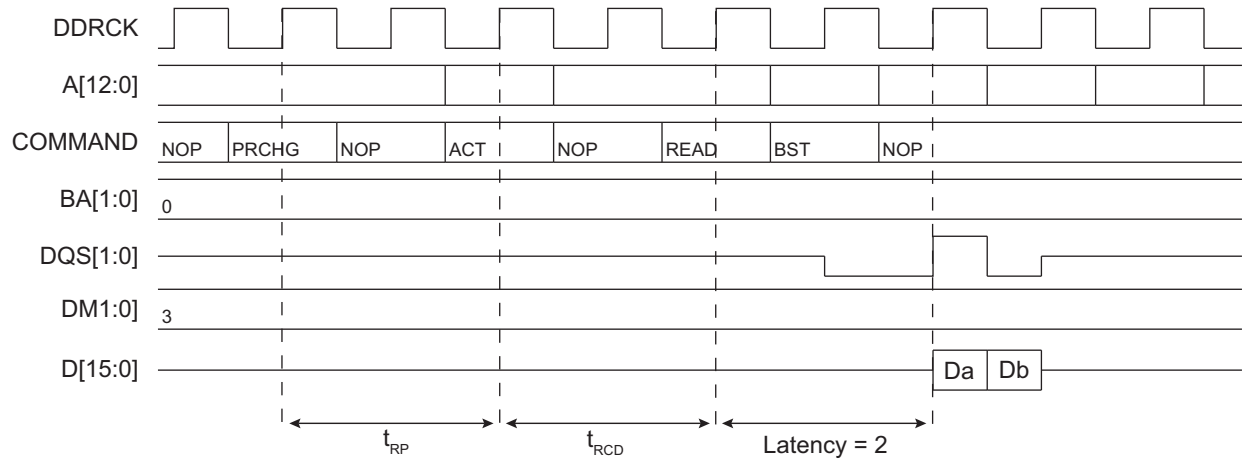
31.5.5 Multiport Functionality

The DDR-SDRAM protocol imposes a check of timings prior to performing a read or a write access, thus decreasing system performance. An access to DDR-SDRAM is performed if banks and rows are open (or active). To activate a row in a particular bank, the last open row must be deactivated and a new row must be open. Two DDR-SDRAM commands must be performed to open a bank: Precharge command and Activate command with respect to T_{RP} timing. Before performing a read or write command, T_{RCD} timing must be checked.

This operation generates a significant bandwidth loss (see the following figure).

Downloaded from Arrow.com.

Figure 31-18. T_{RP} and T_{RCD} Timings

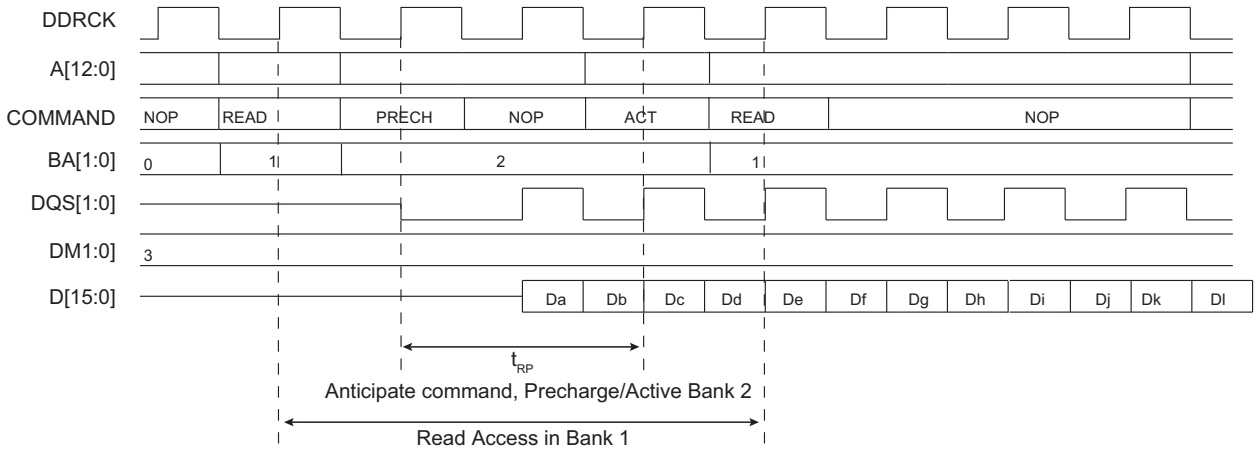


4 cycles before performing a read command

The multiport controller is designed to mask these timings and thus improve the system bandwidth.

The MPDDRC is a multiport controller whereby four masters can simultaneously reach the controller. This feature improves the bandwidth of the system because it can detect four requests on the AHB slave inputs and thus anticipate the commands that follow, Precharge command and Activate command in bank X during the current access in bank Y. This masks t_{RP} and t_{RCD} timings (see the following figure). In the best case, all accesses are done as if the banks and rows were already open. The best condition is met when the four masters work in different banks. In the case of four simultaneous read accesses, when the four or eight banks and associated rows are open, the controller reads with a continuous flow and masks the CAS latency for each access. To allow a continuous flow, the read command must be set at 2, 3 cycles (CAS latency) before the end of the current access. The arbitration scheme must be changed since the round-robin arbitration cannot be respected. If the controller anticipates a read access, and thus a master with a high priority arises before the end of the current access, then this master will not be serviced.

Figure 31-19. Anticipate Precharge/Activate Command in Bank 2 during Read Access in Bank 1



MPDDRC is a multiport controller that embeds three arbitration mechanisms based on round-robin arbitration which allows to share the external device between different masters when two or more masters try to access the DDR-SDRAM device at the same time.

The three arbitration types are round-robin arbitration and two weighted round-robin arbitrations. For weighted round-robin arbitrations, the priority can be given either depending on the number of requests or words per port, or depending on the required bandwidth per port. The type of arbitration can be chosen by setting the ARB field in the Configuration Arbiter register (MPDDRC_CONF_ARBITER) (see MPDDRC Configuration Arbiter Register).

Downloaded from Arrow.com.

31.5.5.1 Round-robin Arbitration

Round-robin arbitration is used when the ARB field is set to 0 (see MPDDRC Configuration Arbiter Register). This algorithm dispatches the requests from different masters to the DDR-SDRAM device in a round-robin manner. If two or more master requests arise at the same time, the master with the lowest number is serviced first, then the others are serviced in a round-robin manner.

To avoid burst breaking and to provide the maximum throughput for the DDR-SDRAM device, arbitration must only take place during the following cycles:

1. Idle cycles: when no master is connected to the DDR-SDRAM device.
2. Single cycles: when a slave is currently doing a single access.
3. End of Burst cycles: when the current cycle is the last cycle of a burst transfer:
   - For bursts of defined length, predicted end of burst matches the size of the transfer.
   - For bursts of undefined length, predicted end of burst is generated at the end of each four-beat boundary inside the INCR transfer.
4. Anticipated Access: when an anticipated read access is done while the current access is not complete, the arbitration scheme can be changed if the anticipated access is not the next access serviced by the arbitration scheme.

31.5.5.2 Request-word Weighted Round-robin Arbitration

In request-word weighted round-robin arbitration, the weight is the number of requests or the number of words per port.

This arbitration scheme is enabled by configuring the ARB field to 1 (see MPDDRC Configuration Arbiter Register). This algorithm grants a port for X[(1)] consecutive first transfer (htrans = NON SEQUENTIAL) of a burst or X single transfer, or for X word transfers. It is possible to choose between an arbitration scheme by request or by word per port by setting the RQ_WD_Px field (see MPDDRC Configuration Arbiter Register).

Note: 1. X is an integer value provided by some master modules to the arbiter.

It is also possible for the user to provide the number of requests or words (by overwriting the information provided by a master) on master basis by configuring the MA_PR_Px field. Depending on the application, it is possible to reduce or increase the number of these requests or words by configuring the NRD_NWD_BDW_Px fields (see MPDDRC Configuration Arbiter Register).

The TIMEOUT_Px field defines the delay between two accesses on the same port in number of cycles before rearbitrating the access to another port. This field allows to avoid a timeout on the system because some masters have the particularity to add idle cycles between two consecutive accesses (see MPDDRC Configuration Arbiter Register).

This algorithm dispatches the requests from different masters to the DDR-SDRAM device in a round-robin manner. If two or more master requests arise at the same time, the master with the lowest number is serviced first, then the others are serviced in a round-robin manner when the number of requests or words is reached or when the timeout value is reached.

To avoid burst breaking and to provide the maximum throughput for the DDR-SDRAM device, arbitration must only take place during the following cycles:

1. Timeout is reached: the delay between two accesses is equal to TIMEOUT_Px.
2. Number of requests or words is reached: when the current cycle is the last cycle of a transfer.

31.5.5.3 Bandwidth Weighted Round-robin Arbitration

In bandwidth weighted round-robin arbitration, a minimum bandwidth is guaranteed per port.

This arbitration scheme is enabled when the ARB field is set to 2 (see MPDDRC Configuration Arbiter Register).

This algorithm grants to each port a percentage of the bandwidth. The NRD_NWD_BDW_Px field defines the percentage allocated to each port.

The percentage of the bandwidth is programmed with the NRD_NWD_BDW_Px fields (see MPDDRC Configuration Arbiter Register).

The TIMEOUT_Px field defines the delay between two accesses on the same port in number of cycles rearbitrating the access to another port. This field allows to avoid a timeout on the system because some masters have the particularity to add idle cycles between two consecutive accesses (see MPDDRC Configuration Arbiter Register).

Downloaded from Arrow.com.

This algorithm dispatches the requests from different masters to the DDR-SDRAM device in a round-robin manner. If two or more master requests arise at the same time, the master with the lowest number is serviced first, then the others are serviced in a round-robin manner when the allocated bandwidth is reached or when the timeout value is reached.

The BDW_BURST field allows to arbitrate either when the current master reaches exactly the programmed bandwidth, or when the current master reaches exactly the programmed bandwidth and the current access is ended (see MPDDRC Configuration Arbiter Register).

To provide the maximum throughput for the DDR-SDRAM device, arbitration must only take place during the following cycles:

1. Timeout is reached: the delay between two accesses is equal to TIMEOUT_Px.
2. Allocated Bandwidth is reached although the current cycle is not ended.
3. Allocated Bandwidth is reached and the current cycle is the last cycle of a transfer.

31.5.5.4 Quality Of Service Arbitration

This arbitration scheme is enabled when the ARB field is set to 3 (see MPDDRC Configuration Arbiter Register).

The arbitration scheme is organized in priority pools corresponding each to an access criticality class as shown in the corresponding Latency Quality of Service column in the following table. When the Latency Quality of Service is enabled for a master-slave pair through the Bus Matrix (refer to AHB Bus Matrix section), the priority pool number to use for arbitration at the slave port is determined from the master. When the Latency Quality of Service is disabled, it is determined through the Bus Matrix user interface. Refer to “Bus Matrix Priority Registers A For Slaves” in AHB Bus Matrix section.

Table 31-3. Arbitration Priority Pools

Priority pool	Latency Quality of Service
3	Latency Critical
2	Latency Sensitive
1	Bandwidth Sensitive
0	Background Transfers

Round-robin priority is used in the highest and lowest priority pools 3 and 0, whereas fixed level priority is used between priority pools and in the intermediate priority pools 2 and 1.

For each slave, each master is assigned to one of the slave priority pools based on the Latency Quality of Service inputs or to the priority registers for slaves (MxPR fields of MATRIX_PRAS and MATRIX_PRBS, refer to AHB Bus Matrix section). When evaluating master requests, this priority pool level always takes precedence.

After reset, most of the masters belong to the lowest priority pool (MxPR = 0, Background Transfer) and are therefore granted bus access in a true round-robin order.

The highest priority pool must be specifically reserved for masters requiring very low access latency. If more than one master belong to this pool, those masters are granted bus access in a biased round-robin manner which enables tight and deterministic maximum access latency from AHB bus requests. In the worst case, any currently occurring high-priority master request is granted after the current bus master access has ended and any other high priority pool master requests have been granted once each.

The lowest priority pool shares the remaining bus bandwidth between AHB masters.

Intermediate priority pools enable fine priority tuning. Typically, a latency-sensitive master or a bandwidth-sensitive master uses such a priority level. The higher the priority level (MxPR value, refer to AHB Bus Matrix section), the higher the master priority.

All combinations of MxPR values are allowed for all masters and slaves. For example, some masters might be assigned the highest priority pool (round-robin), and remaining masters the lowest priority pool (round-robin), with no master for intermediate fixed priority levels.

Some masters, such as LCD or DMA, drive a signal named HNBREQ on the system bus to indicate the number of transfers to be performed. When the field MPDDRC_CONF_ARBITER.KEEP_LAYER is set to 1, the master with the

Downloaded from Arrow.com.

highest LQOS value and a HNBREQ value different from 0 continues to be granted, even during a last data phase with IDLE cycles.

31.5.6 Scrambling/Unscrambling Function

The external data bus can be scrambled in order to prevent intellectual property data located in off-chip memories from being easily recovered by analyzing data at the package pin level of either the microcontroller or the memory device.

The scrambling and unscrambling are performed on-the-fly without additional wait states.

The scrambling method depends on two user-configurable key registers, KEY1 in the "MPDDRC OCMS KEY1 Register" and KEY2 in the "MPDDRC OCMS KEY2 Register". These key registers are only accessible in Write mode.

The key must be securely stored in a reliable non-volatile memory in order to recover data from the off-chip memory. Any data scrambled with a given key cannot be recovered if the key is lost.

The scrambling/unscrambling function can be enabled or disabled by programming the "MPDDRC OCMS Register".

31.5.7 Clearing Scrambling Keys on Tamper Event

On tamper detection event on WKUP pins, it is possible to perform an immediate clear of the scrambling keys (MPDDRC_OCMS_KEY1 and MPDDRC_OCMS_KEY2) if bit MPDDRC_OCMS.TAMPCLR = 1.

31.5.8 Register Write Protection

To prevent any single software error from corrupting MPDDRC behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the MPDDRC Write Protection Mode Register (MPDDRC_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the MPDDRC Write Protection Status Register (MPDDRC_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading MPDDRC_WPSR.

The following registers are write-protected when the bit WPEN is set:

- MPDDRC Mode Register
- MPDDRC Refresh Timer Register
- MPDDRC Configuration Register
- MPDDRC Timing Parameter 0 Register
- MPDDRC Timing Parameter 1 Register
- MPDDRC Memory Device Register
- MPDDRC OCMS Register
- MPDDRC OCMS KEY1 Register
- MPDDRC OCMS KEY2 Register

The following registers are write-protected when the bit WPITEN is set:

- MPDDRC Interrupt Enable Register
- MPDDRC Interrupt Disable Register

31.5.9 Monitor

The MPDDRC embeds a monitor which collects bus transaction information from four MPDDRC ports. This information, such as accumulated latency (MPDDRC_MINFOx (TOTAL_LATENCY) or number of transfers (MPDDRC_MINFOx (NB_TRANSFERS), can be used to calculate the average latency for each port.

Configuration registers (MPDDRRC_MCFGR, MPDDRRC_MADDRx) are used to define the type of transaction collected (read, write or read/write) and the address range snooped.

By default, the monitor is enabled and the address range is 0x3FFFFFFF (address high = 0xFFFF) to 0x20000000 (address low = 0000).

Monitor use example:

1. Clear the configuration register: write 0x00000000 in MPDDRRC_MCFGR.

Downloaded from Arrow.com.

2. Enable the monitor: write 0x00000001 in MPDDRRC_MCFGR.
3. Reset the monitor: write 0x00000003 in MPDDRRC_MCFGR.
4. Enable the monitor: write 0x00000001 in MPDDRRC_MCFGR.
5. Start profiling: write 0x00000011 in MPDDRRC_MCFGR.
6. Profiling is launched. An event can be used to stop monitoring.
7. Stop profiling: write 0x00000001 in MPDDRRC_MCFGR.
8. To know the number of tranfers per port, write 0x00000801 in MPDDRRC_MCFGR and read MPDDRC_MINFOx (NB_TRANSFERS).

31.5.10 Security and Safety Analysis and Reports

Several types of checks are performed when the MPDDRC is accessing the memory device.

The peripheral clock of the MPDDRC is monitored by specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the MPDDRC. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the flag MPDDRC_WPSR.CGD is set, an abnormal condition occurred on the peripheral clock. This flag is not set under normal operating conditions.

The internal sequencer of the MPDDRC is also monitored and if an abnormal state is detected, the flag MPDDRC_WPSR.SEQE is set. This flag is not set under normal operating conditions.

If the flag MPDDRC_WPSR.CGD = 1, a clock glitch has been detected. This flag is not set under normal operating conditions.

The software accesses to the MPDDRC are monitored and if an incorrect access is performed, the flag MPDDRC_WPSR.SWE is set. The type of incorrect/abnormal software access is reported in the MPDDRC_WPSR.SWETYP field (see MPDDRC Write Protection Status Register (MPDDRC_WPSR) for details), e.g., writing a new configuration (MPDDRC_CR, MPDDRC_TPR0/1/2, MPDDRC_MD, MPDDRC_OCMS, MPDDRC_OCMS_KEY1/2) after the initialization of the MPDDRC (i.e., if MPDDRC_TR.COUNT > 0) is an error. MPDDRC_WPSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The flags CGD, SEQE, SWE and WPVS are automatically cleared when MPDDRC_WPSR is read.

If one of these flags is set, the flag MPDDRC_ISR.SECE is set and can trigger an interrupt if the MPDDRC_IMR.SECE bit is ‘1’. SECE is cleared by reading MPDDRC_ISR.

The MPDDRC embeds an automatic periodic check of an address of the memory device. This function can be enabled by writing a 1 to the MPDDRC_SAFETY.EN bit. The address to be checked can be configured by writing the field MPDDRC_SAFETY.ADDRESS. When MPDDRC_SAFETY.EN = 1, the MPDDRC performs read and write accesses with specific, predetermined, data patterns to the configured address, with no impact for the application software,

31.6 Software Interface/SDRAM Organization, Address Mapping

The DDR-SDRAM address space is organized into banks, rows and columns. The MPDDRC maps different memory types depending on values set in the Configuration register (MPDDRC_CR) (see MPDDRC Configuration Register). The tables that follow illustrate the relation between CPU addresses and columns, rows and banks addresses for 16-bit memory data bus widths.

The MPDDRC supports address mapping in Linear mode.

Sequential mode is a method for address mapping where banks alternate at each last DDR-SDRAM page of the current bank.

Interleaved mode is a method for address mapping where banks alternate at each DDR-SDRAM end of page of the current bank.

The MPDDRC makes the DDR-SDRAM device access protocol transparent to the user. The tables that follow illustrate the DDR-SDRAM device memory mapping seen by the user in correlation with the device structure. Various configurations are illustrated.

Downloaded from Arrow.com.

31.6.1 DDR-SDRAM Address Mapping for 16-bit Memory Data Bus Width

Table 31-4. Sequential Mapping for DDR-SDRAM Configuration, 2K Rows, 256/512/1024/2048/4096 Columns, 4 Banks

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						Bk[1:0]	Row[10:0]												Column[7:0]								M0
					Bk[1:0]	Row[10:0]													Column[8:0]								M0
				Bk[1:0]	Row[10:0]												Column[9:0]										M0
			Bk[1:0]	Row[10:0]												Column[10:0]											M0
		Bk[1:0]	Row[10:0]												Column[11:0]												M0

Table 31-5. Interleaved Mapping for DDR-SDRAM Configuration, 2K Rows, 256/512/1024/2048/4096 Columns, 4 Banks

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							Row[10:0]										Bk[1:0]		Column[7:0]								M0
						Row[10:0]										Bk[1:0]			Column[8:0]								M0
					Row[10:0]										Bk[1:0]				Column[9:0]								M0
				Row[10:0]										Bk[1:0]					Column[10:0]								M0
			Row[10:0]										Bk[1:0]						Column[11:0]								M0

Table 31-6. Sequential Mapping for DDR-SDRAM Configuration: 4K Rows, 256/512/1024/2048/4096 Columns, 4 Banks

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					Bk[1:0]		Row[11:0]												Column[7:0]								M0
				Bk[1:0]		Row[11:0]													Column[8:0]								M0
			Bk[1:0]		Row[11:0]												Column[9:0]										M0
		Bk[1:0]		Row[11:0]												Column[10:0]											M0
	Bk[1:0]		Row[11:0]												Column[11:0]												M0

Table 31-7. Interleaved Mapping for DDR-SDRAM Configuration: 4K Rows, 256/512/1024/2048/4096 Columns, 4 Banks

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						Row[11:0]											Bk[1:0]		Column[7:0]								M0
					Row[11:0]											Bk[1:0]			Column[8:0]								M0
				Row[11:0]											Bk[1:0]				Column[9:0]								M0
			Row[11:0]											Bk[1:0]					Column[10:0]								M0

Downloaded from Arrow.com.

.....continued

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		Row[11:0]											Bk[1:0]		Column[11:0]												M0

Table 31-8. Sequential Mapping for DDR-SDRAM Configuration: 8K Rows, 512/1024/2048/4096 Columns, 4 Banks

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			Bk[1:0]		Row[12:0]													Column[8:0]									M0
		Bk[1:0]			Row[12:0]													Column[9:0]									M0
	Bk[1:0]				Row[12:0]													Column[10:0]									M0
Bk[1:0]					Row[12:0]													Column[11:0]									M0

Table 31-9. Interleaved Mapping for DDR-SDRAM Configuration: 8K Rows, 512/1024/2048/4096 Columns, 4 Banks

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			Row[12:0]													Bk[1:0]		Column[8:0]									M0
		Row[12:0]													Bk[1:0]			Column[9:0]									M0
	Row[12:0]													Bk[1:0]				Column[10:0]									M0
Row[12:0]													Bk[1:0]					Column[11:0]									M0

Table 31-10. Sequential Mapping for DDR-SDRAM Configuration: 16K Rows, 512/1024/2048 Columns, 4 Banks

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		Bk[1:0]		Row[13:0]														Column[8:0]									M0
	Bk[1:0]			Row[13:0]														Column[9:0]									M0
Bk[1:0]				Row[13:0]														Column[10:0]									M0

Table 31-11. Interleaved Mapping for DDR-SDRAM Configuration: 16K Rows, 512/1024/2048 Columns, 4 Banks

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		Row[13:0]														Bk[1:0]		Column[8:0]									M0
	Row[13:0]														Bk[1:0]			Column[9:0]									M0
Row[13:0]														Bk[1:0]				Column[10:0]									M0

Downloaded from Arrow.com.

Table 31-12. Sequential Mapping for DDR-SDRAM Configuration: 8K Rows, 1024 Columns, 8 Banks

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Bk[2:0]			Row[12:0]													Column[9:0]										M0

Table 31-13. Interleaved Mapping for DDR-SDRAM Configuration: 8K Rows, 1024 Columns, 8 Banks

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Row[12:0]													Bk[2:0]			Column[9:0]										M0

Table 31-14. Sequential Mapping for DDR-SDRAM Configuration: 16K Rows, 1024 Columns, 8 Banks

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bk[2:0]			Row[13:0]														Column[9:0]										M0

Table 31-15. Interleaved Mapping for DDR-SDRAM Configuration: 16K Rows, 1024 Columns, 8 Banks

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Row[13:0]														Bk[2:0]			Column[9:0]										M0

31.6.2 DDR-SDRAM Address Mapping for Low-cost Memories

Table 31-16. Sequential Mapping for DDR-SDRAM Configuration, 2K Rows, 512 Columns, 2 Banks, 16 Bits

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						Bk	Row[10:0]											Column[8:0]									M0

Table 31-17. Interleaved Mapping for DDR-SDRAM Configuration, 2K Rows, 512 Columns, 2 Banks, 16 Bits

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						Row[10:0]											Bk	Column[8:0]									M0

Downloaded from Arrow.com.

31.7 Register Summary

The User Interface is connected to the APB bus. The MPDDRC is programmed using the registers listed in the following table.

Offset	Name	Bit Pos.								
0x00	MPDDRC_MR	31:24								
		23:16								
		15:8								
		7:0						MODE[2:0]		
0x04	MPDDRC_RTR	31:24								
		23:16								
		15:8					COUNT[11:8]			
		7:0	COUNT[7:0]							
0x08	MPDDRC_CR	31:24								
		23:16	UNAL	DECOD	NDQS	NB	LC_LPDDR1			DQMS
		15:8		OCD[2:0]					DIS_DLL	DIC_DS
		7:0	DLL	CAS[2:0]			NR[1:0]		NC[1:0]	
0x0C	MPDDRC_TPR0	31:24	TMRD[3:0]					TWTR[2:0]		
		23:16	TRRD[3:0]				TRP[3:0]			
		15:8	TRC[3:0]				TWR[3:0]			
		7:0	TRCD[3:0]				TRAS[3:0]			
0x10	MPDDRC_TPR1	31:24					TXP[3:0]			
		23:16	TXSRD[7:0]							
		15:8	TXSNR[7:0]							
		7:0		TRFC[6:0]						
0x14	MPDDRC_TPR2	31:24								
		23:16					TFAW[3:0]			
		15:8		TRTP[2:0]			TRPA[3:0]			
		7:0	TXARDS[3:0]				TXARD[3:0]			
0x18 ... 0x1B	Reserved									
0x1C	MPDDRC_LPR	31:24							SELF_DONE	CHG_FRQ
		23:16			UPD_MR[1:0]					APDE
		15:8		SELFAUTO	TIMEOUT[1:0]			DS[2:0]		
		7:0		PASR[2:0]				CLK_FR	LPCB[1:0]	
0x20	MPDDRC_MD	31:24								
		23:16								
		15:8								
		7:0				DBW		MD[2:0]		
0x24 ... 0x33	Reserved									
0x34	MPDDRC_IO_CALIBR	31:24								
		23:16	CALCODEN[3:0]				CALCODEP[3:0]			
		15:8		TZQIO[6:0]						
		7:0						CK_F_RANGE[2:0]		
0x38	MPDDRC_OCMS	31:24								
		23:16								
		15:8								
		7:0				TAMPCLR				SCR_EN
0x3C	MPDDRC_OCMS_KEY1	31:24	KEY1[31:24]							
		23:16	KEY1[23:16]							
		15:8	KEY1[15:8]							
		7:0	KEY1[7:0]							
0x40	MPDDRC_OCMS_KEY2	31:24	KEY2[31:24]							
		23:16	KEY2[23:16]							
		15:8	KEY2[15:8]							
		7:0	KEY2[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x44	MPDDRC_CONF_ARBITER	31:24					BDW_BURST_P3	BDW_BURST_P2	BDW_BURST_P1	BDW_BURST_P0
		23:16					MA_PR_P3	MA_PR_P2	MA_PR_P1	MA_PR_P0
		15:8					RQ_WD_P3	RQ_WD_P2	RQ_WD_P1	RQ_WD_P0
		7:0					BDW_MAX_CUR		ARB[1:0]	
0x48	MPDDRC_TIMEOUT	31:24								
		23:16								
		15:8	TIMEOUT_P3[3:0]				TIMEOUT_P2[3:0]			
		7:0	TIMEOUT_P1[3:0]				TIMEOUT_P0[3:0]			
0x4C	MPDDRC_REQ_PORT_0123	31:24	NRQ_NWD_BDW_P3[7:0]							
		23:16	NRQ_NWD_BDW_P2[7:0]							
		15:8	NRQ_NWD_BDW_P1[7:0]							
		7:0	NRQ_NWD_BDW_P0[7:0]							
0x50 ... 0x53	Reserved									
0x54	MPDDRC_BDW_PORT_0123	31:24		BDW_P3[6:0]						
		23:16		BDW_P2[6:0]						
		15:8		BDW_P1[6:0]						
		7:0		BDW_P0[6:0]						
0x58 ... 0x5B	Reserved									
0x5C	MPDDRC_RD_DATA_PATH	31:24								
		23:16								
		15:8								
		7:0							SHIFT_SAMPLING[1:0]	
0x60	MPDDRC_MCFGR	31:24								
		23:16								
		15:8			INFO[2:0]			REFR_CALIB	READ_WRITE[1:0]	
		7:0				RUN			SOFT_RESET	EN_MONI
0x64	MPDDRC_MADDR0	31:24	ADDR_HIGH_PORT0[15:8]							
		23:16	ADDR_HIGH_PORT0[7:0]							
		15:8	ADDR_LOW_PORT0[15:8]							
		7:0	ADDR_LOW_PORT0[7:0]							
0x68	MPDDRC_MADDR1	31:24	ADDR_HIGH_PORT1[15:8]							
		23:16	ADDR_HIGH_PORT1[7:0]							
		15:8	ADDR_LOW_PORT1[15:8]							
		7:0	ADDR_LOW_PORT1[7:0]							
0x6C	MPDDRC_MADDR2	31:24	ADDR_HIGH_PORT2[15:8]							
		23:16	ADDR_HIGH_PORT2[7:0]							
		15:8	ADDR_LOW_PORT2[15:8]							
		7:0	ADDR_LOW_PORT2[7:0]							
0x70	MPDDRC_MADDR3	31:24	ADDR_HIGH_PORT3[15:8]							
		23:16	ADDR_HIGH_PORT3[7:0]							
		15:8	ADDR_LOW_PORT3[15:8]							
		7:0	ADDR_LOW_PORT3[7:0]							
0x74 ... 0x83	Reserved									
0x84	MPDDRC_MINFO0 (MAX_WAIT)	31:24						LQOS[1:0]		READ_WRITE
		23:16		SIZE[2:0]				BURST[2:0]		
		15:8	MAX_PORT0_WAITING[15:8]							
		7:0	MAX_PORT0_WAITING[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x84	MPDDRC_MINFO0 (NB_TRANSFERS)	31:24	P0_NB_TRANSFERS[31:24]							
		23:16	P0_NB_TRANSFERS[23:16]							
		15:8	P0_NB_TRANSFERS[15:8]							
		7:0	P0_NB_TRANSFERS[7:0]							
0x84	MPDDRC_MINFO0 (TOTAL_LATENCY)	31:24	P0_TOTAL_LATENCY[31:24]							
		23:16	P0_TOTAL_LATENCY[23:16]							
		15:8	P0_TOTAL_LATENCY[15:8]							
		7:0	P0_TOTAL_LATENCY[7:0]							
0x84	MPDDRC_MINFO0 (TOTAL_LATENCY_QOS01)	31:24	P_TOTAL_LATENCY_QOS1[15:8]							
		23:16	P_TOTAL_LATENCY_QOS1[7:0]							
		15:8	P_TOTAL_LATENCY_QOS0[15:8]							
		7:0	P_TOTAL_LATENCY_QOS0[7:0]							
0x84	MPDDRC_MINFO0 (TOTAL_LATENCY_QOS23)	31:24	P0_TOTAL_LATENCY_QOS3[15:8]							
		23:16	P0_TOTAL_LATENCY_QOS3[7:0]							
		15:8	P0_TOTAL_LATENCY_QOS2[15:8]							
		7:0	P0_TOTAL_LATENCY_QOS2[7:0]							
0x88	MPDDRC_MINFO1 (MAX_WAIT)	31:24						LQOS[1:0]		READ_WRITE
		23:16		SIZE[2:0]				BURST[2:0]		
		15:8	MAX_PORT1_WAITING[15:8]							
		7:0	MAX_PORT1_WAITING[7:0]							
0x88	MPDDRC_MINFO1 (NB_TRANSFERS)	31:24	P1_NB_TRANSFERS[31:24]							
		23:16	P1_NB_TRANSFERS[23:16]							
		15:8	P1_NB_TRANSFERS[15:8]							
		7:0	P1_NB_TRANSFERS[7:0]							
0x88	MPDDRC_MINFO1 (TOTAL_LATENCY)	31:24	P1_TOTAL_LATENCY[31:24]							
		23:16	P1_TOTAL_LATENCY[23:16]							
		15:8	P1_TOTAL_LATENCY[15:8]							
		7:0	P1_TOTAL_LATENCY[7:0]							
0x88	MPDDRC_MINFO1 (TOTAL_LATENCY_QOS01)	31:24	P_TOTAL_LATENCY_QOS1[15:8]							
		23:16	P_TOTAL_LATENCY_QOS1[7:0]							
		15:8	P_TOTAL_LATENCY_QOS0[15:8]							
		7:0	P_TOTAL_LATENCY_QOS0[7:0]							
0x88	MPDDRC_MINFO1 (TOTAL_LATENCY_QOS23)	31:24	P1_TOTAL_LATENCY_QOS3[15:8]							
		23:16	P1_TOTAL_LATENCY_QOS3[7:0]							
		15:8	P1_TOTAL_LATENCY_QOS2[15:8]							
		7:0	P1_TOTAL_LATENCY_QOS2[7:0]							
0x8C	MPDDRC_MINFO2 (MAX_WAIT)	31:24						LQOS[1:0]		READ_WRITE
		23:16		SIZE[2:0]				BURST[2:0]		
		15:8	MAX_PORT2_WAITING[15:8]							
		7:0	MAX_PORT2_WAITING[7:0]							
0x8C	MPDDRC_MINFO2 (NB_TRANSFERS)	31:24	P2_NB_TRANSFERS[31:24]							
		23:16	P2_NB_TRANSFERS[23:16]							
		15:8	P2_NB_TRANSFERS[15:8]							
		7:0	P2_NB_TRANSFERS[7:0]							
0x8C	MPDDRC_MINFO2 (TOTAL_LATENCY)	31:24	P2_TOTAL_LATENCY[31:24]							
		23:16	P2_TOTAL_LATENCY[23:16]							
		15:8	P2_TOTAL_LATENCY[15:8]							
		7:0	P2_TOTAL_LATENCY[7:0]							
0x8C	MPDDRC_MINFO2 (TOTAL_LATENCY_QOS01)	31:24	P2_TOTAL_LATENCY_QOS1[15:8]							
		23:16	P2_TOTAL_LATENCY_QOS1[7:0]							
		15:8	P2_TOTAL_LATENCY_QOS0[15:8]							
		7:0	P2_TOTAL_LATENCY_QOS0[7:0]							
0x8C	MPDDRC_MINFO2 (TOTAL_LATENCY_QOS23)	31:24	P_TOTAL_LATENCY_QOS3[15:8]							
		23:16	P_TOTAL_LATENCY_QOS3[7:0]							
		15:8	P_TOTAL_LATENCY_QOS2[15:8]							
		7:0	P_TOTAL_LATENCY_QOS2[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.									
0x90	MPDDRC_MINFO3 (MAX_WAIT)	31:24						LQOS[1:0]		READ_WRITE	
		23:16		SIZE[2:0]				BURST[2:0]			
		15:8	MAX_PORT3_WAITING[15:8]								
		7:0	MAX_PORT3_WAITING[7:0]								
0x90	MPDDRC_MINFO3 (NB_TRANSFERS)	31:24	P3_NB_TRANSFERS[31:24]								
		23:16	P3_NB_TRANSFERS[23:16]								
		15:8	P3_NB_TRANSFERS[15:8]								
		7:0	P3_NB_TRANSFERS[7:0]								
0x90	MPDDRC_MINFO3 (TOTAL_LATENCY)	31:24	P3_TOTAL_LATENCY[31:24]								
		23:16	P3_TOTAL_LATENCY[23:16]								
		15:8	P3_TOTAL_LATENCY[15:8]								
		7:0	P3_TOTAL_LATENCY[7:0]								
0x90	MPDDRC_MINFO3 (TOTAL_LATENCY_QOS01)	31:24	P3_TOTAL_LATENCY_QOS1[15:8]								
		23:16	P3_TOTAL_LATENCY_QOS1[7:0]								
		15:8	P3_TOTAL_LATENCY_QOS0[15:8]								
		7:0	P3_TOTAL_LATENCY_QOS0[7:0]								
0x90	MPDDRC_MINFO3 (TOTAL_LATENCY_QOS23)	31:24	P_TOTAL_LATENCY_QOS3[15:8]								
		23:16	P_TOTAL_LATENCY_QOS3[7:0]								
		15:8	P_TOTAL_LATENCY_QOS2[15:8]								
		7:0	P_TOTAL_LATENCY_QOS2[7:0]								
0x94 ... 0xBF	Reserved										
0xC0	MPDDRC_IER	31:24									
		23:16									
		15:8									
		7:0								RD_ERR	SEC
0xC4	MPDDRC_IDR	31:24									
		23:16									
		15:8									
		7:0								RD_ERR	SEC
0xC8	MPDDRC_IMR	31:24									
		23:16									
		15:8									
		7:0								RD_ERR	SEC
0xCC	MPDDRC_ISR	31:24									
		23:16									
		15:8									
		7:0								RD_ERR	SEC
0xD0	MPDDRC_SAFETY	31:24				EN	ADDRESS[27:24]				
		23:16	ADDRESS[23:16]								
		15:8	ADDRESS[15:8]								
		7:0	ADDRESS[7:0]								
0xD4 ... 0xE3	Reserved										
0xE4	MPDDRC_WPMR	31:24	WPKEY[23:16]								
		23:16	WPKEY[15:8]								
		15:8	WPKEY[7:0]								
		7:0				FIRSTE				WPITEN	WPEN
0xE8	MPDDRC_WPSR	31:24	ECLASS							SWETYP[1:0]	
		23:16	WPVSRC[15:8]								
		15:8	WPVSRC[7:0]								
		7:0					SWE	SEQE	CGD	WPVS	

Downloaded from Arrow.com.

31.7.1 MPDDRC Mode Register

Name: MPDDRC_MR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the MPDDRC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						MODE[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – MODE[2:0] MPDDRC Command Mode
This field defines the command issued by the MPDDRC when the SDRAM device is accessed. This register is used to initialize the SDRAM device and to activate Deep Powerdown mode.

Value	Name	Description
0	NORMAL_CMD	Normal Mode. Any access to the MPDDRC is decoded normally. To activate this mode, the command must be followed by a write to the DDR-SDRAM.
1	NOP_CMD	The MPDDRC issues a NOP command when the DDR-SDRAM device is accessed regardless of the cycle. To activate this mode, the command must be followed by a write to the DDR-SDRAM.
2	PRCGALL_CMD	The MPDDRC issues the All Banks Precharge command when the DDR-SDRAM device is accessed regardless of the cycle. To activate this mode, the command must be followed by a write to the SDRAM.
3	LMR_CMD	The MPDDRC issues a Load Mode Register command when the DDR-SDRAM device is accessed regardless of the cycle. To activate this mode, the command must be followed by a write to the DDR-SDRAM.
4	RFSH_CMD	The MPDDRC issues an Autorefresh command when the DDR-SDRAM device is accessed regardless of the cycle. Previously, an All Banks Precharge command must be issued. To activate this mode, the command must be followed by a write to the DDR-SDRAM.
5	EXT_LMR_CMD	The MPDDRC issues an Extended Load Mode Register command when the SDRAM device is accessed regardless of the cycle. To activate this mode, the command must be followed by a write to the DDR-SDRAM. The write in the DDR-SDRAM must be done in the appropriate bank.
6	DEEP_MD	Deep Power mode: Access to Deep Powerdown mode

Downloaded from Arrow.com.

31.7.2 MPDDRC Refresh Timer Register

Name: MPDDRC_RTR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the MPDDRC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					COUNT[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – COUNT[11:0] MPDDRC Refresh Timer Count
This 12-bit field is loaded into a timer which generates the refresh pulse. Each time the refresh pulse is generated, a refresh sequence is initiated.
The SDRAM requires autorefresh cycles at an average periodic interval of T_{refi}. The value to be loaded depends on the MPDDRC clock frequency MCK (Master Clock) and average periodic interval of T_{refi}.
For example, for an SDRAM with T_{refi} = 7.8 µs and a 133 MHz (7.5 ns) Master clock, the value of the COUNT field is configured: $((7.8 \times 10^{-6}) / (7.5 \times 10^{-9})) = 1040$ or 0x0410.

Downloaded from Arrow.com.

31.7.3 MPDDRC Configuration Register

Name: MPDDRC_CR
Offset: 0x08
Reset: 0x00207024
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the MPDDRC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	UNAL	DECOD	NDQS	NB	LC_LPDDR1			DQMS
Access	R/W	R/W	R/W	R/W	R/W			R/W
Reset	0	0	1	0	0			0

Bit	15	14	13	12	11	10	9	8
		OCD[2:0]					DIS_DLL	DIC_DS
Access		R/W	R/W	R/W			R/W	R/W
Reset		1	1	1			0	0

Bit	7	6	5	4	3	2	1	0
	DLL	CAS[2:0]			NR[1:0]		NC[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	1	0	0

Bit 23 – UNAL This bit must always be written to 1.

Bit 22 – DECOD Type of Decoding

Value	Name	Description
0	SEQUENTIAL	Method for address mapping where banks alternate at each last DDR-SDRAM page of the current bank.
1	INTERLEAVED	Method for address mapping where banks alternate at each DDR-SDRAM end of page of the current bank.

Bit 21 – NDQS Not DQS.
This bit is found in DDR2-SDRAM devices, in Extended Mode register 1. DQS may be used in Single-ended mode or paired with optional complementary signal NDQS.
To comply with the LPDDR1 standard, DQS must be used in Single-ended mode. NDQS must be disabled.

Value	Name	Description
0	ENABLED	'Not DQS' is enabled.
1	DISABLED	'Not DQS' is disabled.

Bit 20 – NB Number of Banks
LC_LPDDR1 is set to 1, NB is not relevant.

Value	Name	Description
0	4_BANKS	4-bank memory devices
1	8_BANKS	8 banks. Only possible when using DDR2-SDRAM devices.

Bit 19 – LC_LPDDR1 Low-cost Low-power DDR1

Value	Name	Description
0	NOT_2_BANKS	Any type of memory devices except of low cost, low density Low Power DDR1.

Downloaded from Arrow.com.

Value	Name	Description
1	2_BANKS_LPDDR1	Low-cost and low-density low-power DDR1. These devices have a density of 32 Mbits and are organized as two internal banks. To use this feature, the user has to define the type of memory and the data bus width (see 31.7.8 MPDDRC_MD). The 16-bit memory device is organized as 2 banks, 9 columns and 11 rows.

Bit 16 – DQMS Mask Data is Shared

Value	Name	Description
0	NOT_SHARED	DQM is not shared with another controller
1	SHARED	DQM is shared with another controller

Bits 14:12 – OCD[2:0] Off-chip Driver
SDRAM Controller supports only two values for OCD (default calibration and exit from calibration). These values MUST always be programmed during the initialization sequence. The default calibration must be programmed first, after which the exit calibration and maintain settings must be programmed.
This field is found only in the DDR2-SDRAM devices.

Value	Name	Description
0	DDR2_EXITCALIB	Exit from OCD Calibration mode and maintain settings
7	DDR2_DEFAULT_CALIB	OCD calibration default

Bit 9 – DIS_DLL Disable DLL
This value is used during the powerup sequence. It is only found in the DDR2-SDRAM devices.

Value	Description
0	Enable DLL.
1	Disable DLL.

Bit 8 – DIC_DS Output Driver Impedance Control (Drive Strength)
This bit name is described as “DS” in some memory datasheets. It defines the output drive strength. This value is used during the powerup sequence.
This bit is found only in the DDR2-SDRAM devices.

Value	Name	Description
0	DDR2_NORMALSTRENGTH	Normal drive strength (DDR2)
1	DDR2_WEAKSTRENGTH	Weak drive strength (DDR2)

Bit 7 – DLL Reset DLL
This bit defines the value of Reset DLL. It is found only in DDR2-SDRAM devices.
This value is used during the powerup sequence.

Value	Name	Description
0	RESET_DISABLED	Disable DLL reset
1	RESET_ENABLED	Enable DLL reset

Bits 6:4 – CAS[2:0] CAS Latency

Value	Name	Description
2	DDR_CAS2	LPDDR1 CAS Latency 2
3	DDR_CAS3	DDR2/LPDDR1 CAS Latency 3

Bits 3:2 – NR[1:0] Number of Row Bits

Value	Name	Description
0	11_ROW_BITS	11 bits to define the row number, up to 2048 rows
1	12_ROW_BITS	12 bits to define the row number, up to 4096 rows
2	13_ROW_BITS	13 bits to define the row number, up to 8192 rows
3	14_ROW_BITS	14 bits to define the row number, up to 16384 rows

Bits 1:0 – NC[1:0] Number of Column Bits

Downloaded from Arrow.com.

Value	Name	Description
0	DDR9_MDDR8_COL_BITS	9 bits to define the column number, up to 512 columns, for DDR2-SDRAM 8 bits to define the column number, up to 256 columns, for LPDDR1-SDRAM
1	DDR10_MDDR9_COL_BITS	10 bits to define the column number, up to 512 columns, for DDR2-SDRAM 9 bits to define the column number, up to 256 columns, for LPDDR1-SDRAM
2	DDR11_MDDR10_COL_BITS	11 bits to define the column number, up to 512 columns, for DDR2-SDRAM 10 bits to define the column number, up to 256 columns, for LPDDR1-SDRAM
3	DDR12_MDDR11_COL_BITS	12 bits to define the column number, up to 512 columns, for DDR2-SDRAM 11 bits to define the column number, up to 256 columns, for LPDDR1-SDRAM

Downloaded from Arrow.com.

31.7.4 MPDDRC Timing Parameter 0 Register

Name: MPDDRC_TPR0
Offset: 0x0C
Reset: 0x20227225
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the MPDDRC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	TMRD[3:0]					TWTR[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	1	0		0	0	0

Bit	23	22	21	20	19	18	17	16
	TRRD[3:0]				TRP[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8
	TRC[3:0]				TWR[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	1	0	0	1	0

Bit	7	6	5	4	3	2	1	0
	TRCD[3:0]				TRAS[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	1	0	1

Bits 31:28 – TMRD[3:0] Load Mode Register Command to Activate or Refresh Command
This field defines the delay between a Load mode register command and an Activate or Refresh command in number of DDRCK clock cycles. The number of cycles is between 0 and 15.

Bits 26:24 – TWTR[2:0] Internal Write to Read Delay
This field defines the internal Write to Read command time in number of DDRCK clock cycles. The number of cycles is between 1 and 7.

Bits 23:20 – TRRD[3:0] Active BankA to Active BankB
This field defines the delay between an Activate command in BankA and an Activate command in BankB in number of DDRCK clock cycles. The number of cycles is between 1 and 15.

Bits 19:16 – TRP[3:0] Row Precharge Delay
This field defines the delay between a Precharge command and another command in number of DDRCK clock cycles. The number of cycles is between 0 and 15.

Bits 15:12 – TRC[3:0] Row Cycle Delay
This field defines the delay between an Activate command and a Refresh command in number of DDRCK clock cycles. The number of cycles is between 0 and 15.

Bits 11:8 – TWR[3:0] Write Recovery Delay
This field defines the Write Recovery Time in number of DDRCK clock cycles. The number of cycles is between 1 and 15.

Bits 7:4 – TRCD[3:0] Row to Column Delay
This field defines the delay between an Activate command and a Read/Write command in number of DDRCK clock cycles. The number of cycles is between 0 and 15.

Downloaded from Arrow.com.

Bits 3:0 – TRAS[3:0] Active to Precharge Delay
This field defines the delay between an Activate command and a Precharge command in number of DDRCK clock cycles. The number of cycles is between 0 and 15.

Downloaded from Arrow.com.

31.7.5 MPDDRC Timing Parameter 1 Register

Name: MPDDRC_TPR1
Offset: 0x10
Reset: 0x03C80808
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the MPDDRC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
					TXP[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	1	1

Bit	23	22	21	20	19	18	17	16
	TXSRD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8
	TXSNR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	0	0

Bit	7	6	5	4	3	2	1	0
		TRFC[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	1	0	0	0

Bits 27:24 – TXP[3:0] Exit Powerdown Delay to First Command
This field defines the delay between CKE set high and a valid command in number of DDRCK clock cycles. The number of cycles is between 0 and 15.

Bits 23:16 – TXSRD[7:0] Exit Self-refresh Delay to Read Command
This field defines the delay between CKE set high and a Read command in number of DDRCK clock cycles. The number of cycles is between 0 and 255.
This field is found only in DDR2-SDRAM devices.

Bits 15:8 – TXSNR[7:0] Exit Self-refresh Delay to Non-Read Command
This field defines the delay between CKE set high and a Non Read command in number of DDRCK clock cycles. The number of cycles is between 0 and 255. This field is used by the DDR-SDRAM devices. In case of low-power DDR-SDRAM, this field is equivalent to t_{XSR}.

Bits 6:0 – TRFC[6:0] Row Refresh Cycle
This field defines the delay between a Refresh command or a Refresh and Activate command in number of DDRCK clock cycles. The number of cycles is between 0 and 127.

Downloaded from Arrow.com.

31.7.6 MPDDRC Timing Parameter 2 Register

Name: MPDDRC_TPR2
Offset: 0x14
Reset: 0x00042062
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					TFAW[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	1	0	0

Bit	15	14	13	12	11	10	9	8
		TRTP[2:0]			TRPA[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	1	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXARDS[3:0]				TXARD[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	0	0	0	1	0

Bits 19:16 – TFAW[3:0] Four Active Windows
DDR2 devices with eight banks (1 Gbit or larger) have an additional requirement concerning t_{FAW} timing. This requires that no more than four Activate commands may be issued in any given t_{FAW} (MIN) period.
The number of cycles is between 0 and 15.
This field is found only in DDR2-SDRAM.

Bits 14:12 – TRTP[2:0] Read to Precharge
This field defines the delay between a Read command and a Precharge command in number of DDRCK clock cycles.
The number of cycles is between 0 and 7.

Bits 11:8 – TRPA[3:0] Row Precharge All Delay
This field defines the delay between a Precharge All Banks command and another command in number of DDRCK clock cycles. The number of cycles is between 0 and 15.
This field is found only in the DDR2-SDRAM devices.

Bits 7:4 – TXARDS[3:0] Exit Active Power Down Delay to Read Command in Mode “Slow Exit”
This field defines the delay between CKE set high and a Read command in number of DDRCK clock cycles. The number of cycles is between 0 and 15.
This field is found only in the DDR2-SDRAM devices.

Bits 3:0 – TXARD[3:0] Exit Active Power Down Delay to Read Command in Mode “Fast Exit”
This field defines the delay between CKE set high and a Read command in number of DDRCK clock cycles. The number of cycles is between 0 and 15.
This field is found only in the DDR2-SDRAM devices.

Downloaded from Arrow.com.

31.7.7 MPDDRC Low-Power Register

Name: MPDDRC_LPR
Offset: 0x1C
Reset: 0x00010000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							SELF_DONE	CHG_FRQ
Access							R	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
			UPD_MR[1:0]					APDE
Access			R/W	R/W				R/W
Reset			0	0				1

Bit	15	14	13	12	11	10	9	8
		SELFAUTO	TIMEOUT[1:0]			DS[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
		PASR[2:0]				CLK_FR	LPCB[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit 25 – SELF_DONE Self-refresh is Done
This bit indicates that external device is in Self-refresh mode.

Bit 24 – CHG_FRQ Change Clock Frequency During Self-refresh Mode
This mode allows to change the low-power DDR-DRAM input clock frequency. This mode is unique to the low-power DDR-DRAM devices.

Bits 21:20 – UPD_MR[1:0] Update Load Mode Register and Extended Mode Register
This bit is used to enable or disable automatic update of the Load Mode Register and Extended Mode Register. This update depends on the MPDDRC integration in a system. MPDDRC can either share or not an external bus with another controller.

Value	Name	Description
0	NO_UPDATE	Update of Load Mode and Extended Mode registers is disabled.
1	UPDATE_SHAREDBUS	MPDDRC shares an external bus. Automatic update is done during a refresh command and a pending read or write access in the SDRAM device.
2	UPDATE_NOSHAREDBUS	MPDDRC does not share an external bus. Automatic update is done before entering Self-refresh mode.
3	–	Reserved

Bit 16 – APDE Active Power Down Exit Time
This mode is unique to the DDR2-SDRAM devices.
This mode manages the active Powerdown mode which determines performance versus power saving.
After the initialization sequence, as soon as the APDE field is modified, the Extended Mode Register (located in the memory of the external device) is accessed automatically and APDE bits are updated. Depending on the UPD_MR bit, update is done before entering Self-refresh mode or during a refresh command and a pending read or write access

Value	Name	Description
0	DDR2_FAST_EXIT	Fast Exit from Power Down. DDR2-SDRAM devices only.

Downloaded from Arrow.com.

Value	Name	Description
1	DDR2_SLOW_EXIT	Slow Exit from Power Down. DDR2-SDRAM devices only.

Bit 14 – SELFAUTO Self-refresh Exit Autorefresh

Value	Description
1	Upon exiting Self-refresh mode, autorefresh command is immediately performed after tXSNR.
0	Upon exiting Self-refresh mode, active command is immediately performed after tXSNR. The autorefresh command is issued every 15.6 µs or less.

Bits 13:12 – TIMEOUT[1:0] Time Between Last Transfer and Low-Power Mode
This field defines when Low-power mode is activated.

Value	Name	Description
0	NONE	SDRAM Low-power mode is activated immediately after the end of the last transfer.
1	DELAY_64_CLK	SDRAM Low-power mode is activated 64 clock cycles after the end of the last transfer.
2	DELAY_128_CLK	SDRAM Low-power mode is activated 128 clock cycles after the end of the last transfer.
3	–	Reserved

Bits 10:8 – DS[2:0] Drive Strength
This field is unique to low-power DDR1-SDRAM. It selects the output drive strength.
After the initialization sequence, as soon as the DS field is modified, the Extended Mode Register is accessed automatically and DS bits are updated. Depending on the UPD_MR bit, update is done before entering self-refresh mode or during a refresh command and a pending read or write access.

Value	Name	Description
0	DS_FULL	Full drive strength
1	DS_HALF	Half drive strength
2	DS_QUARTER	Quarter drive strength
3	DS_OCTANT	Octant drive strength
4–7	–	Reserved

Bits 6:4 – PASR[2:0] Partial Array Self-refresh
This field is unique to low-power DDR1-SDRAM. It is used to specify whether only one-quarter, one-half or all banks of the DDR-SDRAM array are enabled. Disabled banks are not refreshed in Self-refresh mode.
The values of this field are dependent on the low-power DDR-SDRAM devices.
After the initialization sequence, as soon as the PASR field is modified, the Extended Mode Register in the external device memory is accessed automatically and PASR bits are updated. Depending on the UPD_MR bit, update is done before entering Self-refresh mode or during a refresh command and a pending read or write access.

Bit 2 – CLK_FR Clock Frozen Command Bit
This field sets the clock low during Powerdown mode. Some DDR-SDRAM devices do not support freezing the clock during Powerdown mode. Refer to the relevant DDR-SDRAM device datasheet for details.

Value	Name	Description
0	DISABLED	Clock(s) is/are not frozen.
1	ENABLED	Clock(s) is/are frozen.

Bits 1:0 – LPCB[1:0] Low-power Command Bit

Value	Name	Description
0	NOLOWPOWER	Low-power feature is inhibited. No Powerdown, Self-refresh and Deep power modes are issued to the DDR-SDRAM device.
1	SELFREFRESH	The MPDDRC issues a self-refresh command to the DDR-SDRAM device, the clock(s) is/are deactivated and the CKE signal is set low. The DDR-SDRAM device leaves the Self-refresh mode when accessed and reenters it after the access.
2	POWERDOWN	The MPDDRC issues a Powerdown command to the DDR-SDRAM device after each access, the CKE signal is set low. The DDR-SDRAM device leaves the Powerdown mode when accessed and reenters it after the access.

Downloaded from Arrow.com.

Value	Name	Description
3	DEEPPOWERDOWN	The MPDDRC issues a Deep Powerdown command to the low-power DDR-SDRAM device.

Downloaded from Arrow.com.

31.7.8 MPDDRC Memory Device Register

Name: MPDDRC_MD
Offset: 0x20
Reset: 0x00000013
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the MPDDRC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				DBW		MD[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				1		0	1	1

Bit 4 – DBW Data Bus Width

Value	Name	Description
0	RESERVED	Reserved
1	DBW_16_BITS	Data bus width is 16 bits.

Bits 2:0 – MD[2:0] Memory Device

Value	Name	Description
3	LPDDR_SDRAM	Low-power DDR1-SDRAM
6	DDR2_SDRAM	DDR2-SDRAM

Downloaded from Arrow.com.

31.7.9 MPDDRC I/O Calibration Register

Name: MPDDRC_IO_CALIBR
Offset: 0x34
Reset: 0x00870000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CALCODEN[3:0]				CALCODEP[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	1	0	0	0	0	1	1	1

Bit	15	14	13	12	11	10	9	8
		TZQIO[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						CK_F_RANGE[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 23:20 – CALCODEN[3:0] Number of N-type Transistors
This value gives the number of N-type transistors to perform the calibration.

Bits 19:16 – CALCODEP[3:0] Number of P-type Transistors
This value gives the number of P-type transistors to perform the calibration.

Bits 14:8 – TZQIO[6:0] IO Calibration
This field defines the delay between the start up of the amplifier and the beginning of the calibration in number of DDRCK clock cycles. The value of this field must be set to 600 ns.
The number of cycles is between 0 and 127.
The TZQIO configuration code must be set correctly depending on the clock frequency using the following formula:
TZQIO = (DDRCK × 600e-9) + 1
where DDRCK frequency is in Hz.
For example, for a frequency of 176 MHz, the value of the TZQIO field is configured (176 × 10e6) × (600e-9) + 1.

Bits 2:0 – CK_F_RANGE[2:0] DDRCK Maximum Clock Frequency Range Indicator
This field must always be written to 7.

Downloaded from Arrow.com.

31.7.10 MPDDRC OCMS Register

Name: MPDDRC_OCMS
Offset: 0x38
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the MPDDRC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				TAMPCLR				SCR_EN
Access				R/W				R/W
Reset				0				0

Bit 4 – TAMPCLR Tamper Clear Enable

Value	Description
0	A tamper detection event has no effect on MPDDRC scrambling keys.
1	A tamper detection event immediately clears MPDDRC scrambling keys.

Bit 0 – SCR_EN Scrambling Enable

Value	Description
0	Disables “Off-chip” scrambling for SDRAM access.
1	Enables “Off-chip” scrambling for SDRAM access.

Downloaded from Arrow.com.

31.7.11 MPDDRC OCMS KEY1 Register

Name: MPDDRC_OCMS_KEY1
Offset: 0x3C
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the MPDDRC Write Protection Mode Register. The register can only be written once.

Bit	31	30	29	28	27	26	25	24
	KEY1[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	KEY1[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	KEY1[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	KEY1[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – KEY1[31:0] Off-chip Memory Scrambling (OCMS) Key Part 1
When Off-chip Memory Scrambling is enabled, the data scrambling depends on KEY1 and KEY2 values.

Downloaded from Arrow.com.

31.7.12 MPDDRC OCMS KEY2 Register

Name: MPDDRC_OCMS_KEY2
Offset: 0x40
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the MPDDRC Write Protection Mode Register. The register can only be written once.

Bit	31	30	29	28	27	26	25	24
	KEY2[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	KEY2[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	KEY2[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	KEY2[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – KEY2[31:0] Off-chip Memory Scrambling (OCMS) Key Part 2
When Off-chip Memory Scrambling is enabled, the data scrambling depends on KEY1 and KEY2 values.

Downloaded from Arrow.com.

31.7.13 MPDDRC Configuration Arbiter Register

Name: MPDDRC_CONF_ARBITER
Offset: 0x44
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					BDW_BURST_P3	BDW_BURST_P2	BDW_BURST_P1	BDW_BURST_P0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					MA_PR_P3	MA_PR_P2	MA_PR_P1	MA_PR_P0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					RQ_WD_P3	RQ_WD_P2	RQ_WD_P1	RQ_WD_P0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
					BDW_MAX_CUR		ARB[1:0]	
Access					R/W		R/W	R/W
Reset					0		0	0

Bits 24, 25, 26, 27 – BDW_BURST_Px Bandwidth Arbitration Mode on Port X

Value	Description
0	The arbitration is done when the bandwidth limit defined in MPDDRC_BDW_PORT_0123.BDW_Px is reached. If the bandwidth limit is reached during a burst access, the burst is completed.
1	The arbitration is done when the bandwidth limit defined in MPDDRC_BDW_PORT_0123.BDW_Px is reached. If the bandwidth limit is reached during a burst access, the burst is broken.

Bits 16, 17, 18, 19 – MA_PR_Px Master or Software Provide Information

Value	Description
0	Number of requests or words is provided by the master, if the master supports this feature.
1	Number of requests or words is provided by software, see "NRQ_NWD_BDW_Px: Number of Requests, Number of Words or Bandwidth Allocation from Port 0-1-2-3".

Bits 8, 9, 10, 11 – RQ_WD_Px Request or Word from Port X

Value	Description
0	Number of requests is selected.
1	Number of words is selected.

Bit 3 – BDW_MAX_CUR Bandwidth Max or Current
This field displays the maximum of the bandwidth or the current bandwidth for each port.
The maximum of the bandwidth is computed when at least two ports of MPDDRC are used.
That information is given in MPDDRC Current/Maximum Bandwidth Port 0-1-2-3 Register.

Value	Description
0	Current bandwidth is displayed.
1	Maximum of the bandwidth is displayed.

Downloaded from Arrow.com.

Bits 1:0 – ARB[1:0] Type of Arbitration
This field allows to choose the type of arbitration: round-robin, number of requests per port or bandwidth per port.

Value	Name	Description
0	ROUND	Round-Robin Policy
1	NB_REQUEST	Request Policy
2	BANDWIDTH	Bandwidth Policy
3	LQOS	Quality of Service Policy

Downloaded from Arrow.com.

31.7.14 MPDDRC Timeout Register

Name: MPDDRC_TIMEOUT
Offset: 0x48
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TIMEOUT_P3[3:0]				TIMEOUT_P2[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TIMEOUT_P1[3:0]				TIMEOUT_P0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0:3, 4:7, 8:11, 12:15 – TIMEOUT_Px Timeout for Ports 0, 1, 2, 3
Some masters have the particularity to insert idle state between two accesses. This field defines the delay between two accesses on the same port in number of DDRCK clock cycles before arbitration and handling the access over to another port.
This field is not used with round-robin and bandwidth arbitrations.
The number of cycles is between 1 and 15.

Downloaded from Arrow.com.

31.7.15 MPDDRC Request Port 0-1-2-3 Register

Name: MPDDRC_REQ_PORT_0123
Offset: 0x4C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	NRQ_NWD_BDW_P3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NRQ_NWD_BDW_P2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NRQ_NWD_BDW_P1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NRQ_NWD_BDW_P0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0:7, 8:15, 16:23, 24:31 – NRQ_NWD_BDW_Px Number of Requests, Number of Words or Bandwidth Allocation from Port 0-1-2-3
The number of requests corresponds to the number of start transfers. For example, setting this field to 2 performs two burst accesses regardless of the burst type (INCR4, INCR8, etc.). The number of words corresponds exactly to the number of accesses; setting this field to 2 performs two accesses. In this example, burst accesses will be broken. These values depend on scheme arbitration (see MPDDRC Configuration Arbiter Register).
In case of round-robin arbitration, this field is not used. In case of “bandwidth arbitration”, this field corresponds to percentage allocated for each port. In case of “request” arbitration, this field corresponds to number of start transfers or to number of accesses allocated for each port.

Downloaded from Arrow.com.

31.7.16 MPDDRC Current/Maximum Bandwidth Port 0-1-2-3 Register

Name: MPDDRC_BDW_PORT_0123
Offset: 0x54
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
		BDW_P3[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		BDW_P2[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		BDW_P1[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		BDW_P0[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 0:6, 8:14, 16:22, 24:30 – BDW_Px Current/Maximum Bandwidth from Port 0-1-2-3
This field displays the current bandwidth or the maximum bandwidth for each port. This information is given in the "BDW_MAX_CUR: Bandwidth Max or Current" field description.

Downloaded from Arrow.com.

31.7.17 MPDDRC Read Data Path Register

Name: MPDDRC_RD_DATA_PATH
Offset: 0x5C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							SHIFT_SAMPLING[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – SHIFT_SAMPLING[1:0] Shift Sampling Point of Data
Shifts the sampling point of data coming from the memory device. The higher the memory device clock frequency, the higher the SHIFT_SAMPLING value. Refer to the section "Electrical Characteristics".

Value	Name	Description
0	NO_SHIFT	Initial sampling point.
1	SHIFT_ONE_CYCLE	Sampling point is shifted by one cycle.
2	SHIFT_TWO_CYCLES	Sampling point is shifted by two cycles.
3	SHIFT_THREE_CYCLES	Not applicable for DDR2 and LPDDR1 devices.

Downloaded from Arrow.com.

31.7.18 MPDDRC Monitor Configuration Register

Name: MPDDRC_MCFGR
Offset: 0x60
Reset: 0x00000011
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			INFO[2:0]			REFR_CALIB	READ_WRITE[1:0]	
Access							R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				RUN			SOFT_RESET	EN_MONI
Access				R/W			R/W	R/W
Reset				1			0	1

Bits 13:11 – INFO[2:0] Information Type
This field reports information such as latency and the number of transfers monitored on port x [x = 0..3].

Value	Name	Description
0	MAX_WAIT	Information concerning the transfer with the longest waiting time
1	NB_TRANSFERS	Number of transfers on the port
2	TOTAL_LATENCY	Total latency on the port
3	–	Reserved
4	MAX_WAIT_QOS01	Information concerning the transfer with the longest waiting time, depending on QOS values (0 and 1)
5	MAX_WAIT_QOS23	Information concerning the transfer with the longest waiting time, depending on QOS values (2 and 3)

Bit 10 – REFR_CALIB Refresh Calibration

Value	Description
0	Monitoring does not depend on Autorefresh mode, Self-refresh mode, Powerdown mode, DLL nor calibration impact.
1	Monitoring depends on Autorefresh mode, Self-refresh mode, Powerdown mode, DLL and calibration impact.

Bits 9:8 – READ_WRITE[1:0] Read/Write Access
This field is used to monitor different types of access.

Value	Name	Description
0	TRIG_RD_WR	Read and Write accesses are triggered.
1	TRIG_WR	Only Write accesses are triggered.
2	TRIG_RD	Only Read accesses are triggered.
3	–	Reserved

Bit 4 – RUN Control Monitor

Downloaded from Arrow.com.

Value	Description
0	Monitoring is halted. All counters are stopped.
1	Monitoring is launched.

Bit 1 – SOFT_RESET Soft Reset

Value	Description
0	Soft reset is not performed.
1	Soft reset is performed.

Bit 0 – EN_MONI Enable Monitor

Value	Description
0	Monitor is disabled.
1	Monitor is enabled.

Downloaded from Arrow.com.

31.7.19 MPDDRC Monitor Address High/Low Port x Register

Name: MPDDRC_MADDRx
Offset: 0x64 + x*0x04 [x=0..3]
Reset: 0xFFFF0000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ADDR_HIGH_PORTx[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	ADDR_HIGH_PORTx[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	ADDR_LOW_PORTx[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR_LOW_PORTx[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – ADDR_HIGH_PORTx[15:0] Address High on Port x
Address high which defines the interval to be monitored on port x [x = 0..3]. This address must be programmed according to the memory mapping of the product.

Bits 15:0 – ADDR_LOW_PORTx[15:0] Address Low on Port x
Address low which defines the interval to be monitored on port x [x = 0..3]. This address must be programmed according to the memory mapping of the product.

Downloaded from Arrow.com.

31.7.20 MPDDRC Monitor Information Port x Register (MAX_WAIT)

Name: MPDDRC_MINFOx (MAX_WAIT)
Offset: 0x84 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read-only

The following fields can be read if the INFO field in the MPDDRC Monitor Configuration register is set to 0.

Bit	31	30	29	28	27	26	25	24
						LQOS[1:0]		READ_WRITE
Access						R	R	R
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
		SIZE[2:0]				BURST[2:0]		
Access		R	R	R		R	R	R
Reset		0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
	MAX_PORTx_WAITING[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MAX_PORTx_WAITING[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 26:25 – LQOS[1:0] Value of Quality Of Service on Port x
This field reports the value of quality of service for the maximum waiting time.

Value	Name	Description
0	BACKGROUND	Background transfers
1	BANDWIDTH	Bandwidth sensitive
2	SENSITIVE_LAT	Latency sensitive
3	CRITICAL_LAT	Latency critical

Bit 24 – READ_WRITE Read or Write Access on Port x
This field reports the transfer direction for the maximum waiting time.

Value	Description
0	Read transfer
1	Write transfer

Bits 22:20 – SIZE[2:0] Transfer Size on Port x
This field reports the size of the transfer for the maximum waiting time.

Value	Name	Description
0	8BITS	Byte transfer
1	16BITS	Halfword transfer
2	32BITS	Word transfer
3	64BITS	Dword transfer

Bits 18:16 – BURST[2:0] Type of Burst on Port x
This field reports the type of burst for the maximum waiting time.

Value	Name	Description
0	SINGLE	Single transfer
1	INCR	Incrementing burst of unspecified length

Downloaded from Arrow.com.

Value	Name	Description
2	WRAP4	4-beat wrapping burst
3	INCR4	4-beat incrementing burst
4	WRAP8	8-beat wrapping burst
5	INCR8	8-beat incrementing burst
6	WRAP16	16-beat wrapping burst
7	INCR16	16-beat incrementing burst

Bits 15:0 – MAX_PORTx_WAITING[15:0] Address High on Port x
This field reports the maximum waiting time and the associated type of transfer (burst, size, read or write).

Downloaded from Arrow.com.

31.7.21 MPDDRC Monitor Information Port x Register (NB_TRANSFERS)

Name: MPDDRC_MINFOx (NB_TRANSFERS)
Offset: 0x84 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read-only

The following fields can be read if the INFO field in the MPDDRC Monitor Configuration register is set to 1.

Bit	31	30	29	28	27	26	25	24
	Px_NB_TRANSFERS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	Px_NB_TRANSFERS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	Px_NB_TRANSFERS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	Px_NB_TRANSFERS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – Px_NB_TRANSFERS[31:0] Number of Transfers on Port x
This field can be read if the INFO field is set to 1. This field reports the number of transfers performed within an interval (ADDR_HIGH_PORT and ADDR_LOW_PORT) when the port is used.

Downloaded from Arrow.com.

31.7.22 MPDDRC Monitor Information Port x Register (TOTAL_LATENCY)

Name: MPDDRC_MINFOx (TOTAL_LATENCY)
Offset: 0x84 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read-only

The following fields can be read if the INFO field in the MPDDRC Monitor Configuration register is set to 2.

Bit	31	30	29	28	27	26	25	24
	Px_TOTAL_LATENCY[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	Px_TOTAL_LATENCY[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	Px_TOTAL_LATENCY[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	Px_TOTAL_LATENCY[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – Px_TOTAL_LATENCY[31:0] Total Latency on Port x
This field can be read if the INFO field is set to 2. This field reports the total latency within an interval (ADDR_HIGH_PORT and ADDR_LOW_PORT) when the port is used.

Downloaded from Arrow.com.

31.7.23 MPDDRC Monitor Information Port x Register (TOTAL_LATENCY_QOS01)

Name: MPDDRC_MINFOx (TOTAL_LATENCY_QOS01)
Offset: 0x84 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read-only

The following fields can be read if the INFO field in the MPDDRC Monitor Configuration register is set to 4.

Bit	31	30	29	28	27	26	25	24
	Px_TOTAL_LATENCY_QOS1[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	Px_TOTAL_LATENCY_QOS1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	Px_TOTAL_LATENCY_QOS0[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	Px_TOTAL_LATENCY_QOS0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – Px_TOTAL_LATENCY_QOS1[15:0] Total Latency on Port x when value of qos is 1
This field can be read if the INFO field is set to 4. This field reports the total latency within an interval (ADDR_HIGH_PORT and ADDR_LOW_PORT) when the port is used with qos = 1.

Bits 15:0 – Px_TOTAL_LATENCY_QOS0[15:0] Total Latency on Port x when value of qos is 0
This field can be read if the INFO field is set to 4. This field reports the total latency within an interval (ADDR_HIGH_PORT and ADDR_LOW_PORT) when the port is used with qos = 0.

Downloaded from Arrow.com.

31.7.24 MPDDRC Monitor Information Port x Register (TOTAL_LATENCY_QOS23)

Name: MPDDRC_MINFOx (TOTAL_LATENCY_QOS23)
Offset: 0x84 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read-only

The following fields can be read if the INFO field in the MPDDRC Monitor Configuration register is set to 5.

Bit	31	30	29	28	27	26	25	24
	Px_TOTAL_LATENCY_QOS3[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	Px_TOTAL_LATENCY_QOS3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	Px_TOTAL_LATENCY_QOS2[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	Px_TOTAL_LATENCY_QOS2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – Px_TOTAL_LATENCY_QOS3[15:0] Total Latency on Port x when value of qos is 3
This field can be read if the INFO field is set to 5. This field reports the total latency within an interval (ADDR_HIGH_PORT and ADDR_LOW_PORT) when the port is used with qos = 3.

Bits 15:0 – Px_TOTAL_LATENCY_QOS2[15:0] Total Latency on Port x when value of qos is 2
This field can be read if the INFO field is set to 5. This field reports the total latency within an interval (ADDR_HIGH_PORT and ADDR_LOW_PORT) when the port is used with qos = 2.

Downloaded from Arrow.com.

31.7.25 MPDDRC Interrupt Enable Register

Name: MPDDRC_IER
Offset: 0xC0
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the MPDDRC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access							W	W
Reset							–	–

Bit 1 – RD_ERR Read Error Interrupt Enable

Bit 0 – SEC Security and /or Safety Interrupt Enable

Downloaded from Arrow.com.

31.7.26 MPDDRC Interrupt Disable Register

Name: MPDDRC_IDR
Offset: 0xC4
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the MPDDRC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access							W	W
Reset							–	–

Bit 1 – RD_ERR Read Error Interrupt Disable

Bit 0 – SEC Security and /or Safety Interrupt Disable

Downloaded from Arrow.com.

31.7.27 MPDDRC Interrupt Mask Register

Name: MPDDRC_IMR
Offset: 0xC8
Reset: –
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							RD_ERR	SEC
Access							R	R
Reset							–	–

Bit 1 – RD_ERR Read Error Interrupt Mask

Bit 0 – SEC Security and /or Safety Interrupt Mask

Downloaded from Arrow.com.

31.7.28 MPDDRC Interrupt Status Register

Name: MPDDRC_ISR
Offset: 0xCC
Reset: –
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							RD_ERR	SEC
Access							R	R
Reset							–	–

Bit 1 – RD_ERR Read Error

Value	Description
0	There is no error during memory check.
1	There is one error during memory check.

Bit 0 – SEC Security and /or Safety Event

Value	Description
0	There is no security report in MPDDRC_WPSR.
1	One security flag is set in MPDDRC_WPSR.

Downloaded from Arrow.com.

31.7.29 MPDDRC Safety Register

Name: MPDDRC_SAFETY
Offset: 0xD0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				EN	ADDRESS[27:24]			
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDRESS[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDRESS[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDRESS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 28 – EN Enable Periodic Check of Memory Device

Value	Description
0	Memory check is disabled.
1	Memory check is enabled.

Bits 27:0 – ADDRESS[27:0] Memory Device Address
Memory device address to be checked.

Downloaded from Arrow.com.

31.7.30 MPDDRC Write Protection Mode Register

Name: MPDDRC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

See 31.5.8 Register Write Protection for the list of registers that can be protected.

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				FIRSTE			WPITEN	WPEN
Access				R/W			R/W	R/W
Reset				0			0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x444452	PASSWD	Writing any other value in this field aborts the write operation of the WPEN and WPITEN bits. Always reads as 0.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in MPDDRC_WPSR.WPVSRC and the last software control error type is reported in MPDDRC_WPSR.SWETYP. The MPDDRC_ISR.SEC flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in MPDDRC_WPSR.WPVSRC and only the first software control error type is reported in MPDDRC_WPSR.SWETYP. The MPDDRC_ISR.SEC flag is set at the first error occurrence within a series.

Bit 1 – WPITEN Write Protection Interruption Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x444452 (“DDR” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x444452 (“DDR” in ASCII).

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x444452 (“DDR” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x444452 (“DDR” in ASCII).

Downloaded from Arrow.com.

31.7.31 MPDDRC Write Protection Status Register

Name: MPDDRC_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS						SWETYP[1:0]	
Access	R						R	R
Reset	0						0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SWE	SEQE	CGD	WPVS
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – ECLASS Software Error Class (cleared on read)
0 (WARNING): An abnormal access that does not affect system functionality is performed.
1 (ERROR): An access is performed into some registers after memory device initialization sequence.

Bits 25:24 – SWETYP[1:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	A write-only register has been read (warning).
1	WRITE_RO	MPDDRC is enabled and a write access has been performed on a read-only register (warning).
2	UNDEF_RW	Access to an undefined address (warning).
3	W_AFTER_INIT	Abnormal use of MPDDRC user interface when memory device is already configured and initialized, i.e., if MPDDRC_RTR.COUNT > 0 (error).

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of MPDDRC_WPSR.
1	A software error has occurred since the last read of MPDDRC_WPSR. The field SWE details the type of software error. The associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of MPDDRC_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of MPDDRC_WPSR. This flag can only be set under abnormal operating conditions.

Downloaded from Arrow.com.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	The clock monitoring circuitry has not been corrupted since the last read of MPDDRC_WPSR. Under normal operating conditions, this bit is always cleared.
1	The clock monitoring circuitry has been corrupted since the last read of MPDDRC_WPSR. This flag can only be set in case of an abnormal clock signal waveform (glitch).

Bit 0 – WPVS Write Protection Enable

Value	Description
0	No write protection violation occurred since the last read of this register (MPDDRC_WPSR).
1	A write protection violation occurred since the last read of this register (MPDDRC_WPSR). If this violation is an unauthorized attempt to write a control register, the associated violation is reported into the WPVSRC field.

Downloaded from Arrow.com.

32. SDRAM Controller (SDRAMC)

32.1 Description

The SDRAM Controller (SDRAMC) extends the memory capabilities of a chip by providing the interface to external 16-bit and 32-bit DRAM devices. The page size supports ranges from 2048 to 8192 and the number of columns from 256 to 2048. It supports byte (8-bit), half-word (16-bit) and word (32-bit) accesses.

The SDRAMC supports a read or write burst length of one location. It keeps track of the active row in each bank, thus maximizing SDRAM performance, for example, the application may be placed in one bank and data in the other banks. For optimized performance, it is advisable to avoid accessing different rows in the same bank.

The SDRAMC supports a CAS latency of 2 or 3 and optimizes the read access depending on the frequency.

The different modes available – Self-refresh, Powerdown and Deep Powerdown modes – minimize power consumption on the SDRAM device.

32.2 Embedded Characteristics

- Numerous Configurations Supported
  - 2K, 4K, 8K row address memory parts
  - SDRAM with two or four internal banks
  - SDRAM with 16-bit and 32-bit data path
- Programming Facilities
  - Word, half-word, byte access
  - Automatic Page break when memory boundary has been reached
  - Multibank ping-pong access
  - Timing parameters specified by software
  - Automatic refresh operation, refresh rate is programmable
  - Automatic update of DS, TCR and PASR parameters (mobile SDRAM devices)
- Energy-Saving Capabilities
  - Self-refresh, Powerdown and Deep Power modes Supported
  - Supports mobile SDRAM devices
- Error Detection
  - Refresh error interrupt
- SDRAM Power-up Initialization by Software
- CAS Latency of 2, 3 Supported
- Auto Precharge Command Not Used
- 256-Mbyte address space with 32-bit data path, 128-Mbyte address space with 16-bit data path
- Zero Wait State Scrambling/Unscrambling Function with User Key
- Multiplexed Address/Data Lines or Address/Data/Command Lines for Reduced Pin Count
- Abnormal Software Access and Internal Sequencer Integrity Check Reports

32.3 Signal Description

Table 32-1. Signal Description

Name	Description	Type	Active Level
SDCK	SDRAM Clock	Output	–
SDCKE	SDRAM Clock Enable	Output	High

Downloaded from Arrow.com.

.....continued

Name	Description	Type	Active Level
SDCS	SDRAMC Chip Select	Output	Low
BA[1:0]	Bank Select Signals	Output	–
RAS	Row Signal	Output	Low
CAS	Column Signal	Output	Low
SDWE	SDRAM Write Enable	Output	Low
NBS[3:0]	Data Mask Enable Signals	Output	Low
SDRAMC_A[12:0]	Address Bus	Output	–
D[31:0]	Data Bus	I/O	–

32.4 Software Interface/SDRAM Organization, Address Mapping

The SDRAM address space is organized into banks, rows, and columns. The SDRAMC allows mapping different memory types according to the values set in the Configuration register (SDRAMC_CR).

The SDRAMC makes the SDRAM device access protocol transparent to the user. The following tables illustrate the SDRAM device memory mapping seen by the user in correlation with the device structure. Various configurations are illustrated.

32.4.1 SDRAM Address Mapping for 32-bit Memory Data Bus Width

Table 32-2. SDRAM Configuration Mapping: 2K Rows, 256/512/1024/2048 Columns

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					Bk[1:0]		Row[10:0]											Column[7:0]								M[1:0]	
				Bk[1:0]			Row[10:0]										Column[8:0]									M[1:0]	
			Bk[1:0]		Row[10:0]											Column[9:0]										M[1:0]	
		Bk[1:0]		Row[10:0]											Column[10:0]											M[1:0]	

Note: M[1:0] is the byte address inside a 32-bit word and Bk[1] = BA1, Bk[0] = BA0.

Table 32-3. SDRAM Configuration Mapping: 4K Rows, 256/512/1024/2048 Columns

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				Bk[1:0]		Row[11:0]												Column[7:0]								M[1:0]	
			Bk[1:0]		Row[11:0]												Column[8:0]									M[1:0]	
		Bk[1:0]		Row[11:0]												Column[9:0]										M[1:0]	
	Bk[1:0]		Row[11:0]												Column[10:0]											M[1:0]	

Note: M[1:0] is the byte address inside a 32-bit word and Bk[1] = BA1, Bk[0] = BA0.

Downloaded from Arrow.com.

Table 32-4. SDRAM Configuration Mapping: 8K Rows, 256/512/1024/2048 Columns

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			Bk[1:0]		Row[12:0]													Column[7:0]								M[1:0]	
		Bk[1:0]			Row[12:0]												Column[8:0]									M[1:0]	
	Bk[1:0]				Row[12:0]											Column[9:0]										M[1:0]	
Bk[1:0]			Row[12:0]												Column[10:0]											M[1:0]	

Note: M[1:0] is the byte address inside a 32-bit word and Bk[1] = BA1, Bk[0] = BA0.

32.4.2 SDRAM Address Mapping for 16-bit Memory Data Bus Width

Table 32-5. SDRAM Configuration Mapping: 2K Rows, 256/512/1024/2048 Columns

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						Bk[1:0]		Row[10:0]										Column[7:0]									M0
					Bk[1:0]		Row[10:0]											Column[8:0]									M0
				Bk[1:0]		Row[10:0]											Column[9:0]									M0
			Bk[1:0]		Row[10:0]											Column[10:0]										M0

Note: M0 is the byte address inside a 16-bit half-word and Bk[1] = BA1, Bk[0] = BA0.

Table 32-6. SDRAM Configuration Mapping: 4K Rows, 256/512/1024/2048 Columns

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					Bk[1:0]		Row[11:0]											Column[7:0]									M0
				Bk[1:0]		Row[11:0]												Column[8:0]									M0
			Bk[1:0]		Row[11:0]												Column[9:0]									M0
		Bk[1:0]		Row[11:0]												Column[10:0]										M0

Note: M0 is the byte address inside a 16-bit half-word and Bk[1] = BA1, Bk[0] = BA0.

Table 32-7. SDRAM Configuration Mapping: 8K Rows, 256/512/1024/2048 Columns

CPU Address Line																										
27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				Bk[1:0]		Row[12:0]												Column[7:0]									M0
			Bk[1:0]		Row[12:0]												Column[8:0]									M0
		Bk[1:0]		Row[12:0]													Column[9:0]									M0
	Bk[1:0]		Row[12:0]													Column[10:0]										M0

Note: M0 is the byte address inside a 16-bit half-word and Bk[1] = BA1, Bk[0] = BA0.

Downloaded from Arrow.com.

32.5 Product Dependencies

32.5.1 SDRAM Device Initialization

The initialization sequence is generated by software. The sequence to initialize SDRAM devices is the following:

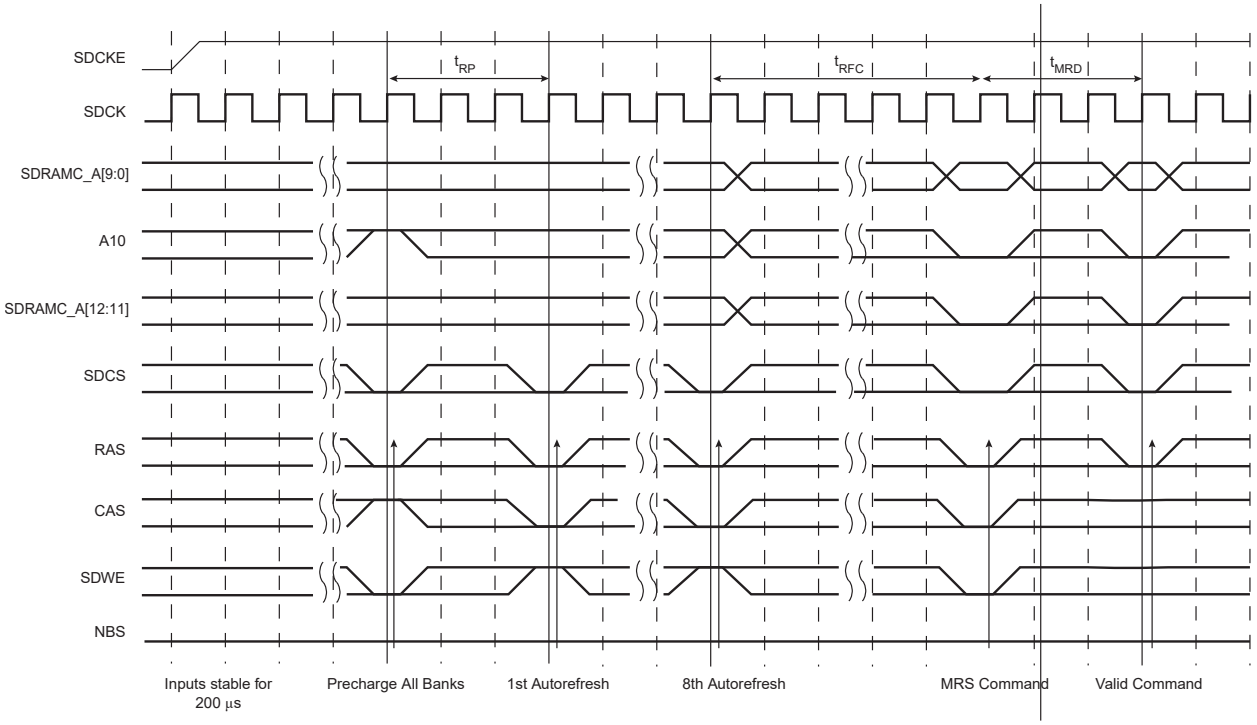
1. Set the SDRAM features in the SDRAMC_CR: asynchronous timings (TRC, TRAS, etc.), number of columns, number of rows, CAS latency and data bus width. Set UNAL bit in SDRAMC_CFR1.
2. For mobile SDRAM, configure temperature-compensated self-refresh (TCSR), drive strength (DS) and partial array self-refresh (PASR) in the Low Power register (SDRAMC_LPR).
3. Select the SDRAM memory device type and the shift sampling value in the Memory Device register (SDRAMC_MDR).
4. A pause of at least 200 µs must be observed before a signal toggle.
5. A NOP command is issued to the SDRAM devices. The application must write a 1 to the MODE field in the Mode register (SDRAMC_MR) (see **Note**). Read the SDRAMC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any SDRAM address.
6. An All Banks Precharge command is issued to the SDRAM. The application must write a 2 to the MODE field in the SDRAMC_MR. Read the SDRAMC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any SDRAM address.
7. Eight autorefresh (CBR) cycles are provided. The application must set the MODE field to 4 in the SDRAMC_MR. Read the SDRAMC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any SDRAM location eight times.
8. A Mode Register set (MRS) cycle is issued to program the parameters of the SDRAM, in particular CAS latency and burst length. The application must write a 3 to the MODE field in the SDRAMC_MR. Read the SDRAMC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the SDRAM. The write address must be chosen so that BA[1:0] are set to 0. For example, with a 16-bit 128 MB SDRAM (12 rows, 9 columns, 4 banks) bank address, the SDRAM write access should be done at the address 0x20000000.
9. For mobile SDRAM initialization, an Extended Mode Register set (EMRS) cycle is issued to program the SDRAM parameters (TCSR, PASR, DS). The application must set the MODE field to 5 in the SDRAMC_MR. Read the SDRAMC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the SDRAM. The write address must be chosen so that BA[1] or BA[0] are set to 1. For example, with a 16-bit 128 MB SDRAM (12 rows, 9 columns, 4 banks) bank address, the SDRAM write access should be done at address 0x20800000 or 0x20400000.
10. The application must go into Normal mode. Configure MODE to 0 in the SDRAMC_MR. Read the SDRAMC_MR and add a memory barrier assembler instruction just after the read. Perform a write access at any location in the SDRAM.
11. Write the refresh rate into the COUNT field in the Refresh Timer register (SDRAMC_TR). (Refresh rate = delay between refresh cycles). The SDRAM device requires a refresh every 15.625 µs or 7.81 µs. With a 100 MHz frequency, the Refresh Timer register must be set with the value 1562 (15.625 µs x 100 MHz) or 781 (7.81 µs x 100 MHz).

After initialization, the SDRAM devices are fully functional.

Note: The instructions stated in Step 5 of the initialization process must be respected to make sure the subsequent commands issued by the SDRAMC are taken into account.

Downloaded from Arrow.com.

Figure 32-1. SDRAM Device Initialization Sequence



32.5.2 I/O Lines

The pins used for interfacing the SDRAMC may be multiplexed with the PIO lines. The programmer must first program the PIO controller to assign the SDRAMC pins to their peripheral function. If I/O lines of the SDRAMC are not used by the application, they can be used for other purposes by the PIO Controller.

32.5.3 Power Management

The SDRAMC may be clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the SDRAMC clock.

The SDRAM clock on pin SDCK is output as soon as the first access to the SDRAM is made during the initialization phase. To stop the SDRAM clock signal, the SDRAMC_LPR must be programmed with the self-refresh command.

32.5.4 Interrupt Sources

Using the SDRAMC interrupt requires the interrupt controller to be programmed first.

The SDRAMC interrupt line (Refresh Error notification) is connected to the interrupt controller through the memory controller (external bus interface) interrupt line.

SDRAMC refresh error notification interrupt line is ORed with other system peripheral interrupt lines and is finally provided as the system interrupt source to the interrupt controller.

32.6 Functional Description

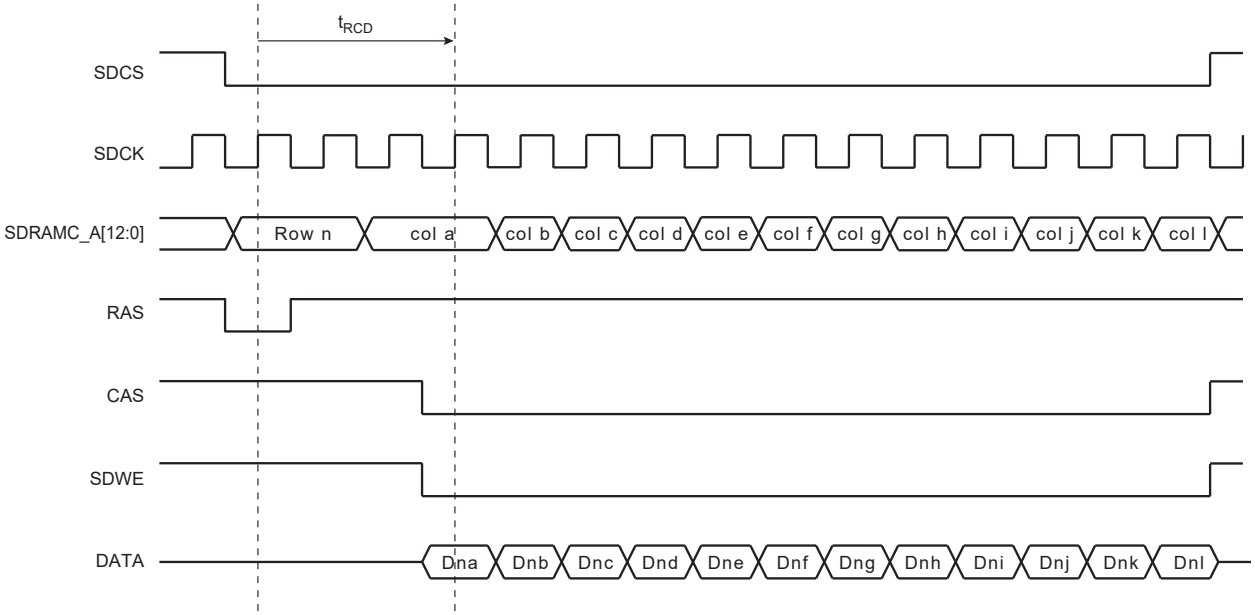
32.6.1 SDRAM Controller Write Cycle

The SDRAMC allows burst access or single access. In both cases, the SDRAMC keeps track of the active row in each bank, thus maximizing performance. To initiate a burst access, the SDRAMC uses the transfer type signal provided by the master requesting the access. If the next access is a sequential write access, writing to the SDRAM device is carried out. If the next access is a write-sequential access, but the current access is to a boundary page, or if the next access is in another row, then the SDRAMC generates a precharge command, activates the new row and initiates a write command. To comply with SDRAM timing parameters, additional clock cycles are inserted between

Downloaded from Arrow.com.

precharge and active commands (t_{RP}), and between active and write commands (t_{RCD}). For definition of these timing parameters, refer to the SDRAMC Configuration Register. Refer to the following figure.

Figure 32-2. Write Burst SDRAM Access



32.6.2 SDRAM Controller Read Cycle

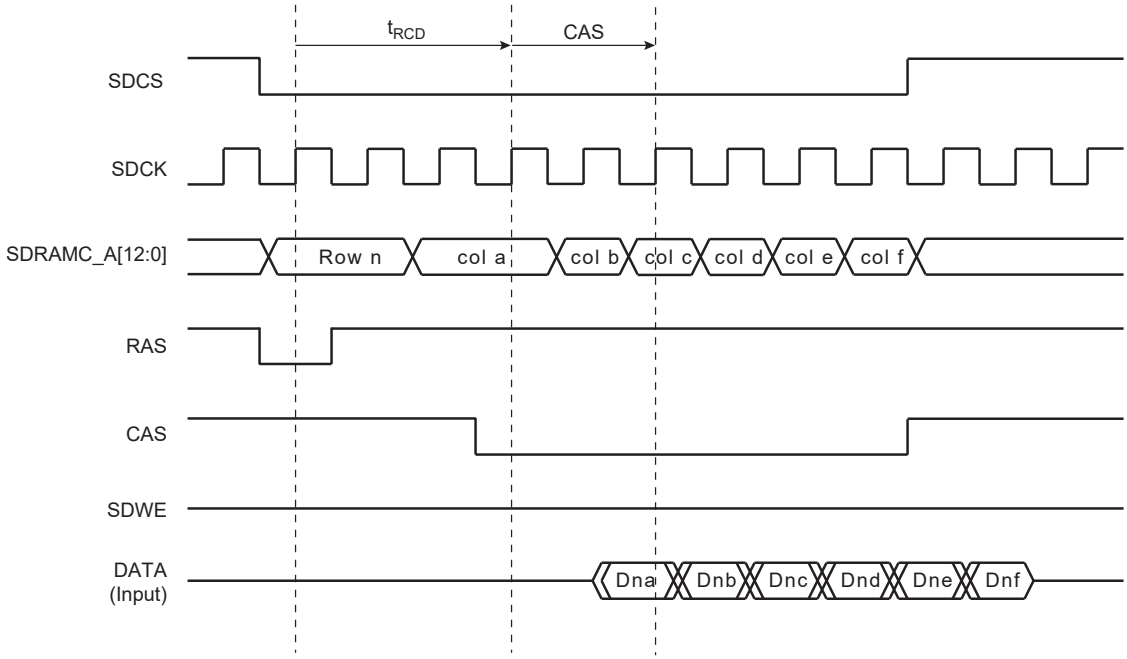
The SDRAMC allows burst access, incremental burst of unspecified length or single access. In all cases, the SDRAMC keeps track of the active row in each bank, thus maximizing performance of the SDRAM. If row and bank addresses do not match the previous row/bank address, then the SDRAMC automatically generates a precharge command, activates the new row and starts the read command. To comply with the SDRAM timing parameters, additional clock cycles on SDCK are inserted between precharge and active commands (t_{RP}), and between active and read commands (t_{RCD}). These two parameters are set in the SDRAMC_CR. After a read command, additional wait states are generated to comply with the CAS latency (2 or 3 clock delays specified in the SDRAMC_CR).

For a single access or an incremented burst of unspecified length, the SDRAMC anticipates the next access. While the last value of the column is returned by the SDRAMC on the bus, the SDRAMC anticipates the read to the next column and thus anticipates the CAS latency. This reduces the effect of the CAS latency on the internal bus.

For burst access of specified length (4, 8, 16 words), access is not anticipated. This case leads to the best performance. If the burst is broken (border, Busy mode, etc.), the next access is handled as an incrementing burst of unspecified length.

Downloaded from Arrow.com.

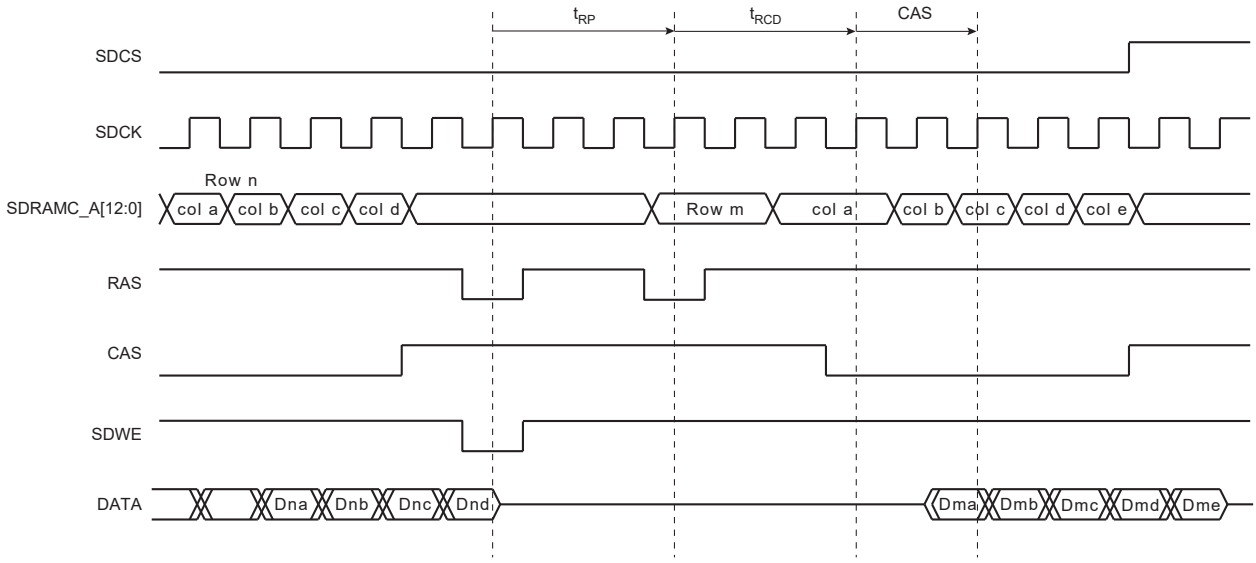
Figure 32-3. Read Burst SDRAM Access



32.6.3 Border Management

When the memory row boundary has been reached, an automatic page break is inserted. In this case, the SDRAMC generates a precharge command, activates the new row and initiates a read or write command. To comply with SDRAM timing parameters, an additional clock cycle is inserted between the precharge and the active command (t_{RP}) and between the active and the read command (t_{RCD}). Refer to the following figure.

Figure 32-4. Read Burst with Boundary Row Access



32.6.4 SDRAM Controller Refresh Cycles

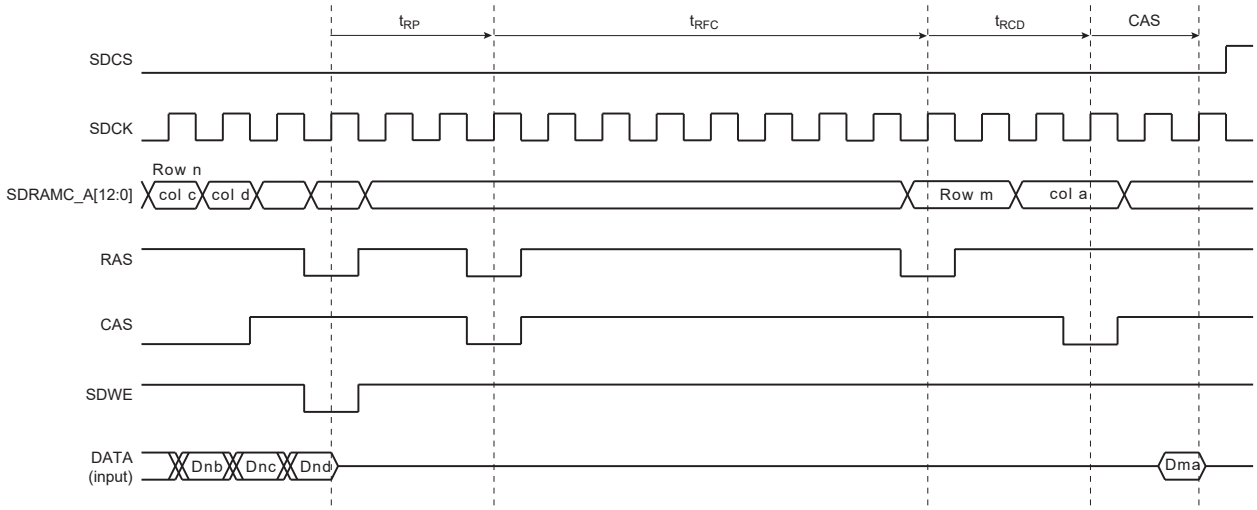
An autorefresh command is used to refresh the SDRAM device. Refresh addresses are generated internally by the SDRAM device and incremented after each autorefresh automatically. The SDRAMC generates these autorefresh commands periodically. An internal timer is loaded with the value in SDRAMC_TR that indicates the number of clock cycles between refresh cycles.

Downloaded from Arrow.com.

A refresh error interrupt is generated when the previous autorefresh command did not perform. It is acknowledged by reading the Interrupt Status register (SDRAMC_ISR).

When the SDRAMC initiates a refresh of the SDRAM device, internal memory accesses are not delayed. However, if the processor tries to access the SDRAM, the slave indicates that the device is busy and the master is held by a wait signal. Refer to the following figure.

Figure 32-5. Refresh Cycle Followed by a Read Access



32.6.5 Power Management

Three low-power modes are available:

- Self-refresh mode: The SDRAM executes its own Autorefresh cycle without control of the SDRAMC. Current drained by the SDRAM is very low.
- Powerdown mode: Autorefresh cycles are controlled by the SDRAMC. Between autorefresh cycles, the SDRAM is in powerdown. Current drained in Powerdown mode is higher than in Self-refresh Mode.
- Deep Powerdown mode (only available with Mobile SDRAM): The SDRAM contents are lost, but the SDRAM does not drain any current.

The SDRAMC activates one low-power mode as soon as the SDRAM device is not selected. It is possible to delay the entry in Self-refresh and Powerdown modes after the last access by programming a timeout value in the SDRAMC_LPR.

32.6.5.1 Self-refresh Mode

This mode is selected by configuring SDRAMC_LPR.LPCB to 1. In Self-refresh mode, the SDRAM device retains data without external clocking and provides its own internal clocking, thus performing its own autorefresh cycles. All the inputs to the SDRAM device become “don’t care” except SDCKE, which remains low. As soon as the SDRAM device is selected, the SDRAMC provides a sequence of commands and exits Self-refresh mode.

Some low-power SDRAMs (e.g., mobile SDRAM) can refresh only one-quarter or a half quarter or all banks of the SDRAM array. This feature reduces the self-refresh current. To configure this feature, Temperature Compensated Self-Refresh (TCSR), Partial Array Self-Refresh (PASR) and Drive Strength (DS) must be set in the SDRAMC_LPR and transmitted to the low-power SDRAM during initialization.

After initialization, as soon as the PASR/DS/TCSR fields are modified and Self-refresh mode is activated, the Extended Mode register is accessed automatically and the PASR/DS/TCSR bits are updated before entry into Self-refresh mode. This feature is not supported when SDRAMC shares an external bus with another controller.

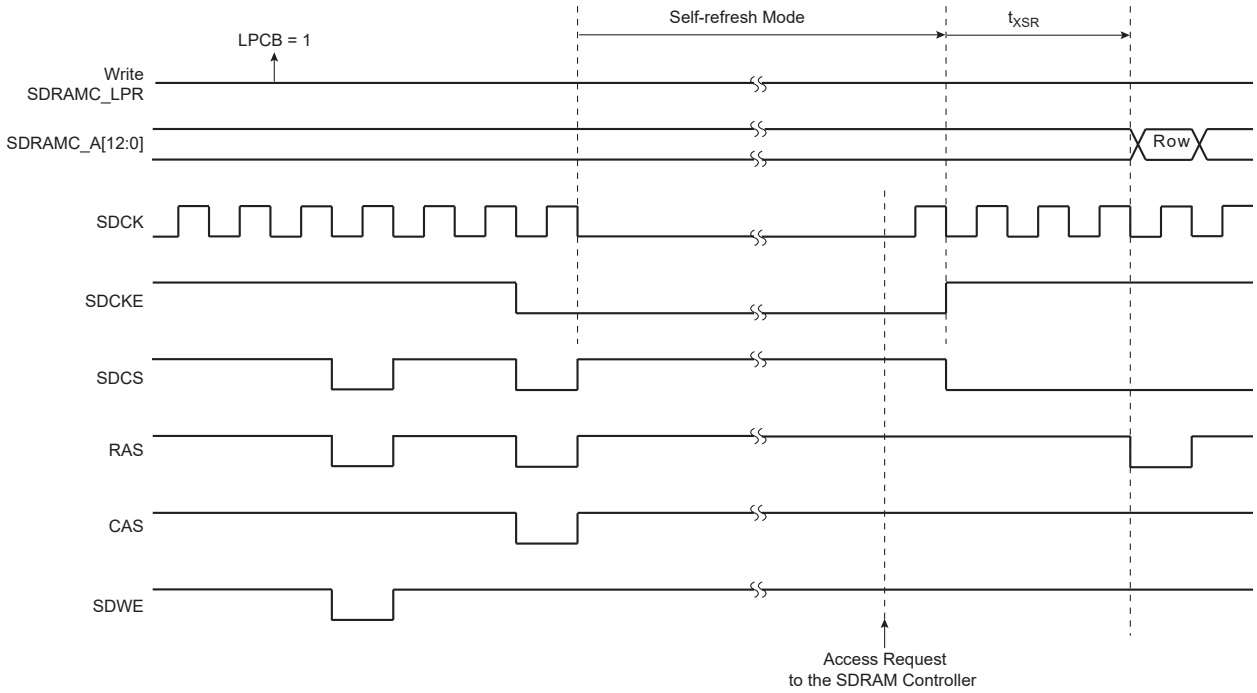
The SDRAM device must remain in Self-refresh mode for a minimum period of t_{RAS} and may remain in Self-refresh mode for an indefinite period. Refer to the following figure.

Upon exiting Self-refresh mode, an autorefresh command can be issued immediately after t_{XSR} by setting the bit SELFAUTO to 1 (refer to the SDRAMC Low-Power Register). Otherwise, an active command is issued immediately after t_{XSR} and an autorefresh command is issued every 7.81 µs or 15.6 µs.

Downloaded from Arrow.com.

Note: Some SDRAM providers impose some cycles of burst autorefresh immediately before self-refresh entry and immediately after self-refresh exit. For example, a SDRAM with 4096 rows will impose 4096 cycles of burst autorefresh. This constraint is not supported.

Figure 32-6. Self-refresh Mode Behavior



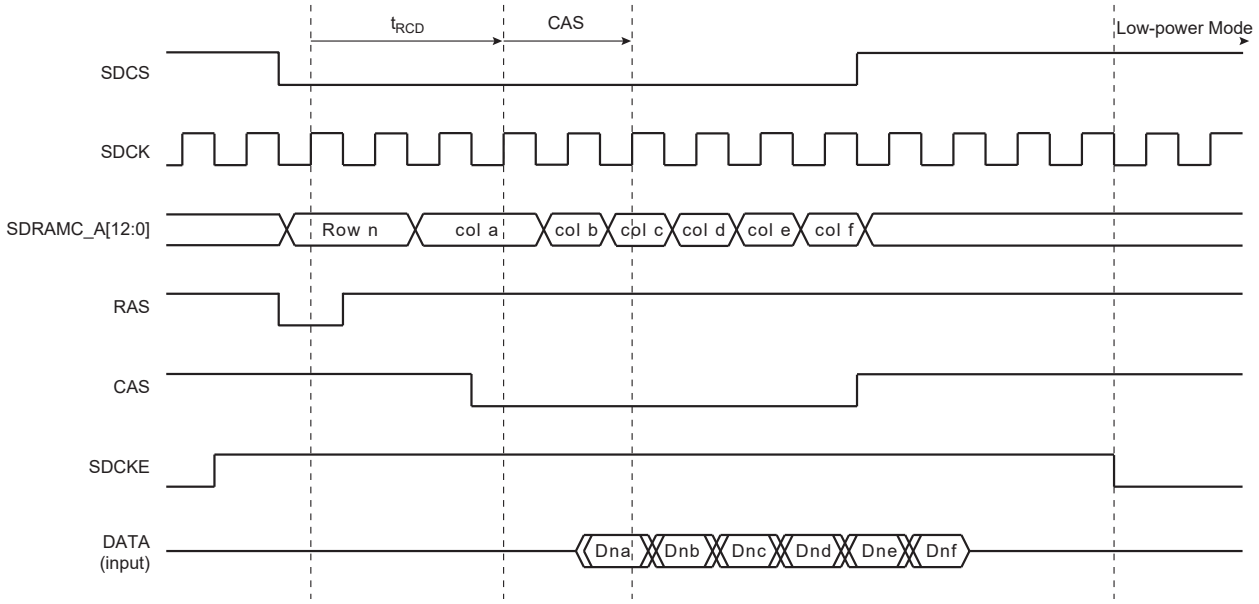
32.6.5.2 Low-power Mode

This mode is selected by configuring SDRAMC_LPR.LPCB to 2. Power consumption is greater than in Self-refresh mode. All the input and output buffers of the SDRAM device are deactivated except SDCKE, which remains low. In contrast to Self-refresh mode, the SDRAM device cannot remain in Low-power mode longer than the refresh period (64 ms for a whole device refresh operation). As no autorefresh operations are performed by the SDRAM itself, the SDRAMC carries out the refresh operation. The exit procedure is faster than in Self-refresh mode.

Refer to the following figure.

Downloaded from Arrow.com.

Figure 32-7. Low-power Mode Behavior



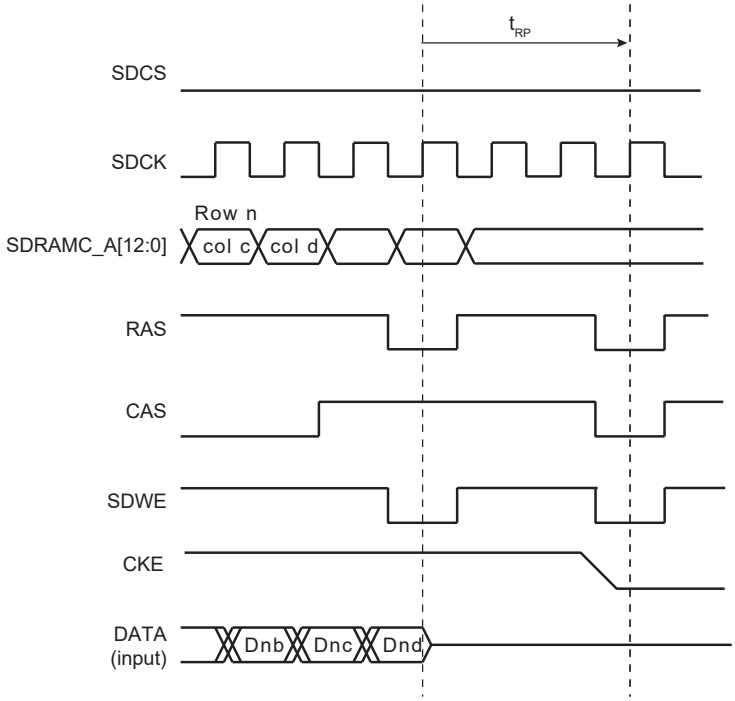
32.6.5.3 Deep Powerdown Mode

This mode is selected by configuring SDRAMC_LPR.LPCB to 3. When this mode is activated, all internal voltage generators inside the SDRAM are stopped and all data is lost.

When this mode is enabled, the application must not access the SDRAM until a new initialization sequence is done (see "SDRAM Device Initialization").

Refer to the following figure.

Figure 32-8. Deep Powerdown Mode Behavior



Downloaded from Arrow.com.

32.6.6 Scrambling/Unscrambling Function

The external data bus can be scrambled in order to prevent intellectual property data located in off-chip memories from being easily recovered by analyzing data at the package pin level of either microcontroller or memory device.

The scrambling and unscrambling are performed on-the-fly without additional wait states.

The scrambling/unscrambling function can be enabled or disabled by configuring the SDR_SE bit in the OCMS register (SDRAMC_OCMS). This bit cannot be reconfigured as long as the external memory device is powered.

The scrambling method depends on two user-configurable key registers, SDRAMC_OCMS_KEY1 and SDRAMC_OCMS_KEY2 plus a random value depending on device processing characteristics. These key registers are only accessible in Write mode.

The scrambling user key or the seed for key generation must be securely stored in a reliable nonvolatile memory in order to recover data from the off-chip memory. Any data scrambled with a given key cannot be recovered if the key is lost.

When multiple chip selects are handled, it is possible to configure the scrambling function per chip select using the OCMS field in the SDRAMC_OCMS registers.

32.6.7 Clearing Scrambling Keys on Tamper Event

On tamper detection event on WKUP pins, it is possible to perform an immediate clear of the scrambling keys (SDRAMC_OCMS_KEY1 and SDRAMC_OCMS_KEY2) if bit SDRAMC_OCMS.TAMPCLR = 1.

32.6.8 Interface with Multiplexed Data/Address Lines and Data/Address/Command Lines

This feature takes advantage of the SDRAM protocol to reduce the pin count by multiplexing address and data lines or address, data and command lines. Up to 16 lines can be reduced depending on the configuration of the SDRAM.

Using this feature reduces the efficiency of the SDRAM by adding one clock cycle during write access.

When address, data and command lines are multiplexed, as SDA10 is multiplexed, refresh operations increase the latency. It is not possible to perform an access to another device during auto-refresh process.

Figure 32-9. Multiplexed Address/Data, Connection to 1x16 SDRAM Interface

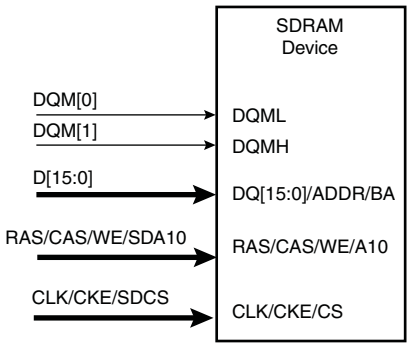


Figure 32-10. Multiplexed Address/Data/Command, Connection to 1x16 SDRAM Interface

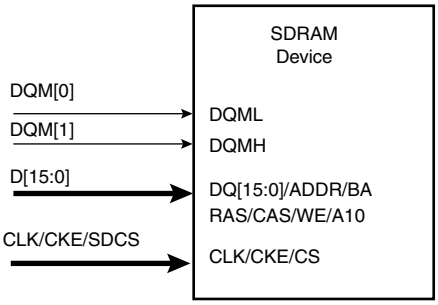


Table 32-8. 16-Mbit SDR-SDRAM, 512k*16* 2 Banks: Multiplexed Address/Data

	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Downloaded from Arrow.com.

CLK, CKE, SDCS, DM[1:0], RAS, CAS, WE	–	–	–	–	BK	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

Table 32-9. 16-Mbit SDR-SDRAM, 512k*16* 2 Banks: Multiplexed Address/Data/Command

	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
CLK, CKE, SDCS, DM[1:0]	RAS	CAS	WE	–	BK	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

Table 32-10. 64-Mbit SDR-SDRAM, 1 Meg*16* 4 Banks, 128-Mbit SDR-SDRAM, 2 Meg*16* 4 Banks: Multiplexed Address/Data

	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
CLK, CKE, SDCS, DM[1:0], RAS, CAS, WE	–	–	BK	BK	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

Table 32-11. 64-Mbit SDR-SDRAM, 1 Meg*16* 4 Banks, 128-Mbit SDR-SDRAM, 2 Meg*16* 4 Banks: Multiplexed Address/Data/ Command

	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
CLK, CKE, SDCS, DM[1:0], RAS	CAS	WE	BK	BK	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

Table 32-12. 256-Mbit SDR-SDRAM, 4 Meg*16* 4 Banks: Multiplexed Address/Data

	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
CLK, CKE, SDCS, DM[1:0], RAS, CAS, WE	–	BK	BK	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

Table 32-13. 256-Mbit SDR-SDRAM, 4 Meg*16* 4 Banks: Multiplexed Address/Data/Command

	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
CLK, CKE, SDCS, DM[1:0], RAS, CAS	WE	BK	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

32.6.9 Register Write Protection

To prevent any single software error from corrupting SDRAMC behavior, some registers in the address space can be write-protected by setting the WPEN and WPITEN bits in the SDRAMC Write Protection Mode Register (SDRAMC_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the SDRAMC Write Protection Status Register (SDRAMC_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the SDRAMC_WPSR.

The following registers can be write-protected:

- SDRAMC Mode Register
- SDRAMC Refresh Timer Register
- SDRAMC Configuration Register
- SDRAMC Memory Device Register
- SDRAMC Configuration Register 1
- SDRAMC OCMS Register
- SDRAMC OCMS KEY1 Register
- SDRAMC OCMS KEY2 Register

The following registers can be write-protected when WPITEN is enabled:

- SDRAMC Interrupt Enable Register
- SDRAMC Interrupt Disable Register

32.6.10 Security and Safety Analysis and Reports

Several types of checks are performed when the SDRAMC is accessing the memory device.

The peripheral clock of the SDRAMC is monitored by specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the SDRAMC. Corruption on the triggering edge of the clock or a pulse with

Downloaded from Arrow.com.

a minimum duration may be identified. If the flag SDRAMC_WPSR.CGD is set, an abnormal condition occurred on the peripheral clock. This flag is not set under normal operating conditions.

The internal sequencer of the SDRAMC is also monitored and if an abnormal state is detected, the flag SDRAMC_WPSR.SEQE is set. This flag is not set under normal operating conditions.

The software accesses to the SDRAMC are monitored and if an incorrect access is performed, the flag SDRAMC_WPSR.SWE is set. The type of incorrect/abnormal software access is reported in the SDRAMC_WPSR.SWETYP field (see 32.7.15 SDRAMC_WPSR for details) , e.g. , writing a new configuration (SDRAMC_CR, SDRAMC_CFR1, SDRAMC_HSR, SDRAMC_OCMS, SDRAMC_KEY1/2) after the initializat ion of the SDRAMC (i .e. , i f SDRAMC_TR.COUNT > 0) i s an er ror . SDRAMC_WPSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The flags CGD, SEQE, SWE and WPVS are automatically cleared when SDRAMC_WPSR is read.

If one of these flags is set, the flag SDRAMC_ISR.SECE is set and can trigger an interrupt if the SDRAMC_IMR.SECE bit is ‘1’. SECE is cleared by reading SDRAMC_ISR.

Downloaded from Arrow.com.

32.7 Register Summary

Offset	Name	Bit Pos.								
0x00	SDRAMC_MR	31:24								
		23:16								
		15:8								
		7:0						MODE[2:0]		
0x04	SDRAMC_TR	31:24								
		23:16								
		15:8					COUNT[11:8]			
		7:0	COUNT[7:0]							
0x08	SDRAMC_CR	31:24	TXSR[3:0]				TRAS[3:0]			
		23:16	TRCD[3:0]				TRP[3:0]			
		15:8	TRC_TRFC[3:0]				TWR[3:0]			
		7:0	DBW	CAS[1:0]		NB	NR[1:0]		NC[1:0]	
0x0C ... 0x0F	Reserved									
0x10	SDRAMC_LPR	31:24								
		23:16								SELFDONE
		15:8		SELFAUTO	TIMEOUT[1:0]		DS[1:0]		TCSR[1:0]	
		7:0		PASR[2:0]					LPCB[1:0]	
0x14	SDRAMC_IER	31:24								
		23:16								
		15:8								
		7:0							SECE	RES
0x18	SDRAMC_IDR	31:24								
		23:16								
		15:8								
		7:0							SECE	RES
0x1C	SDRAMC_IMR	31:24								
		23:16								
		15:8								
		7:0							SECE	RES
0x20	SDRAMC_ISR	31:24								
		23:16								
		15:8								
		7:0							SECE	RES
0x24	SDRAMC_MDR	31:24								
		23:16								
		15:8								
		7:0			SHIFT_SAMPLING[1:0]				MD[1:0]	
0x28	SDRAMC_CFR1	31:24								
		23:16								
		15:8					CMD_MUX	ADD_DATA_MUX		UNAL
		7:0					TMRD[3:0]			
0x2C	SDRAMC_OCMS	31:24								
		23:16								
		15:8								
		7:0				TAMPCLR				SDR_SE
0x30	SDRAMC_OCMS_KEY1	31:24	KEY1[31:24]							
		23:16	KEY1[23:16]							
		15:8	KEY1[15:8]							
		7:0	KEY1[7:0]							
0x34	SDRAMC_OCMS_KEY2	31:24	KEY2[31:24]							
		23:16	KEY2[23:16]							
		15:8	KEY2[15:8]							
		7:0	KEY2[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x38 ... 0x3B	Reserved									
0x3C	SDRAMC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPITEN	WPEN
0x40	SDRAMC_WPSR	31:24	ECLASS					SWETYP[2:0]		
		23:16								
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPEN

Downloaded from Arrow.com.

32.7.1 SDRAMC Mode Register

Name: SDRAMC_MR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SDRAMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						MODE[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – MODE[2:0] SDRAMC Command Mode
This field defines the command issued by the SDRAMC when the SDRAM device is accessed.

Value	Name	Description
0	NORMAL	Normal mode. Any access to the SDRAM is decoded normally. To activate this mode, the command must be followed by a write to the SDRAM.
1	NOP	The SDRAMC issues a NOP command when the SDRAM device is accessed regardless of the cycle. To activate this mode, the command must be followed by a write to the SDRAM.
2	ALLBANKS_PRECHARGE	The SDRAMC issues an “All Banks Precharge” command when the SDRAM device is accessed regardless of the cycle. To activate this mode, the command must be followed by a write to the SDRAM.
3	LOAD_MODEREG	The SDRAMC issues a “Load Mode Register” command when the SDRAM device is accessed regardless of the cycle. To activate this mode, the command must be followed by a write to the SDRAM.
4	AUTO_REFRESH	The SDRAMC issues an “Autorefresh” Command when the SDRAM device is accessed regardless of the cycle. Previously, an “All Banks Precharge” command must be issued. To activate this mode, the command must be followed by a write to the SDRAM.
5	EXT_LOAD_MODEREG	The SDRAMC issues an “Extended Load Mode Register” command when the SDRAM device is accessed regardless of the cycle. To activate this mode, the “Extended Load Mode Register” command must be followed by a write to the SDRAM. The write in the SDRAM must be done in the appropriate bank; most low-power SDRAM devices use the bank 1.
6	DEEP_POWERDOWN	Deep Powerdown mode. Enters Deep Powerdown mode.

Downloaded from Arrow.com.

32.7.2 SDRAMC Refresh Timer Register

Name: SDRAMC_TR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SDRAMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					COUNT[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – COUNT[11:0] SDRAMC Refresh Timer Count
This 12-bit field is loaded into a timer that generates the refresh pulse. Each time the refresh pulse is generated, a refresh burst is initiated. The SDRAM device requires a refresh every 15.625 µs or 7.81 µs. With a 100 MHz frequency, the Refresh Timer Counter Register must be set with the value 1562 (15.625 µs x 100 MHz) or 781 (7.81 µs x 100 MHz).
To refresh the SDRAM device, this 12-bit field must be written. If this condition is not satisfied, no refresh command is issued and no refresh of the SDRAM device is carried out.

Downloaded from Arrow.com.

32.7.3 SDRAMC Configuration Register

Name: SDRAMC_CR
Offset: 0x08
Reset: 0x852372C0
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SDRAMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	TXSR[3:0]				TRAS[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	1	0	1

Bit	23	22	21	20	19	18	17	16
	TRCD[3:0]				TRP[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	1	1

Bit	15	14	13	12	11	10	9	8
	TRC_TRFC[3:0]				TWR[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	1	0	0	1	0

Bit	7	6	5	4	3	2	1	0
	DBW	CAS[1:0]		NB	NR[1:0]		NC[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	0	0	0	0	0	0

Bits 31:28 – TXSR[3:0] Exit Self-Refresh to Active Delay
Reset value is eight cycles.
This field defines the delay between SCKE set high and an Activate Command in number of cycles. Number of cycles is between 0 and 15.

Bits 27:24 – TRAS[3:0] Active to Precharge Delay
Reset value is five cycles.
This field defines the delay between an Activate Command and a Precharge Command in number of cycles. Number of cycles is between 0 and 15.

Bits 23:20 – TRCD[3:0] Row to Column Delay
Reset value is two cycles.
This field defines the delay between an Activate Command and a Read/Write Command in number of cycles. Number of cycles is between 0 and 15.

Bits 19:16 – TRP[3:0] Row Precharge Delay
Reset value is three cycles.
This field defines the delay between a Precharge Command and another Command in number of cycles. Number of cycles is between 0 and 15.

Bits 15:12 – TRC_TRFC[3:0] Row Cycle Delay and Row Refresh Cycle
Reset value is seven cycles.
This field defines two timings:

- the delay (t_{RFC}) between two Refresh commands and between a Refresh command and an Activate command
- the delay (t_{RC}) between two Active commands in number of cycles.

The number of cycles is between 0 and 15. The end user must program max {t_{RC}, t_{RFC}}.

Downloaded from Arrow.com.

Bits 11:8 – TWR[3:0] Write Recovery Delay
Reset value is two cycles.
This field defines the Write Recovery Time in number of cycles. Number of cycles is between 0 and 15.

Bit 7 – DBW Data Bus Width
Reset value is 16 bits.

Value	Description
0	Data bus width is 32 bits.
1	Data bus width is 16 bits.

Bits 6:5 – CAS[1:0] CAS Latency
Reset value is two cycles. In the SDRAMC, only a CAS latency of one, two and three cycles is managed.

Value	Name	Description
0	–	Reserved
1	–	Reserved
2	LATENCY2	2-cycle latency
3	LATENCY3	3-cycle latency

Bit 4 – NB Number of Banks
Reset value is two banks.

Value	Name	Description
0	BANK2	2 banks
1	BANK4	4 banks

Bits 3:2 – NR[1:0] Number of Row Bits
Reset value is 11 row bits.

Value	Name	Description
0	ROW11	11 bits to define the row number, up to 2048 rows
1	ROW12	12 bits to define the row number, up to 4096 rows
2	ROW13	13 bits to define the row number, up to 8192 rows
3	Reserved	

Bits 1:0 – NC[1:0] Number of Column Bits
Reset value is 8 column bits.

Value	Name	Description
0	COL8	8 bits to define the column number, up to 256 columns.
1	COL9	9 bits to define the column number, up to 512 columns.
2	COL10	10 bits to define the column number, up to 1024 columns.
3	COL11	11 bits to define the column number, up to 2048 columns.

Downloaded from Arrow.com.

32.7.4 SDRAMC Low-Power Register

Name: SDRAMC_LPR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Access		R/W	R/W	R/W			R/W	R/W
Reset		0	0	0			0	0

Bit 16 – SELFDONE Self-refresh Done (read-only)
This bit indicates that the external device is in Self-refresh mode.

Bit 14 – SELFAUTO Self-refresh Exit Autorefresh

Value	Description
1	Upon exiting Self-refresh mode, autorefresh command is immediately performed after t_{XSR}.
0	Upon exiting Self-refresh mode, active command is immediately performed after t_{XSR}. The autorefresh command is issued every 15.6 µs or less.

Bits 13:12 – TIMEOUT[1:0] Time to Define When Low-power Mode Is Enabled

Value	Name	Description
0	LP_LAST_XFER	The SDRAMC activates the SDRAM Low-power mode immediately after the end of the last transfer.
1	LP_LAST_XFER_64	The SDRAMC activates the SDRAM Low-power mode 64 clock cycles after the end of the last transfer.
2	LP_LAST_XFER_128	The SDRAMC activates the SDRAM Low-power mode 128 clock cycles after the end of the last transfer.
3	Reserved	

Bits 11:10 – DS[1:0] Drive Strength (only for low-power SDRAM)
DS is transmitted to the SDRAM during initialization to select the SDRAM strength of data output. This parameter must be set according to the SDRAM device specification.
After initialization, as soon as the DS field is modified and Self-refresh mode is activated, the Extended Mode Register is accessed automatically and DS bits are updated before entry in Self-refresh mode. This feature is not supported when SDRAMC shares an external bus with another controller.

Downloaded from Arrow.com.

Bits 9:8 – TCSR[1:0] Temperature Compensated Self-Refresh (only for low-power SDRAM)
TCSR is transmitted to the SDRAM during initialization to set the refresh interval during Self-refresh mode depending on the temperature of the low-power SDRAM. This parameter must be set according to the SDRAM device specification.
After initialization, as soon as the TCSR field is modified and Self-refresh mode is activated, the Extended Mode Register is accessed automatically and TCSR bits are updated before entry in Self-refresh mode. This feature is not supported when SDRAMC shares an external bus with another controller.

Bits 6:4 – PASR[2:0] Partial Array Self-refresh (only for low-power SDRAM)
PASR is transmitted to the SDRAM during initialization to specify whether only one quarter, one half or all banks of the SDRAM array are enabled. Disabled banks are not refreshed in Self-refresh mode. This parameter must be set according to the SDRAM device specification.
After initialization, as soon as the PASR field is modified and Self-refresh mode is activated, the Extended Mode Register is accessed automatically and PASR bits are updated before entry in Self-refresh mode. This feature is not supported when SDRAMC shares an external bus with another controller.

Bits 1:0 – LPCB[1:0] Low-power Configuration Bits

Value	Name	Description
0	DISABLED	The low-power feature is inhibited: no Powerdown, Self-refresh or Deep Powerdown command is issued to the SDRAM device.
1	SELF_REFRESH	The SDRAMC issues a Self-refresh command to the SDRAM device, the SDCK clock is deactivated and the SDCKE signal is set low. The SDRAM device leaves the Self-refresh mode when accessed and enters it after the access.
2	POWER_DOWN	The SDRAMC issues a Powerdown Command to the SDRAM device after each access, the SDCKE signal is set to low. The SDRAM device leaves the Powerdown mode when accessed and enters it after the access.
3	DEEP_POWER_DOWN	The SDRAMC issues a Deep Powerdown command to the SDRAM device. This mode is unique to low-power SDRAM.

Downloaded from Arrow.com.

32.7.5 SDRAMC Interrupt Enable Register

Name: SDRAMC_IER
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the SDRAMC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access							W	W
Reset							–	–

Bit 1 – SECE Security and/or Safety Event Interrupt Enable

Bit 0 – RES Refresh Error Interrupt Enable

Downloaded from Arrow.com.

32.7.6 SDRAMC Interrupt Disable Register

Name: SDRAMC_IDR
Offset: 0x18
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the SDRAMC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access							W	W
Reset							SECE	RES

Bit 1 – SECE Security and/or Safety Event Interrupt Disable

Bit 0 – RES Refresh Error Interrupt Disable

Downloaded from Arrow.com.

32.7.7 SDRAMC Interrupt Mask Register

Name: SDRAMC_IMR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							SECE	RES
Access							W	R
Reset							0	0

Bit 1 – SECE Security and/or Safety Event Interrupt Mask

Value	Description
0	The security and/or safety event interrupt is disabled.
1	The security and/or safety event interrupt is enabled.

Bit 0 – RES Refresh Error Interrupt Mask

Value	Description
0	The refresh error interrupt is disabled.
1	The refresh error interrupt is enabled.

Downloaded from Arrow.com.

32.7.8 SDRAMC Interrupt Status Register

Name: SDRAMC_ISR
Offset: 0x20
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							SECE	RES
Access							W	R
Reset							0	0

Bit 1 – SECE Security and/or Safety Event Event (cleared on read)

Value	Description
0	There is no security report in SDRAMC_WPSR.
1	One security flag is set in SDRAMC_WPSR.

Bit 0 – RES Refresh Error Status (cleared on read)

Value	Description
0	No refresh error has been detected since the register was last read.
1	A refresh error has been detected since the register was last read.

Downloaded from Arrow.com.

32.7.9 SDRAMC Memory Device Register

Name: SDRAMC_MDR
Offset: 0x24
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SDRAMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			SHIFT_SAMPLING[1:0]				MD[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 5:4 – SHIFT_SAMPLING[1:0] Shift Sampling Point of Data
Shifts the sampling point of data coming from the memory device. The higher the memory device clock frequency, the higher the SHIFT_SAMPLING value. Refer to the section "Electrical Characteristics".

Value	Name	Description
0	–	Reserved.
1	SHIFT_ONE_CYCLE	Sampling point is shifted by one cycle.
2	SHIFT_TWO_CYCLES	Sampling point is shifted by two cycles.
3	SHIFT_THREE_CYCLES	Sampling point is shifted by three cycles.

Bits 1:0 – MD[1:0] Memory Device Type

Value	Name	Description
0	SDRAM	SDRAM
1	LPSDRAM	Low-power SDRAM
2	–	Reserved
3	–	Reserved

Downloaded from Arrow.com.

32.7.10 SDRAMC Configuration Register 1

Name: SDRAMC_CFR1
Offset: 0x28
Reset: 0x00000002
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SDRAMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					CMD_MUX	ADD_DATA_MUX		UNAL
Access					R/W	R/W		R/W
Reset					0	0		0

Bit	7	6	5	4	3	2	1	0
					TMRD[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	1	0

Bit 11 – CMD_MUX Commands are Multiplexed with Address and Data
To use this feature, ADD_DATA_MUX must be set to '1'. This feature allows to reduce the number of pins.

Value	Name	Description
0	UNSUPPORTED	Commands are not multiplexed with address and data.
1	SUPPORTED	Commands are multiplexed with address and data.

Bit 10 – ADD_DATA_MUX Multiplexed Address and Data
.This feature allows to reduce the number of pins.

Value	Name	Description
0	UNSUPPORTED	Data and address are not multiplexed
1	SUPPORTED	Data and address are multiplexed

Bit 8 – UNAL Always written to 1
This bit must be always written to 1.

Bits 3:0 – TMRD[3:0] Load Mode Register Command to Active or Refresh Command
This field defines the delay between a "Load Mode Register" command and an active or refresh command in number of cycles. Number of cycles is between 0 and 15.

Downloaded from Arrow.com.

32.7.11 SDRAMC OCMS Register

Name: SDRAMC_OCMS
Offset: 0x2C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SDRAMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				TAMPCLR				SDR_SE
Access				R/W				R/W
Reset				0				0

Bit 4 – TAMPCLR Tamper Clear Enable

Value	Description
0	A tamper detection event has no effect on SDRAMC scrambling keys.
1	A tamper detection event immediately clears SDRAMC scrambling keys.

Bit 0 – SDR_SE SDRAM Memory Controller Scrambling Enable

Value	Description
0	Disables off-chip scrambling for SDR-SDRAM access.
1	Enables off-chip scrambling for SDR-SDRAM access.

Downloaded from Arrow.com.

32.7.12 SDRAMC OCMS KEY1 Register

Name: SDRAMC_OCMS_KEY1
Offset: 0x30
Property: Write-only

This register can only be written if the WPEN bit is cleared in the SDRAMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	KEY1[31:24]							
Access	W	W	W	W	W	W	W	W
Reset								

Bit	23	22	21	20	19	18	17	16
	KEY1[23:16]							
Access	W	W	W	W	W	W	W	W
Reset								

Bit	15	14	13	12	11	10	9	8
	KEY1[15:8]							
Access	W	W	W	W	W	W	W	W
Reset								

Bit	7	6	5	4	3	2	1	0
	KEY1[7:0]							
Access	W	W	W	W	W	W	W	W
Reset								

Bits 31:0 – KEY1[31:0] Off-chip Memory Scrambling (OCMS) Key Part 1
When off-chip memory scrambling is enabled, the data scrambling depends on KEY1 and KEY2 values.

Downloaded from Arrow.com.

32.7.13 SDRAMC OCMS KEY2 Register

Name: SDRAMC_OCMS_KEY2
Offset: 0x34
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the SDRAMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	KEY2[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	KEY2[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	KEY2[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	KEY2[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – KEY2[31:0] Off-chip Memory Scrambling (OCMS) Key Part 2
When off-chip memory scrambling is enabled, the data scrambling depends on KEY1 and KEY2 values.

Downloaded from Arrow.com.

32.7.14 SDRAMC Write Protection Mode Register

Name: SDRAMC_WPMR
Offset: 0x3C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
							WPITEN	WPEN
Access							R/W	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x534452	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection of SDRAMC_IER/IDR if WPKEY corresponds to 0x534452 (SDR in ASCII).
1	Enables the write protection of SDRAMC_IER/IDR if WPKEY corresponds to 0x534452 (SDR in ASCII).

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x534452 ("SDR" in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x534452 ("SDR" in ASCII).

Downloaded from Arrow.com.

32.7.15 SDRAMC Write Protection Status Register

Name: SDRAMC_WPSR
Offset: 0x40
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS					SWETYP[2:0]		
Access	R					R	R	R
Reset	0					0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SWE	SEQE	CGD	WPEN
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – ECLASS Software Error Class (cleared on read)

Value	Name	Description
0	WARNING	An abnormal access is performed but it does not affect system functionality.
1	ERROR	An access is performed into some registers after memory device initialization sequence.

Bits 26:24 – SWETYP[2:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	A write-only register has been read (Warning).
1	WRITE_RO	A write access has been performed on a read-only register (Warning).
2	UNDEF_RW	Access to an undefined address (Warning).
3	W_AFTER_INIT	Write access performed into some configuration registers after memory device initialization, i.e. if SDRAMC_TR.COUNT > 0 (Error).

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.
When WPVS=0 and SWE=1, WPVSRC reports the address of the incorrect software access. As soon as WPVS=1, WPVSRC returns the address of the write-protected violation.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of SDRAMC_WPSR.
1	A software error has occurred since the last read of SDRAMC_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of SDRAMC_WPSR.

Downloaded from Arrow.com.

Value	Description
1	A peripheral internal sequencer error has occurred since the last read of SDRAMC_WPSR. This flag can only be set under abnormal operating conditions.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	The clock monitoring circuitry has not been corrupted since the last read of SDRAMC_WPSR. Under normal operating conditions, this bit is always cleared.
1	The clock monitoring circuitry has been corrupted since the last read of SDRAMC_WPSR. This flag can only be set in case of an abnormal clock signal waveform (glitch).

Bit 0 – WPEN Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the QSPI_WPSR.
1	A write protection violation has occurred since the last read of the QSPI_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

33. Static Memory Controller (SMC)

33.1 Description

The Static Memory Controller (SMC) generates the signals that control the access to the external memory devices or peripheral devices. It has six Chip Selects and a 26-bit address bus. The 32-bit data bus can be configured to interface with 8-, 16-, or 32-bit external devices. Separate read and write control signals allow for direct memory and peripheral interfacing. Read and write signal waveforms are fully parametrizable.

The SMC can manage wait requests from external devices to extend the current access. The SMC is provided with an automatic Slow Clock mode. In Slow Clock mode, it switches from user-programmed waveforms to slow-rate specific waveforms on read and write signals. The SMC supports asynchronous burst read in Page mode access for page size up to 32 bytes.

33.2 Embedded Characteristics

- Six Chip Selects Available
- 64-Mbyte Address Space per Chip Select
- 8-, 16- or 32-bit Data Bus
- Word, Halfword, Byte Transfers
- Byte Write or Byte Select Lines
- Programmable Setup, Pulse And Hold Time for Read Signals per Chip Select
- Programmable Setup, Pulse And Hold Time for Write Signals per Chip Select
- Programmable Data Float Time per Chip Select
- Compliant with LCD Module
- External Wait Request
- Automatic Switch to Slow Clock Mode
- Asynchronous Read in Page Mode Supported: Page Size Ranges from 4 to 32 Bytes

33.3 I/O Lines Description

Table 33-1. I/O Lines Description

Name	Description	Type	Active Level
NCS[5:0]	Static Memory Controller Chip Select Lines	Output	Low
NRD	Read Signal	Output	Low
NWR0/NWE	Write 0/Write Enable Signal	Output	Low
A0/NBS0	Address Bit 0/Byte 0 Select Signal	Output	Low
NWR1/NBS1	Write 1/Byte 1 Select Signal	Output	Low
A1/NWR2/NBS2	Address Bit 1/Write 2/Byte 2 Select Signal	Output	Low
NWR3/NBS3	Write 3/Byte 3 Select Signal	Output	Low
A[25:2]	Address Bus	Output	–
D[31:0]	Data Bus	I/O	–
NWAIT	External Wait Signal	Input	Low

Downloaded from Arrow.com.

33.4 Interrupt Source

The SMC has an interrupt line connected to the interrupt line of the external bus interface. The external bus interface (SMC, SDRAMC, MPDDRC) interrupt line is connected to the interrupt controller. The external bus interface interrupt line can be triggered by SMC or SDRAMC/MPDDRC.

33.5 Multiplexed Signals

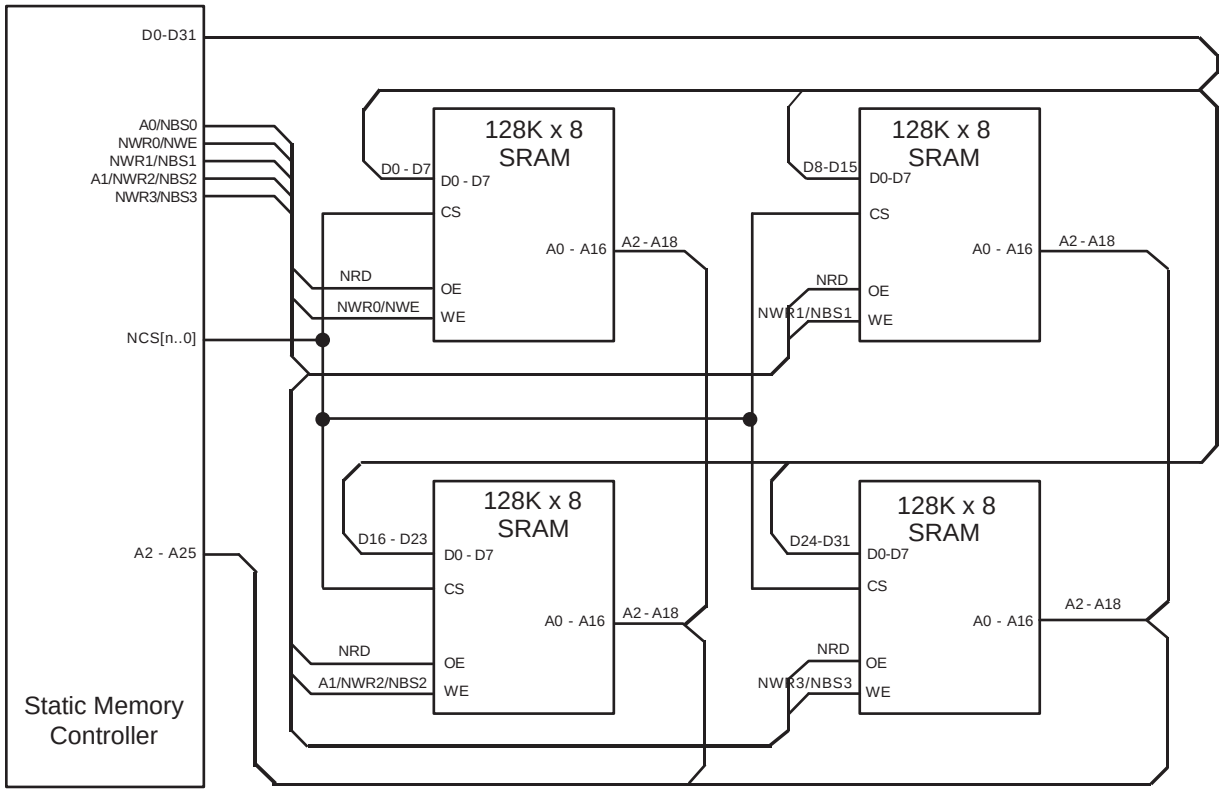
Table 33-2. Static Memory Controller (SMC) Multiplexed Signals

Multiplexed Signals			Related Function
NWR0	NWE	–	Byte-write or byte-select access, see Byte Write or Byte Select Access
A0	NBS0	–	8-bit or 16-/32-bit data bus, see Data Bus Width
NWR1	NBS1	–	Byte-write or byte-select access, see Byte Write or Byte Select Access
A1	NWR2	NBS2	8-/16-bit or 32-bit data bus, see Data Bus Width. Byte-write or byte-select access, see Byte Write or Byte Select Access
NWR3	NBS3	–	Byte-write or byte-select access, see Byte Write or Byte Select Access

33.6 Application Example

33.6.1 Hardware Interface

Figure 33-1. SMC Connections to Static Memory Devices



Downloaded from Arrow.com.

33.7 Product Dependencies

33.7.1 I/O Lines

The pins used for interfacing the Static Memory Controller may be multiplexed with the PIO lines. The programmer must first program the PIO controller to assign the Static Memory Controller pins to their peripheral function. If I/O Lines of the SMC are not used by the application, they can be used for other purposes by the PIO Controller.

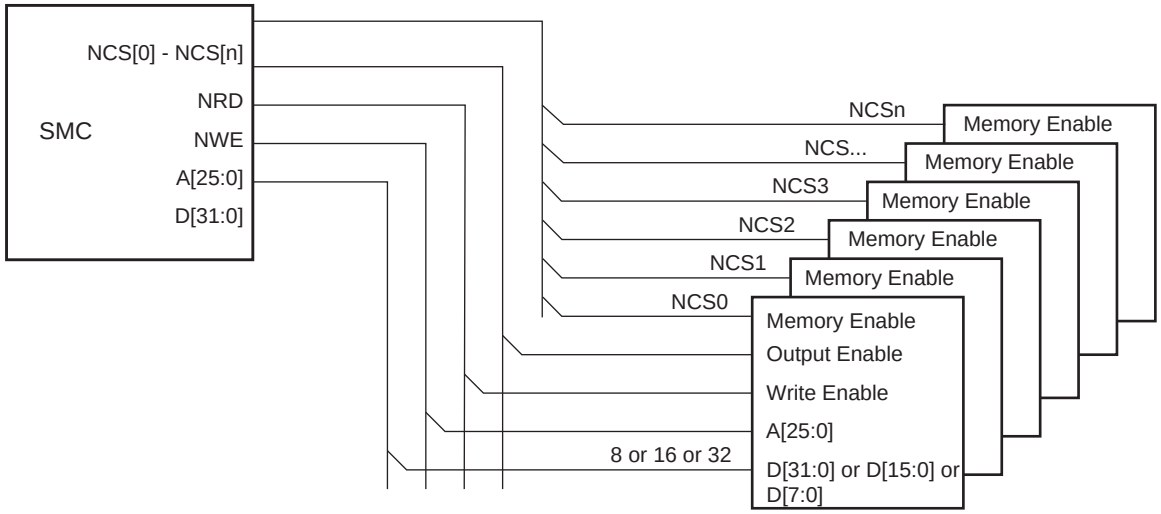
33.8 External Memory Mapping

The SMC provides up to 26 address lines, A[25:0]. This allows each chip select line to address up to 64 Mbytes of memory.

If the physical memory device connected on one chip select is smaller than 64 Mbytes, it wraps around and appears to be repeated within this space. The SMC correctly handles any valid access to the memory device within the page (see the figure below).

A[25:0] is only significant for 8-bit memory, A[25:1] is used for 16-bit memory, A[25:2] is used for 32-bit memory.

Figure 33-2. Memory Connections for Eight External Devices



33.9 Connection to External Devices

33.9.1 Data Bus Width

A data bus width of 8, 16, or 32 bits can be selected for each chip select. This option is controlled by the field DBW in SMC_MODE (Mode register) for the corresponding chip select.

- Figure 33-3 illustrates how to connect a 512K x 8-bit memory on NCS2.
- Figure 33-4 illustrates how to connect a 512K x 16-bit memory on NCS2.
- Figure 33-5 shows two 16-bit memories connected as a single 32-bit memory.

33.9.2 Byte Write or Byte Select Access

Each chip select with a 16-bit or 32-bit data bus can operate with one of two different types of write access: byte write or byte select access. This is controlled by the BAT field of the SMC_MODE register for the corresponding chip select.

Downloaded from Arrow.com.

Figure 33-3. Memory Connection for an 8-bit Data Bus

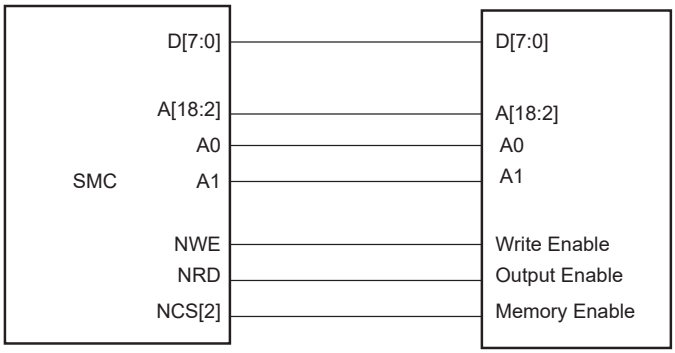


Figure 33-4. Memory Connection for a 16-bit Data Bus

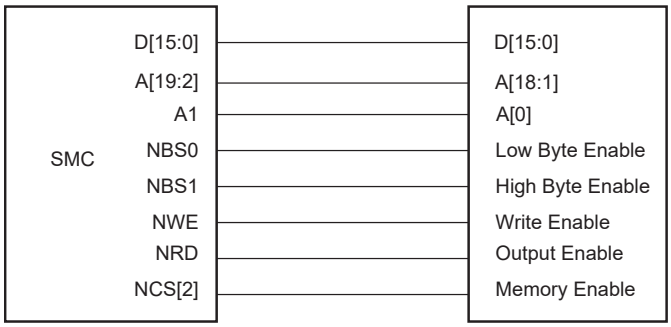
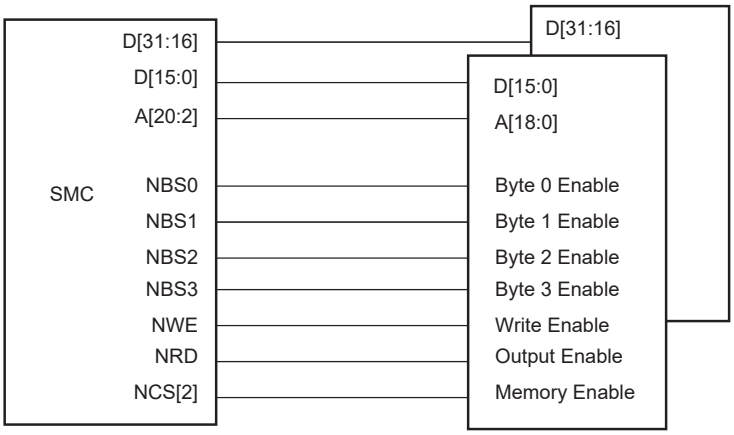


Figure 33-5. Memory Connection for a 32-bit Data Bus



33.9.2.1 Byte Write Access

Byte write access supports one byte write signal per byte of the data bus and a single read signal.

Note that the SMC does not allow boot in Byte Write Access mode.

- For 16-bit devices: the SMC provides NWR0 and NWR1 write signals for respectively byte0 (lower byte) and byte1 (upper byte) of a 16-bit bus. One single read signal (NRD) is provided.
  Byte Write Access is used to connect 2 x 8-bit devices as a 16-bit memory.
- For 32-bit devices: NWR0, NWR1, NWR2 and NWR3, are the write signals of byte0 (lower byte), byte1, byte2 and byte 3 (upper byte) respectively. One single read signal (NRD) is provided.
  Byte Write Access is used to connect 4 x 8-bit devices as a 32-bit memory.

  The Byte Write option is illustrated in Figure 33-6.

33.9.2.2 Byte Select Access

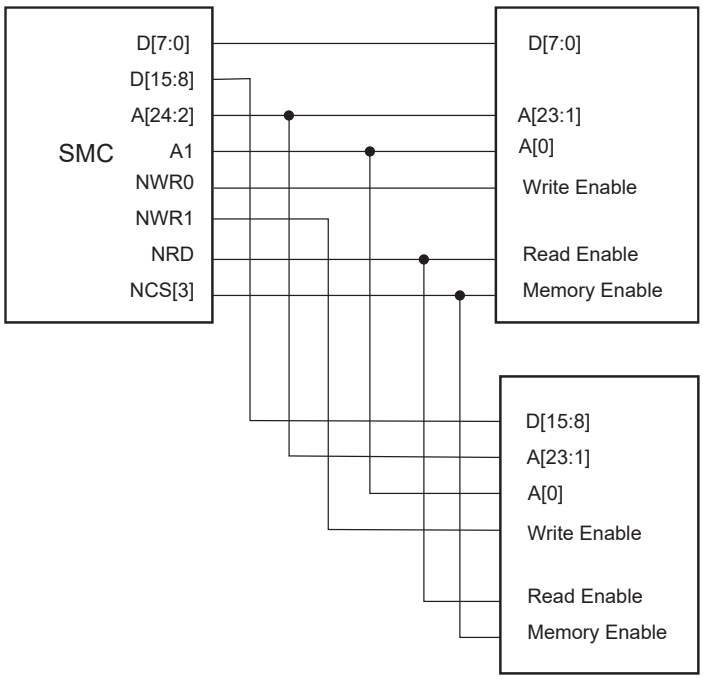
In this mode, read/write operations can be enabled/disabled at a byte level. One byte-select line per byte of the data bus is provided. One NRD and one NWE signal control read and write.

Downloaded from Arrow.com.

- For 16-bit devices: the SMC provides NBS0 and NBS1 selection signals for respectively byte0 (lower byte) and byte1 (upper byte) of a 16-bit bus.
  Byte Select Access is used to connect one 16-bit device.
- For 32-bit devices: NBS0, NBS1, NBS2 and NBS3, are the selection signals of byte0 (lower byte), byte1, byte2 and byte 3 (upper byte) respectively. Byte Select Access is used to connect two 16-bit devices.

Figure 33-7 shows how to connect two 16-bit devices on a 32-bit data bus in Byte Select Access mode, on NCS3 (BAT = Byte Select Access).

Figure 33-6. Connection of 2 x 8-bit Devices on a 16-bit Bus: Byte Write Option



33.9.2.3 Signal Multiplexing

Depending on the byte access type (BAT), only the write signals or the byte select signals are used. To save IOs at the external bus interface, control signals at the SMC interface are multiplexed. The table below shows signal multiplexing depending on the data bus width and the byte access type.

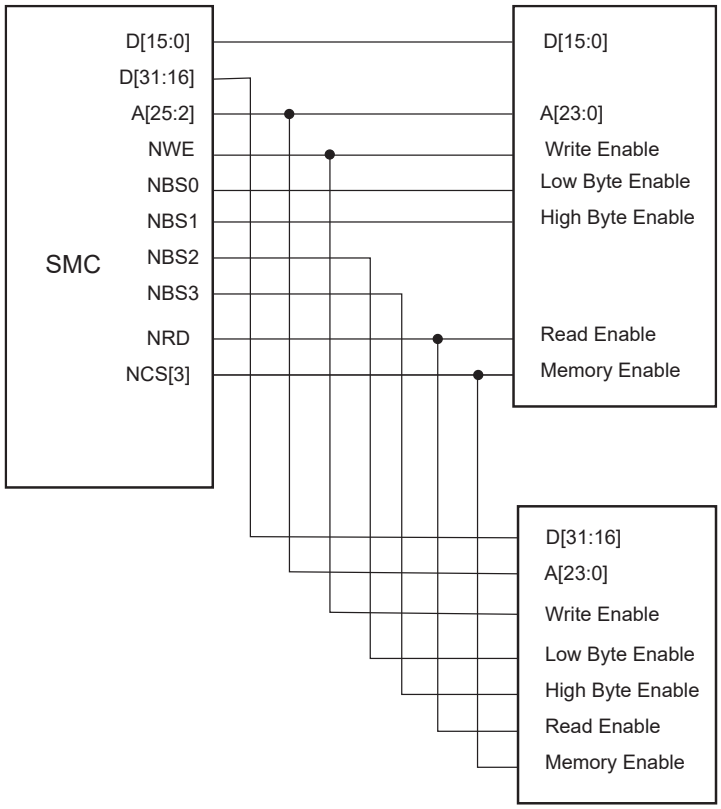
Table 33-3. SMC Multiplexed Signal Translation

Signal Name	32-bit Bus			16-bit Bus		8-bit Bus
Device Type	1 x 32-bit	2 x 16-bit	4 x 8-bit	1 x 16-bit	2 x 8-bit	1 x 8-bit
Byte Access Type (BAT)	Byte Select	Byte Select	Byte Write	Byte Select	Byte Write	–
NBS0_A0	NBS0	NBS0	–	NBS0	–	A0
NWE_NWR0	NWE	NWE	NWR0	NWE	NWR0	NWE
NBS1_NWR1	NBS1	NBS1	NWR1	NBS1	NWR1	–
NBS2_NWR2_A1	NBS2	NBS2	NWR2	A1	A1	A1
NBS3_NWR3	NBS3	NBS3	NWR3	–	–	–

For 32-bit devices, bits A0 and A1 are unused. For 16-bit devices, bit A0 of address is unused. When the Byte Select option is selected, NWR1 to NWR3 are unused. When the Byte Write option is selected, NBS0 to NBS3 are unused.

Downloaded from Arrow.com.

Figure 33-7. Connection of 2x16-bit Data Bus on a 32-bit Data Bus (Byte Select Option)



33.10 Standard Read and Write Protocols

In the following sections, the byte access type is not considered. Byte select lines (NBS0 to NBS3) always have the same timing as the A address bus. NWE represents either the NWE signal in byte select access type or one of the byte write lines (NWR0 to NWR3) in byte write access type. NWR0 to NWR3 have the same timings and protocol as NWE. In the same way, NCS represents one of the NCS[0..5] chip select lines.

33.10.1 Read Waveforms

The following figure shows the read cycle. The read cycle starts with the address setting on the memory address bus:

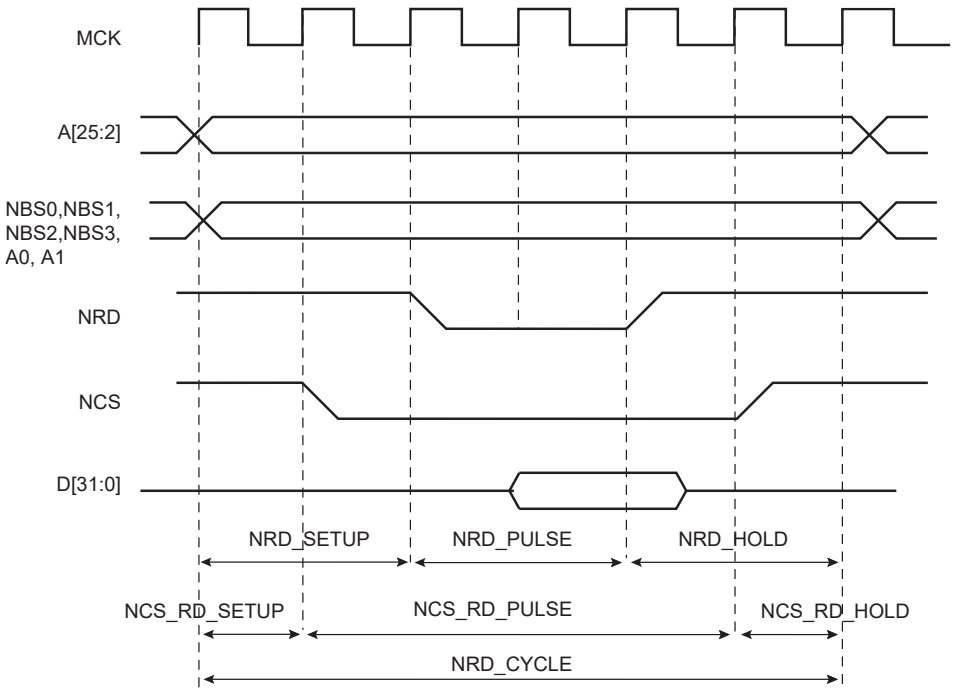
{A[25:2], A1, A0} for 8-bit devices

{A[25:2], A1} for 16-bit devices

A[25:2] for 32-bit devices

Downloaded from Arrow.com.

Figure 33-8. Standard Read Cycle



33.10.1.1 NRD Waveform

The NRD signal is characterized by a setup timing, a pulse width and a hold timing:

- NRD_SETUP—NRD setup time is defined as the setup of address before the NRD falling edge.
- NRD_PULSE—NRD pulse length is the time between NRD falling edge and NRD rising edge.
- NRD_HOLD—NRD hold time is defined as the hold time of address after the NRD rising edge.

33.10.1.2 NCS Waveform

Similar to the NRD signal, the NCS signal can be divided into a setup time, pulse length and hold time:

- NCS_RD_SETUP—NCS setup time is defined as the setup time of address before the NCS falling edge.
- NCS_RD_PULSE—NCS pulse length is the time between NCS falling edge and NCS rising edge;
- NCS_RD_HOLD—NCS hold time is defined as the hold time of address after the NCS rising edge.

33.10.1.3 Read Cycle

The NRD_CYCLE time is defined as the total duration of the read cycle, that is, from the time where address is set on the address bus to the point where address may change. The total read cycle time is defined as:

- NRD_CYCLE = NRD_SETUP + NRD_PULSE + NRD_HOLD

as well as

- NRD_CYCLE = NCS_RD_SETUP + NCS_RD_PULSE + NCS_RD_HOLD

All NRD and NCS timings are defined separately for each chip select as an integer number of Master Clock cycles. The NRD_CYCLE field is common to both the NRD and NCS signals, thus the timing period is of the same duration.

NRD_CYCLE, NRD_SETUP, and NRD_PULSE implicitly define the NRD_HOLD value as:

- NRD_HOLD = NRD_CYCLE - NRD SETUP - NRD PULSE

NRD_CYCLE, NCS_RD_SETUP, and NCS_RD_PULSE implicitly define the NCS_RD_HOLD value as:

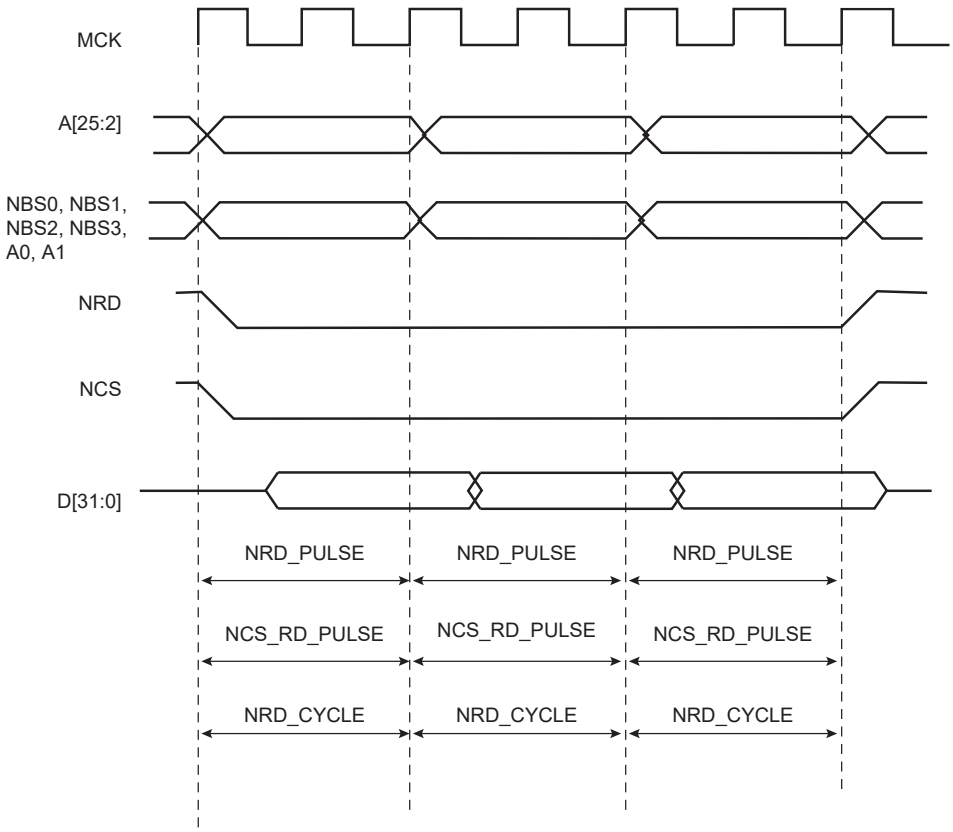
- NCS_RD_HOLD = NRD_CYCLE - NCS_RD_SETUP - NCS_RD_PULSE

33.10.1.4 Null Delay Setup and Hold

If null setup and hold parameters are programmed for NRD and/or NCS, NRD and NCS remain active continuously in case of consecutive read cycles in the same memory. See the following figure.

Downloaded from Arrow.com.

Figure 33-9. No Setup, No Hold On NRD and NCS Read Signals



33.10.1.5 Null Pulse

Programming a null pulse is not permitted. Pulse must be at least set to 1. A null value leads to unpredictable behavior.

33.10.2 Read Mode

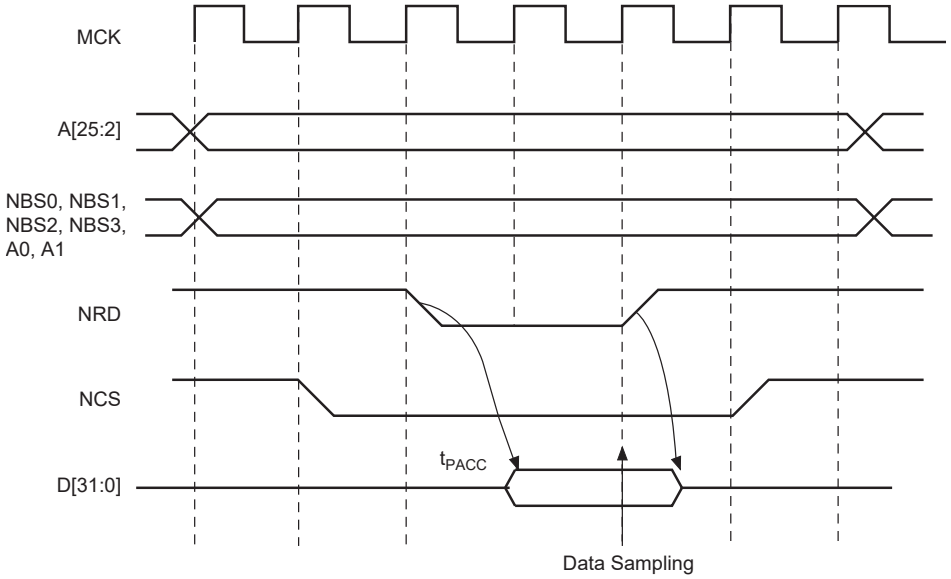
As NCS and NRD waveforms are defined independently of one other, the SMC needs to know when the read data is available on the data bus. The SMC does not compare NCS and NRD timings to know which signal rises first. The READ_MODE parameter in the SMC_MODE register of the corresponding chip select indicates which signal of NRD and NCS controls the read operation.

33.10.2.1 Read is Controlled by NRD (READ_MODE = 1)

The following figure shows the waveforms of a read operation of a typical asynchronous RAM. The read data is available t_{PACC} after the falling edge of NRD, and turns to ‘Z’ after the rising edge of NRD. In this case, the READ_MODE must be set to 1 (read is controlled by NRD), to indicate that data is available with the rising edge of NRD. The SMC samples the read data internally on the rising edge of Master Clock that generates the rising edge of NRD, whatever the programmed waveform of NCS may be.

Downloaded from Arrow.com.

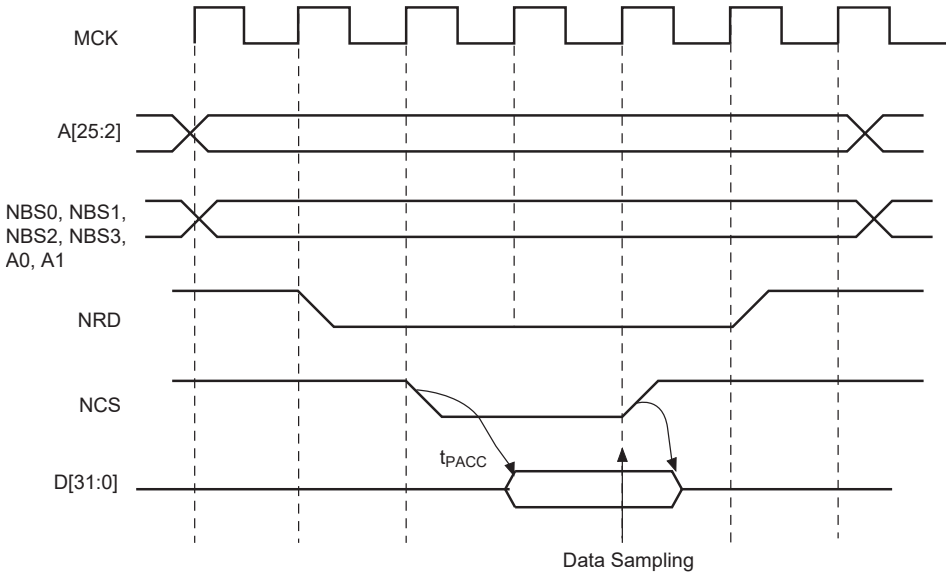
Figure 33-10. READ_MODE = 1 (Data sampled by SMC before rising edge of NRD)



33.10.2.2 Read is Controlled by NCS (READ_MODE = 0)

The following figure shows the typical read cycle of an LCD module. The read data is valid t_{PACC} after the falling edge of the NCS signal and remains valid until the rising edge of NCS. Data must be sampled when NCS is raised. In that case, the READ_MODE must be set to 0 (read is controlled by NCS): the SMC internally samples the data on the rising edge of Master Clock that generates the rising edge of NCS, whatever the programmed waveform of NRD may be.

Figure 33-11. READ_MODE = 0 (Data sampled by SMC before rising edge of NCS)



33.10.3 Write Waveforms

The write protocol (depicted in Figure 33-12) is similar to the read protocol. The write cycle starts with the address setting on the memory address bus.

33.10.3.1 NWE Waveforms

The NWE signal is characterized by a setup timing, a pulse width and a hold timing.

- NWE_SETUP—NWE setup time is defined as the setup of address and data before the NWE falling edge.

Downloaded from Arrow.com.

- NWE_PULSE—NWE pulse length is the time between NWE falling edge and NWE rising edge.
- NWE_HOLD—NWE hold time is defined as the hold time of address and data after the NWE rising edge.

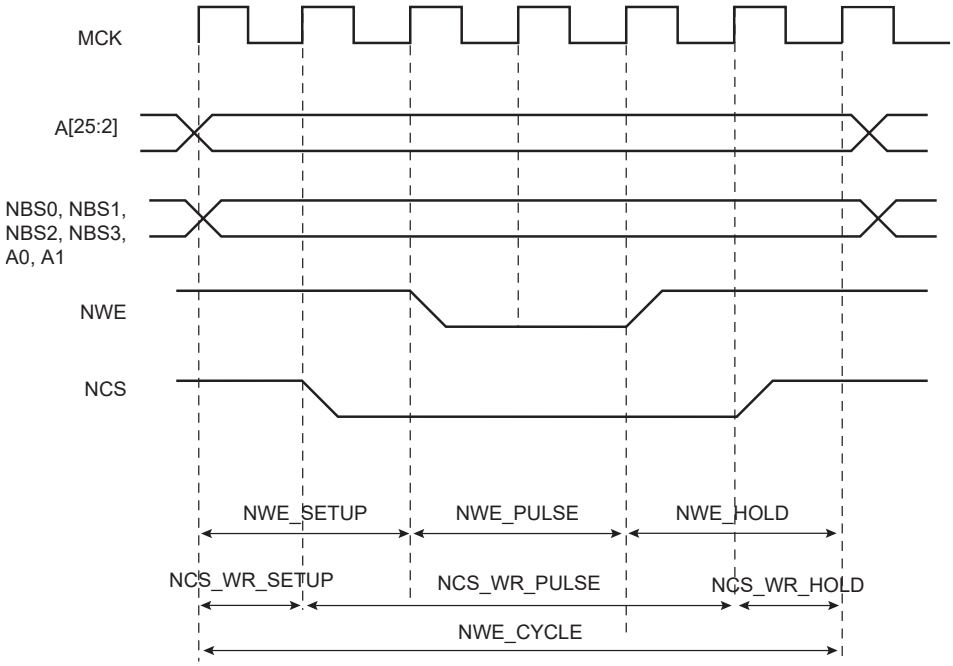
The NWE waveforms apply to all byte-write lines in Byte Write Access mode: NWR0 to NWR3.

33.10.3.2 NCS Waveforms

The NCS signal waveforms in write operation are not the same that those applied in read operations, but are separately defined:

- NCS_WR_SETUP—NCS setup time is defined as the setup time of address before the NCS falling edge.
- NCS_WR_PULSE—NCS pulse length is the time between NCS falling edge and NCS rising edge.
- NCS_WR_HOLD—NCS hold time is defined as the hold time of address after the NCS rising edge.

Figure 33-12. Write Cycle



33.10.3.3 Write Cycle

The write_cycle time is defined as the total duration of the write cycle, that is, from the time where address is set on the address bus to the point where address may change. The total write cycle time is defined as:

- NWE_CYCLE = NWE_SETUP + NWE_PULSE + NWE_HOLD

as well as

- NWE_CYCLE = NCS_WR_SETUP + NCS_WR_PULSE + NCS_WR_HOLD

All NWE and NCS (write) timings are defined separately for each chip select as an integer number of Master Clock cycles. The NWE_CYCLE field is common to both the NWE and NCS signals, thus the timing period is of the same duration.

NWE_CYCLE, NWE_SETUP, and NWE_PULSE implicitly define the NWE_HOLD value as:

- NWE_HOLD = NWE_CYCLE - NWE_SETUP - NWE_PULSE

NWE_CYCLE, NCS_WR_SETUP, and NCS_WR_PULSE implicitly define the NCS_WR_HOLD value as:

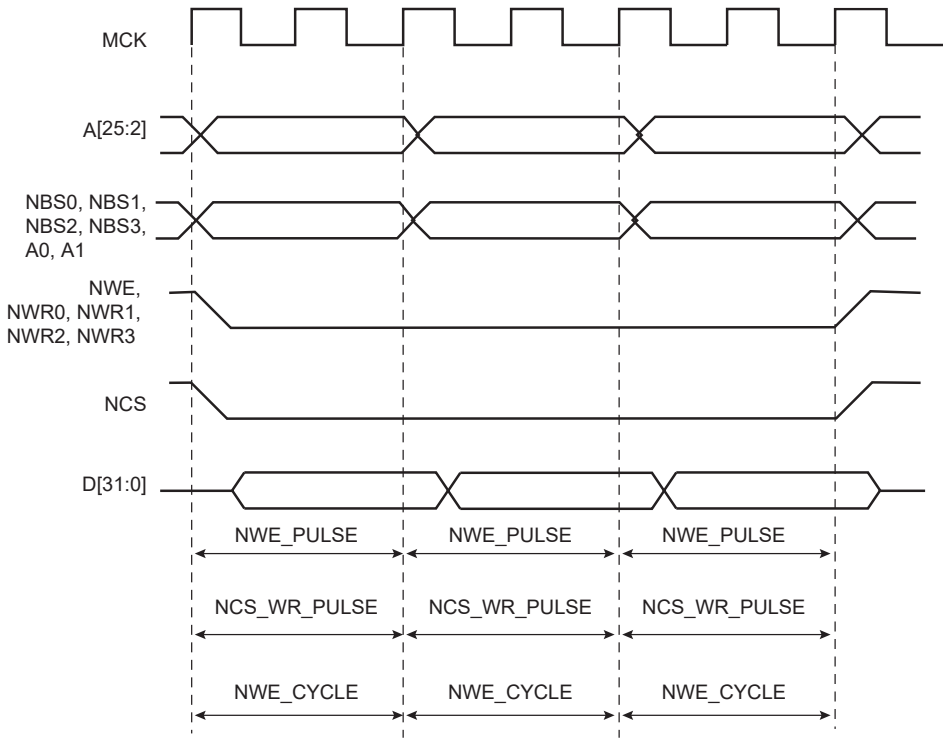
- NCS_WR_HOLD = NWE_CYCLE - NCS_WR_SETUP - NCS_WR_PULSE

33.10.3.4 Null Delay Setup and Hold

If null setup parameters are programmed for NWE and/or NCS, NWE and/or NCS remain active continuously in case of consecutive write cycles in the same memory (see the following figure). However, for devices that perform write operations on the rising edge of NWE or NCS, such as SRAM, either a setup or a hold must be programmed.

Downloaded from Arrow.com.

Figure 33-13. Null Setup and Hold Values of NCS and NWE in Write Cycle



33.10.3.5 Null Pulse

Programming a null pulse is not permitted. Pulse must be at least set to 1. A null value leads to unpredictable behavior.

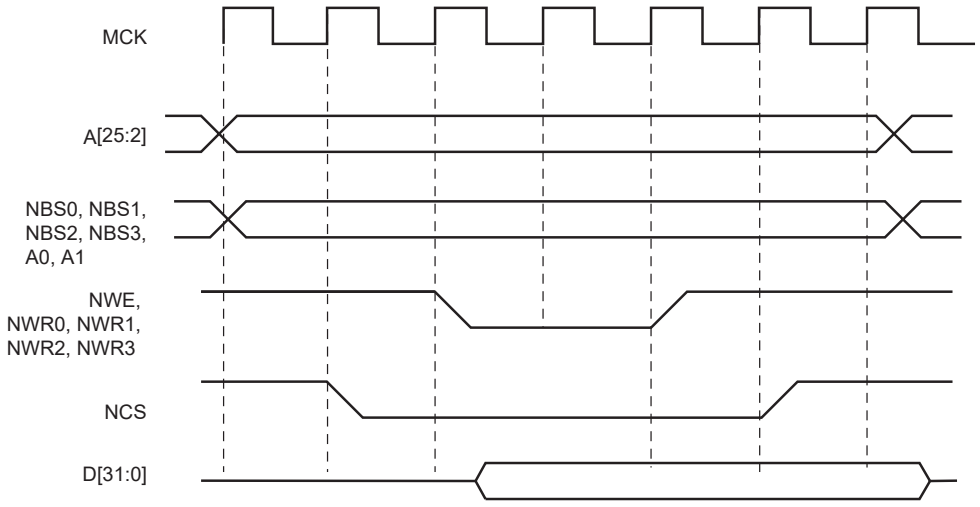
33.10.4 Write Mode

The WRITE_MODE parameter in the SMC_MODE register of the corresponding chip select indicates which signal controls the write operation.

33.10.4.1 Write is Controlled by NWE (WRITE_MODE = 1)

The following figure shows the waveforms of a write operation with WRITE_MODE set to 1. The data is put on the bus during the pulse and hold steps of the NWE signal. The internal data buffers are switched to Output mode after the NWE_SETUP time, and until the end of the write cycle, regardless of the programmed waveform on NCS.

Figure 33-14. WRITE_MODE = 1 (Write Operation Controlled by NWE)

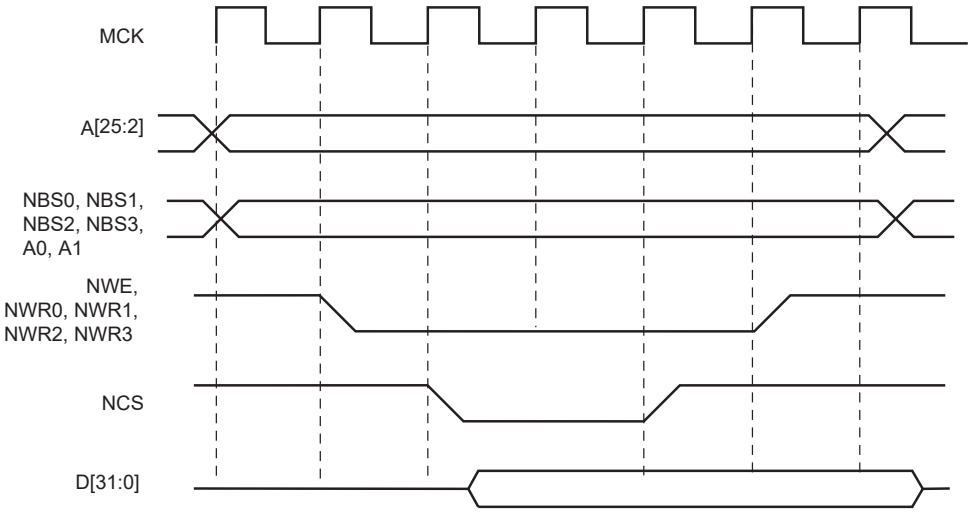


Downloaded from Arrow.com.

33.10.4.2 Write is Controlled by NCS (WRITE_MODE = 0)

The following figure shows the waveforms of a write operation with WRITE_MODE set to 0. The data is put on the bus during the pulse and hold steps of the NCS signal. The internal data buffers are switched to Output mode after the NCS_WR_SETUP time, and until the end of the write cycle, regardless of the programmed waveform on NWE.

Figure 33-15. WRITE_MODE = 0 (Write Operation Controlled by NCS)



33.10.5 Coding Timing Parameters

All timing parameters are defined for one chip select and are grouped together in one SMC_REGISTER according to their type.

The SMC_SETUP register groups the definition of all setup parameters:

- NRD_SETUP, NCS_RD_SETUP, NWE_SETUP, NCS_WR_SETUP

The SMC_PULSE register groups the definition of all pulse parameters:

- NRD_PULSE, NCS_RD_PULSE, NWE_PULSE, NCS_WR_PULSE

The SMC_CYCLE register groups the definition of all cycle parameters:

- NRD_CYCLE, NWE_CYCLE

The following table shows how the timing parameters are coded and their permitted range.

Table 33-4. Coding and Range of Timing Parameters

Coded Value	Number of Bits	Effective Value	Permitted Range	
			Coded Value	Effective Value
setup [5:0]	6	128 x setup[5] + setup[4:0]	0 ≤ 31	0 ≤ 128 + 31
pulse [6:0]	7	256 x pulse[6] + pulse[5:0]	0 ≤ 63	0 ≤ 256 + 63
cycle [8:0]	9	256 x cycle[8:7] + cycle[6:0]	0 ≤ 127	0 ≤ 256 + 127 0 ≤ 512 +1 27 0 ≤ 768 + 127

33.10.6 Reset Values of Timing Parameters

For the default values of timing parameters at reset, see 33.20.1 SMC_SETUPx, 33.20.2 SMC_PULSEx and 33.20.3 SMC_CYCLEx.

Downloaded from Arrow.com.

33.10.7 Usage Restriction

The SMC does not check the validity of the user-programmed parameters. If the sum of SETUP and PULSE parameters is larger than the corresponding CYCLE parameter, this leads to unpredictable behavior of the SMC.

- For read operations:
  Null but positive setup and hold of address and NRD and/or NCS can not be guaranteed at the memory interface because of the propagation delay of theses signals through external logic and pads. If positive setup and hold values must be verified, then it is strictly recommended to program non-null values so as to cover possible skews between address, NCS and NRD signals.
- For write operations:
  If a null hold value is programmed on NWE, the SMC can guarantee a positive hold of address, byte select lines, and NCS signal after the rising edge of NWE. This is true for WRITE_MODE = 1 only. See Early Read Wait State.
- For read and write operations:
  A null value for pulse parameters is forbidden and may lead to unpredictable behavior.

  In read and write cycles, the setup and hold time parameters are defined in reference to the address bus. For external devices that require setup and hold time between NCS and NRD signals (read), or between NCS and NWE signals (write), these setup and hold times must be converted into setup and hold times in reference to the address bus.

33.11 Automatic Wait States

Under certain circumstances, the SMC automatically inserts idle cycles between accesses to avoid bus contention or operation conflict.

33.11.1 Chip Select Wait States

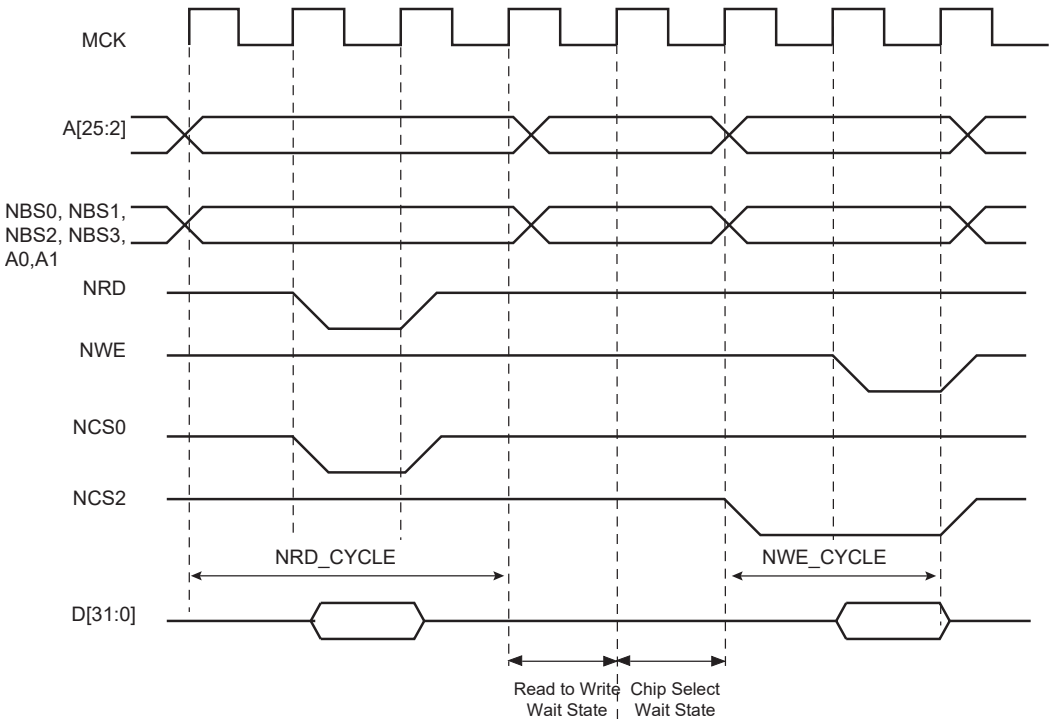
The SMC always inserts an idle cycle between two transfers on separate chip selects. This idle cycle ensures that there is no bus contention between the de-activation of one device and the activation of the next one.

During chip select wait state, all control lines are turned inactive: NBS0 to NBS3, NWR0 to NWR3, NCS[0..5], NRD lines are all set to 1.

The following figure illustrates a chip select wait state between access on Chip Select 0 and Chip Select 2.

Downloaded from Arrow.com.

Figure 33-16. Chip Select Wait State between a Read Access on NCS0 and a Write Access on NCS2



33.11.2 Early Read Wait State

In some cases, the SMC inserts a wait state cycle between a write access and a read access to allow time for the write cycle to end before the subsequent read cycle begins. This wait state is not generated in addition to a chip select wait state. The early read cycle thus only occurs between a write and read access to the same memory device (same chip select).

An early read wait state is automatically inserted if at least one of the following conditions is valid:

- the write controlling signal has no hold time and the read controlling signal has no setup time (Figure 33-17).
- in NCS Write Controlled mode (WRITE_MODE = 0), there is no hold timing on the NCS signal and the NCS_RD_SETUP parameter is set to 0, regardless of the read mode (Figure 33-18). The write operation must end with a NCS rising edge. Without an Early Read Wait State, the write operation could not complete properly.
- in NWE Controlled mode (WRITE_MODE = 1) and if there is no hold timing (NWE_HOLD = 0), the feedback of the write control signal is used to control address, data, chip select and byte select lines. If the external write control signal is not inactivated as expected due to load capacitances, an Early Read Wait State is inserted and address, data and control signals are maintained one more cycle. See Figure 33-19.

Downloaded from Arrow.com.

Figure 33-17. Early Read Wait State: Write with No Hold Followed by Read with No Setup

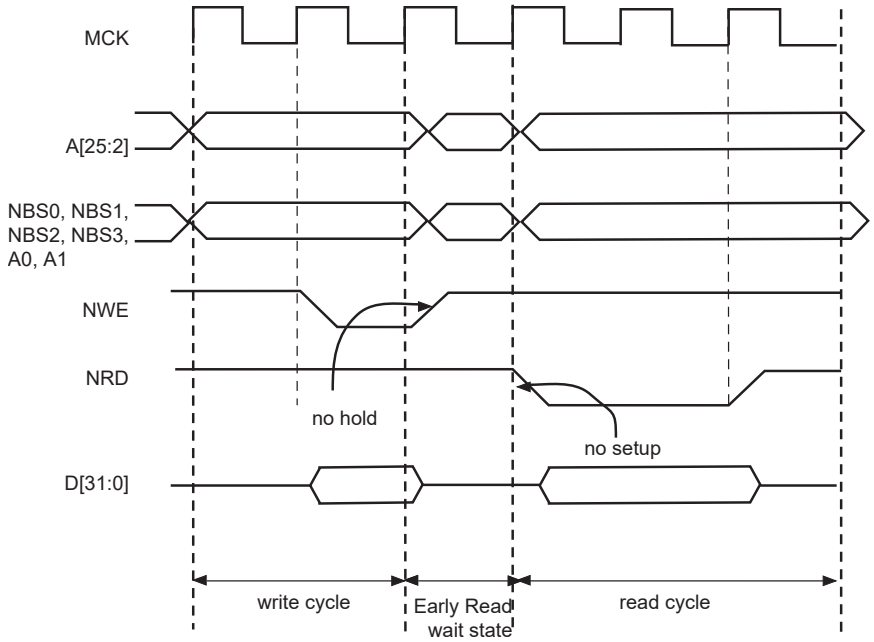
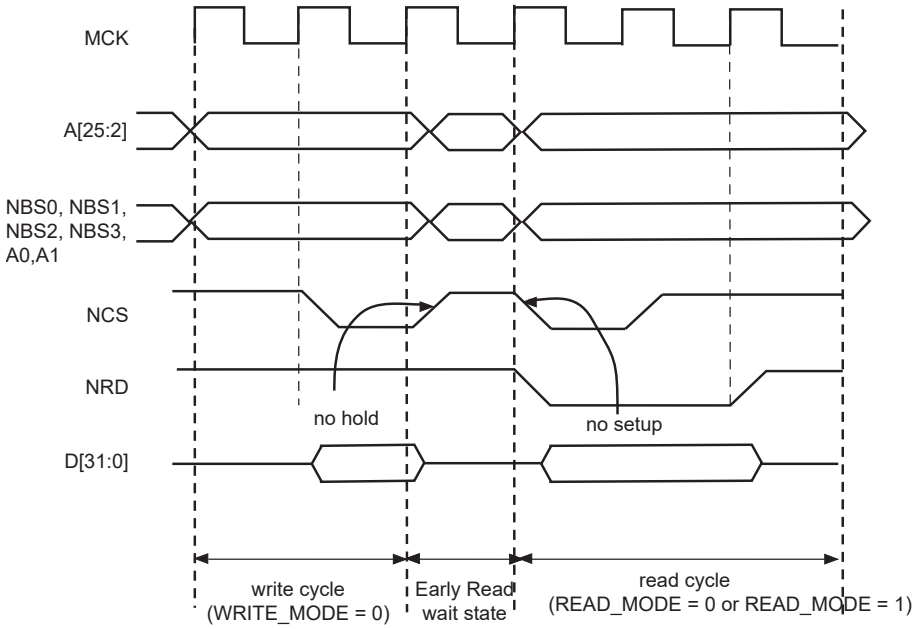
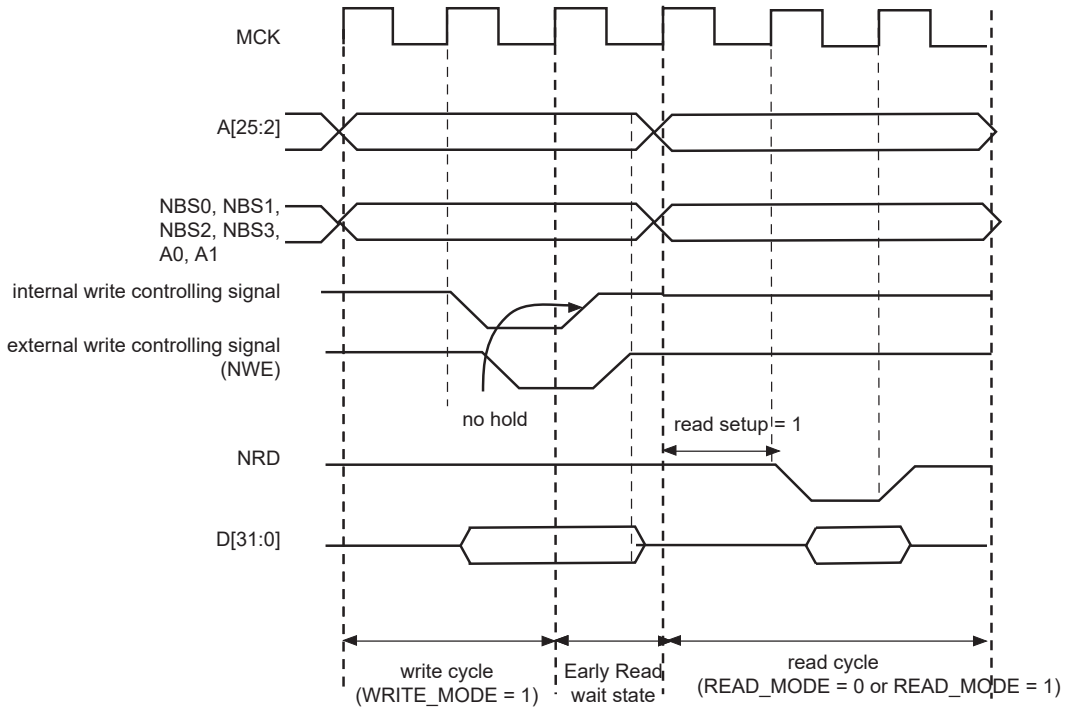


Figure 33-18. Early Read Wait State: NCS Controlled Write with No Hold Followed by a Read with No NCS Setup



Downloaded from Arrow.com.

Figure 33-19. Early Read Wait State: NWE-controlled Write with No Hold Followed by a Read with one Set-up Cycle



33.11.3 Reload User Configuration Wait State

The user may change any of the configuration parameters by writing the SMC user interface.

When detecting that a new user configuration has been written in the user interface, the SMC inserts a wait state before starting the next access. The so called “Reload User Configuration Wait State” is used by the SMC to load the new set of parameters to apply to next accesses.

The Reload Configuration Wait State is not applied in addition to the Chip Select Wait State. If accesses before and after re-programming the user interface are made to different devices (Chip Selects), then one single Chip Select Wait State is applied.

On the other hand, if accesses before and after writing the user interface are made to the same device, a Reload Configuration Wait State is inserted, even if the change does not concern the current Chip Select.

33.11.3.1 User Procedure

To insert a Reload Configuration Wait State, the SMC detects a write access to any SMC_MODE register of the user interface. If the user only modifies timing registers (SMC_SETUP, SMC_PULSE, SMC_CYCLE registers) in the user interface, he must validate the modification by writing the SMC_MODE, even if no change was made on the mode parameters.

The user must not change the configuration parameters of an SMC Chip Select (Setup, Pulse, Cycle, Mode) if accesses are performed on this CS during the modification. Any change of the Chip Select parameters, while fetching the code from a memory connected on this CS, may lead to unpredictable behavior. The instructions used to modify the parameters of an SMC Chip Select can be executed from the internal RAM or from a memory connected to another CS.

33.11.3.2 Slow Clock Mode Transition

A Reload Configuration Wait State is also inserted when the Slow Clock mode is entered or exited, after the end of the current transfer (see 33.14 Slow Clock Mode).

33.11.4 Read to Write Wait State

Due to an internal mechanism, a wait cycle is always inserted between consecutive read and write SMC accesses.

Downloaded from Arrow.com.

This wait cycle is referred to as a read to write wait state in this document.

This wait cycle is applied in addition to chip select and reload user configuration wait states when they are to be inserted. See Figure 33-16.

33.12 Data Float Wait States

Some memory devices are slow to release the external bus. For such devices, it is necessary to add wait states (data float wait states) after a read access:

- before starting a read access to a different external memory
- before starting a write access to the same device or to a different external one.

The Data Float Output Time (t_{DF}) for each external memory device is programmed in the TDF_CYCLES field of the SMC_MODE register for the corresponding chip select. The value of TDF_CYCLES indicates the number of data float wait cycles (between 0 and 15) before the external device releases the bus, and represents the time allowed for the data output to go to high impedance after the memory is disabled.

Data float wait states do not delay internal memory accesses. Hence, a single access to an external memory with long t_{DF} will not slow down the execution of a program from internal memory.

The data float wait states management depends on the READ_MODE and the TDF_MODE fields of the SMC_MODE register for the corresponding chip select.

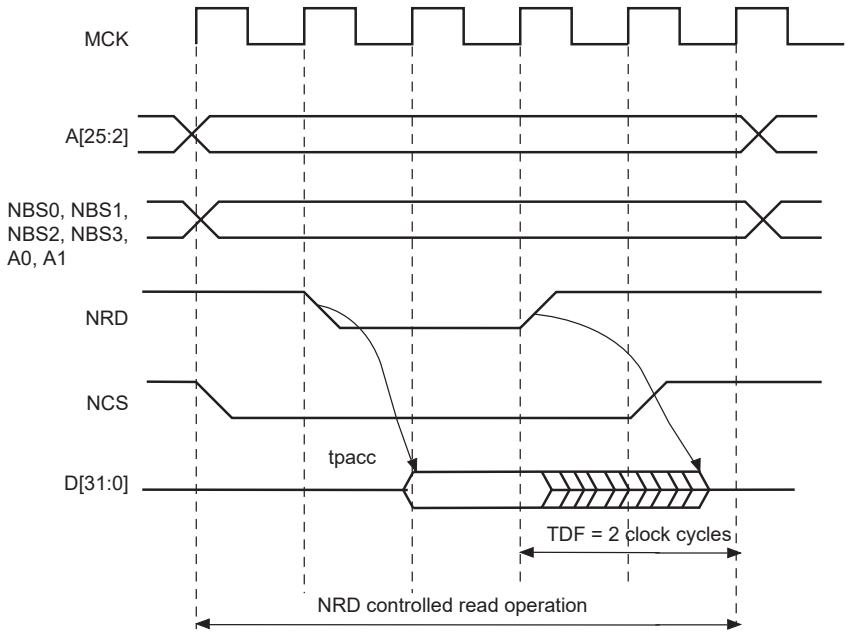
33.12.1 READ_MODE

Setting the READ_MODE to 1 indicates to the SMC that the NRD signal is responsible for turning off the tri-state buffers of the external memory device. The Data Float Period then begins after the rising edge of the NRD signal and lasts TDF_CYCLES MCK cycles.

When the read operation is controlled by the NCS signal (READ_MODE = 0), the TDF field gives the number of MCK cycles during which the data bus remains busy after the rising edge of NCS.

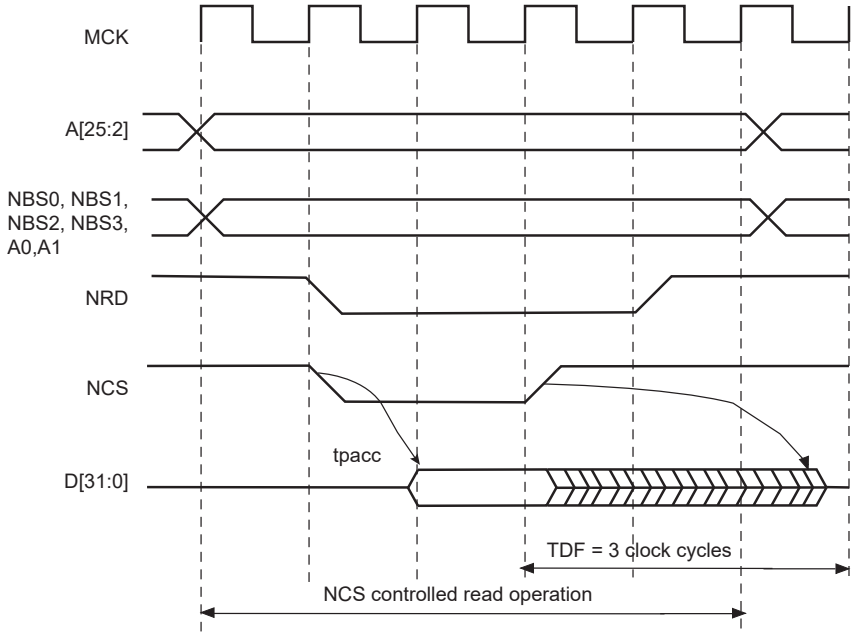
Figure 33-20 illustrates the Data Float Period in NRD-controlled mode (READ_MODE = 1), assuming a data float period of two cycles (TDF_CYCLES = 2). Figure 33-21 shows the read operation when controlled by NCS (READ_MODE = 0) and the TDF_CYCLES parameter equals 3.

Figure 33-20. TDF Period in NRD Controlled Read Access (TDF = 2)



Downloaded from Arrow.com.

Figure 33-21. TDF Period in NCS Controlled Read Operation (TDF = 3)



33.12.2 TDF Optimization Enabled (TDF_MODE = 1)

When the TDF_MODE of the SMC_MODE register is set to 1 (TDF optimization is enabled), the SMC takes advantage of the setup period of the next access to optimize the number of wait states cycle to insert.

The following figure shows a read access controlled by NRD, followed by a write access controlled by NWE, on Chip Select 0. Chip Select 0 has been programmed with:

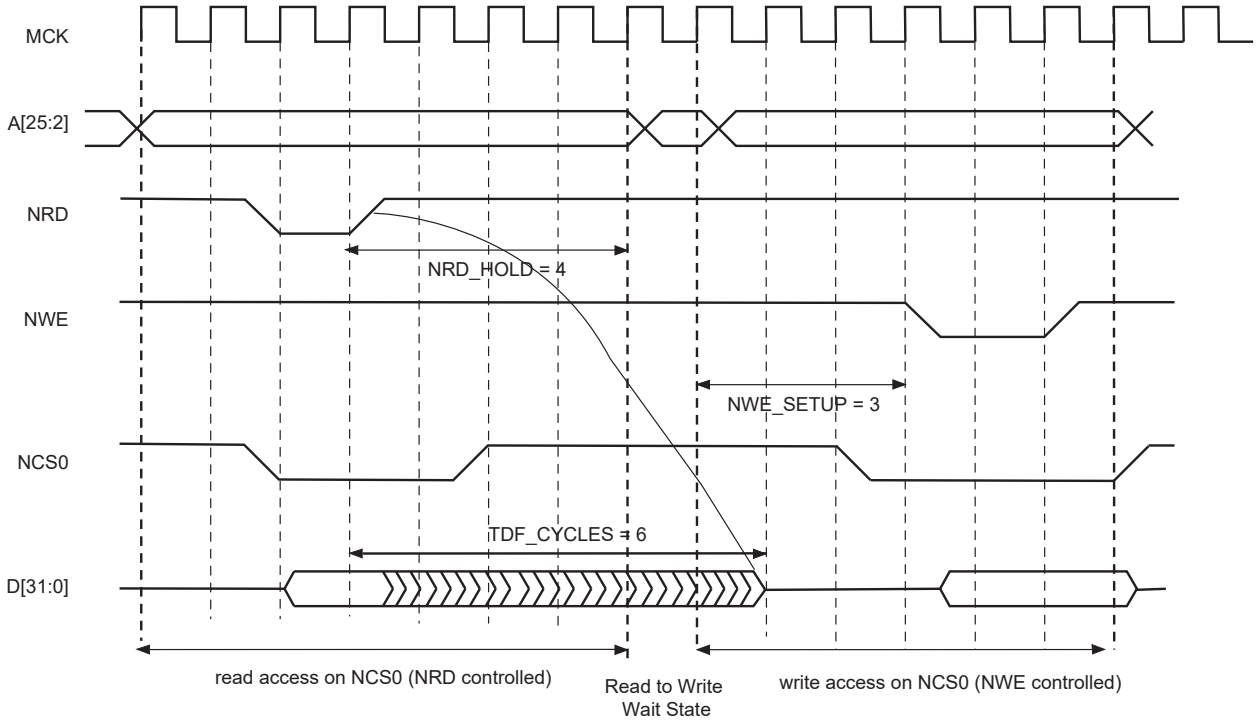
NRD_HOLD = 4; READ_MODE = 1 (NRD controlled)

NWE_SETUP = 3; WRITE_MODE = 1 (NWE controlled)

TDF_CYCLES = 6; TDF_MODE = 1 (optimization enabled).

Downloaded from Arrow.com.

Figure 33-22. TDF Optimization: No TDF wait states are inserted if the TDF period is over when the next access begins



33.12.3 TDF Optimization Disabled (TDF_MODE = 0)

When optimization is disabled, TDF wait states are inserted at the end of the read transfer, so that the data float period is ended when the second access begins. If the hold period of the read1 controlling signal overlaps the data float period, no additional TDF wait states will be inserted.

The three figures below illustrate the following cases, respectively:

- read access followed by a read access on another chip select,
- read access followed by a write access on another chip select,
- read access followed by a write access on the same chip select,

with no TDF optimization.

Downloaded from Arrow.com.

Figure 33-23. TDF Optimization Disabled (TDF Mode = 0): TDF wait states between 2 read accesses on different chip selects

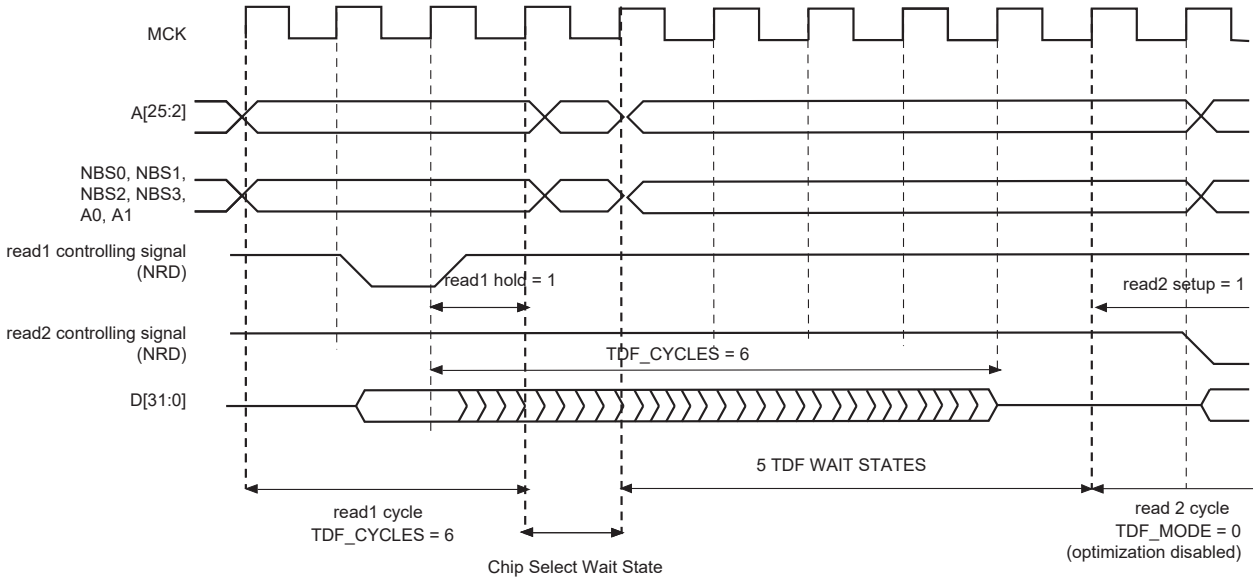
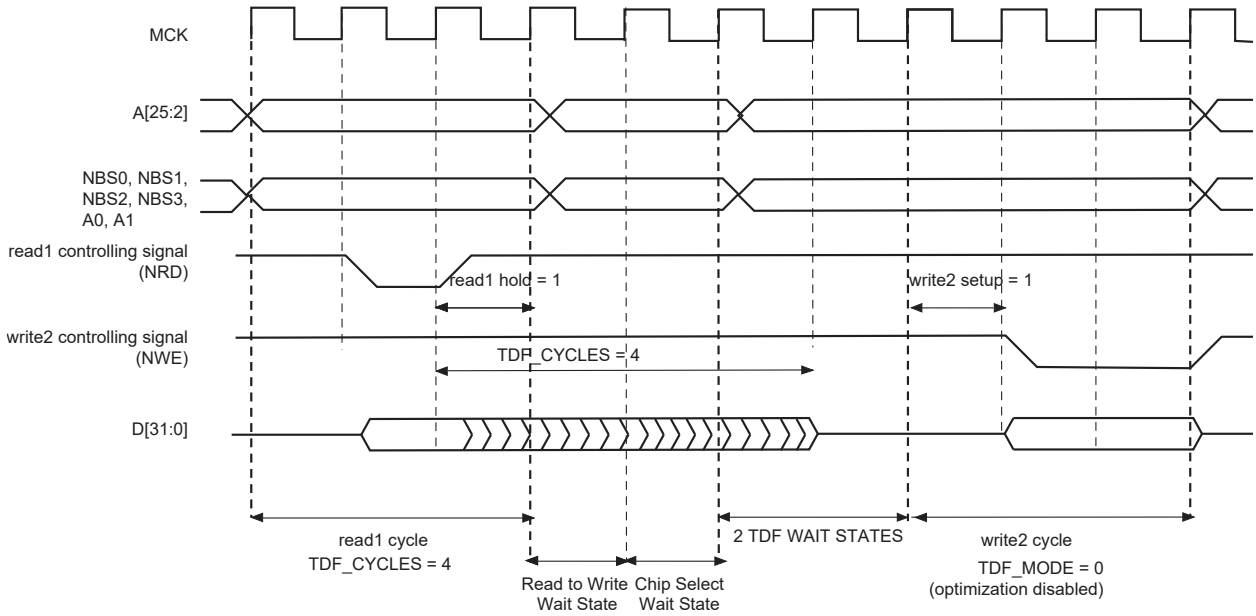
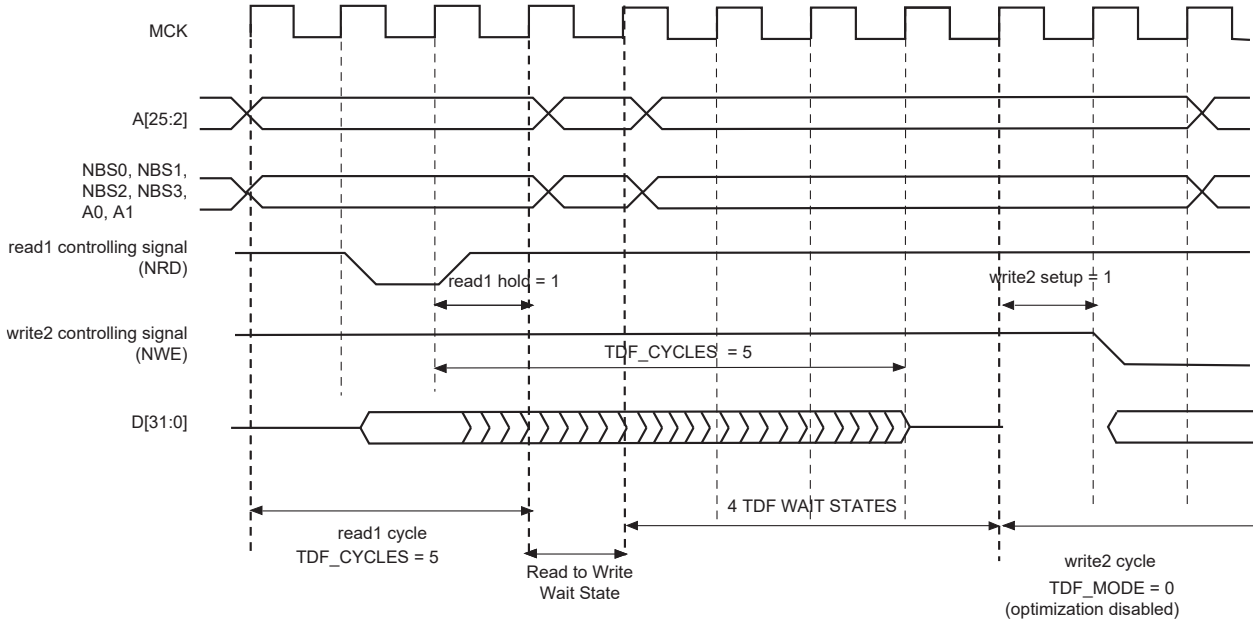


Figure 33-24. TDF Mode = 0: TDF wait states between a read and a write access on different chip selects



Downloaded from Arrow.com.

Figure 33-25. TDF Mode = 0: TDF wait states between read and write accesses on the same chip select



33.13 External Wait

Any access can be extended by an external device using the NWAIT input signal of the SMC. The EXNW_MODE field of the SMC_MODE register on the corresponding chip select must be set to either to ‘10’ (frozen mode) or ‘11’ (ready mode). When the EXNW_MODE is set to ‘00’ (disabled), the NWAIT signal is simply ignored on the corresponding chip select. The NWAIT signal delays the read or write operation in regards to the read or write controlling signal, depending on the read and write modes of the corresponding chip select.

33.13.1 Restriction

When one of the EXNW_MODE is enabled, it is mandatory to program at least one hold cycle for the read/write controlling signal. For that reason, the NWAIT signal cannot be used in Page mode (see 33.15 Asynchronous Page Mode), or in Slow Clock mode (see 33.14 Slow Clock Mode).

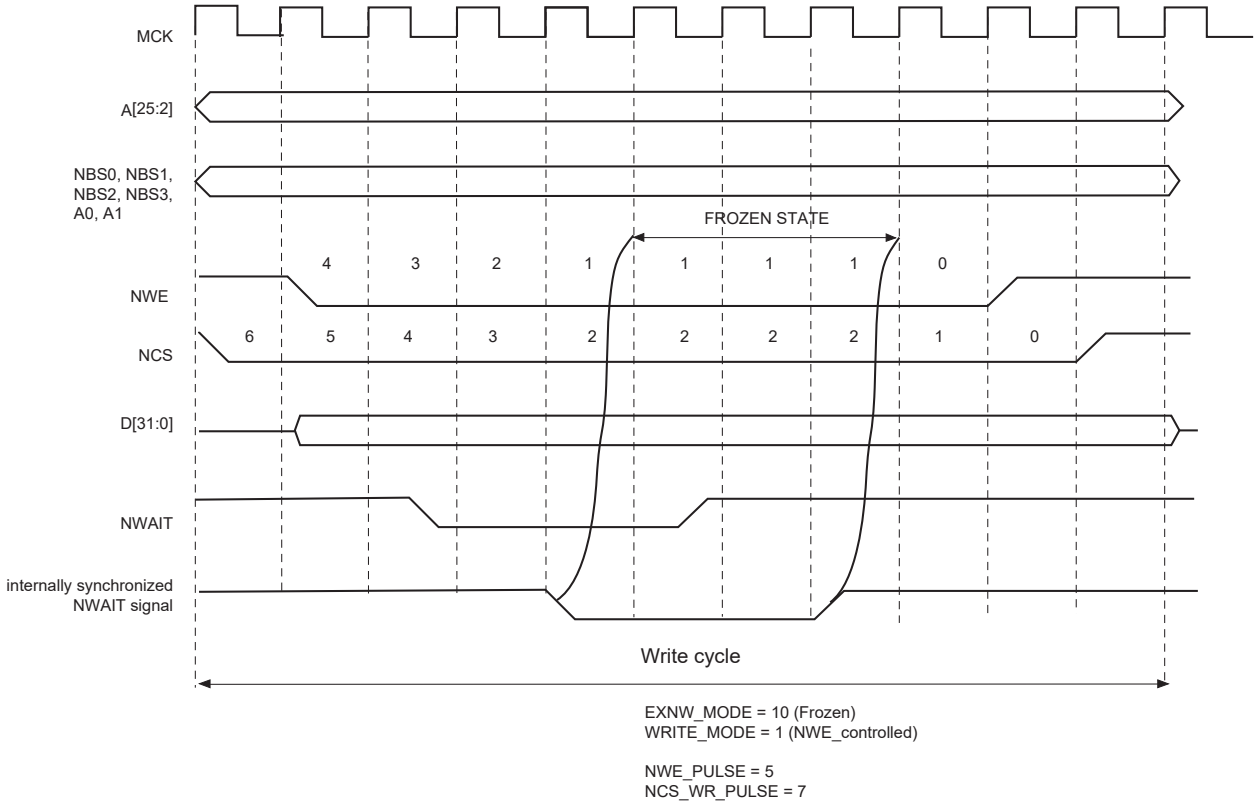
The NWAIT signal is assumed to be a response of the external device to the read/write request of the SMC. Then NWAIT is examined by the SMC only in the pulse state of the read or write controlling signal. The assertion of the NWAIT signal outside the expected period has no impact on SMC behavior.

33.13.2 Frozen Mode

When the external device asserts the NWAIT signal (active low), and after internal synchronization of this signal, the SMC state is frozen, i.e., SMC internal counters are frozen, and all control signals remain unchanged. When the resynchronized NWAIT signal is deasserted, the SMC completes the access, resuming the access from the point where it was stopped. See the following figure. This mode must be selected when the external device uses the NWAIT signal to delay the access and to freeze the SMC.

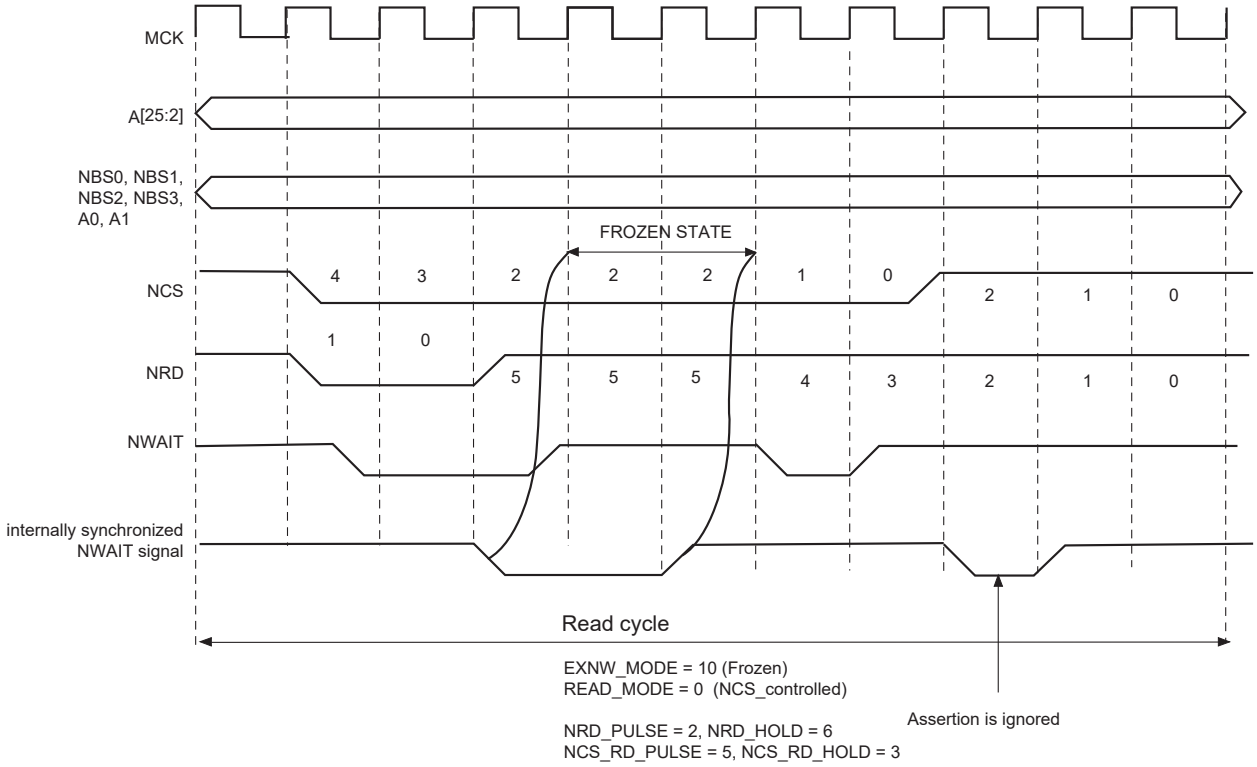
Downloaded from Arrow.com.

Figure 33-26. Write Access with NWAIT Assertion in Frozen Mode (EXNW_MODE = 10)



The assertion of the NWAIT signal outside the expected period is ignored as illustrated in the figure below.

Figure 33-27. Read Access with NWAIT Assertion in Frozen Mode (EXNW_MODE = 10)



Downloaded from Arrow.com.

33.13.3 Ready Mode

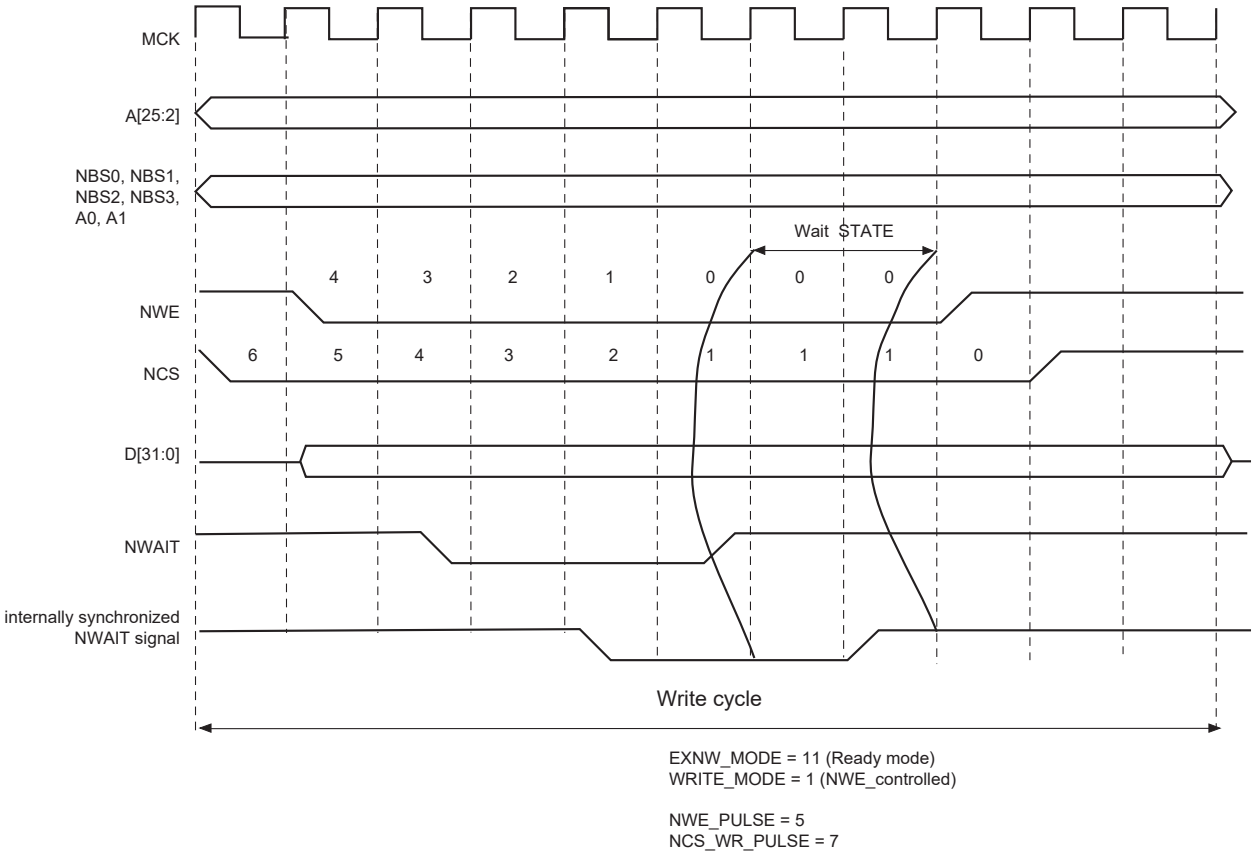
In Ready mode (EXNW_MODE = 11), the SMC behaves differently. Normally, the SMC begins the access by down counting the setup and pulse counters of the read/write controlling signal. In the last cycle of the pulse phase, the resynchronized NWAIT signal is examined.

If asserted, the SMC suspends the access as shown in Figure 33-28 and Figure 33-29. After deassertion, the access is completed: the hold step of the access is performed.

This mode must be selected when the external device uses deassertion of the NWAIT signal to indicate its ability to complete the read or write operation.

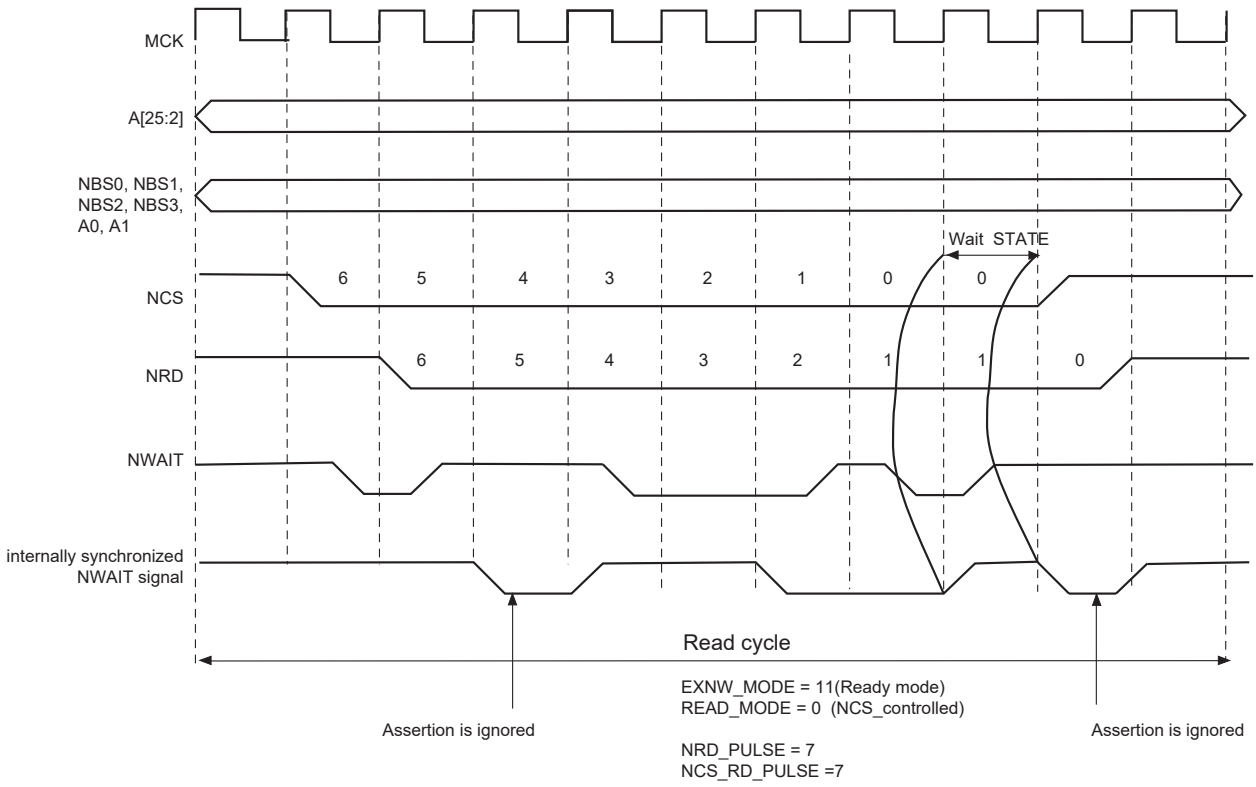
If the NWAIT signal is deasserted before the end of the pulse, or asserted after the end of the pulse of the controlling read/write signal, it has no impact on the access length as shown in Figure 33-29.

Figure 33-28. NWAIT Assertion in Write Access: Ready Mode (EXNW_MODE = 11)



Downloaded from Arrow.com.

Figure 33-29. NWAIT Assertion in Read Access: Ready Mode (EXNW_MODE = 11)



33.13.4 NWAIT Latency and Read/Write Timings

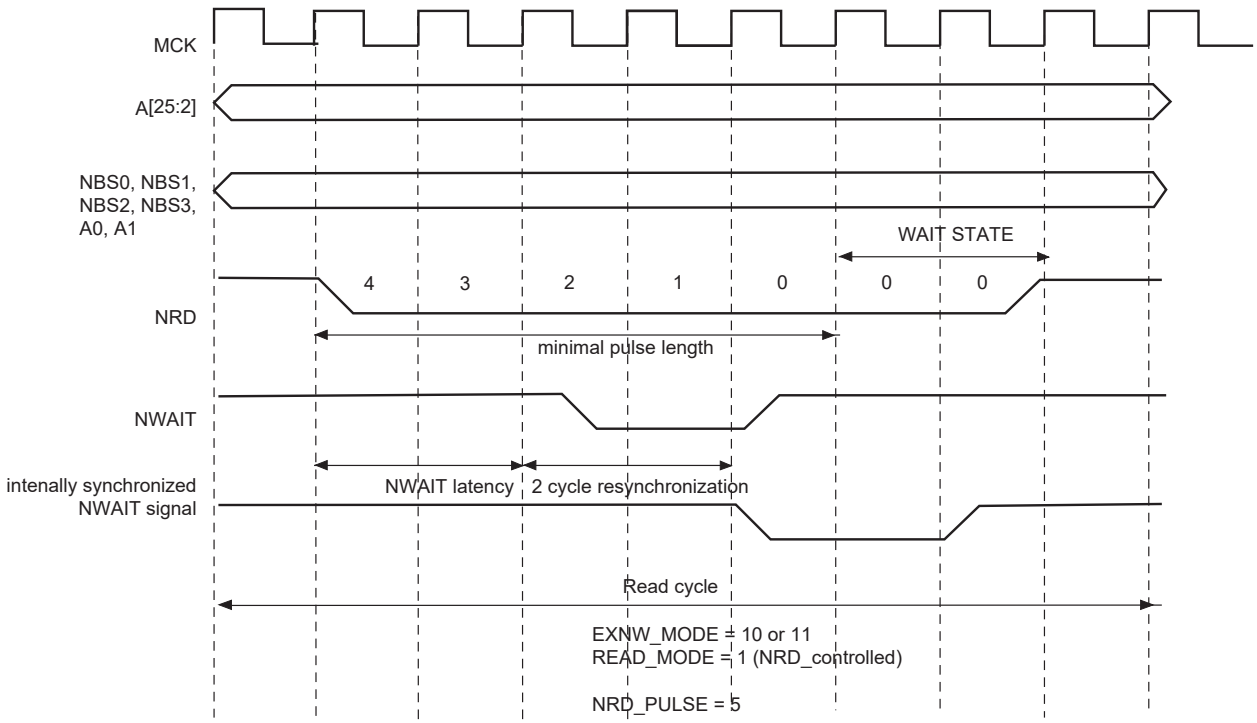
There may be a latency between the assertion of the read/write controlling signal and the assertion of the NWAIT signal by the device. The programmed pulse length of the read/write controlling signal must be at least equal to this latency plus the 2 cycles of resynchronization + 1 cycle. Otherwise, the SMC may enter the hold state of the access without detecting the NWAIT signal assertion. This is true in frozen mode as well as in ready mode. This is illustrated in the following figure.

When EXNW_MODE is enabled (ready or frozen), the user must program a pulse length of the read and write controlling signal of at least:

minimal pulse length = NWAIT latency + 2 resynchronization cycles + 1 cycle

Downloaded from Arrow.com.

Figure 33-30. NWAIT Latency



33.14 Slow Clock Mode

The SMC is able to automatically apply a set of “slow clock mode” read/write waveforms when an internal signal driven by the Power Management Controller is asserted because MCK has been configured to a very slow clock rate (typically 32 kHz clock rate). In this mode, the user-programmed waveforms are ignored and the slow clock mode waveforms are applied. This mode is provided so as to avoid reprogramming the User Interface with appropriate waveforms at very slow clock rate. When activated, the slow mode is active on all chip selects.

33.15 Asynchronous Page Mode

The SMC supports asynchronous burst reads in page mode, providing that the page mode is enabled in the SMC_MODE register (PMEN field). The page size must be configured in the SMC_MODE register (PS field) to 4, 8, 16 or 32 bytes.

The page defines a set of consecutive bytes into memory. A 4-byte page (resp. 8-, 16-, 32-byte page) is always aligned to 4-byte boundaries (resp. 8-, 16-, 32-byte boundaries) of memory. The MSB of data address defines the address of the page in memory, the LSB of address define the address of the data in the page as detailed in the table below.

With page mode memory devices, the first access to one page (t_{pa}) takes longer than the subsequent accesses to the page (t_{sa}) as shown in Figure 33-31. When in page mode, the SMC enables the user to define different read timings for the first access within one page, and next accesses within the page.

Table 33-5. Page Address and Data Address within a Page

Page Size	Page Address (1)	Data Address in the Page (2)
4 bytes	A[25:2]	A[1:0]
8 bytes	A[25:3]	A[2:0]
16 bytes	A[25:4]	A[3:0]

Downloaded from Arrow.com.

.....continued		
Page Size	**Page Address (1)**	**Data Address in the Page (2)**
32 bytes	A[25:5]	A[4:0]

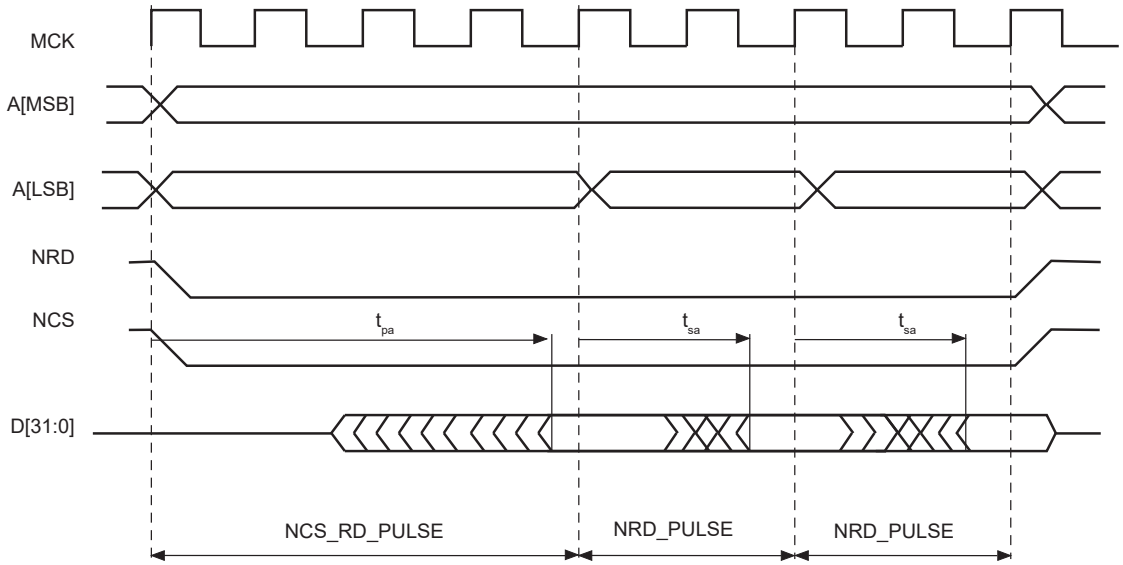
Note:

1. 'A' denotes the address bus of the memory device.
2. For 16-bit devices, bit 0 of the address is ignored. For 32-bit devices, bits [1:0] are ignored.

33.15.1 Protocol and Timings in Page Mode

The following figure shows the NRD and NCS timings in Page mode access. Note that address MSB and LSB are defined in Table 33-5.

Figure 33-31. Page Mode Read Protocol



The NRD and NCS signals are held low during all read transfers, whatever the programmed values of the setup and hold timings in the User Interface may be. Moreover, the NRD and NCS timings are identical. The pulse length of the first access to the page is defined with the NCS_RD_PULSE field of the SMC_PULSE register. The pulse length of subsequent accesses within the page are defined using the NRD_PULSE parameter.

Programming of the read timings in Page mode is described in the following table.

Table 33-6. Programming of Read Timings in Page Mode

Parameter	Value	Definition
READ_MODE	'x'	No impact
NCS_RD_SETUP	'x'	No impact
NCS_RD_PULSE	t_{pa}	Access time of first access to the page
NRD_SETUP	'x'	No impact
NRD_PULSE	t_{sa}	Access time of subsequent accesses in the page
NRD_CYCLE	'x'	No impact

The SMC does not check the coherency of timings. It will always apply the NCS_RD_PULSE timings as page access timing (t_{pa}) and the NRD_PULSE for accesses to the page (t_{sa}), even if the programmed value for t_{pa} is shorter than the programmed value for t_{sa}.

Downloaded from Arrow.com.

33.15.2 Byte Access Type in Page Mode

The byte access type (BAT) configuration remains active in page mode. For 16-bit or 32-bit page mode devices that require byte selection signals, write a 0 to the BAT bit in the SMC Mode Register (SMC_MODE) to select the byte select access type.

33.15.3 Page Mode Restriction

The page mode is not compatible with the use of the NWAIT signal. Using the page mode and the NWAIT signal may lead to unpredictable behavior.

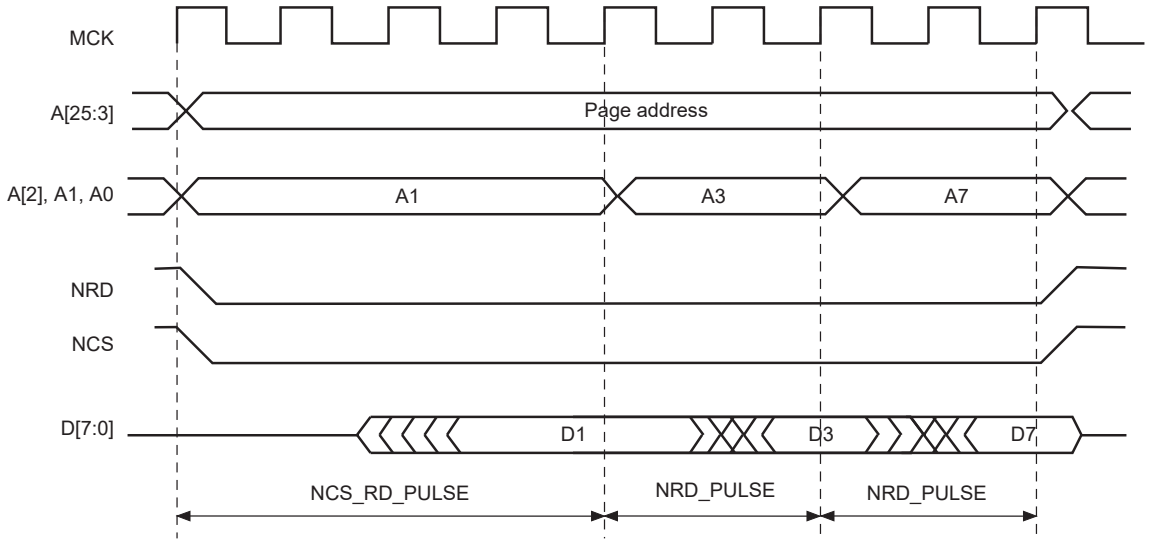
33.15.4 Sequential and Non-sequential Accesses

If the chip select and the MSB of addresses as defined in Table 33-5 are identical, then the current access lies in the same page as the previous one, and no page break occurs.

Using this information, all data within the same page, sequential or not sequential, are accessed with a minimum access time (t_{sa}). The following figure illustrates access to an 8-bit memory device in page mode, with 8-byte pages. Access to D1 causes a page access with a long access time (t_{pa}). Accesses to D3 and D7, though they are not sequential accesses, only require a short access time (t_{sa}).

If the MSB of addresses are different, the SMC performs the access of a new page. In the same way, if the chip select is different from the previous access, a page break occurs. If two sequential accesses are made to the page mode memory, but separated by an other internal or external peripheral access, a page break occurs on the second access because the chip select of the device was deasserted between both accesses.

Figure 33-32. Access to Non-sequential Data within the Same Page



33.16 Register Write Protection

To prevent any single software error from corrupting SMC behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the SMC Write Protection Mode Register (SMC_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the SMC Write Protection Status Register (SMC_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the SMC_WPSR.

The following registers can be write-protected:

- SMC Setup Register
- SMC Pulse Register
- SMC Cycle Register
- SMC Mode Register

Downloaded from Arrow.com.

- SMC Off-Chip Memory Scrambling Register
- SMC Safety Report Interrupt Enable Register

33.17 Security and Safety Analysis and Reports

Several types of checks are performed when the SMC is reading or writing an external memory. The internal sequencer of the SMC is monitored for integrity and if an abnormal state is detected, the flag SMC_WPSR.SEQE is set. This flag is not set under normal operating conditions. The software accesses to the SMC are monitored and if an incorrect access is performed, the flag SMC_WPSR.SWE is set. The type of incorrect/abnormal software access is reported in the SMC_WPSR.SWETYP field (see SMC Write Protection Status Register for details). The flags SEQE, SWE and WPVS are automatically cleared when SMC_WPSR is read. If one of these flags is set, an interrupt can be triggered if the SMC_SRIER.SRIE bit is ‘1’.

33.18 Scrambling/Unscrambling Function

The external data bus can be scrambled to make more difficult the recovery of intellectual property data located in off-chip memories by means of data analysis at the package pin level of either the microcontroller or the memory device.

The scrambling and unscrambling are performed on-the-fly without additional wait states.

The scrambling/unscrambling function can be enabled or disabled by configuring the CSxSE bits in the SMC Off-Chip Memory Scrambling Register (SMC_OCMS).

When multiple chip selects are handled, the scrambling function per chip select is configurable using the CSxSE bits in the SMC_OCMS register.

The scrambling method depends on two user-configurable key registers, SMC_KEY1 and SMC_KEY2 plus a random value depending on device processing characteristics. These key registers cannot be read. They can be written once after a system reset.

The scrambling user key or the seed for key generation must be securely stored in a reliable non-volatile memory in order to recover data from the off-chip memory. Any data scrambled with a given key cannot be recovered if the key is lost.

33.19 Clearing Scrambling Keys on a Tamper Event

On tamper detection event on WKUP pins, it is possible to perform an immediate clear of the scrambling keys (SMC_KEY1 and SMC_KEY2) is performed if SMC_OCMS.TAMPCLR is set.

Downloaded from Arrow.com.

33.20 Register Summary

The SMC is programmed using the registers listed below. For each chip select, a set of four registers is used to program the parameters of the external device connected on it. The number of SMC_SETUP, SMC_PULSE, SMC_CYCLE and SMC_MODE registers depends on the number of chip selects. Sixteen bytes (0x10) are required per chip select.

Note: The user must confirm the SMC configuration by writing any one of the SMC_MODE registers.

Offset	Name	Bit Pos.								
0x00	SMC_SETUP0	31:24			NCS_RD_SETUP[5:0]					
		23:16			NRD_SETUP[5:0]					
		15:8			NCS_WR_SETUP[5:0]					
		7:0			NWE_SETUP[5:0]					
0x04	SMC_PULSE0	31:24		NCS_RD_PULSE[6:0]						
		23:16		NRD_PULSE[6:0]						
		15:8		NCS_WR_PULSE[6:0]						
		7:0		NWE_PULSE[6:0]						
0x08	SMC_CYCLE0	31:24								NRD_CYCLE[8]
		23:16	NRD_CYCLE[7:0]							
		15:8								NWE_CYCLE[8]
		7:0	NWE_CYCLE[7:0]							
0x0C	SMC_MODE0	31:24			PS[1:0]					PMEN
		23:16				TDF_MODE	TDF_CYCLES[3:0]			
		15:8			DBW[1:0]					BAT
		7:0			EXNW_MODE[1:0]				WRITE_MODE	READ_MODE
0x10	SMC_SETUP1	31:24			NCS_RD_SETUP[5:0]					
		23:16			NRD_SETUP[5:0]					
		15:8			NCS_WR_SETUP[5:0]					
		7:0			NWE_SETUP[5:0]					
0x14	SMC_PULSE1	31:24		NCS_RD_PULSE[6:0]						
		23:16		NRD_PULSE[6:0]						
		15:8		NCS_WR_PULSE[6:0]						
		7:0		NWE_PULSE[6:0]						
0x18	SMC_CYCLE1	31:24								NRD_CYCLE[8]
		23:16	NRD_CYCLE[7:0]							
		15:8								NWE_CYCLE[8]
		7:0	NWE_CYCLE[7:0]							
0x1C	SMC_MODE1	31:24			PS[1:0]					PMEN
		23:16				TDF_MODE	TDF_CYCLES[3:0]			
		15:8			DBW[1:0]					BAT
		7:0			EXNW_MODE[1:0]				WRITE_MODE	READ_MODE
0x20	SMC_SETUP2	31:24			NCS_RD_SETUP[5:0]					
		23:16			NRD_SETUP[5:0]					
		15:8			NCS_WR_SETUP[5:0]					
		7:0			NWE_SETUP[5:0]					
0x24	SMC_PULSE2	31:24		NCS_RD_PULSE[6:0]						
		23:16		NRD_PULSE[6:0]						
		15:8		NCS_WR_PULSE[6:0]						
		7:0		NWE_PULSE[6:0]						

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x28	SMC_CYCLE2	31:24								NRD_CYCLE[8]
		23:16	NRD_CYCLE[7:0]							
		15:8								NWE_CYCLE[8]
		7:0	NWE_CYCLE[7:0]							
0x2C	SMC_MODE2	31:24			PS[1:0]					PMEN
		23:16				TDF_MODE	TDF_CYCLES[3:0]			
		15:8			DBW[1:0]					BAT
		7:0			EXNW_MODE[1:0]				WRITE_MODE	READ_MODE
0x30	SMC_SETUP3	31:24			NCS_RD_SETUP[5:0]					
		23:16			NRD_SETUP[5:0]					
		15:8			NCS_WR_SETUP[5:0]					
		7:0			NWE_SETUP[5:0]					
0x34	SMC_PULSE3	31:24		NCS_RD_PULSE[6:0]						
		23:16		NRD_PULSE[6:0]						
		15:8		NCS_WR_PULSE[6:0]						
		7:0		NWE_PULSE[6:0]						
0x38	SMC_CYCLE3	31:24								NRD_CYCLE[8]
		23:16	NRD_CYCLE[7:0]							
		15:8								NWE_CYCLE[8]
		7:0	NWE_CYCLE[7:0]							
0x3C	SMC_MODE3	31:24			PS[1:0]					PMEN
		23:16				TDF_MODE	TDF_CYCLES[3:0]			
		15:8			DBW[1:0]					BAT
		7:0			EXNW_MODE[1:0]				WRITE_MODE	READ_MODE
0x40	SMC_SETUP4	31:24			NCS_RD_SETUP[5:0]					
		23:16			NRD_SETUP[5:0]					
		15:8			NCS_WR_SETUP[5:0]					
		7:0			NWE_SETUP[5:0]					
0x44	SMC_PULSE4	31:24		NCS_RD_PULSE[6:0]						
		23:16		NRD_PULSE[6:0]						
		15:8		NCS_WR_PULSE[6:0]						
		7:0		NWE_PULSE[6:0]						
0x48	SMC_CYCLE4	31:24								NRD_CYCLE[8]
		23:16	NRD_CYCLE[7:0]							
		15:8								NWE_CYCLE[8]
		7:0	NWE_CYCLE[7:0]							
0x4C	SMC_MODE4	31:24			PS[1:0]					PMEN
		23:16				TDF_MODE	TDF_CYCLES[3:0]			
		15:8			DBW[1:0]					BAT
		7:0			EXNW_MODE[1:0]				WRITE_MODE	READ_MODE
0x50	SMC_SETUP5	31:24			NCS_RD_SETUP[5:0]					
		23:16			NRD_SETUP[5:0]					
		15:8			NCS_WR_SETUP[5:0]					
		7:0			NWE_SETUP[5:0]					
0x54	SMC_PULSE5	31:24		NCS_RD_PULSE[6:0]						
		23:16		NRD_PULSE[6:0]						
		15:8		NCS_WR_PULSE[6:0]						
		7:0		NWE_PULSE[6:0]						

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x58	SMC_CYCLE5	31:24								NRD_CYCLE[8]
		23:16	NRD_CYCLE[7:0]							
		15:8								NWE_CYCLE[8]
		7:0	NWE_CYCLE[7:0]							
0x5C	SMC_MODE5	31:24			PS[1:0]					PMEN
		23:16				TDF_MODE	TDF_CYCLES[3:0]			
		15:8			DBW[1:0]					BAT
		7:0			EXNW_MODE[1:0]				WRITE_MODE	READ_MODE
0x60 ... 0x7F	Reserved									
0x80	SMC_OCMS	31:24								
		23:16								
		15:8			CS5SE	CS4SE	CS3SE	CS2SE	CS1SE	CS0SE
		7:0				TAMPCLR				SMSE
0x84	SMC_KEY1	31:24	KEY1[31:24]							
		23:16	KEY1[23:16]							
		15:8	KEY1[15:8]							
		7:0	KEY1[7:0]							
0x88	SMC_KEY2	31:24	KEY2[31:24]							
		23:16	KEY2[23:16]							
		15:8	KEY2[15:8]							
		7:0	KEY2[7:0]							
0x8C ... 0x8F	Reserved									
0x90	SMC_SRIER	31:24								
		23:16								
		15:8								
		7:0								SRIE
0x94 ... 0xE3	Reserved									
0xE4	SMC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0xE8	SMC_WPSR	31:24							SWETYP[1:0]	
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE		WPVS

Downloaded from Arrow.com.

33.20.1 SMC Setup Register

Name: SMC_SETUPx
Offset: 0x00 + x*0x10 [x=0..5]
Reset: 0x01010101
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SMC Write Protection Mode Register.

Note: The number of SMC_SETUP registers depends on the chip select number.

Bit	31	30	29	28	27	26	25	24
			NCS_RD_SETUP[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	1

Bit	23	22	21	20	19	18	17	16
			NRD_SETUP[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
			NCS_WR_SETUP[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
			NWE_SETUP[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	1

Bits 29:24 – NCS_RD_SETUP[5:0] NCS Setup Length in READ Access
In read access, the NCS signal setup length is defined as:
NCS setup length = (128 * NCS_RD_SETUP[5] + NCS_RD_SETUP[4:0]) clock cycles

Bits 21:16 – NRD_SETUP[5:0] NRD Setup Length
The NRD signal setup length is defined in clock cycles as:
NRD setup length = (128 * NRD_SETUP[5] + NRD_SETUP[4:0]) clock cycles

Bits 13:8 – NCS_WR_SETUP[5:0] NCS Setup Length in WRITE Access
In write access, the NCS signal setup length is defined as:
NCS setup length = (128 * NCS_WR_SETUP[5] + NCS_WR_SETUP[4:0]) clock cycles

Bits 5:0 – NWE_SETUP[5:0] NWE Setup Length
The NWE signal setup length is defined as:
NWE setup length = (128 * NWE_SETUP[5] + NWE_SETUP[4:0]) clock cycles

Downloaded from Arrow.com.

33.20.2 SMC Pulse Register

Name: SMC_PULSEx
Offset: 0x04 + x*0x10 [x=0..5]
Reset: 0x01010101
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SMC Write Protection Mode Register.

Note: The number of SMC_PULSE registers depends on the chip select number.

Bit	31	30	29	28	27	26	25	24
		NCS_RD_PULSE[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	1

Bit	23	22	21	20	19	18	17	16
		NRD_PULSE[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
		NCS_WR_PULSE[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
		NWE_PULSE[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	1

Bits 30:24 – NCS_RD_PULSE[6:0] NCS Pulse Length in READ Access
In standard read access, the NCS signal pulse length is defined as:
NCS pulse length = (256 * NCS_RD_PULSE[6] + NCS_RD_PULSE[5:0]) clock cycles
The NCS pulse length must be at least 1 clock cycle.
In page mode read access, the NCS_RD_PULSE parameter defines the duration of the first access to one page.

Bits 22:16 – NRD_PULSE[6:0] NRD Pulse Length
In standard read access, the NRD signal pulse length is defined in clock cycles as:
NRD pulse length = (256 * NRD_PULSE[6] + NRD_PULSE[5:0]) clock cycles
The NRD pulse length must be at least 1 clock cycle.
In page mode read access, the NRD_PULSE parameter defines the duration of the subsequent accesses in the page.

Bits 14:8 – NCS_WR_PULSE[6:0] NCS Pulse Length in WRITE Access
In write access, the NCS signal pulse length is defined as:
NCS pulse length = (256 * NCS_WR_PULSE[6] + NCS_WR_PULSE[5:0]) clock cycles
The NCS pulse length must be at least 1 clock cycle.

Bits 6:0 – NWE_PULSE[6:0] NWE Pulse Length
The NWE signal pulse length is defined as:
NWE pulse length = (256 * NWE_PULSE[6] + NWE_PULSE[5:0]) clock cycles
The NWE pulse length must be at least 1 clock cycle.

Downloaded from Arrow.com.

33.20.3 SMC Cycle Register

Name: SMC_CYCLEx
Offset: 0x08 + x*0x10 [x=0..5]
Reset: 0x00030003
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SMC Write Protection Mode Register.

Note: The number of SMC_CYCLE registers depends on the chip select number.

Bit	31	30	29	28	27	26	25	24
								NRD_CYCLE[8]
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
	NRD_CYCLE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	1

Bit	15	14	13	12	11	10	9	8
								NWE_CYCLE[8]
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	NWE_CYCLE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	1

Bits 24:16 – NRD_CYCLE[8:0] Total Read Cycle Length
The total read cycle length is the total duration in clock cycles of the read cycle. It is equal to the sum of the setup, pulse and hold steps of the NRD and NCS signals. It is defined as:
Read cycle length = (NRD_CYCLE[8:7] * 256 + NRD_CYCLE[6:0]) clock cycles

Bits 8:0 – NWE_CYCLE[8:0] Total Write Cycle Length
The total write cycle length is the total duration in clock cycles of the write cycle. It is equal to the sum of the setup, pulse and hold steps of the NWE and NCS signals. It is defined as:
Write cycle length = (NWE_CYCLE[8:7] * 256 + NWE_CYCLE[6:0]) clock cycles

Downloaded from Arrow.com.

33.20.4 SMC Mode Register

Name: SMC_MODEx
Offset: 0x0C + x*0x10 [x=0..5]
Reset: 0x10001000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SMC Write Protection Mode Register.

The user must confirm the SMC configuration by writing any one of the SMC_MODE registers.

Note: The number of SMC_MODE registers depends on the chip select number.

Bit	31	30	29	28	27	26	25	24
			PS[1:0]					PMEN
Access			R/W	R/W				R/W
Reset			0	1				0

Bit	23	22	21	20	19	18	17	16
				TDF_MODE	TDF_CYCLES[3:0]			
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			DBW[1:0]					BAT
Access			R/W	R/W				R/W
Reset			0	1				0

Bit	7	6	5	4	3	2	1	0
			EXNW_MODE[1:0]				WRITE_MODE	READ_MODE
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 29:28 – PS[1:0] Page Size
If page mode is enabled, this field indicates the size of the page in bytes.

Value	Name	Description
0	BYTE_4	4-byte page
1	BYTE_8	8-byte page
2	BYTE_16	16-byte page
3	BYTE_32	32-byte page

Bit 24 – PMEN Page Mode Enabled

Value	Description
1	Asynchronous burst read in page mode is applied on the corresponding chip select.
0	Standard read is applied.

Bit 20 – TDF_MODE TDF Optimization

Value	Description
1	TDF optimization enabled—The number of TDF wait states is optimized using the setup period of the next read/write access.
0	TDF optimization disabled—The number of TDF wait states is inserted before the next access begins.

Bits 19:16 – TDF_CYCLES[3:0] Data Float Time
This field gives the integer number of clock cycles required by the external device to release the data after the rising edge of the read controlling signal. The SMC always provides one full cycle of bus turnaround after the TDF_CYCLES period. The external bus cannot be used by another chip select during TDF_CYCLES + 1 cycles. From 0 up to 15 TDF_CYCLES can be set.

Downloaded from Arrow.com.

Bits 13:12 – DBW[1:0] Data Bus Width

Value	Name	Description
0	BIT_8	8-bit bus
1	BIT_16	16-bit bus
2	BIT_32	32-bit bus
3	—	Reserved

Bit 8 – BAT Byte Access Type

This field is used only if DBW defines a 16- or 32-bit data bus.

Value	Name	Description
0	BYTE_SELECT	Byte select access type: - Write operation is controlled using NCS, NWE, NBS0, NBS1, NBS2 and NBS3 - Read operation is controlled using NCS, NRD, NBS0, NBS1, NBS2 and NBS3
1	BYTE_WRITE	Byte write access type: - Write operation is controlled using NCS, NWR0, NWR1, NWR2, NWR3 - Read operation is controlled using NCS and NRD

Bits 5:4 – EXNW_MODE[1:0] NWAIT Mode

The NWAIT signal is used to extend the current read or write signal. It is only taken into account during the pulse phase of the read and write controlling signal. When the use of NWAIT is enabled, at least one cycle hold duration must be programmed for the read and write controlling signal.

Value	Name	Description
0	DISABLED	Disabled Mode—The NWAIT input signal is ignored on the corresponding Chip Select.
1	—	Reserved
2	FROZEN	Frozen Mode—If asserted, the NWAIT signal freezes the current read or write cycle. After deassertion, the read/write cycle is resumed from the point where it was stopped.
3	READY	Ready Mode—The NWAIT signal indicates the availability of the external device at the end of the pulse of the controlling read or write signal, to complete the access. If high, the access normally completes. If low, the access is extended until NWAIT returns high.

Bit 1 – WRITE_MODE Selection of the Control Signal for Write Operation

Value	Name	Description
0	NCS_CTRL	Write operation controlled by NCS signal—If TDF optimization is enabled (TDF_MODE = 1), TDF wait states will be inserted after the setup of NCS.
1	NWE_CTRL	Write operation controlled by NWE signal—If TDF optimization is enabled (TDF_MODE = 1), TDF wait states will be inserted after the setup of NWE.

Bit 0 – READ_MODE Selection of the Control Signal for Read Operation

Value	Name	Description
0	NCS_CTRL	Read operation controlled by NCS signal - If TDF cycles are programmed, the external bus is marked busy after the rising edge of NCS. - If TDF optimization is enabled (TDF_MODE = 1), TDF wait states are inserted after the setup of NCS.
1	NRD_CTRL	Read operation controlled by NRD signal - If TDF cycles are programmed, the external bus is marked busy after the rising edge of NRD. - If TDF optimization is enabled (TDF_MODE = 1), TDF wait states are inserted after the setup of NRD.

Downloaded from Arrow.com.

33.20.5 SMC Off-Chip Memory Scrambling Register

Name: SMC_OCMS
Offset: 0x80
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			CS5SE	CS4SE	CS3SE	CS2SE	CS1SE	CS0SE
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				TAMPCLR				SMSE
Access				R/W				R/W
Reset				0				0

Bits 8, 9, 10, 11, 12, 13 – CSSE Chip Select Scrambling Enable

Value	Description
0	Disables scrambling for CSx.
1	Enables scrambling for CSx.

Bit 4 – TAMPCLR Tamper Clear Enable

Value	Description
0	A tamper detection event has no effect on SMC scrambling keys.
1	A tamper detection event immediately clears SMC scrambling keys.

Bit 0 – SMSE Static Memory Controller Scrambling Enable

Value	Description
0	Disables scrambling for SMC access.
1	Enables scrambling for SMC access.

Downloaded from Arrow.com.

33.20.6 SMC Off-Chip Memory Scrambling Key1 Register

Name: SMC_KEY1
Offset: 0x84
Reset: 0x00000000
Property: Write-only

This register is a ‘Write-once’ register: the first write access after a system reset prevents any further modification of the register value.

This register is erased if a tamper is detected on fast wakeup pins and bit SMC_OCMS.TAMPCLR = 1.

Bit	31	30	29	28	27	26	25	24
	KEY1[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	KEY1[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	KEY1[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	KEY1[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – KEY1[31:0] Off-Chip Memory Scrambling (OCMS) Key Part 1
When off-chip memory scrambling is enabled, KEY1 and KEY2 values determine data scrambling.

Downloaded from Arrow.com.

33.20.7 SMC Off-Chip Memory Scrambling Key2 Register

Name: SMC_KEY2
Offset: 0x88
Reset: 0x00000000
Property: Write-only

This register is a ‘Write-once’ register: the first write access after a system reset prevents any further modification of the register value.

This register is erased if a tamper is detected on fast wakeup pins and bit SMC_OCMS.TAMPCLR = 1.

Bit	31	30	29	28	27	26	25	24
	KEY2[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	KEY2[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	KEY2[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	KEY2[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – KEY2[31:0] Off-Chip Memory Scrambling (OCMS) Key Part 2
When off-chip memory scrambling is enabled, KEY1 and KEY2 values determine data scrambling.

Downloaded from Arrow.com.

33.20.8 SMC Safety Report Interrupt Enable Register

Name: SMC_SRIER
Offset: 0x90
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SRIE
Access								R/W
Reset								0

Bit 0 – SRIE Safety Report Interrupt Enable

Value	Description
0	Disables the SMC safety report interrupt from External Bus Interface.
1	Enables the SMC safety report interrupt from External Bus Interface.

Downloaded from Arrow.com.

33.20.9 SMC Write Protection Mode Register

Name: SMC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x534D43	PASSWD	Writing any other value in this field aborts the write operation of bit WPEN. Always reads as 0.

Bit 0 – WPEN Write Protection Enable
See list of write-protected registers in Coding Timing Parameters.

Value	Description
0	Disables write protection if WPKEY value corresponds to 0x534D43 (“SMC” in ASCII).
1	Enables write protection if WPKEY value corresponds to 0x534D43 (“SMC” in ASCII).

Downloaded from Arrow.com.

33.20.10 SMC Write Protection Status Register

Name: SMC_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
							SWETYP[1:0]	
Access							R	R
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SWE	SEQE		WPVS
Access					R	R		R
Reset					0	0		0

Bits 25:24 – SWETYP[1:0] Software Error Type (Cleared on read)

Value	Name	Description
0	READ_WO	A write-only register has been read.
1	WRITE_WO	A write access has been performed on a read-only register.
2	UNDEF_RW	Access to an undefined address.

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 3 – SWE Software Control Error (Cleared on read)

Value	Description
0	No software error has occurred since the last read of SMC_WPSR.
1	A software error has occurred since the last read of SMC_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (Cleared on read)

Value	Description
0	No internal sequencer error has occurred since the last read of SMC_WPSR.
1	An internal sequencer error has occurred since the last read of SMC_WPSR. This flag can only be set under abnormal operating conditions.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the SMC_WPSR.
1	A write protection violation occurred since the last read of the SMC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

34. Programmable Multibit Error Correction Code Controller (PMECC)

34.1 Description

The Programmable Multibit Error Correction Code Controller (PMECC) is a programmable binary BCH (Bose, Chaudhuri and Hocquenghem) encoder/decoder. This controller can be used to generate redundancy information for both Single-Level Cell (SLC) and Multi-level Cell (MLC) NAND Flash devices. It supports redundancy for correction of 2, 4, 8, 12 or 24 bits of error per sector of data.

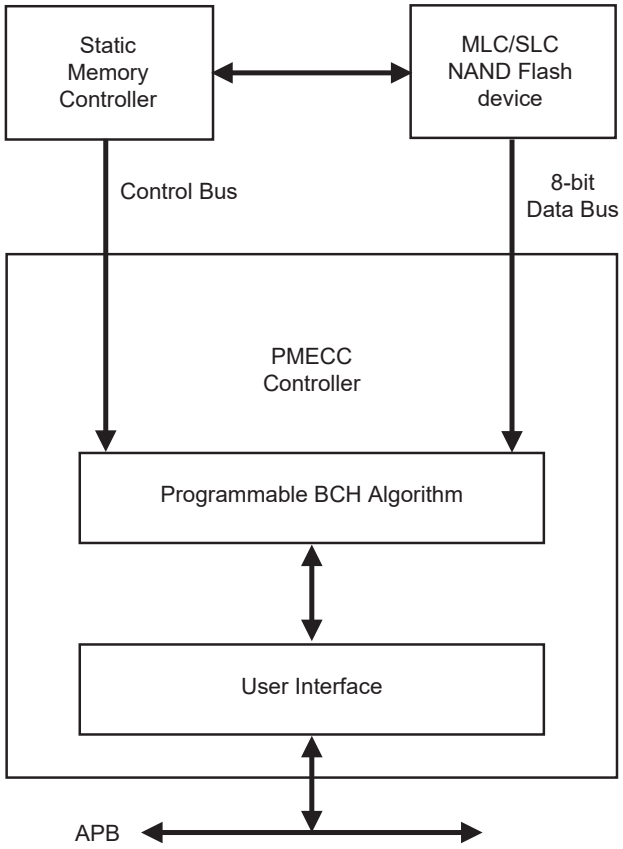
34.2 Embedded Characteristics

- 8-bit NAND Flash Data Bus Support
- Multibit Error Correcting Code
- Algorithm Based on Binary Shortened Bose, Chaudhuri and Hocquenghem (BCH) Codes
- Programmable Error Correcting Capability: 2, 4, 8, 12 and 24 Bits of Error per Sector
- Programmable Sector Size: 512 Bytes or 1024 Bytes
- Programmable Number of Sectors per Page: 1, 2, 4 or 8 Sectors of Data per Page
- Programmable Spare Area Size
- Supports Spare Area ECC Protection
- Supports 8 Kbytes Page Size Using 1024 Bytes per Sector and 4 Kbytes Page Size Using 512 Bytes per Sector
- Configurable through APB Interface
- Interrupt-Driven Multibit Error Detection

Downloaded from Arrow.com.

34.3 Block Diagram

Figure 34-1. Block Diagram



34.4 Functional Description

The NAND Flash sector size is programmable and can be set to 512 bytes or 1024 bytes. The PMECC module generates redundancy at encoding time, when a NAND write page operation is performed. The redundancy is appended to the page and written in the spare area. This operation is performed by the processor. It moves the content of the PMECCx registers into the NAND Flash memory. The number of registers depends on the selected error correction capability, refer to the table “Relevant Redundancy Registers”. This operation is executed for each sector. At decoding time, the PMECC module generates the remainder of the received codeword by minimal polynomials. When all polynomial remainders for a given sector are set to zero, no error occurred. When the polynomial remainders are other than zero, the codeword is corrupted and further processing is required.

The PMECC generates an interrupt indicating that an error occurred. The processor must read the PMECC Interrupt Status Register (PMECC_ISR). This register indicates which sector is corrupted.

To find the error location within a sector, the processor must execute the following decoding steps:

1. Syndrome computation
2. Find the error locator polynomials
3. Find the roots of the error locator polynomial

All decoding steps involve finite field computation, for which a library of finite field arithmetic must be available to perform addition, multiplication and inversion. The finite field arithmetic operations can be performed through the use of a memory mapped lookup table, or direct software implementation. The software implementation presented is based on lookup tables. Two tables named gf_log and gf_antilog are used. If alpha is the primitive element of the field, then a power of alpha is in the field. Assume beta = alpha ^ index, then beta belongs to the field, and

Downloaded from Arrow.com.

gf_log(beta) = gf_log(alpha ^ index) = index. The gf_antilog tables provide exponent inverse of the element, if beta = alpha ^ index, then gf_antilog(index) = beta.

The first step consists of the syndrome computation. The PMECC computes the remainders and software must substitute the power of the primitive element.

The procedure implementation is given in “Remainder Substitution Procedure”.

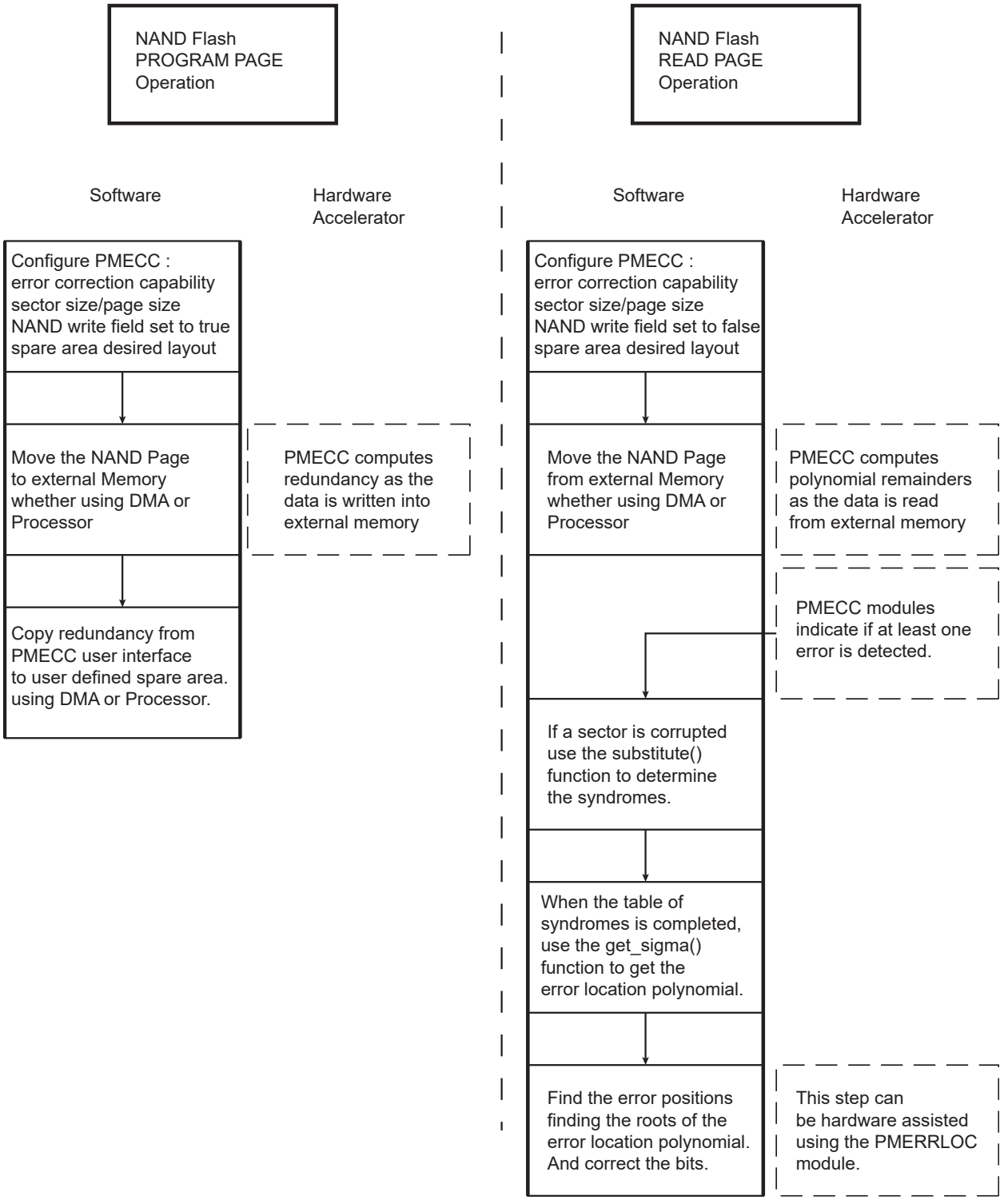
The second step is the most software intensive. It is the Berlekamp’s iterative algorithm for finding the error-location polynomial.

The procedure implementation is given in “Find the Error Location Polynomial Sigma(x)”.

The last step is finding the root of the error location polynomial. This step can be very software intensive, as there is no straightforward method of finding the roots, except by evaluating each element of the field in the error location polynomial. However, a hardware accelerator can be used to find the roots of the polynomial. The Programmable Multibit Error Correction Code Location (PMERRLOC) module provides this kind of hardware acceleration.

Downloaded from Arrow.com.

Figure 34-2. Software/Hardware Multibit Error Correction Dataflow



34.4.1 MLC/SLC Write Page Operation using PMECC

When an MLC write page operation is performed, the PMECC controller is configured with the NANDWR bit in the PMECC Configuration Register (PMECC_CFG) set to one. When the NAND spare area contains file system information and redundancy (PMECCx), the spare area is error protected, then PMECC_CFG.SPAREEN is set to ‘1’. When the NAND spare area contains only redundancy information, SPAREEN is set to ‘0’.

Downloaded from Arrow.com.

When the write page operation is terminated, the user writes the redundancy in the NAND spare area. This operation can be done with DMA assistance.

Table 34-1. Relevant Redundancy Registers

BCH_ERR field	Sector size set to 512 bytes	Sector size set to 1024 bytes
0	PMECC_ECC0	PMECC_ECC0
1	PMECC_ECC0, PMECC_ECC1	PMECC_ECC0, PMECC_ECC1
2	PMECC_ECC0, PMECC_ECC1, PMECC_ECC2, PMECC_ECC3	PMECC_ECC0, PMECC_ECC1, PMECC_ECC2, PMECC_ECC3
3	PMECC_ECC0, PMECC_ECC1, PMECC_ECC2, PMECC_ECC3, PMECC_ECC4, PMECC_ECC5, PMECC_ECC6	PMECC_ECC0, PMECC_ECC1, PMECC_ECC2, PMECC_ECC3, PMECC_ECC4, PMECC_ECC5, PMECC_ECC6
4	PMECC_ECC0, PMECC_ECC1, PMECC_ECC2, PMECC_ECC3, PMECC_ECC4, PMECC_ECC5, PMECC_ECC6, PMECC_ECC7, PMECC_ECC8, PMECC_ECC9	PMECC_ECC0, PMECC_ECC1, PMECC_ECC2, PMECC_ECC3, PMECC_ECC4, PMECC_ECC5, PMECC_ECC6, PMECC_ECC7, PMECC_ECC8, PMECC_ECC9, PMECC_ECC10

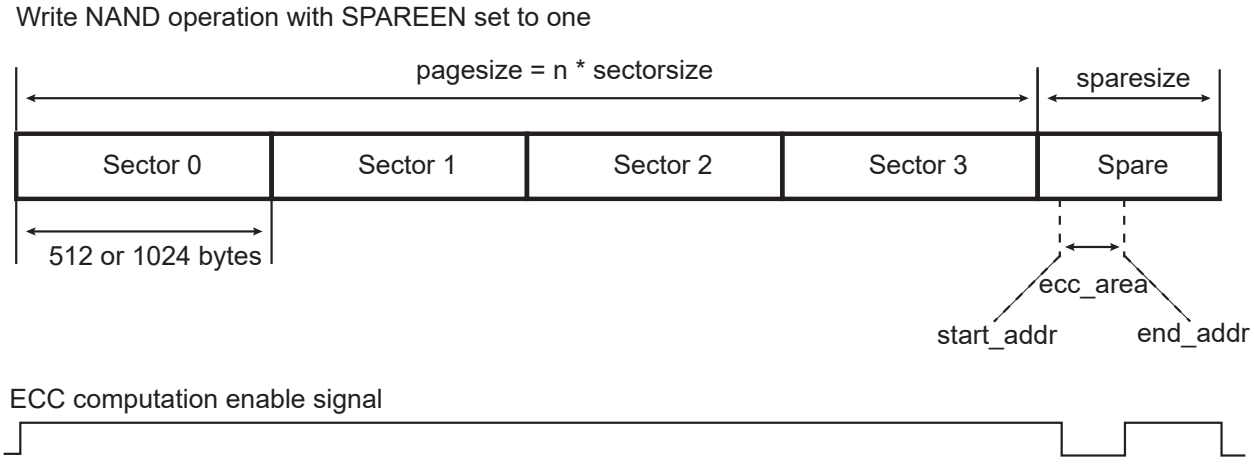
Table 34-2. Number of Relevant ECC bytes per Sector, copied from LSbyte to MSbyte

BCH_ERR field	Sector size set to 512 bytes	Sector size set to 1024 bytes
0	4 bytes	4 bytes
1	7 bytes	7 bytes
2	13 bytes	14 bytes
3	20 bytes	21 bytes
4	39 bytes	42 bytes

34.4.1.1 SLC/MLC Write Operation with Spare Enable Bit Set

When PMECC_CFG.SPAREEN is set to ‘1’, the spare area of the page is encoded with the stream of data of the last sector of the page. This mode is entered by writing one in the DATA bit in the PMECC Control Register (PMECC_CTRL). When the encoding process is over, the redundancy is written to the spare area in user mode, PMECC_CTRL.USER must be set to ‘1’.

Figure 34-3. NAND Write Operation with Spare Encoding

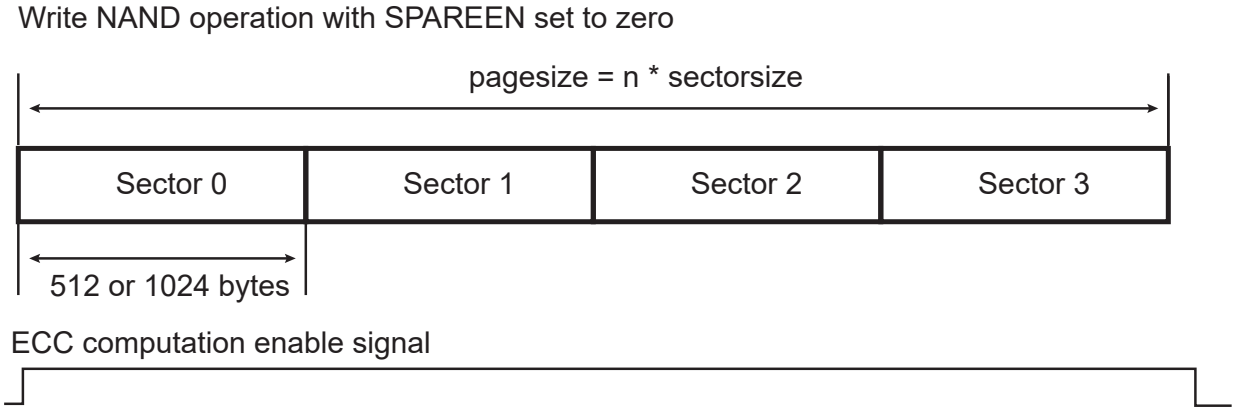


Downloaded from Arrow.com.

34.4.1.2 MLC/SLC Write Operation with Spare Area Disabled

When PMECC_CFG.SPAREEN is set to ‘0’, the spare area is not encoded with the stream of data. This mode is entered by writing ‘1’to PMECC_CTRL.DATA.

Figure 34-4. NAND Write Operation



34.4.2 MLC/SLC Read Page Operation using PMECC

Table 34-3. Relevant Remainders Registers

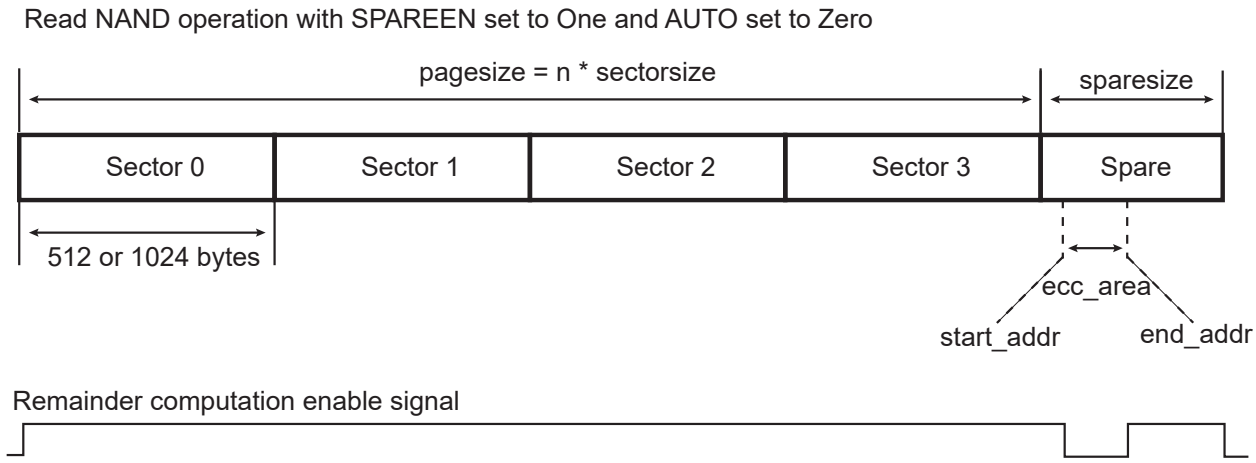
BCH_ERR field	Sector size set to 512 bytes	Sector size set to 1024 bytes
0	PMECC_REM0	PMECC_REM0
1	PMECC_REM0, PMECC_REM1	PMECC_REM0, PMECC_REM1
2	PMECC_REM0, PMECC_REM1, PMECC_REM2, PMECC_REM3,	PMECC_REM0, PMECC_REM1, PMECC_REM2, PMECC_REM3
3	PMECC_REM0, PMECC_REM1, PMECC_REM2, PMECC_REM3, PMECC_REM4, PMECC_REM5, PMECC_REM6, PMECC_REM7	PMECC_REM0, PMECC_REM1, PMECC_REM2, PMECC_REM3, PMECC_REM4, PMECC_REM5, PMECC_REM6, PMECC_REM7
4	PMECC_REM0, PMECC_REM1, PMECC_REM2, PMECC_REM3, PMECC_REM4, PMECC_REM5, PMECC_REM6, PMECC_REM7, PMECC_REM8, PMECC_REM9, PMECC_REM10, PMECC_REM11	PMECC_REM0, PMECC_REM1, PMECC_REM2, PMECC_REM3, PMECC_REM4, PMECC_REM5, PMECC_REM6, PMECC_REM7, PMECC_REM8, PMECC_REM9, PMECC_REM10, PMECC_REM11

34.4.2.1 MLC/SLC Read Operation with Spare Decoding

When the spare area is protected, the spare area contains valid data. As the redundancy may be included in the middle of the information stream, the user programs the start address and the end address of the ECC area. The controller will automatically skip the ECC area. This mode is entered by writing a ‘1’ to PMECC_CTRL.DATA. When the page has been fully retrieved from NAND, the ECC area is read using the user mode by writing a ‘1’ to PMECC_CTRL.USER.

Downloaded from Arrow.com.

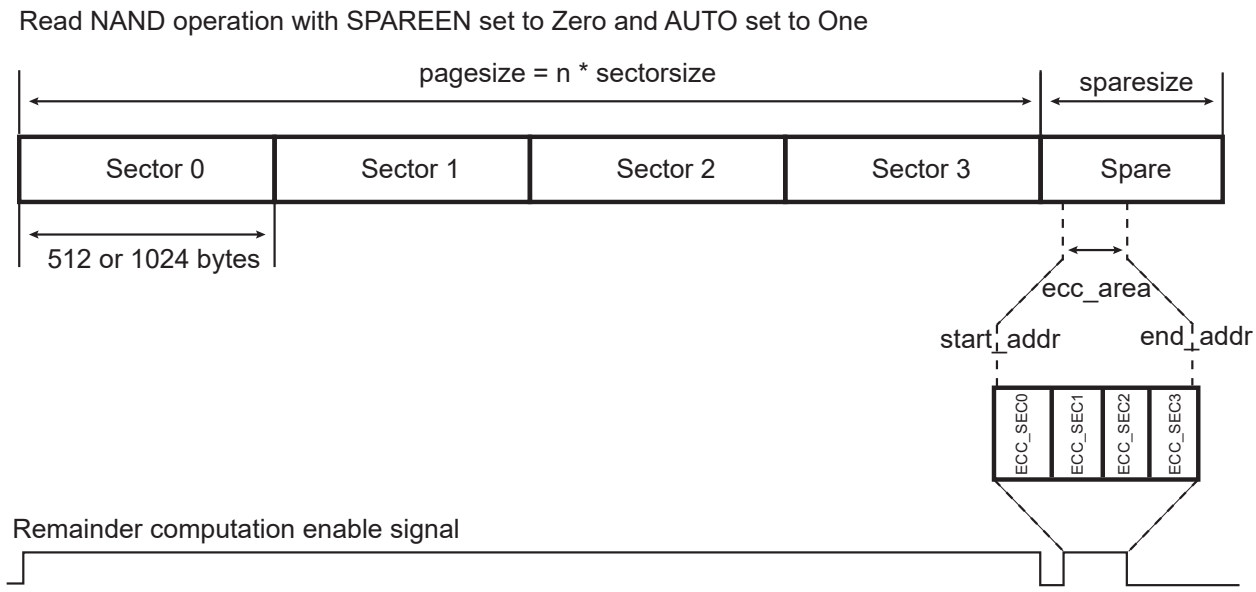
Figure 34-5. Read Operation with Spare Decoding



34.4.2.2 MLC/SLC Read Operation

If the spare area is not protected with the error correcting code, the redundancy area is retrieved directly. This mode is entered by writing a '1' to PMECC_CTRL.DATA. When PMECC_CFG.AUTO is set to '1', the ECC is retrieved automatically, otherwise the ECC must be read using user mode.

Figure 34-6. Read Operation



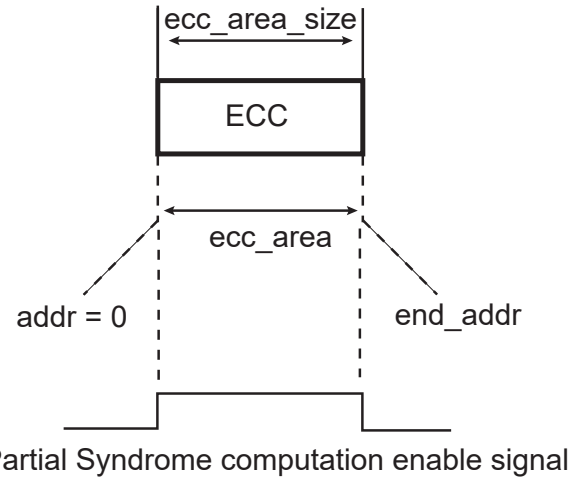
34.4.2.3 MLC/SLC User Read ECC Area

This mode allows a manual retrieve of the ECC.

This mode is entered by writing a '1' to PMECC_CTRL.USER.

Downloaded from Arrow.com.

Figure 34-7. User Read Mode



34.5 Software Implementation

34.5.1 Remainder Substitution Procedure

The substitute function evaluates the polynomial remainder, with different values of the field primitive elements. The finite field arithmetic addition operation is performed with the Exclusive or. The finite field arithmetic multiplication operation is performed through the gf_log, gf_antilog lookup tables.

The REM2NP1 and REMN2NP3 fields of the PMECC Remainder x registers (PMECC_REMx) contain only odd remainders. Each bit indicates whether the coefficient of the polynomial remainder is set to zero or not.

NB_ERROR_MAX defines the maximum value of the error correcting capability.

NB_ERROR defines the error correcting capability selected at encoding/decoding time.

NB_FIELD_ELEMENTS defines the number of elements in the field.

si[] is a table that holds the current syndrome value, an element of that table belongs to the field. This is also a shared variable for the next step of the decoding operation.

oo[] is a table that contains the degree of the remainders.

```
int substitute()
    {
    int i;
    int j;
    for (i = 1; i < 2 * NB_ERROR_MAX;
        i++)
    {
     si[i] = 0;
    }
    for (i = 1; i < 2*NB_ERROR;
        i++)
    {
     for (j = 0; j < oo[i];
      j++)
     {
      if (REM2NPX[i][j])
      {
       si[i] = gf_antilog[(i *
        j)%NB_FIELD_ELEMENTS] ^ si[i];
      }
     }
    }
    return 0;
    }
```


Downloaded from Arrow.com.

34.5.2 Find the Error Location Polynomial Sigma(x)

The sample code below gives a Berlekamp iterative procedure for finding the value of the error location polynomial.

The input of the procedure is the si[] table defined in the remainder substitution procedure.

The output of the procedure is the error location polynomial named smu (sigma mu). The polynomial coefficients belong to the field. The smu[NB_ERROR+1][] is a table that contains all these coefficients.

NB_ERROR_MAX defines the maximum value of the error correcting capability.

NB_ERROR defines the error correcting capability selected at encoding/decoding time.

NB_FIELD_ELEMENTS defines the number of elements in the field.

```
int get_sigma()
    {
    int i;
    int j;
    int k;
    /* mu          */
    int
      mu[NB_ERROR_MAX+2];
    /* sigma ro   */
    int
      sro[2*NB_ERROR_MAX+1];
    /* discrepancy */
    int
      dmu[NB_ERROR_MAX+2];
    /* delta order   */
    int
      delta[NB_ERROR_MAX+2];
    /* index of largest delta
        */
    int ro;
    int largest;
    int diff;
    /*
      */
    /*      First Row
      */
    /*
      */
    /* Mu */
    mu[0]  = -1; /* Actually -1/2
        */
    /* Sigma(x) set to 1
      */
    for (i = 0; i <
        (2*NB_ERROR_MAX+1); i++)
     smu[0][i] = 0;
    smu[0][0] = 1;
    /* discrepancy set to 1
      */
    dmu[0] = 1;
    /* polynom order set to 0
        */
    lmu[0] = 0;
    /* delta set to -1 */
    delta[0]  = (mu[0] * 2 - lmu[0])
        >> 1;
    /*
      */
    /*      Second Row
      */
    /*
      */
    /* Mu */
    mu[1]  = 0;
    /* Sigma(x) set to 1
      */
    for (i = 0; i <
        (2*NB_ERROR_MAX+1); i++)
     smu[1][i] = 0;
    smu[1][0] = 1;
    /* discrepancy set to Syndrome 1
```


Downloaded from Arrow.com.

```
    */
dmu[1] = si[1];
/* polynom order set to 0
    */
lmu[1] = 0;
/* delta set to 0 */
delta[1]  = (mu[1] * 2 - lmu[1])
    >> 1;
for (i=1; i <= NB_ERROR;
    i++)
{
 mu[i+1] = i << 1;

    /*********************************************/
 /*
    */
 /*
    */
 /*          Compute Sigma (Mu+1)
    */
 /*          And L(mu)
    */
 /* check if discrepancy is set
    to 0 */
 if (dmu[i] == 0)
 {
 /* copy polynom */
 for (j=0; j<2*NB_ERROR_MAX+1;
    j++)
 {
  smu[i+1][j] =
  smu[i][j];
 }
 /* copy previous polynom order
    to the next */
 lmu[i+1] = lmu[i];
 }
 else
 {
 ro      = 0;
 largest = -1;
 /* find largest delta with dmu
    != 0 */
 for (j=0; j<i; j++)
  {
  if (dmu[j])
   {
   if (delta[j] >
  largest)
    {
    largest =
  delta[j];
    ro   = j;
    }
   }
  }
 /* initialize signal ro
  */
 for (k = 0; k < 2*NB_ERROR_MAX+1;
    k ++)
 {
  sro[k] = 0;
 }
 /* compute difference
  */
 diff = (mu[i] -
  mu[ro]);
 /* compute X ^ (2(mu-ro))
    */
 for (k = 0; k <
    (2*NB_ERROR_MAX+1); k ++)
 {
  sro[k+diff] =
  smu[ro][k];
 }
 /* multiply by dmu * dmu[ro]^-1
```


Downloaded from Arrow.com.

```
      */
    for (k = 0; k < 2*NB_ERROR_MAX+1;
      k ++)
    {
     /* dmu[ro] is not equal to zero
       by definition */
     /* check that operand are
       different from 0    */
     if (sro[k] &&
     dmu[i])
     {
      /* galois inverse
     */
      sro[k] =
       gf_antilog[(gf_log[dmu[i]] + (NB_FIELD_ELEMENTS-gf_log[dmu[ro]]) + gf_log[sro[k]]) %
       NB_FIELD_ELEMENTS];
     }
    }
    /* multiply by dmu * dmu[ro]^-1
      */
    for (k = 0; k < 2*NB_ERROR_MAX+1;
      k++)
    {
     smu[i+1][k] = smu[i][k] ^
       sro[k];
     if (smu[i+1][k])
     {
      /* find the order of the
       polynom */
      lmu[i+1] = k <<
     1;
     }
    }
   }
   /*
      */
   /*
      */
   /*      End Compute Sigma (Mu+1)
      */
   /*      And L(mu)
      */
   /*********************************************/
   /* In either case compute delta
      */
   delta[i+1]  = (mu[i+1] * 2 -
      lmu[i+1]) >> 1;
   /* In either case compute the
      discrepancy */
   for (k = 0 ; k <= (lmu[i+1]>>1);
      k++)
   {
    if (k == 0)
    dmu[i+1] =
     si[2*(i-1)+3];
    /* check if one operand of the
      multiplier is null, its index is -1 */
    else if (smu[i+1][k] &&
      si[2*(i-1)+3-k])
    dmu[i+1] =
      gf_antilog[(gf_log[smu[i+1][k]] + gf_log[si[2*(i-1)+3-k]])%nn] ^ dmu[i+1];
   }
   }
   return 0;
   }
```

34.5.3 Find the Error Position

The output of the get_sigma() procedure is a polynomial stored in the smu[NB_ERROR+1][] table. The error position is the roots of that polynomial. The degree of this polynomial is very important information, as it gives the number of errors. The PMERRLOC module provides a hardware accelerator for this step.

Downloaded from Arrow.com.

34.6 Register Summary

Note: The blocks of registers listed below are instanced 8 times in the user interface:

- PMECC_ECCx[x=0..10]
- PMECC_REMx[x=0..11]

Offset	Name	Bit Pos.									
0x00	PMECC_CFG	31:24									
		23:16				AUTO					SPAREEN
		15:8				NANDWR				PAGESIZE[1:0]	
		7:0				SECTORSZ			BCH_ERR[2:0]		
0x04	PMECC_SAREA	31:24									
		23:16									
		15:8									SPARESIZE[8]
		7:0	SPARESIZE[7:0]								
0x08	PMECC_SADDR	31:24									
		23:16									
		15:8									STARTADDR[8]
		7:0	STARTADDR[7:0]								
0x0C	PMECC_EADDR	31:24									
		23:16									
		15:8									ENDADDR[8]
		7:0	ENDADDR[7:0]								
0x10	PMECC_CLK	31:24									
		23:16									
		15:8									
		7:0						CLKCTRL[2:0]			
0x14	PMECC_CTRL	31:24									
		23:16									
		15:8									
		7:0			DISABLE	ENABLE		USER	DATA	RST	
0x18	PMECC_SR	31:24									
		23:16									
		15:8									
		7:0				ENABLE				BUSY	
0x1C	PMECC_IER	31:24									
		23:16									
		15:8									
		7:0								ERRIE	
0x20	PMECC_IDR	31:24									
		23:16									
		15:8									
		7:0								ERRID	
0x24	PMECC_IMR	31:24									
		23:16									
		15:8									
		7:0								ERRIM	
0x28	PMECC_ISR	31:24									
		23:16									
		15:8									
		7:0	ERRIS[7:0]								
0x2C ... 0x3F	Reserved										

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.									
0x40	PMECC_ECC0	31:24	ECC[31:24]								
		23:16	ECC[23:16]								
		15:8	ECC[15:8]								
		7:0	ECC[7:0]								
...											
0x68	PMECC_ECC10	31:24	ECC[31:24]								
		23:16	ECC[23:16]								
		15:8	ECC[15:8]								
		7:0	ECC[7:0]								
0x6C ... 0x023F	Reserved										
0x0240	PMECC_REM0	31:24			REM2NP3[13:8]						
		23:16	REM2NP3[7:0]								
		15:8			REM2NP1[13:8]						
		7:0	REM2NP1[7:0]								
...											
0x026C	PMECC_REM11	31:24			REM2NP3[13:8]						
		23:16	REM2NP3[7:0]								
		15:8			REM2NP1[13:8]						
		7:0	REM2NP1[7:0]								

Downloaded from Arrow.com.

34.6.1 PMECC Configuration Register

Name: PMECC_CFG
Offset: 0x000
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
				AUTO				SPAREEN
Access				R/W				R/W
Reset				0				0

Bit	15	14	13	12	11	10	9	8
				NANDWR			PAGESIZE[1:0]	
Access				R/W			R/W	R/W
Reset				0			0	0

Bit	7	6	5	4	3	2	1	0
				SECTORSZ		BCH_ERR[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit 20 – AUTO Automatic Mode Enable
This bit is only relevant in NAND Read Mode, when spare enable is activated.

Value	Description
0	Indicates that the spare area is not protected. In that case the ECC computation takes into account the ECC area located in the spare area. (within the start address and the end address).
1	Indicates that the spare is error protected. In this case, the ECC computation takes into account the whole spare area minus the ECC area in the ECC computation operation.

Bit 16 – SPAREEN Spare Enable

- For NAND write access:
  — 0: The spare area is skipped
  — 1: The spare area is protected with the last sector of data.
- For NAND read access:
  — 0: The spare area is skipped.
  — 1: The spare area contains protected data or only redundancy information.

Bit 12 – NANDWR NAND Write Access

Value	Description
0	NAND read access
1	NAND write access

Bits 9:8 – PAGESIZE[1:0] Number of Sectors in the Page

Value	Name	Description
0	PAGESIZE_1SEC	1 sector for main area (512 or 1024 bytes)
1	PAGESIZE_2SEC	2 sectors for main area (1024 or 2048 bytes)
2	PAGESIZE_4SEC	4 sectors for main area (2048 or 4096 bytes)
3	PAGESIZE_8SEC	8 errors for main area (4096 or 8192 bytes)

Downloaded from Arrow.com.

Bit 4 – SECTORSZ Sector Size

Value	Description
0	The ECC computation is based on a sector of 512 bytes.
1	The ECC computation is based on a sector of 1024 bytes.

Bits 2:0 – BCH_ERR[2:0] Error Correct Capability

Value	Name	Description
0	BCH_ERR2	2 errors
1	BCH_ERR4	4 errors
2	BCH_ERR8	8 errors
3	BCH_ERR12	12 errors
4	BCH_ERR24	24 errors

Downloaded from Arrow.com.

34.6.2 PMECC Spare Area Size Register

Name: PMECC_SAREA
Offset: 0x004
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 8:0 – SPARESIZE[8:0] Spare Area Size
The spare area size is equal to (SPARESIZE+1) bytes.

Downloaded from Arrow.com.

34.6.3 PMECC Start Address Register

Name: PMECC_SADDR
Offset: 0x008
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 8:0 – STARTADDR[8:0] ECC Area Start Address (byte oriented address)
This field indicates the first byte address of the ECC area. Location 0 matches the first byte of the spare area.

Downloaded from Arrow.com.

34.6.4 PMECC End Address Register

Name: PMECC_EADDR
Offset: 0x00C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	ENDADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 8:0 – ENDADDR[8:0] ECC Area End Address (byte oriented address)
This field indicates the last byte address of the ECC area.

Downloaded from Arrow.com.

34.6.5 PMECC Clock Control Register

Name: PMECC_CLK
Offset: 0x010
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CLKCTRL[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – CLKCTRL[2:0] Clock Control Register
The PMECC datapath setup time is set to CLKCTRL+1.
This field indicates the database setup times in number of clock cycles. At 133 MHz, this field must be programmed with 2, indicating that the setup time is 3 clock cycles.

Downloaded from Arrow.com.

34.6.6 PMECC Control Register

Name: PMECC_CTRL
Offset: 0x014
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			DISABLE	ENABLE		USER	DATA	RST
Access			W	W		W	W	W
Reset			–	–		–	–	–

Bit 5 – DISABLE PMECC Module Disable
The PMECC module must always be configured after being deactivated.

Bit 4 – ENABLE PMECC Module Enable
The PMECC module must always be configured before being activated.

Bit 2 – USER Start a User Mode Phase

Bit 1 – DATA Start a Data Phase

Bit 0 – RST Reset the PMECC Module
When set to one, this bit resets the PMECC Controller; configuration registers remain unaffected.

Downloaded from Arrow.com.

34.6.7 PMECC Status Register

Name: PMECC_SR
Offset: 0x018
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				ENABLE				BUSY
Access				R				R
Reset				0				0

Bit 4 – ENABLE PMECC Module Status

Value	Description
0	The PMECC module is disabled and can be configured.
1	The PMECC module is enabled and the configuration registers cannot be written.

Bit 0 – BUSY Kernel of the PMECC is Busy

Downloaded from Arrow.com.

34.6.8 PMECC Interrupt Enable Register

Name: PMECC_IER
Offset: 0x01C
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ERRIE
Access								W
Reset								–

Bit 0 – ERRIE Error Interrupt Enable

Downloaded from Arrow.com.

34.6.9 PMECC Interrupt Disable Register

Name: PMECC_IDR
Offset: 0x020
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ERRID
Access								W
Reset								–

Bit 0 – ERRID Error Interrupt Disable

Downloaded from Arrow.com.

34.6.10 PMECC Interrupt Mask Register

Name: PMECC_IMR
Offset: 0x024
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								ERRIM
Reset								0

Bit 0 – ERRIM Error Interrupt Mask

Downloaded from Arrow.com.

34.6.11 PMECC Interrupt Status Register

Name: PMECC_ISR
Offset: 0x028
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ERRIS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – ERRIS[7:0] Error Interrupt Status
When set to one, bit i of the PMECC_ISR indicates that sector i is corrupted.

Downloaded from Arrow.com.

34.6.12 PMECC ECC x Register

Name: PMECC_ECCx
Offset: 0x40 + x*0x04 [x=0..10]
Reset: 0x00000000
Property: Read-only

Note: The block of registers PMECC_ECCx[x=0..10] is instanced 8 times in the user interface.

Bit	31	30	29	28	27	26	25	24
	ECC[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ECC[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ECC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ECC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ECC[31:0] BCH Redundancy
This register contains the remainder of the division of the codeword by the generator polynomial.

Downloaded from Arrow.com.

34.6.13 PMECC Remainder x Register

Name: PMECC_REMx
Offset: 0x0240 + x*0x04 [x=0..11]
Reset: 0x00000000
Property: Read-only

Note: The block of registers PMECC_REMx[x=0..11] is instanced 8 times in the user interface.

Bit	31	30	29	28	27	26	25	24
			REM2NP3[13:8]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	REM2NP3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			REM2NP1[13:8]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	REM2NP1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 29:16 – REM2NP3[13:0] BCH Remainder 2 * N + 3
When sector size is set to 512 bytes, bit REM2NP3[29] is not used and read as zero.
If bit i of the REM2NP3 field is set to one then the coefficient of the X ^ i is set to one, otherwise the coefficient is zero.

Bits 13:0 – REM2NP1[13:0] BCH Remainder 2 * N + 1
When sector size is set to 512 bytes, bit REM2NP1[13] is not used and read as zero.
If bit i of the REM2NP1 field is set to one then the coefficient of the X ^ i is set to one, otherwise the coefficient is zero.

Downloaded from Arrow.com.

35. Programmable Multibit ECC Error Location Controller (PMERRLOC)

35.1 Description

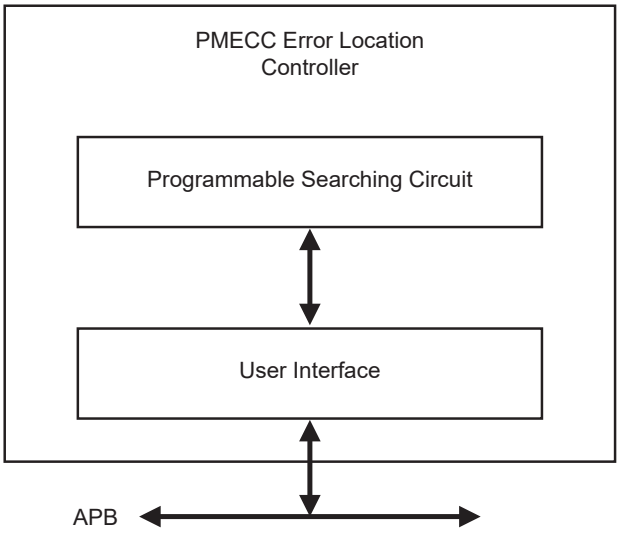
The PMECC Error Location Controller provides hardware acceleration for determining roots of polynomials over two finite fields: GF(2^13) and GF(2^14). It integrates 24 fully programmable coefficients. These coefficients belong to GF(2^13) or GF(2^14). The coefficient programmed in the PMERRLOC_SIGMAx register is the coefficient of degree x in the polynomial.

35.2 Embedded Characteristics

- Provides Hardware Acceleration to Determine Roots of Polynomials Defined over a Finite Field
- Programmable Finite Field GF(2^13) or GF(2^14)
- Finds Roots of Error Locator Polynomial
- Programmable Number of Roots

35.3 Block Diagram

Figure 35-1. Block Diagram



35.4 Functional Description

The PMERRLOC search operation is started as soon as a write access is detected in the PMERRLOC_ELEN register and can be disabled by writing to the PMERRLOC_ELDIS register. PMERRLOC_ELEN.ENINIT shall be initialized with the number of Galois field elements to test. The set of the roots can be limited to a valid range.

Table 35-1. ENINIT Field Value for a Sector Size of 512 Bytes

Error Correcting Capability	ENINIT Value
2	4122
4	4148
8	4200

Downloaded from Arrow.com.

.....continued	
Error Correcting Capability	**ENINIT Value**
12	4252
24	4408

Table 35-2. ENINIT Field Value for a Sector Size of 1024 Bytes

Error Correcting Capability	ENINIT Value
2	8220
4	8248
8	8304
12	8360
24	8528

When the PMERRLOC is searching for roots, PMERRLOC_ELSR.BUSY remains asserted. An interrupt is asserted at the end of the computation, and PMERRLOC_ELISR.DONE is set. PMERRLOC_ELISR.ERR_CNT indicates the number of errors. The error position can be read in the PMERRLOC_ELx registers.

Downloaded from Arrow.com.

35.5 Register Summary

Offset	Name	Bit Pos.								
0x00	PMERRLOC_ELCFG	31:24								
		23:16				ERRNUM[4:0]				
		15:8								
		7:0								SECTORSZ
0x04	PMERRLOC_ELPRIM	31:24								
		23:16								
		15:8	PRIMITIV[15:8]							
		7:0	PRIMITIV[7:0]							
0x08	PMERRLOC_ELEN	31:24								
		23:16								
		15:8			ENINIT[13:8]					
		7:0	ENINIT[7:0]							
0x0C	PMERRLOC_ELDIS	31:24								
		23:16								
		15:8								
		7:0								DIS
0x10	PMERRLOC_ELSR	31:24								
		23:16								
		15:8								
		7:0								BUSY
0x14	PMERRLOC_ELIER	31:24								
		23:16								
		15:8								
		7:0								DONE
0x18	PMERRLOC_ELIDR	31:24								
		23:16								
		15:8								
		7:0								DONE
0x1C	PMERRLOC_ELIMR	31:24								
		23:16								
		15:8								
		7:0								DONE
0x20	PMERRLOC_ELISR	31:24								
		23:16								
		15:8				ERR_CNT[4:0]				
		7:0								DONE
0x24 ... 0x27	Reserved									
0x28	PMERRLOC_SIGMA0	31:24								
		23:16								
		15:8			SIGMA0[13:8]					
		7:0	SIGMA0[7:0]							
0x2C	PMERRLOC_SIGMA1	31:24								
		23:16								
		15:8			SIGMA[13:8]					
		7:0	SIGMA[7:0]							
...										
0x88	PMERRLOC_SIGMA24	31:24								
		23:16								
		15:8			SIGMA[13:8]					
		7:0	SIGMA[7:0]							
0x8C	PMERRLOC_EL0	31:24								
		23:16								
		15:8			ERRLOCN[13:8]					
		7:0	ERRLOCN[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
...										
0xE8	PMERRLOC_EL23	31:24								
		23:16								
		15:8			ERRLOCN[13:8]					
		7:0	ERRLOCN[7:0]							

Downloaded from Arrow.com.

35.5.1 PMERRLOC Configuration Register

Name: PMERRLOC_ELCFG
Offset: 0x000
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
				ERRNUM[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SECTORSZ
Access								R/W
Reset								0

Bits 20:16 – ERRNUM[4:0] Number of Errors

Bit 0 – SECTORSZ Sector Size

Value	Description
0	The ECC computation is based on a 512-byte sector.
1	The ECC computation is based on a 1024-byte sector.

Downloaded from Arrow.com.

35.5.2 PMERRLOC Primitive Register

Name: PMERRLOC_ELPRIM
Offset: 0x004
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PRIMITIV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PRIMITIV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PRIMITIV[15:0] Primitive Polynomial

Downloaded from Arrow.com.

35.5.3 PMERRLOC Enable Register

Name: PMERRLOC_ELEN
Offset: 0x008
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			ENINIT[13:8]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ENINIT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 13:0 – ENINIT[13:0] Initial Number of Bits in the Codeword

Downloaded from Arrow.com.

35.5.4 PMERRLOC Disable Register

Name: PMERRLOC_ELDIS
Offset: 0x00C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								DIS
Access								R/W
Reset								0

Bit 0 – DIS Disable Error Location Engine

Downloaded from Arrow.com.

35.5.5 PMERRLOC Status Register

Name: PMERRLOC_ELSR
Offset: 0x010
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								BUSY
Access								R/W
Reset								0

Bit 0 – BUSY Error Location Engine Busy

Downloaded from Arrow.com.

35.5.6 PMERRLOC Interrupt Enable Register

Name: PMERRLOC_ELIER
Offset: 0x014
Reset: 0x00000000
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								DONE
Access								W
Reset								0

Bit 0 – DONE Computation Terminated Interrupt Enable

Downloaded from Arrow.com.

35.5.7 PMERRLOC Interrupt Disable Register

Name: PMERRLOC_ELIDR
Offset: 0x018
Reset: 0x00000000
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								DONE
Access								W
Reset								0

Bit 0 – DONE Computation Terminated Interrupt Disable

Downloaded from Arrow.com.

35.5.8 PMERRLOC Interrupt Mask Register

Name: PMERRLOC_ELIMR
Offset: 0x01C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								DONE
Access								R
Reset								0

Bit 0 – DONE Computation Terminated Interrupt Mask

Downloaded from Arrow.com.

35.5.9 PMERRLOC Interrupt Status Register

Name: PMERRLOC_ELISR
Offset: 0x020
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access				ERR_CNT[4:0]				
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Access								DONE
Reset								R
								0

Bits 12:8 – ERR_CNT[4:0] Error Counter Value

Bit 0 – DONE Computation Terminated Interrupt Status

Downloaded from Arrow.com.

35.5.10 PMERRLOC SIGMA0 Register

Name: PMERRLOC_SIGMA0
Offset: 0x028
Reset: 0x00000001
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			SIGMA0[13:8]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SIGMA0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	1

Bits 13:0 – SIGMA0[13:0] Coefficient of Degree 0 in the SIGMA Polynomial
SIGMA0 belongs to the finite field GF(2^13) when the sector size is set to 512 bytes.
SIGMA0 belongs to the finite field GF(2^14) when the sector size is set to 1024 bytes.

Downloaded from Arrow.com.

35.5.11 PMERRLOC SIGMAx Register

Name: PMERRLOC_SIGMAx
Offset: 0x2C + (x-1)*0x04 [x=1..24]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			SIGMA[13:8]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SIGMA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 13:0 – SIGMA[13:0] Coefficient of Degree x in the SIGMA Polynomial
SIGMAx belongs to the finite field GF(2^13) when the sector size is set to 512 bytes.
SIGMAx belongs to the finite field GF(2^14) when the sector size is set to 1024 bytes.

Downloaded from Arrow.com.

35.5.12 PMERRLOC Error Location x Register

Name: PMERRLOC_ELx
Offset: 0x8C + x*0x04 [x=0..23]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			ERRLOCN[13:8]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ERRLOCN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 13:0 – ERRLOCN[13:0] Error Position within the Set {sector area, spare area}.
ERRLOCN points to 0 when the first bit of the main area is corrupted.
If the sector size is set to 512 bytes, ERRLOCN points to 4096 when the last bit of the sector area is corrupted.
If the sector size is set to 1024 bytes, ERRLOCN points to 8192 when the last bit of the sector area is corrupted.
If the sector size is set to 512 bytes, ERRLOCN points to 4097 when the first bit of the spare area is corrupted.
If the sector size is set to 1024 bytes, ERRLOCN points to 8193 when the first bit of the spare area is corrupted.

Downloaded from Arrow.com.

36. DMA Controller (XDMAC)

36.1 Description

The DMA Controller (XDMAC) is a AHB-protocol central direct memory access controller. It performs peripheral data transfer and memory move operations over one or two bus ports through the unidirectional communication channel. Each channel is fully programmable and provides both peripheral or memory-to-memory transfers. The channel features are configurable at implementation.

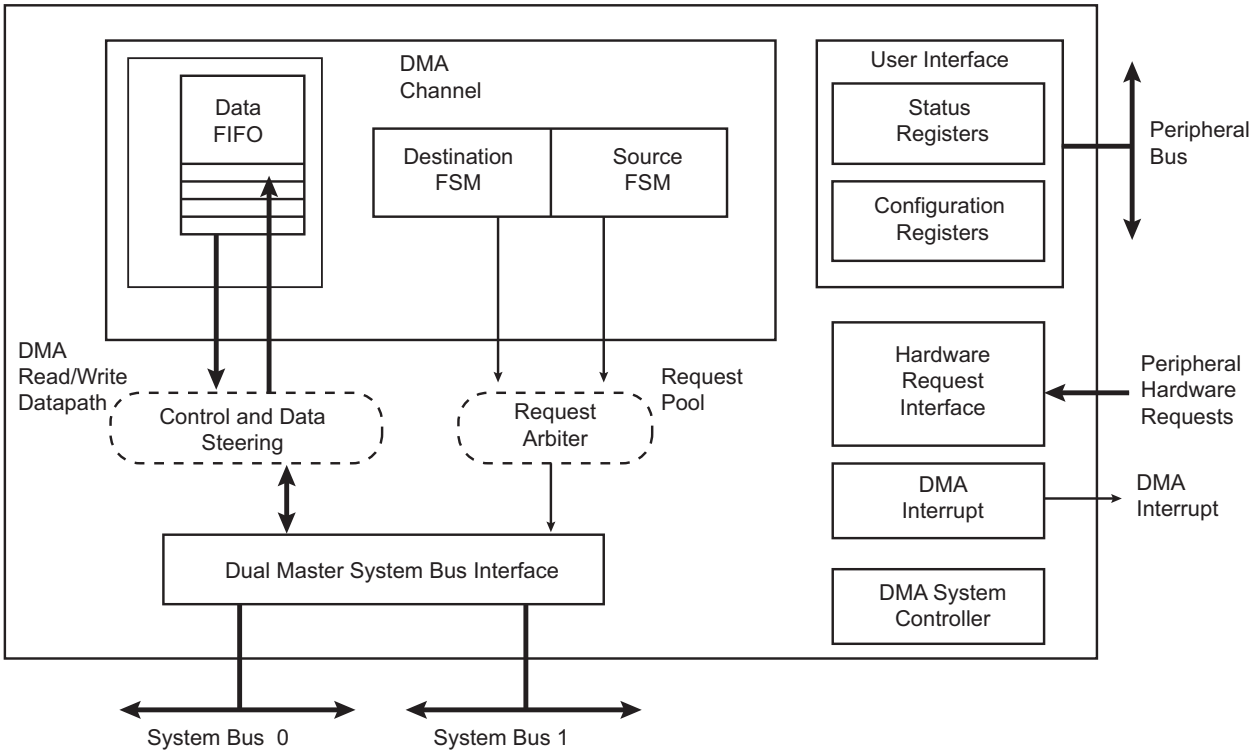
36.2 Embedded Characteristics

- 2 System Bus Master Interfaces
- 16 DMA Channels
- 51 Hardware Requests
- 4 Kbytes Embedded FIFO
- Supports Peripheral-to-Memory, Memory-to-Peripheral, or Memory-to-Memory Transfer Operations
- Peripheral DMA Operation Runs on Bytes (8-bit), Half-Word (16-bit) and Word (32-bit)
- Memory DMA Operation Runs on Bytes (8 bit), Half-Word (16-bit) and Word (32 -bit)
- Supports Hardware and Software Initiated Transfers
- Supports Linked List Operations
- Supports Incrementing or Fixed Addressing Mode
- Supports Programmable Independent Data Striding for Source and Destination
- Supports Programmable Independent Microblock Striding for Source and Destination
- Configurable Priority Group and Arbitration Policy
- Programmable AHB Burst Length
- Configuration Interface on Peripheral Bus
- XDMAC Architecture Includes Multiport FIFO
- Supports Multiple View Channel Descriptor
- Automatic Flush of Channel Trailing Bytes
- Automatic Coarse-Grain and Fine-Grain Clock Gating
- Hardware Acceleration of Memset Pattern
- Supports Configurable Quality of Service per Channel

Downloaded from Arrow.com.

36.3 Block Diagram

Figure 36-1. XDMAC Block Diagram



36.4 DMA Controller Peripheral Connections

DMA Controller 0 manages transfers between peripherals and memory, and receives the triggers from the peripherals listed in the following table.

Table 36-1. DMA Channels Definitions (XDMAC0)

Instance Name	Channel T/R	Interface Number
FLEXCOM0	Transmit	0
FLEXCOM0	Receive	1
FLEXCOM1	Transmit	2
FLEXCOM1	Receive	3
FLEXCOM2	Transmit	4
FLEXCOM2	Receive	5
FLEXCOM3	Transmit	6
FLEXCOM3	Receive	7
FLEXCOM4	Transmit	8
FLEXCOM4	Receive	9
FLEXCOM5	Transmit	10
FLEXCOM5	Receive	11

Downloaded from Arrow.com.

.....continued

Instance Name	Channel T/R	Interface Number
FLEXCOM6	Transmit	12
FLEXCOM6	Receive	13
FLEXCOM7	Transmit	14
FLEXCOM7	Receive	15
FLEXCOM8	Transmit	16
FLEXCOM8	Receive	17
FLEXCOM9	Transmit	18
FLEXCOM9	Receive	19
FLEXCOM10	Transmit	20
FLEXCOM10	Receive	21
FLEXCOM11	Transmit	22
FLEXCOM11	Receive	23
FLEXCOM12	Transmit	24
FLEXCOM12	Receive	25
QSPI	Transmit	26
QSPI	Receive	27
DBGU	Transmit	28
DBGU	Receive	29
TDES	Receive	30
TDES	Transmit	31
AES	Transmit	32
AES	Receive	33
SHA	Transmit	34
CLASSD	Transmit	35
I2SMCC	Transmit	36
I2SMCC	Receive	37
SSC	Transmit	38
SSC	Receive	39
ADC	Receive	40
TC0	Receive	41
TC1	Receive	42
TC1_CPA	Compare Counter A, Timer Channel 1	43
TC4_CPA	Compare Counter A, Timer Channel 4	44
TC1_CPB	Compare Counter B, Timer Channel 1	45
TC4_CPB	Compare Counter B Timer Channel 4	46

Downloaded from Arrow.com.

.....continued

Instance Name	Channel T/R	Interface Number
TC1_CPC	Compare Counter C, Timer Channel 1	47
TC4_CPC	Compare Counter C, Timer Channel 4	48
TC1_ETRG	External Event trigger, timer channel 1 for TC1_ETRG	49
TC4_ETRG	External Event trigger, timer channel 1 for TC4_ETRG	50

36.5 Functional Description

36.5.1 Basic Definitions

Source Peripheral: Slave device, memory mapped on the interconnection network, from where the XDMAC reads data. The source peripheral teams up with a destination peripheral to form a channel. A data read operation is scheduled when the peripheral transfer request is asserted.

Destination Peripheral: Slave device, memory mapped on the interconnection network, to which the XDMAC writes. A write data operation is scheduled when the peripheral transfer request is asserted.

Channel: The data movement between source and destination creates a logical channel.

Stride: Number of address locations between successive elements/data measured in bytes.

Transfer Type: The transfer is hardware-synchronized when it is paced by the peripheral hardware request, otherwise the transfer is self-triggered (memory to memory transfer).

XDMAC Master Transfer: The Master Transfer is composed of a linked list of blocks. The channel address, control and configuration registers can be modified at the inter block boundary. The descriptor structure modifies the channel registers conditionally. Interrupts can be generated on a per block basis or when the end of linked list event occurs.

XDMAC Block: An XDMAC block is composed of a programmable number of microblocks. The channel configuration registers remain unchanged at the inter microblock boundary. The source and destination addresses are conditionally updated with a programmable signed number.

XDMAC Microblock: The microblock is composed of a programmable number of data. The channel configuration registers remain unchanged at the data boundary. The data address may be fixed (a FIFO location, a peripheral transmit or receive register), incrementing (a memory-mapped area) by a programmable signed number.

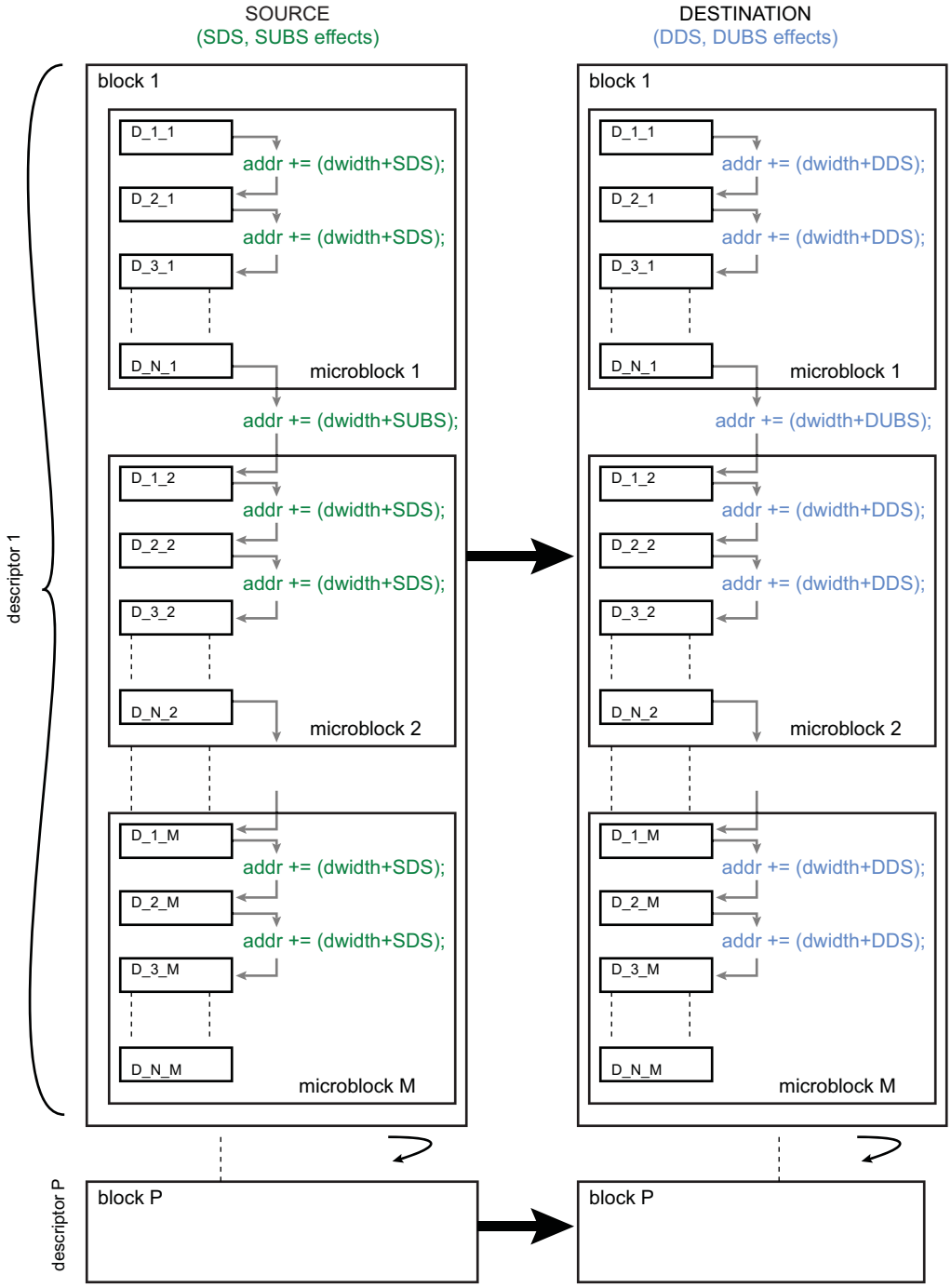
XDMAC Burst and Incomplete Burst: In order to improve the overall performance when accessing dynamic external memory, burst access is mandatory. Each data of the microblock is considered as a part of a memory burst. The programmable burst value indicates the largest memory burst allowed on a per channel basis. When the microblock length is not an integral multiple of the burst size, an incomplete burst is performed to read or write the last trailing bytes.

XDMAC Chunk and Incomplete Chunk: When a peripheral synchronized transfer is activated, the microblock splits into a number of data chunks. The chunk size is programmable. The larger the chunk is, the better the performance is. When the transfer size is not a multiple of the chunk size, the last chunk may be incomplete.

Downloaded from Arrow.com.

36.5.2 Data Striding Diagram

Figure 36-2. Data Striding Diagram



Downloaded from Arrow.com.

36.5.3 Transfer Hierarchy Diagrams

Figure 36-3. XDMAC Memory Transfer Hierarchy

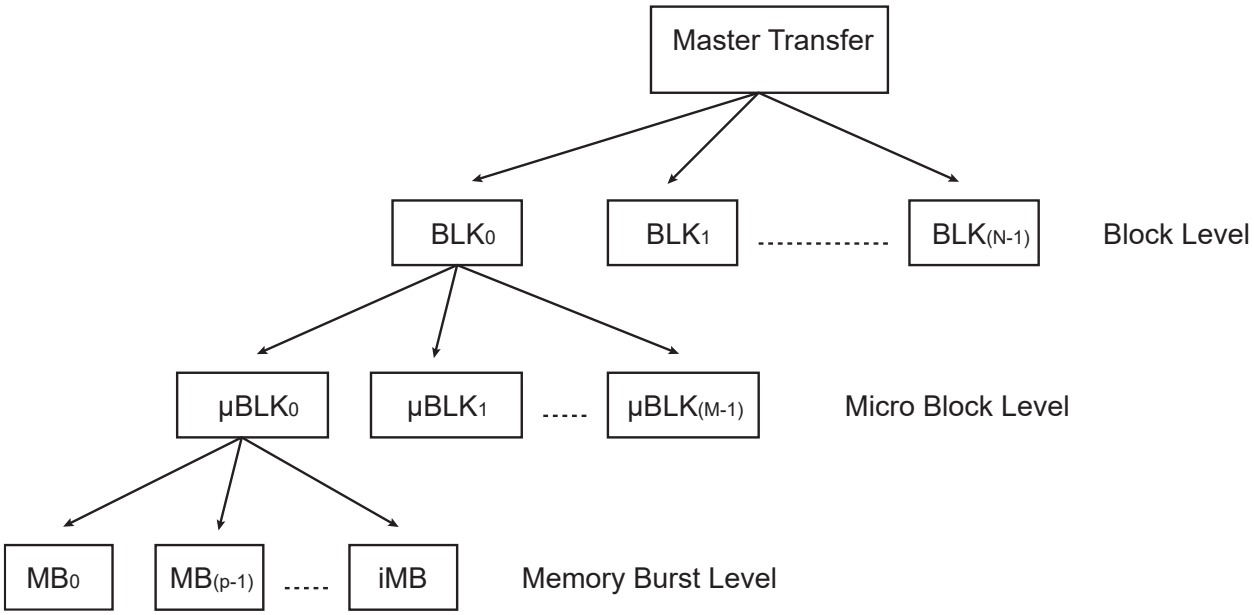
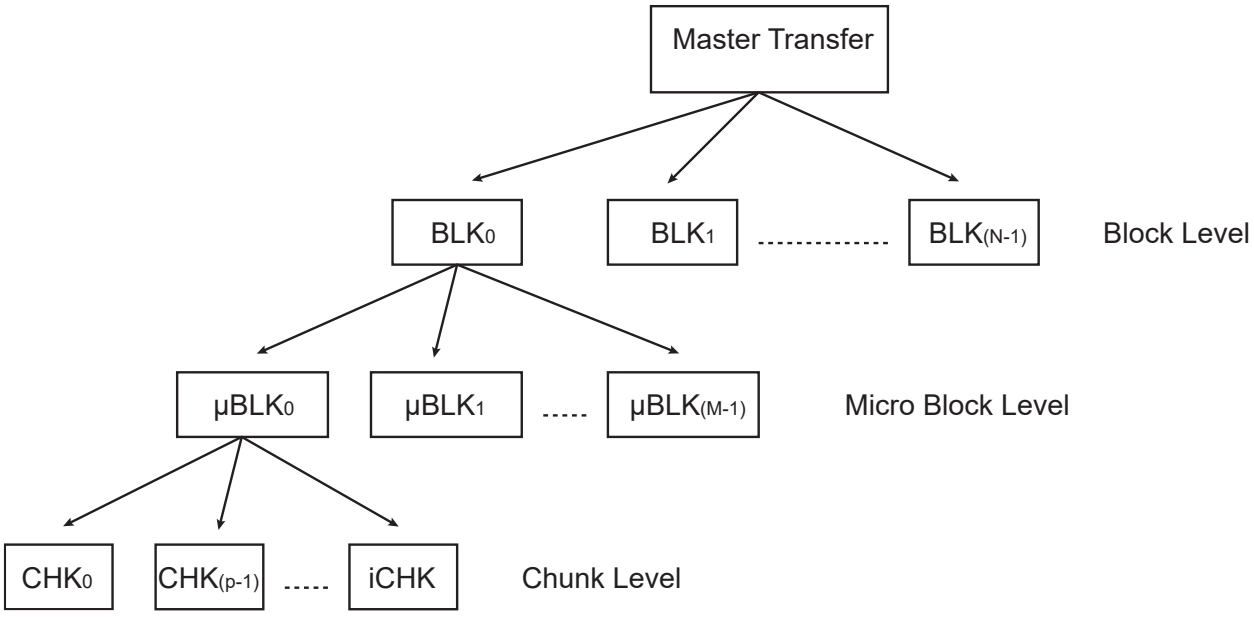


Figure 36-4. XDAMC Peripheral Transfer Hierarchy



36.5.4 Peripheral Synchronized Transfer

A peripheral hardware request interface is used to control the pace of the chunk transfer. When a peripheral is ready to transmit or receive a chunk of data, it asserts its request line and the DMA Controller transfers a data to or from the memory to the peripheral.

36.5.4.1 Software Triggered Synchronized Transfer

The Peripheral hardware request can be software controlled using the SWREQ field of the XDMAC Global Channel Software Request Register (XDMAC_GSWR). The peripheral synchronized transfer is paced using a processor write access in the XDMAC_GSWR. Each bit of that register triggers a transfer request. The XDMAC Global Channel Software Request Status Register (XDMAC_GSWS) indicates the status of the request; when set, the request is still pending.

Downloaded from Arrow.com.

36.5.5 XDMAC Transfer Software Operation

Note:
When a memory-to-memory transfer is performed, configure the field XDMAC_CCx.PERID (where 'x' is the index of the channel used for transfer) to an unused peripheral ID (refer to table "Peripheral Identifiers").

36.5.5.1 Single Block Transfer With Single Microblock

1. Read the XDMAC Global Channel Status Register (XDMAC_GS) to select a free channel.
2. Clear the pending Interrupt Status bit(s) by reading the selected XDMAC Channel x Interrupt Status Register (XDMAC_CISx).
3. Write the XDMAC Channel x Source Address Register (XDMAC_CSAx) for channel x.
4. Write the XDMAC Channel x Destination Address Register (XDMAC_CDAx) for channel x.
5. Program field UBLEN in the XDMAC Channel x Microblock Control Register (XDMAC_CUBCx) with the number of data.
6. Program the XDMAC Channel x Configuration Register (XDMAC_CCx):
    6.1. Clear XDMAC_CCx.TYPE for a memory-to-memory transfer, otherwise set this bit.
    6.2. Configure XDMAC_CCx.MBSIZE to the memory burst size used.
    6.3. Configure XDMAC_CCx.SAM and DAM to Memory Addressing mode.
    6.4. Configure XDMAC_CCx.DSYNC to select the peripheral transfer direction.
    6.5. Configure XDMAC_CCx.CSIZE to configure the channel chunk size (only relevant for peripheral synchronized transfer).
    6.6. Configure XDMAC_CCx.DWIDTH to configure the transfer data width.
    6.7. Configure XDMAC_CCx.SIF, XDMAC_CCx.DIF to configure the master interface used to read data and write data, respectively.
    6.8. Configure XDMAC_CCx.PERID to select the active hardware request line (only relevant for a peripheral synchronized transfer).
    6.9. Set XDMAC_CCx.SWREQ to use a software request (only relevant for a peripheral synchronized transfer).
7. Clear the following five registers:
    – XDMAC Channel x Next Descriptor Control Register (XDMAC_CNDCx)
    – XDMAC Channel x Block Control Register (XDMAC_CBCx)
    – XDMAC Channel x Data Stride Memory Set Pattern Register (XDMAC_CDS_MSPx)
    – XDMAC Channel x Source Microblock Stride Register (XDMAC_CSUSx)
    – XDMAC Channel x Destination Microblock Stride Register (XDMAC_CDUSx)
    This indicates that the linked list is disabled, there is only one block and striding is disabled.
8. Enable the Microblock interrupt by writing a '1' to bit BIE in the XDMAC Channel x Interrupt Enable Register (XDMAC_CIEx). Enable the Channel x Interrupt Enable bit by writing a '1' to bit IEx in the XDMAC Global Interrupt Enable Register (XDMAC_GIE).
9. Enable channel x by writing a '1' to bit ENx in the XDMAC Global Channel Enable Register (XDMAC_GE). XDMAC_GS.STx (XDMAC Channel x Status bit) is set by hardware.
10. Once completed, the DMA channel sets XDMAC_CISx.BIS (End of Block Interrupt Status bit) and generates an interrupt. XDMAC_GS.STx is cleared by hardware. The software can either wait for an interrupt or poll the channel status bit.

36.5.5.2 Single Block Transfer With Multiple Microblock

1. Read the XDMAC_GS register to choose a free channel.
2. Clear the pending Interrupt Status bit by reading the chosen XDMAC_CISx register.
3. Write the XDMAC_CSAx register for channel x.
4. Write the XDMAC_CDAx register for channel x.
5. Program XDMAC_CUBCx.UBLEN with the number of data.
6. Program XDMAC_CCx register (see "Single Block Transfer With Single Microblock").
7. Program XDMAC_CBCx.BLEN with the number of microblocks of data.
8. Clear the following registers:
    – XDMAC_CNDCx

Downloaded from Arrow.com.

– XDMAC_CDS_MSPx
– XDMAC_CSUSx XDMAC_CDUSx
This indicates that the linked list is disabled and striding is disabled.

9. Enable the Block interrupt by writing a ‘1’ to XDMAC_CIEx.BIE, enable the Channel x Interrupt Enable bit by writing a ‘1’ to XDMAC_GIEx.IEx.
10. Enable channel x by writing a ‘1’ to the XDMAC_GE.ENx. XDMAC_GS.STx is set by hardware.
11. Once completed, the DMA channel sets XDMAC_CISx.BIS (End of Block Interrupt Status bit) and generates an interrupt. XDMAC_GS.STx is cleared by hardware. The software can either wait for an interrupt or poll the channel status bit.

36.5.5.3 Master Transfer

1. Read the XDMAC_GS register to choose a free channel.
2. Clear the pending Interrupt Status bit by reading the chosen XDMAC_CISx register.
3. Build a linked list of transfer descriptors in memory. The descriptor view is programmable on a per descriptor basis. The linked list items structure must be word aligned. MBR_UBC.NDE must be configured to 0 in the last descriptor to terminate the list.
4. Configure field NDA in the XDMAC Channel x Next Descriptor Address Register (XDMAC_CNDAx) with the first descriptor address and bit XDMAC_CNDAx.NDAIF with the master interface identifier.
5. Configure the XDMAC_CNDCx register:
   5.1. Set XDMAC_CNDCx.NDE to enable the descriptor fetch.
   5.2. Set XDMAC_CNDCx.NDSUP to update the source address at the descriptor fetch time, otherwise clear this bit.
   5.3. Set XDMAC_CNDCx.NDDUP to update the destination address at the descriptor fetch time, otherwise clear this bit.
   5.4. Configure XDMAC_CNDCx.NDVIEW to define the length of the first descriptor.
6. Enable the End of Linked List interrupt by writing a ‘1’ to XDMAC_CIEx.LIE.
7. Enable channel x by writing a ‘1’ to XDMAC_GE.ENx. XDMAC_GS.STx is set by hardware.
8. Once completed, the DMA channel sets XDMAC_CISx.BIS (End of Block Interrupt Status bit) and generates an interrupt. XDMAC_GS.STx is cleared by hardware. The software can either wait for an interrupt or poll the channel status bit.

36.5.5.4 Disabling A Channel Before Transfer Completion

Under normal operation, the software enables a channel by writing a ‘1’ to XDMAC_GE.ENx, then the hardware disables a channel on transfer completion by clearing bit XDMAC_GS.STx. To disable a channel, write a ‘1’ to bit XDMAC_GD.DIx and poll the XDMAC_GS register.

36.6 Linked List Descriptor Operation

36.6.1 Linked List Descriptor View

36.6.1.1 Channel Next Descriptor View 0–3 Structures

Table 36-2. Channel Next Descriptor View 0–3 Structures

Channel Next Descriptor	Offset	Structure member	Name
View 0 Structure	DSCR_ADDR+0x00	Next Descriptor Address Member	MBR_NDA
	DSCR_ADDR+0x04	Microblock Control Member	MBR_UBC
	DSCR_ADDR+0x08	Transfer Address Member	MBR_TA

Downloaded from Arrow.com.

.....continued

Channel Next Descriptor	Offset	Structure member	Name
View 1 Structure	DSCR_ADDR+0x00	Next Descriptor Address Member	MBR_NDA
	DSCR_ADDR+0x04	Microblock Control Member	MBR_UBC
	DSCR_ADDR+0x08	Source Address Member	MBR_SA
	DSCR_ADDR+0x0C	Destination Address Member	MBR_DA
View 2 Structure	DSCR_ADDR+0x00	Next Descriptor Address Member	MBR_NDA
	DSCR_ADDR+0x04	Microblock Control Member	MBR_UBC
	DSCR_ADDR+0x08	Source Address Member	MBR_SA
	DSCR_ADDR+0x0C	Destination Address Member	MBR_DA
	DSCR_ADDR+0x10	Configuration Register	MBR_CFG
View 3 Structure	DSCR_ADDR+0x00	Next Descriptor Address Member	MBR_NDA
	DSCR_ADDR+0x04	Microblock Control Member	MBR_UBC
	DSCR_ADDR+0x08	Source Address Member	MBR_SA
	DSCR_ADDR+0x0C	Destination Address Member	MBR_DA
	DSCR_ADDR+0x10	Configuration Member	MBR_CFG
	DSCR_ADDR+0x14	Block Control Member	MBR_BC
	DSCR_ADDR+0x18	Data Stride Member	MBR_DS
	DSCR_ADDR+0x1C	Source Microblock Stride Member	MBR_SUS
	DSCR_ADDR+0x20	Destination Microblock Stride Member	MBR_DUS

36.6.2 Descriptor Structure Members Description

Downloaded from Arrow.com.

36.6.2.1 Descriptor Structure Microblock Control Member

Name: MBR_UBC
Property: Read-only

Bit	31	30	29	28	27	26	25	24
		QOS[1:0]		NVIEW[1:0]		NDEN	NSEN	NDE
Access		R	R	R	R	R	R	R
Reset								

Bit	23	22	21	20	19	18	17	16
	UBLEN[23:16]							
Access	R	R	R	R	R	R	R	R
Reset								

Bit	15	14	13	12	11	10	9	8
	UBLEN[15:8]							
Access	R	R	R	R	R	R	R	R
Reset								

Bit	7	6	5	4	3	2	1	0
	UBLEN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset								

Bits 30:29 – QOS[1:0] Channel Quality of Service Level
This field indicates the current quality of service level for the channel. Refer to the section “MATRIX”.

Bits 28:27 – NVIEW[1:0] Next Descriptor View

Value	Name	Description
0	NDV0	Next Descriptor View 0
1	NDV1	Next Descriptor View 1
2	NDV2	Next Descriptor View 2
3	NDV3	Next Descriptor View 3

Bit 26 – NDEN Next Descriptor Destination Update

Value	Description
0	Destination parameters remain unchanged.
1	Destination parameters are updated when the descriptor is retrieved.

Bit 25 – NSEN Next Descriptor Source Update

Value	Description
0	Source parameters remain unchanged.
1	Source parameters are updated when the descriptor is retrieved.

Bit 24 – NDE Next Descriptor Enable

Value	Description
0	Descriptor fetch is disabled.
1	Descriptor fetch is enabled.

Bits 23:0 – UBLEN[23:0] Microblock Length
This field indicates the number of data in the microblock. The microblock contains UBLEN data.

Downloaded from Arrow.com.

36.7 XDMAC Maintenance Software Operations

36.7.1 Disabling a Channel

A disable channel request occurs when a write operation is performed in the XDMAC_GD register. If the channel is source peripheral synchronized (bit XDMAC_CCx.TYPE is set and bit XDMAC_CCx.DSYNC is cleared), then pending bytes (bytes located in the FIFO) are written to memory and bit XDMAC_CISx.DIS is set. If the channel is not source peripheral synchronized, the current channel transaction (read or write) is terminated and XDMAC_CISx.DIS is set. XDMAC_GS.STx is cleared by hardware when the current transfer is completed. The channel is no longer active and can be reused.

36.7.2 Suspending a Channel

A disable channel request occurs when a write operation is performed in the XDMAC_GD register. If the channel is source peripheral synchronized (bit XDMAC_CCx.TYPE is set and bit XDMAC_CCx.DSYNC is cleared), then pending bytes (bytes located in the FIFO) are written to memory and bit XDMAC_CISx.DIS is set. If the channel is not source peripheral synchronized, the current channel transaction (read or write) is terminated and XDMAC_CISx.DIS is set. XDMAC_GS.STx is cleared by hardware when the current transfer is completed. The channel is no longer active and can be reused.

36.7.3 Flushing a Channel

A FIFO flush command is issued by writing to the XDMAC_SWF register. The content of the FIFO is written to memory. XDMAC_CISx.FIS (End of Flush Interrupt Status bit) is set when the last byte is successfully transferred to memory. The channel is not disabled. The flush operation is not blocking, meaning that read operation can be scheduled during the flush write operation. The flush operation is only relevant for peripheral to memory transfer where pending peripheral bytes are buffered into the channel FIFO.

36.7.4 Maintenance Operation Priority

36.7.4.1 Disable Operation Priority

- When a disable request occurs on a suspended channel, the XDMAC_GWS.WSx (Channel x Write Suspend bit) is cleared. If the transfer is source peripheral synchronized, the pending bytes are drained to memory. The bit XDMAC_CISx.DIS is set.
- When a disable request follows a flush request, if the flush last transaction is not yet scheduled, the flush request is discarded and the disable procedure is applied. Bit XDMAC_CISx.FIS is not set. Bit XDMAC_CISx.DIS is set when the disable request is completed. If the flush request transaction is already scheduled, the XDMAC_CISx.FIS is set. XDMAC_CISx.DIS is also set when the disable request is completed.

36.7.4.2 Flush Operation Priority

- When a flush request occurs on a suspended channel, if there are pending bytes in the FIFO, they are written out to memory, XDMAC_CISx.FIS is set. If the FIFO is empty, XDMAC_CISx.FIS is also set.
- If the flush operation is performed after a disable request, the flush command is ignored. XDMAC_CISx.FIS is not set.

36.7.4.3 Suspend Operation Priority

If the suspend operation is performed after a disable request, the write suspend operation is ignored.

36.8 XDMAC Software Requirements

- Write operations to channel registers are not be performed in an active channel after the channel is enabled. If any channel parameters must be reprogrammed, this can only be done after disabling the XDMAC channel.
- XDMAC_CSAx and XDMAC_CDAx channel registers must be programmed with a byte, half-word or word aligned address depending on the Channel x Data Width field (DWIDTH) of the XDMAC Channel x Configuration Register. When a memory-to-peripheral transfer is performed, the XDMAC_CSAx address register has no alignment requirement.
- When a memory-to-memory transfer is performed, configure the field XDMAC_CCx.PERID (where ‘x’ is the index of the channel used for the transfer) to an unused peripheral ID (refer to table “Peripheral Identifiers”).

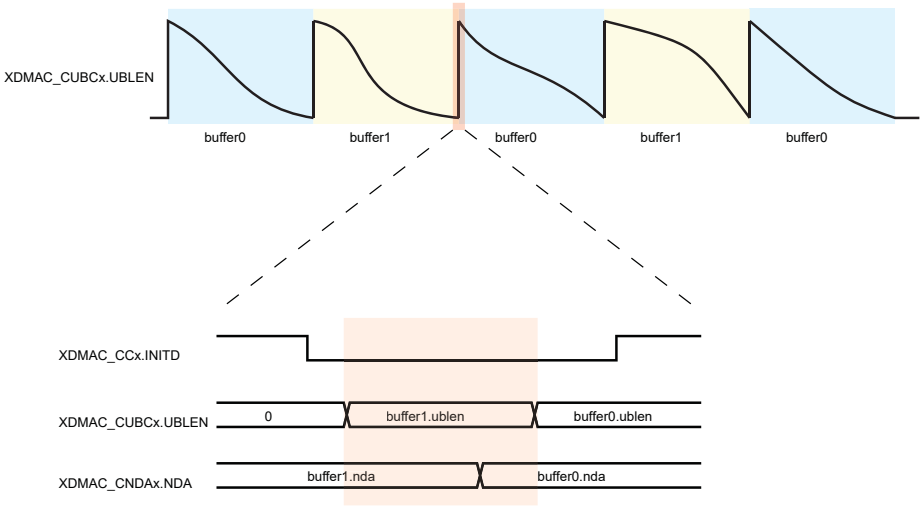
Downloaded from Arrow.com.

- When XDMAC_CC.INITD is set to 0, XDMAC_CUBC.UBLEN and XDMAC_CNDA.NDA field values are unreliable when the descriptor is being updated. The following procedure applies to get the buffer descriptor identifier and the residual bytes:

```
Read XDMAC_CNDAx.NDA(nda0)
Read XDMAC_CCx.INITD(initd0)
Read XDMAC_CCx.INITD(initd0)
Read XDMAC_CUBCx.UBLEN(ublen)
Read XDMAC_CCx.INITD(initd1)
Read XDMA_CNDAx.NDA(nda1)
If (nda0 == nda1 && initd0 == 1 && initd1 == 1).
Then the ublen is correct, the buffer id is nda.
Else retry
```

See the figure below.

Figure 36-5. INITD Timing Diagram



Downloaded from Arrow.com.

36.9 Register Summary

Offset	Name	Bit Pos.								
0x00	XDMAC_GTYPE	31:24								
		23:16		NB_REQ[6:0]						
		15:8	FIFO_SZ[10:3]							
		7:0	FIFO_SZ[2:0]			NB_CH[4:0]				
0x04	XDMAC_GCFG	31:24								
		23:16								
		15:8								BXKBEN
		7:0					CGDISIF	CGDISFIFO	CGDISPIPE	CGDISREG
0x08	XDMAC_GWAC	31:24								
		23:16								
		15:8	PW3[3:0]				PW2[3:0]			
		7:0	PW1[3:0]				PW0[3:0]			
0x0C	XDMAC_GIE	31:24								
		23:16								
		15:8	IE15	IE14	IE13	IE12	IE11	IE10	IE9	IE8
		7:0	IE7	IE6	IE5	IE4	IE3	IE2	IE1	IE0
0x10	XDMAC_GID	31:24								
		23:16								
		15:8	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
		7:0	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
0x14	XDMAC_GIM	31:24								
		23:16								
		15:8	IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8
		7:0	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
0x18	XDMAC_GIS	31:24								
		23:16								
		15:8	IS15	IS14	IS13	IS12	IS11	IS10	IS9	IS8
		7:0	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
0x1C	XDMAC_GE	31:24								
		23:16								
		15:8	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8
		7:0	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
0x20	XDMAC_GD	31:24								
		23:16								
		15:8	DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8
		7:0	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
0x24	XDMAC_GS	31:24								
		23:16								
		15:8	ST15	ST14	ST13	ST12	ST11	ST10	ST9	ST8
		7:0	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
0x28	XDMAC_GRS	31:24								
		23:16								
		15:8	RS15	RS14	RS13	RS12	RS11	RS10	RS9	RS8
		7:0	RS7	RS6	RS5	RS4	RS3	RS2	RS1	RS0
0x2C	XDMAC_GWS	31:24								
		23:16								
		15:8	WS15	WS14	WS13	WS12	WS11	WS10	WS9	WS8
		7:0	WS7	WS6	WS5	WS4	WS3	WS2	WS1	WS0
0x30	XDMAC_GRWS	31:24								
		23:16								
		15:8	RWS15	RWS14	RWS13	RWS12	RWS11	RWS10	RWS9	RWS8
		7:0	RWS7	RWS6	RWS5	RWS4	RWS3	RWS2	RWS1	RWS0
0x34	XDMAC_GRWR	31:24								
		23:16								
		15:8	RWR15	RWR14	RWR13	RWR12	RWR11	RWR10	RWR9	RWR8
		7:0	RWR7	RWR6	RWR5	RWR4	RWR3	RWR2	RWR1	RWR0

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x38	XDMAC_GSWR	31:24								
		23:16								
		15:8	SWREQ15	SWREQ14	SWREQ13	SWREQ12	SWREQ11	SWREQ10	SWREQ9	SWREQ8
		7:0	SWREQ7	SWREQ6	SWREQ5	SWREQ4	SWREQ3	SWREQ2	SWREQ1	SWREQ0
0x3C	XDMAC_GSWS	31:24								
		23:16								
		15:8	SWRS15	SWRS14	SWRS13	SWRS12	SWRS11	SWRS10	SWRS9	SWRS8
		7:0	SWRS7	SWRS6	SWRS5	SWRS4	SWRS3	SWRS2	SWRS1	SWRS0
0x40	XDMAC_GSWF	31:24								
		23:16								
		15:8	SWF15	SWF14	SWF13	SWF12	SWF11	SWF10	SWF9	SWF8
		7:0	SWF7	SWF6	SWF5	SWF4	SWF3	SWF2	SWF1	SWF0
0x44 ... 0x4F	Reserved									
0x50	XDMAC_CIE0	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x54	XDMAC_CID0	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x58	XDMAC_CIM0	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x5C	XDMAC_CIS0	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x60	XDMAC_CSA0	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x64	XDMAC_CDA0	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x68	XDMAC_CNDA0	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x6C	XDMAC_CNDC0	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x70	XDMAC_CUBC0	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x74	XDMAC_CBC0	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x78	XDMAC_CC0	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x7C	XDMAC_CDS_MSP0	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x80	XDMAC_CSUS0	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x84	XDMAC_CDUS0	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x88 ... 0x8F	Reserved									
0x90	XDMAC_CIE1	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x94	XDMAC_CID1	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x98	XDMAC_CIM1	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x9C	XDMAC_CIS1	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0xA0	XDMAC_CSA1	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0xA4	XDMAC_CDA1	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0xA8	XDMAC_CNDA1	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0xAC	XDMAC_CNDC1	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0xB0	XDMAC_CUBC1	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0xB4	XDMAC_CBC1	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0xB8	XDMAC_CC1	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0xBC	XDMAC_CDS_MSP1	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0xC0	XDMAC_CSUS1	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0xC4	XDMAC_CDUS1	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0xC8 ... 0xCF	Reserved									
0xD0	XDMAC_CIE2	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0xD4	XDMAC_CID2	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0xD8	XDMAC_CIM2	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0xDC	XDMAC_CIS2	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0xE0	XDMAC_CSA2	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0xE4	XDMAC_CDA2	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0xE8	XDMAC_CNDA2	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0xEC	XDMAC_CNDC2	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0xF0	XDMAC_CUBC2	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0xF4	XDMAC_CBC2	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0xF8	XDMAC_CC2	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0xFC	XDMAC_CDS_MSP2	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0100	XDMAC_CSUS2	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0104	XDMAC_CDUS2	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0108 ... 0x010F	Reserved									
0x0110	XDMAC_CIE3	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0114	XDMAC_CID3	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0118	XDMAC_CIM3	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x011C	XDMAC_CIS3	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0120	XDMAC_CSA3	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0124	XDMAC_CDA3	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0128	XDMAC_CNDA3	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x012C	XDMAC_CNDC3	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0130	XDMAC_CUBC3	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0134	XDMAC_CBC3	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0138	XDMAC_CC3	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x013C	XDMAC_CDS_MSP3	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0140	XDMAC_CSUS3	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0144	XDMAC_CDUS3	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0148 ... 0x014F	Reserved									
0x0150	XDMAC_CIE4	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0154	XDMAC_CID4	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0158	XDMAC_CIM4	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x015C	XDMAC_CIS4	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0160	XDMAC_CSA4	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0164	XDMAC_CDA4	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0168	XDMAC_CNDA4	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x016C	XDMAC_CNDC4	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0170	XDMAC_CUBC4	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0174	XDMAC_CBC4	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0178	XDMAC_CC4	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x017C	XDMAC_CDS_MSP4	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0180	XDMAC_CSUS4	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0184	XDMAC_CDUS4	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0188 ... 0x018F	Reserved									
0x0190	XDMAC_CIE5	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0194	XDMAC_CID5	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0198	XDMAC_CIM5	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x019C	XDMAC_CIS5	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x01A0	XDMAC_CSA5	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x01A4	XDMAC_CDA5	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x01A8	XDMAC_CNDA5	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x01AC	XDMAC_CNDC5	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x01B0	XDMAC_CUBC5	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x01B4	XDMAC_CBC5	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x01B8	XDMAC_CC5	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x01BC	XDMAC_CDS_MSP5	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x01C0	XDMAC_CSUS5	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x01C4	XDMAC_CDUS5	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x01C8 ... 0x01CF	Reserved									
0x01D0	XDMAC_CIE6	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x01D4	XDMAC_CID6	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x01D8	XDMAC_CIM6	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x01DC	XDMAC_CIS6	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x01E0	XDMAC_CSA6	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x01E4	XDMAC_CDA6	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x01E8	XDMAC_CNDA6	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x01EC	XDMAC_CNDC6	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x01F0	XDMAC_CUBC6	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x01F4	XDMAC_CBC6	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x01F8	XDMAC_CC6	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x01FC	XDMAC_CDS_MSP6	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0200	XDMAC_CSUS6	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0204	XDMAC_CDUS6	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0208 ... 0x020F	Reserved									
0x0210	XDMAC_CIE7	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0214	XDMAC_CID7	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0218	XDMAC_CIM7	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x021C	XDMAC_CIS7	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0220	XDMAC_CSA7	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0224	XDMAC_CDA7	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0228	XDMAC_CNDA7	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x022C	XDMAC_CNDC7	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0230	XDMAC_CUBC7	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0234	XDMAC_CBC7	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0238	XDMAC_CC7	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x023C	XDMAC_CDS_MSP7	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0240	XDMAC_CSUS7	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0244	XDMAC_CDUS7	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0248 ... 0x024F	Reserved									
0x0250	XDMAC_CIE8	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0254	XDMAC_CID8	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0258	XDMAC_CIM8	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x025C	XDMAC_CIS8	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0260	XDMAC_CSA8	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0264	XDMAC_CDA8	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0268	XDMAC_CNDA8	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x026C	XDMAC_CNDC8	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0270	XDMAC_CUBC8	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0274	XDMAC_CBC8	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0278	XDMAC_CC8	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x027C	XDMAC_CDS_MSP8	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0280	XDMAC_CSUS8	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0284	XDMAC_CDUS8	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0288 ... 0x028F	Reserved									
0x0290	XDMAC_CIE9	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0294	XDMAC_CID9	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0298	XDMAC_CIM9	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x029C	XDMAC_CIS9	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x02A0	XDMAC_CSA9	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x02A4	XDMAC_CDA9	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x02A8	XDMAC_CNDA9	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x02AC	XDMAC_CNDC9	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x02B0	XDMAC_CUBC9	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x02B4	XDMAC_CBC9	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x02B8	XDMAC_CC9	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x02BC	XDMAC_CDS_MSP9	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x02C0	XDMAC_CSUS9	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x02C4	XDMAC_CDUS9	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x02C8 ... 0x02CF	Reserved									
0x02D0	XDMAC_CIE10	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x02D4	XDMAC_CID10	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x02D8	XDMAC_CIM10	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x02DC	XDMAC_CIS10	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x02E0	XDMAC_CSA10	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x02E4	XDMAC_CDA10	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x02E8	XDMAC_CNDA10	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x02EC	XDMAC_CNDC10	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x02F0	XDMAC_CUBC10	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x02F4	XDMAC_CBC10	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x02F8	XDMAC_CC10	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x02FC	XDMAC_CDS_MSP10	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0300	XDMAC_CSUS10	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0304	XDMAC_CDUS10	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0308 ... 0x030F	Reserved									
0x0310	XDMAC_CIE11	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0314	XDMAC_CID11	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0318	XDMAC_CIM11	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x031C	XDMAC_CIS11	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0320	XDMAC_CSA11	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0324	XDMAC_CDA11	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0328	XDMAC_CNDA11	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x032C	XDMAC_CNDC11	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0330	XDMAC_CUBC11	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0334	XDMAC_CBC11	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0338	XDMAC_CC11	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x033C	XDMAC_CDS_MSP11	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0340	XDMAC_CSUS11	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0344	XDMAC_CDUS11	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0348 ... 0x034F	Reserved									
0x0350	XDMAC_CIE12	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0354	XDMAC_CID12	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0358	XDMAC_CIM12	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x035C	XDMAC_CIS12	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0360	XDMAC_CSA12	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0364	XDMAC_CDA12	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0368	XDMAC_CNDA12	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x036C	XDMAC_CNDC12	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0370	XDMAC_CUBC12	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0374	XDMAC_CBC12	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0378	XDMAC_CC12	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x037C	XDMAC_CDS_MSP12	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0380	XDMAC_CSUS12	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0384	XDMAC_CDUS12	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0388 ... 0x038F	Reserved									
0x0390	XDMAC_CIE13	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0394	XDMAC_CID13	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0398	XDMAC_CIM13	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x039C	XDMAC_CIS13	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x03A0	XDMAC_CSA13	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x03A4	XDMAC_CDA13	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x03A8	XDMAC_CNDA13	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x03AC	XDMAC_CNDC13	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x03B0	XDMAC_CUBC13	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x03B4	XDMAC_CBC13	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x03B8	XDMAC_CC13	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x03BC	XDMAC_CDS_MSP13	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x03C0	XDMAC_CSUS13	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x03C4	XDMAC_CDUS13	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x03C8 ... 0x03CF	Reserved									
0x03D0	XDMAC_CIE14	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x03D4	XDMAC_CID14	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x03D8	XDMAC_CIM14	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x03DC	XDMAC_CIS14	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x03E0	XDMAC_CSA14	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x03E4	XDMAC_CDA14	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x03E8	XDMAC_CNDA14	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x03EC	XDMAC_CNDC14	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x03F0	XDMAC_CUBC14	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x03F4	XDMAC_CBC14	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x03F8	XDMAC_CC14	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x03FC	XDMAC_CDS_MSP14	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0400	XDMAC_CSUS14	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0404	XDMAC_CDUS14	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0408 ... 0x040F	Reserved									
0x0410	XDMAC_CIE15	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0414	XDMAC_CID15	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0418	XDMAC_CIM15	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x041C	XDMAC_CIS15	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0420	XDMAC_CSA15	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0424	XDMAC_CDA15	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0428	XDMAC_CNDA15	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x042C	XDMAC_CNDC15	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0430	XDMAC_CUBC15	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0434	XDMAC_CBC15	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0438	XDMAC_CC15	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x043C	XDMAC_CDS_MSP15	31:24	DDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		7:0	SDS_MSP[7:0]							
0x0440	XDMAC_CSUS15	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0444	XDMAC_CDUS15	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							

Downloaded from Arrow.com.

36.9.1 XDMAC Global Type Register

Name: XDMAC_GTYPE
Offset: 0x00
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		NB_REQ[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FIFO_SZ[10:3]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FIFO_SZ[2:0]			NB_CH[4:0]				
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 22:16 – NB_REQ[6:0] Number of Peripheral Requests Minus One

Bits 15:5 – FIFO_SZ[10:0] Number of Bytes

Bits 4:0 – NB_CH[4:0] Number of Channels Minus One

Downloaded from Arrow.com.

36.9.2 XDMAC Global Configuration Register

Name: XDMAC_GCFG
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 8 – BXKBEN Boundary X Kilobyte Enable

Value	Description
0	XDMAC generates a non-sequential attribute on the system bus when address crosses the 1-Kilobyte boundary with a burst access.
1	XDMAC does not generate a non-sequential attribute on the system bus when address crosses the 1-Kilobyte boundary with a burst access.

Bit 3 – CGDISIF Bus Interface Clock Gating Disable

Value	Description
0	The automatic clock gating is enabled for the system bus interface.
1	The automatic clock gating is disabled for the system bus interface.

Bit 2 – CGDISFIFO FIFO Clock Gating Disable

Value	Description
0	The automatic clock gating is enabled for the main FIFO.
1	The automatic clock gating is disabled for the main FIFO.

Bit 1 – CGDISPIPE Pipeline Clock Gating Disable

Value	Description
0	The automatic clock gating is enabled for the main pipeline.
1	The automatic clock gating is disabled for the main pipeline.

Bit 0 – CGDISREG Configuration Registers Clock Gating Disable

Value	Description
0	The automatic clock gating is enabled for the configuration registers.
1	The automatic clock gating is disabled for the configuration registers.

Downloaded from Arrow.com.

36.9.3 XDMAC Global Weighted Arbiter Configuration Register

Name: XDMAC_GWAC
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PW3[3:0]				PW2[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PW1[3:0]				PW0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:12 – PW3[3:0] Pool Weight 3
This field indicates the weight of pool 3 in the arbitration scheme of the DMA scheduler.

Bits 11:8 – PW2[3:0] Pool Weight 2
This field indicates the weight of pool 2 in the arbitration scheme of the DMA scheduler.

Bits 7:4 – PW1[3:0] Pool Weight 1
This field indicates the weight of pool 1 in the arbitration scheme of the DMA scheduler.

Bits 3:0 – PW0[3:0] Pool Weight 0
This field indicates the weight of pool 0 in the arbitration scheme of the DMA scheduler.

Downloaded from Arrow.com.

36.9.4 XDMAC Global Interrupt Enable Register

Name: XDMAC_GIE
Offset: 0x0C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	IE15	IE14	IE13	IE12	IE11	IE10	IE9	IE8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	IE7	IE6	IE5	IE4	IE3	IE2	IE1	IE0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – IEx XDMAC Channel x Interrupt Enable

Value	Description
0	This bit has no effect. The Channel x Interrupt Mask bit (XDMAC_GIM.IMx) is not modified.
1	The corresponding mask bit is set. The XDMAC Channel x Interrupt Status register (XDMAC_GIS) can generate an interrupt.

Downloaded from Arrow.com.

36.9.5 XDMAC Global Interrupt Disable Register

Name: XDMAC_GID
Offset: 0x10
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – IDx XDMAC Channel x Interrupt Disable

Value	Description
0	This bit has no effect. The Channel x Interrupt Mask bit (XDMAC_GIM.IMx) is not modified.
1	The corresponding mask bit is reset. The Channel x Interrupt Status register interrupt (XDMAC_GIS) is masked.

Downloaded from Arrow.com.

36.9.6 XDMAC Global Interrupt Mask Register

Name: XDMAC_GIM
Offset: 0x14
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – IMx XDMAC Channel x Interrupt Mask

Value	Description
0	This bit indicates that the channel x interrupt source is masked. The interrupt line is not raised.
1	This bit indicates that the channel x interrupt source is unmasked.

Downloaded from Arrow.com.

36.9.7 XDMAC Global Interrupt Status Register

Name: XDMAC_GIS
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	IS15	IS14	IS13	IS12	IS11	IS10	IS9	IS8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – ISx XDMAC Channel x Interrupt Status

Value	Description
0	This bit indicates that either the interrupt source is masked at the channel level or no interrupt is pending for channel x.
1	This bit indicates that an interrupt is pending for the channel x.

Downloaded from Arrow.com.

36.9.8 XDMAC Global Channel Enable Register

Name: XDMAC_GE
Offset: 0x1C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – ENx XDMAC Channel x Enable

Value	Description
0	This bit has no effect.
1	Enables channel n. This operation is permitted if the Channel x Status bit (XDMAC_GS.STx) was read as '0'.

Downloaded from Arrow.com.

36.9.9 XDMAC Global Channel Disable Register

Name: XDMAC_GD
Offset: 0x20
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – DIx XDMAC Channel x Disable

Value	Description
0	This bit has no effect.
1	Disables channel x.

Downloaded from Arrow.com.

36.9.10 XDMAC Global Channel Status Register

Name: XDMAC_GS
Offset: 0x24
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ST15	ST14	ST13	ST12	ST11	ST10	ST9	ST8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – STx XDMAC Channel x Status

Value	Description
0	This bit indicates that the channel x is disabled.
1	This bit indicates that the channel x is enabled. If a channel disable request is issued, this bit remains asserted until pending transaction is completed.

Downloaded from Arrow.com.

36.9.11 XDMAC Global Channel Read Suspend Register

Name: XDMAC_GRS
Offset: 0x28
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RS15	RS14	RS13	RS12	RS11	RS10	RS9	RS8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RS7	RS6	RS5	RS4	RS3	RS2	RS1	RS0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – RSx XDMAC Channel x Read Suspend

Value	Description
0	The read channel is not suspended.
1	The source requests for channel n are no longer serviced by the system scheduler.

Downloaded from Arrow.com.

36.9.12 XDMAC Global Channel Write Suspend Register

Name: XDMAC_GWS
Offset: 0x2C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WS15	WS14	WS13	WS12	WS11	WS10	WS9	WS8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WS7	WS6	WS5	WS4	WS3	WS2	WS1	WS0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – WSx XDMAC Channel x Write Suspend

Value	Description
0	The write channel is not suspended.
1	Destination requests are no longer routed to the scheduler.

Downloaded from Arrow.com.

36.9.13 XDMAC Global Channel Read Write Suspend Register

Name: XDMAC_GRWS
Offset: 0x30
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RWS15	RWS14	RWS13	RWS12	RWS11	RWS10	RWS9	RWS8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	RWS7	RWS6	RWS5	RWS4	RWS3	RWS2	RWS1	RWS0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – RWSx XDMAC Channel x Read Write Suspend

Value	Description
0	No effect.
1	Read and write requests are suspended.

Downloaded from Arrow.com.

36.9.14 XDMAC Global Channel Read Write Resume Register

Name: XDMAC_GRWR
Offset: 0x34
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RWR15	RWR14	RWR13	RWR12	RWR11	RWR10	RWR9	RWR8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	RWR7	RWR6	RWR5	RWR4	RWR3	RWR2	RWR1	RWR0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – RWRx XDMAC Channel x Read Write Resume

Value	Description
0	No effect.
1	Read and write requests are serviced.

Downloaded from Arrow.com.

36.9.15 XDMAC Global Channel Software Request Register

Name: XDMAC_GSWR
Offset: 0x38
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SWREQ15	SWREQ14	SWREQ13	SWREQ12	SWREQ11	SWREQ10	SWREQ9	SWREQ8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	SWREQ7	SWREQ6	SWREQ5	SWREQ4	SWREQ3	SWREQ2	SWREQ1	SWREQ0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – SWREQx XDMAC Channel x Software Request

Value	Description
0	No effect.
1	Requests a DMA transfer for channel x.

Downloaded from Arrow.com.

36.9.16 XDMAC Global Channel Software Request Status Register

Name: XDMAC_GSWS
Offset: 0x3C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SWRS15	SWRS14	SWRS13	SWRS12	SWRS11	SWRS10	SWRS9	SWRS8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SWRS7	SWRS6	SWRS5	SWRS4	SWRS3	SWRS2	SWRS1	SWRS0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – SWRSx XDMAC Channel x Software Request Status

Value	Description
0	Channel x source request is serviced.
1	Channel x source request is pending.

Downloaded from Arrow.com.

36.9.17 XDMAC Global Channel Software Flush Request Register

Name: XDMAC_GSWF
Offset: 0x40
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SWF15	SWF14	SWF13	SWF12	SWF11	SWF10	SWF9	SWF8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	SWF7	SWF6	SWF5	SWF4	SWF3	SWF2	SWF1	SWF0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – SWFx XDMAC Channel x Software Flush Request

Value	Description
0	No effect.
1	Requests a DMA transfer flush for channel x. This bit is only relevant when the transfer is source peripheral synchronized.

Downloaded from Arrow.com.

36.9.18 XDMAC Channel x Interrupt Enable Register [x=0..15]

Name: XDMAC_CIE
Offset: 0x50 + n*0x40 [n=0..15]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bit 6 – ROIE Request Overflow Error Interrupt Enable

Value	Description
0	No effect.
1	Enables request overflow error interrupt.

Bit 5 – WBIE Write Bus Error Interrupt Enable

Value	Description
0	No effect.
1	Enables write bus error interrupt.

Bit 4 – RBIE Read Bus Error Interrupt Enable

Value	Description
0	No effect.
1	Enables read bus error interrupt.

Bit 3 – FIE End of Flush Interrupt Enable

Value	Description
0	No effect.
1	Enables end of flush interrupt.

Bit 2 – DIE End of Disable Interrupt Enable

Value	Description
0	No effect.
1	Enables end of disable interrupt.

Bit 1 – LIE End of Linked List Interrupt Enable

Value	Description
0	No effect.
1	Enables end of linked list interrupt.

Downloaded from Arrow.com.

Bit 0 – BIE End of Block Interrupt Enable

Value	Description
0	No effect.
1	Enables end of block interrupt.

Downloaded from Arrow.com.

36.9.19 XDMAC Channel x Interrupt Disable Register [x = 0..15]

Name: XDMAC_CID
Offset: 0x54 + n*0x40 [n=0..15]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		ROID	WBEID	RBEID	FID	DID	LID	BID
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bit 6 – ROID Request Overflow Error Interrupt Disable

Value	Description
0	No effect.
1	Disables request overflow error interrupt.

Bit 5 – WBEID Write Bus Error Interrupt Disable

Value	Description
0	No effect.
1	Disables bus error interrupt.

Bit 4 – RBEID Read Bus Error Interrupt Disable

Value	Description
0	No effect.
1	Disables bus error interrupt.

Bit 3 – FID End of Flush Interrupt Disable

Value	Description
0	No effect.
1	Disables end of flush interrupt.

Bit 2 – DID End of Disable Interrupt Disable

Value	Description
0	No effect.
1	Disables end of disable interrupt.

Bit 1 – LID End of Linked List Interrupt Disable

Value	Description
0	No effect.
1	Disables end of linked list interrupt.

Downloaded from Arrow.com.

Bit 0 – BID End of Block Interrupt Disable

Value	Description
0	No effect.
1	Disables end of block interrupt.

Downloaded from Arrow.com.

36.9.20 XDMAC Channel x Interrupt Mask Register [x = 0..15]

Name: XDMAC_CIM
Offset: 0x58 + n*0x40 [n=0..15]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 6 – ROIM Request Overflow Error Interrupt Mask

Value	Description
0	Request overflow interrupt is masked.
1	Request overflow interrupt is activated.

Bit 5 – WBEIM Write Bus Error Interrupt Mask

Value	Description
0	Bus error interrupt is masked.
1	Bus error interrupt is activated.

Bit 4 – RBEIM Read Bus Error Interrupt Mask

Value	Description
0	Bus error interrupt is masked.
1	Bus error interrupt is activated.

Bit 3 – FIM End of Flush Interrupt Mask

Value	Description
0	End of flush interrupt is masked.
1	End of flush interrupt is activated.

Bit 2 – DIM End of Disable Interrupt Mask

Value	Description
0	End of disable interrupt is masked.
1	End of disable interrupt is activated.

Bit 1 – LIM End of Linked List Interrupt Mask

Value	Description
0	End of linked list interrupt is masked.
1	End of linked list interrupt is activated.

Downloaded from Arrow.com.

Bit 0 – BIM End of Block Interrupt Mask

Value	Description
0	Block interrupt is masked.
1	Block interrupt is activated.

Downloaded from Arrow.com.

36.9.21 XDMAC Channel x Interrupt Status Register [x = 0..15]

Name: XDMAC_CIS
Offset: 0x5C + n*0x40 [n=0..15]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 6 – ROIS Request Overflow Error Interrupt Status

Value	Description
0	Overflow condition has not occurred.
1	Overflow condition has occurred at least once. (This information is only relevant for peripheral synchronized transfers.)

Bit 5 – WBEIS Write Bus Error Interrupt Status

Value	Description
0	Write bus error condition has not occurred.
1	At least one bus error has been detected in a write access since the last read of the Status register.

Bit 4 – RBEIS Read Bus Error Interrupt Status

Value	Description
0	Read bus error condition has not occurred.
1	At least one bus error has been detected in a read access since the last read of the Status register.

Bit 3 – FIS End of Flush Interrupt Status

Value	Description
0	End of flush condition has not occurred.
1	End of flush condition has occurred since the last read of the Status register.

Bit 2 – DIS End of Disable Interrupt Status

Value	Description
0	End of disable condition has not occurred.
1	End of disable condition has occurred since the last read of the Status register.

Bit 1 – LIS End of Linked List Interrupt Status

Value	Description
0	End of linked list condition has not occurred.

Downloaded from Arrow.com.

Value	Description
1	End of linked list condition has occurred since the last read of the Status register.

Bit 0 – BIS End of Block Interrupt Status

Value	Description
0	End of block interrupt has not occurred.
1	End of block interrupt has occurred since the last read of the Status register.

Downloaded from Arrow.com.

36.9.22 XDMAC Channel x Source Address Register [x = 0..15]

Name: XDMAC_CSA
Offset: 0x60 + n*0x40 [n=0..15]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	SA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SA[31:0] Channel x Source Address
Program this register with the source address of the DMA transfer.

Downloaded from Arrow.com.

36.9.23 XDMAC Channel x Destination Address Register [x = 0..15]

Name: XDMAC_CDA
Offset: 0x64 + n*0x40 [n=0..15]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	DA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DA[31:0] Channel x Destination Address
Program this register with the destination address of the DMA transfer.

Downloaded from Arrow.com.

36.9.24 XDMAC Channel x Next Descriptor Address Register [x = 0..15]

Name: XDMAC_CNDA
Offset: 0x68 + n*0x40 [n=0..15]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	NDA[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NDA[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NDA[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NDA[5:0]							NDAIF
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bits 31:2 – NDA[29:0] Channel x Next Descriptor Address
The 30-bit width of the NDA field represents the next descriptor address range 31:2. The descriptor is word-aligned and the two least significant register bits 1:0 are ignored.

Bit 0 – NDAIF Channel x Next Descriptor Interface

Value	Description
0	The channel descriptor is retrieved through system interface 0.
1	The channel descriptor is retrieved through system interface 1.

Downloaded from Arrow.com.

36.9.25 XDMAC Channel x Next Descriptor Control Register [x = 0..15]

Name: XDMAC_CNDC
Offset: 0x6C + n*0x40 [n=0..15]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 6:5 – QOS[1:0] Channel Quality Of Service level
This field indicates the current quality of service level for the channel. Refer to the section “MATRIX”.

Bits 4:3 – NDVIEW[1:0] Channel x Next Descriptor View

Value	Name	Description
0	NDV0	Next Descriptor View 0
1	NDV1	Next Descriptor View 1
2	NDV2	Next Descriptor View 2
3	NDV3	Next Descriptor View 3

Bit 2 – NDDUP Channel x Next Descriptor Destination Update
0 (): .
1 ():

Value	Name	Description
0	DST_PARAMS_UNCHANGED	Destination parameters remain unchanged.
1	DST_PARAMS_UPDATED	Destination parameters are updated when the descriptor is retrieved.

Bit 1 – NDSUP Channel x Next Descriptor Source Update

Value	Name	Description
0	SRC_PARAMS_UNCHANGED	Source parameters remain unchanged.
1	SRC_PARAMS_UPDATED	Source parameters are updated when the descriptor is retrieved.

Bit 0 – NDE Channel x Next Descriptor Enable

Value	Name	Description
0	DSCR_FETCH_DIS	Descriptor fetch is disabled.
1	DSCR_FETCH_EN	Descriptor fetch is enabled.

Downloaded from Arrow.com.

36.9.26 XDMAC Channel x Microblock Control Register [x = 0..15]

Name: XDMAC_CUBC
Offset: 0x70 + n*0x40 [n=0..15]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	UBLEN[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UBLEN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UBLEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – UBLEN[23:0] Channel x Microblock Length
This field indicates the number of data in the microblock. The microblock contains UBLEN data.

Downloaded from Arrow.com.

36.9.27 XDMAC Channel x Block Control Register [x = 0..15]

Name: XDMAC_CBC
Offset: 0x74 + n*0x40 [n=0..15]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					BLEN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BLEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – BLEN[11:0] Channel x Block Length
The length of the block is (BLEN+1) microblocks.

Downloaded from Arrow.com.

36.9.28 XDMAC Channel x Configuration Register [x = 0..15]

Name: XDMAC_CC
Offset: 0x78 + n*0x40 [n=0..15]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
		PERID[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
Access	R	R	R		R/W	R/W	R/W	R/W
Reset	0	0	0		0	0	0	0

Bit	15	14	13	12	11	10	9	8
		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE
Access	R/W	R/W		R/W		R/W	R/W	R/W
Reset	0	0		0		0	0	0

Bits 30:24 – PERID[6:0] Channel x Peripheral Hardware Request Line Identifier
This field contains the peripheral hardware request line identifier. PERID refers to identifiers defined in "DMA Controller Peripheral Connections".
Note: When a memory-to-memory transfer is performed, configure PERID to an unused peripheral ID (refer to table "Peripheral Identifiers").

Bit 23 – WRIP Write in Progress

Value	Name	Description
0	DONE	No active write transaction on the bus.
1	IN_PROGRESS	A write transaction is in progress.

Bit 22 – RDIP Read in Progress

Value	Name	Description
0	DONE	No active read transaction on the bus.
1	IN_PROGRESS	A read transaction is in progress.

Bit 21 – INITD Channel Initialization Done
When set to 0, XDMAC_CUBC.UBLEN and XDMAC_CNDA.NDA field values are unreliable each time a descriptor is being updated. See 36.8 XDMAC Software Requirements.

Value	Name	Description
0	IN_PROGRESS	Channel initialization is in progress.
1	TERMINATED	Channel initialization is completed.

Bits 19:18 – DAM[1:0] Channel x Destination Addressing Mode

Value	Name	Description
0	FIXED_AM	The address remains unchanged.
1	INCREMENTED_AM	The addressing mode is incremented (the increment size is set to the data size).
2	UBS_AM	The microblock stride is added at the microblock boundary.

Downloaded from Arrow.com.

Value	Name	Description
3	UBS_DS_AM	The microblock stride is added at the microblock boundary; the data stride is added at the data boundary.

Bits 17:16 – SAM[1:0] Channel x Source Addressing Mode

Value	Name	Description
0	FIXED_AM	The address remains unchanged.
1	INCREMENTED_AM	The addressing mode is incremented (the increment size is set to the data size).
2	UBS_AM	The microblock stride is added at the microblock boundary.
3	UBS_DS_AM	The microblock stride is added at the microblock boundary, the data stride is added at the data boundary.

Bit 14 – DIF Channel x Destination Interface Identifier

Value	Name	Description
0	AHB_IF0	The data is written through system bus interface 0.
1	AHB_IF1	The data is written through system bus interface 1.

Bit 13 – SIF Channel x Source Interface Identifier

Value	Name	Description
0	AHB_IF0	The data is read through system bus interface 0.
1	AHB_IF1	The data is read through system bus interface 1.

Bits 12:11 – DWIDTH[1:0] Channel x Data Width

Value	Name	Description
0	BYTE	The data size is set to 8 bits
1	HALFWORD	The data size is set to 16 bits
2	WORD	The data size is set to 32 bits

Bits 10:8 – CSIZE[2:0] Channel x Chunk Size

Value	Name	Description
0	CHK_1	1 data transferred
1	CHK_2	2 data transferred
2	CHK_4	4 data transferred
3	CHK_8	8 data transferred
4	CHK_16	16 data transferred

Bit 7 – MEMSET Channel x Fill Block of Memory

Value	Name	Description
0	NORMAL_MODE	Memset is not activated.
1	HW_MODE	Sets the block of memory pointed by DA field to the specified value. This operation is performed on 8-, 16- or 32-bit basis.

Bit 6 – SWREQ Channel x Software Request Trigger

Value	Name	Description
0	HWR_CONNECTED	Hardware request line is connected to the peripheral request line.
1	SWR_CONNECTED	Software request is connected to the peripheral request line.

Bit 4 – DSYNC Channel x Synchronization

Value	Name	Description
0	PER2MEM	Peripheral-to-memory transfer
1	MEM2PER	Memory-to-peripheral transfer

Bits 2:1 – MBSIZE[1:0] Channel x Memory Burst Size

Value	Name	Description
0	SINGLE	The memory burst size is set to one.
1	FOUR	The memory burst size is set to four.

Downloaded from Arrow.com.

Value	Name	Description
2	EIGHT	The memory burst size is set to eight.
3	SIXTEEN	The memory burst size is set to sixteen.

Bit 0 – TYPE Channel x Transfer Type

Value	Name	Description
0	MEM_TRAN	Self-triggered mode (memory-to-memory transfer)
1	PER_TRAN	Synchronized mode (peripheral-to-memory or memory-to-peripheral transfer)

Downloaded from Arrow.com.

36.9.29 XDMAC Channel x Data Stride Memory Set Pattern Register [x = 0..15]

Name: XDMAC_CDS_MSP
Offset: 0x7C + n*0x40 [n=0..15]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	DDS_MSP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DDS_MSP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SDS_MSP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SDS_MSP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – DDS_MSP[15:0] Channel x Destination Data Stride or Memory Set Pattern
When XDMAC_CCx.MEMSET = 0, this field indicates the destination data stride.
Number of bytes for the data stride of channel x (two's complement). If the field is set to zero the data is contiguous (see Data Striding Diagram).
The DDS_MSP field is only relevant when XDMAC_CCx.SAM=UBS_DS_AM.
When XDMAC_CCx.MEMSET = 1, this field indicates the memory set pattern.

Bits 15:0 – SDS_MSP[15:0] Channel x Source Data Stride or Memory Set Pattern
When XDMAC_CCx.MEMSET = 0, this field indicates the source data stride.
Number of bytes for the data stride of channel x (two's complement). If the field is set to zero the data is contiguous (see Data Striding Diagram).
The SDS_MSP field is only relevant when XDMAC_CCx.SAM=UBS_DS_AM.
When XDMAC_CCx.MEMSET = 1, this field indicates the memory set pattern.

Downloaded from Arrow.com.

36.9.30 XDMAC Channel x Source Microblock Stride Register [x = 0..15]

Name: XDMAC_CSUS
Offset: 0x80 + n*0x40 [n=0..15]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SUBS[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SUBS[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SUBS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – SUBS[23:0] Channel x Source Microblock Stride
Number of bytes for the microblock stride for channel x (two's complement). If the field is set to zero the data is contiguous (see Figure 36-2).
The SUBS field is only relevant when XDMAC_CCx.SAM=UBS_AM or XDMAC_CCx.SAM=UBS_DS_AM.

Downloaded from Arrow.com.

36.9.31 XDMAC Channel x Destination Microblock Stride Register [x = 0..15]

Name: XDMAC_CDUS
Offset: 0x84 + n*0x40 [n=0..15]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DUBS[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DUBS[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DUBS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – DUBS[23:0] Channel x Destination Microblock Stride
Number of bytes for the microblock stride for channel x (two's complement). If the field is set to zero the data is contiguous (see Figure 36-2).
The DUBS field is only relevant when XDMAC_CCx.SAM=UBS_AM or XDMAC_CCx.SAM=UBS_DS_AM.

Downloaded from Arrow.com.

37. LCD Controller (LCDC)

37.1 Description

The LCD Controller (LCDC) consists of logic for transferring LCD image data from an external display buffer to an LCD module. The LCD has one display input buffer per overlay that fetches pixels through the single system bus master interface and a lookup table to allow palletized display configurations. The LCD controller is programmable on a per overlay basis, and supports different LCD resolutions, window sizes, image formats and pixel depths.

The LCD is connected to the system bus as a master for reading pixel data. It also integrates an APB interface to configure its registers.

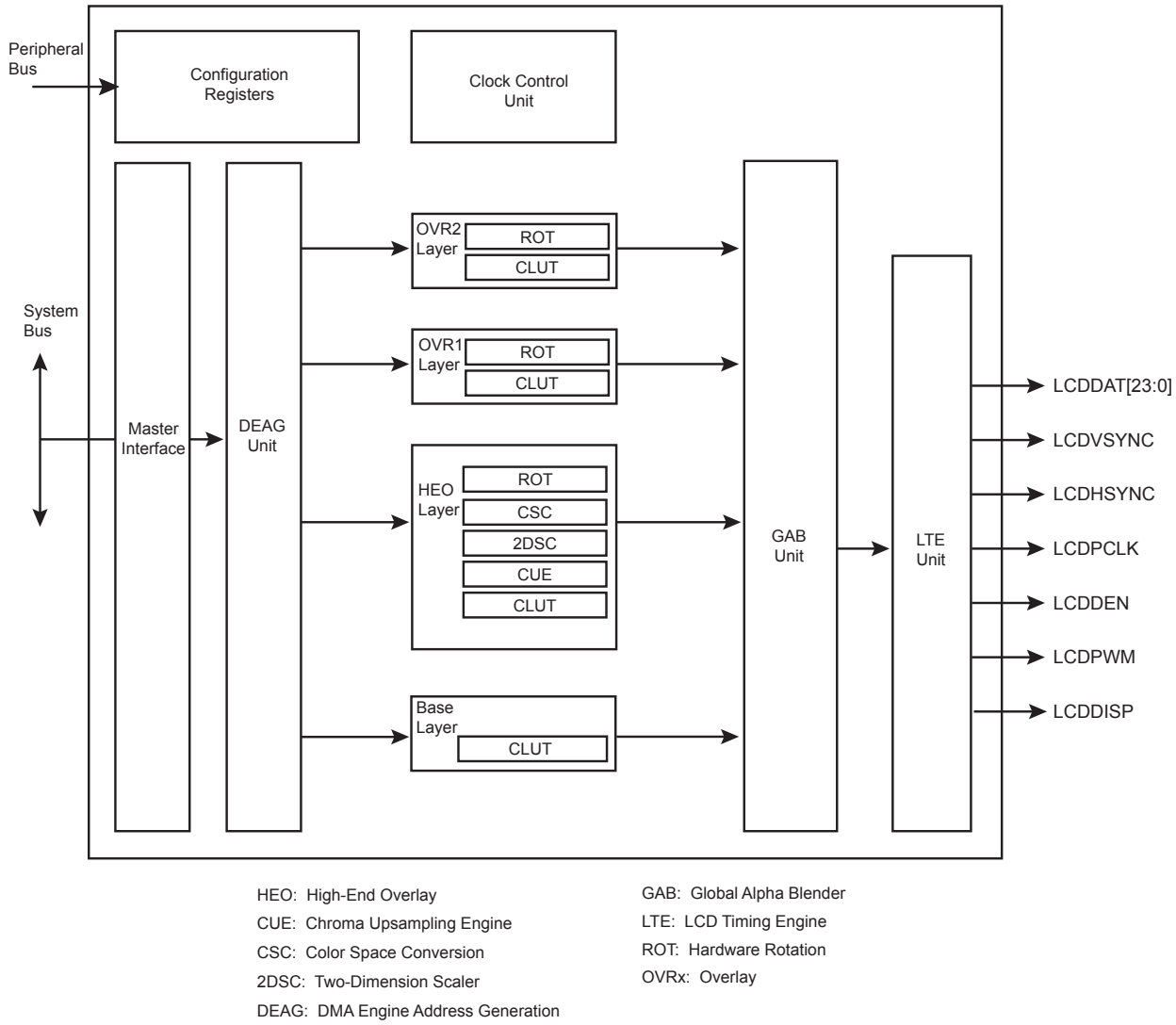
37.2 Embedded Characteristics

- Single System Bus Master Interface
- Supports Single Scan Active TFT Display
- Supports 12-, 16-, 18- and 24-bit Output Mode through the Spatial Dithering Unit
- Asynchronous Output Mode Supported
- 1, 2, 4, 8 bits per Pixel (Palletized)
- 12, 16, 18, 19, 24, 25 and 32 bits per Pixel (Non-palletized)
- Supports One Base Layer (Background)
- Supports Overlay 1 Layer
- Supports Overlay 2 Layer
- Supports High-End Overlay (HEO) Layer
- High-End Overlay supports 4:2:0 Planar Mode and Semiplanar Mode
- High-End Overlay supports 4:2:2 Planar Mode, Semiplanar Mode and Packed
- High-End Overlay includes Chroma Upsampling Unit
- Little Endian Memory Organization
- Programmable Timing Engine, with Integer Clock Divider
- Programmable Polarity for Data, Line Synchro and Frame Synchro
- Up to 1024x768 (XGA) with Overlay (Application-Dependent). Still Image up to WXGA.
- Color Lookup Table with up to 256 Entries and Predefined 8-bit Alpha
- Programmable Negative and Positive Row Striding for all Layers
- Programmable Negative and Positive Pixel Striding for Layers
- Horizontal and Vertical Rescaling Unit with Edge Interpolation and Independent Non-Integer Ratio, up to 1024x768
- Hidden Layer Removal supported
- Integrates Fully Programmable Color Space Conversion
- Blender Function Supports Arbitrary 8-bit Alpha Value and Chroma Keying
- DMA User Interface uses Linked List Structure and Add-to-queue Structure

Downloaded from Arrow.com.

37.3 Block Diagram

Figure 37-1. LCDC Block Diagram



HEO: High-End Overlay
CUE: Chroma Upsampling Engine
CSC: Color Space Conversion
2DSC: Two-Dimension Scaler
DEAG: DMA Engine Address Generation

GAB: Global Alpha Blender
LTE: LCD Timing Engine
ROT: Hardware Rotation
OVRx: Overlay

37.4 I/O Lines Description

Table 37-1. I/O Lines Description

Name	Description	Type
LCDPWM	Contrast control signal, using Pulse Width Modulation	Output
LCDHSYNC	Horizontal Synchronization Pulse	Output
LCDVSYNC	Vertical Synchronization Pulse	Output
LCDDAT[23:0]	LCD 24-bit data bus	Output
LCDDEN	Data Enable	Output
LCDDISP	Display Enable signal	Output

Downloaded from Arrow.com.

.....continued		
Name	**Description**	**Type**
LCDPCLK	Pixel Clock	Output

37.5 Product Dependencies

37.5.1 I/O Lines

The pins used for interfacing the LCD Controller may be multiplexed with PIO lines. The programmer must first program the PIO Controller to assign the pins to their peripheral function. If I/O lines of the LCD Controller are not used by the application, they can be used for other purposes by the PIO Controller.

37.5.2 Power Management

The LCD Controller is not continuously clocked. Before using it, the user must first enable the LCDC peripheral and generic (GCLK) clocks in the Power Management Controller.

37.5.3 Interrupt Sources

The LCD Controller interrupt line is connected to one of the internal sources of the interrupt controller. Using the LCD Controller interrupt requires prior programming of the interrupt controller.

37.6 Functional Description

The LCD module integrates the following digital blocks:

- DMA Engine Address Generation (DEAG)—this block performs data prefetch and requests access to the system bus interface.
- Input Overlay FIFO—stores the stream of pixels
- Color Lookup Table (CLUT)—these 256 RAM-based lookup table entries are selected when the color depth is set to 1, 2, 4 or 8 bpp.
- Chroma Upsampling Engine (CUE)—this block is selected when the input image sampling format is YUV (Y'CbCr) 4:2:0 and converts it to higher quality 4:4:4 image.
- Color Space Conversion (CSC)—changes the color space from YUV to RGB
- Two Dimension Scaler (2DSC)—resizes the image
- Global Alpha Blender (GAB)—performs programmable 256-level alpha blending
- Output FIFO—stores the blended pixel prior to display
- LCD Timing Engine—provides a fully programmable HSYNC-VSYNC interface

The DMA controller reads the image through the system bus master interface. The LCD controller engine formats the display data, then the GAB performs alpha blending if required, and writes the final pixel into the output FIFO. The programmable timing engine drives a valid pixel onto the LCDDAT[23:0] display bus.

37.6.1 Timing Engine Configuration

37.6.1.1 Pixel Clock Period Configuration

The pixel clock (LCDPCLK) generated by the timing engine is the source clock divided by the field CLKDIV in the LCDC_LCDCFG0 register. The source clock is the GCLK clock.

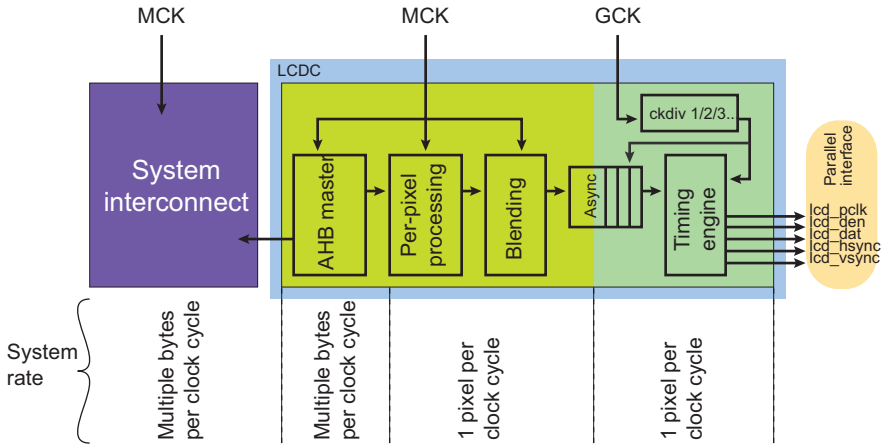
Pixel clock period formula:

$$\text{LCD_PCLK} = \frac{\text{source clock}}{\text{CLKDIV}+2}$$

The pixel clock polarity is also programmable.

Downloaded from Arrow.com.

Figure 37-2. LCD Controller Pixel Processing and Timing Engine Clock Scheme



37.6.1.2 Horizontal and Vertical Synchronization Configuration

The following fields are used to configure the timing engine:

- LCDC_LCDCFG1.HSPW
- LCDC_LCDCFG1.VSPW
- LCDC_LCDCFG2.VFPW
- LCDC_LCDCFG2.VBPW
- LCDC_LCDCFG3.HFPW
- LCDC_LCDCFG3.HBPW
- LCDC_LCDCFG4.PPL
- LCDC_LCDCFG4.RPF

The polarity of output signals is also programmable.

37.6.1.3 Timing Engine Powerup Software Operation

The following sequence is used to enable the display:

1. Configure LCD timing parameters, signal polarity and clock period.
2. Enable the pixel clock by writing a one to bit LCDC_LCDEN.CLKEN.
3. Poll bit LCDC_LCDSR.CLKSTS to check that the clock is running.
4. Enable Horizontal and Vertical Synchronization by writing a one to bit LCDC_LCDEN.SYNCEN.
5. Poll bit LCDC_LCDSR.LCDSTS to check that the synchronization is up.
6. Enable the display power signal by writing a one to bit LCDC_LCDEN.DISPEN.
7. Poll bit LCDC_LCDSR.DISPSTS to check that the power signal is activated.

The field LCDC_LCDCFG5.GUARDTIME is used to configure the number of frames before the assertion of the DISP signal.

37.6.1.4 Timing Engine Powerdown Software Operation

The following sequence is used to disable the display:

1. Disable the DISP signal by writing bit LCDC_LCDDIS.DISPDIS.
2. Poll bit LCDC_LCDSR.DISPSTS to verify that the DISP is no longer activated.
3. Disable the HSYNC and VSYNC signals by writing a one to bit LCDC_LCDDIS.SYNCDIS.
4. Poll bit LCDC_LCDSR.LCDSTS to check that the synchronization is off.
5. Disable the pixel clock by writing a one to bit LCDC_LCDDIS.CLKDIS.

37.6.2 DMA Software Operations

37.6.2.1 DMA Channel Descriptor (DSCR) Alignment and Structure

The DMA Channel Descriptor (DSCR) must be aligned on a 32-bit boundary.

Downloaded from Arrow.com.

The DMA Channel Descriptor structure contains three fields:

- DSCR.CHXADDR: Frame Buffer base address register
- DSCR.CHXCTRL: Transfer Control register
- DSCR.CHXNEXT: Next Descriptor Address register

Table 37-2. DMA Channel Descriptor Structure

System Memory	Structure Field for Channel CHX
DSCR + 0x0	ADDR
DSCR + 0x4	CTRL
DSCR + 0x8	NEXT

37.6.2.2 Enabling a DMA Channel

Follow the steps below to enable a DMA channel:

1. Check the status of the channel by reading the CHXCHSR register.
2. Write the channel descriptor (DSCR) structure in the system memory by writing DSCR.CHXADDR Frame base address, DSCR.CHXCTRL channel control and DSCR.CHXNEXT next descriptor location.
3. If more than one descriptor is expected, the field DFETCH of DSCR.CHXCTRL is set to ‘1’ to enable the descriptor fetch operation.
4. Write the DSCR.CHXNEXT register with the address location of the descriptor structure and set DFETCH field of the DSCR.CHXCTRL register to ‘1’.
5. Enable the relevant channel by writing one to the CHEN field of the CHXCHER register.
6. An interrupt may be raised if unmasked when the descriptor has been loaded.

37.6.2.3 Disabling a DMA Channel

Follow the steps below to disable a DMA channel:

1. Clearing the DFETCH bit in the DSCR.CHXCTRL field of the DSCR structure disables the channel at the end of the frame.
2. Setting the DSCR.CHXNEXT field of the DSCR structure disables the channel at the end of the frame.
3. Writing one to the CHDIS field of the CHXCHDR register disables the channel at the end of the frame.
4. Writing one to the CHRST field of the CHXCHDR register disables the channel immediately. This may occur in the middle of the image.
5. Polling CHSR field in the CHXCHSR register until the channel is successfully disabled.

37.6.2.4 DMA Dynamic Linking of a New Transfer Descriptor

1. Write the new descriptor structure in the system memory.
2. Write the address of the new structure in the CHXHEAD register.
3. Add the new structure to the queue of descriptors by writing one to the A2QEN field of the CHXCHER register.
4. The new descriptor is added to the queue on the next frame.
5. An interrupt is raised if unmasked, when the head descriptor structure has been loaded by the DMA channel.

37.6.2.5 DMA Interrupt Generation

The DMA Controller operation sets the following interrupt flags in the Interrupt Status register CHXISR:

- DMA field indicates that the DMA transfer is completed.
- DSCR field indicates that the descriptor structure is loaded in the DMA controller.
- ADD field indicates that a descriptor has been added to the descriptor queue.
- DONE field indicates that the channel transfer has terminated and the channel is automatically disabled.

37.6.2.6 DMA Address Alignment Requirements

When programming the DSCR.CHXADDR field of the DSCR structure, the following requirement must be met.

Downloaded from Arrow.com.

Table 37-3. DMA Address Alignment when CLUT Mode is Selected

CLUT Mode	DMA Address Alignment
1 bpp	8 bits
2 bpp	8 bits
4 bpp	8 bits
8 bpp	8 bits

Table 37-4. DMA Address Alignment when RGB Mode is Selected

RGB Mode	DMA Address Alignment
12 bpp RGB 444	16 bits
16 bpp ARGB 4444	16 bits
16 bpp RGBA 4444	16 bits
16 bpp RGB 565	16 bits
16 bpp TRGB 1555	16 bits
18 bpp RGB 666	32 bits
18 bpp RGB 666 PACKED	8 bits
19 bpp TRGB 1666	32 bits
19 bpp TRGB 1666	8 bits
24 bpp RGB 888	32 bits
24 bpp RGB 888 PACKED	8 bits
25 bpp TRGB 1888	32 bits
32 bpp ARGB 8888	32 bits
32 bpp RGBA 8888	32 bits

Table 37-5. DMA Address Alignment when YUV Mode is Selected

YUV Mode	DMA Address Alignment
32 bpp AYCrCb	32 bits
16 bpp YCrCb 4:2:2	32 bits
16 bpp semiplanar YCrCb 4:2:2	Y 8 bits
	CrCb 16 bits
16 bpp planar YCrCb 4:2:2	Y 8 bits
	Cr 8 bits
	Cb 8 bits
12 bpp YCrCb 4:2:0	Y 8 bits
	CrCb 16 bits
12 bpp YCrCb 4:2:0	Y 8 bits
	Cr 8 bits
	Cb 8 bits

Downloaded from Arrow.com.

37.6.3 Overlay Software Configuration

37.6.3.1 System Bus Access Attributes

These attributes are defined to improve bandwidth of the overlay.

- LOCKDIS bit—When set to ‘1’, the system bus lock signal is not asserted when the PSTRIDE value is different from zero (rotation in progress).
- ROTDIS bit—When set to ‘1’, the Pixel Striding optimization is disabled.
- DLBO bit—When set to ‘1’, only defined burst lengths are performed when the DMA channel retrieves the data from the memory.
- BLEN field—Defines the maximum burst length of the DMA channel.
- SIF bit—Defines the targeted DMA interface.

37.6.3.2 Color Attributes

- CLUTMODE field—Selects the Color Lookup Table mode.
- RGBMODE field—Selects the RGB mode.
- YUVMODE field—Selects the Luminance Chrominance mode.

37.6.3.3 Window Position, Size, Scaling and Striding Attributes

- XPOS and YPOS fields—Defines the position of the overlay window.
- XSIZE and YSIZE fields—Defines the size of the displayed window.
- XMEMSIZE and YMEMSIZE fields—Defines the size of the image frame buffer.
- XSTRIDE and PSTRIDE fields—Defines the line and pixel striding.
- XFACTOR and YFACTOR fields—Defines the scaling ratio.

The position and size attributes are to be programmed to keep the window within the display area.

When the Color Lookup Table mode is enabled, the restrictions detailed in the following table apply on the horizontal and vertical window sizes.

Table 37-6. Color Lookup Table Mode and Window Size

CLUT Mode	X-Y Size Requirement
1 bpp	Multiple of 8 pixels
2 bpp	Multiple of 4 pixels
4 bpp	Multiple of 2 pixels
8 bpp	Free size

Pixel striding is disabled when CLUT mode is enabled.

When YUV mode is enabled, the restrictions detailed in the following table apply on the window size.

Table 37-7. YUV Mode and Window Size

YUV Mode	X-Y Requirement, Scaling Turned Off	X-Y Requirement, Scaling Turned On
AYUV	Free size	X-Y size is greater than 5
YUV 4:2:2 packed	XSIZE is greater than 2 pixels	X-Y size is greater than 5
YUV 4:2:2 semiplanar	XSIZE is greater than 2 pixels	X-Y size is greater than 5
YUV 4:2:2 planar	XSIZE is greater than 2 pixels	X-Y size is greater than 5
YUV 4:2:0 semiplanar	XSIZE is greater that 2 pixels	X-Y size is greater than 5
YUV 4:2:0 planar	XSIZE is greater than 2 pixels	X-Y size is greater than 5

In RGB mode, there is no restriction on the line length.

Downloaded from Arrow.com.

37.6.3.4 Overlay Blender Attributes

When two or more video layers are used, alpha blending is performed to define the final image displayed. Each window has its own blending attributes.

- CRKEY bit—Enables the chroma keying and match logic.
- INV bit—Performs bit inversion at pixel level.
- ITER2BL bit—When written to '1', the iterated data path is selected.
- ITER bit—When written to '1', the iterated value is used in the iterated data path, otherwise the iterated value is set to 0.
- REVALPHA bit—Uses the reverse alpha value.
- GAEN bit—Enables the global alpha value in the data path.
- LAEN bit—Enables the local alpha value from the pixel.
- OVR bit—When written to '1', the overlay is selected as an input of the blender.
- DMA bit—The DMA data path is activated.
- REP bit—Enables the bit replication to fill the 24-bit internal data path.
- DSTKEY bit—When written to '1', Destination keying is enabled.
- GA field—Defines the global alpha value.

37.6.3.5 Overlay Attributes Software Operation

1. When required, write the overlay attributes configuration registers.
2. Set UPDATEEN field of the CHXCHER register.
3. Poll UPDATESR field in the CHXCHSR, the update applies when that field is reset.

37.6.4 RGB Frame Buffer Memory Bitmap

37.6.4.1 1 bpp Through Color Lookup Table

Table 37-8. 1 bpp Memory Mapping, Little Endian Organization

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 1 bpp	p31	p30	p29	p28	p27	p26	p25	p24	p23	p22	p21	p20	p19	p18	p17	p16	p15	p14	p13	p12	p11	p10	p9	p8	p7	p6	p5	p4	p3	p2	p1	p0

37.6.4.2 2 bpp Through Color Lookup Table

Table 37-9. 2 bpp Memory Mapping, Little Endian Organization

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 2 bpp	p15		p14		p13		p12		p11		p10		p9		p8		p7		p6		p5		p4		p3		p2		p1		p0	

37.6.4.3 4 bpp Through Color Lookup Table

Table 37-10. 4 bpp Memory Mapping, Little Endian Organization

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 4 bpp	p7				p6				p5				p4				p3				p2				p1				p0			

37.6.4.4 8 bpp Through Color Lookup Table

Table 37-11. 8 bpp Memory Mapping, Little Endian Organization

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 8 bpp	p3								p2								p1								p0							

Downloaded from Arrow.com.

37.6.4.5 12 bpp Memory Mapping, RGB 4:4:4

Table 37-12. 12 bpp Memory Mapping, Little Endian Organization

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 12 bpp	-				R1[3:0]				G1[3:0]				B1[3:0]				-				R0[3:0]				G0[3:0]				B0[3:0]			

37.6.4.6 16 bpp Memory Mapping with Alpha Channel, ARGB 4:4:4:4

Table 37-13. 16 bpp Memory Mapping, Little Endian Organization

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 16 bpp	A1[3:0]				R1[3:0]				G1[3:0]				B1[3:0]				A0[3:0]				R0[3:0]				G0[3:0]				B0[3:0]			

37.6.4.7 16 bpp Memory Mapping with Alpha Channel, RGBA 4:4:4:4

Table 37-14. 16 bpp Memory Mapping, Little Endian Organization

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 16 bpp	R1[3:0]				G13:0]				B1[3:0]				A1[3:0]				R0[3:0]				G0[3:0]				B0[3:0]				A0[3:0]			

37.6.4.8 16 bpp Memory Mapping with Alpha Channel, RGB 5:6:5

Table 37-15. 16 bpp Memory Mapping, Little Endian Organization

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 16bpp	R1[4:0]					G1[5:0]						B1[4:0]					R0[4:0]					G0[5:0]						B0[4:0]			

37.6.4.9 16 bpp Memory Mapping with Transparency Bit, ARGB 1:5:5:5

Table 37-16. 16 bpp Memory Mapping, Little Endian Organization

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 4 bpp	A1	R1[4:0]				G1[4:0]						B1[4:0]					A0	R0[4:0]					G0[4:0]						B0[4:0]			

37.6.4.10 18 bpp Unpacked Memory Mapping with Transparency Bit, RGB 6:6:6

Table 37-17. 18 bpp Unpacked Memory Mapping, Little Endian Organization

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 18 bpp	–								–								R0[5:0]								G0[5:0]							

37.6.4.11 18 bpp Packed Memory Mapping with Transparency Bit, RGB 6:6:6

Table 37-18. 18 bpp Packed Memory Mapping, Little Endian Organization at Address 0x0, 0x1, 0x2, 0x3

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 18 bpp	G1[1:0]		B1[5:0]						–								R0[5:0]								G0[5:0]							

Downloaded from Arrow.com.

Table 37-19. 18 bpp Packed Memory Mapping, Little Endian Organization at Address 0x4, 0x5, 0x6, 0x7

Mem address	0x7								0x6								0x5								0x4							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 18 bpp	R2[3:0]				G2[5:0]				B2[5:0]								–				R1[5:2]				G1[5:2]							

Table 37-20. 18 bpp Packed Memory Mapping, Little Endian Organization at Address 0x8, 0x9, 0xA, 0xB

Mem address	0xB								0xA								0x9								0x8							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 18 bpp	G4[1:0]		B4[5:0]						–				R3[5:0]				G3[5:0]								B3[3:0]						R2[5:4]	

37.6.4.12 19 bpp Unpacked Memory Mapping with Transparency Bit, RGB 1:6:6:6

Table 37-21. 19 bpp Unpacked Memory Mapping, Little Endian Organization

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 19 bpp	–								–					A0	R0[5:0]						G0[5:0]				B0[5:0]							

37.6.4.13 19 bpp Packed Memory Mapping with Transparency Bit, ARGB 1:6:6:6

Table 37-22. 19 bpp Packed Memory Mapping, Little Endian Organization at Address 0x0, 0x1, 0x2, 0x3

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 19 bpp	G1[1:0]		B1[5:0]						–					A0	R0[5:0]						G0[5:0]				B0[5:0]							

Table 37-23. 19 bpp Packed Memory Mapping, Little Endian Organization at Address 0x4, 0x5, 0x6, 0x7

Mem address	0x7								0x6								0x5								0x4							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 19 bpp	R2[3:0]				G2[5:0]				B2[5:0]								–					A1	R1[5:2]				G1[5:2]					

Table 37-24. 18 bpp Packed Memory Mapping, Little Endian Organization at Address 0x8, 0x9, 0xA, 0xB

Mem address	0xB								0xA								0x9								0x8							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 19 bpp	G4[1:0]		B4[5:0]						–			A3	R3[5:0]				G3[5:0]								B3[3:0]						R2[5:4]	

37.6.4.14 24 bpp Unpacked Memory Mapping, RGB 8:8:8

Table 37-25. 24 bpp Memory Mapping, Little Endian Organization

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 24 bpp	–								R0[7:0]								G0[7:0]								B0[7:0]							

Downloaded from Arrow.com.

37.6.4.15 24 bpp Packed Memory Mapping, RGB 8:8:8

Table 37-26. 24 bpp Packed Memory Mapping, Little Endian Organization at Address 0x0, 0x1, 0x2, 0x3

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 24 bpp	B1[7:0]								R0[7:0]								G0[7:0]								B0[7:0]							

Table 37-27. 24 bpp Packed Memory Mapping, Little Endian Organization at Address 0x4, 0x5, 0x6, 0x7

Mem addr	0x7								0x6								0x5								0x4							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 24 bpp	G2[7:0]								B2[7:0]								R1[7:0]								G1[7:0]							

37.6.4.16 25 bpp Memory Mapping, ARGB 1:8:8:8

Table 37-28. 25 bpp Memory Mapping, Little Endian Organization

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 25 bpp	–							A0	R0[7:0]								G0[7:0]								B0[7:0]							

37.6.4.17 32 bpp Memory Mapping, ARGB 8:8:8:8

Table 37-29. 32 bpp Memory Mapping, Little Endian Organization

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 32 bpp	A0[7:0]								R0[7:0]								G0[7:0]								B0[7:0]							

37.6.4.18 32 bpp Memory Mapping, RGBA 8:8:8:8

Table 37-30. 32 bpp Memory Mapping, Little Endian Organization

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 32 bpp	R0[7:0]								G0[7:0]								B0[7:0]								A0[7:0]							

37.6.5 YUV Frame Buffer Memory Mapping

37.6.5.1 AYCbCr 4:4:4 Interleaved Frame Buffer Memory Mapping

Table 37-31. 32 bpp Memory Mapping, Little Endian Organization

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 16 bpp	A0[7:0]								Y0[7:0]								Cb0[7:0]								Cr0[7:0]							

37.6.5.2 4:2:2 Interleaved Mode Frame Buffer Memory Mapping

Table 37-32. 16 bpp 4:2:2 Interleaved Mode 0

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 16 bpp	Cr0[7:0]								Y1[7:0]								Cb0[7:0]								Y0[7:0]							

Downloaded from Arrow.com.

Table 37-33. 16 bpp 4:2:2 Interleaved Mode 1

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 16 bpp	Y1[7:0]								Cr0[7:0]								Y0[7:0]								Cb0[7:0]							

Table 37-34. 16 bpp 4:2:2 Interleaved Mode 2

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 16 bpp	Cb0[7:0]								Y1[7:0]								Cr0[7:0]								Y0[7:0]							

Table 37-35. 16 bpp 4:2:2 Interleaved Mode 3

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 16 bpp	Y1[7:0]								Cb0[7:0]								Y0[7:0]								Cr0[7:0]							

37.6.5.3 4:2:2 Semiplanar Mode Frame Buffer Memory Mapping

Table 37-36. 4:2:2 Semiplanar Luminance Memory Mapping, Little Endian Organization for Byte 0x0, 0x1, 0x2, 0x3

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 16 bpp	Y3[7:0]								Y2[7:0]								Y1[7:0]								Y0[7:0]							

Table 37-37. 4:2:2 Semiplanar Chrominance Memory Mapping, Little Endian Organization for Byte 0x0, 0x1, 0x2, 0x3

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 16 bpp	Cb2[7:0]								Cr2[7:0]								Cb0[7:0]								Cr0[7:0]							

37.6.5.4 4:2:2 Planar Mode Frame Buffer Memory Mapping

Table 37-38. 4:2:2 Planar Mode Luminance Memory Mapping, Little Endian Organization for Byte 0x0, 0x1, 0x2, 0x3

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 16 bpp	Y3[7:0]								Y2[7:0]								Y1[7:0]								Y0[7:0]							

Table 37-39. 4:2:2 Planar Mode Chrominance Memory Mapping, Little Endian Organization for Byte 0x0, 0x1, 0x2, 0x3

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 16 bpp	C3[7:0] C3=Cr/Cb								C2[7:0] C2=Cr/Cb								C1[7:0] C1=Cr/Cb								C0[7:0] C0=Cr/Cb							

37.6.5.5 4:2:0 Planar Mode Frame Buffer Memory Mapping

In Planar mode, the three video components Y, Cr and Cb are split into three memory areas and stored in a raster-scan order. These three memory planes are contiguous and always aligned on a 32-bit boundary.

Downloaded from Arrow.com.

Table 37-40. 4:2:0 Planar Mode Luminance Memory Mapping, Little Endian Organization for Byte 0x0, 0x1, 0x2, 0x3

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 12 bpp	Y3[7:0]								Y2[7:0]								Y1[7:0]								Y0[7:0]							

Table 37-41. 4:2:0 Planar Mode Luminance Memory Mapping, Little Endian Organization for Byte 0x4, 0x5, 0x6, 0x7

Mem address	0x7								0x6								0x5								0x4							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 12 bpp	Y7[7:0]								Y6[7:0]								Y5[7:0]								Y4[7:0]							

Table 37-42. 4:2:0 Planar Mode Chrominance Memory Mapping, Little Endian Organization for Byte 0x0, 0x1, 0x2, 0x3

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 12 bpp	C3[7:0] C3=Cr/Cb								C2[7:0] C2=Cr/Cb								C1[7:0] C1=Cr/Cb								C0[7:0] C0=Cr/Cb							

Table 37-43. 4:2:0 Planar Mode Chrominance Memory Mapping, Little Endian Organization for Byte 0x4, 0x5, 0x6, 0x7

Mem address	0x7								0x6								0x5								0x4							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 12 bpp	C7[7:0] C7=Cr/Cb								C6:[7:0] C6=Cr/Cb								C5[7:0] C5=Cr/Cb								C4[7:0] C4=Cr/Cb							

37.6.5.6 4:2:0 Semiplanar Frame Buffer Memory Mapping

Table 37-44. 4:2:0 Semiplanar Mode Luminance Memory Mapping, Little Endian Organization

Mem address	0x7								0x6								0x5								0x4							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 12 bpp	Y3[7:0]								Y2[7:0]								Y1[7:0]								Y0[7:0]							

Table 37-45. 4:2:0 Semiplanar Mode Chrominance Memory Mapping, Little Endian Organization

Mem address	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 12 bpp	Cb1[7:0]								Cr1[7:0]								Cb0[7:0]								Cr0[7:0]							

37.6.6 Chrominance Upsampling Unit

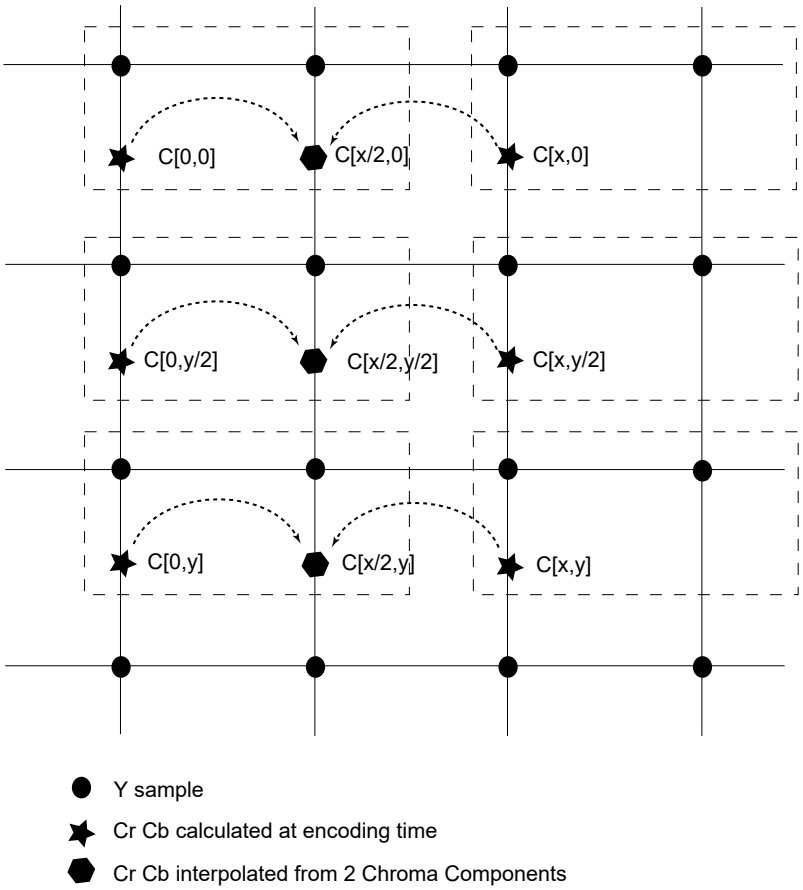
Both the 4:2:2 and the 4:2:0 input formats are supported by the LCD module. In 4:2:2, the two chrominance components are sampled at half the luminance sample rate. The horizontal chrominance resolution is halved. When this input format is selected, the chrominance upsampling unit uses two chrominances to interpolate the missing component.

In 4:2:0, Cr and Cb components are subsampled at a factor of two vertically and horizontally. When this input mode is selected, the chrominance upsampling unit uses two and four chroma components to generate the missing horizontal and vertical components.

Downloaded from Arrow.com.

Figure 37-3. 4:2:2 Upsampling Algorithm

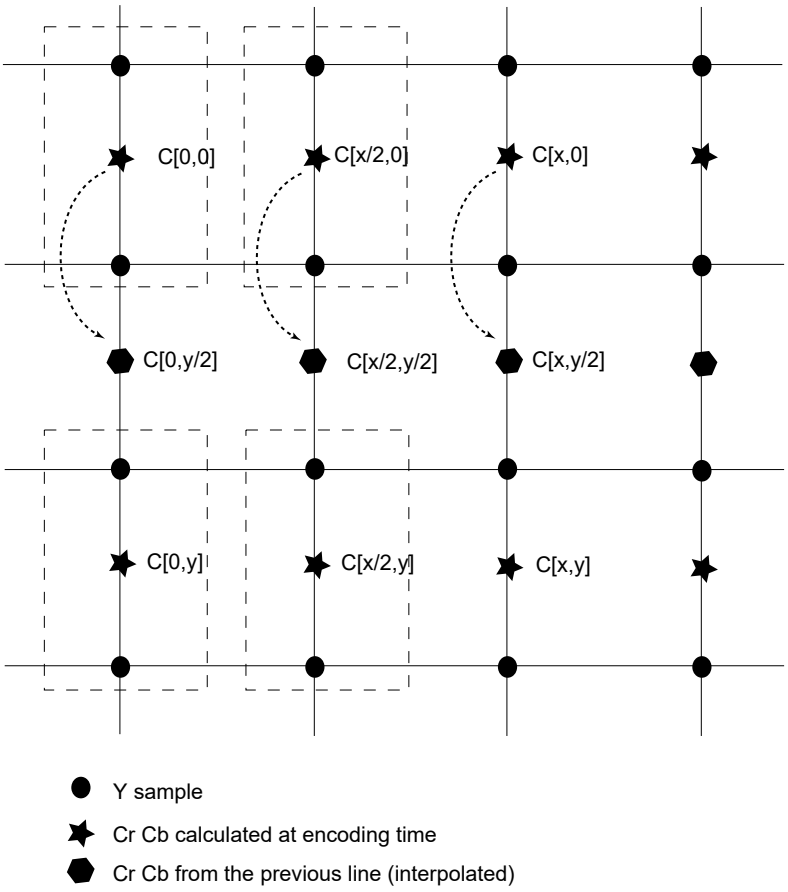
Vertical and Horizontal upsampling 4:2:2 to 4:4:4 conversion 0 or 180 degrees



Downloaded from Arrow.com.

Figure 37-4. 4:2:2 Packed Upsampling Algorithm

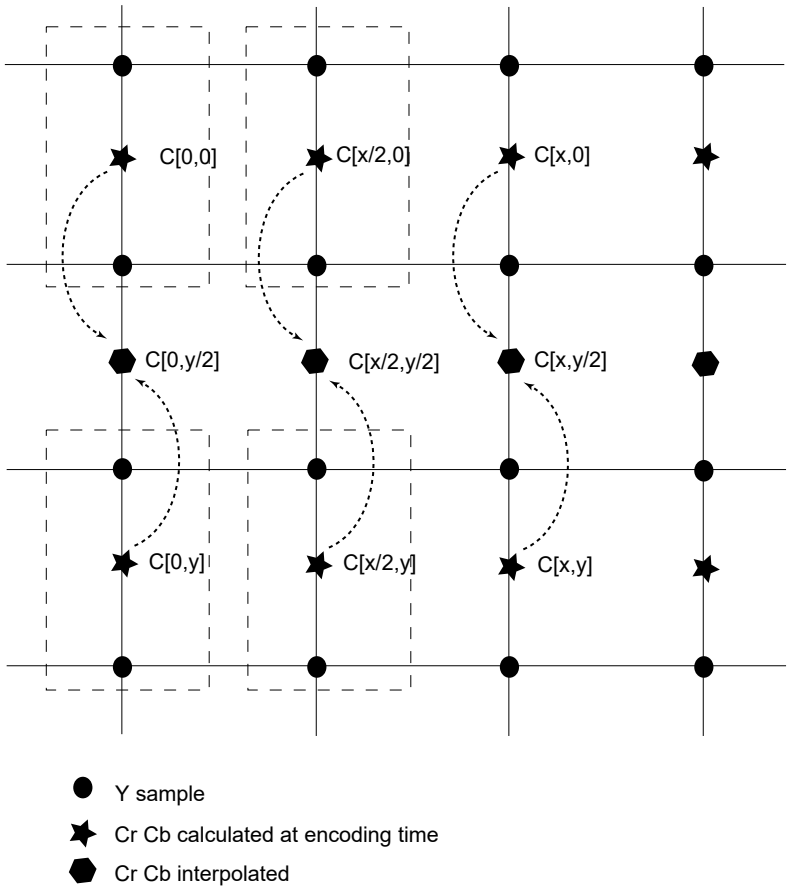
Vertical and Horizontal upsampling 4:2:2 to 4:4:4 conversion 90 or 270 degrees



Downloaded from Arrow.com.

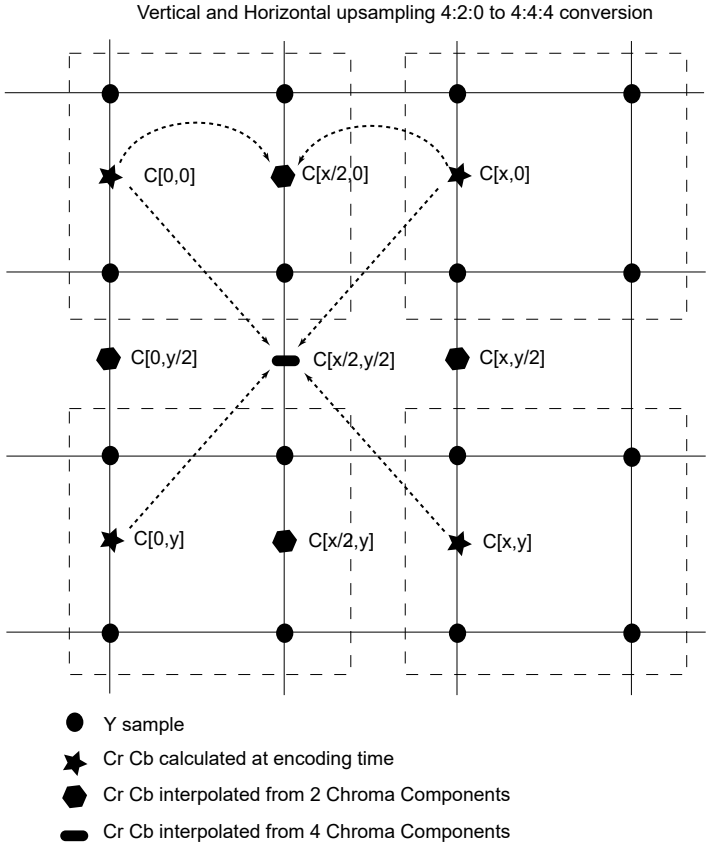
Figure 37-5. 4:2:2 Semiplanar and Planar Upsampling Algorithm - 90 or 270 Degree R Rotation Activated

Vertical and Horizontal upsampling 4:2:2 to 4:4:4 conversion 90 or 270 degrees



Downloaded from Arrow.com.

Figure 37-6. 4:2:0 Upsampling Algorithm



$$\text{Chroma}\left[\frac{x}{2}, 0\right] = \frac{\text{Cr}[0,0] + \text{Cr}[0,x]}{2}$$

$$\text{Chroma}\left[0, \frac{y}{2}\right] = \frac{\text{Cr}[0,0] + C[0,y]}{2}$$

$$\text{Chroma}\left[\frac{x}{2}, \frac{y}{2}\right] = \frac{\text{Cr}[0,0] + \text{Cr}[x,0] + \text{Cr}[y,0] + \text{Cr}[x,y]}{4}$$

$$\text{Chroma}\left[x, \frac{y}{2}\right] = \frac{\text{Cr}[x,0] + \text{Cr}[x,y]}{2}$$

$$\text{Chroma}\left[\frac{x}{2}, y\right] = \frac{\text{Cr}[0,y] + \text{Cr}[x,y]}{2}$$

37.6.6.1 Chrominance Upsampling Algorithm

1. Read line n from chrominance cache and interpolate [x/2,0] chrominance component filling the 1 x 2 kernel with line n. If the chrominance cache is empty, then fetch the first line from external memory and interpolate from the external memory. Duplicate the last chrominance at the end of line.
2. Fetch line n+1 from external memory, write line n + 1 to chrominance cache, read line n from the chrominance cache. Interpolate [0,y/2], [x/2,y/2] and [x, y/2] filling the 2x2 kernel with lines n and n+1. Duplicate the last chrominance line to generate the last interpolated line.
3. Repeat step 1 and step 2.

37.6.7 Line and Pixel Striding

The LCD module includes a technique to increment the memory address by a programmable amount when the end of line has been reached. This offset is referred to as XSTRIDE and is defined on a per overlay basis. Additionally, the PSTRIDE field allows a programmable jump at the pixel level. Pixel stride is the value from one pixel to the next.

Downloaded from Arrow.com.

37.6.7.1 Line Striding

When the end of line has been reached, the DMA address counter points to the next pixel address. The channel DMA address register is added to the XSTRIDE field, and then updated. If XSTRIDE is set to '0', the DMA address register remains unchanged. The XSTRIDE field of the channel configuration register is aligned to the pixel size boundary. The XSTRIDE field is a two's complement number. The following formula applies at the line boundary and indicates how the DMA controller computes the next pixel address. The function Sizeof() returns the number of bytes required to store a pixel.

$$\text{NextPixelAddress} = \text{CurrentPixelAddress} + \text{Sizeof}(\text{pixel}) + \text{XSTRIDE}$$

37.6.7.2 Pixel Striding

The DMA channel engine may optionally fetch non-contiguous pixels. The channel DMA address register is added to the PSTRIDE field and then updated. If PSTRIDE is set to zero, the DMA address register remains unchanged and pixels are contiguous. The PSTRIDE field of the channel configuration register is aligned to the pixel size boundary. The PSTRIDE is a two's complement number. The following formula applies at the pixel boundary and indicates how the DMA controller computes the next pixel address. The function Sizeof() returns the number of bytes required to store a pixel.

$$\text{NextPixelAddress} = \text{CurrentPixelAddress} + \text{Sizeof}(\text{pixel}) + \text{PSTRIDE}$$

37.6.8 Color Space Conversion Unit

The color space conversion unit converts Luminance Chrominance color space into the Red Green Blue color space. The conversion matrix is defined below and is fully programmable through the LCD user interface.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} \text{CSCRY} & \text{CSCRU} & \text{CSCRV} \\ \text{CSCGY} & \text{CSCGU} & \text{CSCGV} \\ \text{CSCBY} & \text{CSCBU} & \text{CSCBV} \end{bmatrix} \cdot \begin{bmatrix} Y - \text{Yoff} \\ \text{Cb} - \text{Cboff} \\ \text{Cr} - \text{Croff} \end{bmatrix}$$

Color space conversion coefficients are defined with the following equation:

$$\text{CSC}_{\text{ij}} = \frac{1}{2^7} \cdot \left[-2^9 \cdot c_9 + \sum_{n=0}^{8} c_n \cdot 2^n \right]$$

Color space conversion coefficients are defined with one sign bit, 2 integer bits and 7 fractional bits. The range of the CSCij coefficients is defined below with a step of 1/128.

$$-4 \leq \text{CSC}_{\text{ij}} \leq 3.9921875$$

Additionally, a set scaling factor {Yoff, Cboff, Croff} can be applied.

37.6.9 Two-Dimension Scaler

The High-End Overlay (HEO) data path includes a hardware scaler that allows an image resize in both the horizontal and the vertical directions.

37.6.9.1 Video Scaler Description

The scaling operation is based on a vertical and horizontal resampling algorithm. The sampling rate of the original image is increased when the video is upscaled, and decreased when the video is downscaled. A Vertical resampler is used to perform a vertical interpolation by a factor of vI, and a decimation by a factor of vD. A Horizontal resampler is used to perform a vertical interpolation by a factor of hI, and a decimation by a factor of hD. The horizontal and vertical low pass filters are both designed to minimize the aliasing effect. The frequency response of the low pass filter has the following characteristics:

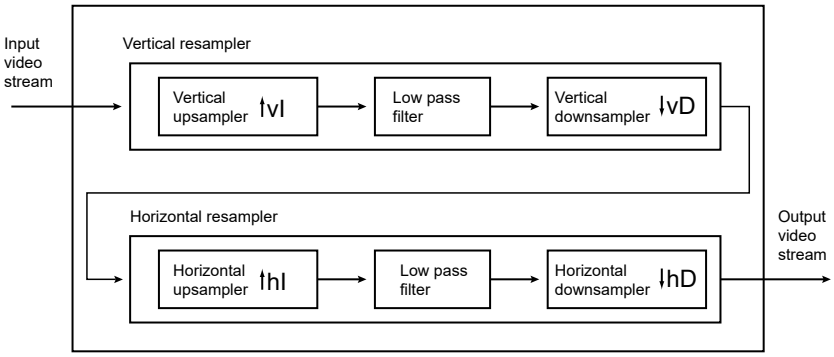
$$H(\omega) = \begin{cases} I & \text{when } 0 \leq |\omega| \leq \min\left(\frac{\pi}{I}, \frac{\pi}{D}\right) \\ 0 & \text{otherwise} \end{cases}$$

Taking into account the linear phase condition and anticipating the filter length M, the desired frequency response is modified.

$$H(\omega) = \begin{cases} Ie^{-\text{j}\omega\frac{M}{2}} & \text{when } 0 \leq |\omega| \leq \min\left(\frac{\pi}{I}, \frac{\pi}{D}\right) \\ 0 & \text{otherwise} \end{cases}$$

Downloaded from Arrow.com.

Figure 37-7. Video Resampler Architecture



The impulse response of the defined low pass filter is:

$$h(n) = \begin{cases} I \times \frac{\omega_c}{\pi} \text{ when } n = 0 \\ I \times \frac{\omega_c}{\pi} \times \frac{\sin\omega cn}{\omega_c n} \text{otherwise} \end{cases}$$

Or, for the filter of length M:

$$h(n) = \begin{cases} I \times \frac{\omega_c}{\pi} \text{ when } n = \frac{M}{2} \\ I \times \frac{\omega_c}{\pi} \times \frac{\sin\omega cn - M2}{\omega_c\left(n - \frac{M}{2}\right)} \text{otherwise} \end{cases}$$

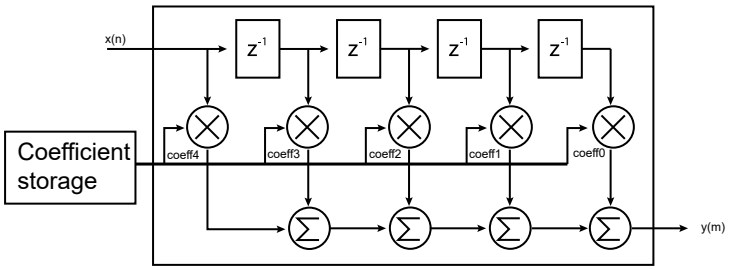
This ideal filter is non-causal and cannot be realized. The unit sample response h(n) is infinite in duration and must be truncated depending on the expected length M of the filter. This truncation is equivalent to the multiplication of the impulse response by a window function w(n).

Table 37-46. Window Function for a Filter Length M

Name of Window Function	Time Domain Sequence w(n)
Barlett	$1 - \frac{2 \times \left\lvert n - \frac{M-1}{2} \right\rvert}{M-1}$
Blackman	$0.42 - 0.5 \times \cos\frac{2\pi n}{M-1} + 0.08 \times \cos\frac{4\pi n}{M-1}$
Hamming	$0.54 - 0.46 \times \cos\frac{2\pi n}{M-1}$
Hanning	$0.5 - 0.5 \times \cos\frac{2\pi n}{M-1}$

The horizontal resampler includes an 8-phase 5-tap filter equivalent to a 40-tap FIR described in the figure below.

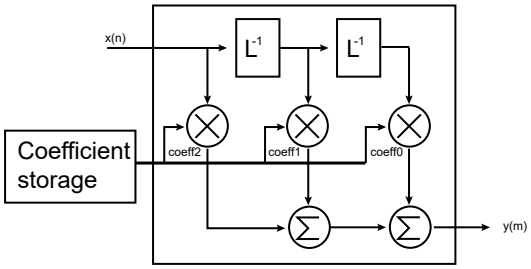
Figure 37-8. Horizontal Resampler Filter Architecture



Downloaded from Arrow.com.

The vertical resampler includes an 8-phase 3-tap filter equivalent to a 24-tap FIR described in the figure below.

Figure 37-9. Vertical Resampler Filter Architecture



37.6.9.2 Horizontal Scaler

The XMEMSIZE field of the LCDC_HEOCFG4 register indicates the horizontal size minus one of the image in the system memory. The XSIZE field of the LCDC_HEOCFG3 register contains the horizontal size minus one of the window. The SCALEN bit of the LCDC_HEOCFG13 register is set to '1'. The scaling factor is programmed in the XFACTOR field of the LCDC_HEOCFG13 register. Use the following algorithm to find the XFACTOR value:

$$\mathrm{XFACTOR}_{1\mathrm{st}} = \mathrm{floor}\left(\frac{8 \times 256 \times \mathrm{XMEMSIZE} - 256 \times \mathrm{XPHIDEF}}{\mathrm{XSIZE}}\right)$$

$$\mathrm{XFACTOR}_{1\mathrm{st}} = \mathrm{XFACTOR}_{1\mathrm{st}} + 1$$

$$\mathrm{XMEMSIZE}_{\max} = \mathrm{floor}\left(\frac{\mathrm{XFACTOR}_{1\mathrm{st}} \times \mathrm{XSIZE} + 256 \times \mathrm{XPHIDEF}}{2048}\right)$$

$$\begin{cases} \mathrm{XFACTOR} = \mathrm{XFACTOR}_{1\mathrm{st}} - 1 \mathrm{when}\left(\mathrm{XMEMSIZE}_{\max} > \mathrm{XMEMSIZE}\right) \\ \mathrm{XFACTOR} = \mathrm{XFACTOR}_{1\mathrm{st}} \mathrm{otherwise} \end{cases}$$

37.6.9.3 Vertical Scaler

The YMEMSIZE field of the LCDC_HEOCFG4 register indicates the vertical size minus one of the image in the system memory. The YSIZE field of the LCDC_HEOCFG3 register contains the vertical size minus one of the window. The SCALEN bit of the LCDC_HEOCFG13 register is set to one. The scaling factor is programmed in the YFACTOR field of the LCDC_HEOCFG13 register.

$$\mathrm{YFACTOR}_{1\mathrm{st}} = \mathrm{floor}\left(\frac{8 \times 256 \times \mathrm{YMEMSIZE} - 256 \times \mathrm{YPHIDEF}}{\mathrm{YSIZE}}\right)$$

$$\mathrm{YFACTOR}_{1\mathrm{st}} = \mathrm{YFACTOR}_{1\mathrm{st}} + 1$$

$$\mathrm{YMEMSIZE}_{\max} = \mathrm{floor}\left(\frac{\mathrm{YFACTOR}_{1\mathrm{st}} \times \mathrm{YSIZE} + 256 \times \mathrm{YPHIDEF}}{2048}\right)$$

$$\begin{cases} \mathrm{YFACTOR} = \mathrm{YFACTOR}_{1\mathrm{st}} - 1 \mathrm{when}\left(\mathrm{YMEMSIZE}_{\max} > \mathrm{YMEMSIZE}\right) \\ \mathrm{YFACTOR} = \mathrm{YFACTOR}_{1\mathrm{st}} \mathrm{otherwise} \end{cases}$$

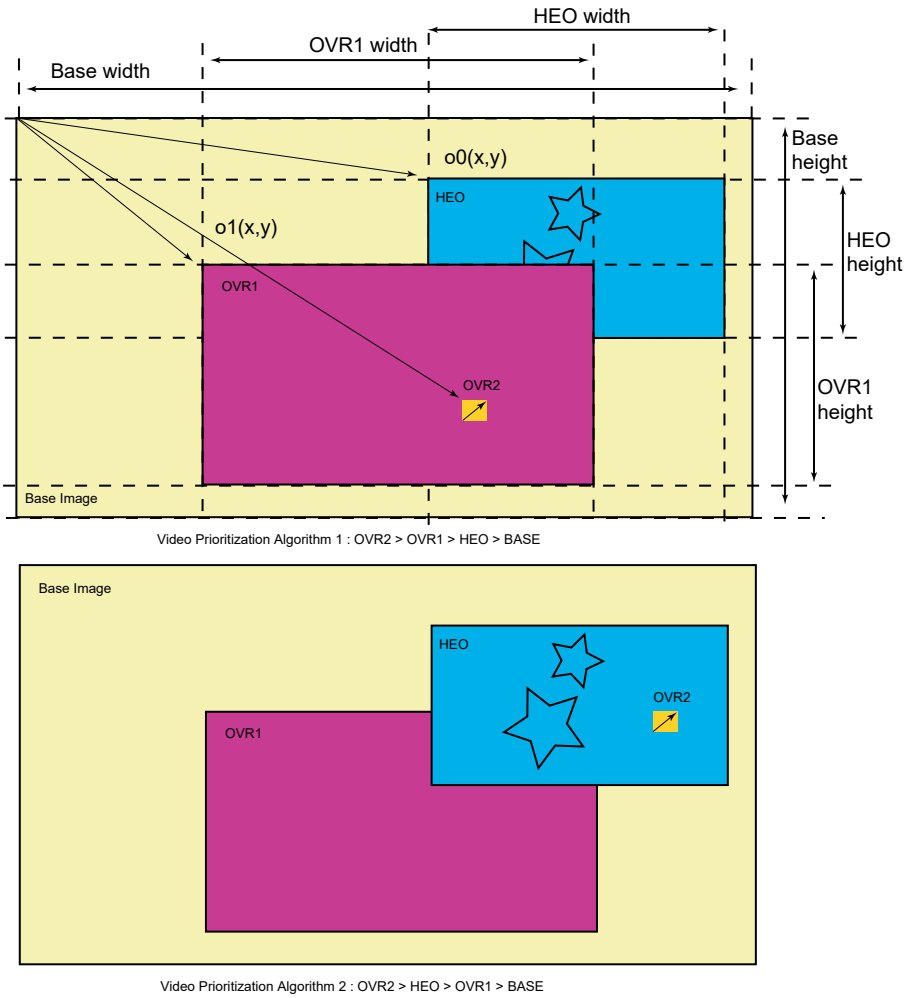
37.6.10 Color Combine Unit

37.6.10.1 Window Overlay

The LCD module provides hardware support for multiple "overlay plane" that can be used to display windows on top of the image without destroying the image located below. The overlay image can use any color depth. Using the overlay alleviates the need to re-render the occluded portion of the image. When pixels are combined together through the alpha blending unit, a new color is created. This new pixel is called an iterated pixel and is passed to the next blending stage. Then, this pixel may be combined again with another pixel. The VIDPRI bit located in the LCDC_HEOCFG12 register configures the video priority algorithm used to display the layers. When the VIDPRI bit is written to '0', the OVR1 layer is located above the HEO layer. When the VIDPRI bit is written to '1', OVR1 is located below the HEO layer.

Downloaded from Arrow.com.

Figure 37-10. Overlay Example with Two Different Video Prioritization Algorithms



37.6.10.2 Base Layer with Window Overlay Optimization

When the base layer is combined with at least one active overlay (100% opacity overlay), by default, the whole base layer frame is retrieved from the memory though it is not visible.

To optimize the system bandwidth, the LCDC can be configured to prevent the unuseful data from being fetched from system memory.

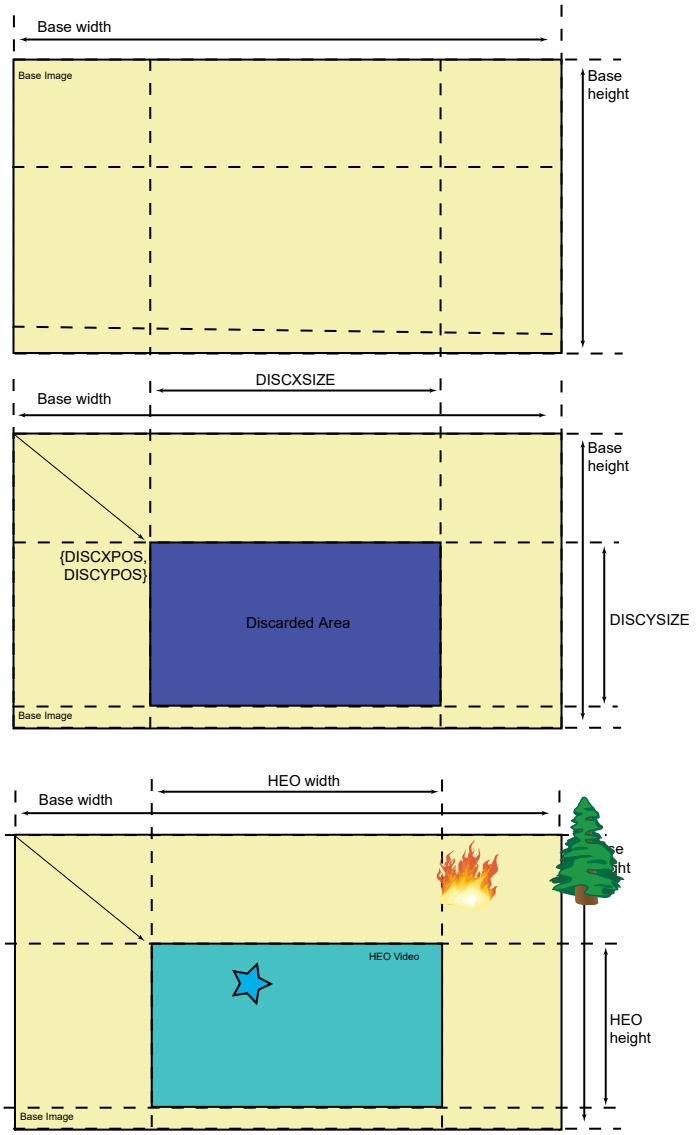
The following registers are used to disable an invisible area of the base layer:

- LCDC_BASECFG5:
  - field DISCXPOS (Discard Area Horizontal Position)
  - field DISCYPOS (Discard Area Vertical Position)
- LCDC_BASECFG6:
  - field DISCXSIZE (Discard Area Horizontal Size)
  - field DISCYSIZE (Discard Area Vertical Size)
- LCDC_BASECFG4: bit DISCEN (Discard Area Enable)

Each time the overlay window is resized and/or moved, these configuration registers must be reconfigured according to the new overlay window features.

Downloaded from Arrow.com.

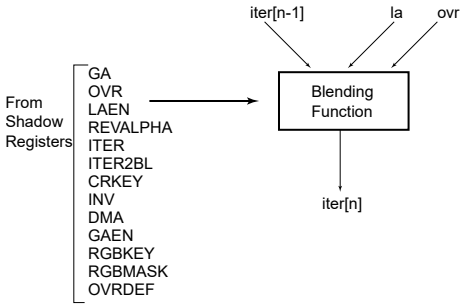
Figure 37-11. Base Layer Discard Area



37.6.10.3 Overlay Blending

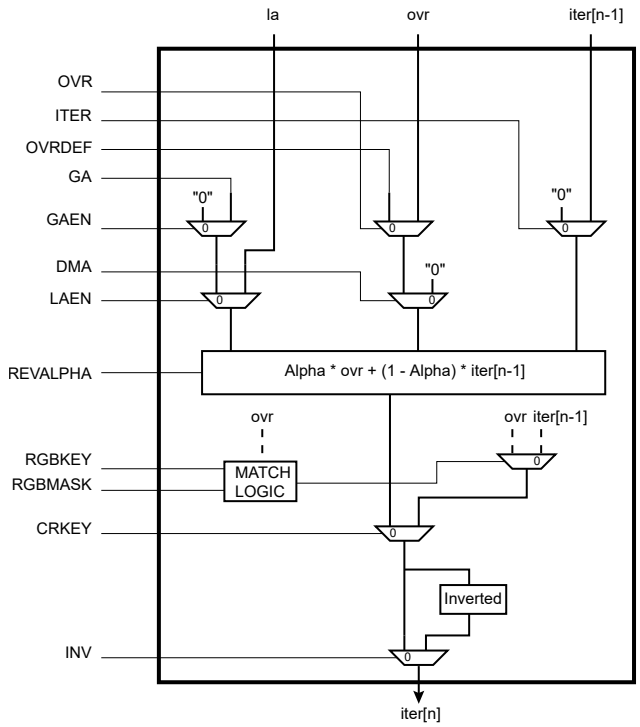
The blending function requires two pixels (one iterated from the previous blending stage and one from the current overlay color) and a set of blending configuration parameters. These parameters define the color operation.

Figure 37-12. Alpha Blender Function



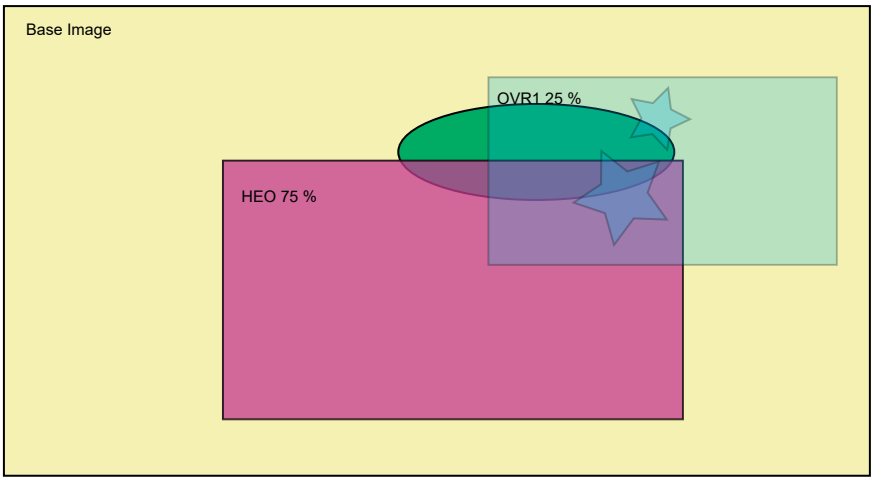
Downloaded from Arrow.com.

Figure 37-13. Alpha Blender Database



37.6.10.4 Window Blending

Figure 37-14. 256-level Alpha Blending



Video Prioritization Algorithm 1: OVR1 > HEO > BASE

37.6.10.5 Color Keying

Color keying involves a method of bit-block image transfer (Blit). This entails blitting one image onto another where not all the pixels are copied. Blitting usually involves two bitmaps: a source bitmap and a destination bitmap. A raster operation (ROP) is performed to define whether the iterated color or the overlay color is to be visible or not.

37.6.10.5.1 Source Color Keying

If the masked overlay color matches the color key, the iterated color is selected and Source Color Keying is activated using the following configuration sequence:

1. Select the overlay to blit.
2. Write a ‘0’ to DSTKEY.

Downloaded from Arrow.com.

3. Activate Color Keying by writing a ‘1’ to CRKEY.
4. Configure the Color Key by writing RKEY, GKEY and BKEY fields.
5. Configure the Color Mask by writing RKEY, GKEY and BKEY fields.

When the field RMASK, GMASK, or BMASK is configured to ‘0’, the comparison is disabled and the raster operation is activated.

37.6.10.5.2 Destination Color Keying

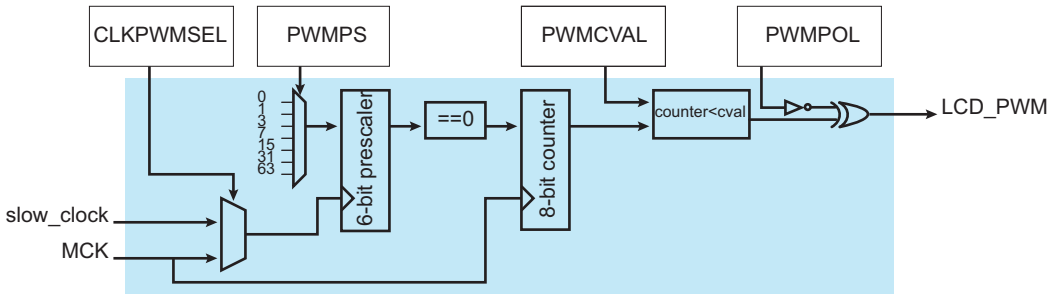
If the iterated masked color matches the color key then the overlay color is selected, Destination Color Keying is activated using the following configuration sequence:

1. Select the overlay to blit.
2. Write a ‘1’ to DSTKEY.
3. Activate Color Keying by writing a ‘1’ to CRKEY bit
4. Configure the Color Key by writing RKEY, GKEY and BKEY fields.
5. Configure the Color Mask by writing RKEY, GKEY and BKEY fields.

When the field RMASK, GMASK, or BMASK is configured to ‘0’, the comparison is disabled and the raster operation is activated.

37.6.11 LCDC PWM Controller

Figure 37-15. PWM Controller Block Diagram



This block generates the LCD contrast control signal (LCDPWM) to make possible the control of the display's contrast by software. This is an 8-bit PWM (Pulse Width Modulation) signal that can be converted to an analog voltage with a simple passive filter.

The PWM module has a free-running counter whose value is compared against a compare register (PWMCVAL field of the LCDC_LCDCFG6 register). If the value in the counter is less than that in the register, the output brings the value of the signal polarity (PWMPOL) bit in the PWM control register: LCDC_LCDCFG6. Otherwise, the opposite value is output. Thus, a periodic waveform with a pulse width proportional to the value in the compare register is generated.

Due to the comparison mechanism, the output pulse has a width between zero and 255 PWM counter cycles. Thus by adding a simple passive filter outside the chip, an analog voltage between 0 and (255/256) × V_{DD} can be obtained (for the positive polarity case, or between (1/256) × V_{DD} and V_{DD} for the negative polarity case). Other voltage values can be obtained by adding active external circuitry.

For PWM mode, the counter frequency can be adjusted to four different values using the PWMPS field of the LCDC_LCDCFG6 register.

The PWM module can be fed with the slow clock or the system clock, depending on the CLKPWMSEL bit of the LCDC_CFG0 register.

LCD display panels have different backlight specifications in terms of minimum/maximum values for PWM frequency. if the LCDC PWM frequency range does not match the LCD display panel, it is possible to use the product standalone PWM controller to drive the backlight.

Downloaded from Arrow.com.

37.6.12 LCD Overall Performance

37.6.12.1 Color Lookup Table (CLUT)

Table 37-47. CLUT Pixel Performance

CLUT Mode	Pixels/Cycle	Rotation	Scaling
1 bpp	64	Not supported	Supported
2 bpp	32	Not supported	Supported
3 bpp	16	Not supported	Supported
4 bpp	8	Not supported	Supported

37.6.12.2 RGB Mode Fetch Performance

Table 37-48. RGB Mode Performance

RGB Mode	Pixels/Cycle Memory Burst Mode	Rotation Peak Random Memory Access (pixels/cycle)		Scaling Burst Mode or Rotation Optimization Available
		Rotation Optimization (see Note 1)	Normal Mode	
12 bpp	4	1	0.2	Supported
16 bpp	4	1	0.2	Supported
18 bpp	2	1	0.2	Supported
18 bpp RGB PACKED	2.666	Not supported	0.2	Supported
19 bpp	2	1	0.2	Supported
19 bpp PACKED	2.666	Not Supported	0.2	Supported
24 bpp	2	1	0.2	Supported
24 bpp PACKED	2.666	Not Supported	0.2	Supported
25 bpp	2	1	0.2	Supported
32 bpp	2	1	0.2	Supported

Note:

1. Rotation optimization = System bus lock asserted on consecutive single access.

37.6.12.3 YUV Mode Fetch Performance

Table 37-49. Single Stream for 0 Wait State Memory

YUV Mode	Pixels/Cycle Memory Burst Mode	Rotation Peak Random Memory Access (pixels/cycle)		Scaling Burst Mode or Rotation Optimization Is Available
		Rotation Optimization (see Note 1)	Normal Mode	
32 bpp AYUV	2	1	0.2	Supported
16 bpp 422	4	Not Supported	Not Supported	Supported

Note:

1. Rotation optimization = System bus lock asserted on consecutive single access.

Downloaded from Arrow.com.

Table 37-50. Multiple Stream for 0 Wait State Memory

YUV Mode	Comp/Cycle Memory Burst Mode	Rotation Peak Random Memory Access (pixels/cycle)		Scaling Burst Mode or Rotation Optimization Is Available
		Rotation Optimization	Normal Mode	
16 bpp 422 semiplanar	8 Y, 4 UV	1 Y, 1 UV (2 streams)	0.2 Y 0.2 UV (2 streams)	Supported
16 bpp 422 planar	8 Y, 8 U, 8 V	1 Y, 1 U, 1 V (3 streams)	0.2 Y, 0.2 U, 0.2 V (3 streams)	Supported
12 bpp 4:2:0 semiplanar	8 Y, 4 UV	1 Y, 1 UV (2 streams)	0.2 Y 0.2 UV (2 streams)	Supported
12 bpp 4:2:0 planar	8 Y, 8 U, 8 V	1 Y, 1 U, 1 V (3 streams)	0.2 Y, 0.2 U, 0.2 V (3 streams)	Supported

Table 37-51. YUV Planar Overall Performance 1 System Bus Interface for 0 Wait State Memory

YUV Mode	Pix/Cycle Memory Burst Mode	Rotation Peak Random Memory Access (pixels/cycle)		Scaling Burst Mode or Rotation Optimization Is Available
		Rotation Optimization	Normal Mode	
16 bpp 422 semiplanar	4	0.66	0.132	Supported
16 bpp 422 planar	4	0.5	0.1	Supported
12 bpp 4:2:0 semiplanar	5.32	0.8	0.16	Supported
12 bpp 4:2:0 planar	5.32	0.66	0.132	Supported

37.6.13 Input FIFO

The LCD module includes one input FIFO per overlay. These input FIFOs are used to buffer the system bus burst and serialize the stream of pixels.

37.6.14 Output FIFO

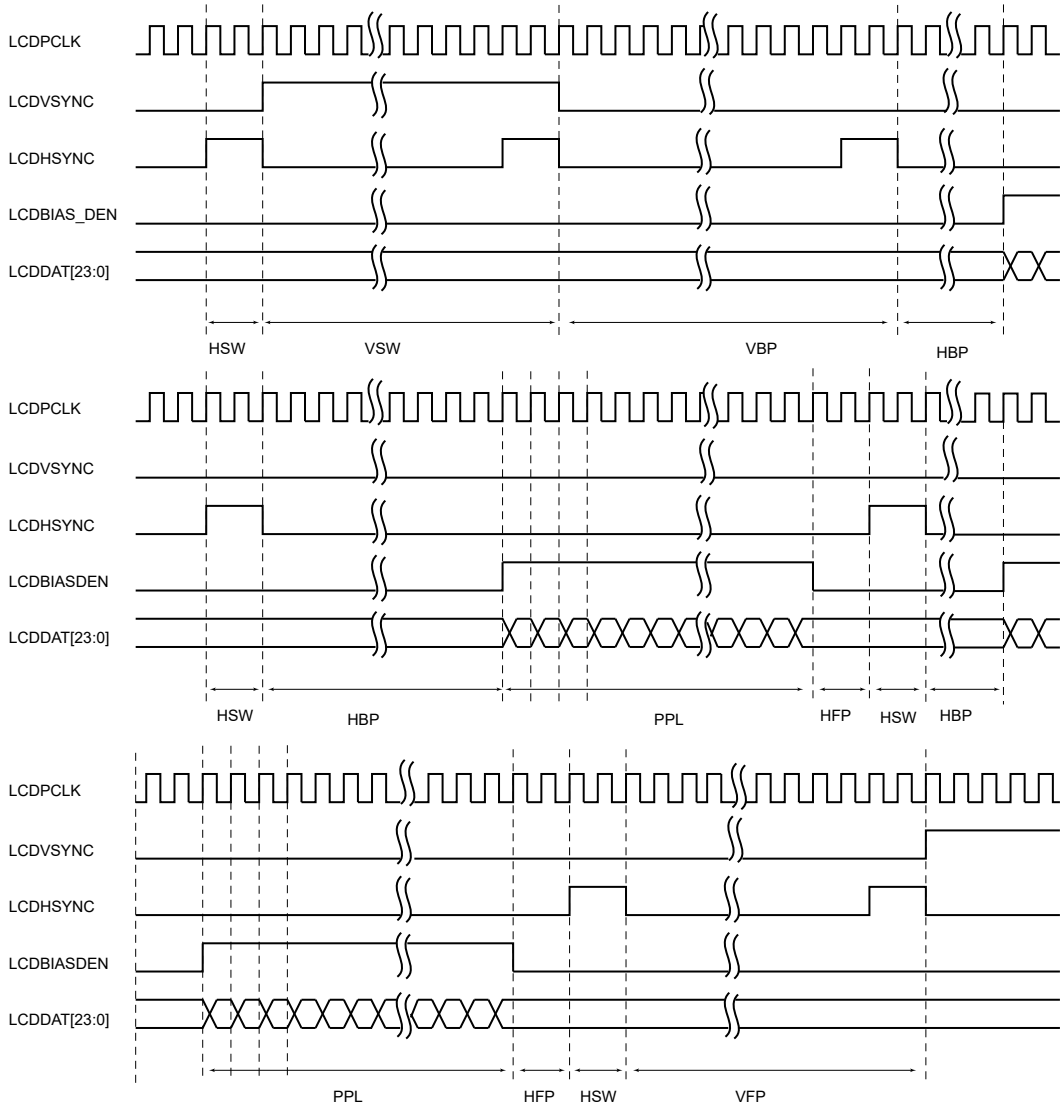
The LCD module includes one output FIFO that stores the blended pixel.

Downloaded from Arrow.com.

37.6.15 Output Timing Generation

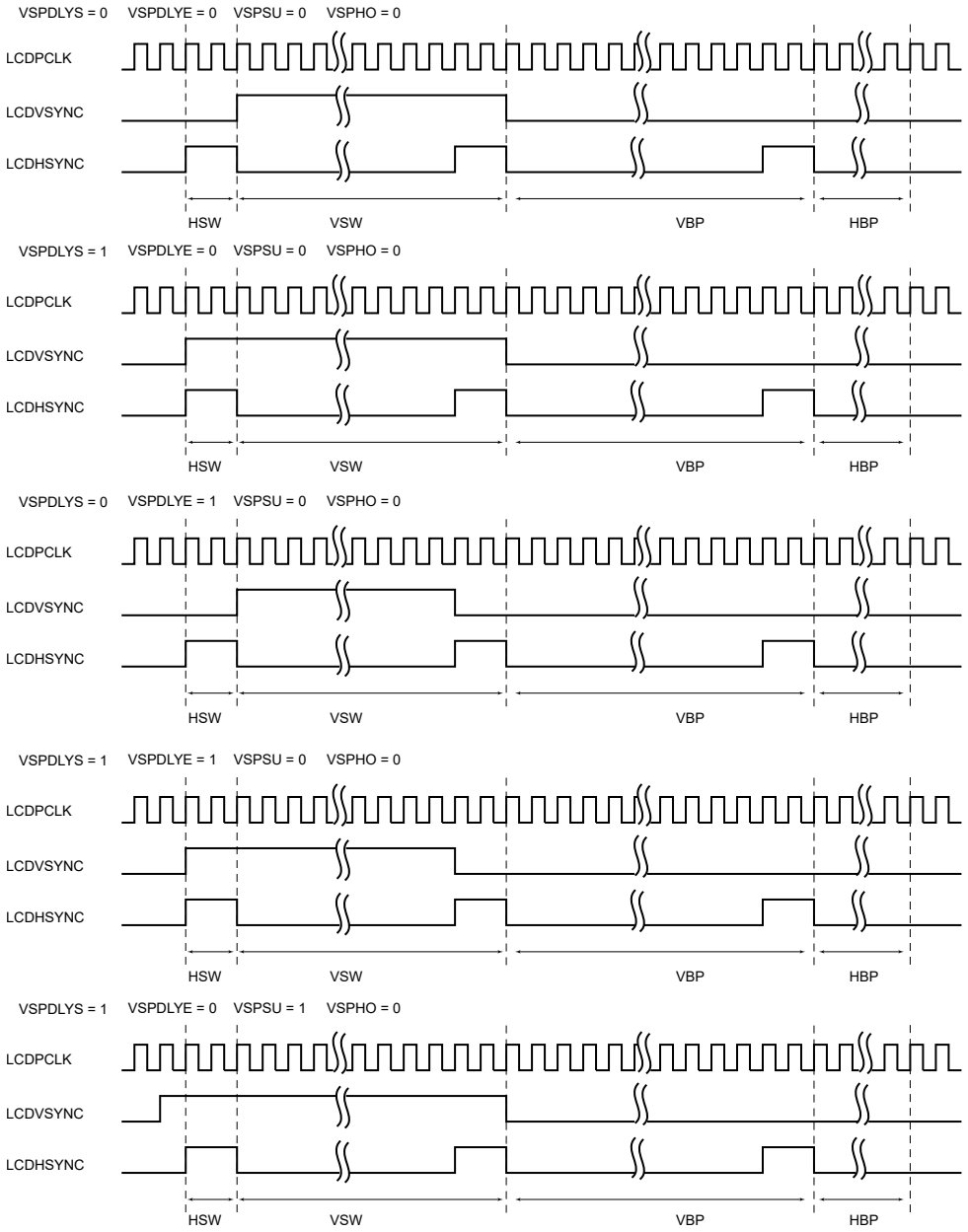
37.6.15.1 Active Display Timing Mode

Figure 37-16. Active Display Timing



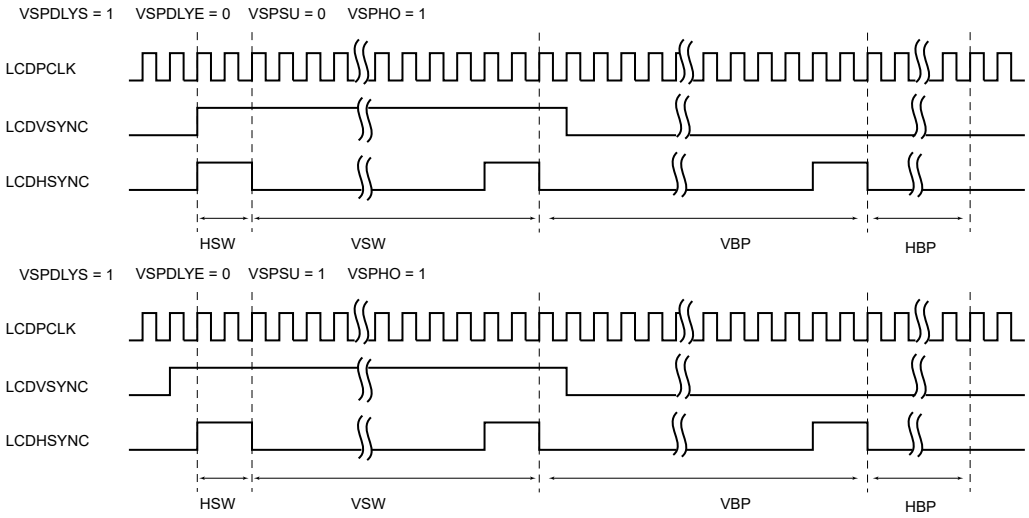
Downloaded from Arrow.com.

Figure 37-17. Vertical Synchronization Timing (part 1)



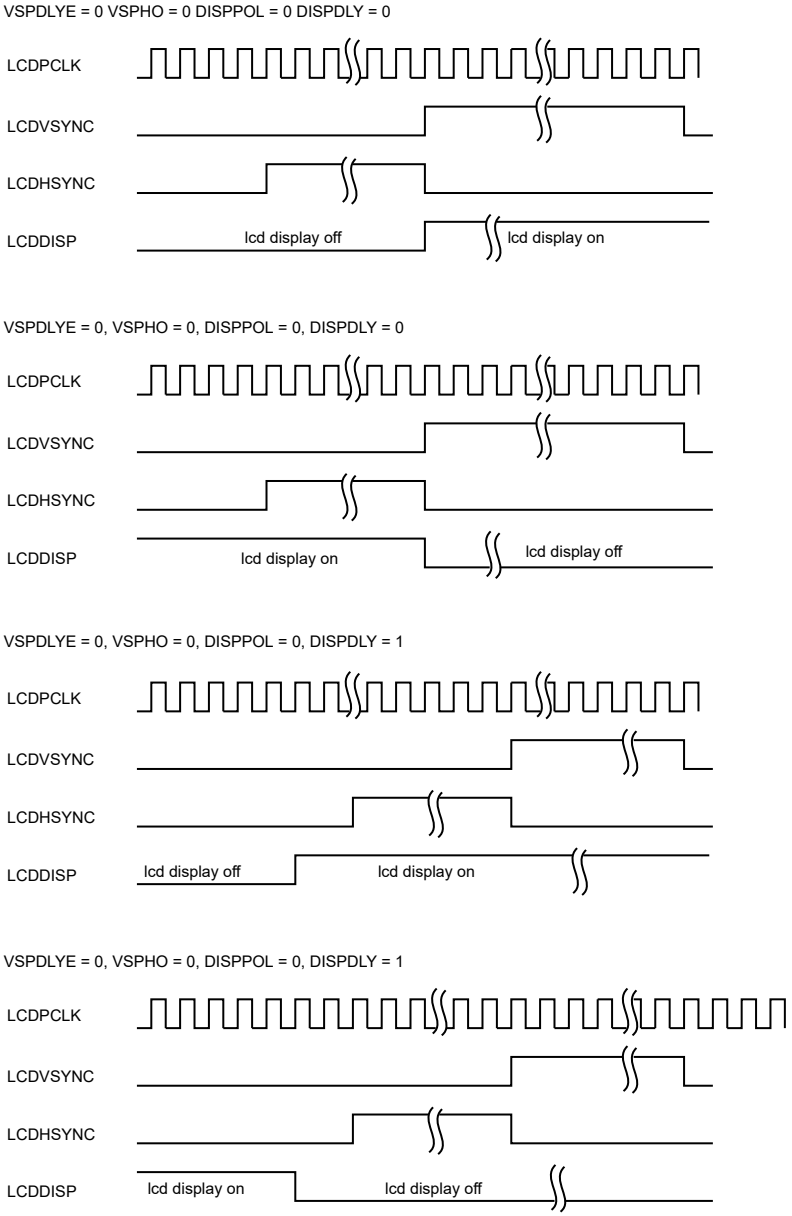
Downloaded from Arrow.com.

Figure 37-18. Vertical Synchronization Timing (part 2)



Downloaded from Arrow.com.

Figure 37-19. DISP Signal Timing Diagram



37.6.16 Output Format

37.6.16.1 Active Mode Output Pin Assignment

Table 37-52. Active Mode Output with 24-bit Bus Interface Configuration

Pin ID	24-bit TFT	18-bit TFT	16-bit TFT	12-bit TFT
LCDDAT[23]	R[7]	R[5]	R[4]	R[3]
LCDDAT[22]	R[6]	R[4]	R[3]	R[2]
LCDDAT[21]	R[5]	R[3]	R[2]	R[1]
LCDDAT[20]	R[4]	R[2]	R[1]	R[0]
LCDDAT[19]	R[3]	R[1]	R[0]	-

Downloaded from Arrow.com.

.....continued

Pin ID	24-bit TFT	18-bit TFT	16-bit TFT	12-bit TFT
LCDDAT[18]	R[2]	R[0]	-	-
LCDDAT[17]	R[1]	-	-	-
LCDDAT[16]	R[0]	-	-	-
LCDDAT[15]	G[7]	G[5]	G[5]	G[3]
LCDDAT[14]	G[6]	G[4]	G[4]	G[2]
LCDDAT[13]	G[5]	G[3]	G[3]	G[1]
LCDDAT[12]	G[4]	G[2]	G[2]	G[0]
LCDDAT[11]	G[3]	G[1]	G[1]	-
LCDDAT[10]	G[2]	G[0]	G[0]	-
LCDDAT[9]	G[1]	-	-	-
LCDDAT[8]	G[0]	-	-	-
LCDDAT[7]	B[7]	B[5]	B[4]	B[3]
LCDDAT[6]	B[6]	B[4]	B[3]	B[2]
LCDDAT[5]	B[5]	B[3]	B[2]	B[1]
LCDDAT[4]	B[4]	B[2]	B[1]	B[0]
LCDDAT[3]	B[3]	B[1]	B[0]	-
LCDDAT[2]	B[2]	B[0]	-	-
LCDDAT[1]	B[1]	-	-	-
LCDDAT[0]	B[0]	-	-	-

Downloaded from Arrow.com.

37.7 Register Summary

Offset	Name	Bit Pos.								
0x00	LCDC_LCDCFG0	31:24								
		23:16	CLKDIV[7:0]							
		15:8			CGDISPP		CGDISHEO	CGDISOVR2	CGDISOVR1	CGDISBASE
		7:0					CLKPWMSEL			CLKPOL
0x04	LCDC_LCDCFG1	31:24							VSPW[9:8]	
		23:16	VSPW[7:0]							
		15:8							HSPW[9:8]	
		7:0	HSPW[7:0]							
0x08	LCDC_LCDCFG2	31:24							VBPW[9:8]	
		23:16	VBPW[7:0]							
		15:8							VFPW[9:8]	
		7:0	VFPW[7:0]							
0x0C	LCDC_LCDCFG3	31:24							HBPW[9:8]	
		23:16	HBPW[7:0]							
		15:8							HFPW[9:8]	
		7:0	HFPW[7:0]							
0x10	LCDC_LCDCFG4	31:24						RPF[10:8]		
		23:16	RPF[7:0]							
		15:8						PPL[10:8]		
		7:0	PPL[7:0]							
0x14	LCDC_LCDCFG5	31:24								
		23:16	GUARDTIME[7:0]							
		15:8			VSPHO	VSPSU			MODE[1:0]	
		7:0	DISPDLY	DITHER		DISPPOL	VSPDLYE	VSPDLYS	VSPOL	HSPOL
0x18	LCDC_LCDCFG6	31:24								
		23:16								
		15:8	PWMCVAL[7:0]							
		7:0				PWMPOL		PWMPS[2:0]		
0x1C	LCDC_LCDCFG7	31:24								
		23:16								
		15:8						ROW[10:8]		
		7:0	ROW[7:0]							
0x20	LCDC_LCDEN	31:24								
		23:16								
		15:8								
		7:0					PWMEN	DISPEN	SYNCEN	CLKEN
0x24	LCDC_LCDDIS	31:24								
		23:16								
		15:8					PWMRST	DISPRST	SYNCRST	CLKRST
		7:0					PWMDIS	DISPDIS	SYNCDIS	CLKDIS
0x28	LCDC_LCDSR	31:24								
		23:16								
		15:8								
		7:0				SIPSTS	PWMSTS	DISPSTS	LCDSTS	CLKSTS
0x2C	LCDC_LCDIER	31:24								
		23:16								
		15:8			PPIE		HEOIE	OVR2IE	OVR1IE	BASEIE
		7:0				FIFOERRIE	ROWIE	DISPIE	DISIE	SOFIE
0x30	LCDC_LCDIDR	31:24								
		23:16								
		15:8			PPID		HEOID	OVR2ID	OVR1ID	BASEID
		7:0				FIFOERRID	ROWID	DISPID	DISID	SOFID
0x34	LCDC_LCDIMR	31:24								
		23:16								
		15:8			PPIM		HEOIM	OVR2IM	OVR1IM	BASEIM
		7:0				FIFOERRIM	ROWIM	DISPIM	DISIM	SOFIM

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x38	LCDC_LCDISR	31:24								
		23:16								
		15:8			PP		HEO	OVR2	OVR1	BASE
		7:0				FIFOERR	ROW	DISP	DIS	SOF
0x3C	LCDC_ATTR	31:24								
		23:16								
		15:8			PPA2Q		HEOA2Q	OVR2A2Q	OVR1A2Q	BASEA2Q
		7:0			PP		HEO	OVR2	OVR1	BASE
0x40	LCDC_QOSCFG	31:24								
		23:16								
		15:8			QOS3CFG[1:0]				QOS2CFG[1:0]	
		7:0			QOS1CFG[1:0]					QOSLOCK
0x44	LCDC_QOS1M	31:24								
		23:16					MET1[19:16]			
		15:8	MET1[15:8]							
		7:0	MET1[7:0]							
0x48	LCDC_QOS2M	31:24								
		23:16					MET1[19:16]			
		15:8	MET1[15:8]							
		7:0	MET1[7:0]							
0x4C	LCDC_QOS3M	31:24								
		23:16					MET1[19:16]			
		15:8	MET1[15:8]							
		7:0	MET1[7:0]							
0x50	LCDC_QOSMIN	31:24								
		23:16								
		15:8								LEVEL[8]
		7:0	LEVEL[7:0]							
0x54 ... 0x5F	Reserved									
0x60	LCDC_BASECHER	31:24								
		23:16								
		15:8								
		7:0						A2QEN	UPDATEEN	CHEN
0x64	LCDC_BASECHDR	31:24								
		23:16								
		15:8								CHRST
		7:0								CHDIS
0x68	LCDC_BASECHSR	31:24								
		23:16								
		15:8								
		7:0						A2QSR	UPDATESR	CHSR
0x6C	LCDC_BASEIER	31:24								
		23:16								
		15:8								
		7:0		OVR	DONE	ADD	DSCR	DMA		
0x70	LCDC_BASEIDR	31:24								
		23:16								
		15:8								
		7:0		OVR	DONE	ADD	DSCR	DMA		
0x74	LCDC_BASEIMR	31:24								
		23:16								
		15:8								
		7:0		OVR	DONE	ADD	DSCR	DMA		
0x78	LCDC_BASEISR	31:24								
		23:16								
		15:8								
		7:0		OVR	DONE	ADD	DSCR	DMA		

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x7C	LCDC_BASEHEAD	31:24	HEAD[29:22]							
		23:16	HEAD[21:14]							
		15:8	HEAD[13:6]							
		7:0	HEAD[5:0]							
0x80	LCDC_BASEADDR	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x84	LCDC_BASECTRL	31:24								
		23:16								
		15:8								
		7:0			DONEIEN	ADDIEN	DSCRIEN	DMAIEN	LFETCH	DFETCH
0x88	LCDC_BASENEXT	31:24	NEXT[31:24]							
		23:16	NEXT[23:16]							
		15:8	NEXT[15:8]							
		7:0	NEXT[7:0]							
0x8C	LCDC_BASECFG0	31:24								
		23:16								
		15:8								DLBO
		7:0			BLEN[1:0]					
0x90	LCDC_BASECFG1	31:24								
		23:16								
		15:8							CLUTMODE[1:0]	
		7:0	RGBMODE[3:0]							CLUTEN
0x94	LCDC_BASECFG2	31:24	XSTRIDE[31:24]							
		23:16	XSTRIDE[23:16]							
		15:8	XSTRIDE[15:8]							
		7:0	XSTRIDE[7:0]							
0x98	LCDC_BASECFG3	31:24								
		23:16	RDEF[7:0]							
		15:8	GDEF[7:0]							
		7:0	BDEF[7:0]							
0x9C	LCDC_BASECFG4	31:24								
		23:16								
		15:8					DISCEN		REP	DMA
		7:0								
0xA0	LCDC_BASECFG5	31:24						DISCYPOS[10:8]		
		23:16	DISCYPOS[7:0]							
		15:8						DISCXPOS[10:8]		
		7:0	DISCXPOS[7:0]							
0xA4	LCDC_BASECFG6	31:24						DISCYSIZE[10:8]		
		23:16	DISCYSIZE[7:0]							
		15:8						DISCXSIZE[10:8]		
		7:0	DISCXSIZE[7:0]							
0xA8 ... 0x015F	Reserved									
0x0160	LCDC_OVR1CHER	31:24								
		23:16								
		15:8								
		7:0						A2QEN	UPDATEEN	CHEN
0x0164	LCDC_OVR1CHDR	31:24								
		23:16								
		15:8								CHRST
		7:0								CHDIS
0x0168	LCDC_OVR1CHSR	31:24								
		23:16								
		15:8								
		7:0						A2QSR	UPDATESR	CHSR

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x016C	LCDC_OVR1IER	31:24								
		23:16								
		15:8								
		7:0		OVR	DONE	ADD	DSCR	DMA		
0x0170	LCDC_OVR1IDR	31:24								
		23:16								
		15:8								
		7:0		OVR	DONE	ADD	DSCR	DMA		
0x0174	LCDC_OVR1IMR	31:24								
		23:16								
		15:8								
		7:0		OVR	DONE	ADD	DSCR	DMA		
0x0178	LCDC_OVR1ISR	31:24								
		23:16								
		15:8								
		7:0		OVR	DONE	ADD	DSCR	DMA		
0x017C	LCDC_OVR1HEAD	31:24	HEAD[29:22]							
		23:16	HEAD[21:14]							
		15:8	HEAD[13:6]							
		7:0	HEAD[5:0]							
0x0180	LCDC_OVR1ADDR	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x0184	LCDC_OVR1CTRL	31:24								
		23:16								
		15:8								
		7:0			DONEIEN	ADDIEN	DSCRIEN	DMAIEN	LFETCH	DFETCH
0x0188	LCDC_OVR1NEXT	31:24	NEXT[31:24]							
		23:16	NEXT[23:16]							
		15:8	NEXT[15:8]							
		7:0	NEXT[7:0]							
0x018C	LCDC_OVR1CFG0	31:24								
		23:16								
		15:8			LOCKDIS	ROTDIS				DLBO
		7:0			BLEN[1:0]					
0x0190	LCDC_OVR1CFG1	31:24								
		23:16								
		15:8							CLUTMODE[1:0]	
		7:0	RGBMODE[3:0]							CLUTEN
0x0194	LCDC_OVR1CFG2	31:24						YPOS[10:8]		
		23:16	YPOS[7:0]							
		15:8						XPOS[10:8]		
		7:0	XPOS[7:0]							
0x0198	LCDC_OVR1CFG3	31:24						YSIZE[10:8]		
		23:16	YSIZE[7:0]							
		15:8						XSIZE[10:8]		
		7:0	XSIZE[7:0]							
0x019C	LCDC_OVR1CFG4	31:24	XSTRIDE[31:24]							
		23:16	XSTRIDE[23:16]							
		15:8	XSTRIDE[15:8]							
		7:0	XSTRIDE[7:0]							
0x01A0	LCDC_OVR1CFG5	31:24	PSTRIDE[31:24]							
		23:16	PSTRIDE[23:16]							
		15:8	PSTRIDE[15:8]							
		7:0	PSTRIDE[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x01A4	LCDC_OVR1CFG6	31:24								
		23:16	RDEF[7:0]							
		15:8	GDEF[7:0]							
		7:0	BDEF[7:0]							
0x01A8	LCDC_OVR1CFG7	31:24								
		23:16	RKEY[7:0]							
		15:8	GKEY[7:0]							
		7:0	BKEY[7:0]							
0x01AC	LCDC_OVR1CFG8	31:24								
		23:16	RMASK[7:0]							
		15:8	GMASK[7:0]							
		7:0	BMASK[7:0]							
0x01B0	LCDC_OVR1CFG9	31:24								
		23:16	GA[7:0]							
		15:8						DSTKEY	REP	DMA
		7:0	OVR	LAEN	GAEN	REVALPHA	ITER	ITER2BL	INV	CRKEY
0x01B4 ... 0x025F	Reserved									
0x0260	LCDC_OVR2CHER	31:24								
		23:16								
		15:8								
		7:0						A2QEN	UPDATEEN	CHEN
0x0264	LCDC_OVR2CHDR	31:24								
		23:16								
		15:8								CHRST
		7:0								CHDIS
0x0268	LCDC_OVR2CHSR	31:24								
		23:16								
		15:8								
		7:0						A2QSR	UPDATESR	CHSR
0x026C	LCDC_OVR2IER	31:24								
		23:16								
		15:8								
		7:0		OVR	DONE	ADD	DSCR	DMA		
0x0270	LCDC_OVR2IDR	31:24								
		23:16								
		15:8								
		7:0		OVR	DONE	ADD	DSCR	DMA		
0x0274	LCDC_OVR2IMR	31:24								
		23:16								
		15:8								
		7:0		OVR	DONE	ADD	DSCR	DMA		
0x0278	LCDC_OVR2ISR	31:24								
		23:16								
		15:8								
		7:0		OVR	DONE	ADD	DSCR	DMA		
0x027C	LCDC_OVR2HEAD	31:24	HEAD[29:22]							
		23:16	HEAD[21:14]							
		15:8	HEAD[13:6]							
		7:0	HEAD[5:0]							
0x0280	LCDC_OVR2ADDR	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x0284	LCDC_OVR2CTRL	31:24								
		23:16								
		15:8								
		7:0			DONEIEN	ADDIEN	DSCRIEN	DMAIEN	LFETCH	DFETCH

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x0288	LCDC_OVR2NEXT	31:24	NEXT[31:24]							
		23:16	NEXT[23:16]							
		15:8	NEXT[15:8]							
		7:0	NEXT[7:0]							
0x028C	LCDC_OVR2CFG0	31:24								
		23:16								
		15:8			LOCKDIS	ROTDIS				DLBO
		7:0			BLEN[1:0]					
0x0290	LCDC_OVR2CFG1	31:24								
		23:16								
		15:8							CLUTMODE[1:0]	
		7:0	RGBMODE[3:0]							CLUTEN
0x0294	LCDC_OVR2CFG2	31:24						YPOS[10:8]		
		23:16	YPOS[7:0]							
		15:8						XPOS[10:8]		
		7:0	XPOS[7:0]							
0x0298	LCDC_OVR2CFG3	31:24						YSIZE[10:8]		
		23:16	YSIZE[7:0]							
		15:8						XSIZE[10:8]		
		7:0	XSIZE[7:0]							
0x029C	LCDC_OVR2CFG4	31:24	XSTRIDE[31:24]							
		23:16	XSTRIDE[23:16]							
		15:8	XSTRIDE[15:8]							
		7:0	XSTRIDE[7:0]							
0x02A0	LCDC_OVR2CFG5	31:24	PSTRIDE[31:24]							
		23:16	PSTRIDE[23:16]							
		15:8	PSTRIDE[15:8]							
		7:0	PSTRIDE[7:0]							
0x02A4	LCDC_OVR2CFG6	31:24								
		23:16	RDEF[7:0]							
		15:8	GDEF[7:0]							
		7:0	BDEF[7:0]							
0x02A8	LCDC_OVR2CFG7	31:24								
		23:16	RKEY[7:0]							
		15:8	GKEY[7:0]							
		7:0	BKEY[7:0]							
0x02AC	LCDC_OVR2CFG8	31:24								
		23:16	RMASK[7:0]							
		15:8	GMASK[7:0]							
		7:0	BMASK[7:0]							
0x02B0	LCDC_OVR2CFG9	31:24								
		23:16	GA[7:0]							
		15:8						DSTKEY	REP	DMA
		7:0	OVR	LAEN	GAEN	REVALPHA	ITER	ITER2BL	INV	CRKEY
0x02B4 ... 0x035F	Reserved									
0x0360	LCDC_HEOCHER	31:24								
		23:16								
		15:8								
		7:0						A2QEN	UPDATEEN	CHEN
0x0364	LCDC_HEOCHDR	31:24								
		23:16								
		15:8								CHRST
		7:0								CHDIS
0x0368	LCDC_HEOCHSR	31:24								
		23:16								
		15:8								
		7:0						A2QSR	UPDATESR	CHSR

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x036C	LCDC_HEOIER	31:24								
		23:16		VOVR	VDONE	VADD	VDSCR	VDMA		
		15:8		UOVR	UDONE	UADD	UDSCR	UDMA		
		7:0		OVR	DONE	ADD	DSCR	DMA		
0x0370	LCDC_HEOIDR	31:24								
		23:16		VOVR	VDONE	VADD	VDSCR	VDMA		
		15:8		UOVR	UDONE	UADD	UDSCR	UDMA		
		7:0		OVR	DONE	ADD	DSCR	DMA		
0x0374	LCDC_HEOIMR	31:24								
		23:16		VOVR	VDONE	VADD	VDSCR	VDMA		
		15:8		UOVR	UDONE	UADD	UDSCR	UDMA		
		7:0		OVR	DONE	ADD	DSCR	DMA		
0x0378	LCDC_HEOISR	31:24								
		23:16		VOVR	VDONE	VADD	VDSCR	VDMA		
		15:8		UOVR	UDONE	UADD	UDSCR	UDMA		
		7:0		OVR	DONE	ADD	DSCR	DMA		
0x037C	LCDC_HEOHEAD	31:24	HEAD[29:22]							
		23:16	HEAD[21:14]							
		15:8	HEAD[13:6]							
		7:0	HEAD[5:0]							
0x0380	LCDC_HEOADDR	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x0384	LCDC_HEOCTRL	31:24								
		23:16								
		15:8								
		7:0			DONEIEN	ADDIEN	DSCRIEN	DMAIEN	LFETCH	DFETCH
0x0388	LCDC_HEONEXT	31:24	NEXT[31:24]							
		23:16	NEXT[23:16]							
		15:8	NEXT[15:8]							
		7:0	NEXT[7:0]							
0x038C	LCDC_HEOUHEAD	31:24	UHEAD[31:24]							
		23:16	UHEAD[23:16]							
		15:8	UHEAD[15:8]							
		7:0	UHEAD[7:0]							
0x0390	LCDC_HEOUADDR	31:24	UADDR[31:24]							
		23:16	UADDR[23:16]							
		15:8	UADDR[15:8]							
		7:0	UADDR[7:0]							
0x0394	LCDC_HEOUCTRL	31:24								
		23:16								
		15:8								
		7:0			UDONEIEN	UADDIEN	UDSCRIEN	UDMAIEN		UDFETCH
0x0398	LCDC_HEOUNEXT	31:24	UNEXT[31:24]							
		23:16	UNEXT[23:16]							
		15:8	UNEXT[15:8]							
		7:0	UNEXT[7:0]							
0x039C	LCDC_HEOVHEAD	31:24	VHEAD[31:24]							
		23:16	VHEAD[23:16]							
		15:8	VHEAD[15:8]							
		7:0	VHEAD[7:0]							
0x03A0	LCDC_HEOVADDR	31:24	VADDR[31:24]							
		23:16	VADDR[23:16]							
		15:8	VADDR[15:8]							
		7:0	VADDR[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x03A4	LCDC_HEOVCTRL	31:24								
		23:16								
		15:8								
		7:0			VDONEIEN	VADDIEN	VDSCRIEN	VDMAIEN		VDFETCH
0x03A8	LCDC_HEOVNEXT	31:24	VNEXT[31:24]							
		23:16	VNEXT[23:16]							
		15:8	VNEXT[15:8]							
		7:0	VNEXT[7:0]							
0x03AC	LCDC_HEOCFG0	31:24								
		23:16								
		15:8			LOCKDIS	ROTDIS				DLBO
		7:0	BLENUV[1:0]		BLEN[1:0]					
0x03B0	LCDC_HEOCFG1	31:24								
		23:16				DSCALEOPT			YUV422SWP	YUV422ROT
		15:8	YUVMODE[3:0]						CLUTMODE[1:0]	
		7:0	RGBMODE[3:0]						YUVEN	CLUTEN
0x03B4	LCDC_HEOCFG2	31:24						YPOS[10:8]		
		23:16	YPOS[7:0]							
		15:8						XPOS[10:8]		
		7:0	XPOS[7:0]							
0x03B8	LCDC_HEOCFG3	31:24						YSIZE[10:8]		
		23:16	YSIZE[7:0]							
		15:8						XSIZE[10:8]		
		7:0	XSIZE[7:0]							
0x03BC	LCDC_HEOCFG4	31:24						YMEMSIZE[10:8]		
		23:16	YMEMSIZE[7:0]							
		15:8						XMEMSIZE[10:8]		
		7:0	XMEMSIZE[7:0]							
0x03C0	LCDC_HEOCFG5	31:24	XSTRIDE[31:24]							
		23:16	XSTRIDE[23:16]							
		15:8	XSTRIDE[15:8]							
		7:0	XSTRIDE[7:0]							
0x03C4	LCDC_HEOCFG6	31:24	PSTRIDE[31:24]							
		23:16	PSTRIDE[23:16]							
		15:8	PSTRIDE[15:8]							
		7:0	PSTRIDE[7:0]							
0x03C8	LCDC_HEOCFG7	31:24	UVXSTRIDE[31:24]							
		23:16	UVXSTRIDE[23:16]							
		15:8	UVXSTRIDE[15:8]							
		7:0	UVXSTRIDE[7:0]							
0x03CC	LCDC_HEOCFG8	31:24	UVPSTRIDE[31:24]							
		23:16	UVPSTRIDE[23:16]							
		15:8	UVPSTRIDE[15:8]							
		7:0	UVPSTRIDE[7:0]							
0x03D0	LCDC_HEOCFG9	31:24								
		23:16	RDEF[7:0]							
		15:8	GDEF[7:0]							
		7:0	BDEF[7:0]							
0x03D4	LCDC_HEOCFG10	31:24								
		23:16	RKEY[7:0]							
		15:8	GKEY[7:0]							
		7:0	BKEY[7:0]							
0x03D8	LCDC_HEOCFG11	31:24								
		23:16	RMASK[7:0]							
		15:8	GMASK[7:0]							
		7:0	BMASK[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x03DC	LCDC_HEOCFG12	31:24								
		23:16	GA[7:0]							
		15:8				VIDPRI		DSTKEY	REP	DMA
		7:0	OVR	LAEN	GAEN	REVALPHA	ITER	ITER2BL	INV	CRKEY
0x03E0	LCDC_HEOCFG13	31:24	SCALEN		YFACTOR[13:8]					
		23:16	YFACTOR[7:0]							
		15:8			XFACTOR[13:8]					
		7:0	XFACTOR[7:0]							
0x03E4	LCDC_HEOCFG14	31:24		CSCYOFF	CSCRV[9:4]					
		23:16	CSCRV[3:0]				CSCRU[9:6]			
		15:8	CSCRU[5:0]						CSCRY[9:8]	
		7:0	CSCRY[7:0]							
0x03E8	LCDC_HEOCFG15	31:24		CSCUOFF	CSCGV[9:4]					
		23:16	CSCGV[3:0]				CSCGU[9:6]			
		15:8	CSCGU[5:0]						CSCGY[9:8]	
		7:0	CSCGY[7:0]							
0x03EC	LCDC_HEOCFG16	31:24		CSCVOFF	CSCBV[9:4]					
		23:16	CSCBV[3:0]				CSCBU[9:6]			
		15:8	CSCBU[5:0]						CSCBY[9:8]	
		7:0	CSCBY[7:0]							
0x03F0	LCDC_HEOCFG17	31:24	XPHI0COEFF3[7:0]							
		23:16	XPHI0COEFF2[7:0]							
		15:8	XPHI0COEFF1[7:0]							
		7:0	XPHI0COEFF0[7:0]							
0x03F4	LCDC_HEOCFG18	31:24								
		23:16								
		15:8								
		7:0	XPHI0COEFF4[7:0]							
0x03F8	LCDC_HEOCFG19	31:24	XPHI1COEFF3[7:0]							
		23:16	XPHI1COEFF2[7:0]							
		15:8	XPHI1COEFF1[7:0]							
		7:0	XPHI1COEFF0[7:0]							
0x03FC	LCDC_HEOCFG20	31:24								
		23:16								
		15:8								
		7:0	XPHI1COEFF4[7:0]							
0x0400	LCDC_HEOCFG21	31:24	XPHI2COEFF3[7:0]							
		23:16	XPHI2COEFF2[7:0]							
		15:8	XPHI2COEFF1[7:0]							
		7:0	XPHI2COEFF0[7:0]							
0x0404	LCDC_HEOCFG22	31:24								
		23:16								
		15:8								
		7:0	XPHI2COEFF4[7:0]							
0x0408	LCDC_HEOCFG23	31:24	XPHI3COEFF3[7:0]							
		23:16	XPHI3COEFF2[7:0]							
		15:8	XPHI3COEFF1[7:0]							
		7:0	XPHI3COEFF0[7:0]							
0x040C	LCDC_HEOCFG24	31:24								
		23:16								
		15:8								
		7:0	XPHI3COEFF4[7:0]							
0x0410	LCDC_HEOCFG25	31:24	XPHI4COEFF3[7:0]							
		23:16	XPHI4COEFF2[7:0]							
		15:8	XPHI4COEFF1[7:0]							
		7:0	XPHI4COEFF0[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x0414	LCDC_HEOCFG26	31:24								
		23:16								
		15:8								
		7:0	XPHI4COEFF4[7:0]							
0x0418	LCDC_HEOCFG27	31:24	XPHI5COEFF3[7:0]							
		23:16	XPHI5COEFF2[7:0]							
		15:8	XPHI5COEFF1[7:0]							
		7:0	XPHI5COEFF0[7:0]							
0x041C	LCDC_HEOCFG28	31:24								
		23:16								
		15:8								
		7:0	XPHI5COEFF4[7:0]							
0x0420	LCDC_HEOCFG29	31:24	XPHI6COEFF3[7:0]							
		23:16	XPHI6COEFF2[7:0]							
		15:8	XPHI6COEFF1[7:0]							
		7:0	XPHI6COEFF0[7:0]							
0x0424	LCDC_HEOCFG30	31:24								
		23:16								
		15:8								
		7:0	XPHI6COEFF4[7:0]							
0x0428	LCDC_HEOCFG31	31:24	XPHI7COEFF3[7:0]							
		23:16	XPHI7COEFF2[7:0]							
		15:8	XPHI7COEFF1[7:0]							
		7:0	XPHI7COEFF0[7:0]							
0x042C	LCDC_HEOCFG32	31:24								
		23:16								
		15:8								
		7:0	XPHI7COEFF4[7:0]							
0x0430	LCDC_HEOCFG33	31:24								
		23:16	YPHI0COEFF2[7:0]							
		15:8	YPHI0COEFF1[7:0]							
		7:0	YPHI0COEFF0[7:0]							
0x0434	LCDC_HEOCFG34	31:24								
		23:16	YPHI1COEFF2[7:0]							
		15:8	YPHI1COEFF1[7:0]							
		7:0	YPHI1COEFF0[7:0]							
0x0438	LCDC_HEOCFG35	31:24								
		23:16	YPHI2COEFF2[7:0]							
		15:8	YPHI2COEFF1[7:0]							
		7:0	YPHI2COEFF0[7:0]							
0x043C	LCDC_HEOCFG36	31:24								
		23:16	YPHI3COEFF2[7:0]							
		15:8	YPHI3COEFF1[7:0]							
		7:0	YPHI3COEFF0[7:0]							
0x0440	LCDC_HEOCFG37	31:24								
		23:16	YPHI4COEFF2[7:0]							
		15:8	YPHI4COEFF1[7:0]							
		7:0	YPHI4COEFF0[7:0]							
0x0444	LCDC_HEOCFG38	31:24								
		23:16	YPHI5COEFF2[7:0]							
		15:8	YPHI5COEFF1[7:0]							
		7:0	YPHI5COEFF0[7:0]							
0x0448	LCDC_HEOCFG39	31:24								
		23:16	YPHI6COEFF2[7:0]							
		15:8	YPHI6COEFF1[7:0]							
		7:0	YPHI6COEFF0[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x044C	LCDC_HEOCFG40	31:24								
		23:16	YPHI7COEFF2[7:0]							
		15:8	YPHI7COEFF1[7:0]							
		7:0	YPHI7COEFF0[7:0]							
0x0450	LCDC_HEOCFG41	31:24								
		23:16						YPHIDEF[2:0]		
		15:8								
		7:0						XPHIDEF[2:0]		
0x0454 ... 0x05FF	Reserved									
0x0600	LCDC_BASECLUT0	31:24								
		23:16	RCLUT[7:0]							
		15:8	GCLUT[7:0]							
		7:0	BCLUT[7:0]							
...										
0x09FC	LCDC_BASECLUT255	31:24								
		23:16	RCLUT[7:0]							
		15:8	GCLUT[7:0]							
		7:0	BCLUT[7:0]							
0x0A00	LCDC_OVR1CLUT0	31:24	ACLUT[7:0]							
		23:16	RCLUT[7:0]							
		15:8	GCLUT[7:0]							
		7:0	BCLUT[7:0]							
...										
0x0DFC	LCDC_OVR1CLUT255	31:24	ACLUT[7:0]							
		23:16	RCLUT[7:0]							
		15:8	GCLUT[7:0]							
		7:0	BCLUT[7:0]							
0x0E00	LCDC_OVR2CLUT0	31:24	ACLUT[7:0]							
		23:16	RCLUT[7:0]							
		15:8	GCLUT[7:0]							
		7:0	BCLUT[7:0]							
...										
0x11FC	LCDC_OVR2CLUT255	31:24	ACLUT[7:0]							
		23:16	RCLUT[7:0]							
		15:8	GCLUT[7:0]							
		7:0	BCLUT[7:0]							
0x1200	LCDC_HEOCLUT0	31:24	ACLUT[7:0]							
		23:16	RCLUT[7:0]							
		15:8	GCLUT[7:0]							
		7:0	BCLUT[7:0]							
...										
0x15FC	LCDC_HEOCLUT255	31:24	ACLUT[7:0]							
		23:16	RCLUT[7:0]							
		15:8	GCLUT[7:0]							
		7:0	BCLUT[7:0]							

Downloaded from Arrow.com.

37.7.1 LCD Controller Configuration Register 0

Name: LCDC_LCDCFG0
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CLKDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			CGDISPP		CGDISHEO	CGDISOVR2	CGDISOVR1	CGDISBASE
Access			R/W		R/W	R/W	R/W	R/W
Reset			0		0	0	0	0

Bit	7	6	5	4	3	2	1	0
					CLKPWMSEL			CLKPOL
Access					R/W			R/W
Reset					0			0

Bits 23:16 – CLKDIV[7:0] LCD Controller Clock Divider
8-bit width clock divider for pixel clock (LCDPCLK). The pixel clock period formula is:
LCDPCLK = source clock / (CLKDIV+2)
where source clock is the GCK clock.

Bit 13 – CGDISPP Clock Gating Disable Control for the Post Processing Layer

Value	Description
0	Automatic Clock Gating is enabled for the Post Processing Layer.
1	Clock is running continuously.

Bit 11 – CGDISHEO Clock Gating Disable Control for the High-End Overlay

Value	Description
0	Automatic Clock Gating is enabled for the High-End Overlay Layer.
1	Clock is running continuously.

Bit 10 – CGDISOVR2 Clock Gating Disable Control for the Overlay 2 Layer

Value	Description
0	Automatic Clock Gating is enabled for the Overlay 2 Layer.
1	Clock is running continuously.

Bit 9 – CGDISOVR1 Clock Gating Disable Control for the Overlay 1 Layer

Value	Description
0	Automatic Clock Gating is enabled for the Overlay 1 Layer.
1	Clock is running continuously.

Bit 8 – CGDISBASE Clock Gating Disable Control for the Base Layer

Value	Description
0	Automatic Clock Gating is enabled for the Base Layer.
1	Clock is running continuously.

Downloaded from Arrow.com.

Bit 3 – CLKPWMSEL LCD Controller PWM Clock Source Selection

Value	Description
0	The slow clock is selected and feeds the PWM module.
1	The system clock is selected and feeds the PWM module.

Bit 0 – CLKPOL LCD Controller Clock Polarity

Value	Description
0	Data/Control signals are launched on the rising edge of the pixel clock.
1	Data/Control signals are launched on the falling edge of the pixel clock.

Downloaded from Arrow.com.

37.7.2 LCD Controller Configuration Register 1

Name: LCDC_LCDCFG1
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							VSPW[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	VSPW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
							HSPW[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	HSPW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 25:16 – VSPW[9:0] Vertical Synchronization Pulse Width
Width of the LCDVSYNC pulse, given in number of lines. Width is (VSPW+1) lines.

Bits 9:0 – HSPW[9:0] Horizontal Synchronization Pulse Width
Width of the LCDHSYNC pulse, given in pixel clock cycles. Width is (HSPW+1) LCDPCLK cycles.

Downloaded from Arrow.com.

37.7.3 LCD Controller Configuration Register 2

Name: LCDC_LCDCFG2
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							VBPW[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	VBPW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
							VFPW[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	VFPW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 25:16 – VBPW[9:0] Vertical Back Porch Width
This field indicates the number of lines at the beginning of the Frame. The blanking interval is equal to VBPW lines.

Bits 9:0 – VFPW[9:0] Vertical Front Porch Width
This field indicates the number of lines at the end of the Frame. The blanking interval is equal to (VFPW+1) lines.

Downloaded from Arrow.com.

37.7.4 LCD Controller Configuration Register 3

Name: LCDC_LCDCFG3
Offset: 0x0C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							HBPW[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	HBPW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
							HFPW[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	HFPW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 25:16 – HBPW[9:0] Horizontal Back Porch Width
Number of pixel clock cycles inserted at the beginning of the line. The interval is equal to (HBPW+1) LCDPCLK cycles.

Bits 9:0 – HFPW[9:0] Horizontal Front Porch Width
Number of pixel clock cycles inserted at the end of the active line. The interval is equal to (HFPW+1) LCDPCLK cycles.

Downloaded from Arrow.com.

37.7.5 LCD Controller Configuration Register 4

Name: LCDC_LCDCFG4
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						RPF[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	RPF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						PPL[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	PPL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – RPF[10:0] Number of Active Row Per Frame
Number of active lines in the frame. The frame height is equal to (RPF+1) lines.

Bits 10:0 – PPL[10:0] Number of Pixels Per Line
Number of pixels in the frame. The number of active pixels in the frame is equal to (PPL+1) pixels.

Downloaded from Arrow.com.

37.7.6 LCD Controller Configuration Register 5

Name: LCDC_LCDCFG5
Offset: 0x14
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	GUARDTIME[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			VSPHO	VSPSU			MODE[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
	DISPDLY	DITHER		DISPPOL	VSPDLYE	VSPDLYS	VSPOL	HSPOL
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

Bits 23:16 – GUARDTIME[7:0] LCD DISPLAY Guard Time
Number of frames inserted during startup before LCDDISP assertion.
Number of frames inserted after LCDDISP reset.

Bit 13 – VSPHO LCD Controller Vertical synchronization Pulse Hold Configuration

Value	Description
0	The vertical synchronization pulse is asserted synchronously with horizontal pulse edge.
1	The vertical synchronization pulse is held active one pixel clock cycle after the horizontal pulse.

Bit 12 – VSPSU LCD Controller Vertical synchronization Pulse Setup Configuration

Value	Description
0	The vertical synchronization pulse is asserted synchronously with horizontal pulse edge.
1	The vertical synchronization pulse is asserted one pixel clock cycle before the horizontal pulse.

Bits 9:8 – MODE[1:0] LCD Controller Output Mode

Value	Name	Description
0	OUTPUT_12BPP	LCD Output mode is set to 12 bits per pixel
1	OUTPUT_16BPP	LCD Output mode is set to 16 bits per pixel
2	OUTPUT_18BPP	LCD Output mode is set to 18 bits per pixel
3	OUTPUT_24BPP	LCD Output mode is set to 24 bits per pixel

Bit 7 – DISPDLY LCD Controller Display Power Signal Synchronization

Value	Description
0	The LCDDISP signal is asserted synchronously with the second active edge of the horizontal pulse.
1	The LCDDISP signal is asserted asynchronously with both edges of the horizontal pulse.

Bit 6 – DITHER LCD Controller Dithering

Value	Description
0	Dithering logical unit is disabled

Downloaded from Arrow.com.

Value	Description
1	Dithering logical unit is activated

Bit 4 – DISPPOL Display Signal Polarity

Value	Description
0	Active High
1	Active Low

Bit 3 – VSPDLYE Vertical Synchronization Pulse End

Value	Description
0	The second active edge of the Vertical synchronization pulse is synchronous with the second edge of the horizontal pulse.
1	The second active edge of the Vertical synchronization pulse is synchronous with the first edge of the horizontal pulse.

Bit 2 – VSPDLYS Vertical Synchronization Pulse Start

Value	Description
0	The first active edge of the Vertical synchronization pulse is synchronous with the second edge of the horizontal pulse.
1	The first active edge of the Vertical synchronization pulse is synchronous with the first edge of the horizontal pulse.

Bit 1 – VSPOL Vertical Synchronization Pulse Polarity

Value	Description
0	Active High
1	Active Low

Bit 0 – HSPOL Horizontal Synchronization Pulse Polarity

Value	Description
0	Active High
1	Active Low

Downloaded from Arrow.com.

37.7.7 LCD Controller Configuration Register 6

Name: LCDC_LCDCFG6
Offset: 0x18
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PWMCVAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				PWMPOL		PWMPS[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bits 15:8 – PWMCVAL[7:0] LCD Controller PWM Compare Value
PWM compare value. Used to adjust the analog value obtained after an external filter to control the contrast of the display.

Bit 4 – PWMPOL LCD Controller PWM Signal Polarity
This bit defines the polarity of the PWM output signal.

Value	Description
0	The output pulses are low level.
1	The output pulses are high level (the output is high whenever the value in the counter is less than value CVAL).

Bits 2:0 – PWMPS[2:0] PWM Clock Prescaler
Selects the configuration of the counter prescaler module.

Value	Name	Description
000	DIV_1	The counter advances at a rate of $f_{COUNTER} = f_{PWM_SELECTED_CLOCK}$
001	DIV_2	The counter advances at a rate of $f_{COUNTER} = f_{PWM_SELECTED_CLOCK/2}$
010	DIV_4	The counter advances at a rate of $f_{COUNTER} = f_{PWM_SELECTED_CLOCK/4}$
011	DIV_8	The counter advances at a rate of $f_{COUNTER} = f_{PWM_SELECTED_CLOCK/8}$
100	DIV_16	The counter advances at a rate of $f_{COUNTER} = f_{PWM_SELECTED_CLOCK/16}$
101	DIV_32	The counter advances at a of rate $f_{COUNTER} = f_{PWM_SELECTED_CLOCK/32}$
110	DIV_64	The counter advances at a of rate $f_{COUNTER} = f_{PWM_SELECTED_CLOCK/64}$

Downloaded from Arrow.com.

37.7.8 LCD Controller Configuration Register 7

Name: LCDC_LCDCFG7
Offset: 0x1C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						ROW[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	ROW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 10:0 – ROW[10:0] Row Identifier For Row Interrupt Signal
When the LCD controller timing engine row pointer reaches the field ROW, an interrupt is triggered.
This field indicates a line in reverse order, i.e., ROW 0 is the last line and ROW height-1 the first line displayed.

Downloaded from Arrow.com.

37.7.9 LCD Controller Enable Register

Name: LCDC_LCDEN
Offset: 0x20
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					PWMEN	DISPEN	SYNCEN	CLKEN
Access					W	W	W	W
Reset					–	–	–	–

Bit 3 – PWMEN LCD Controller Pulse Width Modulation Enable

Value	Description
0	No effect
1	PWM is enabled.

Bit 2 – DISPEN LCD Controller DISP Signal Enable

Value	Description
0	No effect
1	LCDDISP signal is generated.

Bit 1 – SYNCEN LCD Controller Horizontal and Vertical Synchronization Enable

Value	Description
0	No effect
1	Both horizontal and vertical synchronization (LCDVSYNC and LCDHSYNC) signals are generated.

Bit 0 – CLKEN LCD Controller Pixel Clock Enable

Value	Description
0	No effect
1	Pixel clock logical unit is activated.

Downloaded from Arrow.com.

37.7.10 LCD Controller Disable Register

Name: LCDC_LCDDIS
Offset: 0x24
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					PWMRST	DISPRST	SYNCRST	CLKRST
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
					PWMDIS	DISPDIS	SYNCDIS	CLKDIS
Access					W	W	W	W
Reset					–	–	–	–

Bit 11 – PWMRST LCD Controller PWM Reset

Value	Description
0	No effect.
1	Resets the PWM module. The duty cycle may be violated.

Bit 10 – DISPRST LCD Controller DISP Signal Reset

Value	Description
0	No effect.
1	Resets the DISP signal.

Bit 9 – SYNCRST LCD Controller Horizontal and Vertical Synchronization Reset

Value	Description
0	No effect.
1	Resets the timing engine. The horizontal and vertical pulse widths are both violated.

Bit 8 – CLKRST LCD Controller Clock Reset

Value	Description
0	No effect.
1	Resets the pixel clock generator module. The pixel clock duty cycle may be violated.

Bit 3 – PWMDIS LCD Controller Pulse Width Modulation Disable

Value	Description
0	No effect.
1	Disables the pulse width modulation signal.

Bit 2 – DISPDIS LCD Controller DISP Signal Disable

Value	Description
0	No effect.

Downloaded from Arrow.com.

Value	Description
1	Disables the DISP signal.

Bit 1 – SYNCDIS LCD Controller Horizontal and Vertical Synchronization Disable

Value	Description
0	No effect.
1	Disables the synchronization signals after the end of the frame.

Bit 0 – CLKDIS LCD Controller Pixel Clock Disable

Value	Description
0	No effect.
1	Disables the pixel clock.

Downloaded from Arrow.com.

37.7.11 LCD Controller Status Register

Name: LCDC_LCDSR
Offset: 0x28
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				SIPSTS	PWMSTS	DISPSTS	LCDSTS	CLKSTS
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 4 – SIPSTS Synchronization In Progress

Value	Description
0	Clock domain synchronization is terminated.
1	Synchronization is in progress. Access to the registers LCDC_LCDCCFG[0..6], LCDC_LCDEN and LCDC_LCDDIS has no effect.

Bit 3 – PWMSTS LCD Controller PWM Signal Status

Value	Description
0	PWM is disabled.
1	PWM signal is activated.

Bit 2 – DISPSTS LCD Controller DISP Signal Status

Value	Description
0	DISP is disabled.
1	DISP signal is activated.

Bit 1 – LCDSTS LCD Controller Synchronization status

Value	Description
0	Timing engine is disabled.
1	Timing engine is running.

Bit 0 – CLKSTS Clock Status

Value	Description
0	Pixel clock is disabled.
1	Pixel clock is running.

Downloaded from Arrow.com.

37.7.12 LCD Controller Interrupt Enable Register

Name: LCDC_LCDIER
Offset: 0x2C
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			PPIE		HEOIE	OVR2IE	OVR1IE	BASEIE
Access			W		W	W	W	W
Reset			–		–	–	–	–

Bit	7	6	5	4	3	2	1	0
				FIFOERRIE	ROWIE	DISPIE	DISIE	SOFIE
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit 13 – PPIE Post Processing Interrupt Enable

Bit 11 – HEOIE High-End Overlay Interrupt Enable

Bit 10 – OVR2IE Overlay 2 Interrupt Enable

Bit 9 – OVR1IE Overlay 1 Interrupt Enable

Bit 8 – BASEIE Base Layer Interrupt Enable

Bit 4 – FIFOERRIE Output FIFO Error Interrupt Enable

Bit 3 – ROWIE Row Interrupt Enable

Bit 2 – DISPIE Powerup/Powerdown Sequence Terminated Interrupt Enable

Bit 1 – DISIE LCD Disable Interrupt Enable

Bit 0 – SOFIE Start of Frame Interrupt Enable

Downloaded from Arrow.com.

37.7.13 LCD Controller Interrupt Disable Register

Name: LCDC_LCDIDR
Offset: 0x30
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			PPID		HEOID	OVR2ID	OVR1ID	BASEID
Access			W		W	W	W	W
Reset			–		–	–	–	–

Bit	7	6	5	4	3	2	1	0
				FIFOERRID	ROWID	DISPID	DISID	SOFID
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit 13 – PPID Post Processing Interrupt Disable

Bit 11 – HEOID High-End Overlay Interrupt Disable

Bit 10 – OVR2ID Overlay 2 Interrupt Disable

Bit 9 – OVR1ID Overlay 1 Interrupt Disable

Bit 8 – BASEID Base Layer Interrupt Disable

Bit 4 – FIFOERRID Output FIFO Error Interrupt Disable

Bit 3 – ROWID Row Interrupt Disable

Bit 2 – DISPID Powerup/Powerdown Sequence Terminated Interrupt Disable

Bit 1 – DISID LCD Disable Interrupt Disable

Bit 0 – SOFID Start of Frame Interrupt Disable

Downloaded from Arrow.com.

37.7.14 LCD Controller Interrupt Mask Register

Name: LCDC_LCDIMR
Offset: 0x34
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:
0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			PPIM		HEOIM	OVR2IM	OVR1IM	BASEIM
Access			R		R	R	R	R
Reset			0		0	0	0	0

Bit	7	6	5	4	3	2	1	0
				FIFOERRIM	ROWIM	DISPIM	DISIM	SOFIM
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 13 – PPIM Post Processing Interrupt Mask

Bit 11 – HEOIM High-End Overlay Interrupt Mask

Bit 10 – OVR2IM Overlay 2 Interrupt Mask

Bit 9 – OVR1IM Overlay 1 Interrupt Mask

Bit 8 – BASEIM Base Layer Interrupt Mask

Bit 4 – FIFOERRIM Output FIFO Error Interrupt Mask

Bit 3 – ROWIM Row Interrupt Mask

Bit 2 – DISPIM Powerup/Powerdown Sequence Terminated Interrupt Mask

Bit 1 – DISIM LCD Disable Interrupt Mask

Bit 0 – SOFIM Start of Frame Interrupt Mask

Downloaded from Arrow.com.

37.7.15 LCD Controller Interrupt Status Register

Name: LCDC_LCDISR
Offset: 0x38
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			PP		HEO	OVR2	OVR1	BASE
Access			R		R	R	R	R
Reset			0		0	0	0	0

Bit	7	6	5	4	3	2	1	0
				FIFOERR	ROW	DISP	DIS	SOF
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 13 – PP Post Processing Raw Interrupt Status

Value	Description
0	No Post Processing interrupt detected since last read of LCDC_PPISR
1	Indicates that Post Processing interrupt is pending. This flag is reset as soon as the LCDC_PPISR is read.

Bit 11 – HEO High-End Overlay Raw Interrupt Status

Value	Description
0	No High-End layer interrupt detected since last read of LCDC_HEOISR.
1	Indicates that a High-End layer interrupt is pending. This flag is reset as soon as the LCDC_HEOISR is read.

Bit 10 – OVR2 Overlay 2 Raw Interrupt Status

Value	Description
0	No Overlay 2 layer interrupt detected since last read of LCDC_OVR2ISR.
1	Indicates that an Overlay 2 layer interrupt is pending. This flag is reset as soon as the LCDC_OVR2ISR is read.

Bit 9 – OVR1 Overlay 1 Raw Interrupt Status

Value	Description
0	No Overlay 1 layer interrupt detected since last read of LCDC_OVR1ISR.
1	Indicates that an Overlay 1 layer interrupt is pending. This flag is reset as soon as the LCDC_OVR1ISR is read.

Bit 8 – BASE Base Layer Raw Interrupt Status

Value	Description
0	No base layer interrupt detected since last read of LCDC_BASEISR.
1	Indicates that a base layer interrupt is pending. This flag is reset as soon as the LCDC_BASEISR is read.

Downloaded from Arrow.com.

Bit 4 – FIFOERR Output FIFO Error

Value	Description
0	No underflow has occurred in the output FIFO since last read of LCDC_LCDISR.
1	Indicates that an underflow has occurred in the output FIFO. This flag is reset after a read operation.

Bit 3 – ROW Row Interrupt Status

Value	Description
0	No detection since last read of the LCDC_LCDCISR.
1	Indicates that a row event has been detected. This flag is reset after a read operation.

Bit 2 – DISP Powerup/Powerdown Sequence Terminated Interrupt Status

Value	Description
0	Powerup sequence or powerdown sequence has not yet terminated.
1	Indicates the powerup sequence or powerdown sequence has terminated. This flag is reset after a read operation.

Bit 1 – DIS LCD Disable Interrupt Status

Value	Description
0	Horizontal and vertical timing generator has not yet been disabled.
1	Indicates that the horizontal and vertical timing generator has been disabled. This flag is reset after a read operation.

Bit 0 – SOF Start of Frame Interrupt Status

Value	Description
0	No detection since last read of LCDC_LCDISR.
1	Indicates that a start of frame event has been detected. This flag is reset after a read operation.

Downloaded from Arrow.com.

37.7.16 LCD Controller Attribute Register

Name: LCDC_ATTR
Offset: 0x3C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			PPA2Q		HEOA2Q	OVR2A2Q	OVR1A2Q	BASEA2Q
Access			W		W	W	W	W
Reset			–		–	–	–	–

Bit	7	6	5	4	3	2	1	0
			PP		HEO	OVR2	OVR1	BASE
Access			W		W	W	W	W
Reset			–		–	–	–	–

Bit 13 – PPA2Q Post-Processing Update Add To Queue

Value	Description
0	No effect.
1	Add the descriptor pointed to by the LCDC_PPHEAD register to the descriptor list.

Bit 11 – HEOA2Q High-End Overlay Update Add To Queue

Value	Description
0	No effect.
1	Add the descriptor pointed to by the LCDC_HEOHEAD register to the descriptor list.

Bit 10 – OVR2A2Q Overlay 2 Update Add to Queue

Value	Description
0	No effect.
1	Add the descriptor pointed to by the LCDC_OVR2HEAD register to the descriptor list.

Bit 9 – OVR1A2Q Overlay 1 Update Add To Queue

Value	Description
0	No effect.
1	Add the descriptor pointed to by the LCDC_OVR1HEAD register to the descriptor list.

Bit 8 – BASEA2Q Base Layer Update Add To Queue

Value	Description
0	No effect.
1	Add the descriptor pointed to by the LCDC_BASEHEAD register to the descriptor list.

Bit 5 – PP Post-Processing Update Attribute

Value	Description
0	No effect.

Downloaded from Arrow.com.

Value	Description
1	Update the PP window attribute.

Bit 3 – HEO High-End Overlay Update Attribute

Value	Description
0	No effect.
1	Update the HEO window attribute.

Bit 2 – OVR2 Overlay 2 Update Attribute

Value	Description
0	No effect.
1	Update the OVR2 window attribute.

Bit 1 – OVR1 Overlay 1 Update Attribute

Value	Description
0	No effect.
1	Update the OVR1 window attribute.

Bit 0 – BASE Base Layer Update Attribute

Value	Description
0	No effect.
1	Update the BASE window attributes.

Downloaded from Arrow.com.

37.7.17 LCD Controller QoS Configuration Register

Name: LCDC_QOSCFG
Offset: 0x40
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			QOS3CFG[1:0]				QOS2CFG[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
			QOS1CFG[1:0]					QOSLOCK
Access			R/W	R/W				R/W
Reset			0	0				0

Bits 13:12 – QOS3CFG[1:0] Quality Of Service for 2 to 3 Transition
This field indicates the FIFO threshold of the LCD that triggers a QoS change. The basic unit is 1/8 of the output FIFO depth.

Bits 9:8 – QOS2CFG[1:0] Quality Of Service for 1 to 2 Transition
This field indicates the FIFO threshold of the LCD that triggers a QoS change. The basic unit is 1/8 of the output FIFO depth.

Bits 5:4 – QOS1CFG[1:0] Quality Of Service for 0 to 1 Transition
This field indicates the FIFO threshold of the LCD that triggers a QoS change. The basic unit is 1/8 of the output FIFO depth.

Bit 0 – QOSLOCK Quality Of Service Lock

Value	Description
0	Quality of Service operation does not use the bus lock attribute to increase its memory bandwidth
1	Quality of Service operation uses the bus lock attribute to increase its memory bandwidth.

Downloaded from Arrow.com.

37.7.18 LCD Controller QoS 1 Metrics Register

Name: LCDC_QOS1M
Offset: 0x44
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					MET1[19:16]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MET1[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MET1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 19:0 – MET1[19:0] Metrics QoS 1
This field indicates the number of pixels sampled with QoS value set to 1. This field is updated on a per-frame basis.

Downloaded from Arrow.com.

37.7.19 LCD Controller QoS 2 Metrics Register

Name: LCDC_QOS2M
Offset: 0x48
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					MET1[19:16]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MET1[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MET1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 19:0 – MET1[19:0] Metrics QoS 2
This field indicates the number of pixels sampled with QoS value set to 2. This field is updated on a per-frame basis.

Downloaded from Arrow.com.

37.7.20 LCD Controller QoS 3 Metrics Register

Name: LCDC_QOS3M
Offset: 0x4C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					MET1[19:16]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MET1[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MET1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 19:0 – MET1[19:0] Metrics QoS 3
This field indicates the number of pixels sampled with QoS value set to 3. This field is updated on a per-frame basis.

Downloaded from Arrow.com.

37.7.21 LCD Controller QoS Min FIFO Level Register

Name: LCDC_QOSMIN
Offset: 0x50
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								LEVEL[8]
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
	LEVEL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 8:0 – LEVEL[8:0] Minimum FIFO Level
This field indicates the lowest FIFO level reached. This field is updated on a per-frame basis.

Downloaded from Arrow.com.

37.7.22 Base Layer Channel Enable Register

Name: LCDC_BASECHER
Offset: 0x00000060
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						A2QEN	UPDATEEN	CHEN
Access						W	W	W
Reset						–	–	–

Bit 2 – A2QEN Add To Queue Enable

Value	Description
0	No effect
1	Indicates that a valid descriptor has been written to memory, its memory location should be written to the DMA head pointer. The A2QSR status bit is set to one, and it is reset by hardware as soon as the descriptor pointed to by the DMA head pointer is added to the list.

Bit 1 – UPDATEEN Update Overlay Attributes Enable

Value	Description
0	No effect
1	Updates windows attributes on the next start of frame.

Bit 0 – CHEN Channel Enable

Value	Description
0	No effect
1	Enables the DMA channel

Downloaded from Arrow.com.

37.7.23 Base Layer Channel Disable Register

Name: LCDC_BASECHDR
Offset: 0x00000064
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								CHRST
Reset								W
								–

Bit	7	6	5	4	3	2	1	0
Access								CHDIS
Reset								W
								–

Bit 8 – CHRST Channel Reset

Value	Description
0	No effect
1	Resets the layer immediately. The frame is aborted.

Bit 0 – CHDIS Channel Disable

Value	Description
0	No effect
1	Disables the layer at the end of the current frame. The frame is completed.

Downloaded from Arrow.com.

37.7.24 Base Layer Channel Status Register

Name: LCDC_BASECHSR
Offset: 0x00000068
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						A2QSR	UPDATESR	CHSR
Access						R	R	R
Reset						0	0	0

Bit 2 – A2QSR Add To Queue Status

Value	Description
0	Add to queue not pending
1	Add to queue pending

Bit 1 – UPDATESR Update Overlay Attributes In Progress Status

Value	Description
0	No update pending
1	Overlay attributes will be updated on the next frame

Bit 0 – CHSR Channel Status

Value	Description
0	Layer disabled
1	Layer enabled

Downloaded from Arrow.com.

37.7.25 Base Layer Interrupt Enable Register

Name: LCDC_BASEIER
Offset: 0x0000006C
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		OVR	DONE	ADD	DSCR	DMA		
Access		W	W	W	W	W		
Reset		–	–	–	–	–		

Bit 6 – OVR Overflow Interrupt Enable

Bit 5 – DONE End of List Interrupt Enable

Bit 4 – ADD Head Descriptor Loaded Interrupt Enable

Bit 3 – DSCR Descriptor Loaded Interrupt Enable

Bit 2 – DMA End of DMA Transfer Interrupt Enable

Downloaded from Arrow.com.

37.7.26 Base Layer Interrupt Disable Register

Name: LCDC_BASEIDR
Offset: 0x00000070
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		OVR	DONE	ADD	DSCR	DMA		
Access		W	W	W	W	W		
Reset		–	–	–	–	–		

Bit 6 – OVR Overflow Interrupt Disable

Bit 5 – DONE End of List Interrupt Disable

Bit 4 – ADD Head Descriptor Loaded Interrupt Disable

Bit 3 – DSCR Descriptor Loaded Interrupt Disable

Bit 2 – DMA End of DMA Transfer Interrupt Disable

Downloaded from Arrow.com.

37.7.27 Base Layer Interrupt Mask Register

Name: LCDC_BASEIMR
Offset: 0x00000074
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:
0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		OVR	DONE	ADD	DSCR	DMA		
Access		R	R	R	R	R		
Reset		0	0	0	0	0		

Bit 6 – OVR Overflow Interrupt Mask

Bit 5 – DONE End of List Interrupt Mask

Bit 4 – ADD Head Descriptor Loaded Interrupt Mask

Bit 3 – DSCR Descriptor Loaded Interrupt Mask

Bit 2 – DMA End of DMA Transfer Interrupt Mask

Downloaded from Arrow.com.

37.7.28 Base Layer Interrupt Status Register

Name: LCDC_BASEISR
Offset: 0x00000078
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		OVR	DONE	ADD	DSCR	DMA		
Access		R	R	R	R	R		
Reset		0	0	0	0	0		

Bit 6 – OVR Overflow Detected

Value	Description
0	No overflow occurred since last read of LCDC_BASEISR
1	An overflow occurred. This flag is reset after a read operation.

Bit 5 – DONE End of List Detected

Value	Description
0	No End of List condition occurred since last read of LCDC_BASEISR
1	End of List condition has occurred. This flag is reset after a read operation.

Bit 4 – ADD Head Descriptor Loaded

Value	Description
0	No descriptor has been loaded since last read of LCDC_BASEISR
1	The descriptor pointed to by the LCDC_BASEHEAD register has been loaded successfully. This flag is reset after a read operation.

Bit 3 – DSCR DMA Descriptor Loaded

Value	Description
0	No descriptor has been loaded since last read of LCDC_BASEISR
1	A descriptor has been loaded successfully. This flag is reset after a read operation.

Bit 2 – DMA End of DMA Transfer

Value	Description
0	No end of DMA transfer has been detected since last read of LCDC_BASEISR
1	End of Transfer has been detected. This flag is reset after a read operation.

Downloaded from Arrow.com.

37.7.29 Base DMA Head Register

Name: LCDC_BASEHEAD
Offset: 0x0000007C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	HEAD[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HEAD[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HEAD[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	HEAD[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – HEAD[29:0] DMA Head Pointer
The Head Pointer points to a new descriptor.

Downloaded from Arrow.com.

37.7.30 Base DMA Address Register

Name: LCDC_BASEADDR
Offset: 0x00000080
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] DMA Transfer Start Address
Frame buffer base address

Downloaded from Arrow.com.

37.7.31 Base DMA Control Register

Name: LCDC_BASECTRL
Offset: 0x00000084
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			DONEIEN	ADDIEN	DSCRIEN	DMAIEN	LFETCH	DFETCH
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – DONEIEN End of List Interrupt Enable

Value	Description
0	End of list interrupt is disabled
1	End of list interrupt is enabled

Bit 4 – ADDIEN Add Head Descriptor to Queue Interrupt Enable

Value	Description
0	Transfer descriptor added to queue interrupt is enabled
1	Transfer descriptor added to queue interrupt is enabled

Bit 3 – DSCRIEN Descriptor Loaded Interrupt Enable

Value	Description
0	Transfer descriptor loaded interrupt is enabled
1	Transfer descriptor loaded interrupt is disabled

Bit 2 – DMAIEN End of DMA Transfer Interrupt Enable

Value	Description
0	DMA transfer completed interrupt is enabled
1	DMA transfer completed interrupt is disabled

Bit 1 – LFETCH Lookup Table Fetch Enable

Value	Description
0	Lookup Table DMA fetch is disabled
1	Lookup Table DMA fetch is enabled

Bit 0 – DFETCH Transfer Descriptor Fetch Enable

Value	Description
0	Transfer Descriptor fetch is disabled
1	Transfer Descriptor fetch is enabled

Downloaded from Arrow.com.

37.7.32 Base DMA Next Register

Name: LCDC_BASENEXT
Offset: 0x00000088
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	NEXT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NEXT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NEXT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NEXT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NEXT[31:0] DMA Descriptor Next Address
The transfer descriptor address must be aligned on a 64-bit boundary.

Downloaded from Arrow.com.

37.7.33 Base Layer Configuration Register 0

Name: LCDC_BASECFG0
Offset: 0x0000008C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								DLBO
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
			BLEN[1:0]					
Access			R/W	R/W				
Reset			0	0				

Bit 8 – DLBO Defined Length Burst Only For Channel Bus Transaction

Value	Description
0	Undefined length INCR burst is used for a burst of 2 and 3 beats.
1	Only defined length burst is used (SINGLE, INCR4, INCR8 and INCR16).

Bits 5:4 – BLEN[1:0] System Bus Burst Length

Value	Name	Description
0	AHB_SINGLE	System bus access is started as soon as there is enough space in the FIFO to store one data. SINGLE, INCR, INCR4, INCR8 and INCR16 bursts are used. INCR is used for a burst of 2 and 3 beats.
1	AHB_INCR4	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 4 data. A system bus INCR4 Burst is used. SINGLE, INCR and INCR4 bursts are used. INCR is used for a burst of 2 and 3 beats.
2	AHB_INCR8	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 8 data. A system bus INCR8 Burst is used. SINGLE, INCR, INCR4 and INCR8 bursts are used. INCR is used for a burst of 2 and 3 beats.
3	AHB_INCR16	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 16 data. A system bus INCR16 Burst is used. SINGLE, INCR, INCR4, INCR8 and INCR16 bursts are used. INCR is used for a burst of 2 and 3 beats.

Downloaded from Arrow.com.

37.7.34 Base Layer Configuration Register 1

Name: LCDC_BASECFG1
Offset: 0x00000090
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							CLUTMODE[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RGBMODE[3:0]							CLUTEN
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bits 9:8 – CLUTMODE[1:0] Color Lookup Table Mode Input Selection

Value	Name	Description
0	CLUT_1BPP	Color Lookup Table mode set to 1 bit per pixel
1	CLUT_2BPP	Color Lookup Table mode set to 2 bits per pixel
2	CLUT_4BPP	Color Lookup Table mode set to 4 bits per pixel
3	CLUT_8BPP	Color Lookup Table mode set to 8 bits per pixel

Bits 7:4 – RGBMODE[3:0] RGB Mode Input Selection

Value	Name	Description
0	12BPP_RGB_444	12 bpp RGB 444
1	16BPP_ARGB_4444	16 bpp ARGB 4444
2	16BPP_RGBA_4444	16 bpp RGBA 4444
3	16BPP_RGB_565	16 bpp RGB 565
4	16BPP_TRGB_1555	16 bpp TRGB 1555
5	18BPP_RGB_666	18 bpp RGB 666
6	18BPP_RGB_666PACKED	18 bpp RGB 666 PACKED
7	19BPP_TRGB_1666	19 bpp TRGB 1666
8	19BPP_TRGB_PACKED	19 bpp TRGB 1666 PACKED
9	24BPP_RGB_888	24 bpp RGB 888
10	24BPP_RGB_888_PACKED	24 bpp RGB 888 PACKED
11	25BPP_TRGB_1888	25 bpp TRGB 1888
12	32BPP_ARGB_8888	32 bpp ARGB 8888
13	32BPP_RGBA_8888	32 bpp RGBA 8888

Bit 0 – CLUTEN Color Lookup Table Mode Enable

Value	Description
0	RGB mode is selected.
1	Color Lookup Table mode is selected.

Downloaded from Arrow.com.

37.7.35 Base Layer Configuration Register 2

Name: LCDC_BASECFG2
Offset: 0x00000094
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	XSTRIDE[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	XSTRIDE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	XSTRIDE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	XSTRIDE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – XSTRIDE[31:0] Horizontal Stride
XSTRIDE represents the memory offset, in bytes, between two rows of the image memory.

Downloaded from Arrow.com.

37.7.36 Base Layer Configuration Register 3

Name: LCDC_BASECFG3
Offset: 0x00000098
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RDEF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GDEF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BDEF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – RDEF[7:0] Red Default
Default Red color when the Base DMA channel is disabled

Bits 15:8 – GDEF[7:0] Green Default
Default Green color when the Base DMA channel is disabled

Bits 7:0 – BDEF[7:0] Blue Default
Default Blue color when the Base DMA channel is disabled

Downloaded from Arrow.com.

37.7.37 Base Layer Configuration Register 4

Name: LCDC_BASECFG4
Offset: 0x0000009C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					DISCEN		REP	DMA
Access					R/W		R/W	R/W
Reset					0		0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 11 – DISCEN Discard Area Enable

Value	Description
0	The whole frame is retrieved from memory.
1	The DMA channel discards the area located at screen coordinate {DISCXPOS, DISCYPOS}.

Bit 9 – REP Use Replication logic to expand RGB color to 24 bits

Value	Description
0	When the selected pixel depth is less than 24 bpp the pixel is shifted and least significant bits are set to 0.
1	When the selected pixel depth is less than 24 bpp the pixel is shifted and the least significant bit replicates the msb.

Bit 8 – DMA Use DMA Data Path

Value	Description
0	The default color is used on the Base Layer.
1	The DMA channel retrieves the pixels stream from the memory.

Downloaded from Arrow.com.

37.7.38 Base Layer Configuration Register 5

Name: LCDC_BASECFG5
Offset: 0x000000A0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						DISCYPOS[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	DISCYPOS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						DISCXPOS[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	DISCXPOS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – DISCYPOS[10:0] Discard Area Vertical Coordinate
Vertical Position of the Discard Area

Bits 10:0 – DISCXPOS[10:0] Discard Area Horizontal Coordinate
Horizontal Position of the Discard Area

Downloaded from Arrow.com.

37.7.39 Base Layer Configuration Register 6

Name: LCDC_BASECFG6
Offset: 0x000000A4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						DISCYSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	DISCYSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						DISCXSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	DISCXSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – DISCYSIZE[10:0] Discard Area Vertical Size
Discard Vertical size in pixels. The Discard size is set to (DISCYSIZE + 1) pixels vertically.

Bits 10:0 – DISCXSIZE[10:0] Discard Area Horizontal Size
Discard Horizontal size in pixels. The Discard size is set to (DISCXSIZE + 1) pixels horizontally.

Downloaded from Arrow.com.

37.7.40 Overlay 1 Channel Enable Register

Name: LCDC_OVR1CHER
Offset: 0x00000160
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						A2QEN	UPDATEEN	CHEN
Access						W	W	W
Reset						–	–	–

Bit 2 – A2QEN Add To Queue Enable

Value	Description
0	No effect
1	Indicates that a valid descriptor has been written to memory, its memory location should be written to the DMA head pointer. The A2QSR status bit is set to one, and it is reset by hardware as soon as the descriptor pointed to by the DMA head pointer is added to the list.

Bit 1 – UPDATEEN Update Overlay Attributes Enable

Value	Description
0	No effect
1	Updates window attributes (size, alpha blending, etc.) on the next start of frame.

Bit 0 – CHEN Channel Enable

Value	Description
0	No effect
1	Enables the DMA channel

Downloaded from Arrow.com.

37.7.41 Overlay 1 Channel Disable Register

Name: LCDC_OVR1CHDR
Offset: 0x00000164
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								CHRST
Reset								W
								–

Bit	7	6	5	4	3	2	1	0
Access								CHDIS
Reset								W
								–

Bit 8 – CHRST Channel Reset

Value	Description
0	No effect
1	Resets the layer immediately. The frame is aborted.

Bit 0 – CHDIS Channel Disable

Value	Description
0	No effect
1	Disables the layer at the end of the current frame. The frame is completed.

Downloaded from Arrow.com.

37.7.42 Overlay 1 Channel Status Register

Name: LCDC_OVR1CHSR
Offset: 0x00000168
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						A2QSR	UPDATESR	CHSR
Access						R	R	R
Reset						0	0	0

Bit 2 – A2QSR Add To Queue Status

Value	Description
0	Add to queue not pending
1	Add to queue pending

Bit 1 – UPDATESR Update Overlay Attributes In Progress Status

Value	Description
0	No update pending
1	Overlay attributes will be updated on the next frame

Bit 0 – CHSR Channel Status

Value	Description
0	Layer disabled
1	Layer enabled

Downloaded from Arrow.com.

37.7.43 Overlay 1 Interrupt Enable Register

Name: LCDC_OVR1IER
Offset: 0x0000016C
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		OVR	DONE	ADD	DSCR	DMA		
Access		W	W	W	W	W		
Reset		–	–	–	–	–		

Bit 6 – OVR Overflow Interrupt Enable

Bit 5 – DONE End of List Interrupt Enable

Bit 4 – ADD Head Descriptor Loaded Interrupt Enable

Bit 3 – DSCR Descriptor Loaded Interrupt Enable

Bit 2 – DMA End of DMA Transfer Interrupt Enable

Downloaded from Arrow.com.

37.7.44 Overlay 1 Interrupt Disable Register

Name: LCDC_OVR1IDR
Offset: 0x00000170
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		OVR	DONE	ADD	DSCR	DMA		
Access		W	W	W	W	W		
Reset		–	–	–	–	–		

Bit 6 – OVR Overflow Interrupt Disable

Bit 5 – DONE End of List Interrupt Disable

Bit 4 – ADD Head Descriptor Loaded Interrupt Disable

Bit 3 – DSCR Descriptor Loaded Interrupt Disable

Bit 2 – DMA End of DMA Transfer Interrupt Disable

Downloaded from Arrow.com.

37.7.45 Overlay 1 Interrupt Mask Register

Name: LCDC_OVR1IMR
Offset: 0x00000174
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:
0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		OVR	DONE	ADD	DSCR	DMA		
Access		R	R	R	R	R		
Reset		0	0	0	0	0		

Bit 6 – OVR Overflow Interrupt Mask

Bit 5 – DONE End of List Interrupt Mask

Bit 4 – ADD Head Descriptor Loaded Interrupt Mask

Bit 3 – DSCR Descriptor Loaded Interrupt Mask

Bit 2 – DMA End of DMA Transfer Interrupt Mask

Downloaded from Arrow.com.

37.7.46 Overlay 1 Interrupt Status Register

Name: LCDC_OVR1ISR
Offset: 0x00000178
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		OVR	DONE	ADD	DSCR	DMA		
Access		R	R	R	R	R		
Reset		0	0	0	0	0		

Bit 6 – OVR Overflow Detected

Value	Description
0	No overflow occurred since last read of LCDC_OVR1ISR
1	An overflow occurred. This flag is reset after a read operation.

Bit 5 – DONE End of List Detected

Value	Description
0	No End of List condition has occurred since last read of LCDC_OVR1ISR
1	End of List condition has occurred. This flag is reset after a read operation.

Bit 4 – ADD Head Descriptor Loaded

Value	Description
0	No descriptor has been loaded since last read of LCDC_OVR1ISR
1	The descriptor pointed to by the LCDC_OVR1HEAD register has been loaded successfully. This flag is reset after a read operation.

Bit 3 – DSCR DMA Descriptor Loaded

Value	Description
0	No descriptor has been loaded since last read of LCDC_OVR1ISR
1	A descriptor has been loaded successfully. This flag is reset after a read operation.

Bit 2 – DMA End of DMA Transfer

Value	Description
0	No End of Transfer has been detected since last read of LCDC_OVR1ISR
1	End of Transfer has been detected. This flag is reset after a read operation.

Downloaded from Arrow.com.

37.7.47 Overlay 1 Head Register

Name: LCDC_OVR1HEAD
Offset: 0x0000017C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	HEAD[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HEAD[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HEAD[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	HEAD[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – HEAD[29:0] DMA Head Pointer
The Head Pointer points to a new descriptor.

Downloaded from Arrow.com.

37.7.48 Overlay 1 Address Register

Name: LCDC_OVR1ADDR
Offset: 0x00000180
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] DMA Transfer Overlay 1 Address
Overlay 1 frame buffer base address

Downloaded from Arrow.com.

37.7.49 Overlay 1 Control Register

Name: LCDC_OVR1CTRL
Offset: 0x00000184
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			DONEIEN	ADDIEN	DSCRIEN	DMAIEN	LFETCH	DFETCH
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – DONEIEN End of List Interrupt Enable

Value	Description
0	End of list interrupt is disabled
1	End of list interrupt is enabled

Bit 4 – ADDIEN Add Head Descriptor to Queue Interrupt Enable

Value	Description
0	Transfer descriptor added to queue interrupt is enabled
1	Transfer descriptor added to queue interrupt is enabled

Bit 3 – DSCRIEN Descriptor Loaded Interrupt Enable

Value	Description
0	Transfer descriptor loaded interrupt is enabled
1	Transfer descriptor loaded interrupt is disabled

Bit 2 – DMAIEN End of DMA Transfer Interrupt Enable

Value	Description
0	DMA transfer completed interrupt is enabled
1	DMA transfer completed interrupt is disabled

Bit 1 – LFETCH Lookup Table Fetch Enable

Value	Description
0	Lookup Table DMA fetch is disabled
1	Lookup Table DMA fetch is enabled

Bit 0 – DFETCH Transfer Descriptor Fetch Enable

Value	Description
0	Transfer Descriptor fetch is disabled
1	Transfer Descriptor fetch is enabled

Downloaded from Arrow.com.

37.7.50 Overlay 1 Next Register

Name: LCDC_OVR1NEXT
Offset: 0x00000188
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	NEXT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NEXT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NEXT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NEXT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NEXT[31:0] DMA Descriptor Next Address
The transfer descriptor address must be aligned on a 64-bit boundary.

Downloaded from Arrow.com.

37.7.51 Overlay 1 Configuration Register 0

Name: LCDC_OVR1CFG0
Offset: 0x0000018C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			LOCKDIS	ROTDIS				DLBO
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
			BLEN[1:0]					
Access			R/W	R/W				
Reset			0	0				

Bit 13 – LOCKDIS Hardware Rotation Lock Disable

Value	Description
0	System bus lock signal is asserted when a rotation is performed.
1	System bus lock signal is cleared when a rotation is performed.

Bit 12 – ROTDIS Hardware Rotation Optimization Disable

Value	Description
0	Rotation optimization is enabled.
1	Rotation optimization is disabled.

Bit 8 – DLBO Defined Length Burst Only for Channel Bus Transaction

Value	Description
0	Undefined length INCR burst is used for a burst of 2 and 3 beats.
1	Only defined length burst is used (SINGLE, INCR4, INCR8 and INCR16).

Bits 5:4 – BLEN[1:0] System Bus Burst Length

Value	Name	Description
0	AHB_BLEN_SINGLE	System bus access is started as soon as there is enough space in the FIFO to store one data. SINGLE, INCR, INCR4, INCR8 and INCR16 bursts are used. INCR is used for a burst of 2 and 3 beats.
1	AHB_BLEN_INCR4	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 4 data. A system bus INCR4 Burst is used. SINGLE, INCR and INCR4 bursts are used. INCR is used for a burst of 2 and 3 beats.
2	AHB_BLEN_INCR8	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 8 data. A system bus INCR8 Burst is used. SINGLE, INCR, INCR4 and INCR8 bursts are used. INCR is used for a burst of 2 and 3 beats.

Downloaded from Arrow.com.

Value	Name	Description
3	AHB_BLEN_INCR16	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 16 data. A system bus INCR16 Burst is used. SINGLE, INCR, INCR4, INCR8 and INCR16 bursts are used. INCR is used for a burst of 2 and 3 beats.

Downloaded from Arrow.com.

37.7.52 Overlay 1 Configuration Register 1

Name: LCDC_OVR1CFG1
Offset: 0x00000190
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							CLUTMODE[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RGBMODE[3:0]							CLUTEN
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bits 9:8 – CLUTMODE[1:0] Color Lookup Table Mode Input Selection

Value	Name	Description
0	CLUT_1BPP	Color Lookup Table mode set to 1 bit per pixel
1	CLUT_2BPP	Color Lookup Table mode set to 2 bits per pixel
2	CLUT_4BPP	Color Lookup Table mode set to 4 bits per pixel
3	CLUT_8BPP	Color Lookup Table mode set to 8 bits per pixel

Bits 7:4 – RGBMODE[3:0] RGB Mode Input Selection

Value	Name	Description
0	12BPP_RGB_444	12 bpp RGB 444
1	16BPP_ARGB_4444	16 bpp ARGB 4444
2	16BPP_RGBA_4444	16 bpp RGBA 4444
3	16BPP_RGB_565	16 bpp RGB 565
4	16BPP_TRGB_1555	16 bpp TRGB 1555
5	18BPP_RGB_666	18 bpp RGB 666
6	18BPP_RGB_666PACKED	18 bpp RGB 666 PACKED
7	19BPP_TRGB_1666	19 bpp TRGB 1666
8	19BPP_TRGB_PACKED	19 bpp TRGB 1666 PACKED
9	24BPP_RGB_888	24 bpp RGB 888
10	24BPP_RGB_888_PACKED	24 bpp RGB 888 PACKED
11	25BPP_TRGB_1888	25 bpp TRGB 1888
12	32BPP_ARGB_8888	32 bpp ARGB 8888
13	32BPP_RGBA_8888	32 bpp RGBA 8888

Bit 0 – CLUTEN Color Lookup Table Mode Enable

Value	Description
0	RGB mode is selected.
1	Color Lookup Table mode is selected.

Downloaded from Arrow.com.

37.7.53 Overlay 1 Configuration Register 2

Name: LCDC_OVR1CFG2
Offset: 0x00000194
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						YPOS[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	YPOS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						XPOS[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	XPOS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – YPOS[10:0] Vertical Window Position
Overlay 1 Vertical window position.

Bits 10:0 – XPOS[10:0] Horizontal Window Position
Overlay 1 Horizontal window position.

Downloaded from Arrow.com.

37.7.54 Overlay 1 Configuration Register 3

Name: LCDC_OVR1CFG3
Offset: 0x00000198
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						YSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	YSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						XSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	XSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – YSIZE[10:0] Vertical Window Size
Overlay 1 window height in pixels. The window height is set to (YSIZE + 1).
The following constraint must be met: YPOS + YSIZE ≤ RPF

Bits 10:0 – XSIZE[10:0] Horizontal Window Size
Overlay 1 window width in pixels. The window width is set to (XSIZE + 1).
The following constraint must be met: XPOS + XSIZE ≤ PPL

Downloaded from Arrow.com.

37.7.55 Overlay 1 Configuration Register 4

Name: LCDC_OVR1CFG4
Offset: 0x0000019C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	XSTRIDE[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	XSTRIDE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	XSTRIDE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	XSTRIDE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – XSTRIDE[31:0] Horizontal Stride
XSTRIDE represents the memory offset, in bytes, between two rows of the image memory.

Downloaded from Arrow.com.

37.7.56 Overlay 1 Configuration Register 5

Name: LCDC_OVR1CFG5
Offset: 0x000001A0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	PSTRIDE[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PSTRIDE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PSTRIDE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PSTRIDE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – PSTRIDE[31:0] Pixel Stride
PSTRIDE represents the memory offset, in bytes, between two pixels of the image.

Downloaded from Arrow.com.

37.7.57 Overlay 1 Configuration Register 6

Name: LCDC_OVR1CFG6
Offset: 0x000001A4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RDEF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GDEF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BDEF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – RDEF[7:0] Red Default
Default Red color when the Overlay 1 DMA channel is disabled.

Bits 15:8 – GDEF[7:0] Green Default
Default Green color when the Overlay 1 DMA channel is disabled.

Bits 7:0 – BDEF[7:0] Blue Default
Default Blue color when the Overlay 1 DMA channel is disabled.

Downloaded from Arrow.com.

37.7.58 Overlay 1 Configuration Register 7

Name: LCDC_OVR1CFG7
Offset: 0x000001A8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – RKEY[7:0] Red Color Component Chroma Key
Reference Red chroma key used to match the Red color of the current overlay.

Bits 15:8 – GKEY[7:0] Green Color Component Chroma Key
Reference Green chroma key used to match the Green color of the current overlay.

Bits 7:0 – BKEY[7:0] Blue Color Component Chroma Key
Reference Blue chroma key used to match the Blue color of the current overlay.

Downloaded from Arrow.com.

37.7.59 Overlay 1 Configuration Register 8

Name: LCDC_OVR1CFG8
Offset: 0x000001AC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RMASK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GMASK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BMASK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – RMASK[7:0] Red Color Component Chroma Key Mask
Red Mask used when the compare function is used. If a bit is set then this bit is compared.

Bits 15:8 – GMASK[7:0] Green Color Component Chroma Key Mask
Green Mask used when the compare function is used. If a bit is set then this bit is compared.

Bits 7:0 – BMASK[7:0] Blue Color Component Chroma Key Mask
Blue Mask used when the compare function is used. If a bit is set then this bit is compared.

Downloaded from Arrow.com.

37.7.60 Overlay 1 Configuration Register 9

Name: LCDC_OVR1CFG9
Offset: 0x000001B0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	GA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						DSTKEY	REP	DMA
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	OVR	LAEN	GAEN	REVALPHA	ITER	ITER2BL	INV	CRKEY
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – GA[7:0] Blender Global Alpha
Global alpha blender for the current layer.

Bit 10 – DSTKEY Destination Chroma Keying

Value	Description
0	Source Chroma keying is enabled.
1	Destination Chroma keying is used.

Bit 9 – REP Use Replication logic to expand RGB color to 24 bits

Value	Description
0	When the selected pixel depth is less than 24 bpp the pixel is shifted and least significant bits are set to 0.
1	When the selected pixel depth is less than 24 bpp the pixel is shifted and the least significant bit replicates the msb.

Bit 8 – DMA Blender DMA Layer Enable

Value	Description
0	The default color is used on the Overlay 1 Layer.
1	The DMA channel retrieves the pixels stream from the memory.

Bit 7 – OVR Blender Overlay Layer Enable

Value	Description
0	Overlay pixel color is set to the default overlay pixel color.
1	Overlay pixel color is set to the DMA channel pixel color.

Bit 6 – LAEN Blender Local Alpha Enable

Value	Description
0	Local alpha blending coefficient is disabled.
1	Local alpha blending coefficient is enabled.

Downloaded from Arrow.com.

Bit 5 – GAEN Blender Global Alpha Enable

Value	Description
0	Global alpha blending coefficient is disabled.
1	Global alpha blending coefficient is enabled.

Bit 4 – REVALPHA Blender Reverse Alpha

Value	Description
0	Pixel difference is multiplied by alpha.
1	Pixel difference is multiplied by 1 - alpha.

Bit 3 – ITER Blender Use Iterated Color

Value	Description
0	Pixel difference is set to 0.
1	Pixel difference is set to the iterated pixel value.

Bit 2 – ITER2BL Blender Iterated Color Enable

Value	Description
0	Final adder stage operand is set to 0.
1	Final adder stage operand is set to the iterated pixel value.

Bit 1 – INV Blender Inverted Blender Output Enable

Value	Description
0	Iterated pixel is the blended pixel.
1	Iterated pixel is the inverted pixel.

Bit 0 – CRKEY Blender Chroma Key Enable

Value	Description
0	Chroma key matching is disabled.
1	Chroma key matching is enabled.

Downloaded from Arrow.com.

37.7.61 Overlay 2 Channel Enable Register

Name: LCDC_OVR2CHER
Offset: 0x00000260
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						A2QEN	UPDATEEN	CHEN
Access						W	W	W
Reset						–	–	–

Bit 2 – A2QEN Add To Queue Enable

Value	Description
0	No effect
1	Indicates that a valid descriptor has been written to memory, its memory location should be written to the DMA head pointer. The A2QSR status bit is set to one, and it is reset by hardware as soon as the descriptor pointed to by the DMA head pointer is added to the list.

Bit 1 – UPDATEEN Update Overlay Attributes Enable

Value	Description
0	No effect
1	Updates windows attributes on the next start of frame.

Bit 0 – CHEN Channel Enable

Value	Description
0	No effect
1	Enables the DMA channel

Downloaded from Arrow.com.

37.7.62 Overlay 2 Channel Disable Register

Name: LCDC_OVR2CHDR
Offset: 0x00000264
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								CHRST
Reset								W
								–

Bit	7	6	5	4	3	2	1	0
Access								CHDIS
Reset								W
								–

Bit 8 – CHRST Channel Reset

Value	Description
0	No effect
1	Resets the layer immediately. The frame is aborted.

Bit 0 – CHDIS Channel Disable

Value	Description
0	No effect
1	Disables the layer at the end of the current frame. The frame is completed.

Downloaded from Arrow.com.

37.7.63 Overlay 2 Channel Status Register

Name: LCDC_OVR2CHSR
Offset: 0x00000268
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						A2QSR	UPDATESR	CHSR
Access						R	R	R
Reset						0	0	0

Bit 2 – A2QSR Add To Queue Status

Value	Description
0	Add to queue not pending
1	Add to queue pending

Bit 1 – UPDATESR Update Overlay Attributes In Progress Status

Value	Description
0	No update pending
1	Overlay attributes will be updated on the next frame

Bit 0 – CHSR Channel Status

Value	Description
0	Layer disabled
1	Layer enabled

Downloaded from Arrow.com.

37.7.64 Overlay 2 Interrupt Enable Register

Name: LCDC_OVR2IER
Offset: 0x0000026C
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		OVR	DONE	ADD	DSCR	DMA		
Access		W	W	W	W	W		
Reset		–	–	–	–	–		

Bit 6 – OVR Overflow Interrupt Enable

Bit 5 – DONE End of List Interrupt Enable

Bit 4 – ADD Head Descriptor Loaded Interrupt Enable

Bit 3 – DSCR Descriptor Loaded Interrupt Enable

Bit 2 – DMA End of DMA Transfer Interrupt Enable

Downloaded from Arrow.com.

37.7.65 Overlay 2 Interrupt Disable Register

Name: LCDC_OVR2IDR
Offset: 0x00000270
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		OVR	DONE	ADD	DSCR	DMA		
Access		W	W	W	W	W		
Reset		–	–	–	–	–		

Bit 6 – OVR Overflow Interrupt Disable

Bit 5 – DONE End of List Interrupt Disable

Bit 4 – ADD Head Descriptor Loaded Interrupt Disable

Bit 3 – DSCR Descriptor Loaded Interrupt Disable

Bit 2 – DMA End of DMA Transfer Interrupt Disable

Downloaded from Arrow.com.

37.7.66 Overlay 2 Interrupt Mask Register

Name: LCDC_OVR2IMR
Offset: 0x00000274
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:
0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		OVR	DONE	ADD	DSCR	DMA		
Access		R	R	R	R	R		
Reset		0	0	0	0	0		

Bit 6 – OVR Overflow Interrupt Mask

Bit 5 – DONE End of List Interrupt Mask

Bit 4 – ADD Head Descriptor Loaded Interrupt Mask

Bit 3 – DSCR Descriptor Loaded Interrupt Mask

Bit 2 – DMA End of DMA Transfer Interrupt Mask

Downloaded from Arrow.com.

37.7.67 Overlay 2 Interrupt Status Register

Name: LCDC_OVR2ISR
Offset: 0x00000278
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		OVR	DONE	ADD	DSCR	DMA		
Access		R	R	R	R	R		
Reset		0	0	0	0	0		

Bit 6 – OVR Overflow Detected

Value	Description
0	No overflow occurred since last read of LCDC_OVR2ISR
1	An overflow occurred. This flag is reset after a read operation.

Bit 5 – DONE End of List Detected

Value	Description
0	No End of List condition occurred since last read of LCDC_OVR2ISR
1	End of List condition has occurred. This flag is reset after a read operation.

Bit 4 – ADD Head Descriptor Loaded

Value	Description
0	No descriptor has been loaded since last read of LCDC_OVR2ISR
1	The descriptor pointed to by the LCDC_OVR2HEAD register has been loaded successfully. This flag is reset after a read operation.

Bit 3 – DSCR DMA Descriptor Loaded

Value	Description
0	No descriptor has been loaded since last read of LCDC_OVR2ISR
1	A descriptor has been loaded successfully. This flag is reset after a read operation.

Bit 2 – DMA End of DMA Transfer

Value	Description
0	No End of Transfer has been detected since last read of LCDC_OVR2ISR
1	End of Transfer has been detected. This flag is reset after a read operation.

Downloaded from Arrow.com.

37.7.68 Overlay 2 Head Register

Name: LCDC_OVR2HEAD
Offset: 0x0000027C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	HEAD[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HEAD[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HEAD[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	HEAD[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – HEAD[29:0] DMA Head Pointer
The Head Pointer points to a new descriptor.

Downloaded from Arrow.com.

37.7.69 Overlay 2 Address Register

Name: LCDC_OVR2ADDR
Offset: 0x00000280
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] DMA Transfer Overlay 2 Address
Overlay 2 frame buffer base address.

Downloaded from Arrow.com.

37.7.70 Overlay 2 Control Register

Name: LCDC_OVR2CTRL
Offset: 0x00000284
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			DONEIEN	ADDIEN	DSCRIEN	DMAIEN	LFETCH	DFETCH
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – DONEIEN End of List Interrupt Enable

Value	Description
0	End of list interrupt is disabled.
1	End of list interrupt is enabled.

Bit 4 – ADDIEN Add Head Descriptor to Queue Interrupt Enable

Value	Description
0	Transfer descriptor added to queue interrupt is enabled.
1	Transfer descriptor added to queue interrupt is enabled.

Bit 3 – DSCRIEN Descriptor Loaded Interrupt Enable

Value	Description
0	Transfer descriptor loaded interrupt is enabled.
1	Transfer descriptor loaded interrupt is disabled.

Bit 2 – DMAIEN End of DMA Transfer Interrupt Enable

Value	Description
0	DMA transfer completed interrupt is enabled.
1	DMA transfer completed interrupt is disabled.

Bit 1 – LFETCH Lookup Table Fetch Enable

Value	Description
0	Lookup Table DMA fetch is disabled.
1	Lookup Table DMA fetch is enabled.

Bit 0 – DFETCH Transfer Descriptor Fetch Enable

Value	Description
0	Transfer Descriptor fetch is disabled.
1	Transfer Descriptor fetch is enabled.

Downloaded from Arrow.com.

37.7.71 Overlay 2 Next Register

Name: LCDC_OVR2NEXT
Offset: 0x00000288
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	NEXT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NEXT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NEXT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NEXT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NEXT[31:0] DMA Descriptor Next Address
The transfer descriptor address must be aligned on a 64-bit boundary.

Downloaded from Arrow.com.

37.7.72 Overlay 2 Configuration Register 0

Name: LCDC_OVR2CFG0
Offset: 0x0000028C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			LOCKDIS	ROTDIS				DLBO
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
			BLEN[1:0]					
Access			R/W	R/W				
Reset			0	0				

Bit 13 – LOCKDIS Hardware Rotation Lock Disable

Value	Description
0	System bus lock signal is asserted when a rotation is performed.
1	System bus lock signal is cleared when a rotation is performed.

Bit 12 – ROTDIS Hardware Rotation Optimization Disable

Value	Description
0	Rotation optimization is enabled.
1	Rotation optimization is disabled.

Bit 8 – DLBO Defined Length Burst Only For Channel Bus Transaction

Value	Description
0	Undefined length INCR burst is used for 2 and 3 beats burst.
1	Only defined length burst is used (SINGLE, INCR4, INCR8 and INCR16).

Bits 5:4 – BLEN[1:0] System Bus Burst Length

Value	Name	Description
0	AHB_SINGLE	System bus access is started as soon as there is enough space in the FIFO to store one data. SINGLE, INCR, INCR4, INCR8 and INCR16 bursts are used. INCR is used for a burst of 2 and 3 beats.
1	AHB_INCR4	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 4 data. A system bus INCR4 Burst is used. SINGLE, INCR and INCR4 bursts are used. INCR is used for a burst of 2 and 3 beats.
2	AHB_INCR8	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 8 data. A system bus INCR8 Burst is used. SINGLE, INCR, INCR4 and INCR8 bursts are used. INCR is used for a burst of 2 and 3 beats.
3	AHB_INCR16	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 16 data. A system bus INCR16 Burst is used. SINGLE, INCR, INCR4, INCR8 and INCR16 bursts are used. INCR is used for a burst of 2 and 3 beats.

Downloaded from Arrow.com.

37.7.73 Overlay 2 Configuration Register 1

Name: LCDC_OVR2CFG1
Offset: 0x00000290
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							CLUTMODE[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RGBMODE[3:0]							CLUTEN
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bits 9:8 – CLUTMODE[1:0] Color Lookup Table Mode Input Selection

Value	Name	Description
0	CLUT_1BPP	Color Lookup Table mode set to 1 bit per pixel
1	CLUT_2BPP	Color Lookup Table mode set to 2 bits per pixel
2	CLUT_4BPP	Color Lookup Table mode set to 4 bits per pixel
3	CLUT_8BPP	Color Lookup Table mode set to 8 bits per pixel

Bits 7:4 – RGBMODE[3:0] RGB Mode Input Selection

Value	Name	Description
0	12BPP_RGB_444	12 bpp RGB 444
1	16BPP_ARGB_4444	16 bpp ARGB 4444
2	16BPP_RGBA_4444	16 bpp RGBA 4444
3	16BPP_RGB_565	16 bpp RGB 565
4	16BPP_TRGB_1555	16 bpp TRGB 1555
5	18BPP_RGB_666	18 bpp RGB 666
6	18BPP_RGB_666PACKED	18 bpp RGB 666 PACKED
7	19BPP_TRGB_1666	19 bpp TRGB 1666
8	19BPP_TRGB_PACKED	19 bpp TRGB 1666 PACKED
9	24BPP_RGB_888	24 bpp RGB 888
10	24BPP_RGB_888_PACKED	24 bpp RGB 888 PACKED
11	25BPP_TRGB_1888	25 bpp TRGB 1888
12	32BPP_ARGB_8888	32 bpp ARGB 8888
13	32BPP_RGBA_8888	32 bpp RGBA 8888

Bit 0 – CLUTEN Color Lookup Table Mode Enable

Value	Description
0	RGB mode is selected.
1	Color Lookup Table mode is selected.

Downloaded from Arrow.com.

37.7.74 Overlay 2 Configuration Register 2

Name: LCDC_OVR2CFG2
Offset: 0x00000294
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						YPOS[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	YPOS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						XPOS[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	XPOS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – YPOS[10:0] Vertical Window Position
Overlay 2 Vertical window position.

Bits 10:0 – XPOS[10:0] Horizontal Window Position
Overlay 2 Horizontal window position.

Downloaded from Arrow.com.

37.7.75 Overlay 2 Configuration Register 3

Name: LCDC_OVR2CFG3
Offset: 0x00000298
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						YSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	YSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						XSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	XSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – YSIZE[10:0] Vertical Window Size
Overlay 2 window height in pixels. The window height is set to (YSIZE + 1).
The following constraint must be met: YPOS + YSIZE ≤ RPF

Bits 10:0 – XSIZE[10:0] Horizontal Window Size
Overlay 2 window width in pixels. The window width is set to (XSIZE + 1).
The following constraint must be met: XPOS + XSIZE ≤ PPL

Downloaded from Arrow.com.

37.7.76 Overlay 2 Configuration Register 4

Name: LCDC_OVR2CFG4
Offset: 0x0000029C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	XSTRIDE[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	XSTRIDE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	XSTRIDE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	XSTRIDE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – XSTRIDE[31:0] Horizontal Stride
XSTRIDE represents the memory offset, in bytes, between two rows of the image memory.

Downloaded from Arrow.com.

37.7.77 Overlay 2 Configuration Register 5

Name: LCDC_OVR2CFG5
Offset: 0x000002A0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	PSTRIDE[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PSTRIDE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PSTRIDE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PSTRIDE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – PSTRIDE[31:0] Pixel Stride
PSTRIDE represents the memory offset, in bytes, between two pixels of the image memory.

Downloaded from Arrow.com.

37.7.78 Overlay 2 Configuration Register 6

Name: LCDC_OVR2CFG6
Offset: 0x000002A4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RDEF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GDEF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BDEF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – RDEF[7:0] Red Default
Default Red color when the Overlay 1 DMA channel is disabled.

Bits 15:8 – GDEF[7:0] Green Default
Default Green color when the Overlay 1 DMA channel is disabled.

Bits 7:0 – BDEF[7:0] Blue Default
Default Blue color when the Overlay 1 DMA channel is disabled.

Downloaded from Arrow.com.

37.7.79 Overlay 2 Configuration Register 7

Name: LCDC_OVR2CFG7
Offset: 0x000002A8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – RKEY[7:0] Red Color Component Chroma Key
Reference Red chroma key used to match the Red color of the current overlay.

Bits 15:8 – GKEY[7:0] Green Color Component Chroma Key
Reference Green chroma key used to match the Green color of the current overlay.

Bits 7:0 – BKEY[7:0] Blue Color Component Chroma Key
Reference Blue chroma key used to match the Blue color of the current overlay.

Downloaded from Arrow.com.

37.7.80 Overlay 2 Configuration Register 8

Name: LCDC_OVR2CFG8
Offset: 0x000002AC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RMASK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GMASK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BMASK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – RMASK[7:0] Red Color Component Chroma Key Mask
Red Mask used when the compare function is used. If a bit is set then this bit is compared.

Bits 15:8 – GMASK[7:0] Green Color Component Chroma Key Mask
Green Mask used when the compare function is used. If a bit is set then this bit is compared.

Bits 7:0 – BMASK[7:0] Blue Color Component Chroma Key Mask
Blue Mask used when the compare function is used. If a bit is set then this bit is compared.

Downloaded from Arrow.com.

37.7.81 Overlay 2 Configuration Register 9

Name: LCDC_OVR2CFG9
Offset: 0x000002B0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	GA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						DSTKEY	REP	DMA
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	OVR	LAEN	GAEN	REVALPHA	ITER	ITER2BL	INV	CRKEY
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – GA[7:0] Blender Global Alpha
Global alpha blender for the current layer.

Bit 10 – DSTKEY Destination Chroma Keying

Value	Description
0	Source Chroma keying is enabled.
1	Destination Chroma keying is used.

Bit 9 – REP Use Replication logic to expand RGB color to 24 bits

Value	Description
0	When the selected pixel depth is less than 24 bpp the pixel is shifted and least significant bits are set to 0.
1	When the selected pixel depth is less than 24 bpp the pixel is shifted and the least significant bit replicates the msb.

Bit 8 – DMA Blender DMA Layer Enable

Value	Description
0	The default color is used on the Overlay 1 Layer.
1	The DMA channel retrieves the pixels stream from the memory.

Bit 7 – OVR Blender Overlay Layer Enable

Value	Description
0	Overlay pixel color is set to the default overlay pixel color.
1	Overlay pixel color is set to the DMA channel pixel color.

Bit 6 – LAEN Blender Local Alpha Enable

Value	Description
0	Local alpha blending coefficient is disabled.
1	Local alpha blending coefficient is enabled.

Downloaded from Arrow.com.

Bit 5 – GAEN Blender Global Alpha Enable

Value	Description
0	Global alpha blending coefficient is disabled.
1	Global alpha blending coefficient is enabled.

Bit 4 – REVALPHA Blender Reverse Alpha

Value	Description
0	Pixel difference is multiplied by alpha.
1	Pixel difference is multiplied by 1 - alpha.

Bit 3 – ITER Blender Use Iterated Color

Value	Description
0	Pixel difference is set to 0.
1	Pixel difference is set to the iterated pixel value.

Bit 2 – ITER2BL Blender Iterated Color Enable

Value	Description
0	Final adder stage operand is set to 0.
1	Final adder stage operand is set to the iterated pixel value.

Bit 1 – INV Blender Inverted Blender Output Enable

Value	Description
0	Iterated pixel is the blended pixel.
1	Iterated pixel is the inverted pixel.

Bit 0 – CRKEY Blender Chroma Key Enable

Value	Description
0	Chroma key matching is disabled.
1	Chroma key matching is enabled.

Downloaded from Arrow.com.

37.7.82 High-End Overlay Channel Enable Register

Name: LCDC_HEOCHER
Offset: 0x00000360
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						A2QEN	UPDATEEN	CHEN
Access						W	W	W
Reset						–	–	–

Bit 2 – A2QEN Add To Queue Enable

Value	Description
0	No effect
1	Indicates that a valid descriptor has been written to memory, its memory location should be written to the DMA head pointer. The A2QSR status bit is set to one, and it is reset by hardware as soon as the descriptor pointed to by the DMA head pointer is added to the list.

Bit 1 – UPDATEEN Update Overlay Attributes Enable

Value	Description
0	No effect
1	Updates windows attributes on the next start of frame.

Bit 0 – CHEN Channel Enable

Value	Description
0	No effect
1	Enables the DMA channel

Downloaded from Arrow.com.

37.7.83 High-End Overlay Channel Disable Register

Name: LCDC_HEOCHDR
Offset: 0x00000364
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								CHRST
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								CHDIS
Access								W
Reset								–

Bit 8 – CHRST Channel Reset

Value	Description
0	No effect
1	Resets the layer immediately. The frame is aborted.

Bit 0 – CHDIS Channel Disable

Value	Description
0	No effect
1	Disables the layer at the end of the current frame. The frame is completed.

Downloaded from Arrow.com.

37.7.84 High-End Overlay Channel Status Register

Name: LCDC_HEOCHSR
Offset: 0x00000368
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						A2QSR	UPDATESR	CHSR
Access						R	R	R
Reset						0	0	0

Bit 2 – A2QSR Add To Queue Status

Value	Description
0	Add to queue not pending
1	Add to queue pending

Bit 1 – UPDATESR Update Overlay Attributes In Progress Status

Value	Description
0	No update pending
1	Overlay attributes will be updated on the next frame

Bit 0 – CHSR Channel Status

Value	Description
0	Layer disabled
1	Layer enabled

Downloaded from Arrow.com.

37.7.85 High-End Overlay Interrupt Enable Register

Name: LCDC_HEOIER
Offset: 0x0000036C
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		VOVR	VDONE	VADD	VDSCR	VDMA		
Access		W	W	W	W	W		
Reset		–	–	–	–	–		

Bit	15	14	13	12	11	10	9	8
		UOVR	UDONE	UADD	UDSCR	UDMA		
Access		W	W	W	W	W		
Reset		–	–	–	–	–		

Bit	7	6	5	4	3	2	1	0
		OVR	DONE	ADD	DSCR	DMA		
Access		W	W	W	W	W		
Reset		–	–	–	–	–		

Bit 22 – VOVR Overflow for V Chrominance Interrupt Enable

Bit 21 – VDONE End of List for V Chrominance Interrupt Enable

Bit 20 – VADD Head Descriptor Loaded for V Chrominance Interrupt Enable

Bit 19 – VDSCR Descriptor Loaded for V Chrominance Interrupt Enable

Bit 18 – VDMA End of DMA for V Chrominance Transfer Interrupt Enable

Bit 14 – UOVR Overflow for U or UV Chrominance Interrupt Enable

Bit 13 – UDONE End of List for U or UV Chrominance Interrupt Enable

Bit 12 – UADD Head Descriptor Loaded for U or UV Chrominance Interrupt Enable

Bit 11 – UDSCR Descriptor Loaded for U or UV Chrominance Interrupt Enable

Bit 10 – UDMA End of DMA Transfer for U or UV Chrominance Interrupt Enable

Bit 6 – OVR Overflow Interrupt Enable

Bit 5 – DONE End of List Interrupt Enable

Bit 4 – ADD Head Descriptor Loaded Interrupt Enable

Downloaded from Arrow.com.

Bit 3 – DSCR Descriptor Loaded Interrupt Enable

Bit 2 – DMA End of DMA Transfer Interrupt Enable

Downloaded from Arrow.com.

37.7.86 High-End Overlay Interrupt Disable Register

Name: LCDC_HEOIDR
Offset: 0x00000370
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		VOVR	VDONE	VADD	VDSCR	VDMA		
Access		W	W	W	W	W		
Reset		–	–	–	–	–		

Bit	15	14	13	12	11	10	9	8
		UOVR	UDONE	UADD	UDSCR	UDMA		
Access		W	W	W	W	W		
Reset		–	–	–	–	–		

Bit	7	6	5	4	3	2	1	0
		OVR	DONE	ADD	DSCR	DMA		
Access		W	W	W	W	W		
Reset		–	–	–	–	–		

Bit 22 – VOVR Overflow for V Chrominance Component Interrupt Disable

Bit 21 – VDONE End of List for V Chrominance Component Interrupt Disable

Bit 20 – VADD Head Descriptor Loaded for V Chrominance Component Interrupt Disable

Bit 19 – VDSCR Descriptor Loaded for V Chrominance Component Interrupt Disable

Bit 18 – VDMA End of DMA Transfer for V Chrominance Component Interrupt Disable

Bit 14 – UOVR Overflow Interrupt for U or UV Chrominance Component Disable

Bit 13 – UDONE End of List Interrupt for U or UV Chrominance Component Disable

Bit 12 – UADD Head Descriptor Loaded for U or UV Chrominance Component Interrupt Disable

Bit 11 – UDSCR Descriptor Loaded for U or UV Chrominance Component Interrupt Disable

Bit 10 – UDMA End of DMA Transfer for U or UV Chrominance Component Interrupt Disable

Bit 6 – OVR Overflow Interrupt Disable

Bit 5 – DONE End of List Interrupt Disable

Bit 4 – ADD Head Descriptor Loaded Interrupt Disable

Downloaded from Arrow.com.

Bit 3 – DSCR Descriptor Loaded Interrupt Disable

Bit 2 – DMA End of DMA Transfer Interrupt Disable

Downloaded from Arrow.com.

37.7.87 High-End Overlay Interrupt Mask Register

Name: LCDC_HEOIMR
Offset: 0x00000374
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:
0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		VOVR	VDONE	VADD	VDSCR	VDMA		
Access		R	R	R	R	R		
Reset		0	0	0	0	0		

Bit	15	14	13	12	11	10	9	8
		UOVR	UDONE	UADD	UDSCR	UDMA		
Access		R	R	R	R	R		
Reset		0	0	0	0	0		

Bit	7	6	5	4	3	2	1	0
		OVR	DONE	ADD	DSCR	DMA		
Access		R	R	R	R	R		
Reset		0	0	0	0	0		

Bit 22 – VOVR Overflow for V Chrominance Interrupt Mask

Bit 21 – VDONE End of List for V Chrominance Component Mask

Bit 20 – VADD Head Descriptor Loaded for V Chrominance Component Mask

Bit 19 – VDSCR Descriptor Loaded for V Chrominance Component Interrupt Mask

Bit 18 – VDMA End of DMA Transfer for V Chrominance Component Interrupt Mask

Bit 14 – UOVR Overflow for U Chrominance Interrupt Mask

Bit 13 – UDONE End of List for U or UV Chrominance Component Mask

Bit 12 – UADD Head Descriptor Loaded for U or UV Chrominance Component Mask

Bit 11 – UDSCR Descriptor Loaded for U or UV Chrominance Component Interrupt Mask

Bit 10 – UDMA End of DMA Transfer for U or UV Chrominance Component Interrupt Mask

Bit 6 – OVR Overflow Interrupt Mask

Bit 5 – DONE End of List Interrupt Mask

Bit 4 – ADD Head Descriptor Loaded Interrupt Mask

Downloaded from Arrow.com.

Bit 3 – DSCR Descriptor Loaded Interrupt Mask

Bit 2 – DMA End of DMA Transfer Interrupt Mask

Downloaded from Arrow.com.

37.7.88 High-End Overlay Interrupt Status Register

Name: LCDC_HEOISR
Offset: 0x00000378
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		VOVR	VDONE	VADD	VDSCR	VDMA		
Access		R	R	R	R	R		
Reset		0	0	0	0	0		

Bit	15	14	13	12	11	10	9	8
		UOVR	UDONE	UADD	UDSCR	UDMA		
Access		R	R	R	R	R		
Reset		0	0	0	0	0		

Bit	7	6	5	4	3	2	1	0
		OVR	DONE	ADD	DSCR	DMA		
Access		R	R	R	R	R		
Reset		0	0	0	0	0		

Bit 22 – VOVR Overflow Detected for V Component

Value	Description
0	No overflow occurred since last read of LCDC_HEOISR
1	An overflow occurred. This flag is reset after a read operation.

Bit 21 – VDONE End of List Detected for V Component

Value	Description
0	No End of List condition occurred since last read of LCDC_HEOISR
1	End of List condition has occurred. This flag is reset after a read operation.

Bit 20 – VADD Head Descriptor Loaded for V Component

Value	Description
0	No descriptor has been loaded since last read of LCDC_HEOISR
1	The descriptor pointed to by the LCDC_HEOVHEAD register has been loaded successfully. This flag is reset after a read operation.

Bit 19 – VDSCR DMA Descriptor Loaded for V Component

Value	Description
0	No descriptor has been loaded since last read of LCDC_HEOISR
1	A descriptor has been loaded successfully. This flag is reset after a read operation.

Bit 18 – VDMA End of DMA Transfer for V Component

Value	Description
0	No End of Transfer has been detected since last read of LCDC_HEOISR
1	End of Transfer has been detected. This flag is reset after a read operation.

Bit 14 – UOVR Overflow Detected for U Component

Value	Description
0	No overflow occurred since last read of LCDC_HEOISR

Downloaded from Arrow.com.

Value	Description
1	An overflow occurred. This flag is reset after a read operation.

Bit 13 – UDONE End of List Detected for U Component

Value	Description
0	No End of List condition occurred since last read of LCDC_HEOISR
1	End of List condition has occurred. This flag is reset after a read operation.

Bit 12 – UADD Head Descriptor Loaded for U Component

Value	Description
0	No descriptor has been loaded since last read of LCDC_HEOISR
1	The descriptor pointed to by the LCDC_HEOUHEAD register has been loaded successfully. This flag is reset after a read operation.

Bit 11 – UDSCR DMA Descriptor Loaded for U Component

Value	Description
0	No descriptor has been loaded since last read of LCDC_HEOISR
1	A descriptor has been loaded successfully. This flag is reset after a read operation.

Bit 10 – UDMA End of DMA Transfer for U Component

Value	Description
0	No End of Transfer has been detected since last read of LCDC_HEOISR
1	End of Transfer has been detected. This flag is reset after a read operation.

Bit 6 – OVR Overflow Detected

Value	Description
0	No overflow occurred since last read of LCDC_HEOISR
1	An overflow occurred. This flag is reset after a read operation.

Bit 5 – DONE End of List Detected

Value	Description
0	No End of List condition occurred since last read of LCDC_HEOISR
1	End of List condition has occurred. This flag is reset after a read operation.

Bit 4 – ADD Head Descriptor Loaded

Value	Description
0	No descriptor has been loaded since last read of LCDC_HEOISR
1	The descriptor pointed to by the LCDC_HEOHEAD register has been loaded successfully. This flag is reset after a read operation.

Bit 3 – DSCR DMA Descriptor Loaded

Value	Description
0	No descriptor has been loaded since last read of LCDC_HEOISR
1	A descriptor has been loaded successfully. This flag is reset after a read operation.

Bit 2 – DMA End of DMA Transfer

Value	Description
0	No end of transfer has been detected since last read of LCDC_HEOISR
1	End of Transfer has been detected. This flag is reset after a read operation.

Downloaded from Arrow.com.

37.7.89 High-End Overlay DMA Head Register

Name: LCDC_HEOHEAD
Offset: 0x0000037C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	HEAD[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HEAD[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HEAD[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	HEAD[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – HEAD[29:0] DMA Head Pointer
The Head Pointer points to a new descriptor.

Downloaded from Arrow.com.

37.7.90 High-End Overlay DMA Address Register

Name: LCDC_HEOADDR
Offset: 0x00000380
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] DMA Transfer Start Address
Frame Buffer Base Address.

Downloaded from Arrow.com.

37.7.91 High-End Overlay DMA Control Register

Name: LCDC_HEOCTRL
Offset: 0x00000384
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			DONEIEN	ADDIEN	DSCRIEN	DMAIEN	LFETCH	DFETCH
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – DONEIEN End of List Interrupt Enable

Value	Description
0	End of list interrupt is disabled.
1	End of list interrupt is enabled.

Bit 4 – ADDIEN Add Head Descriptor to Queue Interrupt Enable

Value	Description
0	Transfer descriptor added to queue interrupt is enabled.
1	Transfer descriptor added to queue interrupt is enabled.

Bit 3 – DSCRIEN Descriptor Loaded Interrupt Enable

Value	Description
0	Transfer descriptor loaded interrupt is enabled.
1	Transfer descriptor loaded interrupt is disabled.

Bit 2 – DMAIEN End of DMA Transfer Interrupt Enable

Value	Description
0	DMA transfer completed interrupt is enabled.
1	DMA transfer completed interrupt is disabled.

Bit 1 – LFETCH Lookup Table Fetch Enable

Value	Description
0	Lookup Table DMA fetch is disabled.
1	Lookup Table DMA fetch is enabled.

Bit 0 – DFETCH Transfer Descriptor Fetch Enable

Value	Description
0	Transfer Descriptor fetch is disabled.
1	Transfer Descriptor fetch is enabled.

Downloaded from Arrow.com.

37.7.92 High-End Overlay DMA Next Register

Name: LCDC_HEONEXT
Offset: 0x00000388
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	NEXT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NEXT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NEXT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NEXT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NEXT[31:0] DMA Descriptor Next Address
The transfer descriptor address must be aligned on a 64-bit boundary.

Downloaded from Arrow.com.

37.7.93 High-End Overlay U-UV DMA Head Register

Name: LCDC_HEOUHEAD
Offset: 0x0000038C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	UHEAD[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	UHEAD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UHEAD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UHEAD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – UHEAD[31:0] DMA Head Pointer
The Head Pointer points to a new descriptor.

Downloaded from Arrow.com.

37.7.94 High-End Overlay U-UV DMA Address Register

Name: LCDC_HEOUADDR
Offset: 0x00000390
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	UADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	UADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – UADDR[31:0] DMA Transfer Start Address for U or UV Chrominance
U or UV frame buffer address.

Downloaded from Arrow.com.

37.7.95 High-End Overlay U-UV DMA Control Register

Name: LCDC_HEOUCTRL
Offset: 0x00000394
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			UDONEIEN	UADDIEN	UDSCRIEN	UDMAIEN		UDFETCH
Access			R/W	R/W	R/W	R/W		R/W
Reset			0	0	0	0		0

Bit 5 – UDONEIEN End of List Interrupt Enable

Value	Description
0	End of list interrupt is disabled.
1	End of list interrupt is enabled.

Bit 4 – UADDIEN Add Head Descriptor to Queue Interrupt Enable

Value	Description
0	Transfer descriptor added to queue interrupt is enabled.
1	Transfer descriptor added to queue interrupt is enabled.

Bit 3 – UDSCRIEN Descriptor Loaded Interrupt Enable

Value	Description
0	Transfer descriptor loaded interrupt is enabled.
1	Transfer descriptor loaded interrupt is disabled.

Bit 2 – UDMAIEN End of DMA Transfer Interrupt Enable

Value	Description
0	DMA transfer completed interrupt is enabled.
1	DMA transfer completed interrupt is disabled.

Bit 0 – UDFETCH Transfer Descriptor Fetch Enable

Value	Description
0	Transfer Descriptor fetch is disabled.
1	Transfer Descriptor fetch is enabled.

Downloaded from Arrow.com.

37.7.96 High-End Overlay U-UV DMA Next Register

Name: LCDC_HEOUNEXT
Offset: 0x00000398
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	UNEXT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	UNEXT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UNEXT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UNEXT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – UNEXT[31:0] DMA Descriptor Next Address
The transfer descriptor address must be aligned on a 64-bit boundary.

Downloaded from Arrow.com.

37.7.97 High-End Overlay V DMA Head Register

Name: LCDC_HEOVHEAD
Offset: 0x0000039C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	VHEAD[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	VHEAD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	VHEAD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VHEAD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – VHEAD[31:0] DMA Head Pointer
The Head Pointer points to a new descriptor.

Downloaded from Arrow.com.

37.7.98 High-End Overlay V DMA Address Register

Name: LCDC_HEOVADDR
Offset: 0x000003A0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	VADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	VADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	VADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – VADDR[31:0] DMA Transfer Start Address for V Chrominance Frame Buffer Base Address.

Downloaded from Arrow.com.

37.7.99 High-End Overlay V DMA Control Register

Name: LCDC_HEOVCTRL
Offset: 0x000003A4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			VDONEIEN	VADDIEN	VDSCRIEN	VDMAIEN		VDFETCH
Access			R/W	R/W	R/W	R/W		R/W
Reset			0	0	0	0		0

Bit 5 – VDONEIEN End of List Interrupt Enable

Value	Description
0	End of list interrupt is disabled.
1	End of list interrupt is enabled.

Bit 4 – VADDIEN Add Head Descriptor to Queue Interrupt Enable

Value	Description
0	Transfer descriptor added to queue interrupt is enabled.
1	Transfer descriptor added to queue interrupt is enabled.

Bit 3 – VDSCRIEN Descriptor Loaded Interrupt Enable

Value	Description
0	Transfer descriptor loaded interrupt is enabled.
1	Transfer descriptor loaded interrupt is disabled.

Bit 2 – VDMAIEN End of DMA Transfer Interrupt Enable

Value	Description
0	DMA transfer completed interrupt is enabled.
1	DMA transfer completed interrupt is disabled.

Bit 0 – VDFETCH Transfer Descriptor Fetch Enable

Value	Description
0	Transfer Descriptor fetch is disabled.
1	Transfer Descriptor fetch is enabled.

Downloaded from Arrow.com.

37.7.100 High-End Overlay V DMA Next Register

Name: LCDC_HEOVNEXT
Offset: 0x000003A8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	VNEXT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	VNEXT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	VNEXT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VNEXT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – VNEXT[31:0] DMA Descriptor Next Address
The transfer descriptor address must be aligned on a 64-bit boundary.

Downloaded from Arrow.com.

37.7.101 High-End Overlay Configuration Register 0

Name: LCDC_HEOCFG0
Offset: 0x000003AC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			LOCKDIS	ROTDIS				DLBO
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
	BLENUV[1:0]		BLEN[1:0]					
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit 13 – LOCKDIS Hardware Rotation Lock Disable

Value	Description
0	System bus lock signal is asserted when a rotation is performed.
1	System bus lock signal is cleared when a rotation is performed.

Bit 12 – ROTDIS Hardware Rotation Optimization Disable

Value	Description
0	Rotation optimization is enabled.
1	Rotation optimization is disabled.

Bit 8 – DLBO Defined Length Burst Only For Channel Bus Transaction

Value	Description
0	Undefined length INCR burst is used for a burst of 2 and 3 beats.
1	Only defined length burst is used (SINGLE, INCR4, INCR8 and INCR16).

Bits 7:6 – BLENUV[1:0] System Bus Burst Length for U-V Channel

Value	Name	Description
0	AHB_SINGLE	System bus access is started as soon as there is enough space in the FIFO to store one data. SINGLE, INCR, INCR4, INCR8 and INCR16 bursts are used. INCR is used for a burst of 2 and 3 beats.
1	AHB_INCR4	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 4 data. A system bus INCR4 Burst is used. SINGLE, INCR and INCR4 bursts are used. INCR is used for a burst of 2 and 3 beats.
2	AHB_INCR8	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 8 data. A system bus INCR8 Burst is used. SINGLE, INCR, INCR4 and INCR8 bursts are used. INCR is used for a burst of 2 and 3 beats.
3	AHB_INCR16	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 16 data. A system bus INCR16 Burst is used. SINGLE, INCR, INCR4, INCR8 and INCR16 bursts are used. INCR is used for a burst of 2 and 3 beats.

Downloaded from Arrow.com.

Bits 5:4 – BLEN[1:0] System Bus Burst Length

Value	Name	Description
0	AHB_BLEN_SINGLE	System bus access is started as soon as there is enough space in the FIFO to store one data. SINGLE, INCR, INCR4, INCR8 and INCR16 bursts are used. INCR is used for a burst of 2 and 3 beats.
1	AHB_BLEN_INCR4	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 4 data. A system bus INCR4 Burst is used. SINGLE, INCR and INCR4 bursts are used. INCR is used for a burst of 2 and 3 beats.
2	AHB_BLEN_INCR8	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 8 data. A system bus INCR8 Burst is used. SINGLE, INCR, INCR4 and INCR8 bursts are used. INCR is used for a burst of 2 and 3 beats.
3	AHB_BLEN_INCR16	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 16 data. A system bus INCR16 Burst is used. SINGLE, INCR, INCR4, INCR8 and INCR16 bursts are used. INCR is used for a burst of 2 and 3 beats.

Downloaded from Arrow.com.

37.7.102 High-End Overlay Configuration Register 1

Name: LCDC_HEOCFG1
Offset: 0x000003B0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
				DSCALEOPT			YUV422SWP	YUV422ROT
Access				R/W			R/W	R/W
Reset				0			0	0

Bit	15	14	13	12	11	10	9	8
	YUVMODE[3:0]						CLUTMODE[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	RGBMODE[3:0]						YUVEN	CLUTEN
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit 20 – DSCALEOPT Down Scaling Bandwidth Optimization

Value	Description
0	Scaler Optimization is disabled.
1	Scaler Optimization is enabled; only relevant pixels are retrieved from memory to fill the scaler filter.

Bit 17 – YUV422SWP YUV 4:2:2 Swap

Value	Description
0	The two Y components of the YUV 4:2:2 packed data stream are not swapped
1	The two Y components of the YUV 4:2:2 packed data stream are swapped

Bit 16 – YUV422ROT YUV 4:2:2 Rotation
This bit is relevant only when a rotation angle of 90 degrees or 270 degrees is used.

Value	Description
0	Chroma Upsampling kernel is configured to use 0 and 180 degrees algorithm
1	Chroma Upsampling kernel is configured to use the 4:2:2 Rotation Algorithm.

Bits 15:12 – YUVMODE[3:0] YUV Mode Input Selection

Value	Name	Description
0	32BPP_AYCBCR	32 bpp AYCbCr 444
1	16BPP_YCBCR_MODE0	16 bpp Cr(n)Y(n+1)Cb(n)Y(n) 422
2	16BPP_YCBCR_MODE1	16 bpp Y(n+1)Cr(n)Y(n)Cb(n) 422
3	16BPP_YCBCR_MODE2	16 bpp Cb(n)Y(+1)Cr(n)Y(n) 422
4	16BPP_YCBCR_MODE3	16 bpp Y(n+1)Cb(n)Y(n)Cr(n) 422
5	16BPP_YCBCR_SEMIPLANAR	16 bpp Semiplanar 422 YCbCr
6	16BPP_YCBCR_PLANAR	16 bpp Planar 422 YCbCr
7	12BPP_YCBCR_SEMIPLANAR	12 bpp Semiplanar 420 YCbCr
8	12BPP_YCBCR_PLANAR	12 bpp Planar 420 YCbCr

Downloaded from Arrow.com.

Bits 9:8 – CLUTMODE[1:0] Color Lookup Table Mode Input Selection

Value	Name	Description
0	CLUT_1BPP	Color Lookup Table mode set to 1 bit per pixel
1	CLUT_2BPP	Color Lookup Table mode set to 2 bits per pixel
2	CLUT_4BPP	Color Lookup Table mode set to 4 bits per pixel
3	CLUT_8BPP	Color Lookup Table mode set to 8 bits per pixel

Bits 7:4 – RGBMODE[3:0] RGB Mode Input Selection

Value	Name	Description
0	12BPP_RGB_444	12 bpp RGB 444
1	16BPP_ARGB_4444	16 bpp ARGB 4444
2	16BPP_RGBA_4444	16 bpp RGBA 4444
3	16BPP_RGB_565	16 bpp RGB 565
4	16BPP_TRGB_1555	16 bpp TRGB 1555
5	18BPP_RGB_666	18 bpp RGB 666
6	18BPP_RGB_666PACKED	18 bpp RGB 666 PACKED
7	19BPP_TRGB_1666	19 bpp TRGB 1666
8	19BPP_TRGB_PACKED	19 bpp TRGB 1666 PACKED
9	24BPP_RGB_888	24 bpp RGB 888
10	24BPP_RGB_888_PACKED	24 bpp RGB 888 PACKED
11	25BPP_TRGB_1888	25 bpp TRGB 1888
12	32BPP_ARGB_8888	32 bpp ARGB 8888
13	32BPP_RGBA_8888	32 bpp RGBA 8888

Bit 1 – YUVEN YUV Color Space Enable
Color Space is YUV

Value	Description
0	Color space is RGB
1	Color space is YUV

Bit 0 – CLUTEN Color Lookup Table Mode Enable

Value	Description
0	RGB mode is selected.
1	Color Lookup Table mode is selected.

Downloaded from Arrow.com.

37.7.103 High-End Overlay Configuration Register 2

Name: LCDC_HEOCFG2
Offset: 0x000003B4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						YPOS[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	YPOS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						XPOS[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	XPOS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – YPOS[10:0] Vertical Window Position
High-End Overlay Vertical window position.

Bits 10:0 – XPOS[10:0] Horizontal Window Position
High-End Overlay Horizontal window position.

Downloaded from Arrow.com.

37.7.104 High-End Overlay Configuration Register 3

Name: LCDC_HEOCFG3
Offset: 0x000003B8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						YSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	YSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						XSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	XSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – YSIZE[10:0] Vertical Window Size
High-End Overlay window height in pixels. The window height is set to (YSIZE + 1).
The following constraint must be met: YPOS + YSIZE ≤ RPF

Bits 10:0 – XSIZE[10:0] Horizontal Window Size
High-End Overlay window width in pixels. The window width is set to (XSIZE + 1).
The following constraint must be met: XPOS + XSIZE ≤ PPL

Downloaded from Arrow.com.

37.7.105 High-End Overlay Configuration Register 4

Name: LCDC_HEOCFG4
Offset: 0x000003BC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						YMEMSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	YMEMSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
						XMEMSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	XMEMSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:16 – YMEMSIZE[10:0] Vertical image Size in Memory
High-End Overlay image height in pixels. The image height is set to (YMEMSIZE + 1).

Bits 10:0 – XMEMSIZE[10:0] Horizontal image Size in Memory
High-End Overlay image width in pixels. The image width is set to (XMEMSIZE + 1).

Downloaded from Arrow.com.

37.7.106 High-End Overlay Configuration Register 5

Name: LCDC_HEOCFG5
Offset: 0x000003C0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	XSTRIDE[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	XSTRIDE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	XSTRIDE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	XSTRIDE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – XSTRIDE[31:0] Horizontal Stride
XSTRIDE represents the memory offset, in bytes, between two rows of the image memory.

Downloaded from Arrow.com.

37.7.107 High-End Overlay Configuration Register 6

Name: LCDC_HEOCFG6
Offset: 0x000003C4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	PSTRIDE[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PSTRIDE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PSTRIDE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PSTRIDE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – PSTRIDE[31:0] Pixel Stride
PSTRIDE represents the memory offset, in bytes, between two pixels of the image memory.

Downloaded from Arrow.com.

37.7.108 High-End Overlay Configuration Register 7

Name: LCDC_HEOCFG7
Offset: 0x000003C8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	UVXSTRIDE[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	UVXSTRIDE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UVXSTRIDE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UVXSTRIDE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – UVXSTRIDE[31:0] UV Horizontal Stride
UVXSTRIDE represents the memory offset, in bytes, between two rows of the image memory.

Downloaded from Arrow.com.

37.7.109 High-End Overlay Configuration Register 8

Name: LCDC_HEOCFG8
Offset: 0x000003CC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	UVPSTRIDE[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	UVPSTRIDE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UVPSTRIDE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UVPSTRIDE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – UVPSTRIDE[31:0] UV Pixel Stride
UVPSTRIDE represents the memory offset, in bytes, between two pixels of the image memory.

Downloaded from Arrow.com.

37.7.110 High-End Overlay Configuration Register 9

Name: LCDC_HEOCFG9
Offset: 0x000003D0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RDEF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GDEF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BDEF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – RDEF[7:0] Red Default
Default Red color when the High-End Overlay DMA channel is disabled.

Bits 15:8 – GDEF[7:0] Green Default
Default Green color when the High-End Overlay DMA channel is disabled.

Bits 7:0 – BDEF[7:0] Blue Default
Default Blue color when the High-End Overlay DMA channel is disabled.

Downloaded from Arrow.com.

37.7.111 High-End Overlay Configuration Register 10

Name: LCDC_HEOCFG10
Offset: 0x000003D4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – RKEY[7:0] Red Color Component Chroma Key
Reference Red chroma key used to match the Red color of the current overlay.

Bits 15:8 – GKEY[7:0] Green Color Component Chroma Key
Reference Green chroma key used to match the Green color of the current overlay.

Bits 7:0 – BKEY[7:0] Blue Color Component Chroma Key
Reference Blue chroma key used to match the Blue color of the current overlay.

Downloaded from Arrow.com.

37.7.112 High-End Overlay Configuration Register 11

Name: LCDC_HEOCFG11
Offset: 0x000003D8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RMASK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GMASK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BMASK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – RMASK[7:0] Red Color Component Chroma Key Mask
Red Mask used when the compare function is used. If a bit is set then this bit is compared.

Bits 15:8 – GMASK[7:0] Green Color Component Chroma Key Mask
Green Mask used when the compare function is used. If a bit is set then this bit is compared.

Bits 7:0 – BMASK[7:0] Blue Color Component Chroma Key Mask
Blue Mask used when the compare function is used. If a bit is set then this bit is compared.

Downloaded from Arrow.com.

37.7.113 High-End Overlay Configuration Register 12

Name: LCDC_HEOCFG12
Offset: 0x000003DC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	GA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				VIDPRI		DSTKEY	REP	DMA
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
	OVR	LAEN	GAEN	REVALPHA	ITER	ITER2BL	INV	CRKEY
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – GA[7:0] Blender Global Alpha
Global alpha blender for the current layer.

Bit 12 – VIDPRI Video Priority

Value	Description
0	OVR1 layer is above HEO layer.
1	OVR1 layer is below HEO layer.

Bit 10 – DSTKEY Destination Chroma Keying

Value	Description
0	Source Chroma keying is enabled.
1	Destination Chroma keying is used.

Bit 9 – REP Use Replication logic to expand RGB color to 24 bits

Value	Description
0	When the selected pixel depth is less than 24 bpp the pixel is shifted and least significant bits are set to 0.
1	When the selected pixel depth is less than 24 bpp the pixel is shifted and the least significant bit replicates the msb.

Bit 8 – DMA Blender DMA Layer Enable

Value	Description
0	The default color is used on the Overlay 1 Layer.
1	The DMA channel retrieves the pixels stream from the memory.

Bit 7 – OVR Blender Overlay Layer Enable

Value	Description
0	Overlay pixel color is set to the default overlay pixel color.
1	Overlay pixel color is set to the DMA channel pixel color.

Downloaded from Arrow.com.

Bit 6 – LAEN Blender Local Alpha Enable

Value	Description
0	Local alpha blending coefficient is disabled.
1	Local alpha blending coefficient is enabled.

Bit 5 – GAEN Blender Global Alpha Enable

Value	Description
0	Global alpha blending coefficient is disabled.
1	Global alpha blending coefficient is enabled.

Bit 4 – REVALPHA Blender Reverse Alpha

Value	Description
0	Pixel difference is multiplied by alpha.
1	Pixel difference is multiplied by 1 - alpha.

Bit 3 – ITER Blender Use Iterated Color

Value	Description
0	Pixel difference is set to 0.
1	Pixel difference is set to the iterated pixel value.

Bit 2 – ITER2BL Blender Iterated Color Enable

Value	Description
0	Final adder stage operand is set to 0.
1	Final adder stage operand is set to the iterated pixel value.

Bit 1 – INV Blender Inverted Blender Output Enable

Value	Description
0	Iterated pixel is the blended pixel.
1	Iterated pixel is the inverted pixel.

Bit 0 – CRKEY Blender Chroma Key Enable

Value	Description
0	Chroma key matching is disabled.
1	Chroma key matching is enabled.

Downloaded from Arrow.com.

37.7.114 High-End Overlay Configuration Register 13

Name: LCDC_HEOCFG13
Offset: 0x000003E0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	SCALEN		YFACTOR[13:8]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	YFACTOR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			XFACTOR[13:8]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	XFACTOR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – SCALEN Hardware Scaler Enable

Value	Description
0	Scaler is disabled
1	Scaler is enabled.

Bits 29:16 – YFACTOR[13:0] Vertical Scaling Factor
Scaler Vertical Factor.

Bits 13:0 – XFACTOR[13:0] Horizontal Scaling Factor
Scaler Horizontal Factor.

Downloaded from Arrow.com.

37.7.115 High-End Overlay Configuration Register 14

Name: LCDC_HEOCFG14
Offset: 0x000003E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
		CSCYOFF	CSCRV[9:4]					
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CSCRV[3:0]				CSCRU[9:6]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CSCRU[5:0]						CSCRY[9:8]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CSCRY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 30 – CSCYOFF Color Space Conversion Offset

Value	Description
0	Offset is set to 0.
1	Offset is set to 16.

Bits 29:20 – CSCRV[9:0] Color Space Conversion V coefficient for Red Component 1:2:7 format
Color Space Conversion coefficient format is 1 sign bit, 2 magnitude bits and 7 fractional bits.

Bits 19:10 – CSCRU[9:0] Color Space Conversion U coefficient for Red Component 1:2:7 format
Color Space Conversion coefficient format is 1 sign bit, 2 magnitude bits and 7 fractional bits.

Bits 9:0 – CSCRY[9:0] Color Space Conversion Y coefficient for Red Component 1:2:7 format
Color Space Conversion coefficient format is 1 sign bit, 2 magnitude bits and 7 fractional bits.

Downloaded from Arrow.com.

37.7.116 High-End Overlay Configuration Register 15

Name: LCDC_HEOCFG15
Offset: 0x000003E8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
		CSCUOFF	CSCGV[9:4]					
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CSCGV[3:0]				CSCGU[9:6]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CSCGU[5:0]						CSCGY[9:8]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CSCGY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 30 – CSCUOFF Color Space Conversion Offset

Value	Description
0	Offset is set to 0.
1	Offset is set to 128.

Bits 29:20 – CSCGV[9:0] Color Space Conversion V coefficient for Green Component 1:2:7 format
Color Space Conversion coefficient format is 1 sign bit, 2 magnitude bits and 7 fractional bits.

Bits 19:10 – CSCGU[9:0] Color Space Conversion U coefficient for Green Component 1:2:7 format
Color Space Conversion coefficient format is 1 sign bit, 2 magnitude bits and 7 fractional bits.

Bits 9:0 – CSCGY[9:0] Color Space Conversion Y coefficient for Green Component 1:2:7 format
Color Space Conversion coefficient format is 1 sign bit, 2 magnitude bits and 7 fractional bits.

Downloaded from Arrow.com.

37.7.117 High-End Overlay Configuration Register 16

Name: LCDC_HEOCFG16
Offset: 0x000003EC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
		CSCVOFF	CSCBV[9:4]					
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CSCBV[3:0]				CSCBU[9:6]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CSCBU[5:0]						CSCBY[9:8]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CSCBY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 30 – CSCVOFF Color Space Conversion Offset

Value	Description
0	Offset is set to 0.
1	Offset is set to 128.

Bits 29:20 – CSCBV[9:0] Color Space Conversion V coefficient for Blue Component 1:2:7 format
Color Space Conversion coefficient format is 1 sign bit, 2 magnitude bits and 7 fractional bits.

Bits 19:10 – CSCBU[9:0] Color Space Conversion U coefficient for Blue Component 1:2:7 format
Color Space Conversion coefficient format is 1 sign bit, 2 magnitude bits and 7 fractional bits.

Bits 9:0 – CSCBY[9:0] Color Space Conversion Y coefficient for Blue Component 1:2:7 format
Color Space Conversion coefficient format is 1 sign bit, 2 magnitude bits and 7 fractional bits.

Downloaded from Arrow.com.

37.7.118 High-End Overlay Configuration Register 17

Name: LCDC_HEOCFG17
Offset: 0x000003F0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	XPHI0COEFF3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	XPHI0COEFF2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	XPHI0COEFF1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	XPHI0COEFF0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – XPHI0COEFF3[7:0] Horizontal Coefficient for phase 0 tap 3
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 23:16 – XPHI0COEFF2[7:0] Horizontal Coefficient for phase 0 tap 2
Coefficient format is 1 magnitude bit and 7 fractional bits.

Bits 15:8 – XPHI0COEFF1[7:0] Horizontal Coefficient for phase 0 tap 1
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 7:0 – XPHI0COEFF0[7:0] Horizontal Coefficient for phase 0 tap 0
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.119 High-End Overlay Configuration Register 18

Name: LCDC_HEOCFG18
Offset: 0x000003F4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	XPHI0COEFF4[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – XPHI0COEFF4[7:0] Horizontal Coefficient for phase 0 tap 4
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.120 High-End Overlay Configuration Register 19

Name: LCDC_HEOCFG19
Offset: 0x000003F8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	XPHI1COEFF3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	XPHI1COEFF2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	XPHI1COEFF1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	XPHI1COEFF0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – XPHI1COEFF3[7:0] Horizontal Coefficient for phase 1 tap 3
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 23:16 – XPHI1COEFF2[7:0] Horizontal Coefficient for phase 1 tap 2
Coefficient format is 1 magnitude bit and 7 fractional bits.

Bits 15:8 – XPHI1COEFF1[7:0] Horizontal Coefficient for phase 1 tap 1
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 7:0 – XPHI1COEFF0[7:0] Horizontal Coefficient for phase 1 tap 0
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.121 High-End Overlay Configuration Register 20

Name: LCDC_HEOCFG20
Offset: 0x000003FC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	XPHI1COEFF4[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – XPHI1COEFF4[7:0] Horizontal Coefficient for phase 1 tap 4
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.122 High-End Overlay Configuration Register 21

Name: LCDC_HEOCFG21
Offset: 0x00000400
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	XPHI2COEFF3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	XPHI2COEFF2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	XPHI2COEFF1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	XPHI2COEFF0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – XPHI2COEFF3[7:0] Horizontal Coefficient for phase 2 tap 3
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 23:16 – XPHI2COEFF2[7:0] Horizontal Coefficient for phase 2 tap 2
Coefficient format is 1 magnitude bit and 7 fractional bits.

Bits 15:8 – XPHI2COEFF1[7:0] Horizontal Coefficient for phase 2 tap 1
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 7:0 – XPHI2COEFF0[7:0] Horizontal Coefficient for phase 2 tap 0
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.123 High-End Overlay Configuration Register 22

Name: LCDC_HEOCFG22
Offset: 0x00000404
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	XPHI2COEFF4[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – XPHI2COEFF4[7:0] Horizontal Coefficient for phase 2 tap 4
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.124 High-End Overlay Configuration Register 23

Name: LCDC_HEOCFG23
Offset: 0x00000408
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	XPHI3COEFF3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	XPHI3COEFF2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	XPHI3COEFF1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	XPHI3COEFF0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – XPHI3COEFF3[7:0] Horizontal Coefficient for phase 3 tap 3
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 23:16 – XPHI3COEFF2[7:0] Horizontal Coefficient for phase 3 tap 2
Coefficient format is 1 magnitude bit and 7 fractional bits.

Bits 15:8 – XPHI3COEFF1[7:0] Horizontal Coefficient for phase 3 tap 1
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 7:0 – XPHI3COEFF0[7:0] Horizontal Coefficient for phase 3 tap 0
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.125 High-End Overlay Configuration Register 24

Name: LCDC_HEOCFG24
Offset: 0x0000040C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	XPHI3COEFF4[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – XPHI3COEFF4[7:0] Horizontal Coefficient for phase 3 tap 4
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.126 High-End Overlay Configuration Register 25

Name: LCDC_HEOCFG25
Offset: 0x00000410
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	XPHI4COEFF3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	XPHI4COEFF2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	XPHI4COEFF1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	XPHI4COEFF0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – XPHI4COEFF3[7:0] Horizontal Coefficient for phase 4 tap 3
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 23:16 – XPHI4COEFF2[7:0] Horizontal Coefficient for phase 4 tap 2
Coefficient format is 1 magnitude bit and 7 fractional bits.

Bits 15:8 – XPHI4COEFF1[7:0] Horizontal Coefficient for phase 4 tap 1
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 7:0 – XPHI4COEFF0[7:0] Horizontal Coefficient for phase 4 tap 0
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.127 High-End Overlay Configuration Register 26

Name: LCDC_HEOCFG26
Offset: 0x00000414
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	XPHI4COEFF4[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – XPHI4COEFF4[7:0] Horizontal Coefficient for phase 4 tap 4
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.128 High-End Overlay Configuration Register 27

Name: LCDC_HEOCFG27
Offset: 0x00000418
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	XPHI5COEFF3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	XPHI5COEFF2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	XPHI5COEFF1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	XPHI5COEFF0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – XPHI5COEFF3[7:0] Horizontal Coefficient for phase 5 tap 3
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 23:16 – XPHI5COEFF2[7:0] Horizontal Coefficient for phase 5 tap 2
Coefficient format is 1 magnitude bit and 7 fractional bits.

Bits 15:8 – XPHI5COEFF1[7:0] Horizontal Coefficient for phase 5 tap 1
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 7:0 – XPHI5COEFF0[7:0] Horizontal Coefficient for phase 5 tap 0
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.129 High-End Overlay Configuration Register 28

Name: LCDC_HEOCFG28
Offset: 0x0000041C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	XPHI5COEFF4[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – XPHI5COEFF4[7:0] Horizontal Coefficient for phase 5 tap 4
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.130 High-End Overlay Configuration Register 29

Name: LCDC_HEOCFG29
Offset: 0x00000420
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	XPHI6COEFF3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	XPHI6COEFF2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	XPHI6COEFF1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	XPHI6COEFF0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – XPHI6COEFF3[7:0] Horizontal Coefficient for phase 6 tap 3
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 23:16 – XPHI6COEFF2[7:0] Horizontal Coefficient for phase 6 tap 2
Coefficient format is 1 magnitude bit and 7 fractional bits.

Bits 15:8 – XPHI6COEFF1[7:0] Horizontal Coefficient for phase 6 tap 1
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 7:0 – XPHI6COEFF0[7:0] Horizontal Coefficient for phase 6 tap 0
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.131 High-End Overlay Configuration Register 30

Name: LCDC_HEOCFG30
Offset: 0x00000424
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	XPHI6COEFF4[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – XPHI6COEFF4[7:0] Horizontal Coefficient for phase 6 tap 4
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.132 High-End Overlay Configuration Register 31

Name: LCDC_HEOCFG31
Offset: 0x00000428
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	XPHI7COEFF3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	XPHI7COEFF2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	XPHI7COEFF1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	XPHI7COEFF0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – XPHI7COEFF3[7:0] Horizontal Coefficient for phase 7 tap 3
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 23:16 – XPHI7COEFF2[7:0] Horizontal Coefficient for phase 7 tap 2
Coefficient format is 1 magnitude bit and 7 fractional bits.

Bits 15:8 – XPHI7COEFF1[7:0] Horizontal Coefficient for phase 7 tap 1
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 7:0 – XPHI7COEFF0[7:0] Horizontal Coefficient for phase 7 tap 0
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.133 High-End Overlay Configuration Register 32

Name: LCDC_HEOCFG32
Offset: 0x0000042C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	XPHI7COEFF4[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – XPHI7COEFF4[7:0] Horizontal Coefficient for phase 7 tap 4
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.134 High-End Overlay Configuration Register 33

Name: LCDC_HEOCFG33
Offset: 0x00000430
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	YPHI0COEFF2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	YPHI0COEFF1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	YPHI0COEFF0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – YPHI0COEFF2[7:0] Vertical Coefficient for phase 0 tap 2
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 15:8 – YPHI0COEFF1[7:0] Vertical Coefficient for phase 0 tap 1
Coefficient format is 1 magnitude bit and 7 fractional bits.

Bits 7:0 – YPHI0COEFF0[7:0] Vertical Coefficient for phase 0 tap 0
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.135 High-End Overlay Configuration Register 34

Name: LCDC_HEOCFG34
Offset: 0x00000434
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	YPHI1COEFF2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	YPHI1COEFF1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	YPHI1COEFF0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – YPHI1COEFF2[7:0] Vertical Coefficient for phase 1 tap 2
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 15:8 – YPHI1COEFF1[7:0] Vertical Coefficient for phase 1 tap 1
Coefficient format is 1 magnitude bit and 7 fractional bits.

Bits 7:0 – YPHI1COEFF0[7:0] Vertical Coefficient for phase 1 tap 0
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.136 High-End Overlay Configuration Register 35

Name: LCDC_HEOCFG35
Offset: 0x00000438
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	YPHI2COEFF2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	YPHI2COEFF1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	YPHI2COEFF0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – YPHI2COEFF2[7:0] Vertical Coefficient for phase 2 tap 2
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 15:8 – YPHI2COEFF1[7:0] Vertical Coefficient for phase 2 tap 1
Coefficient format is 1 magnitude bit and 7 fractional bits.

Bits 7:0 – YPHI2COEFF0[7:0] Vertical Coefficient for phase 2 tap 0
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.137 High-End Overlay Configuration Register 36

Name: LCDC_HEOCFG36
Offset: 0x0000043C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	YPHI3COEFF2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	YPHI3COEFF1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	YPHI3COEFF0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – YPHI3COEFF2[7:0] Vertical Coefficient for phase 3 tap 2
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 15:8 – YPHI3COEFF1[7:0] Vertical Coefficient for phase 3 tap 1
Coefficient format is 1 magnitude bit and 7 fractional bits.

Bits 7:0 – YPHI3COEFF0[7:0] Vertical Coefficient for phase 3 tap 0
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.138 High-End Overlay Configuration Register 37

Name: LCDC_HEOCFG37
Offset: 0x00000440
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	YPHI4COEFF2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	YPHI4COEFF1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	YPHI4COEFF0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – YPHI4COEFF2[7:0] Vertical Coefficient for phase 4 tap 2
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 15:8 – YPHI4COEFF1[7:0] Vertical Coefficient for phase 4 tap 1
Coefficient format is 1 magnitude bit and 7 fractional bits.

Bits 7:0 – YPHI4COEFF0[7:0] Vertical Coefficient for phase 4 tap 0
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.139 High-End Overlay Configuration Register 38

Name: LCDC_HEOCFG38
Offset: 0x00000444
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	YPHI5COEFF2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	YPHI5COEFF1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	YPHI5COEFF0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – YPHI5COEFF2[7:0] Vertical Coefficient for phase 5 tap 2
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 15:8 – YPHI5COEFF1[7:0] Vertical Coefficient for phase 5 tap 1
Coefficient format is 1 magnitude bit and 7 fractional bits.

Bits 7:0 – YPHI5COEFF0[7:0] Vertical Coefficient for phase 5 tap 0
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.140 High-End Overlay Configuration Register 39

Name: LCDC_HEOCFG39
Offset: 0x00000448
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	YPHI6COEFF2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	YPHI6COEFF1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	YPHI6COEFF0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – YPHI6COEFF2[7:0] Vertical Coefficient for phase 6 tap 2
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 15:8 – YPHI6COEFF1[7:0] Vertical Coefficient for phase 6 tap 1
Coefficient format is 1 magnitude bit and 7 fractional bits.

Bits 7:0 – YPHI6COEFF0[7:0] Vertical Coefficient for phase 6 tap 0
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.141 High-End Overlay Configuration Register 40

Name: LCDC_HEOCFG40
Offset: 0x0000044C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	YPHI7COEFF2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	YPHI7COEFF1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	YPHI7COEFF0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – YPHI7COEFF2[7:0] Vertical Coefficient for phase 7 tap 2
Coefficient format is 1 sign bit and 7 fractional bits.

Bits 15:8 – YPHI7COEFF1[7:0] Vertical Coefficient for phase 7 tap 1
Coefficient format is 1 magnitude bit and 7 fractional bits.

Bits 7:0 – YPHI7COEFF0[7:0] Vertical Coefficient for phase 7 tap 0
Coefficient format is 1 sign bit and 7 fractional bits.

Downloaded from Arrow.com.

37.7.142 High-End Overlay Configuration Register 41

Name: LCDC_HEOCFG41
Offset: 0x00000450
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						YPHIDEF[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						XPHIDEF[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 18:16 – YPHIDEF[2:0] Vertical Filter Phase Offset
XPHIDEF defines the index of the first coefficient set used when the vertical resampling operation is started.

Bits 2:0 – XPHIDEF[2:0] Horizontal Filter Phase Offset
XPHIDEF defines the index of the first coefficient set used when the horizontal resampling operation is started.

Downloaded from Arrow.com.

37.7.143 Base CLUT Register x

Name: LCDC_BASECLUTx
Offset: 0x0600 + x*0x04 [x=0..255]
Reset: 0
Property: Read/Write

Note: CLUT registers are located in embedded RAM.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RCLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GCLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BCLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – RCLUT[7:0] Red Color Entry
This field indicates the 8-bit width Red color of the color lookup table.

Bits 15:8 – GCLUT[7:0] Green Color Entry
This field indicates the 8-bit width Green color of the color lookup table.

Bits 7:0 – BCLUT[7:0] Blue Color Entry
This field indicates the 8-bit width Blue color of the color lookup table.

Downloaded from Arrow.com.

37.7.144 Overlay 1 CLUT Register x

Name: LCDC_OVR1CLUTx
Offset: 0x0A00 + x*0x04 [x=0..255]
Reset: 0
Property: Read/Write

Note: CLUT registers are located in embedded RAM.

Bit	31	30	29	28	27	26	25	24
	ACLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RCLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GCLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BCLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – ACLUT[7:0] Alpha Color Entry
This field indicates the 8-bit width Alpha channel of the color lookup table.

Bits 23:16 – RCLUT[7:0] Red Color Entry
This field indicates the 8-bit width Red color of the color lookup table.

Bits 15:8 – GCLUT[7:0] Green Color Entry
This field indicates the 8-bit width Green color of the color lookup table.

Bits 7:0 – BCLUT[7:0] Blue Color Entry
This field indicates the 8-bit width Blue color of the color lookup table.

Downloaded from Arrow.com.

37.7.145 Overlay 2 CLUT Register x

Name: LCDC_OVR2CLUTx
Offset: 0x0E00 + x*0x04 [x=0..255]
Reset: 0
Property: Read/Write

Note: CLUT registers are located in embedded RAM.

Bit	31	30	29	28	27	26	25	24
	ACLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RCLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GCLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BCLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – ACLUT[7:0] Alpha Color Entry
This field indicates the 8-bit width Alpha channel of the color lookup table.

Bits 23:16 – RCLUT[7:0] Red Color Entry
This field indicates the 8-bit width Red color of the color lookup table.

Bits 15:8 – GCLUT[7:0] Green Color Entry
This field indicates the 8-bit width Green color of the color lookup table.

Bits 7:0 – BCLUT[7:0] Blue Color Entry
This field indicates the 8-bit width Blue color of the color lookup table.

Downloaded from Arrow.com.

37.7.146 High-End Overlay CLUT Register x

Name: LCDC_HEOCLUTx
Offset: 0x1200 + x*0x04 [x=0..255]
Reset: 0
Property: Read/Write

Note: CLUT registers are located in embedded RAM.

Bit	31	30	29	28	27	26	25	24
	ACLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RCLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GCLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BCLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – ACLUT[7:0] Alpha Color Entry
This field indicates the 8-bit width Alpha channel of the color lookup table.

Bits 23:16 – RCLUT[7:0] Red Color Entry
This field indicates the 8-bit width Red color of the color lookup table.

Bits 15:8 – GCLUT[7:0] Green Color Entry
This field indicates the 8-bit width Green color of the color lookup table.

Bits 7:0 – BCLUT[7:0] Blue Color Entry
This field indicates the 8-bit width Blue color of the color lookup table.

Downloaded from Arrow.com.

38. 2D Graphics Engine (GFX2D)

38.1 Description

The 2D Graphics Engine (GFX2D) is a AHB-protocol drawing engine. It performs memory data move operation over one master bus port.

38.2 Embedded Characteristics

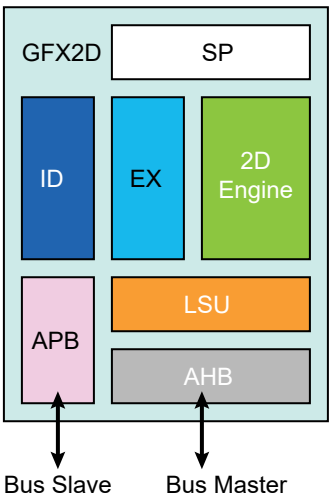
- 32-bit AHB Master Interface
- Ring Buffer Command Interface
- FILL, COPY, BLEND and ROP Instructions
- Classic and Special Blending Functions Available
- Alpha Channel, Indexed Color and True Color Supported
- Linear-to-Cartesian Coordinate Transformation
- Automatic Transfer Regulation

38.3 Block Diagram

The GFX2D module integrates the blocks listed in the following figure, where:

- SP is the Scratch Pad memory, used to store local pixels
- ID is the Instruction Decoder
- EX is the Execution Unit
- 2D Engine is the pixel processor
- LSU is the Load and Store Unit
- AHB and APB are the pixel/instruction interface and the configuration interface, respectively

Figure 38-1. GFX2D Block Diagram



38.4 Functional Description

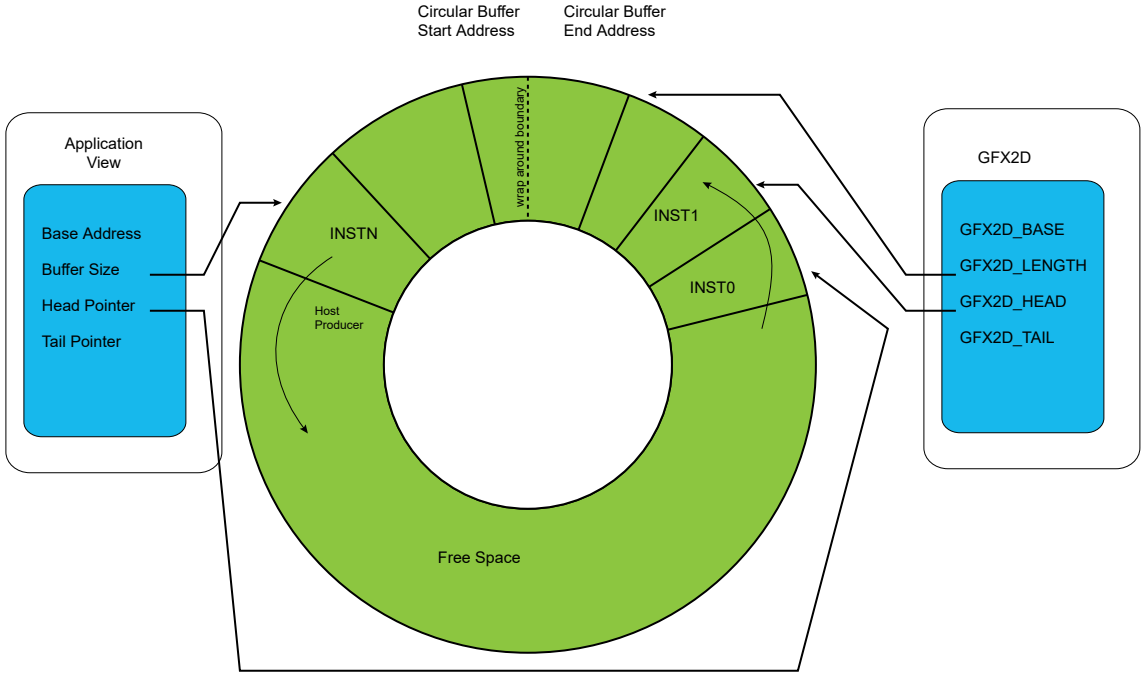
The GFX2D controller is an AHB-Bus master that executes a sequence of instructions stored in a ring buffer located in memory. Once the instruction decoding is complete, the graphics engine executes the instruction, performs memory accesses and, optionally, raises an interrupt. A single instruction does not modify the whole set of internal registers. Host software must use a load register instruction to change the graphics context.

Downloaded from Arrow.com.

38.4.1 Ring Buffer Management

38.4.1.1 Ring Buffer Model Diagram

Figure 38-2. Ring Buffer Model



38.4.1.2 Ring Buffer Allocation

The instruction Ring Buffer (RB) is a contiguous memory area shared between the processor and the graphics engine. The size of the ring buffer is 24 x (GFX2D_LEN.LEN+1) bytes.

It is an efficient and effective means of passing instructions to the graphics. It is non-blocking and it facilitates batch processing of surfaces. The total RB length is software-programmable, and is a multiple of the RB memory allocation unit. A large RB is recommended to reduce CPU intervention.

Use the following software procedure to allocate a new ring buffer:

1. Wait for command completion by polling the GFX2D_GS.BUSY bit.
2. Disable the ring buffer by writing to GFX2D_GD.DISABLE.
3. Initialize the tail and head pointers by setting GFX2D_HEAD.HEAD and GFX2D_TAIL.TAIL to ‘0’.
4. Program the ring buffer base address by setting GFX2D_BASE.BASE to the base address. The address must be 24 bytes aligned.
5. Program the ring buffer length by setting GFX2D_LEN.LEN to the length.
6. Enable the ring buffer by writing to GFX2D_GE.ENABLE.

38.4.1.3 Ring Buffer Push-Pull Model

The Ring Buffer uses the producer/consumer model. The processor pushes the required instructions to render frames into the RB. The graphics consume the instructions and inform the processor that these free slots can be recycled. The RB is full when the number of free slots is equal to one. When the tail pointer is equal to the read pointer, the RB is empty. A new instruction can be pushed only if there is enough space available to store the whole instruction. When the memory is updated with one or more instructions, the head pointer is incremented with the total number of words of the batch of instructions. If the end of the RB is reached, the head pointer wraps around.

Use the following software procedure to add a command to the ring buffer queue:

1. Monitor the Ring Buffer status from the Graphics side by reading the GFX2D_TAIL register. This register is updated by the hardware when the instruction has been successfully read from the memory.
2. Verify that there is enough space left in the buffer to insert the command. The queue is full when you have inserted N-1 instructions, where N is the number of entries.

Downloaded from Arrow.com.

3. Write the instruction at the memory location defined by GFX2D_BASE + GFX2D_HEAD.HEAD*4.
4. Increment the local head pointer and the GFX2D_HEAD.HEAD register to inform the graphics that new instructions have been added to the queue.
5. Wait for command completion by polling the BUSY bit in the GFX2D_GS register or wait for an interrupt if programmed.

38.4.1.4 Ring Buffer Disable

Use the following software procedure to halt the GFX2D:

1. Wait for command completion by polling the GFX2D_GS.BUSY bit.
2. Disable the ring buffer by writing '1' to GFX2D_GD.DISABLE.
3. Reset the tail and head pointers by writing '0' to GFX2D_HEAD.HEAD and GFX2D_TAIL.TAIL.

38.4.2 GFX2D Surface Memory Format

38.4.2.1 Source Surface Memory Format

38.4.2.1.1 4-bit Alpha Channel with 4-bit Indexed Color

Table 38-1. 4-bit Alpha Channel 4-bit Indexed Color Memory Mapping, Little Endian Organization

Mem. addr.	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel A4IDX4	A3				IDX3				A2				IDX2				A1				IDX1				A0				IDX0			

The 4-bit Indexed Color mode uses the Color Look-Up Table (CLUT) defined by the GFX2D_CFGx.IDXCX bit.

Table 38-2. 32-bpp Final Color Used at the Input of the Alpha Blending Stage, with Alpha Expansion

Mem. addr.	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st pixel	A0				A0				CLUT[IDX0].R								CLUT[IDX0].G								CLUT[IDX0].B							
2nd pixel	A1				A1				CLUT[IDX1].R								CLUT[IDX1].G								CLUT[IDX1].B							
3rd pixel	A2				A2				CLUT[IDX2].R								CLUT[IDX2].G								CLUT[IDX2].B							
4th pixel	A3				A3				CLUT[IDX3].R								CLUT[IDX3].G								CLUT[IDX3].B							

38.4.2.1.2 8-bit Alpha Channel

Table 38-3. 8-bit Alpha Channel Memory Mapping, Little Endian Organization

Mem. addr.	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 8 bpp	A3								A2								A1								A0							

The alpha channel is loaded from the memory and the color information is retrieved from the general-purpose register defined by the instruction's REG field.

Table 38-4. Constant Color in the General-Purpose Register Defined by Instruction's REG Field

Mem. addr.	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Downloaded from Arrow.com.

.....continued			
Mem. addr.	**0x3**	**0x2**	**0x1**	**0x0**
Pixel 8 bpp	A_C	R_C	G_C	B_C

Table 38-5. 32-bpp Final Color Used at the Input of the Alpha Blending Stage

Mem. addr.	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st pixel	A0								R_C								G_C								B_C							
2nd pixel	A1								R_C								G_C								B_C							
3rd pixel	A2								R_C								G_C								B_C							
4th pixel	A3								R_C								G_C								B_C							

38.4.2.1.3 8-bit Indexed Color

Table 38-6. 8-bit Indexed Color Memory Mapping, Little Endian Organization

Mem. addr.	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 8 bpp	IDX3								IDX2								IDX1								IDX0							

The 8-bit Indexed Color mode uses the Color Look-Up Table (CLUT) defined by the GFX2D_CFGx.IDXCX bit.

Table 38-7. 32-bpp Final Color Used at the Input of the Alpha Blending Stage

Mem. addr.	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st pixel	CLUT[IDX0].A								CLUT[IDX0].R								CLUT[IDX0].G								CLUT[IDX0].B							
2nd pixel	CLUT[IDX1].A								CLUT[IDX1].R								CLUT[IDX1].G								CLUT[IDX1].B							
3rd pixel	CLUT[IDX2].A								CLUT[IDX2].R								CLUT[IDX2].G								CLUT[IDX2].B							
4th pixel	CLUT[IDX3].A								CLUT[IDX3].R								CLUT[IDX3].G								CLUT[IDX3].B							

38.4.2.1.4 8-bit Alpha Channel with 8-bit Indexed Color

Table 38-8. 8-bit Alpha Channel 8-bit Indexed Color Memory Mapping, Little Endian Organization

Mem. addr.	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 8 bpp	A1								IDX1								A0								IDX0							

Table 38-9. 32-bpp Final Color Used at the Input of the Alpha Blending Stage

Mem. addr.	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st pixel	A0								CLUT[IDX0].R								CLUT[IDX0].G								CLUT[IDX0].B							

Downloaded from Arrow.com.

.....continued			
Mem. addr.	**0x3**	**0x2**	**0x1**	**0x0**
2nd pixel	A1	CLUT[IDX1].R	CLUT[IDX1].G	CLUT[IDX1].B

38.4.2.1.5 12-bpp Memory Mapping, RGB 4:4:4

Table 38-10. 12-bpp Memory Mapping, Little Endian Organization

Mem. addr.	**0x3**								**0x2**								**0x1**								**0x0**							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 12 bpp	–				R1[3:0]				G1[3:0]				B1[3:0]				–				R0[3:0]				G0[3:0]				B0[3:0]			

Table 38-11. 32-bpp Final Color Used at the Input of the Alpha Blending Stage with Color Expansion

Mem. addr.	**0x3**								**0x2**								**0x1**								**0x0**							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st pixel	0xFF								R0[3:0]				R0[3:0]				G0[3:0]				G0[3:0]				B0[3:0]				B0[3:0]			
2nd pixel	0xFF								R1[3:0]				R1[3:0]				G1[3:0]				G1[3:0]				B1[3:0]				B1[3:0]			

38.4.2.1.6 16-bpp Memory Mapping with 4-bit Alpha Channel, ARGB 4:4:4:4

Table 38-12. 32-bpp Final Color Used at the Input of the Alpha Blending Stage with Color Expansion

Mem. addr.	**0x3**								**0x2**								**0x1**								**0x0**							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st pixel	A0[3:0]				A0[3:0]				R0[3:0]				R0[3:0]				G0[3:0]				G0[3:0]				B0[3:0]				B0[3:0]			
2nd pixel	A1[3:0]				A1[3:0]				R1[3:0]				R1[3:0]				G1[3:0]				G1[3:0]				B1[3:0]				B1[3:0]			

38.4.2.1.7 16-bpp Memory Mapping with Transparency Bit, TRGB 5:5:5

Table 38-13. 16-bpp Memory Mapping, Little Endian Organization

Mem. addr.	**0x3**								**0x2**								**0x1**								**0x0**							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 4 bpp	T1	R1[4:0]					G1[4:0]					B1[4:0]					T0	R0[4:0]					G0[4:0]					B0[4:0]				

Table 38-14. 32-bpp Final Color Used at the Input of the Alpha Blending Stage with Color Expansion

Mem. addr.	**0x3**								**0x2**								**0x1**								**0x0**							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st pixel	T0	T0	T0	T0	T0	T0	T0	T0	R0[4:0]					R0[4:2]			G0[4:0]					G0[4:2]			B0[4:0]					B0[4:2]		
2nd pixel	T1	T1	T1	T1		T1	T1	T1	R1[4:0]					R1[4:2]			G1[4:0					G1[4:2]			B1[4:0]					B1[4:2]		

Downloaded from Arrow.com.

38.4.2.1.8 16-bpp Memory Mapping with Transparency Bit, RGBT 5:5:5:1

Table 38-15. 16-bpp Memory Mapping, Little Endian Organization

Mem. addr.	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 4 bpp	R1[4:0]					G1[4:0]			B1[4:0]							T1	R0[4:0]					G0[4:0]			B0[4:0]							T0

Table 38-16. 32-bpp Final Color Used at the Input of the Alpha Blending Stage with Color Expansion

Mem. addr.	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st pixel	T0	T0	T0	T0	T0	T0	T0	T0	R0[4:0]					R0[4:2]			G0[4:0]					G0[4:2]			B0[4:0]					B0[4:2]		
2nd pixel	T1	T1	T1	T1		T1	T1	T1	R1[4:0]					R1[4:2]			G1[4:0					G1[4:2]			B1[4:0]					B1[4:2]		

38.4.2.1.9 16-bpp Memory Mapping with Alpha Channel, RGB 5:6:5

Table 38-17. 16-bpp Memory Mapping, Little Endian Organization

Mem. addr.	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 16 bpp	R1[4:0]					G1[5:0]						B1[4:0]					R0[4:0]					G0[5:0]						B0[4:0]				

Table 38-18. 32-bpp Final Color Used at the Input of the Alpha Blending Stage with Color Expansion

Mem. addr.	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st pixel	0xFF								R0[4:0]					R0[4:2]			G0[5:0]						G0[5:4]		B0[4:0]					B0[4:2]		
2nd pixel	0xFF								R1[4:0]					R1[4:2]			G1[5:0]						G1[5:4]		B1[4:0]					B1[4:2]		

38.4.2.1.10 32-bpp Memory Mapping, ARGB 8:8:8:8

Table 38-19. 32-bpp Memory Mapping, Little Endian Organization

Mem. addr.	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 32 bpp	A0[7:0]								R0[7:0]								G0[7:0]								B0[7:0]							

38.4.2.1.11 32-bpp Memory Mapping, RGBA 8:8:8:8

Table 38-20. 32-bpp Memory Mapping, Little Endian Organization

Mem. addr.	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Downloaded from Arrow.com.

.....continued				
Mem. addr.	**0x3**	**0x2**	**0x1**	**0x0**
Pixel 32 bpp	R0[7:0]	G0[7:0]	B0[7:0]	A0[7:0]

38.4.2.2 Destination Surface Memory Format

38.4.2.2.1 4-bit Alpha Channel with 4-bit Luminance

The 24-bpp RGB final pixel is converted into an 8-bit luminance value and then truncated.

$$Y = \frac{0.299R + 0.587G + 0.114B}{16}$$

Table 38-21. 4-bit Alpha Channel with 4-bit Luminance Memory Mapping, Little Endian Organization

Mem. addr.	**0x3**								**0x2**								**0x1**								**0x0**							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel A4IDX4	A3				Y3				A2				Y2				A1				Y1				A0				Y0			

38.4.2.2.2 8-bit Alpha Channel

Table 38-22. 8-bit Alpha Memory Mapping, Little Endian Organization

Mem. addr.	**0x3**								**0x2**								**0x1**								**0x0**							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Alpha	A3								A2								A1								A0							

38.4.2.2.3 8-bit Luminance Channel

The 32-bpp ARGB final pixel is converted into an 8-bit luminance value.

$$Y = 0.299R + 0.587G + 0.114B$$

Table 38-23. 8-bit Luminance Memory Mapping, Little Endian Organization

Mem. addr.	**0x3**								**0x2**								**0x1**								**0x0**							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Alpha	Y3								Y2								Y1								Y0							

38.4.2.2.4 8-bit Alpha Channel with 8-bit Luminance

$$Y = 0.299R + 0.587G + 0.114B$$

The 32-bpp ARGB final pixel is converted into an 8-bit luminance value.

Table 38-24. 8-bit Alpha with 8-bit Luminance Memory Mapping, Little Endian Organization

Mem. addr.	**0x3**								**0x2**								**0x1**								**0x0**							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Alpha	A1								Y1								A0								Y0							

38.4.2.3 Color Look-Up Table (CLUT)

The GFX2D module includes two CLUTs. Each CLUT includes 256 entries containing 32 bits. Each entry contains a 32-bpp ARGB color.

Downloaded from Arrow.com.

Table 38-25. Color Look-Up Table Entry

Mem. addr.	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Color entry	A								R								G								B							

The CLUT can be loaded using an LDR instruction with the LDRC bit set. The LDRCID bit indicates the targeted CLUT. The LDRCS bit provides the amount of data to be loaded. Using a reduced CLUT mapping (i.e., 16 entries) reduces the number of accessible colors but improves loading time. The data structure is loaded from a memory location defined by the address in the REG field.

38.4.3 GFX2D Visible Registers

38.4.3.1 Register Naming

A set of 15 registers is defined and accessible through the use of the load and store architecture.

Table 38-26. Visible Registers

Value	Name	Description
0	GFX2D_PA0	Surface 0 Physical Address Register
1	GFX2D_PITCH0	Surface 0 Pitch Register
2	GFX2D_CFG0	Surface 0 Configuration Register
3	GFX2D_PA1	Surface 1 Physical Address Register
4	GFX2D_PITCH1	Surface 1 Pitch Register
5	GFX2D_CFG1	Surface 1 Configuration Register
6	GFX2D_PA2	Surface 2 Physical Address Register
7	GFX2D_PITCH2	Surface 2 Pitch Register
8	GFX2D_CFG2	Surface 2 Configuration Register
9	GFX2D_PA3	Surface 3 Physical Address Register
10	GPREG0	General-Purpose Register 0
11	GPREG1	General-Purpose Register 1
12	GPREG2	General-Purpose Register 2
13	GPREG3	General-Purpose Register 3
14	GPREG4	General-Purpose Register 4
15	GPREG5	General-Purpose Register 5

General-purpose registers are generic registers. They can hold:

- an address,
- a constant color for a FILL or BLEND operation,
- a constant alpha value for a BLEND, or
- a user-defined 32-bit data for a STORE operation.

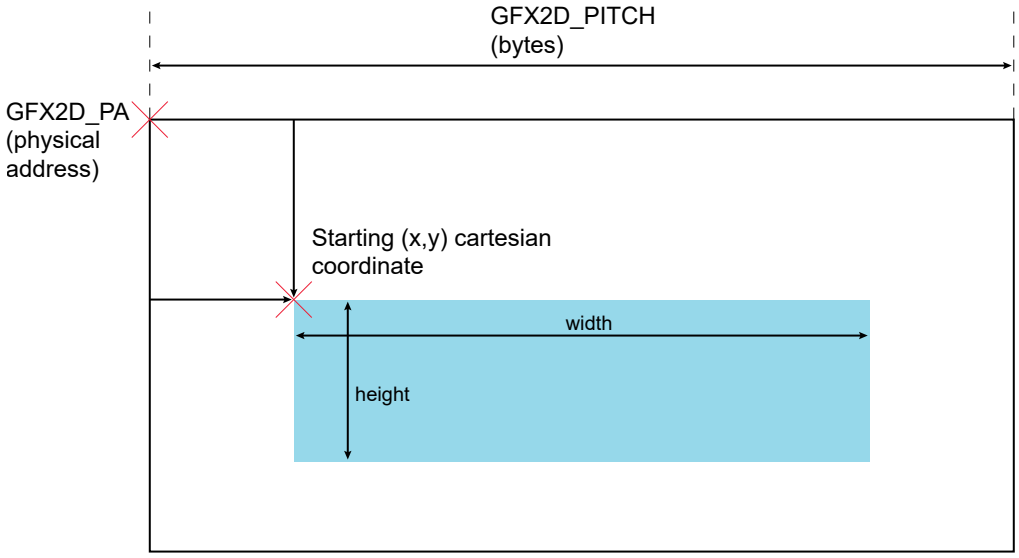
Downloaded from Arrow.com.

Table 38-27. General-Purpose Registers

Operation/ Instruction	General-Purpose Register
FILL	The register holds the 32-bit fill color pattern when the FILL arguments are less than 2. In that case, the color is defined by the REG field of the FILL instruction.
BLEND	The register holds the constant 32-bit color (A_c, R_c, G_c, B_c). It can be used for CONSTANT_ALPHA, CONSTANT_ALPHA parameters.
STR	The store operation requires a valid address defined by the REGAD field register index, and a data defined by the REG field register index. The general-purpose register can hold a 32-bit address or data.
LDR	The LDR instruction can load a general-purpose register.

38.4.3.2 Register Descriptions

Figure 38-3. Surface Registers



38.4.4 Traffic Balancing Using Outstanding Regulation

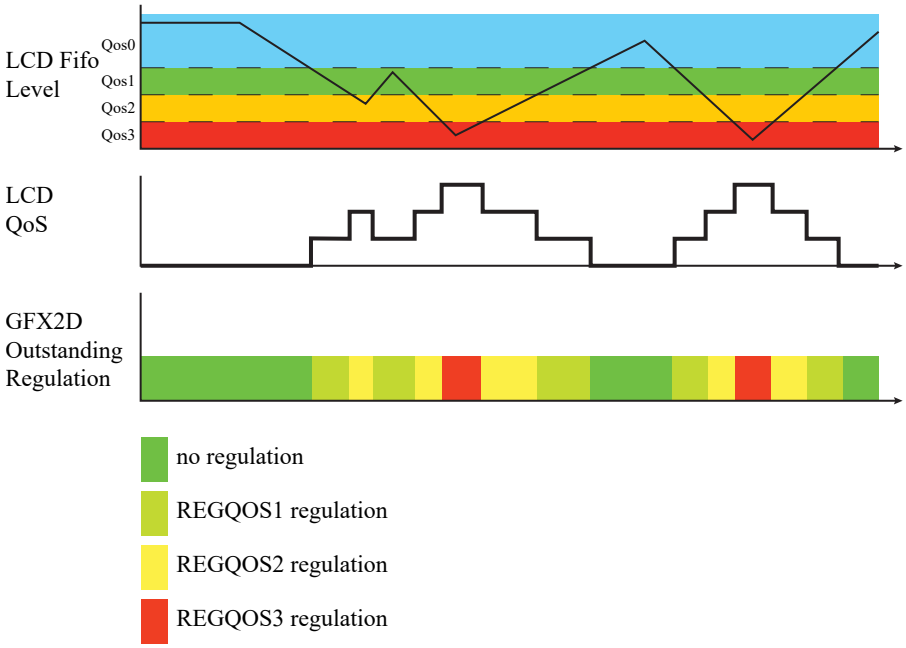
Global system performance is significantly impacted when the LCD controller is reading a frame buffer from memory to refresh the LCD screen. To ensure optimal performance, the LCD forwards its quality of service to other bus masters to limit their usage of the bus. The GFX2D frame rate is limited to allow latency-critical LCD operation.

Use the following procedure to activate traffic balancing:

1. Program the GFX2D_GC.REGQOS1, GFX2D_GC.REGQOS2 and GFX2D_GC.REGQOS3 registers and enable the outstanding issuance regulation by setting the GFX2D_GC.REGEN.
2. Configure the GFX2D_PCX register, and read the GFX2D_MCX register to adjust the regulation.

Downloaded from Arrow.com.

Figure 38-4. Traffic Balancing Using Outstanding Regulation



38.4.5 Data Flow Instructions

38.4.5.1 LDR Instruction

The LDR instruction length is variable and depends on the context.

- When the LDR instruction is used to load a register, the instruction length is 2 words.
- When the LDR instruction is used to load a CLUT, the instruction length is a single word.

Downloaded from Arrow.com.

38.4.5.1.1 LDR_WD0

Name: LDR_WD0

Bit	31	30	29	28	27	26	25	24
	OPCODE[3:0]							IE
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					REG[3:0]			
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						LDRCS	LDRCID	LDRC
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						ARGS[2:0]		
Access								
Reset								

Bits 31:28 – OPCODE[3:0] Instruction Code
This field must be set to '0x8'.

Bit 24 – IE Interrupt Enable

Value	Description
0	The End of Instruction interrupt is disabled.
1	The End of Instruction interrupt is enabled.

Bits 19:16 – REG[3:0] Register Identifier
This is the target register for a load operation. When the LDR instruction is used to load a Color Look-Up Table, the REG field points to the general-purpose register that holds the physical base address of the table.

Bit 10 – LDRCS Color Look-Up Table Size

Value	Description
0	256 entries.
1	16 entries.

Bit 9 – LDRCID Color Look-Up Table Selection

Value	Description
0	Table 0 is selected.
1	Table 1 is selected.

Bit 8 – LDRC Load Color Look-Up Table

Value	Description
0	Loads the register with the immediate value.
1	Loads the Color Look-Up Table with the content of the memory.

Bits 2:0 – ARGS[2:0] Arguments Length
This field must be set to '0'.

Downloaded from Arrow.com.

38.4.5.1.2 LDR_WD1

Name: LDR_WD1

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access								
Reset								

Bits 31:0 – DATA[31:0] 32-bit Data
The 32-bit data word loaded into the register.

38.4.5.2 STR Instruction

The instruction length is 1 word.

Downloaded from Arrow.com.

38.4.5.2.1 STR_WD0

Name: STR_WD0

Bit	31	30	29	28	27	26	25	24
	OPCODE[3:0]							IE
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					REG[3:0]			
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	REGAD[3:0]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						ARGS[2:0]		
Access								
Reset								

Bits 31:28 – OPCODE[3:0] Instruction Code
Must be set to '0x9'.

Bit 24 – IE Interrupt Enable

Value	Description
0	The End of Instruction interrupt is disabled.
1	The End of Instruction interrupt is enabled.

Bits 19:16 – REG[3:0] Register Identifier
Points to the general-purpose register that holds the data written to memory.

Bits 15:12 – REGAD[3:0] Register Identifier for Stored Address
Points to the general-purpose register that holds the address of the memory store operation.

Bits 2:0 – ARGS[2:0] Arguments Length
This field must be set to '0'.

38.4.5.3 WFE Instruction

The instruction length is 1 word.

Downloaded from Arrow.com.

38.4.5.3.1 WFE_WD0

Name: WFE_WD0

Bit	31	30	29	28	27	26	25	24
	OPCODE[3:0]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

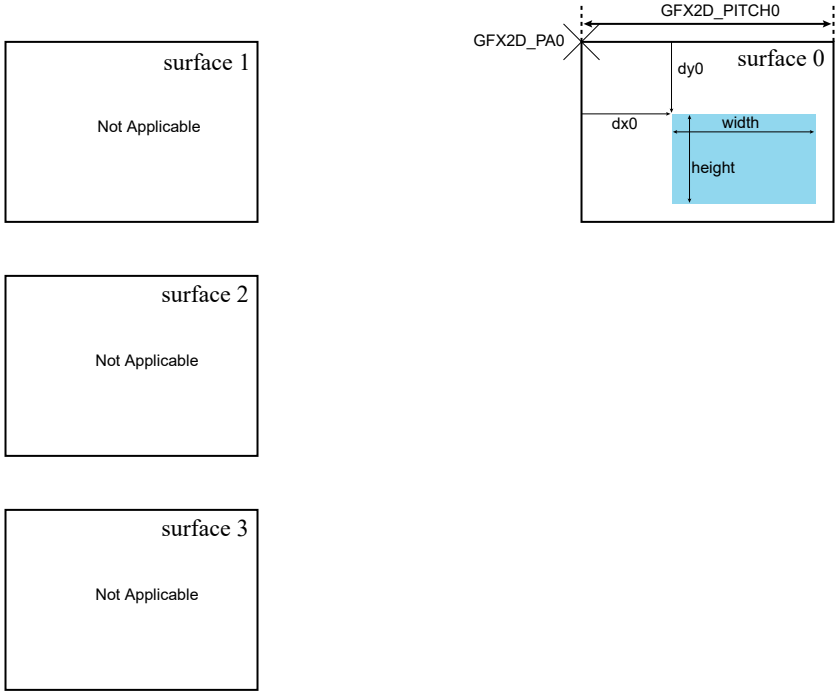
Bits 31:28 – OPCODE[3:0] Instruction Code
This field must be set to '0xA'.

38.4.6 Graphics Instructions

38.4.6.1 FILL Instruction

The FILL instruction length is 4.

Figure 38-5. Fill Operation



Downloaded from Arrow.com.

Table 38-28. FILL Instructions

Word Identifier	Description
FILL_WD0	The instruction header
FILL_WD1	The width and height of the destination-filled area
FILL_WD2	The x and y offsets of the filled area
FILL_WD3	The fill color: a 32-bit ARGB color resized to the area pixel format

Downloaded from Arrow.com.

38.4.6.1.1 FILL_WD0

Name: FILL_WD0

Bit	31	30	29	28	27	26	25	24
	OPCODE[3:0]							IE
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					REG[3:0]			
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							DIR[1:0]	
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						ARGS[2:0]		
Access								
Reset								

Bits 31:28 – OPCODE[3:0] Instruction Code
This field must be set to '0xB'.

Bit 24 – IE Interrupt Enable

Value	Description
0	The End of Instruction interrupt is disabled.
1	The End of Instruction interrupt is enabled.

Bits 19:16 – REG[3:0] Register Identifier
This field is relevant when ARGS is less than 2. In that case, the REG value points to a register with a valid 32-bit ARGB color.

Bits 9:8 – DIR[1:0] Transfer Direction

Value	Name	Description
0	XY00	Horizontal forward, vertical forward
1	XY01	Horizontal forward, vertical backward
2	XY10	Horizontal backward, vertical forward
3	XY11	Horizontal backward, vertical backward

Bits 2:0 – ARGS[2:0] Arguments Length
Must be less than or equal to 2.

Downloaded from Arrow.com.

38.4.6.1.2 FILL_WD1

Name: FILL_WD1

Bit	31	30	29	28	27	26	25	24
				HEIGHT[12:8]				
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	HEIGHT[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				WIDTH[12:8]				
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	WIDTH[7:0]							
Access								
Reset								

Bits 28:16 – HEIGHT[12:0] Destination Surface Height
The surface height is set to (HEIGHT+1) pixels.

Bits 12:0 – WIDTH[12:0] Destination Surface Width
The surface width is set to (WIDTH+1) pixels.

Downloaded from Arrow.com.

38.4.6.1.3 FILL_WD2

Name: FILL_WD2

Bit	31	30	29	28	27	26	25	24
				DY0[12:8]				
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DY0[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				DX0[12:8]				
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DX0[7:0]							
Access								
Reset								

Bits 28:16 – DY0[12:0] Destination Y Position
This field indicates the vertical position of the window.

Bits 12:0 – DX0[12:0] Destination X Position
This field indicates the horizontal position of the window.

Downloaded from Arrow.com.

38.4.6.1.4 FILL_WD3

Name: FILL_WD3

Bit	31	30	29	28	27	26	25	24
	A[7:0]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	R[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	G[7:0]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	B[7:0]							
Access								
Reset								

Bits 31:24 – A[7:0] Alpha Color Component Value
Program this field with the desired Alpha value.

Bits 23:16 – R[7:0] Red Color Component Value
Program this field with the desired Red value.

Bits 15:8 – G[7:0] Green Color Component Value
Program this field with the desired Green value.

Bits 7:0 – B[7:0] Blue Color Component Value
Program this field with the desired Blue value.

38.4.6.2 COPY Instruction

The instruction length is 4 words.

Downloaded from Arrow.com.

Figure 38-6. COPY Operation

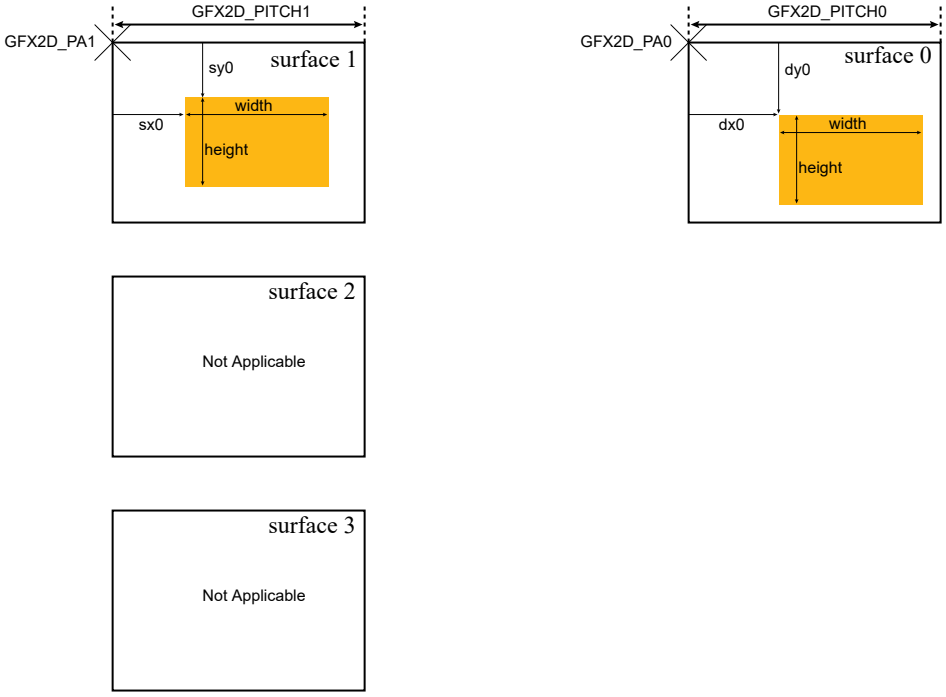


Table 38-29. COPY Instructions

Word Identifier	Description	Surface Register
COPY_WD0	The instruction header	Not applicable
COPY_WD1	The width and height of the destination area	Surface 0 register
COPY_WD2	The x and y offsets of the destination area	Surface 0 register
COPY_WD3	The x and y offsets of the source area	Surface 1 register

Downloaded from Arrow.com.

38.4.6.2.1 COPY_WD0

Name: COPY_WD0

Bit	31	30	29	28	27	26	25	24
	OPCODE[3:0]							IE
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					REG[3:0]			
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							DIR[1:0]	
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						ARGS[2:0]		
Access								
Reset								

Bits 31:28 – OPCODE[3:0] Instruction Code
This field must be set to '0xC'.

Bit 24 – IE Interrupt Enable

Value	Description
0	The End of Instruction interrupt is disabled.
1	The End of Instruction interrupt is enabled.

Bits 19:16 – REG[3:0] Register Identifier
When a constant is used, the constant value is located in the register defined by the REG value.

Bits 9:8 – DIR[1:0] Transfer Direction

Value	Name	Description
0	XY00	Horizontal forward, vertical forward
1	XY01	Horizontal forward, vertical backward
2	XY10	Horizontal backward, vertical forward
3	XY11	Horizontal backward, vertical backward

Bits 2:0 – ARGS[2:0] Arguments Length
Must be set to '2'.

Downloaded from Arrow.com.

38.4.6.2.2 COPY_WD1

Name: COPY_WD1

Bit	31	30	29	28	27	26	25	24
				HEIGHT[12:8]				
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	HEIGHT[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				WIDTH[12:8]				
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	WIDTH[7:0]							
Access								
Reset								

Bits 28:16 – HEIGHT[12:0] Destination Surface Height
The surface height is set to (HEIGHT+1) pixels.

Bits 12:0 – WIDTH[12:0] Destination Surface Width
The surface width is set to (WIDTH+1) pixels.

Downloaded from Arrow.com.

38.4.6.2.3 COPY_WD2

Name: COPY_WD2

Bit	31	30	29	28	27	26	25	24
				DY0[12:8]				
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DY0[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				DX0[12:8]				
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DX0[7:0]							
Access								
Reset								

Bits 28:16 – DY0[12:0] Destination Y Position
This field indicates the vertical position of the window.

Bits 12:0 – DX0[12:0] Destination X Position
This field indicates the horizontal position of the window.

Downloaded from Arrow.com.

38.4.6.2.4 COPY_WD3

Name: COPY_WD3

Bit	31	30	29	28	27	26	25	24
				SY0[12:8]				
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SY0[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				SX0[12:8]				
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	SX0[7:0]							
Access								
Reset								

Bits 28:16 – SY0[12:0] Source Y Position
This field indicates the vertical position of the window.

Bits 12:0 – SX0[12:0] Source X Position
This field indicates the horizontal position of the window.

38.4.6.3 BLEND Instruction

The instruction length is 6 words.

Downloaded from Arrow.com.

Figure 38-7. BLEND Operation

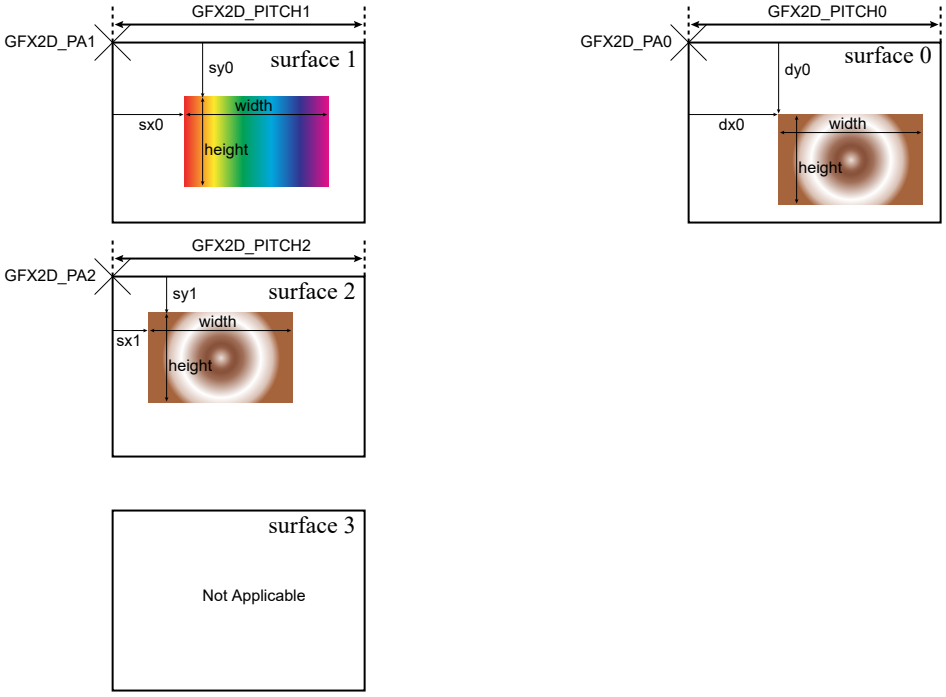


Table 38-30. BLEND Instructions

Word Identifier	Description	Surface Register
BLEND_WD0	The Instruction header	Not applicable
BLEND_WD1	The width and height of the destination area (C_f)	Surface 0 registers
BLEND_WD2	The x and y offsets of the destination area (C_f)	Surface 0 registers
BLEND_WD3	The x and y offsets of source 0 (C_d)	Surface 1 registers
BLEND_WD4	The x and y offsets of source 1 (C_s)	Surface 2 registers
BLEND_WD5	This register configures the blending functions and factors	Not applicable

Downloaded from Arrow.com.

38.4.6.3.1 BLEND_WD0

Name: BLEND_WD0

Bit	31	30	29	28	27	26	25	24
	OPCODE[3:0]							IE
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					REG[3:0]			
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							DIR[1:0]	
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						ARGS[2:0]		
Access								
Reset								

Bits 31:28 – OPCODE[3:0] Instruction Code
This field must be set to '0xD'.

Bit 24 – IE Interrupt Enable

Value	Description
0	The End of Instruction interrupt is disabled.
1	The End of Instruction interrupt is enabled.

Bits 19:16 – REG[3:0] Register Identifier
When a constant is used, the constant value is located in the register defined by the REG value.

Bits 9:8 – DIR[1:0] Transfer Direction

Value	Name	Description
0	XY00	Horizontal forward, vertical forward
1	XY01	Horizontal forward, vertical backward
2	XY10	Horizontal backward, vertical forward
3	XY11	Horizontal backward, vertical backward

Bits 2:0 – ARGS[2:0] Arguments Length
Must be set to '4'.

Downloaded from Arrow.com.

38.4.6.3.2 BLEND_WD1

Name: BLEND_WD1

Bit	31	30	29	28	27	26	25	24
				HEIGHT[12:8]				
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	HEIGHT[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				WIDTH[12:8]				
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	WIDTH[7:0]							
Access								
Reset								

Bits 28:16 – HEIGHT[12:0] Destination Surface Height
The surface height is set to (HEIGHT+1) pixels.

Bits 12:0 – WIDTH[12:0] Destination Surface Width
The surface width is set to (WIDTH+1) pixels.

Downloaded from Arrow.com.

38.4.6.3.3 BLEND_WD2

Name: BLEND_WD2

Bit	31	30	29	28	27	26	25	24
				DY0[12:8]				
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DY0[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				DX0[12:8]				
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DX0[7:0]							
Access								
Reset								

Bits 28:16 – DY0[12:0] Destination Y Position
This field indicates the vertical position of the window.

Bits 12:0 – DX0[12:0] Destination X Position
This field indicates the horizontal position of the window.

Downloaded from Arrow.com.

38.4.6.3.4 BLEND_WD3

Name: BLEND_WD3

Bit	31	30	29	28	27	26	25	24
				SY0[12:8]				
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SY0[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				SX0[12:8]				
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	SX0[7:0]							
Access								
Reset								

Bits 28:16 – SY0[12:0] Source 0 Y Position
This field indicates the vertical position of the window.

Bits 12:0 – SX0[12:0] Source 0 X Position
This field indicates the horizontal position of the window.

Downloaded from Arrow.com.

38.4.6.3.5 BLEND_WD3

Name: BLEND_WD3

Bit	31	30	29	28	27	26	25	24
				SY1[12:8]				
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SY1[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				SX1[12:8]				
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	SX1[7:0]							
Access								
Reset								

Bits 28:16 – SY1[12:0] Source 1 Y Position
This field indicates the vertical position of the window.

Bits 12:0 – SX1[12:0] Source 1 X Position
This field indicates the horizontal position of the window.

Downloaded from Arrow.com.

38.4.6.3.6 BLEND_WD5

Name: BLEND_WD5

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access	SPE[3:0]					FUNC[2:0]		
Reset								

Bit	7	6	5	4	3	2	1	0
Access					SFACT[3:0]			
Reset								

Bits 15:12 – SPE[3:0] Special Blend

Value	Name	Description
0	LIGHTEN	Based on the color information in each channel, selects the base color or the blend color, whichever is lighter, as the resulting color.
1	DARKEN	Based on the color information in each channel, selects the base color or the blend color, whichever is darker, as the resulting color.
2	MULTIPLY	Based on the color information in each channel, multiplies the base color by the blend color. The result is always a darker color.
3	AVERAGE	Based on the color information in each channel, averages the base color and the blend color to generate the resulting color.
4	ADD	This blend mode adds pixel values.
5	SUBTRACT	Based on the color information in each channel, subtracts the blend color from the base color.
6	DIFFERENCE	Based on the color information in each channel, subtracts either the blend color from the base color or the base color from the blend color, depending on which has the greater brightness value.
7	NEGATION	This mode produces the opposite effect to difference.
8	SCREEN	Based on each channel's color information, multiplies the inverse of the blend and base colors. The resulting color is always a lighter color.
9	OVERLAY	Multiplies or screens the colors, depending on the base color.
10	DODGE	Based on the color information in each channel, brightens the base color to reflect the blend color by decreasing the contrast between the two.
11	BURN	Based on the color information in each channel, darkens the base color to reflect the blend color by increasing the contrast between the two.
12	REFLECT	This blend mode can be used to add shiny objects or area if light. Black pixels in the blend are ignored as if they were transparent.
13	GLOW	The reverse of the REFLECT mode. GLOW effectively brightens the composition by the amount of brightness in the blend layer. Black pixels in the blend layer are rendered as if they were transparent.

Bits 10:8 – FUNC[2:0] Blending Function

Downloaded from Arrow.com.

Value	Name	Description
0	ADD	$C_f=S*C_s+D*C_d$
1	SUBTRACT	$C_f=S*C_s-D*C_d$
2	REVERSE	$C_f=D*C_d+S*C_s$
3	MIN	$C_f=Min(C_s, C_d)$
4	MAX	$C_f=Max(C_s, C_d)$
5	SPE	Special blending functions

Bits 3:0 – SFACT[3:0] Source Factor

Value	Name	Description
0	ZERO	(0, 0, 0, 0)
1	ONE	(1, 1, 1, 1)
2	SRC_COLOR	(A_s, R_s, G_s, B_s)
3	ONE_MINUS_SRC_COLOR	$(1, 1, 1, 1)-(A_s, R_s, G_s, B_s)$
4	DST_COLOR	(A_d, R_d, G_d, B_d)
5	ONE_MINUS_DST_COLOR	$(1, 1, 1, 1)-(A_d, R_d, G_d, B_d)$
6	SRC_ALPHA	(A_s, A_s, A_s, A_s)
7	ONE_MINUS_SRC_ALPHA	$(1, 1, 1, 1)-(A_s, A_s, A_s, A_s)$
8	DST_ALPHA	(A_d, A_d, A_d, A_d)
9	ONE_MINUS_DST_ALPHA	$(1, 1, 1, 1)-(A_d, A_d, A_d, A_d)$
10	CONSTANT_COLOR	(A_c, R_c, G_c, B_c)
11	ONE_MINUS_CONSTANT_COLOR	$(1, 1, 1, 1)-(A_c, R_c, G_c, B_c)$
12	CONSTANT_ALPHA	(A_c, A_c, A_c, A_c)
13	ONE_MINUS_CONSTANT_ALPHA	$(1, 1, 1, 1)-(A_c, A_c, A_c, A_c)$
14	SRC_ALPHA_SATURATE	(1, i, i, i) where i is equal to the minimum between A_s and $1-A_d$

Bits 3:0 – DFACT[3:0] Destination Factor

Value	Name	Description
0	ZERO	(0, 0, 0, 0)
1	ONE	(1, 1, 1, 1)
2	SRC_COLOR	(A_s, R_s, G_s, B_s)
3	ONE_MINUS_SRC_COLOR	$(1, 1, 1, 1)-(A_s, R_s, G_s, B_s)$
4	DST_COLOR	(A_d, R_d, G_d, B_d)
5	ONE_MINUS_DST_COLOR	$(1, 1, 1, 1)-(A_d, R_d, G_d, B_d)$
6	SRC_ALPHA	(A_s, A_s, A_s, A_s)
7	ONE_MINUS_SRC_ALPHA	$(1, 1, 1, 1)-(A_s, A_s, A_s, A_s)$
8	DST_ALPHA	(A_d, A_d, A_d, A_d)
9	ONE_MINUS_DST_ALPHA	$(1, 1, 1, 1)-(A_d, A_d, A_d, A_d)$
10	CONSTANT_COLOR	(A_c, R_c, G_c, B_c)
11	ONE_MINUS_CONSTANT_COLOR	$(1, 1, 1, 1)-(A_c, R_c, G_c, B_c)$
12	CONSTANT_ALPHA	(A_c, A_c, A_c, A_c)
13	ONE_MINUS_CONSTANT_ALPHA	$(1, 1, 1, 1)-(A_c, A_c, A_c, A_c)$
14	SRC_ALPHA_SATURATE	(1, i, i, i) where i is equal to the minimum between A_s and $1-A_d$

38.4.6.4 ROP Instruction

The instruction length is 7 words.

Downloaded from Arrow.com.

Figure 38-8. ROP Operations

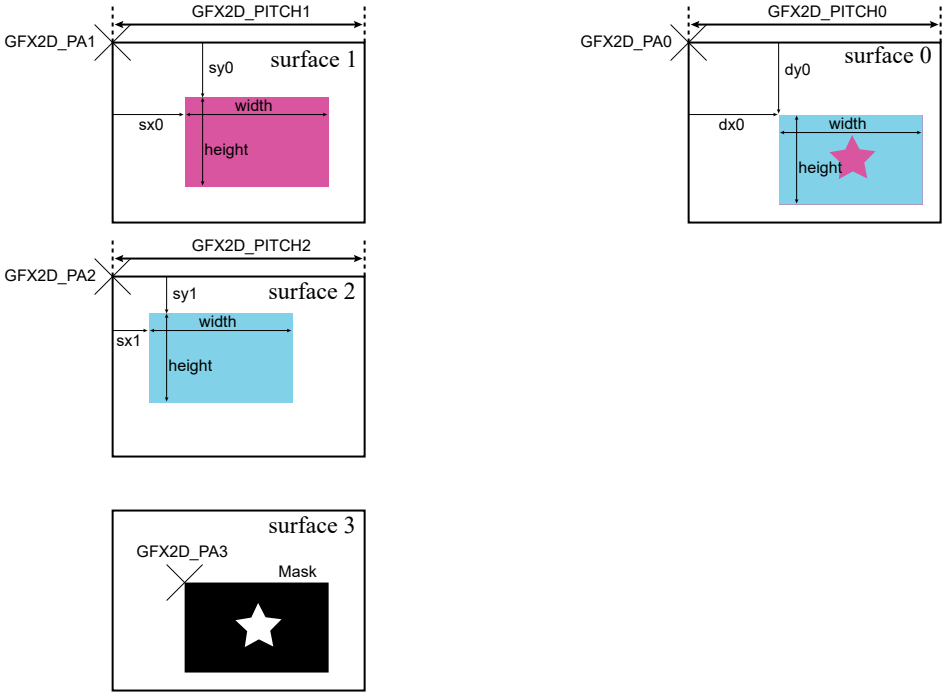


Table 38-31. ROP Instruction, Classic Operations

Word Identifier	Description	Surface Register
ROP_WD0	The instruction header	Not applicable
ROP_WD1	The width and height of the destination area D	Surface 0 registers
ROP_WD2	The x and y offsets of the destination area D	Surface 0 registers
ROP_WD3	The x and y offsets of source S	Surface 1 registers
ROP_WD4	The x and y offsets of pattern P	Surface 2 registers
ROP_WD5	A pointer to raster mask M	Surface 3 registers
ROP_WD6	Indicates the Raster Operation mode	Not applicable

Downloaded from Arrow.com.

38.4.6.4.1 ROP_WD0

Name: ROP_WD0

Bit	31	30	29	28	27	26	25	24
	OPCODE[3:0]							IE
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					REG[3:0]			
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						ARGS[2:0]		
Access								
Reset								

Bits 31:28 – OPCODE[3:0] Instruction Code
This field must be set to '0xE'.

Bit 24 – IE Interrupt Enable

Value	Description
0	The End of Instruction interrupt is disabled.
1	The End of Instruction interrupt is enabled.

Bits 19:16 – REG[3:0] Register Identifier
When a constant is used, the constant value is located in the register defined by the REG value.

Bits 2:0 – ARGS[2:0] Arguments Length
Must be set to '5'.

Downloaded from Arrow.com.

38.4.6.4.2 ROP_WD1

Name: ROP_WD1

Bit	31	30	29	28	27	26	25	24
				HEIGHT[12:8]				
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	HEIGHT[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				WIDTH[12:8]				
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	WIDTH[7:0]							
Access								
Reset								

Bits 28:16 – HEIGHT[12:0] Destination Surface Height
The surface height is set to (HEIGHT+1) pixels.

Bits 12:0 – WIDTH[12:0] Destination Surface Width
The surface width is set to (WIDTH+1) pixels.

Downloaded from Arrow.com.

38.4.6.4.3 ROP_WD2

Name: ROP_WD2

Bit	31	30	29	28	27	26	25	24
				DY0[12:8]				
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DY0[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				DX0[12:8]				
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DX0[7:0]							
Access								
Reset								

Bits 28:16 – DY0[12:0] Destination Y Position
This field indicates the vertical position of the window.

Bits 12:0 – DX0[12:0] Destination X Position
This field indicates the horizontal position of the window.

Downloaded from Arrow.com.

38.4.6.4.4 ROP_WD3

Name: ROP_WD3

Bit	31	30	29	28	27	26	25	24
				SY0[12:8]				
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SY0[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				SX0[12:8]				
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	SX0[7:0]							
Access								
Reset								

Bits 28:16 – SY0[12:0] Source Y Position
This field indicates the vertical position of the window.

Bits 12:0 – SX0[12:0] Source X Position
This field indicates the horizontal position of the window.

Downloaded from Arrow.com.

38.4.6.4.5 ROP_WD4

Name: ROP_WD4

Bit	31	30	29	28	27	26	25	24
				PY1[12:8]				
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	PY1[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				PX1[12:8]				
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PX1[7:0]							
Access								
Reset								

Bits 28:16 – PY1[12:0] Pattern Y Position
This field indicates the pattern Y position of the window.

Bits 12:0 – PX1[12:0] Pattern X Position
This field indicates the pattern X position of the window.

Downloaded from Arrow.com.

38.4.6.4.6 ROP_WD5

Name: ROP_WD5

Bit	31	30	29	28	27	26	25	24
	PMASK[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	PMASK[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PMASK[15:8]							
Access								
Reset								

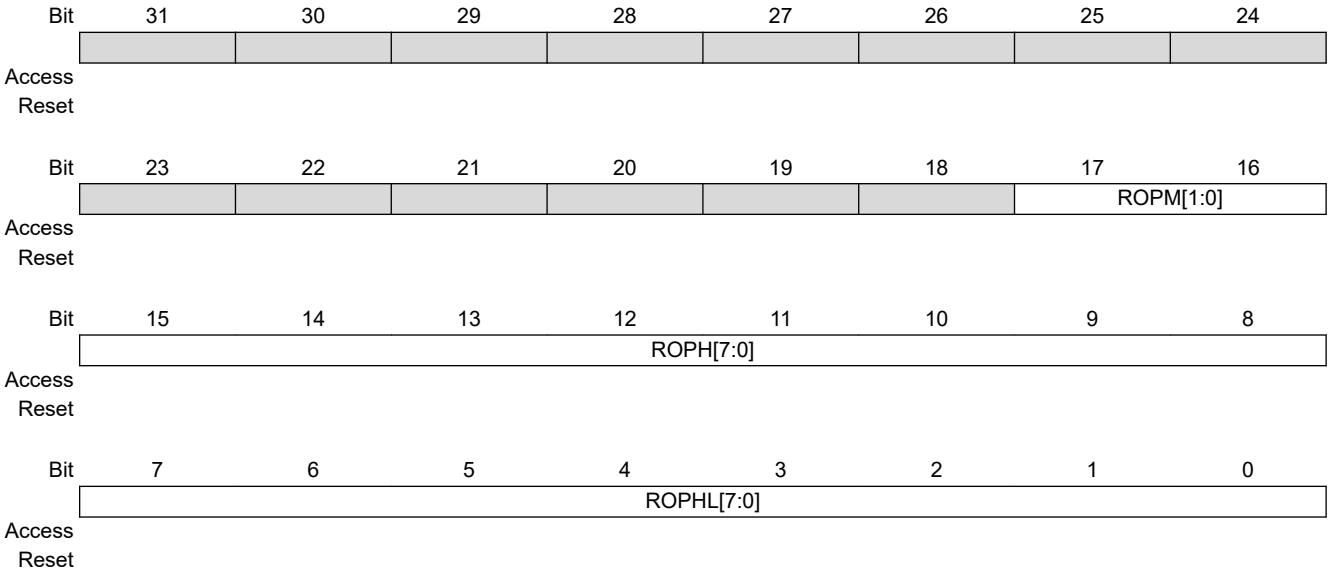
Bit	7	6	5	4	3	2	1	0
	PMASK[7:0]							
Access								
Reset								

Bits 31:0 – PMASK[31:0] Mask Pointer
This field must be set to the 32-bit address of the mask memory location.

Downloaded from Arrow.com.

38.4.6.4.7 ROP_WD6

Name: ROP_WD6



Bits 17:16 – ROPM[1:0] Raster Operation Mode

Value	Name	Description
0	ROP2	Binary raster operation (D, S) Two surfaces are used (required), 16 operations are defined, D and S required. Mask set to 0 internally Pattern set to 0 internally Source read from memory ROPL field value is used to generate the logical expression. ROPL can be any value between 0 and 15 (0x00 to 0x0F). ROPH field is ignored. ROP2_BLACK D = 0 (always 0, black in RGB mode) ROP2_NOTMERGESOURCE D = ~(D|S) Inverse of R2_MERGESOURCE ROP2_MASKNOTSOURCE D = D &~S Conjunction of destination and inverse of source ROP2_NOTCOPYSOURCE D = ~S Inverse of source color ROP2_MASKSOURCENOT D = S & ~D Conjunction of source and inverse of destination ROP2_NOT D = ~D Inverse of destination ROP2_XORSOURCE D = D ^ S Exclusive OR of destination and source ROP2_NOTMASKSOURCE D = ~(D&S) Inverse of ROP2_MASKSOURCE ROP2_MASKSOURCE D = D & P Conjunction of destination and source ROP2_NOTXORSOURCE D = ~(D ^ S) Inverse of ROP2_XORSOURCE ROP2_NOP D = D No change (copy) ROP2_MERGENOTSOURCE D = D | ~S Disjunction of destination and inverse of source ROP2_COPYSOURCE D = S

Downloaded from Arrow.com.

Value	Name	Description
		ROP2_MERGESOURCENOT D = S | ~D Disjunction of source and inverse of destination ROP2_MERGESOURCE D = S | D Conjunction of source and destination ROP2_WHITE D = 1
1	ROP3	Ternary raster operation (D, P, S) Three surfaces used, 256 operations Mask value set to 0 Source read from memory Pattern read from memory ROPL used to generate the logical expression 256 raster operations from 0x00 to 0xFF
2	ROP4	4-operand raster operation (D, P, S, M) Four surfaces used, 256 operations Mask value is read from memory (pointed by PMASK), mask is bit level. Source read from memory Pattern read from memory ROPL used to generate the logical expression when mask bit is cleared ROPH used to generate the logical expression when mask bit is set

Bits 15:8 – ROPH[7:0] 8-bit Raster Operation High

Bits 7:0 – ROPHL[7:0] 8-bit Raster Operation Low

Downloaded from Arrow.com.

38.5 Register Summary

Offset	Name	Bit Pos.								
0x00	GFX2D_GC	31:24								
		23:16					REGQOS3[3:0]			
		15:8	REGQOS2[3:0]				REGQOS1[3:0]			
		7:0		MTY		REGEN		CGDISFIFO	CGDISAXI	CGDISCORE
0x04	GFX2D_GE	31:24								
		23:16								
		15:8								
		7:0								ENABLE
0x08	GFX2D_GD	31:24								
		23:16								
		15:8								WFERES
		7:0								DISABLE
0x0C	GFX2D_GS	31:24								
		23:16								
		15:8								WFEIP
		7:0				BUSY				STATUS
0x10	GFX2D_IE	31:24								
		23:16								
		15:8								
		7:0				IERR	BERR	RERR	EXEND	RBEMPTY
0x14	GFX2D_ID	31:24								
		23:16								
		15:8								
		7:0				IERR	BERR	RERR	EXEND	RBEMPTY
0x18	GFX2D_IM	31:24								
		23:16								
		15:8								
		7:0				IERR	BERR	RERR	EXEND	RBEMPTY
0x1C	GFX2D_IS	31:24								
		23:16								
		15:8								
		7:0				IERR	BERR	RERR	EXEND	RBEMPTY
0x20	GFX2D_PC0	31:24								
		23:16								
		15:8								
		7:0		FILT[2:0]					SEL[1:0]	
0x24	GFX2D_MC0	31:24	COUNTER[31:24]							
		23:16	COUNTER[23:16]							
		15:8	COUNTER[15:8]							
		7:0	COUNTER[7:0]							
0x28	GFX2D_PC1	31:24								
		23:16								
		15:8								
		7:0		FILT[2:0]					SEL[1:0]	
0x2C	GFX2D_MC1	31:24	COUNTER[31:24]							
		23:16	COUNTER[23:16]							
		15:8	COUNTER[15:8]							
		7:0	COUNTER[7:0]							
0x30	GFX2D_BASE	31:24	BASE[23:16]							
		23:16	BASE[15:8]							
		15:8	BASE[7:0]							
		7:0								
0x34	GFX2D_LEN	31:24								
		23:16								
		15:8								
		7:0					LEN[3:0]			

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x38	GFX2D_HEAD	31:24								
		23:16								
		15:8							HEAD[9:8]	
		7:0	HEAD[7:0]							
0x3C	GFX2D_TAIL	31:24								
		23:16								
		15:8							TAIL[9:8]	
		7:0	TAIL[7:0]							
0x40	GFX2D_PA0	31:24	PA[31:24]							
		23:16	PA[23:16]							
		15:8	PA[15:8]							
		7:0	PA[7:0]							
0x44	GFX2D_PITCH0	31:24								
		23:16								
		15:8	PITCH[15:8]							
		7:0	PITCH[7:0]							
0x48	GFX2D_CFG0	31:24								
		23:16								
		15:8								
		7:0				IDXCX	PF[3:0]			
0x4C ... 0x4F	Reserved									
0x50	GFX2D_PA1	31:24	PA[31:24]							
		23:16	PA[23:16]							
		15:8	PA[15:8]							
		7:0	PA[7:0]							
0x54	GFX2D_PITCH1	31:24								
		23:16								
		15:8	PITCH[15:8]							
		7:0	PITCH[7:0]							
0x58	GFX2D_CFG1	31:24								
		23:16								
		15:8								
		7:0				IDXCX	PF[3:0]			
0x5C ... 0x5F	Reserved									
0x60	GFX2D_PA2	31:24	PA[31:24]							
		23:16	PA[23:16]							
		15:8	PA[15:8]							
		7:0	PA[7:0]							
0x64	GFX2D_PITCH2	31:24								
		23:16								
		15:8	PITCH[15:8]							
		7:0	PITCH[7:0]							
0x68	GFX2D_CFG2	31:24								
		23:16								
		15:8								
		7:0				IDXCX	PF[3:0]			
0x6C ... 0x6F	Reserved									
0x70	GFX2D_PA3	31:24	PA[31:24]							
		23:16	PA[23:16]							
		15:8	PA[15:8]							
		7:0	PA[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x74	GFX2D_PITCH3	31:24								
		23:16								
		15:8	PITCH[15:8]							
		7:0	PITCH[7:0]							
0x78	GFX2D_CFG3	31:24								
		23:16								
		15:8								
		7:0				IDXCX	PF[3:0]			

Downloaded from Arrow.com.

38.5.1 GFX2D Global Configuration Register

Name: GFX2D_GC
Offset: 0x00
Reset: 0x00065400
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					REGQOS3[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	1	1	0

Bit	15	14	13	12	11	10	9	8
	REGQOS2[3:0]				REGQOS1[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	1	0	1	0	0

Bit	7	6	5	4	3	2	1	0
		MTY		REGEN		CGDISFIFO	CGDISAXI	CGDISCORE
Access		R/W		R/W		R/W	R/W	R/W
Reset		0		0		0	0	0

Bits 19:16 – REGQOS3[3:0] Regulation for QoS Level 3
This register indicates the number of clock cycles inserted between outstanding transactions. The number of clock cycles added is calculated as follows.

$$\text{Latency} = 2^{\text{REGQOS3}} - 1$$

The maximum number of clock cycles is 1023.

Bits 15:12 – REGQOS2[3:0] Regulation for QoS Level 2
This register indicates the number of clock cycles inserted between outstanding transactions. The number of clock cycles added is calculated as follows.

$$\text{Latency} = 2^{\text{REGQOS2}} - 1$$

The maximum number of clock cycles is 1023.

Bits 11:8 – REGQOS1[3:0] Regulation for QoS Level 1
This register indicates the number of clock cycles inserted between outstanding transactions. The number of clock cycles added is calculated as follows.

$$\text{Latency} = 2^{\text{REGQOS1}} - 1$$

The maximum number of clock cycles is 1023.

Bit 6 – MTY Memory Tile Access

Value	Description
0	GFX2D uses tile accesses.
1	GFX2D uses linear accesses.

Bit 4 – REGEN Outstanding Regulation Enable

Value	Description
0	Outstanding Regulation is disabled.

Downloaded from Arrow.com.

Value	Description
1	Outstanding Regulation is enabled.

Bit 2 – CGDISFIFO Clock Gating Disable FIFO

Value	Description
0	Clock gating is activated for the FIFO.
1	Clock gating is disabled for the FIFO.

Bit 1 – CGDISAXI Clock Gating Disable AXI

Value	Description
0	Clock gating is activated for the AXI interface.
1	Clock gating is disabled for the AXI interface.

Bit 0 – CGDISCORE Clock Gating Disable Core

Value	Description
0	Clock gating is activated for the graphics engine.
1	Clock gating is disabled for the graphics engine.

Downloaded from Arrow.com.

38.5.2 GFX2D Global Enable Register

Name: GFX2D_GE
Offset: 0x04
Reset: 0x00000000
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ENABLE
Access								W
Reset								0

Bit 0 – ENABLE GFX2D Enable

Value	Description
0	No effect.
1	Enables the GFX2D controller. This operation is permitted if the global status bit was read as ‘0’.

Downloaded from Arrow.com.

38.5.3 GFX2D Global Disable Register

Name: GFX2D_GD
Offset: 0x08
Reset: 0x00000000
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								WFERES
Access								W
Reset								0

Bit	7	6	5	4	3	2	1	0
								DISABLE
Access								W
Reset								0

Bit 8 – WFERES WFE Software Resume bit

Value	Description
0	No effect.
1	Exits the WFE state in software.

Bit 0 – DISABLE GFX2D Disable Bit

Value	Description
0	No effect.
1	Disables the graphics engine.

Downloaded from Arrow.com.

38.5.4 GFX2D Global Status Register

Name: GFX2D_GS
Offset: 0x0C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								WFEIP
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
				BUSY				STATUS
Access				R				R
Reset				0				0

Bit 8 – WFEIP Wait For Event Status bit

Value	Description
0	The graphics core is running.
1	The graphics core is waiting for an event.

Bit 4 – BUSY GFX2D Busy Bit

Value	Description
0	The graphics core is in idle state.
1	The graphics core is busy.

Bit 0 – STATUS GFX2D Status Bit

Value	Description
0	The graphics engine is disabled.
1	The graphics engine is enabled.

Downloaded from Arrow.com.

38.5.5 GFX2D Interrupt Enable Register

Name: GFX2D_IE
Offset: 0x10
Reset: 0x00000000
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				IERR	BERR	RERR	EXEND	RBEMPTY
Access				W	W	W	W	W
Reset				0	0	0	0	0

Bit 4 – IERR Illegal Instruction Interrupt Enable Bit

Bit 3 – BERR Write Data Bus Error Interrupt Enable Bit

Bit 2 – RERR Read Data Bus Error Interrupt Enable Bit

Bit 1 – EXEND End of Execution Interrupt Enable Bit

Bit 0 – RBEMPTY Ring Buffer Empty Interrupt Enable Bit

Downloaded from Arrow.com.

38.5.6 GFX2D Interrupt Disable Register

Name: GFX2D_ID
Offset: 0x14
Reset: 0x00000000
Property: Write-only

The following configuration values are valid for all listed bit names of this register:
0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				IERR	BERR	RERR	EXEND	RBEMPTY
Access				W	W	W	W	W
Reset				0	0	0	0	0

Bit 4 – IERR Illegal Instruction Interrupt Disable bit

Bit 3 – BERR Write Access Error Interrupt Disable bit

Bit 2 – RERR Read Access Error Interrupt Disable Bit

Bit 1 – EXEND End of Execution Interrupt Disable Bit

Bit 0 – RBEMPTY Ring Buffer Empty Interrupt Disable Bit

Downloaded from Arrow.com.

38.5.7 GFX2D Interrupt Mask Register

Name: GFX2D_IM
Offset: 0x18
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:
0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				IERR	BERR	RERR	EXEND	RBEMPTY
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 4 – IERR Illegal Instruction Interrupt Mask Bit

Bit 3 – BERR Write Error Interrupt Mask Bit

Bit 2 – RERR Read Error Interrupt Mask Bit

Bit 1 – EXEND Execution Ended Empty Interrupt Mask Bit

Bit 0 – RBEMPTY Ring Buffer Empty Interrupt Mask Bit

Downloaded from Arrow.com.

38.5.8 GFX2D Interrupt Status Register

Name: GFX2D_IS
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				IERR	BERR	RERR	EXEND	RBEMPTY
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 4 – IERR Illegal Instruction Interrupt Status Bit

Value	Description
0	The interrupt source is masked or no Illegal Instruction interrupt is pending.
1	An Illegal Instruction interrupt is pending.

Bit 3 – BERR Write Error Interrupt Status Bit

Value	Description
0	Either the interrupt source is masked or no Write Error interrupt is pending.
1	A Write Error interrupt is pending.

Bit 2 – RERR Read Error Interrupt Status Bit

Value	Description
0	Either the interrupt source is masked or no read error interrupt is pending.
1	An interrupt is pending.

Bit 1 – EXEND End of Execution Status Bit

Value	Description
0	Either the interrupt source is masked or no End of Execution interrupt is pending.
1	An End of Execution interrupt is pending (i.e. EXEND is '1' when GFX2D_GS.BUSY is set to '0').

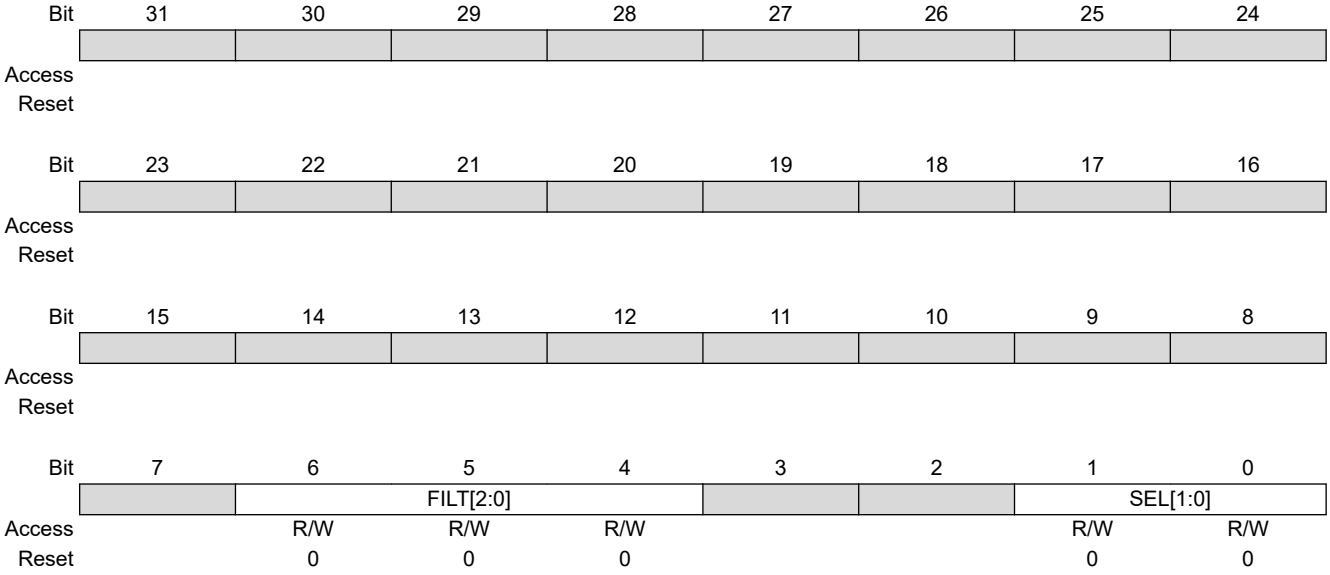
Bit 0 – RBEMPTY Ring Buffer Empty Interrupt Status Bit

Value	Description
0	The Ring Buffer Empty interrupt has not occurred.
1	The Ring Buffer Empty interrupt has occurred since the last read of the status register.

Downloaded from Arrow.com.

38.5.9 GFX2D Performance Configuration 0 Register

Name: GFX2D_PC0
Offset: 0x20
Reset: 0x00000000
Property: Read/Write



Bits 6:4 – FILT[2:0] Filter Configuration

Value	Name	Description
0	DISABLED	The filter is disabled.
1	QOS0	Events are valid when input QoS is equal to 0.
2	QOS1	Events are valid when input QoS is equal to 1.
3	QOS2	Events are valid when input QoS is equal to 2.
4	QOS3	Events are valid when input QoS is equal to 3.

Bits 1:0 – SEL[1:0] Performance Metrics Selection

Value	Name	Description
0	DISABLED	The performance counter is disabled and reset.
1	READ	The performance counter is incremented when a Read access is performed.
2	WRITE	The performance counter is incremented when a Write access is performed
3	CYCLE	Number of clock cycles

Downloaded from Arrow.com.

38.5.10 GFX2D Metrics Counter 0 Register

Name: GFX2D_MC0
Offset: 0x24
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	COUNTER[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNTER[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNTER[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNTER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNTER[31:0] Metrics Counter

Downloaded from Arrow.com.

38.5.11 GFX2D Performance Configuration 1 Register

Name: GFX2D_PC1
Offset: 0x28
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		FILT[2:0]					SEL[1:0]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		0	0	0			0	0

Bits 6:4 – FILT[2:0] Filter Configuration

Value	Name	Description
0	DISABLED	The filter is disabled.
1	QOS0	Events are valid when input QoS is equal to 0.
2	QOS1	Events are valid when input QoS is equal to 1.
3	QOS2	Events are valid when input QoS is equal to 2.
4	QOS3	Events are valid when input QoS is equal to 3.

Bits 1:0 – SEL[1:0] Performance Metrics Selection

Value	Name	Description
0	DISABLED	The performance counter is disabled and reset.
1	READ	The performance counter is incremented when a Read access is performed.
2	WRITE	The performance counter is incremented when a Write access is performed
3	CYCLE	Number of clock cycles

Downloaded from Arrow.com.

38.5.12 GFX2D Metrics Counter 1

Name: GFX2D_MC1
Offset: 0x2C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	COUNTER[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNTER[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNTER[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNTER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNTER[31:0] Metrics Counter

Downloaded from Arrow.com.

38.5.13 GFX2D Ring Buffer Base Register

Name: GFX2D_BASE
Offset: 0x30
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	BASE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BASE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BASE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 31:8 – BASE[23:0] Ring Buffer Base Register
This field is programmed with the Ring Buffer base address and is aligned on the allocation unit size. Ring buffer allocation unit is 2^8=256 bytes. The base address is 256 bytes aligned.

Downloaded from Arrow.com.

38.5.14 GFX2D Ring Buffer Length Register

Name: GFX2D_LEN
Offset: 0x34
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					LEN[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – LEN[3:0] Ring Buffer Length Multiplier
Program this field with the desired buffer length multiplier. Buffer length = 256*(LEN+1) bytes.

Downloaded from Arrow.com.

38.5.15 GFX2D Ring Buffer Head Register

Name: GFX2D_HEAD
Offset: 0x38
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							HEAD[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	HEAD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – HEAD[9:0] Ring Buffer Head Pointer
The head pointer is updated by software in Functional mode to indicate the number of words written to memory.

Downloaded from Arrow.com.

38.5.16 GFX2D Ring Buffer Tail Register

Name: GFX2D_TAIL
Offset: 0x3C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							TAIL[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	TAIL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – TAIL[9:0] Ring Buffer Tail Pointer
The TAIL pointer is updated by the graphics engine in Functional mode to indicate how many words have been consumed.

Downloaded from Arrow.com.

38.5.17 GFX2D Surface x Physical Address Register

Name: GFX2D_PAx
Offset: 0x40 + x*0x10 [x=0..3]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	PA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – PA[31:0] Surface Physical Start Address
This address must be aligned with the surface pixel format.

Downloaded from Arrow.com.

38.5.18 GFX2D Surface x Pitch Register

Name: GFX2D_PITCHx
Offset: 0x44 + x*0x10 [x=0..3]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PITCH[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PITCH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PITCH[15:0] Surface Pitch
This field indicates the surface pitch size in bytes.

Downloaded from Arrow.com.

38.5.19 GFX2D Surface x Configuration Register

Name: GFX2D_CFGx
Offset: 0x48 + x*0x10 [x=0..3]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				IDXCX	PF[3:0]			
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 4 – IDXCX Color Look-Up Table Selection

Value	Description
0	The indexed color is retrieved in Color Look-Up Table 0.
1	The indexed color is retrieved in Color Look-Up Table 1.

Bits 3:0 – PF[3:0] Pixel Format

Value	Name	Description
0	A4IDX4	4-bit indexed color, with 4-bit alpha value
1	A8	8 bits per pixel alpha, with user-defined constant color
2	IDX8	8-bit indexed color, uses the Color Look-Up Table to expand to true color
3	A8IDX8	8-bit indexed color, with 8-bit alpha value
4	RGB12	12 bits per pixel, 4 bits per color channel
5	ARGB16	16 bits per pixel with 4-bit width alpha value, and 4 bits per color channel
6	RGB15	15 bits per pixel, 5 bits per color channel
7	TRGB16	16 bits per pixel, 5 bits for the red and blue channels and 6 bits for the green channel
8	RGBT16	16 bits per pixel, with 1 bit for transparency and 5 bits for color channels
9	RGB16	16 bits per pixel, 5 bits for the red and blue channels and 6 bits for the green channel
10	RGB24	24 bits per pixel, 8 bits for alpha and color channels
11	ARGB32	32 bits per pixel, 8 bits for alpha and color channels
12	RGBA32	32 bits per pixel, 8 bits for alpha and color channels

Downloaded from Arrow.com.

39. Ethernet MAC 10/100 (EMAC)

39.1 Description

The EMAC module implements a 10/100 Ethernet MAC compatible with the IEEE 802.3 standard using an address checker, statistics and control registers, receive and transmit blocks, and a DMA interface.

The address checker recognizes four specific 48-bit addresses and contains a 64-bit hash register for matching multicast and unicast addresses. It can recognize the broadcast address of all ones, copy all frames, and act on an external address match signal.

The statistics register block contains registers for counting various types of event associated with transmit and receive operations. These registers, along with the status words stored in the receive buffer list, enable software to generate network management statistics compatible with IEEE 802.3.

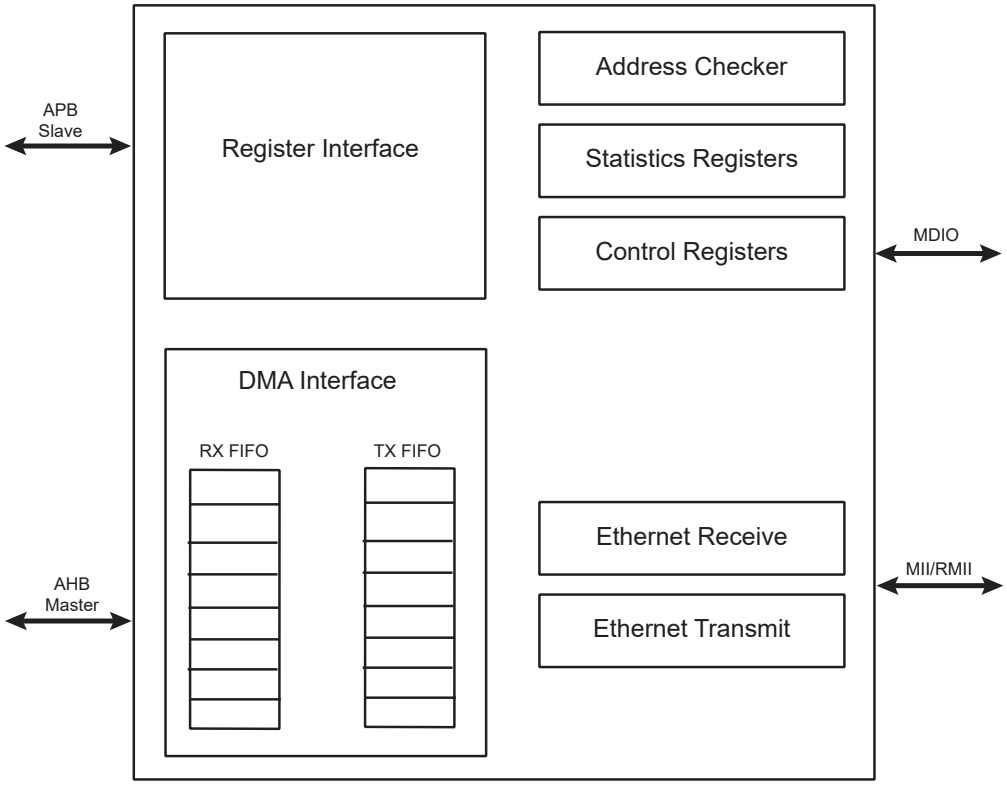
39.2 Embedded Characteristics

- EMAC0 supports MII and RMII Interfaces to the Physical Layer (EMAC1 supports RMII only)
- Compatible with IEEE Standard 802.3
- 10 and 100 Mbit/s Operation
- Full- and Half-duplex Operation
- Statistics Counter Registers
- Interrupt Generation to Signal Receive and Transmit Completion
- DMA Master on Receive and Transmit Channels
- Transmit and Receive FIFOs
- Automatic Pad and CRC Generation on Transmitted Frames
- Automatic Discard of Frames Received with Errors
- Address Checking Logic Supports Up to Four Specific 48-bit Addresses
- Supports Promiscuous Mode Where All Valid Received Frames are Copied to Memory
- Hash Matching of Unicast and Multicast Destination Addresses
- External Address Matching of Received Frames
- Physical Layer Management through MDIO Interface
- Half-duplex Flow Control by Forcing Collisions on Incoming Frames
- Full-duplex Flow Control with Recognition of Incoming Pause Frames and Hardware Generation of Transmitted Pause Frames
- Support for 802.1Q VLAN Tagging with Recognition of Incoming VLAN and Priority Tagged Frames
- Multiple Buffers per Receive and Transmit Frame
- Wake-on-LAN Support
- Jumbo Frames Up to 10240 bytes Supported

Downloaded from Arrow.com.

39.3 Block Diagram

Figure 39-1. EMAC Block Diagram



39.4 Functional Description

The EMAC has several clock domains:

- System bus clock (AHB and APB): DMA and register blocks
- Transmit clock: transmit block
- Receive clock: receive and address checker block

The system bus clock must run at least as fast as the receive clock and transmit clock (25 MHz at 100 Mbit/s, and 2.5 MHz at 10 Mbit/s).

The above block diagram illustrates the different blocks of the EMAC module.

The control registers drive the MDIO interface, setup up DMA activity, start frame transmission and select modes of operation such as full- or half-duplex.

The receive block checks for valid preamble, FCS, alignment and length, and presents received frames to the address checking block and DMA interface.

The transmit block takes data from the DMA interface, adds preamble and, if necessary, pad and FCS, and transmits data according to the CSMA/CD (carrier sense multiple access with collision detect) protocol. The start of transmission is deferred if CRS (carrier sense) is active.

If COL (collision) becomes active during transmission, a jam sequence is asserted and the transmission is retried after a random back off. CRS and COL have no effect in full duplex mode.

The DMA block connects to external memory through its AHB bus interface. It contains receive and transmit FIFOs for buffering frame data. It loads the transmit FIFO and empties the receive FIFO using AHB bus master operations. Receive data is not sent to memory until the address checking logic has determined that the frame should be copied. Receive or transmit frames are stored in one or more buffers. Receive buffers have a fixed length of 128 bytes.

Downloaded from Arrow.com.

Transmit buffers range in length between 0 and 2047 bytes, and up to 128 buffers are permitted per frame. The DMA block manages the transmit and receive framebuffer queues. These queues can hold multiple frames.

39.4.1 Clock

Synchronization module in the EMAC requires that the bus clock (MCK) runs at the speed of the macb_tx/rx_clk at least, which is 25 MHz at 100 Mbit/s, and 2.5 MHz at 10 Mbit/s.

39.4.2 Memory Interface

Frame data is transferred to and from the EMAC through the DMA interface. All transfers are 32-bit words and may be single accesses or bursts of 2, 3 or 4 words. Burst accesses do not cross sixteen-byte boundaries. Bursts of four words are the default data transfer; single accesses or bursts of less than four words may be used to transfer data at the beginning or the end of a buffer.

The DMA controller performs six types of operation on the bus. In order of priority, these are:

1. Receive buffer manager write
2. Receive buffer manager read
3. Transmit data DMA read
4. Receive data DMA write
5. Transmit buffer manager read
6. Transmit buffer manager write

39.4.2.1 FIFO

The FIFO depths are 128 bytes for receive and 128 bytes for transmit and are a function of the system clock speed, memory latency and network speed.

Data is typically transferred into and out of the FIFOs in bursts of four words. For receive, a bus request is asserted when the FIFO contains four words and has space for 28 more. For transmit, a bus request is generated when there is space for four words, or when there is space for 27 words if the next transfer is to be only one or two words.

Thus the bus latency must be less than the time it takes to load the FIFO and transmit or receive three words (112 bytes) of data.

At 100 Mbit/s, it takes 8960 ns to transmit or receive 112 bytes of data. In addition, six master clock cycles should be allowed for data to be loaded from the bus and to propagate through the FIFOs. For a 133 MHz master clock this takes 45 ns, making the bus latency requirement 8915 ns.

39.4.2.2 Receive Buffers

Received frames, including CRC/FCS optionally, are written to receive buffers stored in memory. Each receive buffer is 128 bytes long. The start location for each receive buffer is stored in memory in a list of receive buffer descriptors at a location pointed to by the Receive Buffer Queue Pointer Register (EMAC_RBQP). The receive buffer start location is a word address. For the first buffer of a frame, the start location can be offset by up to three bytes depending on the value written to bits 14 and 15 of the Network Configuration Register (EMAC_NCFGR). If the start location of the buffer is offset the available length of the first buffer of a frame is reduced by the corresponding number of bytes.

Each list entry consists of two words, the first being the address of the receive buffer and the second being the receive status. If the length of a receive frame exceeds the buffer length, the status word for the used buffer is written with zeroes except for the ‘Start of Frame’ bit and the offset bits, if appropriate. Bit 0 of the address field is written to one to show the buffer has been used. The receive buffer manager then reads the location of the next receive buffer and fills that with receive frame data. The final buffer descriptor status word contains the complete frame status. Refer to the table below for details of the receive buffer descriptor list.

Table 39-1. Receive Buffer Descriptor Entry

Bit	Function
Word 0	
31:2	Address of beginning of buffer
1	Wrap - marks last descriptor in receive buffer descriptor list.

Downloaded from Arrow.com.

.....continued	
Bit	**Function**
0	Ownership - needs to be zero for the EMAC to write data to the receive buffer. The EMAC sets this to one once it has successfully written a frame to memory. Software has to clear this bit before the buffer can be used again.
Word 1	
31	Global all ones broadcast address detected
30	Multicast hash match
29	Unicast hash match
28	External address match
27	Reserved for future use
26	Specific address register 1 match
25	Specific address register 2 match
24	Specific address register 3 match
23	Specific address register 4 match
22	Type ID match
21	VLAN tag detected (i.e., type ID of 0x8100)
20	Priority tag detected (i.e., type ID of 0x8100 and null VLAN identifier)
19:17	VLAN priority (only valid if bit 21 is set)
16	Concatenation format indicator (CFI) bit (only valid if bit 21 is set)
15	End of frame - when set the buffer contains the end of a frame. If end of frame is not set, then the only other valid status are bits 12, 13 and 14.
14	Start of frame - when set the buffer contains the start of a frame. If both bits 15 and 14 are set, then the buffer contains a whole frame.
13:12	Receive buffer offset - indicates the number of bytes by which the data in the first buffer is offset from the word address. Updated with the current values of the EMAC_NCFGR. If jumbo frame mode is enabled through bit 3 of the EMAC_NCFGR, then bits 13:12 of the receive buffer descriptor entry are used to indicate bits 13:12 of the frame length.
11:0	Length of frame including FCS (if selected). Bits 13:12 are also used if jumbo frame mode is selected.

To receive frames, the buffer descriptors must be initialized by writing an appropriate address to bits 31 to 2 in the first word of each list entry. Bit zero must be written with zero. Bit 1 is the wrap bit and indicates the last entry in the list.

The start location of the receive buffer descriptor list must be written to the EMAC_RBQP register before setting the ‘Receive Enable’ bit in the Network Control Register (EMAC_NCR) to enable receive. As soon as the receive block starts writing received frame data to the receive FIFO, the receive buffer manager reads the first receive buffer location pointed to by the EMAC_RBQP register.

If the filter block then indicates that the frame should be copied to memory, the receive data DMA operation starts writing data into the receive buffer. If an error occurs, the buffer is recovered. If the current buffer pointer has its wrap bit set or is the 1024th descriptor, the next receive buffer location is read from the beginning of the receive descriptor list. Otherwise, the next receive buffer location is read from the next word in memory.

There is an 11-bit counter to count out the 2048 word locations of a maximum length, receive buffer descriptor list. This is added with the value originally written to the EMAC_RBQP register to produce a pointer into the list. A read of the EMAC_RBQP register returns the pointer value, which is the queue entry currently being accessed. The counter

Downloaded from Arrow.com.

is reset after receive status is written to a descriptor that has its wrap bit set or rolls over to zero after 1024 descriptors have been accessed. The value written to the EMAC_RBQP register may be any word-aligned address, provided that there are at least 2048 word locations available between the pointer and the top of the memory.

Section 3.6 of the AMBA 2.0 specification states that bursts should not cross 1K boundaries. As receive buffer manager writes are bursts of two words, to ensure that this does not occur, it is best to write the EMAC_RBQP register with the least three significant bits set to zero. As receive buffers are used, the receive buffer manager sets bit 0 of the first word of the descriptor to indicate used. If a receive error is detected the receive buffer currently being written is recovered. Previous buffers are not recovered. Software should search through the used bits in the buffer descriptors to find out how many frames have been received. It should be checking the ‘Start of Frame’ and ‘End of Frame’ bits, and not rely on the value returned by the EMAC_RBQP register which changes continuously as more buffers are used.

For CRC errored frames, excessive length frames or length field mismatched frames, all of which are counted in the statistics registers, it is possible that a frame fragment might be stored in a sequence of receive buffers. Software can detect this by looking for ‘Start of Frame’ bit set in a buffer following a buffer with no ‘End of Frame’ bit set.

For a properly working Ethernet system, there should be no excessively long frames or frames greater than 128 bytes with CRC/FCS errors. Collision fragments are less than 128 bytes long. Therefore, it is a rare occurrence to find a frame fragment in a receive buffer.

If bit 0 is set when the receive buffer manager reads the location of the receive buffer, then the buffer has already been used and cannot be used again until software has processed the frame and cleared bit 0. In this case, the DMA block sets the ‘Buffer Not Available’ bit in the Receive Status Register (EMAC_RSR) and triggers an interrupt.

If bit 0 is set when the receive buffer manager reads the location of the receive buffer and a frame is being received, the frame is discarded and the Receive Resource Errors Register (EMAC_RRE) is incremented.

A receive overrun condition occurs when bus was not granted in time or because HRESP was not OK (bus error). In a receive overrun condition, the receive overrun interrupt is asserted and the buffer currently being written is recovered. The next frame received with an address that is recognized reuses the buffer.

If bit 17 of the EMAC_NCFGR is set, the FCS of received frames shall not be copied to memory. The frame length indicated in the receive status field shall be reduced by four bytes in this case.

39.4.2.3 Transmit Buffer

Frames to be transmitted are stored in one or more transmit buffers. Transmit buffers can be between 0 and 2047 bytes long, so it is possible to transmit frames longer than the maximum length specified in IEEE Standard 802.3. Zero length buffers are allowed. The maximum number of buffers permitted for each transmit frame is 128.

The start location for each transmit buffer is stored in memory in a list of transmit buffer descriptors at a location pointed to by the Transmit Buffer Queue Pointer Register (EMAC_TBQP). Each list entry consists of two words, the first being the byte address of the transmit buffer and the second containing the transmit control and status. Frames can be transmitted with or without automatic CRC generation. If CRC is automatically generated, pad is also automatically generated to take frames to a minimum length of 64 bytes. The table below defines an entry in the transmit buffer descriptor list. To transmit frames, the buffer descriptors must be initialized by writing an appropriate byte address to bits 31 to 0 in the first word of each list entry. The second transmit buffer descriptor is initialized with control information that indicates the length of the buffer, whether or not it is to be transmitted with CRC and whether the buffer is the last buffer in the frame.

After transmission, the control bits are written back to the second word of the first buffer along with the ‘used’ bit and other status information. Bit 31 is the ‘used’ bit which must be zero when the control word is read if transmission is to happen. It is written to one when a frame has been transmitted. Bits 27, 28 and 29 indicate various transmit error conditions. Bit 30 is the ‘wrap’ bit which can be set for any buffer within a frame. If no wrap bit is encountered after 1024 descriptors, the queue pointer rolls over to the start in a similar fashion to the receive queue.

The EMAC_TBQP register must not be written while transmit is active. If a new value is written to the EMAC_TBQP register, the queue pointer resets itself to point to the beginning of the new queue. If transmit is disabled by writing to bit 3 of the EMAC_NCR, the EMAC_TBQP register resets to point to the beginning of the transmit queue. Note that disabling receive does not have the same effect on the receive queue pointer.

Once the transmit queue is initialized, transmit is activated by writing to bit 9, the ‘Start Transmission’ bit of the EMAC_NCR. Transmit is halted when a buffer descriptor with its ‘used’ bit set is read, or if a transmit error occurs, or

Downloaded from Arrow.com.

by writing to the ‘Transmit Halt’ bit of the EMAC_NCR. (Transmission is suspended if a pause frame is received while the ‘Pause Enable’ bit is set in the EMAC_NCFGR.) Rewriting the start bit while transmission is active is allowed.

Transmission control is implemented with a Tx_go variable which is readable in the Transmit Status Register (EMAC_TSR) at bit location 3. The Tx_go variable is reset when:

- transmit is disabled
- a buffer descriptor with its ownership bit set is read
- a new value is written to the EMAC_TBQP register
- bit 10, tx_halt, of the EMAC_NCR is written
- there is a transmit error such as too many retries or a transmit underrun.

To set tx_go, write to bit 9, tx_start, of the EMAC_NCR. Transmit halt does not take effect until any ongoing transmit finishes. If a collision occurs during transmission of a multi-buffer frame, transmission automatically restarts from the first buffer of the frame. If a ‘used’ bit is read midway through transmission of a multi-buffer frame, this is treated as a transmit error. Transmission stops, tx_er is asserted and the FCS is bad.

If transmission stops due to a transmit error, the transmit queue pointer resets to point to the beginning of the transmit queue. Software needs to re-initialize the transmit queue after a transmit error.

If transmission stops due to a ‘used’ bit being read at the start of the frame, the transmission queue pointer is not reset and transmission starts from the same transmit buffer descriptor when the ‘Start Transmission’ bit is written.

Table 39-2. Transmit Buffer Descriptor Entry

Bit	Function
Word 0	
31:0	Byte Address of buffer
Word 1	
31	Used. Needs to be zero for the EMAC to read data from the transmit buffer. The EMAC sets this to one for the first buffer of a frame once it has been successfully transmitted. Software has to clear this bit before the buffer can be used again. **Note:** This bit is only set for the first buffer in a frame unlike receive where all buffers have the Used bit set once used.
30	Wrap. Marks last descriptor in transmit buffer descriptor list.
29	Retry limit exceeded, transmit error detected
28	Transmit underrun, occurs either when hresp is not OK (bus error) or the transmit data could not be fetched in time or when buffers are exhausted in mid-frame.
27	Buffers exhausted in mid-frame
26:17	Reserved
16	No CRC. When set, no CRC is appended to the current frame. This bit only needs to be set for the last buffer of a frame.
15	Last buffer. When set, this bit indicates the last buffer in the current frame has been reached.
14:11	Reserved
10:0	Length of buffer

39.4.3 Transmit Block

This block transmits frames in accordance with the Ethernet IEEE 802.3 CSMA/CD protocol. Frame assembly starts by adding preamble and the start frame delimiter. Data is taken from the transmit FIFO a word at a time. Data is transmitted least significant nibble first. If necessary, padding is added to increase the frame length to 60 bytes. CRC is calculated as a 32-bit polynomial. This is inverted and appended to the end of the frame, taking the frame length to

Downloaded from Arrow.com.

a minimum of 64 bytes. If the No CRC bit is set in the second word of the last buffer descriptor of a transmit frame, neither pad nor CRC are appended.

In full-duplex mode, frames are transmitted immediately. Back-to-back frames are transmitted at least 96 bit times apart to guarantee the interframe gap.

In half-duplex mode, the transmitter checks carrier sense. If asserted, it waits for it to de-assert and then starts transmission after the interframe gap of 96 bit times. If the collision signal is asserted during transmission, the transmitter transmits a jam sequence of 32 bits taken from the data register and then retry transmission after the back off time has elapsed.

The back-off time is based on an XOR of the 10 least significant bits of the data coming from the transmit FIFO and a 10-bit pseudo random number generator. The number of bits used depends on the number of collisions seen. After the first collision, one bit is used, after the second collision, two bits are used, and so on up to 10. Above 10, all 10 bits are used. An error is indicated and no further attempts are made if 16 attempts cause collisions.

If transmit DMA underruns, bad CRC is automatically appended using the same mechanism as jam insertion and the tx_er signal is asserted. For a properly configured system, this should never happen.

If the ‘Back Pressure’ bit is set in the EMAC_NCR in half duplex mode, the transmit block transmits 64 bits of data, which can consist of 16 nibbles of 1011 or in bit-rate mode 64 ones, whenever it sees an incoming frame to force a collision. This provides a way of implementing flow control in half-duplex mode.

39.4.4 Pause Frame Support

The following table summarizes the start of an 802.3 pause frame.

Table 39-3. Start of an 802.3 Pause Frame

Destination Address	Source Address	Type (MAC Control Frame)	Pause Opcode	Pause Time
0x0180C2000001	6 bytes	0x8808	0x0001	2 bytes

The EMAC_NCFGR contains a receive ‘Pause Enable’ bit (13). If a valid pause frame is received, the Pause Time Register (EMAC_PTR) is updated with the frame’s pause time, regardless of its current contents and regardless of the state of the EMAC_NCFGR bit 13. An interrupt (12) is triggered when a pause frame is received, assuming it is enabled in the Interrupt Mask Register (EMAC_IMR). If bit 13 is set in the EMAC_NCFGR and the value of the EMAC_PTR is non-zero, no new frame is transmitted until the EMAC_PTR has decremented to zero.

The loading of a new pause time, and hence the pausing of transmission, only occurs when the EMAC is configured for full-duplex operation. If the EMAC is configured for half-duplex, there is no transmission pause, but the pause frame received interrupt is still triggered.

A valid pause frame is defined as having a destination address that matches either the address stored in specific address register 1 or matches 0x0180C2000001 and has the MAC control frame type ID of 0x8808 and the pause opcode of 0x0001. Pause frames that have FCS or other errors are treated as invalid and are discarded. Valid pause frames received increment the Pause Frames Received Register (EMAC_PFR).

The EMAC_PTR decrements every 512 bit times (i.e., 128 rx_clks in nibble mode) once transmission has stopped. For test purposes, the register decrements every rx_clk cycle once transmission has stopped if bit 12 (‘Retry Test’) is set in the EMAC_NCFGR. If the ‘Pause Enable’ bit (13) is not set in the EMAC_NCFGR, then the decrementing occurs regardless of whether transmission has stopped or not.

An interrupt (13) is asserted whenever the EMAC_PTR decrements to zero (assuming it is enabled in the EMAC_IMR). Automatic transmission of pause frames is supported through the transmit pause frame bits of the EMAC_NCR and the tx_pause and tx_pause_zero inputs. If either bit 11 or bit 12 of the EMAC_NCR is written to with a one, or if the input signal tx_pause is toggled, a pause frame is transmitted only if full duplex is selected in the EMAC_NCFGR and transmit is enabled in the EMAC_NCR.

Pause frame transmission occurs immediately if transmit is inactive or if transmit is active between the current frame and the next frame due to be transmitted.

Transmitted pause frames comprise the following:

- A destination address of 01-80-C2-00-00-01

Downloaded from Arrow.com.

- A source address taken from the specific address 1 register
- A type ID of 88-08 (MAC control frame)
- A pause opcode of 00-01
- A pause quantum
- Fill of 00 to take the frame to minimum frame length
- Valid FCS

The pause quantum used in the generated frame depends on the trigger source for the frame as follows:

1. If bit 11 is written with a one, the pause quantum comes from the Transmit Pause Quantum Register (EMAC_TPQ). The EMAC_TPQ register resets to a value of 0xFFFF giving a maximum pause quantum as a default.
2. If bit 12 is written with a one, the pause quantum is zero.
3. If the tx_pause input is toggled and the tx_pause_zero input is held low until the next toggle, the pause quantum comes from the EMAC_TPQ register.
4. If the tx_pause input is toggled and the tx_pause_zero input is held high until the next toggle, the pause quantum is zero.

After transmission, no interrupts are generated and the only statistics register that is incremented is the Transmitted Pause Frames Register (EMAC_TPF).

39.4.5 Receive Block

The receive block checks for valid preamble, FCS, alignment and length, presents received frames to the DMA block and stores the frames destination address for use by the address checking block. If, during frame reception, the frame is found to be too long or rx_er is asserted, a bad frame indication is sent to the DMA block. The DMA block then ceases sending data to memory. At the end of frame reception, the receive block indicates to the DMA block whether the frame is good or bad. The DMA block recovers the current receive buffer if the frame was bad. The receive block signals the register block to increment the alignment error, the CRC (FCS) error, the short frame, long frame, jabber error, the receive symbol error statistics and the length field mismatch statistics.

The enable bit for jumbo frames in the EMAC_NCFGR allows the EMAC to receive jumbo frames of up to 10240 bytes in size. This operation does not form part of the IEEE802.3 specification and is disabled by default. When jumbo frames are enabled, frames received with a frame size greater than 10240 bytes are discarded.

39.4.6 Address Checking Block

The address checking (or filter) block indicates to the DMA block which receive frames should be copied to memory. Whether a frame is copied depends on what is enabled in the EMAC_NCFGR, the state of the external match pin, the contents of the specific address and hash registers and the frame's destination address. In this implementation of the EMAC, the frame's source address is not checked. Provided that bit 18 of the EMAC_NCFGR is not set, a frame is not copied to memory if the EMAC is transmitting in half duplex mode at the time a destination address is received. If bit 18 of the EMAC_NCFGR is set, frames can be received while transmitting in half-duplex mode.

Ethernet frames are transmitted a byte at a time, least significant bit first. The first six bytes (48 bits) of an Ethernet frame make up the destination address. The first bit of the destination address, the LSB of the first byte of the frame, is the group/individual bit: this is One for multicast addresses and Zero for unicast. The All Ones address is the broadcast address, and a special case of multicast.

The EMAC supports recognition of four specific addresses. Each specific address requires two registers, specific address register bottom and specific address register top. Specific address register bottom stores the first four bytes of the destination address and specific address register top contains the last two bytes. The addresses stored can be specific, group, local or universal.

The destination address of received frames is compared against the data stored in the specific address registers once they have been activated. The addresses are deactivated at reset or when their corresponding specific address register bottom is written. They are activated when specific address register top is written. If a receive frame address matches an active address, the frame is copied to memory.

The following example illustrates the use of the address match registers for a MAC address of 21:43:65:87:A9:CB.

Preamble 55

SFD D5

Downloaded from Arrow.com.

DA (Octet0 - LSB) 21

DA (Octet 1) 43

DA (Octet 2) 65

DA (Octet 3) 87

DA (Octet 4) A9

DA (Octet5 - MSB) CB

SA (LSB) 00

SA 00

SA 00

SA 00

SA 00

SA (MSB) 43

SA (LSB) 21

The sequence above shows the beginning of an Ethernet frame. Byte order of transmission is from top to bottom as shown. For a successful match to specific address 1, the following address matching registers must be set up:

- Base address + 0x98 0x87654321 (Bottom)
- Base address + 0x9C 0x0000CBA9 (Top)

And for a successful match to the Type ID Checking Register (EMAC_TID), the following should be set up:

- Base address + 0xB8 0x00004321

39.4.7 Broadcast Address

The broadcast address of 0xFFFFFFFFFFFF is recognized if the 'No Broadcast' bit in the EMAC_NCFGR is zero.

39.4.8 Hash Addressing

The hash address register is 64 bits long and takes up two locations in the memory map. The least significant bits are stored in hash register bottom and the most significant bits in hash register top.

The 'Unicast Hash Enable' and the 'Multicast Hash Enable' bits in the EMAC_NCFGR enable the reception of hash matched frames. The destination address is reduced to a 6-bit index into the 64-bit hash register using the following hash function. The hash function is an exclusive or of every sixth bit of the destination address.

```
hash_index[5] = da[5] ^ da[11] ^ da[17] ^ da[23] ^ da[29] ^ da[35] ^ da[41] ^ da[47]
hash_index[4] = da[4] ^ da[10] ^ da[16] ^ da[22] ^ da[28] ^ da[34] ^ da[40] ^ da[46]
hash_index[3] = da[3] ^ da[09] ^ da[15] ^ da[21] ^ da[27] ^ da[33] ^ da[39] ^ da[45]
hash_index[2] = da[2] ^ da[08] ^ da[14] ^ da[20] ^ da[26] ^ da[32] ^ da[38] ^ da[44]
hash_index[1] = da[1] ^ da[07] ^ da[13] ^ da[19] ^ da[25] ^ da[31] ^ da[37] ^ da[43]
hash_index[0] = da[0] ^ da[06] ^ da[12] ^ da[18] ^ da[24] ^ da[30] ^ da[36] ^ da[42]
```

da[0] represents the least significant bit of the first byte received, that is, the multicast/unicast indicator, and da[47] represents the most significant bit of the last byte received.

If the hash index points to a bit that is set in the hash register, then the frame is matched according to whether the frame is multicast or unicast.

A multicast match is signalled if the 'Multicast Hash Enable' bit is set. da[0] is 1 and the hash index points to a bit set in the hash register.

A unicast match is signalled if the 'Unicast Hash Enable' bit is set. da[0] is 0 and the hash index points to a bit set in the hash register.

To receive all multicast frames, the hash register should be set with all ones and the 'Multicast Hash Enable' bit should be set in the EMAC_NCFGR.

Downloaded from Arrow.com.

39.4.9 External Address Matching

The external address signal (eam) is enabled by bit 9 in the EMAC_NCFGR. When enabled, the filter block sends the store frame and the external address match status signal to the DMA block if the external address match signal is asserted (from a source external to the EMAC) and the destination address has been received and the frame has not completed.

For the DMA block to be able to copy the frame to memory, the external address signal must be asserted before four words have been loaded into the receive FIFO.

39.4.10 Copy All Frames (or Promiscuous Mode)

If the 'Copy All Frames' bit is set in the EMAC_NCFGR, then all non-errored frames are copied to memory. For example, frames that are too long, too short, or have FCS errors or rx_er asserted during reception are discarded and all others are received. Frames with FCS errors are copied to memory if bit 19 in the EMAC_NCFGR is set.

39.4.11 Type ID Checking

The contents of the EMAC_TID register are compared against the length/type ID of received frames (i.e., bytes 13 and 14). Bit 22 in the receive buffer descriptor status is set if there is a match. The reset state of this register is zero which is unlikely to match the length/type ID of any valid Ethernet frame.

Note: A type ID match does not affect whether a frame is copied to memory.

39.4.12 VLAN Support

The following table describes an Ethernet encoded 802.1Q VLAN tag.

Table 39-4. 802.1Q VLAN Tag

TPID (Tag Protocol Identifier) 16 bits	TCI (Tag Control Information) 16 bits
0x8100	First 3 bits priority, then CFI bit, last 12 bits VID

The VLAN tag is inserted at the 13th byte of the frame, adding an extra four bytes to the frame. If the VID (VLAN identifier) is null (0x000), this indicates a priority-tagged frame. The MAC can support frame lengths up to 1536 bytes, 18 bytes more than the original Ethernet maximum frame length of 1518 bytes. This is achieved by setting bit 8 in the EMAC_NCFGR.

The following bits in the receive buffer descriptor status word give information about VLAN tagged frames:

- Bit 21 set if receive frame is VLAN tagged (i.e., type ID of 0x8100)
- Bit 20 set if receive frame is priority tagged (i.e., type ID of 0x8100 and null VID). (If bit 20 is set, bit 21 is set also.)
- Bit 19, 18 and 17 set to priority if bit 21 is set
- Bit 16 set to CFI if bit 21 is set

39.4.13 Wake-on-LAN Support

The receive block supports Wake-on-LAN by detecting the following events on incoming receive frames:

- Magic packet
- ARP request to the device IP address
- Specific address 1 filter match
- Multicast hash filter match

If one of these events occurs Wake-on-LAN detection is indicated by asserting the wol output pin for 64 rx_clk cycles. These events can be individually enabled through bits 19:16 of the Wake-on-LAN Register (EMAC_WOL). Also, for Wake-on-LAN detection to occur, receive enable must be set in the EMAC_NCR, however a receive buffer does not have to be available. wol assertion due to ARP request, specific address 1 or multicast filter events occurs even if the frame is errored. For magic packet events, the frame must be correctly formed and error free.

A magic packet event is detected if all of the following are true:

- magic packet events are enabled through bit 16 of the EMAC_WOL register

Downloaded from Arrow.com.

- the frame's destination address matches specific address 1
- the frame is correctly formed with no errors
- the frame contains at least 6 bytes of 0xFF for synchronization
- there are 16 repetitions of the contents of specific address 1 register immediately following the synchronization

An ARP request event is detected if all of the following are true:

- ARP request events are enabled through bit 17 of the EMAC_WOL register
- broadcasts are allowed by bit 5 in the EMAC_NCFGR
- the frame has a broadcast destination address (bytes 1 to 6)
- the frame has a type ID field of 0x0806 (bytes 13 and 14)
- the frame has an ARP operation field of 0x0001 (bytes 21 and 22)
- the least significant 16 bits of the frame's ARP target protocol address (bytes 41 and 42) match the value programmed in bits 15:0 of the EMAC_WOL register

The decoding of the ARP fields adjusts automatically if a VLAN tag is detected within the frame. The reserved value of 0x0000 for the Wake-on-LAN target address value does not cause an ARP request event, even if matched by the frame.

A specific address 1 filter match event occurs if all of the following are true:

- specific address 1 events are enabled through bit 18 of the EMAC_WOL register
- the frame's destination address matches the value programmed in the specific address 1 registers

A multicast filter match event occurs if all of the following are true:

- multicast hash events are enabled through bit 19 of the EMAC_WOL register
- multicast hash filtering is enabled through bit 6 of the EMAC_NCFGR
- the frame's destination address matches against the multicast hash filter
- the frame's destination address is not a broadcast

39.4.14 PHY Maintenance

The PHY Maintenance Register (EMAC_MAN) enables the EMAC to communicate with a PHY by means of the MDIO interface. It is used during auto-negotiation to ensure that the EMAC and the PHY are configured for the same speed and duplex configuration.

The EMAC_MAN register is implemented as a shift register. Writing to the register starts a shift operation which is signalled as complete when bit 2 is set in the Network Status Register (EMAC_NSR) (about 2000 MCK cycles later when bit 10 is set to zero, and bit 11 is set to one in the EMAC_NCFGR). An interrupt is generated as this bit is set. During this time, the MSB of the register is output on the MDIO pin and the LSB updated from the MDIO pin with each MDC cycle. This causes transmission of a PHY management frame on MDIO.

Reading during the shift operation returns the current contents of the shift register. At the end of management operation, the bits have shifted back to their original locations. For a read operation, the data bits are updated with data read from the PHY. It is important to write the correct values to the register to ensure a valid PHY management frame is produced.

The MDIO interface can read IEEE 802.3 clause 45 PHYs as well as clause 22 PHYs. To read clause 45 PHYs, bits 31:28 should be written as 0x0011. To write clause 45 PHYs, bits 31:28 should be written as 0x0001. See the table below.

Table 39-5. Clause 22/Clause 45 PHYs Read/Write Access Configuration

PHY	Access	Field Configuration (EMAC_MAN Bits 31:28)	
		SOF[1:0]	RW[1:0]
Clause 22	Read	01	10
	Write	01	01

Downloaded from Arrow.com.

.....continued			
PHY	Access	Field Configuration (EMAC_MAN Bits 31:28)	
		SOF[1:0]	RW[1:0]
Clause 45	Read	00	11
	Write	00	01
	Read + Address	00	10

For a description of MDC generation, see Network Configuration Register.

39.4.15 Physical Interface

Depending on products, the Ethernet MAC is capable of interfacing to RMII or MII Interface. The 'RMII' bit in the User Input/Output Register (EMAC_USRIO) controls the interface that is selected. When this bit is set, the RMII interface is selected, else the MII interface is selected.

The MII and RMII interfaces are capable of both 10 Mbit/s and 100 Mbit/s data rates as described in the IEEE 802.3u standard. The signals used by the MII and RMII interfaces are described in the table below.

Table 39-6. Pin Configuration

Pin Name	MII	RMII
ETXCK_EREFCK	ETXCK: Transmit Clock	EREFCK: Reference Clock
ECRS	ECRS: Carrier Sense	–
ECOL	ECOL: Collision Detect	–
ERXDV	ERXDV: Data Valid	ECRSDV: Carrier Sense/Data Valid
ERX0–ERX3	ERX0–ERX3: 4-bit Receive Data	ERX0–ERX1: 2-bit Receive Data
ERXER	ERXER: Receive Error	ERXER: Receive Error
ERXCK	ERXCK: Receive Clock	–
ETXEN	ETXEN: Transmit Enable	ETXEN: Transmit Enable
ETX0–ETX3	ETX0–ETX3: 4-bit Transmit Data	ETX0–ETX1: 2-bit Transmit Data
ETXER	ETXER: Transmit Error	–
EMDC	Management Data Clock	Management Data Clock
EMDIO	Management Data Input Output	Management Data Input Output

The RMII provides a reduced pin count alternative to the IEEE 802.3u MII. It uses two bits for transmit (ETX0 and ETX1) and two bits for receive (ERX0 and ERX1). There is a Transmit Enable (ETXEN), a Receive Error (ERXER), a Carrier Sense (ECRS_DV), and a 50 MHz Reference Clock (ETXCK_EREFCK) for 100 Mbit/s data rate.

39.4.15.1 RMII Transmit and Receive Operation

The same signals are used internally for both the RMII and the MII operations. The RMII maps the signals in a more pin-efficient manner. The transmit and receive bits are converted from a 4-bit parallel format to a 2-bit parallel scheme that is clocked at twice the rate. The carrier sense and data valid signals are combined into the ECRSDV signal. This signal contains information on carrier sense, FIFO status, and validity of the data. Transmit error bit (ETXER) and collision detect (ECOL) are not used in RMII mode.

Downloaded from Arrow.com.

39.5 Programming Interface

39.5.1 Initialization

39.5.1.1 Configuration

Initialization of the EMAC configuration (e.g., loop-back mode, frequency ratios) must be done while the transmit and receive circuits are disabled. See the description of the EMAC_NCR and EMAC_NCFGR earlier in this document.

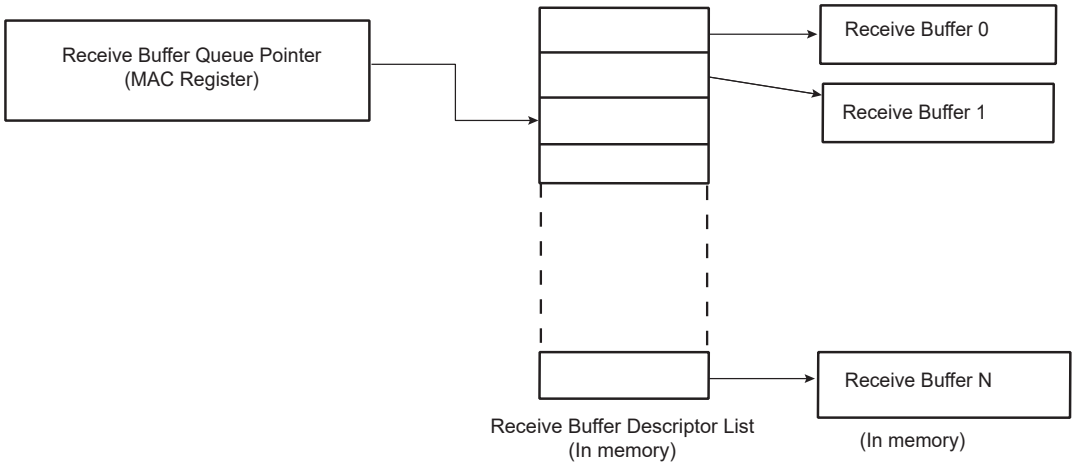
To change loop-back mode, the following sequence of operations must be followed:

1. Write to EMAC_NCR to disable transmit and receive circuits.
2. Write to EMAC_NCR to change loop-back mode.
3. Write to EMAC_NCR to re-enable transmit or receive circuits.
   Note: These writes to EMAC_NCR cannot be combined in any way.

39.5.1.2 Receive Buffer List

Receive data is written to areas of data (i.e., buffers) in system memory. These buffers are listed in another data structure that also resides in main memory. This data structure (receive buffer queue) is a sequence of descriptor entries as defined in Receive Buffer Descriptor Entry. It points to this data structure.

Figure 39-2. Receive Buffer List



To create the list of buffers:

1. Allocate a number (n) of buffers of 128 bytes in system memory.
2. Allocate an area 2n words for the receive buffer descriptor entry in system memory and create n entries in this list. Mark all entries in this list as owned by EMAC, i.e., bit 0 of word 0 set to zero.
3. If less than 1024 buffers are defined, the last descriptor must be marked with the wrap bit (bit 1 in word 0 set to one).
4. Write address of receive buffer descriptor entry to the EMAC_RBQP register.
5. The receive circuits can then be enabled by writing to the address recognition registers and then to the EMAC_NCR.

39.5.1.3 Transmit Buffer List

Transmit data is read from areas of data (the buffers) in system memory These buffers are listed in another data structure that also resides in main memory. This data structure (Transmit Buffer Queue) is a sequence of descriptor entries (as defined in Transmit Buffer Description Entry) that points to this data structure.

To create this list of buffers:

1. Allocate a number (n) of buffers of between 1 and 2047 bytes of data to be transmitted in system memory. Up to 128 buffers per frame are allowed.
2. Allocate an area 2n words for the transmit buffer descriptor entry in system memory and create N entries in this list. Mark all entries in this list as owned by EMAC, i.e., bit 31 of word 1 set to zero.

Downloaded from Arrow.com.

3. If fewer than 1024 buffers are defined, the last descriptor must be marked with the wrap bit — bit 30 in word 1 set to one.
4. Write address of transmit buffer descriptor entry to EMAC_TBQP register.
5. The transmit circuits can then be enabled by writing to the EMAC_NCR.

39.5.1.4 Address Matching

The EMAC register-pair hash address and the four specific address register-pairs must be written with the required values. Each register-pair comprises a bottom register and top register, with the bottom register being written first. The address matching is disabled for a particular register-pair after the bottom-register has been written and re-enabled when the top register is written. See Address Checking Block for details of address matching. Each register-pair may be written at any time, regardless of whether the receive circuits are enabled or disabled.

39.5.1.5 Interrupts

There are 15 interrupt conditions that are detected within the EMAC. These are ORed to make a single interrupt. Depending on the overall system design, this may be passed through a further level of interrupt collection (interrupt controller). On receipt of the interrupt signal, the CPU enters the interrupt handler (Refer to the Interrupt Controller). To ascertain which interrupt has been generated, read the Interrupt Status Register (EMAC_ISR). Note that this register clears itself when read. At reset, all interrupts are disabled.

To enable an interrupt, write to the Interrupt Enable Register (EMAC_IER) with the pertinent interrupt bit set to one.

To disable an interrupt, write to the Interrupt Disable Register (EMAC_IDR) with the pertinent interrupt bit set to one.

To check whether an interrupt is enabled or disabled, read the EMAC_IMR; if the bit is set to one, the interrupt is disabled.

39.5.1.6 Transmitting Frames

To set up a frame for transmission:

1. Enable transmit in the EMAC_NCR.
2. Allocate an area of system memory for transmit data. This does not have to be contiguous, varying byte lengths can be used as long as they conclude on byte borders.
3. Set-up the transmit buffer list.
4. Set the EMAC_NCR to enable transmission and enable interrupts.
5. Write data for transmission into these buffers.
6. Write the address to transmit buffer descriptor queue pointer.
7. Write control and length to word one of the transmit buffer descriptor entry.
8. Write to the ‘Start Transmission’ bit in the EMAC_NCR.

39.5.1.7 Receiving Frames

When a frame is received and the receive circuits are enabled, the EMAC checks the address and, in the following cases, the frame is written to system memory:

- if it matches one of the four specific address registers.
- if it matches the hash address function.
- if it is a broadcast address (0xFFFFFFFFFFFF) and broadcasts are allowed.
- if the EMAC is configured to copy all frames.
- if the EAM is asserted before four words have been loaded into the receive FIFO.

The EMAC_RBQP register points to the next entry (see Receive Buffer Descriptor Entry) and the EMAC uses this as the address in system memory to write the frame to. Once the frame has been completely and successfully received and written to system memory, the EMAC then updates the receive buffer descriptor entry with the reason for the address match and marks the area as being owned by software. Once this is complete an interrupt receive complete is set. Software is then responsible for handling the data in the buffer and then releasing the buffer by writing the ownership bit back to zero.

If the EMAC is unable to write the data at a rate to match the incoming frame, then an interrupt receive overrun is set. If there is no receive buffer available, i.e., the next buffer is still owned by software, the interrupt receive buffer not available is set. If the frame is not successfully received, a statistics register is incremented and the frame is discarded without informing software.

Downloaded from Arrow.com.

39.6 Register Summary

Offset	Name	Bit Pos.								
0x00	EMAC_NCR	31:24								
		23:16								
		15:8				TZQ	TPFR	THALT	TSTART	BP
		7:0	WESTAT	INCSTAT	CLRSTAT	MPE	TE	RE	LLB	LB
0x04	EMAC_NCFGR	31:24								
		23:16					IRXFCS	EFRHD	DRFCS	RLCE
		15:8	RBOF[1:0]		PAE	RTY	CLK[1:0]		EAE	BIG
		7:0	UNI	MTI	NBC	CAF	JFRAME		FD	SPD
0x08	EMAC_NSR	31:24								
		23:16								
		15:8								
		7:0						IDLE	MDIO	LINKR
0x0C ... 0x13	Reserved									
0x14	EMAC_TSR	31:24								
		23:16								
		15:8								
		7:0		UND	COMP	BEX	TGO	RLES	COL	UBR
0x18	EMAC_RBQP	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1C	EMAC_TBQP	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x20	EMAC_RSR	31:24								
		23:16								
		15:8								
		7:0						OVR	REC	BNA
0x24	EMAC_ISR	31:24								
		23:16								
		15:8		WOL	PTZ	PFRE	HRESP	ROVR	LINK	
		7:0	TCOMP	TXERR	RLEX	TUND	TXUBR	RXUBR	RCOMP	MFD
0x28	EMAC_IER	31:24								
		23:16								
		15:8		WOL	PTZ	PFR	HRESP	ROVR	LINK	
		7:0	TCOMP	TXERR	RLE	TUND	TXUBR	RXUBR	RCOMP	MFD
0x2C	EMAC_IDR	31:24								
		23:16								
		15:8		WOL	PTZ	PFR	HRESP	ROVR	LINK	
		7:0	TCOMP	TXERR	RLE	TUND	TXUBR	RXUBR	RCOMP	MFD
0x30	EMAC_IMR	31:24								
		23:16								
		15:8		WOL	PTZ	PFR	HRESP	ROVR	LINK	
		7:0	TCOMP	TXERR	RLE	TUND	TXUBR	RXUBR	RCOMP	MFD
0x34	EMAC_MAN	31:24	SOF[1:0]		RW[1:0]		PHYA[4:1]			
		23:16	PHYA[0]	REGA[4:0]					CODE[1:0]	
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x38	EMAC_PTR	31:24								
		23:16								
		15:8	PTIME[15:8]							
		7:0	PTIME[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x3C	EMAC_PFR	31:24								
		23:16								
		15:8	FROK[15:8]							
		7:0	FROK[7:0]							
0x40	EMAC_FTO	31:24								
		23:16	FTOK[23:16]							
		15:8	FTOK[15:8]							
		7:0	FTOK[7:0]							
0x44	EMAC_SCF	31:24								
		23:16								
		15:8	SCF[15:8]							
		7:0	SCF[7:0]							
0x48	EMAC_MCF	31:24								
		23:16								
		15:8	MCF[15:8]							
		7:0	MCF[7:0]							
0x4C	EMAC_FRO	31:24								
		23:16	FROK[23:16]							
		15:8	FROK[15:8]							
		7:0	FROK[7:0]							
0x50	EMAC_FCSE	31:24								
		23:16								
		15:8								
		7:0	FCSE[7:0]							
0x54	EMAC_ALE	31:24								
		23:16								
		15:8								
		7:0	ALE[7:0]							
0x58	EMAC_DTF	31:24								
		23:16								
		15:8	DTF[15:8]							
		7:0	DTF[7:0]							
0x5C	EMAC_LCOL	31:24								
		23:16								
		15:8								
		7:0	LCOL[7:0]							
0x60	EMAC_ECOL	31:24								
		23:16								
		15:8								
		7:0	EXCOL[7:0]							
0x64	EMAC_TUND	31:24								
		23:16								
		15:8								
		7:0	TUND[7:0]							
0x68	EMAC_CSE	31:24								
		23:16								
		15:8								
		7:0	CSE[7:0]							
0x6C	EMAC_RRE	31:24								
		23:16								
		15:8	RRE[15:8]							
		7:0	RRE[7:0]							
0x70	EMAC_ROV	31:24								
		23:16								
		15:8								
		7:0	ROVR[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x74	EMAC_RSE	31:24								
		23:16								
		15:8								
		7:0	RSE[7:0]							
0x78	EMAC_ELE	31:24								
		23:16								
		15:8								
		7:0	EXL[7:0]							
0x7C	EMAC_RJA	31:24								
		23:16								
		15:8								
		7:0	RJB[7:0]							
0x80	EMAC_USF	31:24								
		23:16								
		15:8								
		7:0	USF[7:0]							
0x84	EMAC_STE	31:24								
		23:16								
		15:8								
		7:0	SQER[7:0]							
0x88	EMAC_RLE	31:24								
		23:16								
		15:8								
		7:0	RLFM[7:0]							
0x8C	EMAC_TPF	31:24								
		23:16								
		15:8	TPF[15:8]							
		7:0	TPF[7:0]							
0x90	EMAC_HRB	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x94	EMAC_HRT	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x98	EMAC_SA1B	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x9C	EMAC_SA1T	31:24								
		23:16								
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0xA0	EMAC_SA2B	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0xA4	EMAC_SA2T	31:24								
		23:16								
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0xA8	EMAC_SA3B	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0xAC	EMAC_SA3T	31:24								
		23:16								
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0xB0	EMAC_SA4B	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0xB4	EMAC_SA4T	31:24								
		23:16								
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0xB8	EMAC_TID	31:24								
		23:16								
		15:8	TID[15:8]							
		7:0	TID[7:0]							
0xBC	EMAC_TPQ	31:24								
		23:16								
		15:8	TPQ[15:8]							
		7:0	TPQ[7:0]							
0xC0	EMAC_USRIO	31:24								
		23:16								
		15:8								
		7:0							CLKEN	RMII
0xC4	EMAC_WOL	31:24								
		23:16					MTI	SA1	ARP	MAG
		15:8	IP[15:8]							
		7:0	IP[7:0]							

Downloaded from Arrow.com.

39.6.1 Network Control Register

Name: EMAC_NCR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				TZQ	TPFR	THALT	TSTART	BP
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WESTAT	INCSTAT	CLRSTAT	MPE	TE	RE	LLB	LB
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 12 – TZQ Transmit Zero Quantum Pause Frame
Writing a one to this bit transmits a pause frame with zero pause quantum at the next available transmitter idle time.

Bit 11 – TPFR Transmit Pause Frame
Writing one to this bit transmits a pause frame with the pause quantum from the EMAC_TPQ register at the next available transmitter idle time.

Bit 10 – THALT Transmit Halt
Writing one to this bit halts transmission as soon as any ongoing frame transmission ends.

Bit 9 – TSTART Start Transmission
Writing one to this bit starts transmission.

Bit 8 – BP Back Pressure
If set in half duplex mode, forces collisions on all received frames.

Bit 7 – WESTAT Write Enable for Statistics Registers
Setting this bit to one makes the statistics registers writable for functional test purposes.

Bit 6 – INCSTAT Increment Statistics Registers
This bit is write only. Writing a one increments all the statistics registers by one for test purposes.

Bit 5 – CLRSTAT Clear Statistics Registers
This bit is write only. Writing a one clears the statistics registers.

Bit 4 – MPE Management Port Enable

Value	Description
0	Forces MDIO to high impedance state and MDC low
1	Enables the management port

Downloaded from Arrow.com.

Bit 3 – TE Transmit Enable
When set, enables the Ethernet transmitter to send data. When reset transmission, stops immediately, the transmit FIFO and control registers are cleared and the EMAC_TBQP register resets to point to the start of the transmit descriptor list.

Bit 2 – RE Receive Enable
When set, enables the EMAC to receive data. When reset, frame reception stops immediately and the receive FIFO is cleared. The EMAC_RBQP register is unaffected.

Bit 1 – LLB LoopBack Local
Connects txd to rxd, tx_en to rx_dv, forces full duplex and drives rx_clk and tx_clk with MCK divided by 4. rx_clk and tx_clk may glitch as the EMAC is switched into and out of internal loop back. It is important that receive and transmit circuits have already been disabled when making the switch into and out of internal loop back.

Bit 0 – LB LoopBack
Asserts the loopback signal to the PHY.

Downloaded from Arrow.com.

39.6.2 Network Configuration Register

Name: EMAC_NCFGR
Offset: 0x04
Reset: 0x00000800
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					IRXFCS	EFRHD	DRFCS	RLCE
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RBOF[1:0]		PAE	RTY	CLK[1:0]		EAE	BIG
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	0	0

Bit	7	6	5	4	3	2	1	0
	UNI	MTI	NBC	CAF	JFRAME		FD	SPD
Access	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Reset	0	0	0	0	0		0	0

Bit 19 – IRXFCS Ignore RX FCS

Value	Description
0	Normal operation
1	Frames with FCS/CRC errors are not rejected and no FCS error statistics are counted.

Bit 18 – EFRHD Enable Frames Received in Half Duplex
Enable Frames to be received in half-duplex mode while transmitting.

Bit 17 – DRFCS Discard Receive FCS
When set, the FCS field of received frames is not copied to memory.

Bit 16 – RLCE Receive Length Field Checking Enable
When set, frames with measured lengths shorter than their length fields are discarded. Frames containing a type ID in bytes 13 and 14 — length/type ID = 0600 — are not counted as length errors.

Bits 15:14 – RBOF[1:0] Receive Buffer Offset
Indicates the number of bytes by which the received data is offset from the start of the first receive buffer.

Value	Name	Description
0	OFFSET_0	No offset from start of receive buffer
1	OFFSET_1	One-byte offset from start of receive buffer
2	OFFSET_2	Two-byte offset from start of receive buffer
3	OFFSET_3	Three-byte offset from start of receive buffer

Bit 13 – PAE Pause Enable
When set, transmission pauses when a valid pause frame is received.

Bit 12 – RTY Retry Test

Value	Description
0	Normal operation

Downloaded from Arrow.com.

Value	Description
1	The back off between collisions is always one slot time. Setting this bit helps in testing the ‘too many retries’ condition. Also used in the pause frame tests to reduce the pause counters decrement time from 512 bit times, to every rx_clk cycle.

Bits 11:10 – CLK[1:0] MDC Clock Divider
Set according to system clock speed. This determines by what number system clock is divided to generate MDC. For conformance with 802.3, MDC must not exceed 2.5 MHz (MDC is only active during MDIO read and write operations).

Value	Name	Description
0	MCK_8	MCK divided by 8 (MCK up to 20 MHz)
1	MCK_16	MCK divided by 16 (MCK up to 40 MHz)
2	MCK_32	MCK divided by 32 (MCK up to 80 MHz)
3	MCK_64	MCK divided by 64 (MCK up to 160 MHz)

Bit 9 – EAE External Address Match Enable
When set, the eam pin can be used to copy frames to memory.

Bit 8 – BIG Receive 1536 Bytes Frames
Setting this bit means the EMAC receives frames up to 1536 bytes in length. Normally, the EMAC would reject any frame above 1518 bytes.

Bit 7 – UNI Unicast Hash Enable
When set to one, unicast frames are received when the 6-bit hash function of the destination address points to a bit that is set in the hash register.

Bit 6 – MTI Multicast Hash Enable
When set to one, multicast frames are received when the 6-bit hash function of the destination address points to a bit that is set in the hash register.

Bit 5 – NBC No Broadcast
When set to one, frames addressed to the broadcast address of all ones are not received.

Bit 4 – CAF Copy All Frames
When set to one, all valid frames are received.

Bit 3 – JFRAME Jumbo Frames
Set to one to enable jumbo frames of up to 10240 bytes to be accepted.

Bit 1 – FD Full Duplex
If set to one, the transmit block ignores the state of collision and carrier sense and allows receive while transmitting.

Bit 0 – SPD Speed

Value	Description
0	10 Mbit/s operation
1	100 Mbit/s operation

Downloaded from Arrow.com.

39.6.3 Network Status Register

Name: EMAC_NSR
Offset: 0x08
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						IDLE	MDIO	LINKR
Access						R	R	R
Reset						0	x	0

Bit 2 – IDLE PHY Management Logic Status

Value	Description
0	The PHY logic is running.
1	The PHY management logic is idle (i.e., has completed).

Bit 1 – MDIO MDIO Input Status
Returns status of the EMDIO pin. Use the PHY Maintenance Register to read managed frames rather than this bit.

Bit 0 – LINKR Link Pin Status
Returns status of link pin.

Downloaded from Arrow.com.

39.6.4 Transmit Status Register

Name: EMAC_TSR
Offset: 0x14
Reset: 0x00000000
Property: Read/Write

This register, when read, provides details of the status of a transmit. Once read, individual bits may be cleared by writing a one to them. It is not possible to set a bit to one by writing to the register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		UND	COMP	BEX	TGO	RLES	COL	UBR
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 6 – UND Transmit Underrun (cleared by writing a one to this bit)
Set when transmit DMA was not able to read data from memory, either because the bus was not granted in time, because a not OK hresp(bus error) was returned or because a used bit was read midway through frame transmission. If this occurs, the transmitter forces bad CRC.

Bit 5 – COMP Transmit Complete (cleared by writing a one to this bit)
Set when a frame has been transmitted.

Bit 4 – BEX Buffers Exhausted Mid-frame (cleared by writing a one to this bit)
If the buffers run out during transmission of a frame, then transmission stops, FCS shall be bad and tx_er asserted.

Bit 3 – TGO Transmit Go
If high transmit is active.

Bit 2 – RLES Retry Limit Exceeded (cleared by writing a one to this bit)

Bit 1 – COL Collision Occurred (cleared by writing a one to this bit)
Set by the assertion of collision.

Bit 0 – UBR Used Bit Read (cleared by writing a one to this bit)
Set when a transmit buffer descriptor is read with its used bit set.

Downloaded from Arrow.com.

39.6.5 Receive Buffer Queue Pointer Register

Name: EMAC_RBQP
Offset: 0x18
Reset: 0x00000000
Property: Read/Write

This register points to the entry in the receive buffer queue (descriptor list) currently being used. It is written with the start location of the receive buffer descriptor list. The lower order bits increment as buffers are used up and wrap to their original values after either 1024 buffers or when the wrap bit of the entry is set.

Reading this register returns the location of the descriptor currently being accessed. This value increments as buffers are used. Software should not use this register for determining where to remove received frames from the queue as it constantly changes as new frames are received. Software should instead work its way through the buffer descriptor queue checking the used bits.

Receive buffer writes also comprise bursts of two words and, as with transmit buffer reads, it is recommended that bit 2 is always written with zero to prevent a burst crossing a 1K boundary, in violation of section 3.6 of the AMBA specification.

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Receive Buffer Queue Pointer Address
Written with the address of the start of the receive queue, reads as a pointer to the current buffer being used. ADDR[1:0] must be 0.

Downloaded from Arrow.com.

39.6.6 Transmit Buffer Queue Pointer Register

Name: EMAC_TBQP
Offset: 0x1C
Reset: 0x00000000
Property: Read/Write

This register points to the entry in the transmit buffer queue (descriptor list) currently being used. It is written with the start location of the transmit buffer descriptor list. The lower order bits increment as buffers are used up and wrap to their original values after either 1024 buffers or when the wrap bit of the entry is set. This register can only be written when bit 3 in the EMAC_TSR is low.

As transmit buffer reads consist of bursts of two words, it is recommended that bit 2 is always written with zero to prevent a burst crossing a 1K boundary, in violation of section 3.6 of the AMBA specification.

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Transmit Buffer Queue Pointer Address
Written with the address of the start of the transmit queue, reads as a pointer to the first buffer of the frame being transmitted or about to be transmitted.
ADDR[1:0] must be 0.

Downloaded from Arrow.com.

39.6.7 Receive Status Register

Name: EMAC_RSR
Offset: 0x20
Reset: 0x00000000
Property: Read/Write

This register, when read, provides details of the status of a receive. Once read, individual bits may be cleared by writing a one to them. It is not possible to set a bit to one by writing to the register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						OVR	REC	BNA
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – OVR Receive Overrun (cleared by writing a one to this bit)
The DMA block was unable to store the receive frame to memory, either because the bus was not granted in time or because a not OK hresp(bus error) was returned. The buffer is recovered if this happens.

Bit 1 – REC Frame Received (cleared by writing a one to this bit)
One or more frames have been received and placed in memory.

Bit 0 – BNA Buffer Not Available (cleared by writing a one to this bit)
An attempt was made to get a new buffer and the pointer indicated that it was owned by the processor. The DMA rereads the pointer each time a new frame starts until a valid pointer is found. This bit is set at each attempt that fails even if it has not had a successful pointer read since it has been cleared.

Downloaded from Arrow.com.

39.6.8 Interrupt Status Register

Name: EMAC_ISR
Offset: 0x24
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		WOL	PTZ	PFRE	HRESP	ROVR	LINK	
Access		R/W	R/W	R/W	R/W	R/W	R/W	
Reset		0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0
	TCOMP	TXERR	RLEX	TUND	TXUBR	RXUBR	RCOMP	MFD
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 14 – WOL Wake-On-LAN (cleared on read)
Set when a WOL event has been triggered (This flag can be set even if the EMAC is not clocked).

Bit 13 – PTZ Pause Time Zero (cleared on read)
Set when the EMAC_PTR, 0x38 decrements to zero.

Bit 12 – PFRE Pause Frame Received (cleared on read)
Indicates a valid pause has been received.

Bit 11 – HRESP Hresp Not OK (cleared on read)
Set when the DMA block sees a bus error.

Bit 10 – ROVR Receive Overrun (cleared on read)
Set when the ‘Receive Overrun’ bit in EMAC_ISR gets set.

Bit 9 – LINK Link Change (cleared on read)
Set when the external link signal changes.

Bit 7 – TCOMP Transmit Complete (cleared on read)
Set when a frame has been transmitted.

Bit 6 – TXERR Transmit Error (cleared on read)
Transmit buffers exhausted in mid-frame - transmit error.

Bit 5 – RLEX Retry Limit Exceeded (cleared on read)

Bit 4 – TUND Ethernet Transmit Buffer Underrun (cleared on read)
The transmit DMA did not fetch frame data in time for it to be transmitted or hresp returned not OK. Also set if a used bit is read mid-frame or when a new transmit queue pointer is written.

Downloaded from Arrow.com.

Bit 3 – TXUBR Transmit Used Bit Read (cleared on read)
Set when a transmit buffer descriptor is read with its used bit set.

Bit 2 – RXUBR Receive Used Bit Read (cleared on read)
Set when a receive buffer descriptor is read with its used bit set.

Bit 1 – RCOMP Receive Complete (cleared on read)
A frame has been stored in memory.

Bit 0 – MFD Management Frame Done (cleared on read)
The PHY Maintenance Register has completed its operation.

Downloaded from Arrow.com.

39.6.9 Interrupt Enable Register

Name: EMAC_IER
Offset: 0x28
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		WOL	PTZ	PFR	HRESP	ROVR	LINK	
Access		W	W	W	W	W	W	
Reset		–	–	–	–	–	–	

Bit	7	6	5	4	3	2	1	0
	TCOMP	TXERR	RLE	TUND	TXUBR	RXUBR	RCOMP	MFD
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 14 – WOL Wake-On-LAN
Enable Wake-On-LAN interrupt.

Bit 13 – PTZ Pause Time Zero
Enable pause time zero interrupt.

Bit 12 – PFR Pause Frame Received
Enable pause frame received interrupt.

Bit 11 – HRESP Hresp Not OK
Enable Hresp not OK interrupt.

Bit 10 – ROVR Receive Overrun
Enable receive overrun interrupt.

Bit 9 – LINK Link Change
Enable link change interrupt.

Bit 7 – TCOMP Transmit Complete
Enable transmit complete interrupt.

Bit 6 – TXERR Transmit Error
Enable transmit buffers exhausted in mid-frame interrupt.

Bit 5 – RLE Retry Limit Exceeded
Enable retry limit exceeded interrupt.

Bit 4 – TUND Ethernet Transmit Buffer Underrun
Enable transmit underrun interrupt.

Downloaded from Arrow.com.

Bit 3 – TXUBR Transmit Used Bit Read
Enable transmit used bit read interrupt.

Bit 2 – RXUBR Receive Used Bit Read
Enable receive used bit read interrupt.

Bit 1 – RCOMP Receive Complete
Enable receive complete interrupt.

Bit 0 – MFD Management Frame Done
Enable management done interrupt.

Downloaded from Arrow.com.

39.6.10 Interrupt Disable Register

Name: EMAC_IDR
Offset: 0x2C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		WOL	PTZ	PFR	HRESP	ROVR	LINK	
Access		W	W	W	W	W	W	
Reset		–	–	–	–	–	–	

Bit	7	6	5	4	3	2	1	0
	TCOMP	TXERR	RLE	TUND	TXUBR	RXUBR	RCOMP	MFD
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 14 – WOL Wake-On-LAN
Disable Wake-On-LAN interrupt.

Bit 13 – PTZ Pause Time Zero
Disable pause time zero interrupt.

Bit 12 – PFR Pause Frame Received
Disable pause frame received interrupt.

Bit 11 – HRESP Hresp Not OK
Disable Hresp not OK interrupt.

Bit 10 – ROVR Receive Overrun
Disable receive overrun interrupt.

Bit 9 – LINK Link Change
Disable link change interrupt.

Bit 7 – TCOMP Transmit Complete
Disable transmit complete interrupt.

Bit 6 – TXERR Transmit Error
Disable transmit buffers exhausted in mid-frame interrupt.

Bit 5 – RLE Retry Limit Exceeded
Disable retry limit exceeded interrupt.

Bit 4 – TUND Ethernet Transmit Buffer Underrun
Disable transmit underrun interrupt.

Downloaded from Arrow.com.

Bit 3 – TXUBR Transmit Used Bit Read
Disable transmit used bit read interrupt.

Bit 2 – RXUBR Receive Used Bit Read
Disable receive used bit read interrupt.

Bit 1 – RCOMP Receive Complete
Disable receive complete interrupt.

Bit 0 – MFD Management Frame Done
Disable management done interrupt.

Downloaded from Arrow.com.

39.6.11 Interrupt Mask Register

Name: EMAC_IMR
Offset: 0x30
Reset: 0x00007FFF
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		WOL	PTZ	PFR	HRESP	ROVR	LINK	
Access		R	R	R	R	R	R	
Reset		1	1	1	1	1	1	

Bit	7	6	5	4	3	2	1	0
	TCOMP	TXERR	RLE	TUND	TXUBR	RXUBR	RCOMP	MFD
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	1	1

Bit 14 – WOL Wake-On-LAN
Wake-On-LAN interrupt masked.

Bit 13 – PTZ Pause Time Zero
Pause time zero interrupt masked.

Bit 12 – PFR Pause Frame Received
Pause frame received interrupt masked.

Bit 11 – HRESP Hresp Not OK
Hresp not OK interrupt masked.

Bit 10 – ROVR Receive Overrun
Receive overrun interrupt masked.

Bit 9 – LINK Link Change
Link change interrupt masked.

Bit 7 – TCOMP Transmit Complete
Transmit complete interrupt masked.

Bit 6 – TXERR Transmit Error
Transmit buffers exhausted in mid-frame interrupt masked.

Bit 5 – RLE Retry Limit Exceeded
Retry limit exceeded interrupt masked.

Bit 4 – TUND Ethernet Transmit Buffer Underrun
Transmit underrun interrupt masked.

Downloaded from Arrow.com.

Bit 3 – TXUBR Transmit Used Bit Read
Transmit used bit read interrupt masked.

Bit 2 – RXUBR Receive Used Bit Read
Receive used bit read interrupt masked.

Bit 1 – RCOMP Receive Complete
Receive complete interrupt masked.

Bit 0 – MFD Management Frame Done
Management done interrupt masked.

Downloaded from Arrow.com.

39.6.12 PHY Maintenance Register

Name: EMAC_MAN
Offset: 0x34
Reset: 0x00000000
Property: Read/Write

To read clause 45 PHYs, bits 31:28 should be written as 0x0011. This overlaps the SOF and RW fields.

Bit	31	30	29	28	27	26	25	24
	SOF[1:0]		RW[1:0]		PHYA[4:1]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PHYA[0]	REGA[4:0]					CODE[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:30 – SOF[1:0] Start of Frame
Must be written to one for a valid frame.

Bits 29:28 – RW[1:0] PHY Read/Write Command
Any other value is an invalid PHY management frame.

Value	Description
1	Write command
2	Read command

Bits 27:23 – PHYA[4:0] PHY Address

Bits 22:18 – REGA[4:0] PHY Register Address
Specifies the register in the PHY to access.

Bits 17:16 – CODE[1:0] Must Be Two
Must be written to 2. Reads as written.

Bits 15:0 – DATA[15:0] PHY Transmit or Receive Data
For a write operation this is written with the data to be written to the PHY.
After a read operation this contains the data read from the PHY.

Downloaded from Arrow.com.

39.6.13 Pause Time Register

Name: EMAC_PTR
Offset: 0x38
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PTIME[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PTIME[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PTIME[15:0] Pause Time
Stores the current value of the EMAC_PTR which is decremented every 512 bit times.

Downloaded from Arrow.com.

39.6.14 EMAC Statistics Registers

Name: EMAC_PFR
Offset: 0x3C
Reset: 0x00000000
Property: Read/Write

EMAC_PFR is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	FROK[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FROK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – FROK[15:0] Pause Frames Received OK
A 16-bit register counting the number of good pause frames received. A good frame has a length of 64 to 1518 (1536 if bit 8 set in EMAC_NCFGR) and has no FCS, alignment or receive symbol errors.

Downloaded from Arrow.com.

39.6.15 Frames Transmitted OK Register

Name: EMAC_FTO
Offset: 0x40
Reset: 0x00000000
Property: Read/Write

EMAC_FTO is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	FTOK[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FTOK[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FTOK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – FTOK[23:0] Frames Transmitted OK
A 24-bit register counting the number of frames successfully transmitted, i.e., no underrun and not too many retries.

Downloaded from Arrow.com.

39.6.16 Single Collision Frames Register

Name: EMAC_SCF
Offset: 0x44
Reset: 0x00000000
Property: Read/Write

EMAC_SCF is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SCF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SCF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – SCF[15:0] Single Collision Frames
A 16-bit register counting the number of frames experiencing a single collision before being successfully transmitted, i.e., no underrun.

Downloaded from Arrow.com.

39.6.17 Multicollision Frames Register

Name: EMAC_MCF
Offset: 0x48
Reset: 0x00000000
Property: Read/Write

EMAC_MCF is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	MCF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MCF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – MCF[15:0] Multicollision Frames
A 16-bit register counting the number of frames experiencing between two and fifteen collisions prior to being successfully transmitted, i.e., no underrun and not too many retries.

Downloaded from Arrow.com.

39.6.18 Frames Received OK Register

Name: EMAC_FRO
Offset: 0x4C
Reset: 0x00000000
Property: Read/Write

EMAC_FRO is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	FROK[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FROK[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FROK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – FROK[23:0] Frames Received OK
A 24-bit register counting the number of good frames received, i.e., address recognized and successfully copied to memory. A good frame is of length 64 to 1518 bytes (1536 if bit 8 set in EMAC_NCFGR) and has no FCS, alignment or receive symbol errors.

Downloaded from Arrow.com.

39.6.19 Frames Check Sequence Errors Register

Name: EMAC_FCSE
Offset: 0x50
Reset: 0x00000000
Property: Read/Write

EMAC_FCSE is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FCSE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – FCSE[7:0] Frame Check Sequence Errors
An 8-bit register counting frames that are an integral number of bytes, have bad CRC and are between 64 and 1518 bytes in length (1536 if bit 8 set in EMAC_NCFGR). This register is also incremented if a symbol error is detected and the frame is of valid length and has an integral number of bytes.

Downloaded from Arrow.com.

39.6.20 Alignment Errors Register

Name: EMAC_ALE
Offset: 0x54
Reset: 0x00000000
Property: Read/Write

EMAC_ALE is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ALE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – ALE[7:0] Alignment Errors
An 8-bit register counting frames that are not an integral number of bytes long and have bad CRC when their length is truncated to an integral number of bytes and are between 64 and 1518 bytes in length (1536 if bit 8 set in EMAC_NCFGR). This register is also incremented if a symbol error is detected and the frame is of valid length and does not have an integral number of bytes.

Downloaded from Arrow.com.

39.6.21 Deferred Transmission Frames Register

Name: EMAC_DTF
Offset: 0x58
Reset: 0x00000000
Property: Read/Write

EMAC_DTF is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DTF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DTF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – DTF[15:0] Deferred Transmission Frames
A 16-bit register counting the number of frames experiencing deferral due to carrier sense being active on their first attempt at transmission. Frames involved in any collision are not counted nor are frames that experienced a transmit underrun.

Downloaded from Arrow.com.

39.6.22 Late Collisions Register

Name: EMAC_LCOL
Offset: 0x5C
Reset: 0x00000000
Property: Read/Write

EMAC_LCOL is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	LCOL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – LCOL[7:0] Late Collisions
An 8-bit register counting the number of frames that experience a collision after the slot time (512 bits) has expired. A late collision is counted twice; i.e., both as a collision and a late collision.

Downloaded from Arrow.com.

39.6.23 Excessive Collisions Register

Name: EMAC_ECOL
Offset: 0x60
Reset: 0x00000000
Property: Read/Write

EMAC_ECOL is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EXCOL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – EXCOL[7:0] Excessive Collisions
An 8-bit register counting the number of frames that failed to be transmitted because they experienced 16 collisions.

Downloaded from Arrow.com.

39.6.24 Transmit Underrun Errors Register

Name: EMAC_TUND
Offset: 0x64
Reset: 0x00000000
Property: Read/Write

EMAC_TUND is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	TUND[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – TUND[7:0] Transmit Underruns
An 8-bit register counting the number of frames not transmitted due to a transmit DMA underrun. If this register is incremented, then no other statistics register is incremented.

Downloaded from Arrow.com.

39.6.25 Carrier Sense Errors Register

Name: EMAC_CSE
Offset: 0x68
Reset: 0x00000000
Property: Read/Write

EMAC_CSE is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CSE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CSE[7:0] Carrier Sense Errors
An 8-bit register counting the number of frames transmitted where carrier sense was not seen during transmission or where carrier sense was deasserted after being asserted in a transmit frame without collision (no underrun). Only incremented in half-duplex mode. The only effect of a carrier sense error is to increment this register. The behavior of the other statistics registers is unaffected by the detection of a carrier sense error.

Downloaded from Arrow.com.

39.6.26 Receive Resource Errors Register

Name: EMAC_RRE
Offset: 0x6C
Reset: 0x00000000
Property: Read/Write

EMAC_RRE is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.
This receive statistics register is only incremented when the 'Receive Enable' bit is set in the Network Control Register (EMAC_NCR). To write to this register, bit 7 must be set in EMAC_NCR.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RRE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RRE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RRE[15:0] Receive Resource Errors
A 16-bit register counting the number of frames that were address matched but could not be copied to memory because no receive buffer was available.

Downloaded from Arrow.com.

39.6.27 Receive Overrun Errors Register

Name: EMAC_ROV
Offset: 0x70
Reset: 0x00000000
Property: Read/Write

EMAC_ROV is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.
This receive statistics register is only incremented when the 'Receive Enable' bit is set in the Network Control Register (EMAC_NCR). To write to this register, bit 7 must be set in EMAC_NCR.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ROVR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – ROVR[7:0] Receive Overrun
An 8-bit register counting the number of frames that are address recognized but were not copied to memory due to a receive DMA overrun.

Downloaded from Arrow.com.

39.6.28 Receive Symbol Errors Register

Name: EMAC_RSE
Offset: 0x74
Reset: 0x00000000
Property: Read/Write

EMAC_RSE is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.
This receive statistics register is only incremented when the 'Receive Enable' bit is set in the Network Control Register (EMAC_NCR). To write to this register, bit 7 must be set in EMAC_NCR.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RSE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – RSE[7:0] Receive Symbol Errors
An 8-bit register counting the number of frames that had rx_er asserted during reception. Receive symbol errors are also counted as an FCS or alignment error if the frame is between 64 and 1518 bytes in length (1536 if bit 8 is set in the EMAC_NCFGR). If the frame is larger, it is recorded as a jabber error.

Downloaded from Arrow.com.

39.6.29 Excessive Length Errors Register

Name: EMAC_ELE
Offset: 0x78
Reset: 0x00000000
Property: Read/Write

EMAC_ELE is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EXL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – EXL[7:0] Excessive Length Errors
An 8-bit register counting the number of frames received exceeding 1518 bytes (1536 if bit 8 set in EMAC_NCFGR) in length but do not have either a CRC error, an alignment error nor a receive symbol error.

Downloaded from Arrow.com.

39.6.30 Receive Jabbers Register

Name: EMAC_RJA
Offset: 0x7C
Reset: 0x00000000
Property: Read/Write

EMAC_RJA is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.
This receive statistics register is only incremented when the 'Receive Enable' bit is set in the Network Control Register (EMAC_NCR). To write to this register, bit 7 must be set in EMAC_NCR.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RJB[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – RJB[7:0] Receive Jabbers
An 8-bit register counting the number of frames received exceeding 1518 bytes (1536 if bit 8 set in EMAC_NCFGR) in length and have either a CRC error, an alignment error or a receive symbol error.

Downloaded from Arrow.com.

39.6.31 Undersize Frames Register

Name: EMAC_USF
Offset: 0x80
Reset: 0x00000000
Property: Read/Write

EMAC_USF is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	USF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – USF[7:0] Undersize Frames
An 8-bit register counting the number of frames received less than 64 bytes in length but do not have either a CRC error, an alignment error or a receive symbol error.

Downloaded from Arrow.com.

39.6.32 SQE Test Errors Register

Name: EMAC_STE
Offset: 0x84
Reset: 0x00000000
Property: Read/Write

EMAC_STE is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	SQER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – SQER[7:0] SQE Test Errors
An 8-bit register counting the number of frames where col was not asserted within 96 bit times (an interframe gap) of tx_en being deasserted in half duplex mode.

Downloaded from Arrow.com.

39.6.33 Received Length Field Mismatch Register

Name: EMAC_RLE
Offset: 0x88
Reset: 0x00000000
Property: Read/Write

EMAC_RLE is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RLFM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – RLFM[7:0] Receive Length Field Mismatch
An 8-bit register counting the number of frames received that have a measured length shorter than that extracted from its length field. Checking is enabled through bit 16 of the EMAC_NCFGR. Frames containing a type ID in bytes 13 and 14 (i.e., length/type ID = 0x0600) are not counted as length field errors, neither are excessive length frames.

Downloaded from Arrow.com.

39.6.34 Transmitted Pause Frames Register

Name: EMAC_TPF
Offset: 0x8C
Reset: 0x00000000
Property: Read/Write

EMAC_TPF is an EMAC statistics register. It resets to zero on a read and sticks at all ones when it counts to its maximum value. It should be read frequently enough to prevent loss of data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TPF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TPF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TPF[15:0] Transmitted Pause Frames
A 16-bit register counting the number of pause frames transmitted.

Downloaded from Arrow.com.

39.6.35 Hash Register Bottom

Name: EMAC_HRB
Offset: 0x90
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Hash Address Bottom
Bits 31:0 of the hash address register. See Hash Addressing.

Downloaded from Arrow.com.

39.6.36 Hash Register Top

Name: EMAC_HRT
Offset: 0x94
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Hash Address Top
Bits 63:32 of the hash address register. See 39.4.8 Hash Addressing.

Downloaded from Arrow.com.

39.6.37 Specific Address 1 Bottom Register

Name: EMAC_SA1B
Offset: 0x98
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Specific Address 1 Bottom
Least significant bits of the destination address. Bit zero indicates whether the address is multicast or unicast and corresponds to the least significant bit of the first byte received.

Downloaded from Arrow.com.

39.6.38 Specific Address 1 Top Register

Name: EMAC_SA1T
Offset: 0x9C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – ADDR[15:0] Specific Address 1 Top
The most significant bits of the destination address, that is bits 47 to 32.

Downloaded from Arrow.com.

39.6.39 Specific Address 2 Bottom Register

Name: EMAC_SA2B
Offset: 0xA0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Specific Address 2 Bottom
Least significant bits of the destination address. Bit zero indicates whether the address is multicast or unicast and corresponds to the least significant bit of the first byte received.

Downloaded from Arrow.com.

39.6.40 Specific Address 2 Top Register

Name: EMAC_SA2T
Offset: 0xA4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – ADDR[15:0] Specific Address 2 Top
The most significant bits of the destination address, that is bits 47 to 32.

Downloaded from Arrow.com.

39.6.41 Specific Address 3 Bottom Register

Name: EMAC_SA3B
Offset: 0xA8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Specific Address 3 Bottom
Least significant bits of the destination address. Bit zero indicates whether the address is multicast or unicast and corresponds to the least significant bit of the first byte received.

Downloaded from Arrow.com.

39.6.42 Specific Address 3 Top Register

Name: EMAC_SA3T
Offset: 0xAC
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – ADDR[15:0] Specific Address 3 Top
The most significant bits of the destination address, that is bits 47 to 32.

Downloaded from Arrow.com.

39.6.43 Specific Address 4 Bottom Register

Name: EMAC_SA4B
Offset: 0xB0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Specific Address 4 Bottom
Least significant bits of the destination address. Bit zero indicates whether the address is multicast or unicast and corresponds to the least significant bit of the first byte received.

Downloaded from Arrow.com.

39.6.44 Specific Address 4 Top Register

Name: EMAC_SA4T
Offset: 0xB4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – ADDR[15:0] Specific Address 4 Top
The most significant bits of the destination address, that is bits 47 to 32.

Downloaded from Arrow.com.

39.6.45 Type ID Checking Register

Name: EMAC_TID
Offset: 0xB8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TID[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TID[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TID[15:0] Type ID Checking
For use in comparisons with received frames TypeID/Length field.

Downloaded from Arrow.com.

39.6.46 Transmit Pause Quantum Register

Name: EMAC_TPQ
Offset: 0xBC
Reset: 0x0000FFFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TPQ[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	TPQ[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 15:0 – TPQ[15:0] Transmit Pause Quantum
Used in hardware generation of transmitted pause frames as value for pause quantum.

Downloaded from Arrow.com.

39.6.47 User Input/Output Register

Name: EMAC_USRIO
Offset: 0xC0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							CLKEN	RMII
Access							R/W	R/W
Reset							0	0

Bit 1 – CLKEN Clock Enable

Value	Description
0	Reduces power consumption when the treasurer is not used
1	Enables the transceiver input clock

Bit 0 – RMII Reduced MII
When set, this bit enables the RMII operation mode. When reset, it selects the MII mode.

Downloaded from Arrow.com.

39.6.48 Wake-on-LAN Register

Name: EMAC_WOL
Offset: 0xC4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					MTI	SA1	ARP	MAG
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 19 – MTI Multicast Hash Event Enable
When set, multicast hash events causes the wol output to be asserted.

Bit 18 – SA1 Specific Address Register 1 Event Enable
When set, specific address 1 events causes the wol output to be asserted.

Bit 17 – ARP ARP Request Event Enable
When set, ARP request events causes the wol output to be asserted.

Bit 16 – MAG Magic Packet Event Enable
When set, magic packet events causes the wol output to be asserted.

Bits 15:0 – IP[15:0] ARP Request IP Address
Written to define the least significant 16 bits of the target IP address that is matched to generate a Wake-on-LAN event. A value of zero does not generate an event, even if this is matched by the received frame.

Downloaded from Arrow.com.

40. USB High Speed Device Port (UDPHS)

40.1 Description

The USB High Speed Device Port (UDPHS) is compliant with the Universal Serial Bus (USB), rev 2.0 High Speed device specification.

Each endpoint can be configured in one of several USB transfer types. It can be associated with one, two or three banks of a Dual-port RAM used to store the current data payload. If two or three banks are used, one DPR bank is read or written by the processor, while the other is read or written by the USB device peripheral. This feature is mandatory for isochronous endpoints.

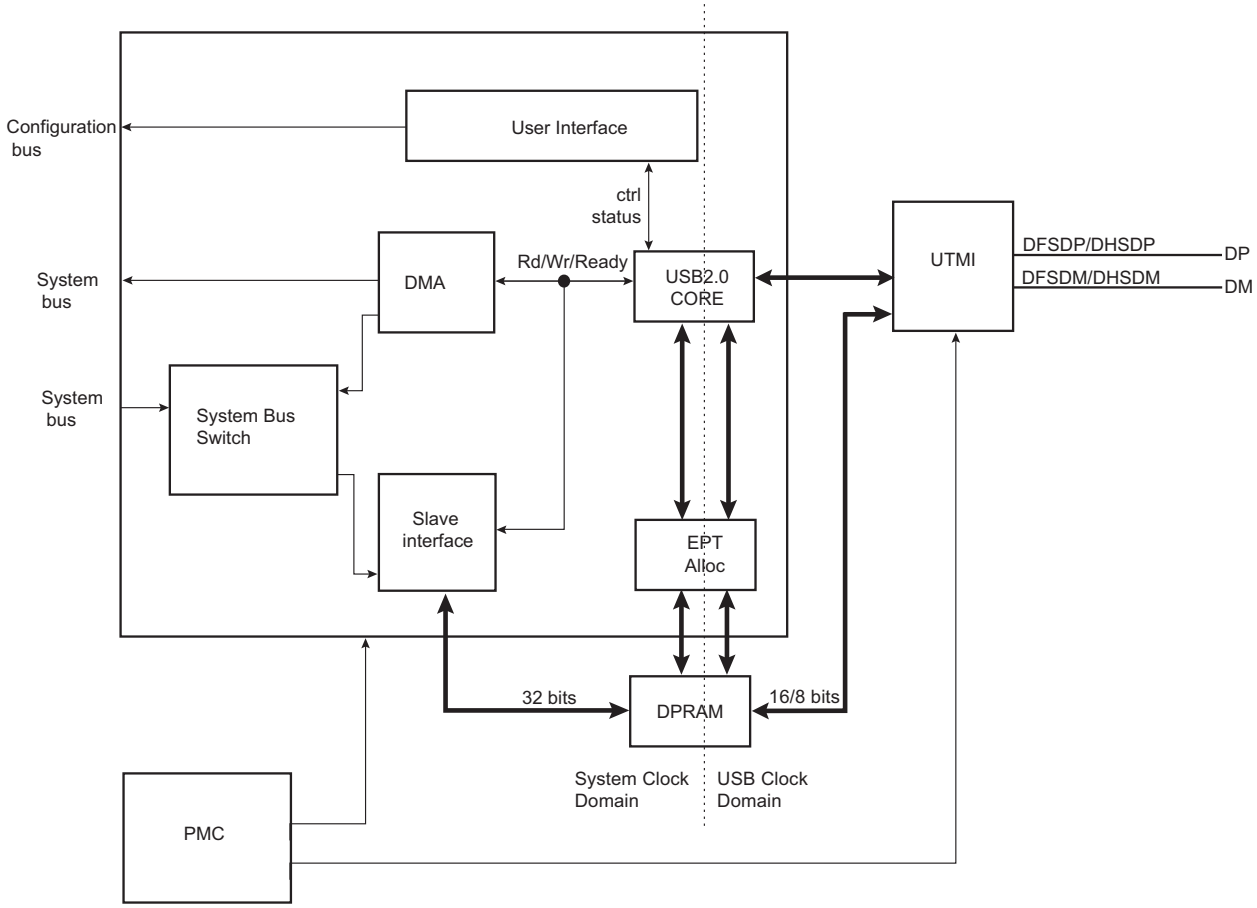
40.2 Embedded Characteristics

- 1 High-speed Device
- 1 UTMI transceiver shared between Host and Device
- USB v2.0 High Speed (480 Mbits/s) Compliant
- 7 Endpoints up to 1024 bytes
- Embedded Dual-port RAM for Endpoints
- Suspend/Resume Logic (Command of UTMI)
- Up to Three Memory Banks for Endpoints (Not for Control Endpoint)
- 16448 bytes of DPRAM

Downloaded from Arrow.com.

40.3 Block Diagram

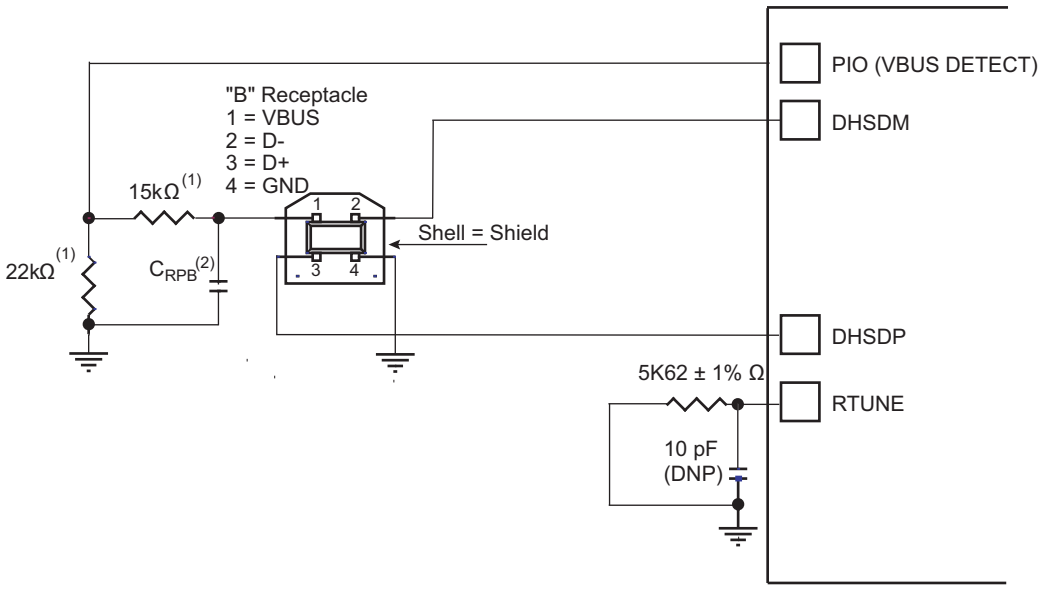
Figure 40-1. UDPHS Block Diagram



Downloaded from Arrow.com.

40.4 Typical Connection

Figure 40-2. Board Schematic



Note:

1. The values shown on the 22 kΩ and 15 kΩ resistors are only valid with 3.3V-supplied PIOs.
2. C_{RPB}: Upstream Facing Port Bypass Capacitance of 1 µF to 10 µF (refer to “DC Electrical Characteristics” in Universal Serial Bus Specification Rev. 2)
3. 10 pF capacitor on VBG is a provision and may not be populated.

40.5 Product Dependencies

40.5.1 Power Management

The UDPHS is not continuously clocked.

To use the UDPHS, the programmer must first enable the UDPHS clock in the Power Management Controller (PMC). Then, UTMI (PLL and transceiver) must be enabled by first enabling the bandgap, then the voltage regulator in PMC.

However, if the application does not require UDPHS operations, the UDPHS clock can be stopped when not needed and restarted later.

40.5.2 Interrupt Sources

The UDPHS interrupt line is connected on one of the internal sources of the interrupt controller. Using the UDPHS interrupt requires the interrupt controller to be programmed first.

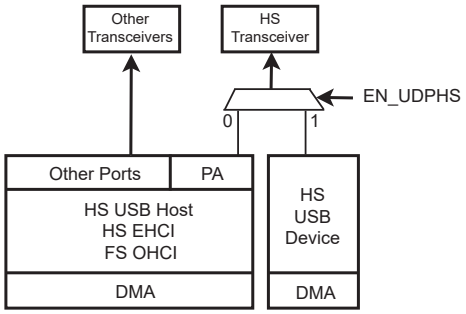
40.6 Functional Description

40.6.1 UTMI Transceivers Sharing

The High Speed USB Host Port A is shared with the High Speed USB Device port and connected to the second UTMI transceiver. The selection between Host Port A and USB Device is controlled by the UDPHS enable bit (EN_UDPHS) located in the UDPHS_CTRL register.

Downloaded from Arrow.com.

Figure 40-3. USB Selection



40.6.2 USB V2.0 High Speed Device Port Introduction

The USB V2.0 High Speed Device Port provides communication services between host and attached USB devices. Each device is offered with a collection of communication flows (pipes) associated with each endpoint. Software on the host communicates with a USB Device through a set of communication flows.

40.6.3 USB V2.0 High Speed Transfer Types

A communication flow is carried over one of four transfer types defined by the USB device.

A device provides several logical communication pipes with the host. To each logical pipe is associated an endpoint. Transfer through a pipe belongs to one of the four transfer types:

- Control Transfers: Used to configure a device at attach time and can be used for other device-specific purposes, including control of other pipes on the device.
- Bulk Data Transfers: Generated or consumed in relatively large burst quantities and have wide dynamic latitude in transmission constraints.
- Interrupt Data Transfers: Used for timely but reliable delivery of data, for example, characters or coordinates with human-perceptible echo or feedback response characteristics.
- Isochronous Data Transfers: Occupy a prenegotiated amount of USB bandwidth with a prenegotiated delivery latency. (Also called streaming real time transfers.)

As indicated below, transfers are sequential events carried out on the USB bus.

Endpoints must be configured according to the transfer type they handle.

Table 40-1. USB Communication Flow

Transfer	Direction	Bandwidth	Endpoint Size	Error Detection	Retrying
Control	Bidirectional	Not guaranteed	8, 16, 32, 64	Yes	Automatic
Isochronous	Unidirectional	Guaranteed	8–1024	Yes	No
Interrupt	Unidirectional	Not guaranteed	8–1024	Yes	Yes
Bulk	Unidirectional	Not guaranteed	8–512	Yes	Yes

40.6.4 USB Transfer Event Definitions

A transfer is composed of one or several transactions as shown in the table below.

Table 40-2. USB Transfer Events

Transfer		Transaction
Direction	Type	
CONTROL (bidirectional)	Control Transfer (1)	• Setup transaction → Data IN transactions → Status OUT transaction • Setup transaction → Data OUT transactions → Status IN transaction • Setup transaction → Status IN transaction

Downloaded from Arrow.com.

.....continued

Transfer		Transaction
Direction	Type	
IN (device toward host)	Bulk IN Transfer	• Data IN transaction → Data IN transaction
	Interrupt IN Transfer	• Data IN transaction → Data IN transaction
	Isochronous IN Transfer (2)	• Data IN transaction → Data IN transaction
OUT (host toward device)	Bulk OUT Transfer	• Data OUT transaction → Data OUT transaction
	Interrupt OUT Transfer	• Data OUT transaction → Data OUT transaction
	Isochronous OUT Transfer (2)	• Data OUT transaction → Data OUT transaction

Note:

1. Control transfer must use endpoints with one bank and can be aborted using a stall handshake.
2. Isochronous transfers must use endpoints configured with two or three banks.

An endpoint handles all transactions related to the type of transfer for which it has been configured.

Table 40-3. UDPHS Endpoint Description

Endpoint #	Mnemonic	Nb Banks	DMA	High Bandwidth	Max. Endpoint Size	Endpoint Type
0	EPT_0	1	N	N	64	Control
1	EPT_1	2	Y	N	1024	Ctrl/Bulk/Iso(1)/Interrupt
2	EPT_2	2	Y	N	1024	Ctrl/Bulk/Iso(1)/Interrupt
3	EPT_3	3	Y	Y	1024	Ctrl/Bulk/Iso(1)/Interrupt
4	EPT_4	3	Y	Y	1024	Ctrl/Bulk/Iso(1)/Interrupt
5	EPT_5	3	Y	Y	1024	Ctrl/Bulk/Iso(1)/Interrupt
6	EPT_6	3	Y	Y	1024	Ctrl/Bulk/Iso(1)/Interrupt

Note:

1. In Isochronous (Iso) mode, it is preferable that the high bandwidth capability is available.

The size of the internal DPRAM is 16448 bytes, covering the memory need for the seven endpoints, hence enabling static allocation of the memory for all endpoints.

Suspend and resume are automatically detected by the UDPHS device, which notifies the processor by raising an interrupt.

40.6.5 USB V2.0 High Speed BUS Transactions

Each transfer results in one or more transactions over the USB bus.

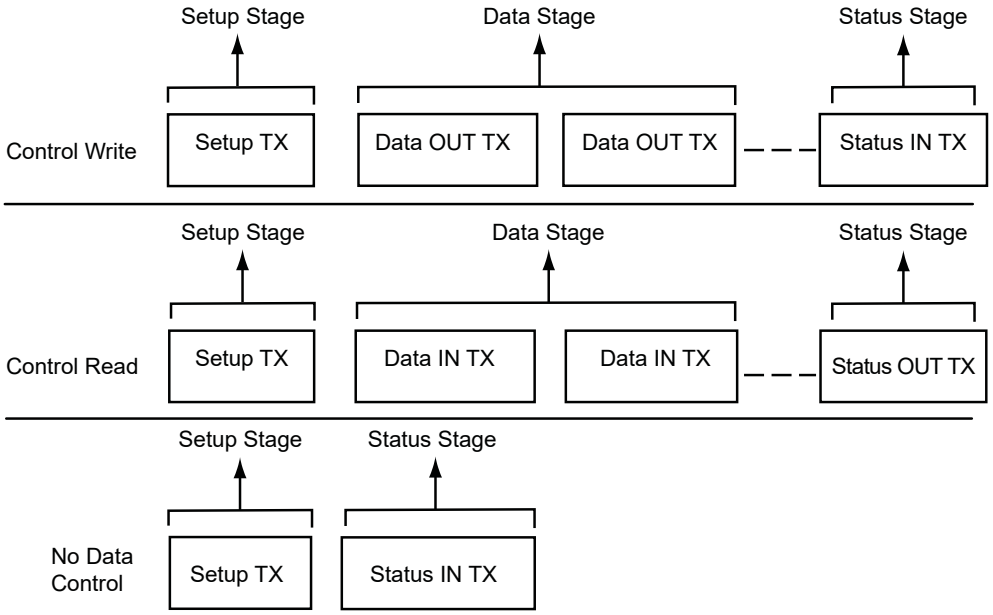
Five types of transaction flow across the bus in packets:

1. Setup Transaction
2. Data IN Transaction
3. Data OUT Transaction
4. Status IN Transaction
5. Status OUT Transaction

A status IN or OUT transaction is identical to a data IN or OUT transaction.

Downloaded from Arrow.com.

Figure 40-4. Control Read and Write Sequences



40.6.6 Endpoint Configuration

The endpoint 0 is always a control endpoint, it must be programmed and active in order to be enabled when the End Of Reset interrupt occurs.

To configure the endpoints:

- Fill the configuration register (UDPHS_EPTCFG) with the endpoint size, direction (IN or OUT), type (CTRL, Bulk, IT, ISO) and the number of banks.
- Fill the number of transactions (NB_TRANS) for isochronous endpoints.

Note: For control endpoints the direction has no effect.

- Verify that the EPT_MAPD flag is set. This flag is set if the endpoint size and the number of banks are correct for this endpoint.
- Configure control flags of the endpoint and enable it in UDPHS_EPTCTLENBx according to the section UDPHS Endpoint Control Disable Register (Isochronous Endpoint).

Control endpoints can generate interrupts and use only 1 bank.

All endpoints (except endpoint 0) can be configured either as Bulk, Interrupt or Isochronous. Refer to the table "UDPHS Endpoint Description" in 40.6.4 USB Transfer Event Definitions.

The maximum packet size they can accept corresponds to the maximum endpoint size.

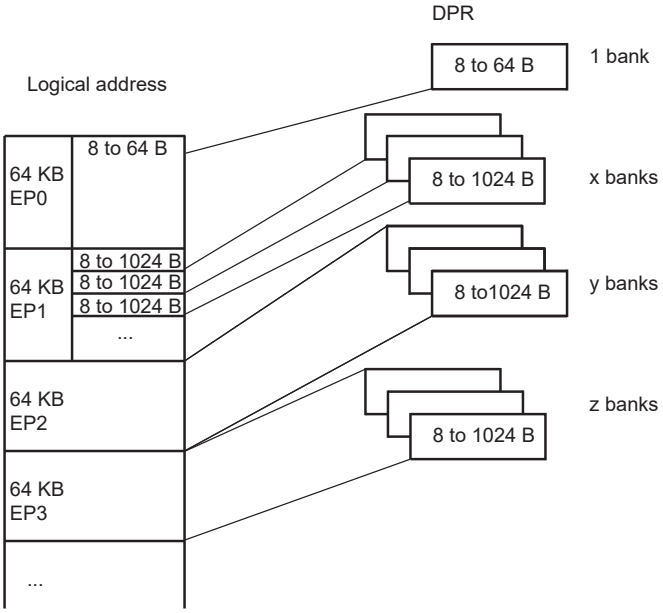
Note: The endpoint size of 1024 is reserved for isochronous endpoints.

The application has access to the physical block of DPR reserved for the endpoint through a 64-Kbyte logical address space.

The physical block of DPR reserved for the endpoint is remapped all along the 64-Kbyte logical address space. The application can write a 64-Kbyte buffer linearly.

Downloaded from Arrow.com.

Figure 40-5. Logical Address Space for DPR Access



Configuration examples of UDPHS_EPTCTLx (UDPHS Endpoint Control Disable Register (Isochronous Endpoint)) for Bulk IN endpoint type follow below.

- With DMA
  - AUTO_VALID: Automatically validate the packet and switch to the next bank.
  - EPT_ENABL: Enable endpoint.
- Without DMA:
  - TXRDY: An interrupt is generated after each transmission.
  - EPT_ENABL: Enable endpoint.

Configuration examples of Bulk OUT endpoint type follow below.

- With DMA
  - AUTO_VALID: Automatically validate the packet and switch to the next bank.
  - EPT_ENABL: Enable endpoint.
- Without DMA
  - RXRDY_TXKL: An interrupt is sent after a new packet has been stored in the endpoint FIFO.
  - EPT_ENABL: Enable endpoint.

40.6.7 DPRAM Management

Endpoints can be configured in any order.

Disabling an endpoint, by writing a one to the Endpoint Disable bit in the UDPHS Endpoint Control Disable Register (UDPHS_EPTCTLDISx.EPT_DISABL), does not reset its configuration:

- Endpoint Banks (UDPHS_EPTCFGx.BK_NUMBER)
- Endpoint Size (UDPHS_EPTCFGx.EPT_SIZE)
- Endpoint Direction (UDPHS_EPTCFGx.EPT_DIR)
- Endpoint Type (UDPHS_EPTCFGx.EPT_TYPE)

Downloaded from Arrow.com.

Note:

1. There is no way the data of the endpoint 0 can be lost (except if it is de-allocated) as the memory allocation and de-allocation may affect only higher endpoints.
2. Deactivating then reactivating the same endpoint with the same configuration only modifies temporarily the controller DPRAM pointer and size for this endpoint. Nothing changes in the DPRAM, higher endpoints seem not to have been moved and their data is preserved as far as nothing has been written or received into them while changing the allocation state of the first endpoint.
3. When the user writes a value different from zero to the UDPHS_EPTCFGx.BK_NUMBER field, the Endpoint Mapped bit (UDPHS_EPTCFGx.EPT_MAPD) is set only if the configured size and number of banks are correct as compared to the endpoint maximal allowed values and to the maximal FIFO size (i.e., the DPRAM size). The UDPHS_EPTCFGx.EPT_MAPD value does not consider memory allocation conflicts.

40.6.8 Transfer With DMA

USB packets of any length may be transferred when required by the UDPHS device. These transfers always feature sequential addressing.

Packet data AHB bursts may be locked on a DMA buffer basis for drastic overall AHB bus bandwidth performance boost with paged memories. These clock-cycle consuming memory row (or bank) changes will then likely not occur, or occur only once instead of several times, during a single big USB packet DMA transfer in case another AHB master addresses the memory. The locked bursts result in up to 128-word single-cycle unbroken AHB bursts for bulk endpoints and 256-word single-cycle unbroken bursts for isochronous endpoints.

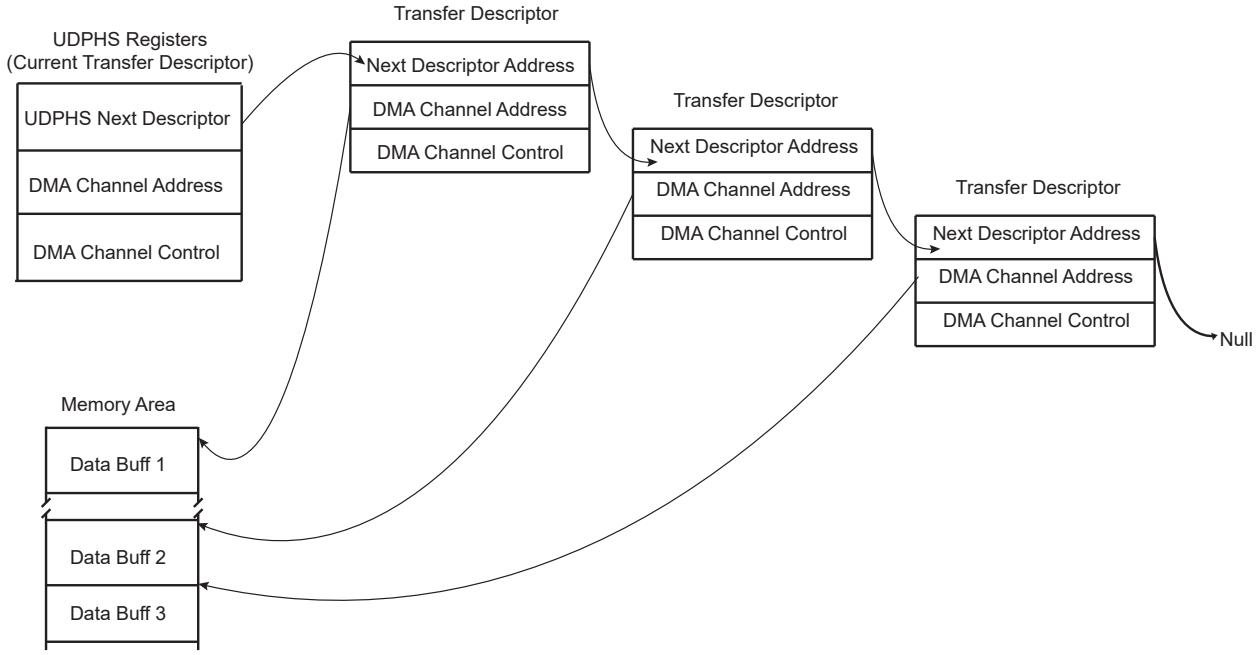
This maximum burst length is then controlled by the lowest programmed USB endpoint size (EPT_SIZE field in the UDPHS_EPTCFGx register) and DMA Size (BUFF_LENGTH field in the UDPHS_DMACONTROLx register).

The USB 2.0 device average throughput may be up to nearly 60 Mbyte/s. Its internal slave average access latency decreases as burst length increases due to the 0 wait-state side effect of unchanged endpoints. If at least 0 wait-state word burst capability is also provided by the external DMA AHB bus slaves, each of both DMA AHB busses need less than 50% bandwidth allocation for full USB 2.0 bandwidth usage at 30 MHz, and less than 25% at 60 MHz.

The UDPHS DMA Channel Transfer Descriptor is described in the section UDPHS DMA Channel Transfer Descriptor.

Note: In case of debug, be careful to address the DMA to an SRAM address even if a remap is done.

Figure 40-6. Example of DMA Chained List



Downloaded from Arrow.com.

40.6.9 Transfer Without DMA

Important: If the DMA is not to be used, it is necessary to disable it, otherwise it can be enabled by previous versions of software without warning. If this should occur, the DMA can process data before an interrupt without knowledge of the user.

The recommended means to disable DMA are as follows:

```
// Reset IP UDPHS
  AT91C_BASE_UDPHS->UDPHS_CTRL &= ~AT91C_UDPHS_EN_UDPHS;
  AT91C_BASE_UDPHS->UDPHS_CTRL |= AT91C_UDPHS_EN_UDPHS;//
With OR without DMA !!!
  for( i=1; i<=((AT91C_BASE_UDPHS->UDPHS_IPFEATURES &
AT91C_UDPHS_DMA_CHANNEL_NBR)>>4); i++ ) {
// RESET endpoint canal DMA:
    // DMA stop channel command
   AT91C_BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMACONTROL = 0; // STOP
command
// Disable endpoint
  AT91C_BASE_UDPHS->UDPHS_EPT[i].UDPHS_EPTCTLDIS |= 0XFFFFFFFF;
// Reset endpoint config
  AT91C_BASE_UDPHS->UDPHS_EPT[i].UDPHS_EPTCTLCFG = 0;
// Reset DMA channel (Buff count and Control field)
  AT91C_BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMACONTROL = 0x02; // NON
STOP command
// Reset DMA channel 0 (STOP)
  AT91C_BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMACONTROL = 0; // STOP
command
// Clear DMA channel status (read the register for clear it)
  AT91C_BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMASTATUS =
  AT91C_BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMASTATUS;
}
```

40.6.10 Handling Transactions with USB V2.0 Device Peripheral

40.6.10.1 Setup Transaction

The setup packet is valid in the DPR while RX_SETUP is set. Once RX_SETUP is cleared by the application, the UDPHS accepts the next packets sent over the device endpoint.

When a valid setup packet is accepted by the UDPHS:

- The UDPHS device automatically acknowledges the setup packet (sends an ACK response)
- Payload data is written in the endpoint
- Sets the RX_SETUP interrupt
- The BYTE_COUNT field in the UDPHS_EPTSTAx register is updated

An endpoint interrupt is generated while RX_SETUP in the UDPHS_EPTSTAx register is not cleared. This interrupt is carried out to the microcontroller if interrupts are enabled for this endpoint.

Thus, firmware must detect RX_SETUP polling UDPHS_EPTSTAx or catching an interrupt, read the setup packet in the FIFO, then clear the RX_SETUP bit in the UDPHS_EPTCLRSTA register to acknowledge the setup stage.

If STALL_SNT was set to 1, then this bit is automatically reset when a setup token is detected by the device. Then, the device still accepts the setup stage. (See the section STALL).

40.6.10.2 NYET

NYET is a High Speed only handshake. It is returned by a High Speed endpoint as part of the PING protocol.

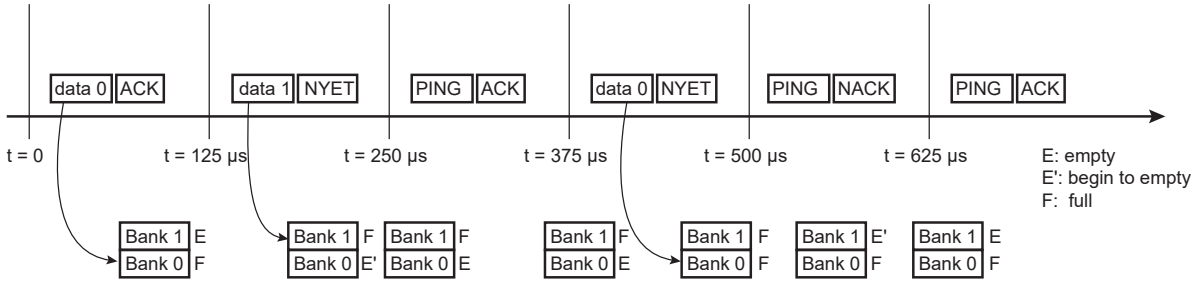
High Speed devices must support an improved NAK mechanism for Bulk OUT and control endpoints (except setup stage). This mechanism allows the device to tell the host whether it has sufficient endpoint space for the next OUT transfer (refer to USB 2.0 spec 8.5.1 NAK Limiting via Ping Flow Control).

The NYET/ACK response to a High Speed Bulk OUT transfer and the PING response are automatically handled by hardware in the UDPHS_EPTCTLx register (except when the user wants to force a NAK response by using the NYET_DIS bit).

Downloaded from Arrow.com.

If the endpoint responds instead to the OUT/DATA transaction with an NYET handshake, this means that the endpoint accepted the data but does not have room for another data payload. The host controller must return to using a PING token until the endpoint indicates it has space available.

Figure 40-7. NYET Example with Two Endpoint Banks



40.6.10.3 Data IN

• Bulk IN or Interrupt IN

Data IN packets are sent by the device during the data or the status stage of a control transfer or during an (interrupt/bulk/isochronous) IN transfer. Data buffers are sent packet by packet under the control of the application or under the control of the DMA channel.

There are three ways for an application to transfer a buffer in several packets over the USB:

- packet by packet (see Bulk IN or Interrupt IN: Sending a Packet Under Application Control (Device to Host) below)
- 64 Kbytes (see Bulk IN or Interrupt IN: Sending a Packet Under Application Control (Device to Host) below)
- DMA (see Bulk IN or Interrupt IN: Sending a Buffer Using DMA (Device to Host) below)
  • Bulk IN or Interrupt IN: Sending a Packet Under Application Control (Device to Host)

• Bulk IN or Interrupt IN: Sending a Packet Under Application Control (Device to Host)

The application can write one or several banks.

A simple algorithm can be used by the application to send packets regardless of the number of banks associated to the endpoint.

Algorithm Description for Each Packet

- The application waits for the TXRDY flag to be cleared in the UDPHS_EPTSTAx register before it can perform a write access to the DPR.
- The application writes one USB packet of data in the DPR through the 64 Kbytes endpoint logical memory window.
- The application sets TXRDY flag in the UDPHS_EPTSETSTAx register.

The application is notified that it is possible to write a new packet to the DPR by the TXRDY interrupt. This interrupt can be enabled or masked by setting the TXRDY bit in the UDPHS_EPTCTLENB/UDPHS_EPTCTLDIS register.

Algorithm Description to Fill Several Packets

Using the previous algorithm, the application is interrupted for each packet. It is possible to reduce the application overhead by writing linearly several banks at the same time. The AUTO_VALID bit in the UDPHS_EPTCTLx must be set by writing the AUTO_VALID bit in the UDPHS_EPTCTLENBx register.

The auto-valid-bank mechanism allows the transfer of data (IN and OUT) without the intervention of the CPU. This means that bank validation (set TXRDY or clear the RXRDY_TXKL bit) is done by hardware.

- The application checks the BUSY_BANK_STA field in the UDPHS_EPTSTAx register. The application must wait that at least one bank is free.
- The application writes a number of bytes inferior to the number of free DPR banks for the endpoint. Each time the application writes the last byte of a bank, the TXRDY signal is automatically set by the UDPHS.
- If the last packet is incomplete (i.e., the last byte of the bank has not been written) the application must set the TXRDY bit in the UDPHS_EPTSETSTAx register.

Downloaded from Arrow.com.

The application is notified that all banks are free, so that it is possible to write another burst of packets by the BUSY_BANK interrupt. This interrupt can be enabled or masked by setting the BUSY_BANK flag in the UDPHS_EPTCTLENB and UDPHS_EPTCTLDIS registers.

This algorithm must not be used for isochronous transfer. In this case, the ping-pong mechanism does not operate.

A Zero Length Packet can be sent by setting just the TXRDY flag in the UDPHS_EPTSETSTAx register.

• Bulk IN or Interrupt IN: Sending a Buffer Using DMA (Device to Host)

The UDPHS integrates a DMA host controller. This DMA controller can be used to transfer a buffer from the memory to the DPR or from the DPR to the processor memory under the UDPHS control. The DMA can be used for all transfer types except control transfer.

Example DMA configuration:

1. Program UDPHS_DMAADDRESS x with the address of the buffer that should be transferred.
2. Enable the interrupt of the DMA in UDPHS_IEN
3. Program UDPHS_ DMACONTROLx:
   – Size of buffer to send: size of the buffer to be sent to the host.
   – END_B_EN: The endpoint can validate the packet (according to the values programmed in the AUTO_VALID and SHRT_PCKT fields of UDPHS_EPTCTLx.) (see section UDPHS Endpoint Control Disable Register (Isochronous Endpoint) and Figure 40-12)
   – END_BUFFIT: generate an interrupt when the BUFF_COUNT in UDPHS_DMASTATUSx reaches 0.
   – CHANN_ENB: Run and stop at end of buffer

The auto-valid-bank mechanism allows the transfer of data (IN & OUT) without the intervention of the CPU. This means that bank validation (set TXRDY or clear the RXRDY_TXKL bit) is done by hardware.

A transfer descriptor can be used. Instead of programming the register directly, a descriptor should be programmed and the address of this descriptor is then given to UDPHS_DMANXTDSC to be processed after setting the LDNXT_DSC field (Load Next Descriptor Now) in UDPHS_DMACONTROLx register.

The structure that defines this transfer descriptor must be aligned.

Each buffer to be transferred must be described by a DMA Transfer descriptor (see section UDPHS DMA Channel Transfer Descriptor). Transfer descriptors are chained. Before executing transfer of the buffer, the UDPHS may fetch a new transfer descriptor from the memory address pointed by the UDPHS_DMANXTDSCx register. Once the transfer is complete, the transfer status is updated in the UDPHS_DMASTATUSx register.

To chain a new transfer descriptor with the current DMA transfer, the DMA channel must be stopped. To do so, INTDIS_DMA and TXRDY may be set in the UDPHS_EPTCTLENBx register. It is also possible for the application to wait for the completion of all transfers. In this case the LDNXT_DSC bit in the last transfer descriptor UDPHS_DMACONTROLx register must be set to 0 and the CHANN_ENB bit set to 1.

Then the application can chain a new transfer descriptor.

The INTDIS_DMA can be used to stop the current DMA transfer if an enabled interrupt is triggered. This can be used to stop DMA transfers in case of errors.

The application can be notified at the end of any buffer transfer (ENB_BUFFIT bit in the UDPHS_DMACONTROLx register).

Downloaded from Arrow.com.

Figure 40-8. Data IN Transfer for Endpoint with One Bank

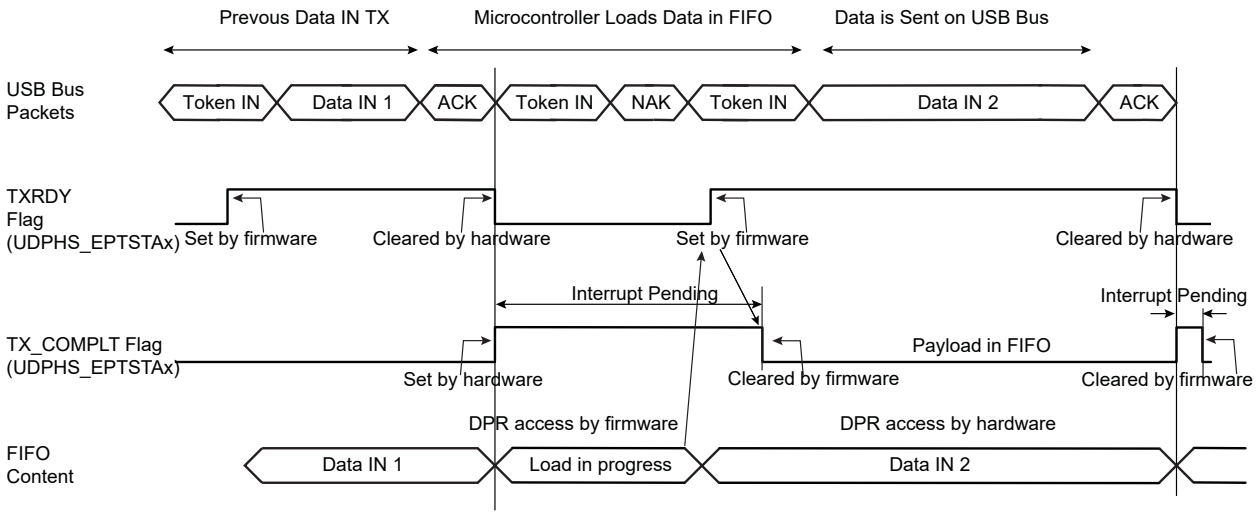
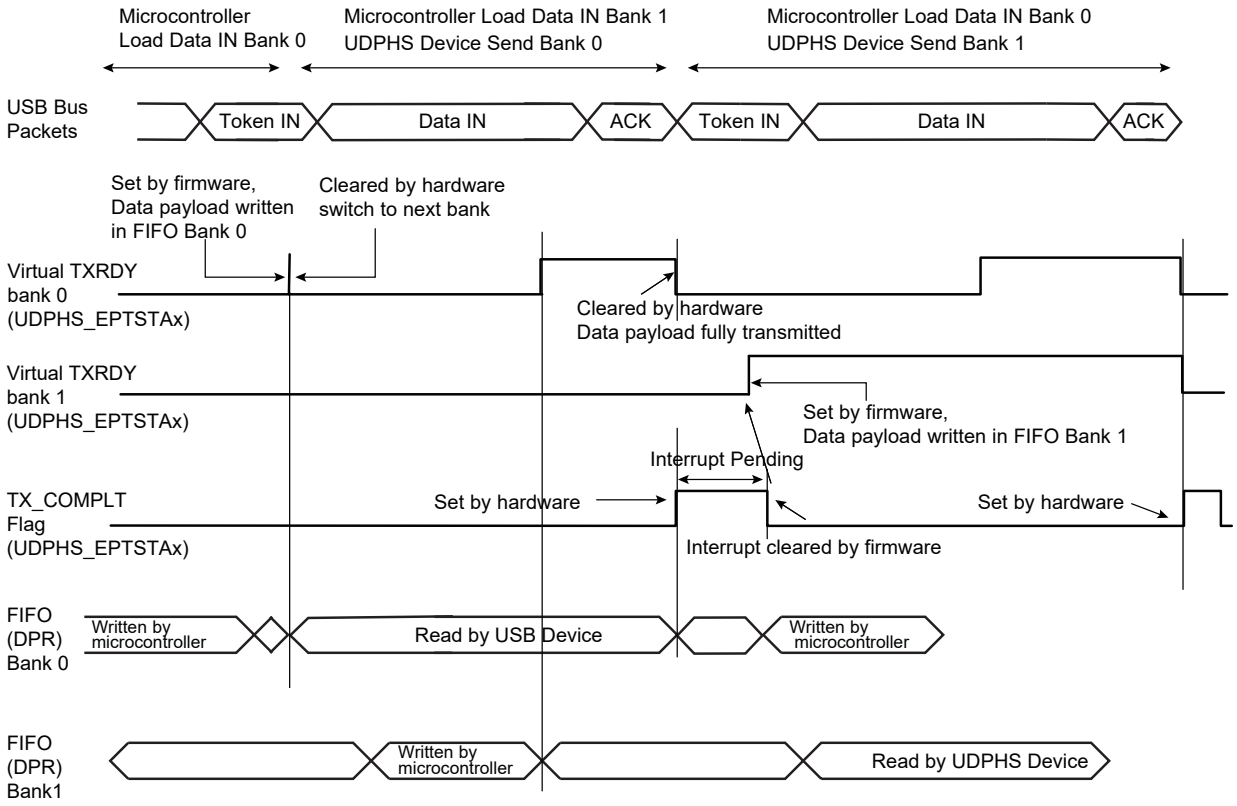
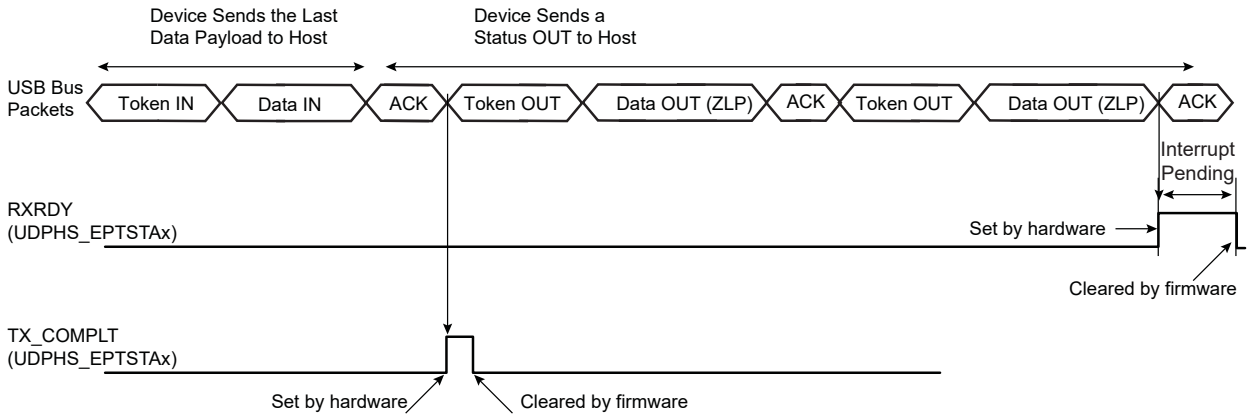


Figure 40-9. Data IN Transfer for Endpoint with Two Banks



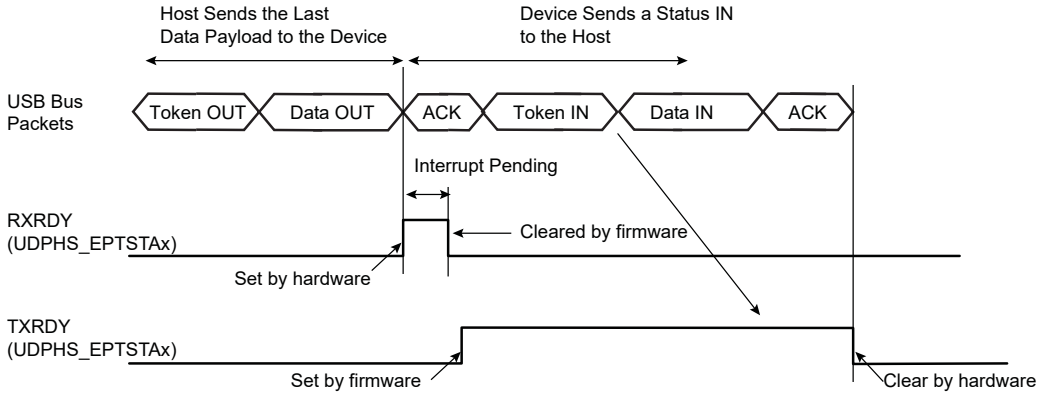
Downloaded from Arrow.com.

Figure 40-10. Data IN Followed By Status OUT Transfer at the End of a Control Transfer



Note: A NAK handshake is always generated at the first status stage token.

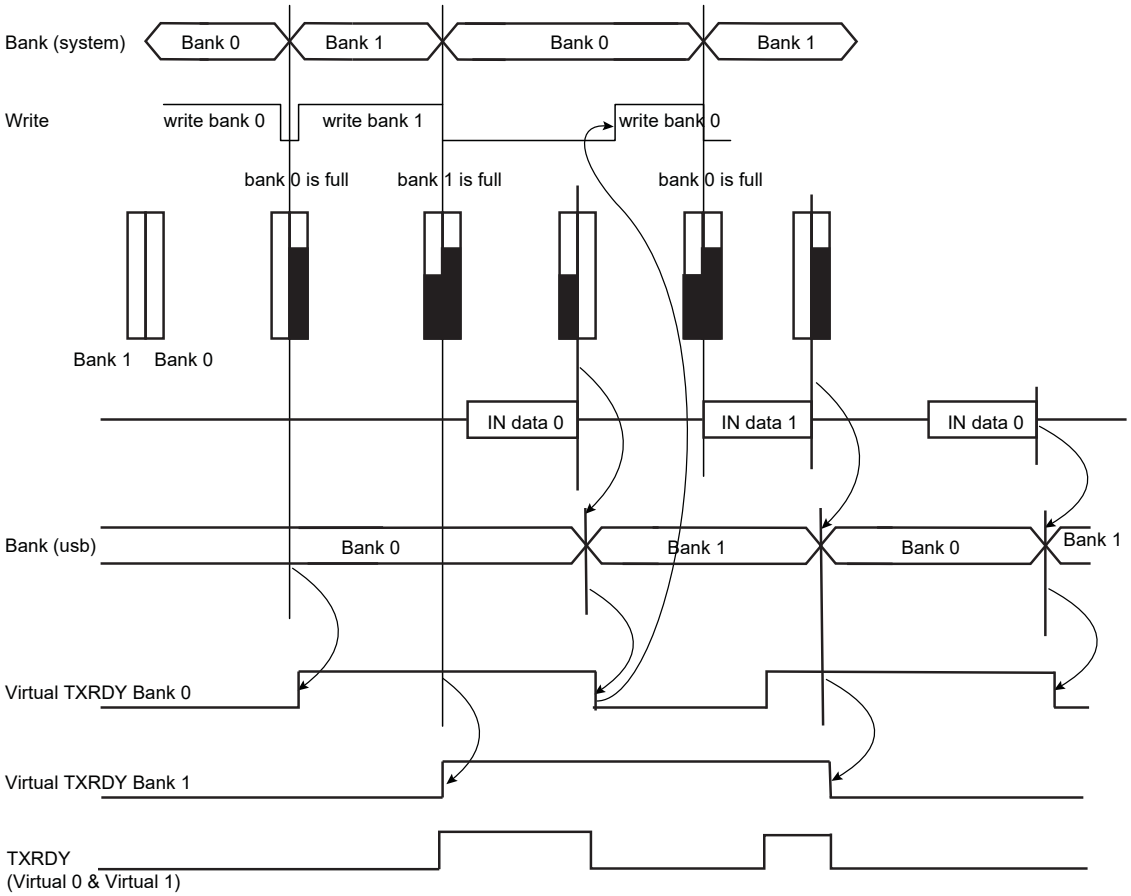
Figure 40-11. Data OUT Followed by Status IN Transfer



Note: Before proceeding to the status stage, the software should determine that there is no risk of extra data from the host (data stage). If not certain (non-predictable data stage length), then the software should wait for a NAK-IN interrupt before proceeding to the status stage. This precaution should be taken to avoid collision in the FIFO.

Downloaded from Arrow.com.

Figure 40-12. Autovalid with DMA



Note: In the illustration above Autovalid validates a bank as full, although this might not be the case, in order to continue processing data and to send to DMA.

• Isochronous IN

Isochronous-IN is used to transmit a stream of data whose timing is implied by the delivery rate. Isochronous transfer provides periodic, continuous communication between host and device.

It guarantees bandwidth and low latencies appropriate for telephony, audio, video, etc.

If the endpoint is not available (TXRDY_TRER = 0), then the device does not answer to the host. An ERR_FL_ISO interrupt is generated in the UDPHS_EPTSTAx register and once enabled, then sent to the CPU.

The STALL_SNT command bit is not used for an ISO-IN endpoint.

• High Bandwidth Isochronous Endpoint Handling: IN Example

For high bandwidth isochronous endpoints, the DMA can be programmed with the number of transactions (BUFF_LENGTH field in UDPHS_DMACONTROLx) and the system should provide the required number of packets per microframe, otherwise, the host will notice a sequencing problem.

A response should be made to the first token IN recognized inside a microframe under the following conditions:

- If at least one bank has been validated, the correct DATAx corresponding to the programmed Number Of Transactions per Microframe (NB_TRANS) should be answered. In case of a subsequent missed or corrupted token IN inside the microframe, the USB 2.0 Core available data bank(s) that should normally have been transmitted during that microframe shall be flushed at its end. If this flush occurs, an error condition is flagged (ERR_FLUSH is set in UDPHS_EPTSTAx).
- If no bank is validated yet, the default DATA0 ZLP is answered and underflow is flagged (ERR_FL_ISO is set in UDPHS_EPTSTAx). Then, no data bank is flushed at microframe end.

Downloaded from Arrow.com.

- If no data bank has been validated at the time when a response should be made for the second transaction of NB_TRANS = 3 transactions microframe, a DATA1 ZLP is answered and underflow is flagged (ERR_FL_ISO is set in UDPHS_EPTSTAx). If and only if remaining untransmitted banks for that microframe are available at its end, they are flushed and an error condition is flagged (ERR_FLUSH is set in UDPHS_EPTSTAx).
- If no data bank has been validated at the time when a response should be made for the last programmed transaction of a microframe, a DATA0 ZLP is answered and underflow is flagged (ERR_FL_ISO is set in UDPHS_EPTSTAx). If and only if the remaining untransmitted data bank for that microframe is available at its end, it is flushed and an error condition is flagged (ERR_FLUSH is set in UDPHS_EPTSTAx).
- If at the end of a microframe no valid token IN has been recognized, no data bank is flushed and no error condition is reported.

At the end of a microframe in which at least one data bank has been transmitted, if less than NB_TRANS banks have been validated for that microframe, an error condition is flagged (ERR_TRANS is set in UDPHS_EPTSTAx).

Cases of Error (in UDPHS_EPTSTAx)

- ERR_FL_ISO: There was no data to transmit inside a microframe, so a ZLP is answered by default.
- ERR_FLUSH: At least one packet has been sent inside the microframe, but the number of token INs received is less than the number of transactions actually validated (TXRDY_TRER) and likewise with the NB_TRANS programmed.
- ERR_TRANS: At least one packet has been sent inside the microframe, but the number of token INs received is less than the number of programmed NB_TRANS transactions and the packets not requested were not validated.
- ERR_FL_ISO + ERR_FLUSH: At least one packet has been sent inside the microframe, but the data has not been validated in time to answer one of the following token INs.
- ERR_FL_ISO + ERR_TRANS: At least one packet has been sent inside the microframe, but the data has not been validated in time to answer one of the following token INs and the data can be discarded at the microframe end.
- ERR_FLUSH + ERR_TRANS: The first token IN has been answered and it was the only one received, a second bank has been validated but not the third, whereas NB_TRANS was waiting for three transactions.
- ERR_FL_ISO + ERR_FLUSH + ERR_TRANS: The first token IN has been treated, the data for the second Token IN was not available in time, but the second bank has been validated before the end of the microframe. The third bank has not been validated, but three transactions have been set in NB_TRANS.

40.6.10.4 Data OUT

• Bulk OUT or Interrupt OUT

Like data IN, data OUT packets are sent by the host during the data or the status stage of control transfer or during an interrupt/bulk/isochronous OUT transfer. Data buffers are sent packet by packet under the control of the application or under the control of the DMA channel.

• Bulk OUT or Interrupt OUT: Receiving a Packet Under Application Control (Host to Device)

Algorithm Description for Each Packet:

- The application enables an interrupt on RXRDY_TXKL.
- When an interrupt on RXRDY_TXKL is received, the application knows that UDPHS_EPTSTAx register BYTE_COUNT bytes have been received.
- The application reads the BYTE_COUNT bytes from the endpoint.
- The application clears RXRDY_TXKL.
  Note: If the application does not know the size of the transfer, it may not be a good option to use AUTO_VALID. Because if a zero-length-packet is received, the RXRDY_TXKL is automatically cleared by the AUTO_VALID hardware and if the endpoint interrupt is triggered, the software will not find its originating flag when reading the UDPHS_EPTSTAx register.

Algorithm to Fill Several Packets

- The application enables the interrupts of BUSY_BANK and AUTO_VALID.
- When a BUSY_BANK interrupt is received, the application knows that all banks available for the endpoint have been filled. Thus, the application can read all banks available.

Downloaded from Arrow.com.

If the application does not know the size of the receive buffer, instead of using the BUSY_BANK interrupt, the application must use RXRDY_TXKL.

• Bulk OUT or Interrupt OUT: Sending a Buffer Using DMA (Host To Device)

To use the DMA setting, the AUTO_VALID field is mandatory.

See Bulk IN or Interrupt IN: Sending a Buffer Using DMA (Device to Host) for more information.

DMA Configuration Example:

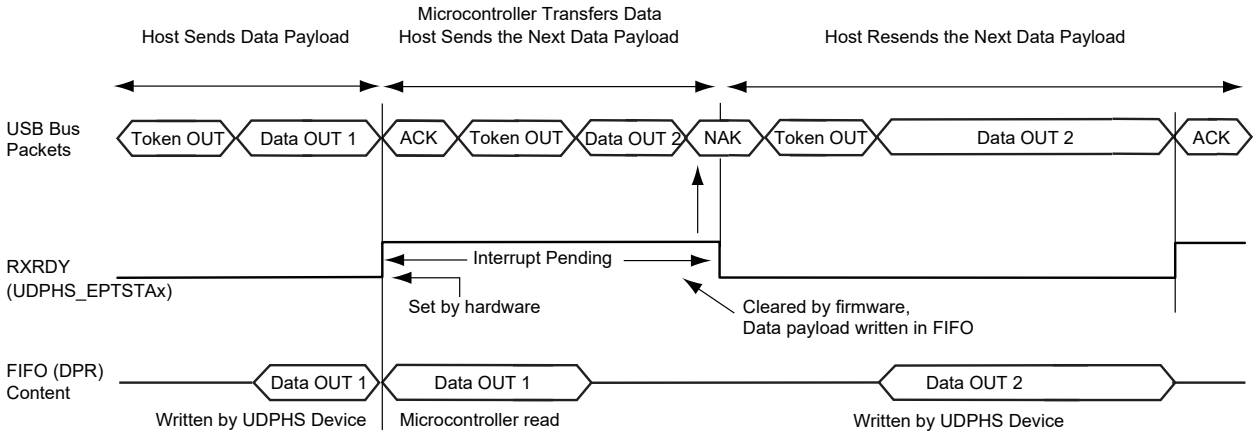
1. First program UDPHS_DMAADDRESSx with the address of the buffer that should be transferred.
2. Enable the interrupt of the DMA in UDPHS_IEN
3. Program the DMA Channelx Control Register:
   – Size of buffer to be sent.
   – END_B_EN: Can be used for OUT packet truncation (discarding of unbuffered packet data) at the end of DMA buffer.
   – END_BUFFIT: Generate an interrupt when BUFF_COUNT in the UDPHS_DMASTATUSx register reaches 0.
   – END_TR_EN: End of transfer enable, the UDPHS device can put an end to the current DMA transfer, in case of a short packet.
   – END_TR_IT: End of transfer interrupt enable, an interrupt is sent after the last USB packet has been transferred by the DMA, if the USB transfer ended with a short packet. (Beneficial when the receive size is unknown.)
   – CHANN_ENB: Run and stop at end of buffer.

For OUT transfer, the bank will be automatically cleared by hardware when the application has read all the bytes in the bank (the bank is empty).

Note:

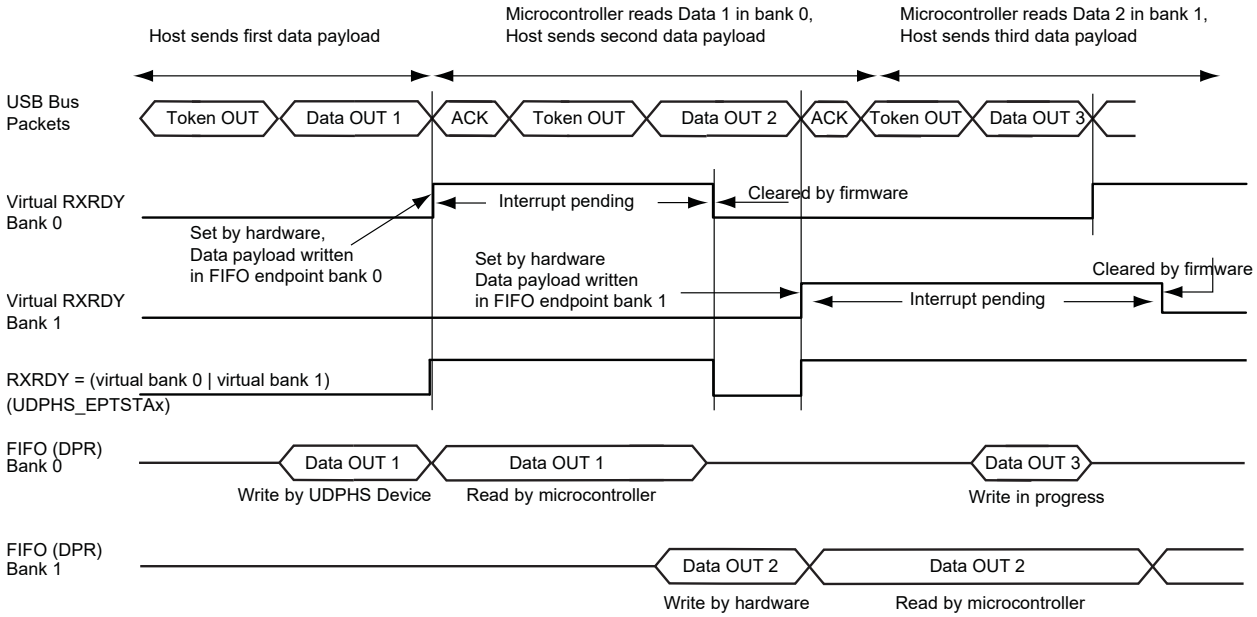
1. When a zero-length-packet is received, RXRDY_TXKL bit in UDPHS_EPTSTAx is cleared automatically by AUTO_VALID, and the application knows of the end of buffer by the presence of the END_TR_IT.
2. If the host sends a zero-length packet, and the endpoint is free, then the device sends an ACK. No data is written in the endpoint, the RXRDY_TXKL interrupt is generated, and the BYTE_COUNT field in UDPHS_EPTSTAx is null.

Figure 40-13. Data OUT Transfer for Endpoint with One Bank



Downloaded from Arrow.com.

Figure 40-14. Data OUT Transfer for an Endpoint with Two Banks



• High Bandwidth Isochronous Endpoint OUT

USB 2.0 supports individual High Speed isochronous endpoints that require data rates up to 192 Mb/s (24 MB/s): 3x1024 data bytes per microframe.

To support such a rate, two or three banks may be used to buffer the three consecutive data packets. The microcontroller (or the DMA) should be able to empty the banks very rapidly (at least 24 MB/s on average).

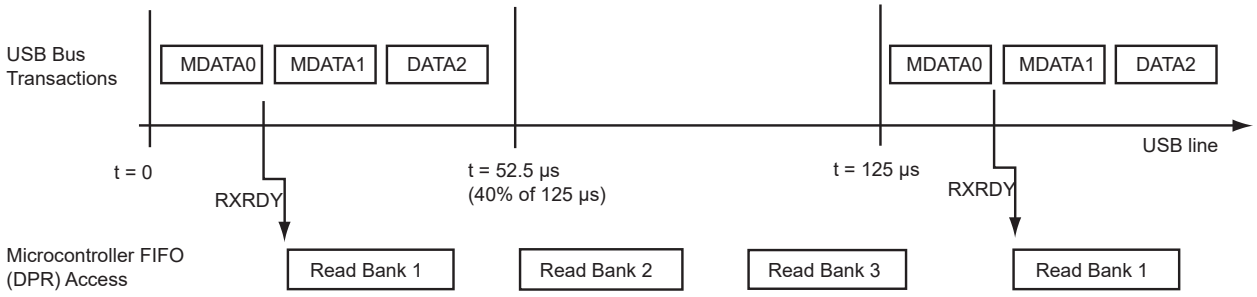
NB_TRANS field in UDPHS_EPTCFGx register = Number Of Transactions per Microframe.

If NB_TRANS > 1 then it is High Bandwidth.

Example:

- If NB_TRANS = 3, the sequence should be either
  - MData0
  - MData0/Data1
  - MData0/Data1/Data2
- If NB_TRANS = 2, the sequence should be either
  - MData0
  - MData0/Data1
- If NB_TRANS = 1, the sequence should be
  - Data0

Figure 40-15. Bank Management, Example of Three Transactions per Microframe



• Isochronous Endpoint Handling: OUT Example

Downloaded from Arrow.com.

The user can ascertain the bank status (free or busy), and the toggle sequencing of the data packet for each bank with the UDPHS_EPTSTAx register in the three fields as follows:

- TOGGLESQ_STA: PID of the data stored in the current bank
- CURBK: Number of the bank currently being accessed by the microcontroller.
- BUSY_BANK_STA: Number of busy bank

This is particularly useful in case of a missing data packet.

If the inter-packet delay between the OUT token and the Data is greater than the USB standard, then the ISO-OUT transaction is ignored. (Payload data is not written, no interrupt is generated to the CPU.)

If there is a data CRC (Cyclic Redundancy Check) error, the payload is, none the less, written in the endpoint. The ERR_CRC_NTR flag is set in UDPHS_EPTSTAx register.

If the endpoint is already full, the packet is not written in the DPRAM. The ERR_FL_ISO flag is set in UDPHS_EPTSTAx.

If the payload data is greater than the maximum size of the endpoint, then the ERR_OVFLW flag is set. It is the task of the CPU to manage this error. The data packet is written in the endpoint (except the extra data).

If the host sends a Zero Length Packet, and the endpoint is free, no data is written in the endpoint, the RXRDY_TXKL flag is set, and the BYTE_COUNT field in UDPHS_EPTSTAx register is null.

The FRCESTALL command bit is unused for an isochronous endpoint.

Otherwise, payload data is written in the endpoint, the RXRDY_TXKL interrupt is generated and the BYTE_COUNT in UDPHS_EPTSTAx register is updated.

40.6.10.5 STALL

STALL is returned by a function in response to an IN token or after the data phase of an OUT or in response to a PING transaction. STALL indicates that a function is unable to transmit or receive data, or that a control pipe request is not supported.

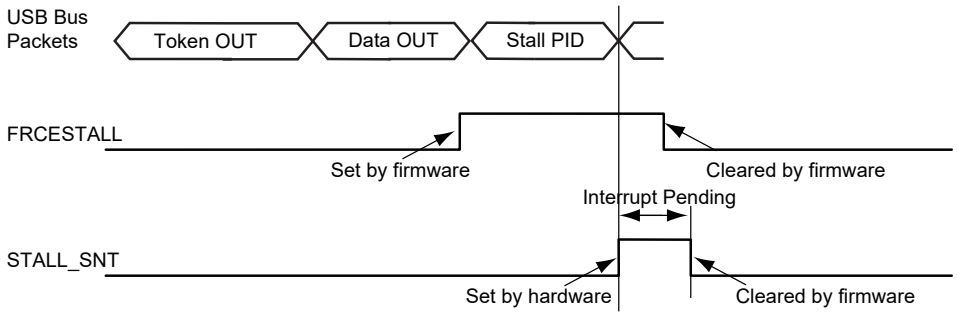
- OUT

To stall an endpoint, set the FRCESTALL bit in UDPHS_EPTSETSTAx register and after the STALL_SNT flag has been set, set the TOGGLE_SEG bit in the UDPHS_EPTCLRSTAx register.

- IN

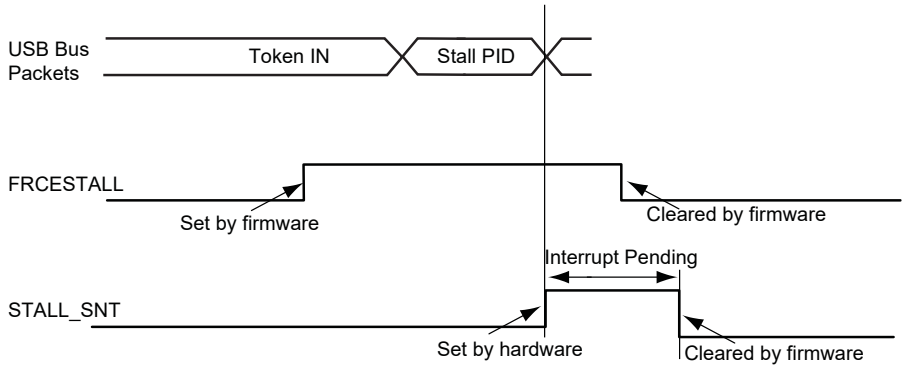
Set the FRCESTALL bit in UDPHS_EPTSETSTAx register.

Figure 40-16. Stall Handshake Data OUT Transfer



Downloaded from Arrow.com.

Figure 40-17. Stall Handshake Data IN Transfer



40.6.11 Speed Identification

The high speed reset is managed by hardware.

At the connection, the host makes a reset which could be a classic reset (full speed) or a high speed reset.

At the end of the reset process (full or high), the ENDRESET interrupt is generated.

Then the CPU should read the SPEED bit in UDPHS_INTSTAx to ascertain the speed mode of the device.

40.6.12 USB V2.0 High Speed Global Interrupt

Interrupts are defined in UDPHS Interrupt Enable Register (UDPHS_IEN) and in UDPHS Interrupt Status Register (UDPHS_INTSTA).

40.6.13 Endpoint Interrupts

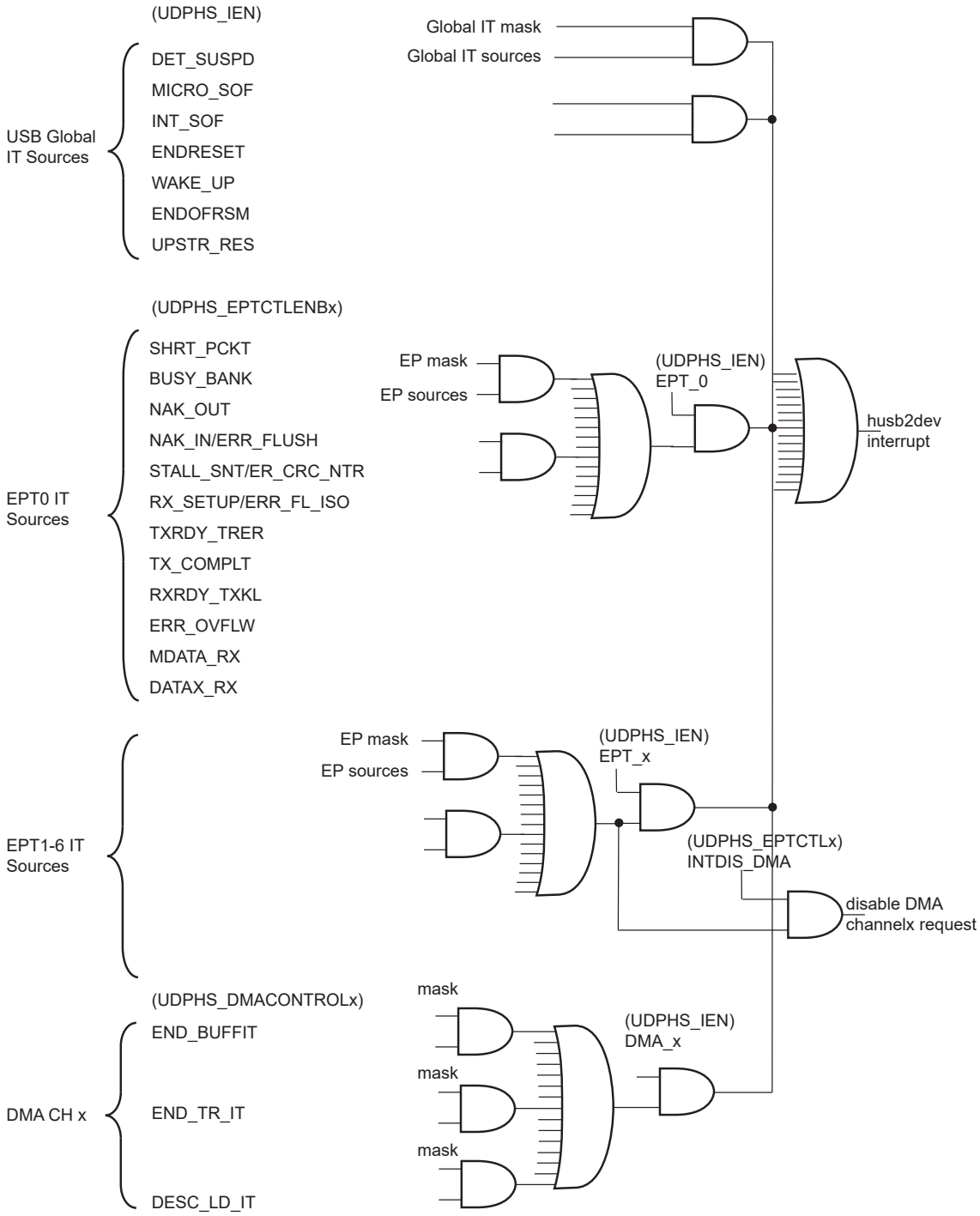
Interrupts are enabled in UDPHS_IEN (see UDPHS Interrupt Enable Register) and individually masked in UDPHS_EPTCTLENBx (see UDPHS Endpoint Control Enable Register (Control, Bulk, Interrupt Endpoints)).

Table 40-4. Endpoint Interrupt Source Masks

SHRT_PCKT	Short Packet Interrupt
BUSY_BANK	Busy Bank Interrupt
NAK_OUT	NAKOUT Interrupt
NAK_IN/ERR_FLUSH	NAKIN/Error Flush Interrupt
STALL_SNT/ERR_CRC_NTR	Stall Sent/CRC error/Number of Transaction Error Interrupt
RX_SETUP/ERR_FL_ISO	Received SETUP/Error Flow Interrupt
TXRDY_TRER	TX Packet Read/Transaction Error Interrupt
TX_COMPLT	Transmitted IN Data Complete Interrupt
RXRDY_TXKL	Received OUT Data Interrupt
ERR_OVFLW	Overflow Error Interrupt
MDATA_RX	MDATA Interrupt
DATAX_RX	DATAx Interrupt

Downloaded from Arrow.com.

Figure 40-18. UDPHS Interrupt Control Interface



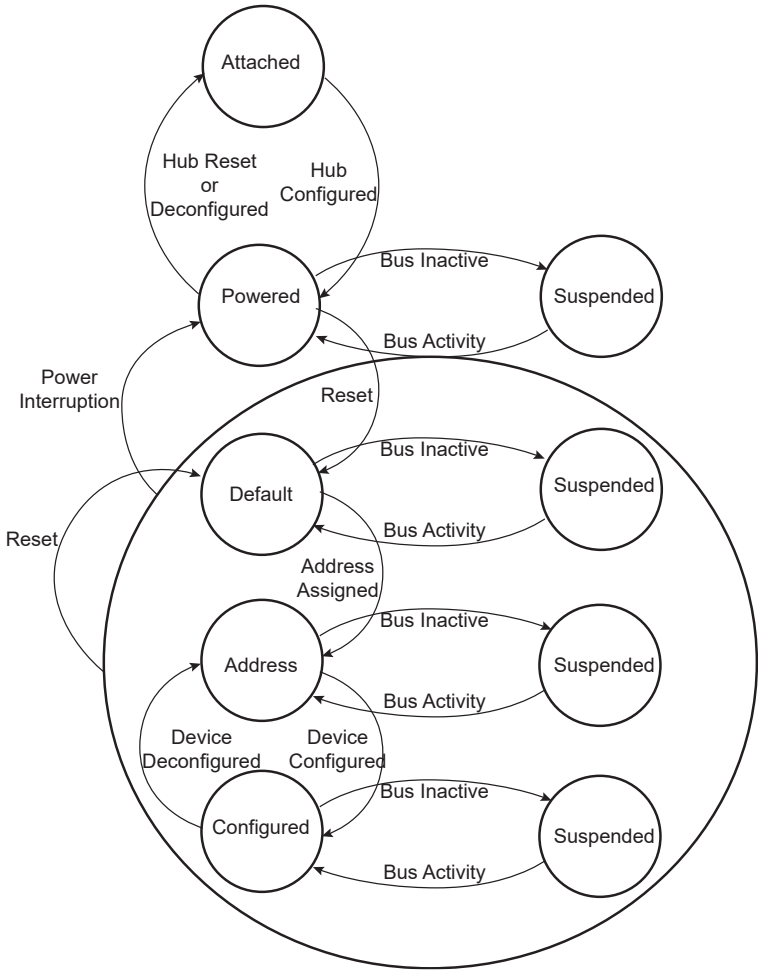
40.6.14 Power Modes

40.6.14.1 Controlling Device States

A USB device has several possible states. Refer to Chapter 9 (USB Device Framework) of the Universal Serial Bus Specification, Rev 2.0.

Downloaded from Arrow.com.

Figure 40-19. UDPHS Device State Diagram



Movement from one state to another depends on the USB bus state or on standard requests sent through control transactions via the default endpoint (endpoint 0).

After a period of bus inactivity, the USB device enters Suspend mode. Accepting Suspend/Resume requests from the USB host is mandatory. Constraints in Suspend mode are very strict for bus-powered applications; devices may not consume more than 500 µA on the USB bus.

While in Suspend mode, the host may wake up a device by sending a resume signal (bus activity) or a USB device may send a wakeup request to the host, e.g., waking up a PC by moving a USB mouse.

The wakeup feature is not mandatory for all devices and must be negotiated with the host.

40.6.14.2 Not Powered State

Self powered devices can detect 5V VBUS using a PIO. When the device is not connected to a host, device power consumption can be reduced by the DETACH bit in UDPHS_CTRL. Disabling the transceiver is automatically done. HSDM, HSDP, FSDP and FSDP lines are tied to GND pulldowns integrated in the hub downstream ports.

40.6.14.3 Entering Attached State

When no device is connected, the USB FSDP and FSDM signals are tied to GND by 15 KΩ pulldowns integrated in the hub downstream ports. When a device is attached to an hub downstream port, the device connects a 1.5 KΩ pullup on FSDP. The USB bus line goes into IDLE state, FSDP is pulled up by the device 1.5 KΩ resistor to 3.3V and FSDM is pulled down by the 15 KΩ resistor to GND of the host.

After pullup connection, the device enters the powered state. The transceiver remains disabled until bus activity is detected.

Downloaded from Arrow.com.

In case of low power consumption need, the device can be stopped. When the device detects the VBUS, the software must enable the USB transceiver by enabling the EN_UDPHS bit in UDPHS_CTRL register.

The software can detach the pullup by setting DETACH bit in UDPHS_CTRL register.

40.6.14.4 From Powered State to Default State (Reset)

After its connection to a USB host, the USB device waits for an end-of-bus reset. The unmasked flag ENDRESET is set in the UDPHS_IEN register and an interrupt is triggered.

Once the ENDRESET interrupt has been triggered, the device enters Default State. In this state, the UDPHS software must:

- Enable the default endpoint, setting the EPT_ENABL flag in the UDPHS_EPTCTLENB[0] register and, optionally, enabling the interrupt for endpoint 0 by writing 1 in EPT_0 of the UDPHS_IEN register. The enumeration then begins by a control transfer.
- Configure the Interrupt Mask Register which has been reset by the USB reset detection
- Enable the transceiver.

In this state, the EN_UDPHS bit in UDPHS_CTRL register must be enabled.

40.6.14.5 From Default State to Address State (Address Assigned)

After a Set Address standard device request, the USB host peripheral enters the address state.

WARNING Before the device enters address state, it must achieve the Status IN transaction of the control transfer, i.e., the UDPHS device sets its new address once the TX_COMPLT flag in the UDPHS_EPTCTL[0] register has been received and cleared.

To move to address state, the driver software sets the DEV_ADDR field and the FADDR_EN flag in the UDPHS_CTRL register.

40.6.14.6 From Address State to Configured State (Device Configured)

Once a valid Set Configuration standard request has been received and acknowledged, the device enables endpoints corresponding to the current configuration. This is done by setting the BK_NUMBER, EPT_TYPE, EPT_DIR and EPT_SIZE fields in the UDPHS_EPTCFGx registers and enabling them by setting the EPT_ENABL flag in the UDPHS_EPTCTLENBx registers, and, optionally, enabling corresponding interrupts in the UDPHS_IEN register.

40.6.14.7 Entering Suspend State (Bus Activity)

When a Suspend (no bus activity on the USB bus) is detected, the DET_SUSPD signal in the UDPHS_STA register is set. This triggers an interrupt if the corresponding bit is set in the UDPHS_IEN register. This flag is cleared by writing to the UDPHS_CLRINT register. Then the device enters Suspend mode.

In this state bus powered devices must drain less than 500 µA from the 5V VBUS. As an example, the microcontroller switches to slow clock, disables the PLL and main oscillator, and goes into Idle mode. It may also switch off other devices on the board.

The UDPHS device peripheral clocks can be switched off. Resume event is asynchronously detected.

40.6.14.8 Receiving a Host Resume

In Suspend mode, a resume event on the USB bus line is detected asynchronously, transceiver and clocks disabled (however, the pullup should not be removed).

Once the resume is detected on the bus, the signal WAKE_UP in the UDPHS_INTSTA is set. It may generate an interrupt if the corresponding bit in the UDPHS_IEN register is set. This interrupt may be used to wake up the core, enable PLL and main oscillators and configure clocks.

40.6.14.9 Sending an External Resume

In Suspend State it is possible to wake up the host by sending an external resume.

The device waits at least 5 ms after being entered in Suspend State before sending an external resume.

The device must force a K state from 1 to 15 ms to resume the host.

Downloaded from Arrow.com.

40.6.15 Test Mode

A device must support the TEST_MODE feature when in the Default, Address or Configured High Speed device states.

TEST_MODE can be:

- Test_J
- Test_K
- Test_Packet
- Test_SEO_NAK

(See UDPHS Test Register for definitions of each test mode.)

```
const char test_packet_buffer[] = {
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,                 // JKJKJKJK * 9
0xAA,0xAA,0xAA,0xAA,0xAA,0xAA,0xAA,0xAA,                      // JJKKJJKK * 8
0xEE,0xEE,0xEE,0xEE,0xEE,0xEE,0xEE,0xEE,                      // JJKKJJKK * 8
0xFE,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,  // JJJJJJJKKKKKKK * 8
0x7F,0xBF,0xDF,0xEF,0xF7,0xFB,0xFD,                           // JJJJJJJK * 8
0xFC,0x7E,0xBF,0xDF,0xEF,0xF7,0xFB,0xFD,0x7E                  // {JKKKKKKK * 10}, JK
};
```


Downloaded from Arrow.com.

40.7 Register Summary

Note: The registers below have two modes: Control, Bulk, Interrupt Endpoints mode and Isochronous Endpoints mode. In this register summary, both modes are displayed at the same offset.

- UDPHS_EPTCTLENB
- UDPHS_EPTCTLDIS
- UDPHS_EPTCTL
- UDPHS_EPTSETSTA
- UDPHS_EPTCLRSTA
- UDPHS_EPTSTA

Offset	Name	Bit Pos.								
0x00	UDPHS_CTRL	31:24								
		23:16								
		15:8					PULLD_DIS	REWAKEUP	DETACH	EN_UDPHS
		7:0	FADDR_EN	DEV_ADDR[6:0]						
0x04	UDPHS_FNUM	31:24	FNUM_ERR							
		23:16								
		15:8			FRAME_NUMBER[10:5]					
		7:0	FRAME_NUMBER[4:0]					MICRO_FRAME_NUM[2:0]		
0x08 ... 0x0F	Reserved									
0x10	UDPHS_IEN	31:24	DMA_7	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	
		23:16	EPT_15	EPT_14	EPT_13	EPT_12	EPT_11	EPT_10	EPT_9	EPT_8
		15:8	EPT_7	EPT_6	EPT_5	EPT_4	EPT_3	EPT_2	EPT_1	EPT_0
		7:0	UPSTR_RES	ENDOFRSM	WAKE_UP	ENDRESET	INT_SOF	MICRO_SOF	DET_SUSPD	
0x14	UDPHS_INTSTA	31:24	DMA_7	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	
		23:16	EPT_15	EPT_14	EPT_13	EPT_12	EPT_11	EPT_10	EPT_9	EPT_8
		15:8	EPT_7	EPT_6	EPT_5	EPT_4	EPT_3	EPT_2	EPT_1	EPT_0
		7:0	UPSTR_RES	ENDOFRSM	WAKE_UP	ENDRESET	INT_SOF	MICRO_SOF	DET_SUSPD	SPEED
0x18	UDPHS_CLRINT	31:24								
		23:16								
		15:8								
		7:0	UPSTR_RES	ENDOFRSM	WAKE_UP	ENDRESET	INT_SOF	MICRO_SOF	DET_SUSPD	
0x1C	UDPHS_EPTRST	31:24								
		23:16								
		15:8	EPT_15	EPT_14	EPT_13	EPT_12	EPT_11	EPT_10	EPT_9	EPT_8
		7:0	EPT_7	EPT_6	EPT_5	EPT_4	EPT_3	EPT_2	EPT_1	EPT_0
0x20 ... 0xDF	Reserved									
0xE0	UDPHS_TST	31:24								
		23:16								
		15:8								
		7:0			OPMODE2	TST_PKT	TST_K	TST_J	SPEED_CFG[1:0]	
0xE4 ... 0xFF	Reserved									
0x0100	UDPHS_EPTCFG0	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x0104	UDPHS_EPTCTLENB0	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x0104	UDPHS_EPTCTLENB0	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0108	UDPHS_EPTCTLDIS0	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x0108	UDPHS_EPTCTLDIS0	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x010C	UDPHS_EPTCTL0	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x010C	UDPHS_EPTCTL0	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0110 ... 0x0113	Reserved									
0x0114	UDPHS_EPTSETSTA0	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x0114	UDPHS_EPTSETSTA0	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x0118	UDPHS_EPTCLRSTA0	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x0118	UDPHS_EPTCLRSTA0	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x011C	UDPHS_EPTSTA0	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x011C	UDPHS_EPTSTA0	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x0120	UDPHS_EPTCFG1	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x0124	UDPHS_EPTCTLENB1	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0124	UDPHS_EPTCTLENB1	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0128	UDPHS_EPTCTLDIS1	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x0128	UDPHS_EPTCTLDIS1	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x012C	UDPHS_EPTCTL1	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x012C	UDPHS_EPTCTL1	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0130 ... 0x0133	Reserved									
0x0134	UDPHS_EPTSETSTA1	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x0134	UDPHS_EPTSETSTA1	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x0138	UDPHS_EPTCLRSTA1	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x0138	UDPHS_EPTCLRSTA1	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x013C	UDPHS_EPTSTA1	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x013C	UDPHS_EPTSTA1	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x0140	UDPHS_EPTCFG2	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x0144	UDPHS_EPTCTLENB2	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0144	UDPHS_EPTCTLENB2	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0148	UDPHS_EPTCTLDIS2	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x0148	UDPHS_EPTCTLDIS2	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x014C	UDPHS_EPTCTL2	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x014C	UDPHS_EPTCTL2	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0150 ... 0x0153	Reserved									
0x0154	UDPHS_EPTSETSTA2	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x0154	UDPHS_EPTSETSTA2	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x0158	UDPHS_EPTCLRSTA2	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x0158	UDPHS_EPTCLRSTA2	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x015C	UDPHS_EPTSTA2	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x015C	UDPHS_EPTSTA2	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x0160	UDPHS_EPTCFG3	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x0164	UDPHS_EPTCTLENB3	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0164	UDPHS_EPTCTLENB3	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0168	UDPHS_EPTCTLDIS3	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x0168	UDPHS_EPTCTLDIS3	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x016C	UDPHS_EPTCTL3	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x016C	UDPHS_EPTCTL3	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0170 ... 0x0173	Reserved									

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x0174	UDPHS_EPTSETSTA3	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x0174	UDPHS_EPTSETSTA3	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x0178	UDPHS_EPTCLRSTA3	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x0178	UDPHS_EPTCLRSTA3	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x017C	UDPHS_EPTSTA3	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x017C	UDPHS_EPTSTA3	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x0180	UDPHS_EPTCFG4	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x0184	UDPHS_EPTCTLENB4	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0184	UDPHS_EPTCTLENB4	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0188	UDPHS_EPTCTLDIS4	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x0188	UDPHS_EPTCTLDIS4	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x018C	UDPHS_EPTCTL4	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x018C	UDPHS_EPTCTL4	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x0190 ... 0x0193	Reserved									
0x0194	UDPHS_EPTSETSTA4	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x0194	UDPHS_EPTSETSTA4	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x0198	UDPHS_EPTCLRSTA4	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x0198	UDPHS_EPTCLRSTA4	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x019C	UDPHS_EPTSTA4	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x019C	UDPHS_EPTSTA4	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x01A0	UDPHS_EPTCFG5	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x01A4	UDPHS_EPTCTLENB5	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x01A4	UDPHS_EPTCTLENB5	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x01A8	UDPHS_EPTCTLDIS5	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x01A8	UDPHS_EPTCTLDIS5	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x01AC	UDPHS_EPTCTL5	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x01AC	UDPHS_EPTCTL5	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x01B0 ... 0x01B3	Reserved									
0x01B4	UDPHS_EPTSETSTA5	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x01B4	UDPHS_EPTSETSTA5	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x01B8	UDPHS_EPTCLRSTA5	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x01B8	UDPHS_EPTCLRSTA5	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x01BC	UDPHS_EPTSTA5	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x01BC	UDPHS_EPTSTA5	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							
0x01C0	UDPHS_EPTCFG6	31:24	EPT_MAPD							
		23:16								
		15:8							NB_TRANS[1:0]	
		7:0	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
0x01C4	UDPHS_EPTCTLENB6	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x01C4	UDPHS_EPTCTLENB6	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x01C8	UDPHS_EPTCTLDIS6	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x01C8	UDPHS_EPTCTLDIS6	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
0x01CC	UDPHS_EPTCTL6	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x01CC	UDPHS_EPTCTL6	31:24	SHRT_PCKT							
		23:16						BUSY_BANK		
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
0x01D0 ... 0x01D3	Reserved									
0x01D4	UDPHS_EPTSETSTA6	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					
0x01D4	UDPHS_EPTSETSTA6	31:24								
		23:16								
		15:8					TXRDY_TRER		RXRDY_TXKL	
		7:0								
0x01D8	UDPHS_EPTCLRSTA6	31:24								
		23:16								
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ	FRCESTALL					
0x01D8	UDPHS_EPTCLRSTA6	31:24								
		23:16								
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
		7:0		TOGGLESQ						
0x01DC	UDPHS_EPTSTA6	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
		15:8	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]		FRCESTALL					
0x01DC	UDPHS_EPTSTA6	31:24	SHRT_PCKT	BYTE_COUNT[10:4]						
		23:16	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
		15:8		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
		7:0	TOGGLESQ_STA[1:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.									
0x01E0 ... 0x030F	Reserved										
0x0310	UDPHS_DMANXTDSC1	31:24	NXT_DSC_ADD[31:24]								
		23:16	NXT_DSC_ADD[23:16]								
		15:8	NXT_DSC_ADD[15:8]								
		7:0	NXT_DSC_ADD[7:0]								
0x0314	UDPHS_DMAADDRESS1	31:24	BUFF_ADD[31:24]								
		23:16	BUFF_ADD[23:16]								
		15:8	BUFF_ADD[15:8]								
		7:0	BUFF_ADD[7:0]								
0x0318	UDPHS_DMACONTROL1	31:24	BUFF_LENGTH[15:8]								
		23:16	BUFF_LENGTH[7:0]								
		15:8									
		7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB	
0x031C	UDPHS_DMASTATUS1	31:24	BUFF_COUNT[15:8]								
		23:16	BUFF_COUNT[7:0]								
		15:8									
		7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB	
0x0320	UDPHS_DMANXTDSC2	31:24	NXT_DSC_ADD[31:24]								
		23:16	NXT_DSC_ADD[23:16]								
		15:8	NXT_DSC_ADD[15:8]								
		7:0	NXT_DSC_ADD[7:0]								
0x0324	UDPHS_DMAADDRESS2	31:24	BUFF_ADD[31:24]								
		23:16	BUFF_ADD[23:16]								
		15:8	BUFF_ADD[15:8]								
		7:0	BUFF_ADD[7:0]								
0x0328	UDPHS_DMACONTROL2	31:24	BUFF_LENGTH[15:8]								
		23:16	BUFF_LENGTH[7:0]								
		15:8									
		7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB	
0x032C	UDPHS_DMASTATUS2	31:24	BUFF_COUNT[15:8]								
		23:16	BUFF_COUNT[7:0]								
		15:8									
		7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB	
0x0330	UDPHS_DMANXTDSC3	31:24	NXT_DSC_ADD[31:24]								
		23:16	NXT_DSC_ADD[23:16]								
		15:8	NXT_DSC_ADD[15:8]								
		7:0	NXT_DSC_ADD[7:0]								
0x0334	UDPHS_DMAADDRESS3	31:24	BUFF_ADD[31:24]								
		23:16	BUFF_ADD[23:16]								
		15:8	BUFF_ADD[15:8]								
		7:0	BUFF_ADD[7:0]								
0x0338	UDPHS_DMACONTROL3	31:24	BUFF_LENGTH[15:8]								
		23:16	BUFF_LENGTH[7:0]								
		15:8									
		7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB	
0x033C	UDPHS_DMASTATUS3	31:24	BUFF_COUNT[15:8]								
		23:16	BUFF_COUNT[7:0]								
		15:8									
		7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB	
0x0340	UDPHS_DMANXTDSC4	31:24	NXT_DSC_ADD[31:24]								
		23:16	NXT_DSC_ADD[23:16]								
		15:8	NXT_DSC_ADD[15:8]								
		7:0	NXT_DSC_ADD[7:0]								
0x0344	UDPHS_DMAADDRESS4	31:24	BUFF_ADD[31:24]								
		23:16	BUFF_ADD[23:16]								
		15:8	BUFF_ADD[15:8]								
		7:0	BUFF_ADD[7:0]								

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x0348	UDPHS_DMACONTROL4	31:24	BUFF_LENGTH[15:8]							
		23:16	BUFF_LENGTH[7:0]							
		15:8								
		7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB
0x034C	UDPHS_DMASTATUS4	31:24	BUFF_COUNT[15:8]							
		23:16	BUFF_COUNT[7:0]							
		15:8								
		7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB
0x0350	UDPHS_DMANXTDSC5	31:24	NXT_DSC_ADD[31:24]							
		23:16	NXT_DSC_ADD[23:16]							
		15:8	NXT_DSC_ADD[15:8]							
		7:0	NXT_DSC_ADD[7:0]							
0x0354	UDPHS_DMAADDRESS5	31:24	BUFF_ADD[31:24]							
		23:16	BUFF_ADD[23:16]							
		15:8	BUFF_ADD[15:8]							
		7:0	BUFF_ADD[7:0]							
0x0358	UDPHS_DMACONTROL5	31:24	BUFF_LENGTH[15:8]							
		23:16	BUFF_LENGTH[7:0]							
		15:8								
		7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB
0x035C	UDPHS_DMASTATUS5	31:24	BUFF_COUNT[15:8]							
		23:16	BUFF_COUNT[7:0]							
		15:8								
		7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB
0x0360	UDPHS_DMANXTDSC6	31:24	NXT_DSC_ADD[31:24]							
		23:16	NXT_DSC_ADD[23:16]							
		15:8	NXT_DSC_ADD[15:8]							
		7:0	NXT_DSC_ADD[7:0]							
0x0364	UDPHS_DMAADDRESS6	31:24	BUFF_ADD[31:24]							
		23:16	BUFF_ADD[23:16]							
		15:8	BUFF_ADD[15:8]							
		7:0	BUFF_ADD[7:0]							
0x0368	UDPHS_DMACONTROL6	31:24	BUFF_LENGTH[15:8]							
		23:16	BUFF_LENGTH[7:0]							
		15:8								
		7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB
0x036C	UDPHS_DMASTATUS6	31:24	BUFF_COUNT[15:8]							
		23:16	BUFF_COUNT[7:0]							
		15:8								
		7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB

Downloaded from Arrow.com.

40.7.1 UDPHS Control Register

Name: UDPHS_CTRL
Offset: 0x00
Reset: 0x00000200
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					PULLD_DIS	REWAKEUP	DETACH	EN_UDPHS
Access					R/W	R/W	R/W	R/W
Reset					0	0	1	0

Bit	7	6	5	4	3	2	1	0
	FADDR_EN	DEV_ADDR[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 11 – PULLD_DIS Pulldown Disable (cleared upon USB reset)
When set, there is no pulldown on DP & DM. (DM Pulldown = DP Pulldown = 0).
Note: If the DETACH bit is also set, device DP & DM are left in high impedance state.

(See description of bit “DETACH”.)

DETACH	PULLD_DIS	DP	DM	Condition
0	0	Pullup	Pulldown	Not recommended
0	1	Pullup	High impedance state	VBUS present
1	0	Pulldown	Pulldown	No VBUS
1	1	High impedance state	High impedance state	VBUS present & software disconnect

Bit 10 – REWAKEUP Send Remote Wakeup (cleared upon USB reset)
An Upstream Resume is sent only after the UDPHS bus has been in SUSPEND state for at least 5 ms.
This bit is automatically cleared by hardware at the end of the Upstream Resume.

Value	Description
0	Remote Wakeup is disabled (read), or this bit has no effect (write).
1	Remote Wakeup is enabled (read), or this bit forces an external interrupt on the UDPHS controller for Remote Wakeup purposes.

Bit 9 – DETACH Detach Command
See description of bit “PULL_DIS”.

Value	Description
0	UDPHS is attached (read), or this bit pulls up the DP line (attach command) (write).
1	UDPHS is detached, UTMI transceiver is suspended (read), or this bit simulates a detach on the UDPHS line and forces the UTMI transceiver into suspend state (Suspend M = 0) (write).

Downloaded from Arrow.com.

Bit 8 – EN_UDPHS UDPHS Enable

Value	Description
0	UDPHS is disabled (read), or this bit disables and resets the UDPHS controller (write). Switch the host to UTMI.
1	UDPHS is enabled (read), or this bit enables the UDPHS controller (write). Switch the host to UTMI.

Bit 7 – FADDR_EN Function Address Enable (cleared upon USB reset)

Value	Description
0	Device is not in address state (read), or only the default function address is used (write).
1	Device is in address state (read), or this bit is set by the device firmware after a successful status phase of a SET_ADDRESS transaction (write). When set, the only address accepted by the UDPHS controller is the one stored in the UDPHS Address field. It will not be cleared afterwards by the device firmware. It is cleared by hardware on hardware reset, or when UDPHS bus reset is received.

Bits 6:0 – DEV_ADDR[6:0] UDPHS Address (cleared upon USB reset)
This field contains the default address (0) after power-up or UDPHS bus reset (read), or it is written with the value set by a SET_ADDRESS request received by the device firmware (write).

Downloaded from Arrow.com.

40.7.2 UDPHS Frame Number Register

Name: UDPHS_FNUM
Offset: 0x04
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	FNUM_ERR							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			FRAME_NUMBER[10:5]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FRAME_NUMBER[4:0]					MICRO_FRAME_NUM[2:0]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 31 – FNUM_ERR Frame Number CRC Error (cleared upon USB reset)
This bit is set by hardware when a corrupted Frame Number in Start of Frame packet (or Micro SOF) is received. This bit and the INT_SOF (or MICRO_SOF) interrupt are updated at the same time.

Bits 13:3 – FRAME_NUMBER[10:0] Frame Number as defined in the Packet Field Formats (cleared upon USB reset)
This field is provided in the last received SOF packet (see INT_SOF in the UDPHS Interrupt Status Register).

Bits 2:0 – MICRO_FRAME_NUM[2:0] Microframe Number (cleared upon USB reset)
Number of the received microframe (0 to 7) in one frame.This field is reset at the beginning of each new frame (1 ms).
One microframe is received each 125 microseconds (1 ms/8).

Downloaded from Arrow.com.

40.7.3 UDPHS Interrupt Enable Register

Name: UDPHS_IEN
Offset: 0x10
Reset: 0x00000010
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	DMA_7	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	

Bit	23	22	21	20	19	18	17	16
	EPT_15	EPT_14	EPT_13	EPT_12	EPT_11	EPT_10	EPT_9	EPT_8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	EPT_7	EPT_6	EPT_5	EPT_4	EPT_3	EPT_2	EPT_1	EPT_0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UPSTR_RES	ENDOFRSM	WAKE_UP	ENDRESET	INT_SOF	MICRO_SOF	DET_SUSPD	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	1	0	0	0	

Bits 25, 26, 27, 28, 29, 30, 31 – DMA_x DMA Channel x Interrupt Enable (cleared upon USB reset)

Value	Description
0	Disable the interrupts for this channel.
1	Enable the interrupts for this channel.

Bits 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23 – EPT_x Endpoint x Interrupt Enable (cleared upon USB reset)

Value	Description
0	Disable the interrupts for this endpoint.
1	Enable the interrupts for this endpoint.

Bit 7 – UPSTR_RES Upstream Resume Interrupt Enable (cleared upon USB reset)

Value	Description
0	Disable Upstream Resume Interrupt.
1	Enable Upstream Resume Interrupt.

Bit 6 – ENDOFRSM End Of Resume Interrupt Enable (cleared upon USB reset)

Value	Description
0	Disable Resume Interrupt.
1	Enable Resume Interrupt.

Bit 5 – WAKE_UP Wake Up CPU Interrupt Enable (cleared upon USB reset)

Value	Description
0	Disable Wake-up CPU Interrupt.
1	Enable Wake-up CPU Interrupt.

Bit 4 – ENDRESET End Of Reset Interrupt Enable (cleared upon USB reset)

Downloaded from Arrow.com.

Value	Description
0	Disable End Of Reset Interrupt.
1	Enable End Of Reset Interrupt. Automatically enabled after USB reset.

Bit 3 – INT_SOF SOF Interrupt Enable (cleared upon USB reset)

Value	Description
0	Disable SOF Interrupt.
1	Enable SOF Interrupt.

Bit 2 – MICRO_SOF Micro-SOF Interrupt Enable (cleared upon USB reset)

Value	Description
0	Disable Micro-SOF Interrupt.
1	Enable Micro-SOF Interrupt.

Bit 1 – DET_SUSPD Suspend Interrupt Enable (cleared upon USB reset)

Value	Description
0	Disable Suspend Interrupt.
1	Enable Suspend Interrupt.

Downloaded from Arrow.com.

40.7.4 UDPHS Interrupt Status Register

Name: UDPHS_INTSTA
Offset: 0x14
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	DMA_7	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	
Access	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	

Bit	23	22	21	20	19	18	17	16
	EPT_15	EPT_14	EPT_13	EPT_12	EPT_11	EPT_10	EPT_9	EPT_8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	EPT_7	EPT_6	EPT_5	EPT_4	EPT_3	EPT_2	EPT_1	EPT_0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UPSTR_RES	ENDOFRSM	WAKE_UP	ENDRESET	INT_SOF	MICRO_SOF	DET_SUSPD	SPEED
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 25, 26, 27, 28, 29, 30, 31 – DMA_x DMA Channel x Interrupt

Value	Description
0	Reset when the UDPHS_DMASTATUSx interrupt source is cleared.
1	Set by hardware when an interrupt is triggered by the DMA Channelx and this endpoint interrupt is enabled by the DMA_x bit in UDPHS_IEN.

Bits 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23 – EPT_x Endpoint x Interrupt (cleared upon USB reset)

Value	Description
0	Reset when the UDPHS_EPTSTAx interrupt source is cleared.
1	Set by hardware when an interrupt is triggered by the UDPHS_EPTSTAx register and this endpoint interrupt is enabled by the EPT_x bit in UDPHS_IEN.

Bit 7 – UPSTR_RES Upstream Resume Interrupt

Value	Description
0	Cleared by setting the UPSTR_RES bit in UDPHS_CLRINT.
1	Set by hardware when the UDPHS controller is sending a resume signal called “upstream resume”. This triggers a UDPHS interrupt when the UPSTR_RES bit is set in UDPHS_IEN.

Bit 6 – ENDOFRSM End Of Resume Interrupt

Value	Description
0	Cleared by setting the ENDOFRSM bit in UDPHS_CLRINT.
1	Set by hardware when the UDPHS controller detects a good end of resume signal initiated by the host. This triggers a UDPHS interrupt when the ENDOFRSM bit is set in UDPHS_IEN.

Bit 5 – WAKE_UP Wake Up CPU Interrupt

Value	Description
0	Cleared by setting the WAKE_UP bit in UDPHS_CLRINT.

Downloaded from Arrow.com.

Value	Description
1	Set by hardware when the UDPHS controller is in SUSPEND state and is re-activated by a filtered non-idle signal from the UDPHS line (not by an upstream resume). This triggers a UDPHS interrupt when the WAKE_UP bit is set in UDPHS_IEN register. When receiving this interrupt, the user has to enable the device controller clock prior to operation. **Note:** this interrupt is generated even if the device controller clock is disabled.

Bit 4 – ENDRESET End Of Reset Interrupt

Value	Description
0	Cleared by setting the ENDRESET bit in UDPHS_CLRINT.
1	Set by hardware when an End Of Reset has been detected by the UDPHS controller. This triggers a UDPHS interrupt when the ENDRESET bit is set in UDPHS_IEN.

Bit 3 – INT_SOF Start Of Frame Interrupt
Note: The Micro Start Of Frame Interrupt (MICRO_SOF), and the Start Of Frame Interrupt (INT_SOF) are not generated at the same time.

Value	Description
0	Cleared by setting the INT_SOF bit in UDPHS_CLRINT.
1	Set by hardware when an UDPHS Start Of Frame PID (SOF) has been detected (every 1 ms) or synthesized by the macro. This triggers a UDPHS interrupt when the INT_SOF bit is set in UDPHS_IEN register. In case of detected SOF, in High Speed mode, the MICRO_FRAME_NUMBER field is cleared in UDPHS_FNUM register and the FRAME_NUMBER field is updated.

Bit 2 – MICRO_SOF Micro Start Of Frame Interrupt
Note: The Micro Start Of Frame Interrupt (MICRO_SOF), and the Start Of Frame Interrupt (INT_SOF) are not generated at the same time.

Value	Description
0	Cleared by setting the MICRO_SOF bit in UDPHS_CLRINT register.
1	Set by hardware when an UDPHS micro start of frame PID (SOF) has been detected (every 125 us) or synthesized by the macro. This triggers a UDPHS interrupt when the MICRO_SOF bit is set in UDPHS_IEN. In case of detected SOF, the MICRO_FRAME_NUM field in UDPHS_FNUM register is incremented and the FRAME_NUMBER field does not change.

Bit 1 – DET_SUSPD Suspend Interrupt

Value	Description
0	Cleared by setting the DET_SUSPD bit in UDPHS_CLRINT register.
1	Set by hardware when a UDPHS Suspend (Idle bus for three frame periods, a J state for 3 ms) is detected. This triggers a UDPHS interrupt when the DET_SUSPD bit is set in UDPHS_IEN register.

Bit 0 – SPEED Speed Status

Value	Description
0	Reset by hardware when the hardware is in Full Speed mode.
1	Set by hardware when the hardware is in High Speed mode.

Downloaded from Arrow.com.

40.7.5 UDPHS Clear Interrupt Register

Name: UDPHS_CLRINT
Offset: 0x18
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	UPSTR_RES	ENDOFRSM	WAKE_UP	ENDRESET	INT_SOF	MICRO_SOF	DET_SUSPD	
Access	W	W	W	W	W	W	W	
Reset	–	–	–	–	–	–	–	

Bit 7 – UPSTR_RES Upstream Resume Interrupt Clear

Value	Description
0	No effect.
1	Clear the UPSTR_RES bit in UDPHS_INTSTA.

Bit 6 – ENDOFRSM End Of Resume Interrupt Clear

Value	Description
0	No effect.
1	Clear the ENDOFRSM bit in UDPHS_INTSTA.

Bit 5 – WAKE_UP Wake Up CPU Interrupt Clear

Value	Description
0	No effect.
1	Clear the WAKE_UP bit in UDPHS_INTSTA.

Bit 4 – ENDRESET End Of Reset Interrupt Clear

Value	Description
0	No effect.
1	Clear the ENDRESET bit in UDPHS_INTSTA.

Bit 3 – INT_SOF Start Of Frame Interrupt Clear

Value	Description
0	No effect.
1	Clear the INT_SOF bit in UDPHS_INTSTA.

Bit 2 – MICRO_SOF Micro Start Of Frame Interrupt Clear

Value	Description
0	No effect.
1	Clear the MICRO_SOF bit in UDPHS_INTSTA.

Downloaded from Arrow.com.

Bit 1 – DET_SUSPD Suspend Interrupt Clear

Value	Description
0	No effect.
1	Clear the DET_SUSPD bit in UDPHS_INTSTA.

Downloaded from Arrow.com.

40.7.6 UDPHS Endpoints Reset Register

Name: UDPHS_EPTRST
Offset: 0x1C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EPT_15	EPT_14	EPT_13	EPT_12	EPT_11	EPT_10	EPT_9	EPT_8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	EPT_7	EPT_6	EPT_5	EPT_4	EPT_3	EPT_2	EPT_1	EPT_0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – EPT_x Endpoint x Reset
Setting this bit clears all bits in Endpoint Status register (UDPHS_EPTSTAx), except the TOGGLESQ_STA field.

Value	Description
0	No effect.
1	Reset the Endpointx state.

Downloaded from Arrow.com.

40.7.7 UDPHS Test Register

Name: UDPHS_TST
Offset: 0xE0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			OPMODE2	TST_PKT	TST_K	TST_J	SPEED_CFG[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – OPMODE2 OpMode2
Note: For the Test mode, Test_SE0_NAK (refer to Universal Serial Bus Specification, Revision 2.0: 7.1.20, Test Mode Support). Force the device in High Speed mode, and configure a bulk-type endpoint. Do not fill this endpoint for sending NAK to the host.
Upon command, a port's transceiver must enter the High Speed Receive mode and remain in that mode until the exit action is taken. This enables the testing of output impedance, low level output voltage and loading characteristics. In addition, while in this mode, upstream facing ports (and only upstream facing ports) must respond to any IN token packet with a NAK handshake (only if the packet CRC is determined to be correct) within the normal allowed device response time. This enables testing of the device squelch level circuitry and, additionally, provides a general purpose stimulus/response test for basic functional testing.

Value	Description
0	No effect.
1	Set to force the OpMode signal (UTMI interface) to "10", to disable the bit-stuffing and the NRZI encoding.

Bit 4 – TST_PKT Test Packet Mode

Value	Description
0	No effect.
1	Set to repetitively transmit the packet stored in the current bank. This enables the testing of rise and fall times, eye patterns, jitter, and any other dynamic waveform specifications.

Bit 3 – TST_K Test K Mode

Value	Description
0	No effect.
1	Set to send the K state on the UDPHS line. This enables the testing of the high output drive level on the D- line.

Bit 2 – TST_J Test J Mode

Downloaded from Arrow.com.

Value	Description
0	No effect.
1	Set to send the J state on the UDPHS line. This enables the testing of the high output drive level on the D+ line.

Bits 1:0 – SPEED_CFG[1:0] Speed Configuration

Value	Name	Description
0	NORMAL	Normal mode: The macro is in Full Speed mode, ready to make a High Speed identification, if the host supports it and then to automatically switch to High Speed mode.
1	–	Reserved
2	HIGH_SPEED	Force High Speed: Set this value to force the hardware to work in High Speed mode. Only for debug or test purpose.
3	FULL_SPEED	Force Full Speed: Set this value to force the hardware to work only in Full Speed mode. In this configuration, the macro will not respond to a High Speed reset handshake.

Downloaded from Arrow.com.

40.7.8 UDPHS Endpoint Configuration Register

Name: UDPHS_EPTCFGx
Offset: 0x0100 + x*0x20 [x=0..6]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	EPT_MAPD							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							NB_TRANS[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	BK_NUMBER[1:0]		EPT_TYPE[1:0]		EPT_DIR	EPT_SIZE[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – EPT_MAPD Endpoint Mapped (cleared upon USB reset)

Value	Description
0	The user should reprogram the register with correct values.
1	Set by hardware when the endpoint size (EPT_SIZE) and the number of banks (BK_NUMBER) are correct regarding: – The max endpoint size for this endpoint – The number of allowed banks for this endpoint

Bits 9:8 – NB_TRANS[1:0] Number Of Transactions per Microframe (cleared upon USB reset)
The number of transactions per microframe is set by software.
Note: Meaningful for high bandwidth isochronous endpoint only.

Bits 7:6 – BK_NUMBER[1:0] Number of Banks (cleared upon USB reset)
Set this field according to the endpoint's number of banks (see section Endpoint Configuration).

Value	Name	Description
0	0	Zero bank, the endpoint is not mapped in memory
1	1	One bank (bank 0)
2	2	Double bank (Ping-Pong: bank0/bank1)
3	3	Triple bank (bank0/bank1/bank2)

Bits 5:4 – EPT_TYPE[1:0] Endpoint Type (cleared upon USB reset)
Set this field according to the endpoint type (see section Endpoint Configuration).
(Endpoint 0 should always be configured as control).

Value	Name	Description
0	CTRL8	Control endpoint
1	ISO	Isochronous endpoint
2	BULK	Bulk endpoint
3	INT	Interrupt endpoint

Downloaded from Arrow.com.

Bit 3 – EPT_DIR Endpoint Direction (cleared upon USB reset)
For Control endpoints this bit has no effect and should be left at zero.

Value	Description
0	Clear this bit to configure OUT direction for Bulk, Interrupt and Isochronous endpoints.
1	Set this bit to configure IN direction for Bulk, Interrupt and Isochronous endpoints.

Bits 2:0 – EPT_SIZE[2:0] Endpoint Size (cleared upon USB reset)
Set this field according to the endpoint size in bytes (see the section Endpoint Configuration). Note that 1024 bytes is only for isochronous endpoints.

Value	Name	Description
0	8	8 bytes
1	16	16 bytes
2	32	32 bytes
3	64	64 bytes
4	128	128 bytes
5	256	256 bytes
6	512	512 bytes
7	1024	1024 bytes

Downloaded from Arrow.com.

40.7.9 UDPHS Endpoint Control Enable Register (Control, Bulk, Interrupt Endpoints)

Name: UDPHS_EPTCTLENBx
Offset: 0x0104 + x*0x20 [x=0..6]
Reset: –
Property: Write-only

This register view is relevant only if EPT_TYPE = 0x0, 0x2 or 0x3 in UDPHS Endpoint Configuration Register.

For additional information, see UDPHS Endpoint Control Register (Control, Bulk, Interrupt Endpoints).

Bit	31	30	29	28	27	26	25	24
	SHRT_PCKT							
Access	W							
Reset	–							

Bit	23	22	21	20	19	18	17	16
						BUSY_BANK		
Access						W		
Reset						–		

Bit	15	14	13	12	11	10	9	8
	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
Access				W	W		W	W
Reset				–	–		–	–

Bit 31 – SHRT_PCKT Short Packet Send/Short Packet Interrupt Enable
For IN endpoints: Guarantees short packet at end of DMA Transfer if the UDPHS_DMACONTROLx register END_B_EN and UDPHS_EPTCTLx register AUTOVALID bits are also set.
For OUT endpoints:

Value	Description
0	No effect.
1	Enable Short Packet Interrupt.

Bit 18 – BUSY_BANK Busy Bank Interrupt Enable

Value	Description
0	No effect.
1	Enable Busy Bank Interrupt.

Bit 15 – NAK_OUT NAKOUT Interrupt Enable

Value	Description
0	No effect.
1	Enable NAKOUT Interrupt.

Bit 14 – NAK_IN NAKIN Interrupt Enable

Value	Description
0	No effect.
1	Enable NAKIN Interrupt.

Bit 13 – STALL_SNT Stall Sent Interrupt Enable

Value	Description
0	No effect.

Downloaded from Arrow.com.

Value	Description
1	Enable Stall Sent Interrupt.

Bit 12 – RX_SETUP Received SETUP

Value	Description
0	No effect.
1	Enable RX_SETUP Interrupt.

Bit 11 – TXRDY TX Packet Ready Interrupt Enable

Value	Description
0	No effect.
1	Enable TX Packet Ready/Transaction Error Interrupt.

Bit 10 – TX_COMPLT Transmitted IN Data Complete Interrupt Enable

Value	Description
0	No effect.
1	Enable Transmitted IN Data Complete Interrupt.

Bit 9 – RXRDY_TXKL Received OUT Data Interrupt Enable

Value	Description
0	No effect.
1	Enable Received OUT Data Interrupt.

Bit 8 – ERR_OVFLW Overflow Error Interrupt Enable

Value	Description
0	No effect.
1	Enable Overflow Error Interrupt.

Bit 4 – NYET_DIS NYET Disable (Only for High Speed Bulk OUT endpoints)

Value	Description
0	No effect.
1	Forces an ACK response to the next High Speed Bulk OUT transfer instead of a NYET response.

Bit 3 – INTDIS_DMA Interrupts Disable DMA

Value	Description
0	No effect.
1	If set, when an enabled endpoint-originated interrupt is triggered, the DMA request is disabled.

Bit 1 – AUTO_VALID Packet Auto-Valid Enable

Value	Description
0	No effect.
1	Enable this bit to automatically validate the current packet and switch to the next bank for both IN and OUT transfers.

Bit 0 – EPT_ENABL Endpoint Enable

Value	Description
0	No effect.
1	Enable endpoint according to the device configuration.

Downloaded from Arrow.com.

40.7.10 UDPHS Endpoint Control Enable Register (Isochronous Endpoints)

Name: UDPHS_EPTCTLENBx
Offset: 0x0104 + x*0x20 [x=0..6]
Reset: –
Property: Write-only

This register view is relevant only if EPT_TYPE = 0x1 in UDPHS Endpoint Configuration Register.

For additional information, see UDPHS Endpoint Control Register (Isochronous Endpoint).

Bit	31	30	29	28	27	26	25	24
	SHRT_PCKT							
Access	W							
Reset	–							

Bit	23	22	21	20	19	18	17	16
						BUSY_BANK		
Access						W		
Reset						–		

Bit	15	14	13	12	11	10	9	8
		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
Access	W	W			W		W	W
Reset	–	–			–		–	–

Bit 31 – SHRT_PCKT Short Packet Send/Short Packet Interrupt Enable
For IN endpoints: Guarantees short packet at end of DMA Transfer if the UDPHS_DMACONTROLx register END_B_EN and UDPHS_EPTCTLx register AUTOVALID bits are also set.
For OUT endpoints:

Value	Description
0	No effect.
1	Enable Short Packet Interrupt.

Bit 18 – BUSY_BANK Busy Bank Interrupt Enable

Value	Description
0	No effect.
1	Enable Busy Bank Interrupt.

Bit 14 – ERR_FLUSH Bank Flush Error Interrupt Enable

Value	Description
0	No effect.
1	Enable Bank Flush Error Interrupt.

Bit 13 – ERR_CRC_NTR ISO CRC Error/Number of Transaction Error Interrupt Enable

Value	Description
0	No effect.
1	Enable Error CRC ISO/Error Number of Transaction Interrupt.

Bit 12 – ERR_FL_ISO Error Flow Interrupt Enable

Downloaded from Arrow.com.

Value	Description
0	No effect.
1	Enable Error Flow ISO Interrupt.

Bit 11 – TXRDY_TRER TX Packet Ready/Transaction Error Interrupt Enable

Value	Description
0	No effect.
1	Enable TX Packet Ready/Transaction Error Interrupt.

Bit 10 – TX_COMPLT Transmitted IN Data Complete Interrupt Enable

Value	Description
0	No effect.
1	Enable Transmitted IN Data Complete Interrupt.

Bit 9 – RXRDY_TXKL Received OUT Data Interrupt Enable

Value	Description
0	No effect.
1	Enable Received OUT Data Interrupt.

Bit 8 – ERR_OVFLW Overflow Error Interrupt Enable

Value	Description
0	No effect.
1	Enable Overflow Error Interrupt.

Bit 7 – MDATA_RX MDATA Interrupt Enable (Only for high bandwidth Isochronous OUT endpoints)

Value	Description
0	No effect.
1	Enable MDATA Interrupt.

Bit 6 – DATAX_RX DATAx Interrupt Enable (Only for high bandwidth Isochronous OUT endpoints)

Value	Description
0	No effect.
1	Enable DATAx Interrupt.

Bit 3 – INTDIS_DMA Interrupts Disable DMA

Value	Description
0	No effect.
1	If set, when an enabled endpoint-originated interrupt is triggered, the DMA request is disabled.

Bit 1 – AUTO_VALID Packet Auto-Valid Enable

Value	Description
0	No effect.
1	Enable this bit to automatically validate the current packet and switch to the next bank for both IN and OUT transfers.

Bit 0 – EPT_ENABL Endpoint Enable

Value	Description
0	No effect.
1	Enable endpoint according to the device configuration.

Downloaded from Arrow.com.

40.7.11 UDPHS Endpoint Control Disable Register (Control, Bulk, Interrupt Endpoints)

Name: UDPHS_EPTCTLDISx
Offset: 0x0108 + x*0x20 [x=0..6]
Reset: –
Property: Write-only

This register view is relevant only if EPT_TYPE = 0x0, 0x2 or 0x3 in UDPHS Endpoint Configuration Register.

For additional information, see UDPHS Endpoint Control Register (Control, Bulk, Interrupt Endpoints).

Bit	31	30	29	28	27	26	25	24
	SHRT_PCKT							
Access	W							
Reset	–							

Bit	23	22	21	20	19	18	17	16
						BUSY_BANK		
Access						W		
Reset						–		

Bit	15	14	13	12	11	10	9	8
	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_DISABL
Access				W	W		W	W
Reset				–	–		–	–

Bit 31 – SHRT_PCKT Short Packet Interrupt Disable
For IN endpoints: Never automatically add a zero length packet at end of DMA transfer.
For OUT endpoints:

Value	Description
0	No effect.
1	Disable Short Packet Interrupt.

Bit 18 – BUSY_BANK Busy Bank Interrupt Disable

Value	Description
0	No effect.
1	Disable Busy Bank Interrupt.

Bit 15 – NAK_OUT NAKOUT Interrupt Disable

Value	Description
0	No effect.
1	Disable NAKOUT Interrupt.

Bit 14 – NAK_IN NAKIN Interrupt Disable

Value	Description
0	No effect.
1	Disable NAKIN Interrupt.

Bit 13 – STALL_SNT Stall Sent Interrupt Disable

Value	Description
0	No effect.
1	Disable Stall Sent Interrupt.

Downloaded from Arrow.com.

Bit 12 – RX_SETUP Received SETUP Interrupt Disable

Value	Description
0	No effect.
1	Disable RX_SETUP Interrupt.

Bit 11 – TXRDY TX Packet Ready Interrupt Disable

Value	Description
0	No effect.
1	Disable TX Packet Ready/Transaction Error Interrupt.

Bit 10 – TX_COMPLT Transmitted IN Data Complete Interrupt Disable

Value	Description
0	No effect.
1	Disable Transmitted IN Data Complete Interrupt.

Bit 9 – RXRDY_TXKL Received OUT Data Interrupt Disable

Value	Description
0	No effect.
1	Disable Received OUT Data Interrupt.

Bit 8 – ERR_OVFLW Overflow Error Interrupt Disable

Value	Description
0	No effect.
1	Disable Overflow Error Interrupt.

Bit 4 – NYET_DIS NYET Enable (Only for High Speed Bulk OUT endpoints)

Value	Description
0	No effect.
1	Let the hardware handle the handshake response for the High Speed Bulk OUT transfer.

Bit 3 – INTDIS_DMA Interrupts Disable DMA

Value	Description
0	No effect.
1	Disable the “Interrupts Disable DMA”.

Bit 1 – AUTO_VALID Packet Auto-Valid Disable

Value	Description
0	No effect.
1	Disable this bit to not automatically validate the current packet.

Bit 0 – EPT_DISABL Endpoint Disable

Value	Description
0	No effect.
1	Disable endpoint.

Downloaded from Arrow.com.

40.7.12 UDPHS Endpoint Control Disable Register (Isochronous Endpoint)

Name: UDPHS_EPTCTLDISx
Offset: 0x0108 + x*0x20 [x=0..6]
Reset: –
Property: Write-only

This register view is relevant only if EPT_TYPE = 0x1 in UDPHS Endpoint Configuration Register.

For additional information, see “UDPHS Endpoint Control Register (Isochronous Endpoint)”.

Bit	31	30	29	28	27	26	25	24
	SHRT_PCKT							
Access	W							
Reset	–							

Bit	23	22	21	20	19	18	17	16
						BUSY_BANK		
Access						W		
Reset						–		

Bit	15	14	13	12	11	10	9	8
		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_DISABL
Access	W	W			W		W	W
Reset	–	–			–		–	–

Bit 31 – SHRT_PCKT Short Packet Interrupt Disable
For IN endpoints: Never automatically add a zero length packet at end of DMA transfer.
For OUT endpoints:

Value	Description
0	No effect.
1	Disable Short Packet Interrupt.

Bit 18 – BUSY_BANK Busy Bank Interrupt Disable

Value	Description
0	No effect.
1	Disable Busy Bank Interrupt.

Bit 14 – ERR_FLUSH bank flush error Interrupt Disable

Value	Description
0	No effect.
1	Disable Bank Flush Error Interrupt.

Bit 13 – ERR_CRC_NTR ISO CRC Error/Number of Transaction Error Interrupt Disable

Value	Description
0	No effect.
1	Disable Error CRC ISO/Error Number of Transaction Interrupt.

Bit 12 – ERR_FL_ISO Error Flow Interrupt Disable

Value	Description
0	No effect.

Downloaded from Arrow.com.

Value	Description
1	Disable Error Flow ISO Interrupt.

Bit 11 – TXRDY_TRER TX Packet Ready/Transaction Error Interrupt Disable

Value	Description
0	No effect.
1	Disable TX Packet Ready/Transaction Error Interrupt.

Bit 10 – TX_COMPLT Transmitted IN Data Complete Interrupt Disable

Value	Description
0	No effect.
1	Disable Transmitted IN Data Complete Interrupt.

Bit 9 – RXRDY_TXKL Received OUT Data Interrupt Disable

Value	Description
0	No effect.
1	Disable Received OUT Data Interrupt.

Bit 8 – ERR_OVFLW Overflow Error Interrupt Disable

Value	Description
0	No effect.
1	Disable Overflow Error Interrupt.

Bit 7 – MDATA_RX MDATA Interrupt Disable (Only for High Bandwidth Isochronous OUT endpoints)

Value	Description
0	No effect.
1	Disable MDATA Interrupt.

Bit 6 – DATAX_RX DATAx Interrupt Disable (Only for High Bandwidth Isochronous OUT endpoints)

Value	Description
0	No effect.
1	Disable DATAx Interrupt.

Bit 3 – INTDIS_DMA Interrupts Disable DMA

Value	Description
0	No effect.
1	Disable the “Interrupts Disable DMA”.

Bit 1 – AUTO_VALID Packet Auto-Valid Disable

Value	Description
0	No effect.
1	Disable this bit to not automatically validate the current packet.

Bit 0 – EPT_DISABL Endpoint Disable

Value	Description
0	No effect.
1	Disable endpoint.

Downloaded from Arrow.com.

40.7.13 UDPHS Endpoint Control Register (Control, Bulk, Interrupt Endpoints)

Name: UDPHS_EPTCTLx
Offset: 0x010C + x*0x20 [x=0..6]
Reset: 0x00000000
Property: Read-only

This register view is relevant only if EPT_TYPE = 0x0, 0x2 or 0x3 in UDPHS Endpoint Configuration Register.

The reset value for UDPHS_EPTCTL0 is 0x00000001.

Bit	31	30	29	28	27	26	25	24
	SHRT_PCKT							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
						BUSY_BANK		
Access						R		
Reset						0		

Bit	15	14	13	12	11	10	9	8
	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				NYET_DIS	INTDIS_DMA		AUTO_VALID	EPT_ENABL
Access				R	R		R	R
Reset				0	0		0	0

Bit 31 – SHRT_PCKT Short Packet Interrupt Enabled (cleared upon USB reset)
For OUT endpoints: sends an Interrupt when a Short Packet has been received.
For IN endpoints: a Short Packet transmission is guaranteed upon end of the DMA Transfer, thus signaling a BULK or INTERRUPT end of transfer, but only if the UDPHS_DMACONTROLx register END_B_EN and UDPHS_EPTCTLx register AUTO_VALID bits are also set.

Value	Description
0	Short Packet Interrupt is masked.
1	Short Packet Interrupt is enabled.

Bit 18 – BUSY_BANK Busy Bank Interrupt Enabled (cleared upon USB reset)
For OUT endpoints: an interrupt is sent when all banks are busy.
For IN endpoints: an interrupt is sent when all banks are free.

Value	Description
0	BUSY_BANK Interrupt is masked.
1	BUSY_BANK Interrupt is enabled.

Bit 15 – NAK_OUT NAKOUT Interrupt Enabled (cleared upon USB reset)

Value	Description
0	NAKOUT Interrupt is masked.
1	NAKOUT Interrupt is enabled.

Bit 14 – NAK_IN NAKIN Interrupt Enabled (cleared upon USB reset)

Value	Description
0	NAKIN Interrupt is masked.
1	NAKIN Interrupt is enabled.

Downloaded from Arrow.com.

Bit 13 – STALL_SNT Stall Sent Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Stall Sent Interrupt is masked.
1	Stall Sent Interrupt is enabled.

Bit 12 – RX_SETUP Received SETUP Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Received SETUP is masked.
1	Received SETUP is enabled.

Bit 11 – TXRDY TX Packet Ready Interrupt Enabled (cleared upon USB reset)

⚠ CAUTION Interrupt source is active as long as the corresponding UDPHS_EPTSTAx register TXRDY flag remains low. If there are no more banks available for transmitting after the software has set UDPHS_EPTSTAx/TXRDY for the last transmit packet, then the interrupt source remains inactive until the first bank becomes free again to transmit at UDPHS_EPTSTAx/TXRDY hardware clear.

Value	Description
0	TX Packet Ready Interrupt is masked.
1	TX Packet Ready Interrupt is enabled.

Bit 10 – TX_COMPLT Transmitted IN Data Complete Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Transmitted IN Data Complete Interrupt is masked.
1	Transmitted IN Data Complete Interrupt is enabled.

Bit 9 – RXRDY_TXKL Received OUT Data Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Received OUT Data Interrupt is masked.
1	Received OUT Data Interrupt is enabled.

Bit 8 – ERR_OVFLW Overflow Error Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Overflow Error Interrupt is masked.
1	Overflow Error Interrupt is enabled.

Bit 4 – NYET_DIS NYET Disable (Only for High Speed Bulk OUT Endpoints) (cleared upon USB reset)
Note: According to the Universal Serial Bus Specification, Rev 2.0 (8.5.1.1 NAK Responses to OUT/DATA During PING Protocol), a NAK response to an HS Bulk OUT transfer is expected to be an unusual occurrence.

Value	Description
0	Lets the hardware handle the handshake response for the High Speed Bulk OUT transfer.
1	Forces an ACK response to the next High Speed Bulk OUT transfer instead of a NYET response.

Bit 3 – INTDIS_DMA Interrupt Disables DMA (cleared upon USB reset)
If set, when an enabled endpoint-originated interrupt is triggered, the DMA request is disabled regardless of the UDPHS_IEN register EPT_x bit for this endpoint. Then, the firmware will have to clear or disable the interrupt source or clear this bit if transfer completion is needed.
If the exception raised is associated with the new system bank packet, then the previous DMA packet transfer is normally completed, but the new DMA packet transfer is not started (not requested).
If the exception raised is not associated to a new system bank packet (NAK_IN, NAK_OUT, etc.), then the request cancellation may happen at any time and may immediately stop the current DMA transfer.
This may be used, for example, to identify or prevent an erroneous packet to be transferred into a buffer or to complete a DMA buffer by software after reception of a short packet.

Downloaded from Arrow.com.

Bit 1 – AUTO_VALID Packet Auto-Valid Enabled (Not for CONTROL Endpoints) (cleared upon USB reset)
Set this bit to automatically validate the current packet and switch to the next bank for both IN and OUT endpoints.
For IN Transfer:
If this bit is set, the UDPHS_EPTSTAx register TXRDY bit is set automatically when the current bank is full and at the end of DMA buffer if the UDPHS_DMACONTROLx register END_B_EN bit is set.
The user may still set the UDPHS_EPTSTAx register TXRDY bit if the current bank is not full, unless the user needs to send a Zero Length Packet by software.
For OUT Transfer:
If this bit is set, the UDPHS_EPTSTAx register RXRDY_TXKL bit is automatically reset for the current bank when the last packet byte has been read from the bank FIFO or at the end of DMA buffer if the UDPHS_DMACONTROLx register END_B_EN bit is set. For example, to truncate a padded data packet when the actual data transfer size is reached.
The user may still clear the UDPHS_EPTSTAx register RXRDY_TXKL bit, for example, after completing a DMA buffer by software if UDPHS_DMACONTROLx register END_B_EN bit was disabled or in order to cancel the read of the remaining data bank(s).

Bit 0 – EPT_ENABL Endpoint Enable (cleared upon USB reset)

Value	Description
0	The endpoint is disabled according to the device configuration. Endpoint 0 should always be enabled after a hardware or UDPHS bus reset and participate in the device configuration.
1	The endpoint is enabled according to the device configuration.

Downloaded from Arrow.com.

40.7.14 UDPHS Endpoint Control Register (Isochronous Endpoint)

Name: UDPHS_EPTCTLx
Offset: 0x010C + x*0x20 [x=0..6]
Reset: 0x00000000
Property: Read-only

This register view is relevant only if EPT_TYPE = 0x1 in UDPHS Endpoint Configuration Register.

The reset value for UDPHS_EPTCTL0 is 0x00000001.

Bit	31	30	29	28	27	26	25	24
	SHRT_PCKT							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
						BUSY_BANK		
Access						R		
Reset						0		

Bit	15	14	13	12	11	10	9	8
		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MDATA_RX	DATAX_RX			INTDIS_DMA		AUTO_VALID	EPT_ENABL
Access	R	R			R		R	R
Reset	0	0			0		0	0

Bit 31 – SHRT_PCKT Short Packet Interrupt Enabled (cleared upon USB reset)
For OUT endpoints: Send an Interrupt when a Short Packet has been received.
For IN endpoints: A Short Packet transmission is guaranteed upon end of the DMA Transfer, thus signaling an end of isochronous (micro-)frame data, but only if the UDPHS_DMACONTROLx register END_B_EN and UDPHS_EPTCTLx register AUTO_VALID bits are also set.

Value	Description
0	Short Packet Interrupt is masked.
1	Short Packet Interrupt is enabled.

Bit 18 – BUSY_BANK Busy Bank Interrupt Enabled (cleared upon USB reset)
For OUT endpoints: An interrupt is sent when all banks are busy.
For IN endpoints: An interrupt is sent when all banks are free.

Value	Description
0	BUSY_BANK Interrupt is masked.
1	BUSY_BANK Interrupt is enabled.

Bit 14 – ERR_FLUSH Bank Flush Error Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Bank Flush Error Interrupt is masked.
1	Bank Flush Error Interrupt is enabled.

Bit 13 – ERR_CRC_NTR ISO CRC Error/Number of Transaction Error Interrupt Enabled (cleared upon USB reset)

Value	Description
0	ISO CRC error/number of Transaction Error Interrupt is masked.
1	ISO CRC error/number of Transaction Error Interrupt is enabled.

Downloaded from Arrow.com.

Bit 12 – ERR_FL_ISO Error Flow Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Error Flow Interrupt is masked.
1	Error Flow Interrupt is enabled.

Bit 11 – TXRDY_TRER TX Packet Ready/Transaction Error Interrupt Enabled (cleared upon USB reset)

⚠ CAUTION Interrupt source is active as long as the corresponding UDPHS_EPTSTAx register TXRDY_TRER flag remains low. If there are no more banks available for transmitting after the software has set UDPHS_EPTSTAx/TXRDY_TRER for the last transmit packet, then the interrupt source remains inactive until the first bank becomes free again to transmit at UDPHS_EPTSTAx/TXRDY_TRER hardware clear.

Value	Description
0	TX Packet Ready/Transaction Error Interrupt is masked.
1	TX Packet Ready/Transaction Error Interrupt is enabled.

Bit 10 – TX_COMPLT Transmitted IN Data Complete Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Transmitted IN Data Complete Interrupt is masked.
1	Transmitted IN Data Complete Interrupt is enabled.

Bit 9 – RXRDY_TXKL Received OUT Data Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Received OUT Data Interrupt is masked.
1	Received OUT Data Interrupt is enabled.

Bit 8 – ERR_OVFLW Overflow Error Interrupt Enabled (cleared upon USB reset)

Value	Description
0	Overflow Error Interrupt is masked.
1	Overflow Error Interrupt is enabled.

Bit 7 – MDATA_RX MDATA Interrupt Enabled (Only for High Bandwidth Isochronous OUT endpoints) (cleared upon USB reset)

Value	Description
0	No effect.
1	Send an interrupt when an MDATA packet has been received and so at least one packet of the microframe data payload has been received.

Bit 6 – DATAX_RX DATAx Interrupt Enabled (Only for High Bandwidth Isochronous OUT endpoints) (cleared upon USB reset)

Value	Description
0	No effect.
1	Send an interrupt when a DATA2, DATA1 or DATA0 packet has been received meaning the whole microframe data payload has been received.

Bit 3 – INTDIS_DMA Interrupt Disables DMA (cleared upon USB reset)
If set, when an enabled endpoint-originated interrupt is triggered, the DMA request is disabled regardless of the UDPHS_IEN register EPT_x bit for this endpoint. Then, the firmware will have to clear or disable the interrupt source or clear this bit if transfer completion is needed.
If the exception raised is associated with the new system bank packet, then the previous DMA packet transfer is normally completed, but the new DMA packet transfer is not started (not requested).
If the exception raised is not associated to a new system bank packet (ex: ERR_FL_ISO), then the request cancellation may happen at any time and may immediately stop the current DMA transfer.
This may be used, for example, to identify or prevent an erroneous packet to be transferred into a buffer or to complete a DMA buffer by software after reception of a short packet, or to perform buffer truncation on ERR_FL_ISO interrupt for adaptive rate.

Downloaded from Arrow.com.

Bit 1 – AUTO_VALID Packet Auto-Valid Enabled (cleared upon USB reset)
Set this bit to automatically validate the current packet and switch to the next bank for both IN and OUT endpoints.
For IN Transfer:
If this bit is set, the UDPHS_EPTSTAx register TXRDY_TRER bit is set automatically when the current bank is full and at the end of DMA buffer if the UDPHS_DMACONTROLx register END_B_EN bit is set.
The user may still set the UDPHS_EPTSTAx register TXRDY_TRER bit if the current bank is not full, unless the user needs to send a Zero Length Packet by software.
For OUT Transfer:
If this bit is set, the UDPHS_EPTSTAx register RXRDY_TXKL bit is automatically reset for the current bank when the last packet byte has been read from the bank FIFO or at the end of DMA buffer if the UDPHS_DMACONTROLx register END_B_EN bit is set. For example, to truncate a padded data packet when the actual data transfer size is reached.
The user may still clear the UDPHS_EPTSTAx register RXRDY_TXKL bit, for example, after completing a DMA buffer by software if UDPHS_DMACONTROLx register END_B_EN bit was disabled or in order to cancel the read of the remaining data bank(s).

Bit 0 – EPT_ENABL Endpoint Enable (cleared upon USB reset)

Value	Description
0	The endpoint is disabled according to the device configuration. Endpoint 0 should always be enabled after a hardware or UDPHS bus reset and participate in the device configuration.
1	The endpoint is enabled according to the device configuration.

Downloaded from Arrow.com.

40.7.15 UDPHS Endpoint Set Status Register (Control, Bulk, Interrupt Endpoints)

Name: UDPHS_EPTSETSTAx
Offset: 0x0114 + x*0x20 [x=0..6]
Reset: –
Property: Write-only

This register view is relevant only if EPT_TYPE = 0x0, 0x2 or 0x3 in UDPHS Endpoint Configuration Register.

For additional information, see UDPHS Endpoint Status Register (Control, Bulk, Interrupt Endpoints).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					TXRDY		RXRDY_TXKL	
Access					W		W	
Reset					–		–	

Bit	7	6	5	4	3	2	1	0
			FRCESTALL					
Access			W					
Reset			–					

Bit 11 – TXRDY TX Packet Ready Set

Value	Description
0	No effect.
1	Set this bit after a packet has been written into the endpoint FIFO for IN data transfers – This flag is used to generate a Data IN transaction (device to host). – Device firmware checks that it can write a data payload in the FIFO, checking that TXRDY is cleared. – Transfer to the FIFO is done by writing in the “Buffer Address” register. – Once the data payload has been transferred to the FIFO, the firmware notifies the UDPHS device setting TXRDY to one. – UDPHS bus transactions can start. – TXCOMP is set once the data payload has been received by the host. – Data should be written into the endpoint FIFO only after this bit has been cleared. – Set this bit without writing data to the endpoint FIFO to send a Zero Length Packet.

Bit 9 – RXRDY_TXKL KILL Bank Set (for IN Endpoint)

Value	Description
0	No effect.
1	Kill the last written bank.

Bit 5 – FRCESTALL Stall Handshake Request Set
Refer to chapters 8.4.5 (Handshake Packets) and 9.4.5 (Get Status) of the Universal Serial Bus Specification, Rev 2.0 for more information on the STALL handshake.

Downloaded from Arrow.com.

Value	Description
0	No effect.
1	Set this bit to request a STALL answer to the host for the next handshake

Downloaded from Arrow.com.

40.7.16 UDPHS Endpoint Set Status Register (Isochronous Endpoint)

Name: UDPHS_EPTSETSTAx
Offset: 0x0114 + x*0x20 [x=0..6]
Reset: –
Property: Write-only

This register view is relevant only if EPT_TYPE = 0x1 in UDPHS Endpoint Configuration Register.

For additional information, see UDPHS Endpoint Status Register (Isochronous Endpoint).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access					W		W	
Reset					–		–	

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 11 – TXRDY_TRER TX Packet Ready Set

Value	Description
0	No effect.
1	Set this bit after a packet has been written into the endpoint FIFO for IN data transfers – This flag is used to generate a Data IN transaction (device to host). – Device firmware checks that it can write a data payload in the FIFO, checking that TXRDY_TRER is cleared. – Transfer to the FIFO is done by writing in the “Buffer Address” register. – Once the data payload has been transferred to the FIFO, the firmware notifies the UDPHS device setting TXRDY_TRER to one. – UDPHS bus transactions can start. – TXCOMP is set once the data payload has been sent. – Data should be written into the endpoint FIFO only after this bit has been cleared. – Set this bit without writing data to the endpoint FIFO to send a Zero Length Packet.

Bit 9 – RXRDY_TXKL KILL Bank Set (for IN Endpoint)

Value	Description
0	No effect.
1	Kill the last written bank.

Downloaded from Arrow.com.

40.7.17 UDPHS Endpoint Clear Status Register (Control, Bulk, Interrupt Endpoints)

Name: UDPHS_EPTCLRSTAx
Offset: 0x0118 + x*0x20 [x=0..6]
Reset: –
Property: Write-only

This register view is relevant only if EPT_TYPE = 0x0, 0x2 or 0x3 in UDPHS Endpoint Configuration Register.

For additional information, see UDPHS Endpoint Status Register (Control, Bulk, Interrupt Endpoints).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP		TX_COMPLT	RXRDY_TXKL	
Access	W	W	W	W		W	W	
Reset	–	–	–	–		–	–	

Bit	7	6	5	4	3	2	1	0
		TOGGLESQ	FRCESTALL					
Access		W	W					
Reset		–	–					

Bit 15 – NAK_OUT NAKOUT Clear

Value	Description
0	No effect.
1	Clear the NAK_OUT flag of UDPHS_EPTSTAx.

Bit 14 – NAK_IN NAKIN Clear

Value	Description
0	No effect.
1	Clear the NAK_IN flags of UDPHS_EPTSTAx.

Bit 13 – STALL_SNT Stall Sent Clear

Value	Description
0	No effect.
1	Clear the STALL_SNT flags of UDPHS_EPTSTAx.

Bit 12 – RX_SETUP Received SETUP Clear

Value	Description
0	No effect.
1	Clear the RX_SETUP flags of UDPHS_EPTSTAx.

Bit 10 – TX_COMPLT Transmitted IN Data Complete Clear

Value	Description
0	No effect.
1	Clear the TX_COMPLT flag of UDPHS_EPTSTAx.

Bit 9 – RXRDY_TXKL Received OUT Data Clear

Downloaded from Arrow.com.

Value	Description
0	No effect.
1	Clear the RXRDY_TXKL flag of UDPHS_EPTSTAx.

Bit 6 – TOGGLESQ Data Toggle Clear
For OUT endpoints, the next received packet should be a DATA0.
For IN endpoints, the next packet will be sent with a DATA0 PID.

Value	Description
0	No effect.
1	Clear the PID data of the current bank

Bit 5 – FRCESTALL Stall Handshake Request Clear

Value	Description
0	No effect.
1	Clear the STALL request. The next packets from host will not be STALLed.

Downloaded from Arrow.com.

40.7.18 UDPHS Endpoint Clear Status Register (Isochronous Endpoint)

Name: UDPHS_EPTCLRSTAx
Offset: 0x0118 + x*0x20 [x=0..6]
Reset: –
Property: Write-only

This register view is relevant only if EPT_TYPE = 0x1 in UDPHS Endpoint Configuration Register.

For additional information, see UDPHS Endpoint Status Register (Isochronous Endpoint).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		ERR_FLUSH	ERR_CRC_NTR	ERR_FL_ISO		TX_COMPLT	RXRDY_TXKL	
Access		W	W	W		W	W	
Reset		–	–	–		–	–	

Bit	7	6	5	4	3	2	1	0
		TOGGLESQ						
Access		W						
Reset		–						

Bit 14 – ERR_FLUSH Bank Flush Error Clear

Value	Description
0	No effect.
1	Clear the ERR_FLUSH flags of UDPHS_EPTSTAx.

Bit 13 – ERR_CRC_NTR Number of Transaction Error Clear

Value	Description
0	No effect.
1	Clear the ERR_CRC_NTR flags of UDPHS_EPTSTAx.

Bit 12 – ERR_FL_ISO Error Flow Clear

Value	Description
0	No effect.
1	Clear the ERR_FL_ISO flags of UDPHS_EPTSTAx.

Bit 10 – TX_COMPLT Transmitted IN Data Complete Clear

Value	Description
0	No effect.
1	Clear the TX_COMPLT flag of UDPHS_EPTSTAx.

Bit 9 – RXRDY_TXKL Received OUT Data Clear

Value	Description
0	No effect.
1	Clear the RXRDY_TXKL flag of UDPHS_EPTSTAx.

Downloaded from Arrow.com.

Bit 6 – TOGGLESQ Data Toggle Clear
For OUT endpoints, the next received packet should be a DATA0.
For IN endpoints, the next packet will be sent with a DATA0 PID.

Value	Description
0	No effect.
1	Clear the PID data of the current bank

Downloaded from Arrow.com.

40.7.19 UDPHS Endpoint Status Register (Control, Bulk, Interrupt Endpoints)

Name: UDPHS_EPTSTAx
Offset: 0x011C + x*0x20 [x=0..6]
Reset: 0x00000040
Property: Read-only

This register view is relevant only if EPT_TYPE = 0x0, 0x2 or 0x3 in UDPHS Endpoint Configuration Register.

Bit	31	30	29	28	27	26	25	24
	SHRT_PCKT	BYTE_COUNT[10:4]						
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK_CTLDIR[1:0]	
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NAK_OUT	NAK_IN	STALL_SNT	RX_SETUP	TXRDY	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TOGGLESQ_STA[1:0]		FRCESTALL					
Access	R	R	R					
Reset	0	1	0					

Bit 31 – SHRT_PCKT Short Packet (cleared upon USB reset)
An OUT Short Packet is detected when the receive byte count is less than the configured UDPHS_EPTCFGx register EPT_Size.
This bit is updated at the same time as the BYTE_COUNT field.
It is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

Bits 30:20 – BYTE_COUNT[10:0] UDPHS Byte Count (cleared upon USB reset)
Byte count of a received data packet.
This field is incremented after each write into the endpoint (to prepare an IN transfer). It is decremented after each reading into the endpoint (OUT transfer).
This field is also updated at RXRDY_TXKL flag clear with the next bank, and at TXRDY flag set with the next bank.
This field is reset by UDPHS_EPTRST.EPT_x.

Bits 19:18 – BUSY_BANK_STA[1:0] Busy Bank Number (cleared upon USB reset)
These bits are set by hardware to indicate the number of busy banks.
IN endpoint: Indicates the number of busy banks filled by the user, ready for IN transfer.
OUT endpoint: Indicates the number of busy banks filled by OUT transaction from the Host.

Value	Name	Description
0	0BUSYBANK	All banks are free
1	1BUSYBANK	1 busy bank
2	2BUSYBANKS	2 busy banks
3	3BUSYBANKS	3 busy banks

Bits 17:16 – CURBK_CTLDIR[1:0] Current Bank/Control Direction (cleared upon USB reset)
Control Direction (for Control endpoint only):
0: A Control Write is requested by the Host.
1: A Control Read is requested by the Host.

Downloaded from Arrow.com.

Note:

1. Corresponds to the the 7th bit of the bmRequestType (Byte 0 of the Setup Data).
2. Updated after receiving new setup data.

Current Bank (not relevant for Control endpoint):

- Set by hardware to indicate the number of the current bank.
- Reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).
- The current bank is updated each time the user:
  - Sets the TX Packet Ready bit to prepare the next IN transfer and to switch to the next bank.
  - Clears the received OUT data bit to access the next bank.

Value	Name	Description
0	BANK0	Bank 0 (or single bank)
1	BANK1	Bank 1
2	BANK2	Bank 2

Bit 15 – NAK_OUT NAK OUT (cleared upon USB reset)
This bit is set by hardware when a NAK handshake has been sent in response to an OUT or PING request from the Host.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by EPT_CTL_DISx (disable endpoint).

Bit 14 – NAK_IN NAK IN (cleared upon USB reset)
This bit is set by hardware when a NAK handshake has been sent in response to an IN request from the Host.
This bit is cleared by software.

Bit 13 – STALL_SNT Stall Sent (cleared upon USB reset)
For Control, Bulk and Interrupt endpoints.
This bit is set by hardware after a STALL handshake has been sent as requested by the UDPHS_EPTSTAx register FRCESTALL bit.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

Bit 12 – RX_SETUP Received SETUP (cleared upon USB reset)
For Control endpoint only.
This bit is set by hardware when a valid SETUP packet has been received from the host.
It is cleared by the device firmware after reading the SETUP data from the endpoint FIFO.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint), and by UDPHS_EPTCTLDISx (disable endpoint).

Bit 11 – TXRDY TX Packet Ready (cleared upon USB reset)
This bit is cleared by hardware after the host has acknowledged the packet.
For Multi-bank endpoints, this bit may remain clear even after software is set if another bank is available to transmit.
Hardware clear of this bit may generate an interrupt if enabled by the UDPHS_EPTCTLx register TXRDY bit.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint), and by UDPHS_EPTCTLDISx (disable endpoint).

Bit 10 – TX_COMPLT Transmitted IN Data Complete (cleared upon USB reset)
This bit is set by hardware after an IN packet has been accepted (ACK'ed) by the host.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint), and by UDPHS_EPTCTLDISx (disable endpoint).

Bit 9 – RXRDY_TXKL Received OUT Data/KILL Bank (cleared upon USB reset)
Received OUT Data (for OUT endpoint or Control endpoint):

- This bit is set by hardware after a new packet has been stored in the endpoint FIFO.
- This bit is cleared by the device firmware after reading the OUT data from the endpoint.
- For multi-bank endpoints, this bit may remain active even when cleared by the device firmware, this if an other packet has been received meanwhile.

Downloaded from Arrow.com.

- Hardware assertion of this bit may generate an interrupt if enabled by the UDPHS_EPTCTLx register RXRDY_TXKL bit.
- This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

KILL Bank (for IN endpoint):

- The bank is really cleared or the bank is sent, BUSY_BANK_STA is decremented.
- The bank is not cleared but sent on the IN transfer, TX_COMPLT
- The bank is not cleared because it was empty. The user should wait that this bit is cleared before trying to clear another packet.
  Note: "Kill a packet" may be refused if at the same time, an IN token is coming and the current packet is sent on the UDPHS line. In this case, the TX_COMPLT bit is set. Take notice however, that if at least two banks are ready to be sent, there is no problem to kill a packet even if an IN token is coming. In fact, in that case, the current bank is sent (IN transfer) and the last bank is killed.

Bit 8 – ERR_OVFLW Overflow Error (cleared upon USB reset)
This bit is set by hardware when a new too-long packet is received.
Example: If the user programs an endpoint 64 bytes wide and the host sends 128 bytes in an OUT transfer, then the Overflow Error bit is set.
This bit is updated at the same time as the BYTE_COUNT field.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

Bits 7:6 – TOGGLESQ_STA[1:0] Toggle Sequencing (cleared upon USB reset)
In OUT transfer, the Toggle information is meaningful only when the current bank is busy (Received OUT Data = 1).
This field is updated for OUT transfer:

- A new data has been written into the current bank.
- The user has just cleared the Received OUT Data bit to switch to the next bank.

This field is reset to DATA1 by the UDPHS_EPTCLRSTAx register TOGGLESQ bit, and by UDPHS_EPTCTLDISx (disable endpoint).
Toggle Sequencing:

- IN endpoint: Indicates the PID Data Toggle that will be used for the next packet sent. This is not relative to the current bank.
- CONTROL and OUT endpoints: Set by hardware to indicate the PID data of the current bank.

Value	Name	Description
0	DATA0	DATA0
1	DATA1	DATA1
2	DATA2	Reserved for High Bandwidth Isochronous Endpoint
3	MDATA	Reserved for High Bandwidth Isochronous Endpoint

Bit 5 – FRCESTALL Stall Handshake Request (cleared upon USB reset)
This bit is reset by hardware upon received SETUP.

Value	Description
0	No effect.
1	If set a STALL answer will be done to the host for the next handshake.

Downloaded from Arrow.com.

40.7.20 UDPHS Endpoint Status Register (Isochronous Endpoint)

Name: UDPHS_EPTSTAx
Offset: 0x011C + x*0x20 [x=0..6]
Reset: 0x00000040
Property: Read-only

This register view is relevant only if EPT_TYPE = 0x1 in “UDPHS Endpoint Configuration Register”.

Bit	31	30	29	28	27	26	25	24
	SHRT_PCKT	BYTE_COUNT[10:4]						
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BYTE_COUNT[3:0]				BUSY_BANK_STA[1:0]		CURBK[1:0]	
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		ERR_FLUSH	ERR_CRC_NT	ERR_FL_ISO	TXRDY_TRER	TX_COMPLT	RXRDY_TXKL	ERR_OVFLW
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TOGGLESQ_STA[1:0]							
Access	R	R						
Reset	0	1						

Bit 31 – SHRT_PCKT Short Packet (cleared upon USB reset)
An OUT Short Packet is detected when the receive byte count is less than the configured UDPHS_EPTCFGx register EPT_Size.
This bit is updated at the same time as the BYTE_COUNT field.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

Bits 30:20 – BYTE_COUNT[10:0] UDPHS Byte Count (cleared upon USB reset)
Byte count of a received data packet.
This field is incremented after each write into the endpoint (to prepare an IN transfer).
This field is decremented after each reading into the endpoint (OUT transfer).
This field is also updated at RXRDY_TXKL flag clear with the next bank.
This field is also updated at TXRDY_TRER flag set with the next bank.
This field is reset by EPT_x of UDPHS_EPTRST register.

Bits 19:18 – BUSY_BANK_STA[1:0] Busy Bank Number (cleared upon USB reset)
These bits are set by hardware to indicate the number of busy banks.
IN endpoint: It indicates the number of busy banks filled by the user, ready for IN transfer.
OUT endpoint: It indicates the number of busy banks filled by OUT transaction from the Host.

Value	Name	Description
0	0BUSYBANK	All banks are free
1	1BUSYBANK	1 busy bank
2	2BUSYBANKS	2 busy banks
3	3BUSYBANKS	3 busy banks

Bits 17:16 – CURBK[1:0] Current Bank (cleared upon USB reset)
These bits are set by hardware to indicate the number of the current bank.

Downloaded from Arrow.com.

The current bank is updated each time the user:

- Sets the TX Packet Ready bit to prepare the next IN transfer and to switch to the next bank.
- Clears the received OUT data bit to access the next bank.

This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

Value	Name	Description
0	BANK0	Bank 0 (or single bank)
1	BANK1	Bank 1
2	BANK2	Bank 2

Bit 14 – ERR_FLUSH Bank Flush Error (cleared upon USB reset)
For High Bandwidth Isochronous IN endpoints.
This bit is set when flushing unsent banks at the end of a microframe.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by EPT_CTL_DISx (disable endpoint).

Bit 13 – ERR_CRC_NTR CRC ISO Error/Number of Transaction Error (cleared upon USB reset)
CRC ISO Error (for Isochronous OUT endpoints) (Read-only):
This bit is set by hardware if the last received data is corrupted (CRC error on data).
This bit is updated by hardware when new data is received (Received OUT Data bit).
Number of Transaction Error (for High Bandwidth Isochronous IN endpoints):
This bit is set at the end of a microframe in which at least one data bank has been transmitted, if less than the number of transactions per micro-frame banks (UDPHS_EPTCFGx register NB_TRANS) have been validated for transmission inside this microframe.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

Bit 12 – ERR_FL_ISO Error Flow (cleared upon USB reset)
This bit is set by hardware when a transaction error occurs.

- Isochronous IN transaction is missed, the micro has no time to fill the endpoint (underflow).
- Isochronous OUT data is dropped because the bank is busy (overflow).

This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

Bit 11 – TXRDY_TRER TX Packet Ready/Transaction Error (cleared upon USB reset)
TX Packet Ready
This bit is cleared by hardware, as soon as the packet has been sent.
For Multi-bank endpoints, this bit may remain clear even after software is set if another bank is available to transmit.
Hardware clear of this bit may generate an interrupt if enabled by the UDPHS_EPTCTLx register TXRDY_TRER bit.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint), and by UDPHS_EPTCTLDISx (disable endpoint).
Transaction Error (for high bandwidth isochronous OUT endpoints) (Read-Only):
This bit is set by hardware when a transaction error occurs inside one microframe.
If one toggle sequencing problem occurs among the n-transactions (n = 1, 2 or 3) inside a microframe, then this bit is still set as long as the current bank contains one “bad” n-transaction (see CURBK field description). As soon as the current bank is relative to a new “good” n-transactions, then this bit is reset.
Note:

1. A transaction error occurs when the toggle sequencing does not comply with the Universal Serial Bus Specification, Rev 2.0 (5.9.2 High Bandwidth Isochronous endpoints) (Bad PID, missing data, etc.)
2. When a transaction error occurs, the user may empty all the “bad” transactions by clearing the Received OUT Data flag (RXRDY_TXKL).

If this bit is reset, then the user should consider that a new n-transaction is coming.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint), and by UDPHS_EPTCTLDISx (disable endpoint).

Downloaded from Arrow.com.

Bit 10 – TX_COMPLT Transmitted IN Data Complete (cleared upon USB reset)
This bit is set by hardware after an IN packet has been sent.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint), and by UDPHS_EPTCTLDISx (disable endpoint).

Bit 9 – RXRDY_TXKL Received OUT Data/KILL Bank (cleared upon USB reset)
Received OUT Data (for OUT endpoint or Control endpoint):

- This bit is set by hardware after a new packet has been stored in the endpoint FIFO.
- This bit is cleared by the device firmware after reading the OUT data from the endpoint.
- For multi-bank endpoints, this bit may remain active even when cleared by the device firmware, this if an other packet has been received meanwhile.
- Hardware assertion of this bit may generate an interrupt if enabled by the UDPHS_EPTCTLx register RXRDY_TXKL bit.
- This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

KILL Bank (for IN endpoint):

- The bank is really cleared or the bank is sent, BUSY_BANK_STA is decremented.
- The bank is not cleared but sent on the IN transfer, TX_COMPLT
- The bank is not cleared because it was empty. The user should wait that this bit is cleared before trying to clear another packet.
  Note: "Kill a packet" may be refused if at the same time, an IN token is coming and the current packet is sent on the UDPHS line. In this case, the TX_COMPLT bit is set. Take notice however, that if at least two banks are ready to be sent, there is no problem to kill a packet even if an IN token is coming. In fact, in that case, the current bank is sent (IN transfer) and the last bank is killed.

Bit 8 – ERR_OVFLW Overflow Error (cleared upon USB reset)
This bit is set by hardware when a new too-long packet is received.
Example: If the user programs an endpoint 64 bytes wide and the host sends 128 bytes in an OUT transfer, then the Overflow Error bit is set.
This bit is updated at the same time as the BYTE_COUNT field.
This bit is reset by UDPHS_EPTRST register EPT_x (reset endpoint) and by UDPHS_EPTCTLDISx (disable endpoint).

Bits 7:6 – TOGGLESQ_STA[1:0] Toggle Sequencing (cleared upon USB reset)
In OUT transfer, the Toggle information is meaningful only when the current bank is busy (Received OUT Data = 1).
This field is updated for OUT transfer:

- A new data has been written into the current bank.
- The user has just cleared the Received OUT Data bit to switch to the next bank.

For High Bandwidth Isochronous Out endpoint, it is recommended to check the UDPHS_EPTSTAx/TXRDY_TRER bit to know if the toggle sequencing is correct or not.
This field is reset to DATA1 by the UDPHS_EPTCLRSTAx register TOGGLESQ bit, and by UDPHS_EPTCTLDISx (disable endpoint).
Toggle Sequencing:

- IN endpoint: Indicates the PID Data Toggle that will be used for the next packet sent. This is not relative to the current bank.
- OUT endpoint: Set by hardware to indicate the PID data of the current bank:

Value	Name	Description
0	DATA0	DATA0
1	DATA1	DATA1
2	DATA2	Data2 (only for High Bandwidth Isochronous Endpoint)
3	MDATA	MData (only for High Bandwidth Isochronous Endpoint)

Downloaded from Arrow.com.

40.7.21 UDPHS DMA Channel Transfer Descriptor

The DMA channel transfer descriptor is loaded from the memory. Be careful with the alignment of this buffer. The structure of the DMA channel transfer descriptor is defined by three parameters as described below:

- Offset 0:
  - The address must be aligned: 0xXXXX0
  - Next Descriptor Address Register: UDPHS_DMANXTDSCx
- Offset 4:
  - The address must be aligned: 0xXXXX4
  - DMA Channelx Address Register: UDPHS_DMAADDRESSx
- Offset 8:
  - The address must be aligned: 0xXXXX8
  - DMA Channelx Control Register: UDPHS_DMACONTROLx

To use the DMA channel transfer descriptor, fill the structures with the correct value (as described in the following pages). Then write directly in UDPHS_DMANXTDSCx the address of the descriptor to be used first. Then write '1' in the LDNXT_DSC bit of UDPHS_DMACONTROLx (load next channel transfer descriptor). The descriptor is automatically loaded upon Endpointx request for packet transfer.

Downloaded from Arrow.com.

40.7.22 UDPHS DMA Next Descriptor Address Register

Name: UDPHS_DMANXTDSCx
Offset: 0x0310 + (x-1)*0x10 [x=1..6]
Reset: 0x00000000
Property: Read/Write

Channel 0 is not used.

Bit	31	30	29	28	27	26	25	24
	NXT_DSC_ADD[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NXT_DSC_ADD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NXT_DSC_ADD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NXT_DSC_ADD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NXT_DSC_ADD[31:0] Next Descriptor Address
This field points to the next channel descriptor to be processed. This channel descriptor must be aligned, so bits 0 to 3 of the address must be equal to zero.

Downloaded from Arrow.com.

40.7.23 UDPHS DMA Channel Address Register

Name: UDPHS_DMAADDRESSx
Offset: 0x0314 + (x-1)*0x10 [x=1..6]
Reset: 0x00000000
Property: Read/Write

Channel 0 is not used.

Bit	31	30	29	28	27	26	25	24
	BUFF_ADD[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BUFF_ADD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BUFF_ADD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BUFF_ADD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BUFF_ADD[31:0] Buffer Address
This field determines the AHB bus starting address of a DMA channel transfer.
Channel start and end addresses may be aligned on any byte boundary.
The firmware may write this field only when the UDPHS_DMASTATUS.CHANN_ENB bit is clear.
This field is updated at the end of the address phase of the current access to the AHB bus. It is incrementing of the access byte width. The access width is 4 bytes (or less) at packet start or end, if the start or end address is not aligned on a word boundary.
The packet start address is either the channel start address or the next channel address to be accessed in the channel buffer.
The packet end address is either the channel end address or the latest channel address accessed in the channel buffer.
The channel start address is written by software or loaded from the descriptor, whereas the channel end address is either determined by the end of buffer or the UDPHS device, USB end of transfer if the UDPHS_DMACONTROL.END_TR_EN bit is set.

Downloaded from Arrow.com.

40.7.24 UDPHS DMA Channel Control Register

Name: UDPHS_DMACONTROLx
Offset: 0x0318 + (x-1)*0x10 [x=1..6]
Reset: 0x00000000
Property: Read/Write

Channel 0 is not used.

Bits [31:2] are only writable when issuing a channel Control Command other than “Stop Now”.

For reliability it is highly recommended to wait for both UDPHS_DMASTATUS.CHAN_ACT and CHAN_ENB flags at 0, thus ensuring the channel has been stopped before issuing a command other than “Stop Now”.

Bit	31	30	29	28	27	26	25	24
	BUFF_LENGTH[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BUFF_LENGTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – BUFF_LENGTH[15:0] Buffer Byte Length (Write-only)
This field determines the number of bytes to be transferred until end of buffer. The maximum channel transfer size (64 Kbytes) is reached when this field is 0 (default value). If the transfer size is unknown, this field should be set to 0, but the transfer end may occur earlier under UDPHS device control.
When this field is written, the UDPHS_DMASTATUS.BUFF_COUNT field is updated with the write value.

Bit 7 – BURST_LCK Burst Lock Enable

Value	Description
0	The DMA never locks bus access.
1	USB packets AHB data bursts are locked for maximum optimization of the bus bandwidth usage and maximization of fly-by AHB burst duration.

Bit 6 – DESC_LD_IT Descriptor Loaded Interrupt Enable

Value	Description
0	UDPHS_DMASTATUS.DESC_LDST rising will not trigger any interrupt.
1	An interrupt is generated when a descriptor has been loaded from the bus.

Bit 5 – END_BUFFIT End of Buffer Interrupt Enable

Value	Description
0	UDPHS_DMASTATUS.END_BF_ST rising will not trigger any interrupt.
1	An interrupt is generated when the UDPHS_DMASTATUS.BUFF_COUNT reaches zero.

Bit 4 – END_TR_IT End of Transfer Interrupt Enable

Downloaded from Arrow.com.

Value	Description
0	UDPHS device initiated buffer transfer completion will not trigger any interrupt at UDPHS_STATUS.END_TR_ST rising.
1	An interrupt is sent after the buffer transfer is complete, if the UDPHS device has ended the buffer transfer. Use when the receive size is unknown.

Bit 3 – END_B_EN End of Buffer Enable (Control)

Value	Description
0	DMA Buffer End has no impact on USB packet transfer.
1	Endpoint can validate the packet (according to the values programmed in the UDPHS_EPTCTL.AUTO_VALID and SHRT_PCKT fields) at DMA Buffer End, i.e., when the UDPHS_DMASTATUS.BUFF_COUNT reaches 0. This is mainly for short packet IN validation initiated by the DMA reaching end of buffer, but could be used for OUT packet truncation (discarding of unwanted packet data) at the end of DMA buffer.

Bit 2 – END_TR_EN End of Transfer Enable (Control)
Used for OUT transfers only.

Value	Description
0	USB end of transfer is ignored.
1	UDPHS device can put an end to the current buffer transfer. When set, a BULK or INTERRUPT short packet or the last packet of an ISOCHRONOUS (micro) frame (DATAX) will close the current buffer and the UDPHS_DMASTATUS.END_TR_ST flag will be raised. This is intended for UDPHS non-prenegotiated end of transfer (BULK or INTERRUPT) or ISOCHRONOUS microframe data buffer closure.

Bit 1 – LDNXT_DSC Load Next Channel Transfer Descriptor Enable (Command)
If the CHANN_ENB bit is cleared, the next descriptor is immediately loaded upon transfer request.
DMA Channel Control Command Summary:

LDNXT_DSC	CHANN_ENB	Description
0	0	Stop now
0	1	Run and stop at end of buffer
1	0	Load next descriptor now
1	1	Run and link at end of buffer

Value	Description
0	No channel register is loaded after the end of the channel transfer.
1	The channel controller loads the next descriptor after the end of the current transfer, i.e., when the UDPHS_DMASTATUS.CHANN_ENB bit is reset.

Bit 0 – CHANN_ENB (Channel Enable Command)

Downloaded from Arrow.com.

Value	Description
0	DMA channel is disabled at and no transfer will occur upon request. This bit is also cleared by hardware when the channel source bus is disabled at end of buffer. If the UDPHS_DMACONTROL.LDNXT_DSC bit has been cleared by descriptor loading, the firmware will have to set the corresponding CHANN_ENB bit to start the described transfer, if needed. If the LDNXT_DSC bit is cleared, the channel is frozen and the channel registers may then be read and/or written reliably as soon as both UDPHS_DMASTATUS.CHANN_ENB and CHANN_ACT flags read as 0. If a channel request is currently serviced when this bit is cleared, the DMA FIFO buffer is drained until it is empty, then the UDPHS_DMASTATUS.CHANN_ENB bit is cleared. If the LDNXT_DSC bit is set at or after this bit clearing, then the currently loaded descriptor is skipped (no data transfer occurs) and the next descriptor is immediately loaded.
1	The UDPHS_DMASTATUS.CHANN_ENB bit will be set, thus enabling DMA channel data transfer. Then, any pending request will start the transfer. This may be used to start or resume any requested transfer.

Downloaded from Arrow.com.

40.7.25 UDPHS DMA Channel Status Register

Name: UDPHS_DMASTATUSx
Offset: 0x031C + (x-1)*0x10 [x=1..6]
Reset: 0x00000000
Property: Read/Write

Channel 0 is not used.

Bit	31	30	29	28	27	26	25	24
	BUFF_COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BUFF_COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB
Access		R/W	R/W	R/W			R/W	R/W
Reset		0	0	0			0	0

Bits 31:16 – BUFF_COUNT[15:0] Buffer Byte Count
This field determines the current number of bytes still to be transferred for this buffer. It is decremented from the AHB source bus access byte width at the end of this bus address phase.
The access byte width is 4 by default, or less, at DMA start or end, if the start or end address is not aligned on a word boundary.
At the end of buffer, the DMA accesses the UDPHS device only for the number of bytes needed to complete it.
This field value is reliable (stable) only if the channel has been stopped or frozen (the UDPHS_EPTCTLx.NT_DIS_DMA bit is used to disable the channel request) and the channel is no longer active (CHANN_ACT flag is 0).
Note: For OUT endpoints, if the receive buffer byte length (BUFF_LENGTH) has been defaulted to zero because the USB transfer length is unknown, the actual buffer byte length received is 0x10000-BUFF_COUNT.

Bit 6 – DESC_LDST Descriptor Loaded Status
Valid until the CHANN_ENB flag is cleared at the end of the next buffer transfer.

Value	Description
0	Cleared automatically when read by software.
1	Set by hardware when a descriptor has been loaded from the system bus.

Bit 5 – END_BF_ST End of Channel Buffer Status
Valid until the CHANN_ENB flag is cleared at the end of the next buffer transfer.

Value	Description
0	Cleared automatically when read by software.
1	Set by hardware when the BUFF_COUNT countdown reaches zero.

Bit 4 – END_TR_ST End of Channel Transfer Status
Valid until the CHANN_ENB flag is cleared at the end of the next buffer transfer.

Value	Description
0	Cleared automatically when read by software.

Downloaded from Arrow.com.

Value	Description
1	Set by hardware when the last packet transfer is complete, if the UDPHS device has ended the transfer.

Bit 1 – CHANN_ACT Channel Active Status
When a packet transfer is ended, this bit is automatically reset.
When a packet transfer cannot be completed due to an END_BF_ST, this flag stays set during the next channel descriptor load (if any) and potentially until UDPHS packet transfer completion, if allowed by the new descriptor.

Value	Description
0	The DMA channel is no longer trying to source the packet data.
1	The DMA channel is currently trying to source packet data, i.e., selected as the highest-priority requesting channel.

Bit 0 – CHANN_ENB Channel Enable Status
When any transfer is ended either due to an elapsed byte count or a UDPHS device initiated transfer end, this bit is automatically reset.
This bit is normally set or cleared by writing into the UDPHS_DMACONTROLx.CHANN_ENB bit either by software or descriptor loading.
If a channel request is currently serviced when the CHANN_ENB bit is cleared, the DMA FIFO buffer is drained until it is empty, then this status bit is cleared.

Value	Description
0	The DMA channel no longer transfers data, and may load the next descriptor if the UDPHS_DMACONTROLx.LDNXT_DSC bit is set.
1	The DMA channel is currently enabled and transfers data upon request.

Downloaded from Arrow.com.

41. USB Host High Speed Port (UHPHS)

41.1 Description

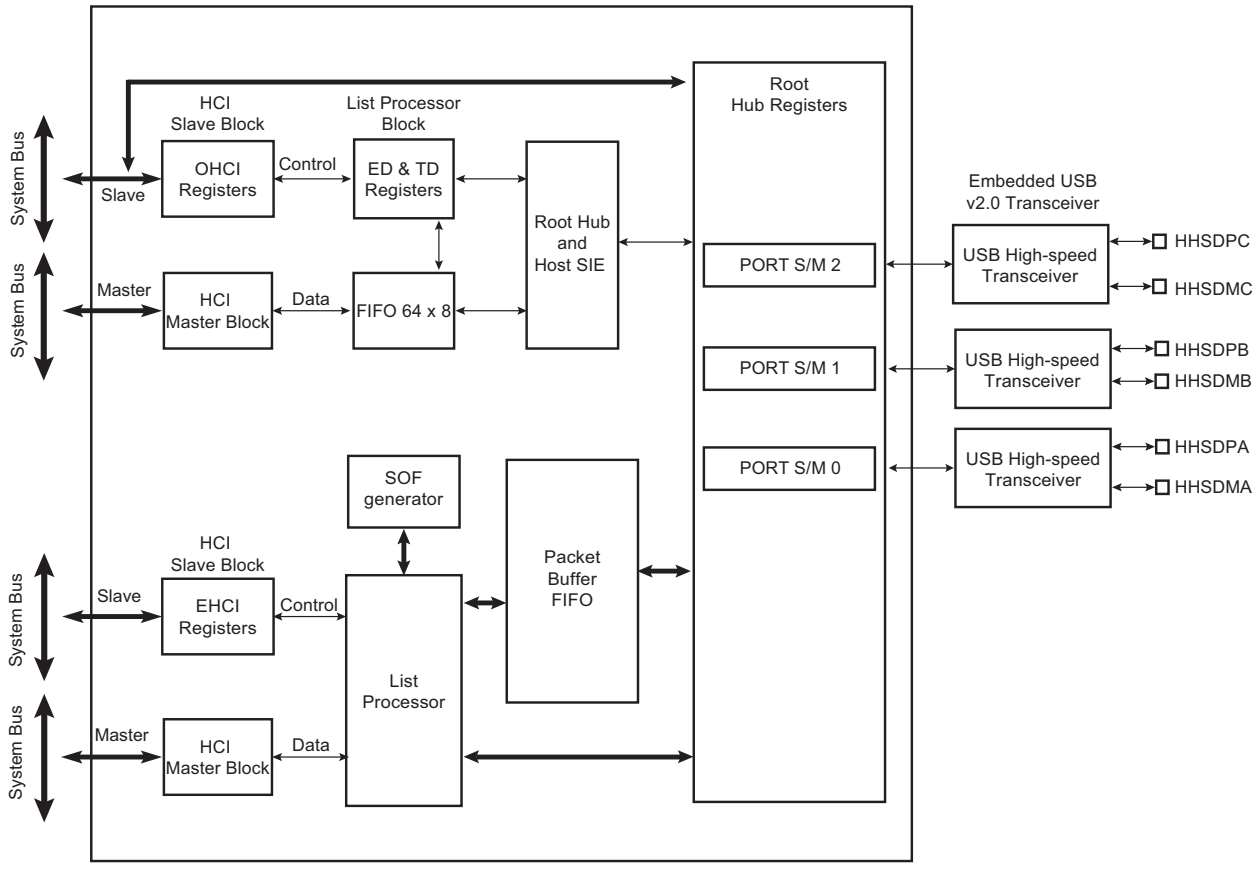
The USB Host High Speed Port (UHPHS) interfaces the USB with the host application. It handles Open HCI protocol (Open Host Controller Interface) as well as Enhanced HCI protocol (Enhanced Host Controller Interface).

41.2 Embedded Characteristics

- Compliant with Enhanced HCI Rev 1.0 Specification
  - Compliant with USB V2.0 High-speed Specification
  - Supports High-speed 480 Mbps
- Compliant with Open HCI Rev 1.0 Specification
  - Compliant with USB V2.0 Full-speed and Low-speed Specification
  - Supports both Low-speed 1.5 Mbps and Full-speed 12 Mbps USB devices
- Root Hub Integrated with 2 Downstream USB HS Ports and 1 FS Port
- Embedded USB Transceivers
- Supports Power Management
- 3 Hosts (A, B, and C) High Speed (EHCI), Port A shared with UDPHS

41.3 Block Diagram

Figure 41-1. Block Diagram



Downloaded from Arrow.com.

Access to the USB host operational registers is achieved through the system bus slave interface. The Open HCI host controller and Enhanced HCI host controller initialize master DMA transfers through the system bus master interface as follows:

- Fetches endpoint descriptors and transfer descriptors
- Accesses endpoint data from system memory
- Accesses HC communication area
- Writes status and retires transfer descriptor

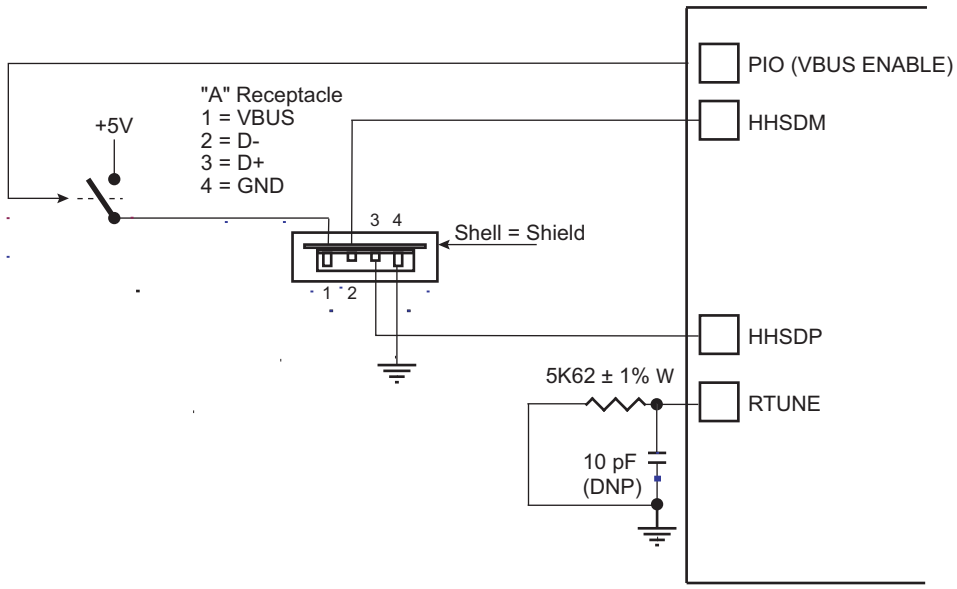
Memory access errors (abort, misalignment) lead to an “Unrecoverable Error” indicated by the corresponding flag in the host controller operational registers.

The USB root hub is integrated in the USB host. Several USB downstream ports are available. The number of downstream ports can be determined by the software driver reading the root hub’s operational registers. Device connection is automatically detected by the USB host port logic.

USB physical transceivers are integrated in the product and driven by the root hub’s ports.

41.4 Typical Connection

Figure 41-2. Board Schematic to Interface UHP High-speed Host Controller



41.5 Product Dependencies

41.5.1 I/O Lines

HHSDPs and HHSDMs are not controlled by any PIO controllers. The embedded USB High Speed physical transceivers are controlled by the USB host controller.

One transceiver is shared with the USB High Speed Device (port A). The selection between Host Port A and USB Device is controlled by the UDPHS enable bit (EN_UDPHS) located in the UDPHS_CTRL register.

In the case the port A is driven by the USB High Speed Device, the output signals are DHSDP and DHSDM. The transceiver is automatically selected for Device operation once the USB High Speed Device is enabled.

In the case the port A is driven by the USB High Speed Host, the output signals are HHSDPA and HHSDMA.

41.5.2 Power Management

The system embeds 3 transceivers.

Downloaded from Arrow.com.

The USB Host High Speed requires a 480 MHz clock for the embedded High-speed transceivers. This clock (UPLLCK) is provided by the UTMI PLL.

In case power consumption is saved by stopping the UTMI PLL, high-speed operations are not possible. Nevertheless, OHCI Full-speed operations remain possible by selecting PLLACK as the input clock of OHCI.

The High-speed transceiver returns a 30 MHz clock to the USB Host controller.

The USB Host controller requires 48 MHz and 12 MHz clocks for OHCI full-speed operations. These clocks must be generated by a PLL with a correct accuracy of +/-0.25% using the USBDIV field.

Thus the USB Host peripheral receives three clocks from the Power Management Controller (PMC): the Peripheral Clock (MCK domain), the UHP48M and the UHP12M (built-in UHP48M divided by four) used by the OHCI to interface with the bus USB signals (recovered 12 MHz domain) in Full-speed operations.

For High-speed operations, the user has to perform the following:

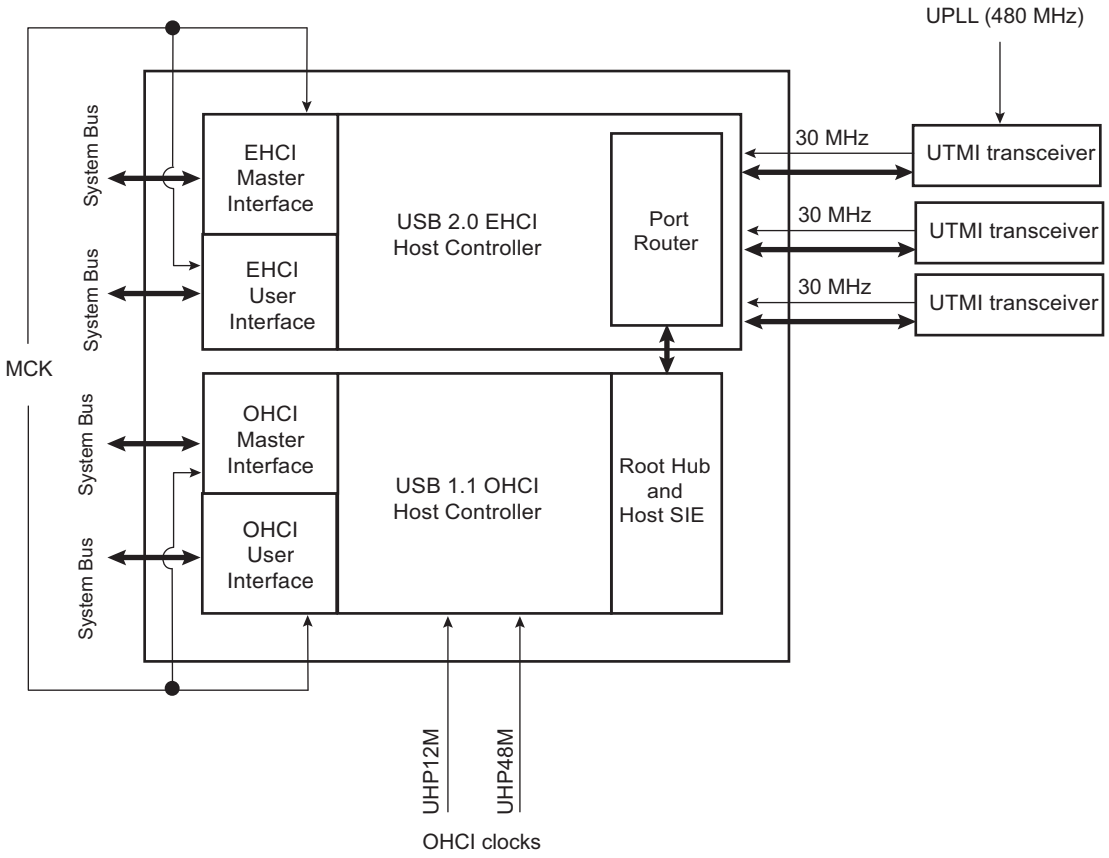
- Enable UHPHS peripheral clock in the PMC.
- Start the UPLL with a 480 MHz output frequency following the recommendation provided in section "USB Clock Controller" in chapter "Power Management Controller (PMC)".
- Select UPLLCK as Input clock of OHCI part (USBS bit in PMC_USB register).
- Program OHCI clocks (UHP48M and UHP12M) with USBDIV field in PMC_USB register. USBDIV must be 9 (division by 10) if UPLLCK is selected.
- Enable OHCI clocks with UHP bit in PMC_SCER.

For OHCI Full-speed operations only, the user can use the PLLA, and in this case has to perform the following:

- Enable UHPHS peripheral clock in the PMC.
- Select PLLACK as Input clock of OHCI part (USBS bit in PMC_USB register).
- Program OHCI clocks (UHP48M and UHP12M) with USBDIV field in PMC_USB register. USBDIV value is to be calculated according to the PLLACK value and USB Full-speed accuracy.
- Enable the OHCI clocks with UHP bit in PMC_SCER.

Downloaded from Arrow.com.

Figure 41-3. UHP Clock Trees



41.5.3 Interrupt Sources

The USB host interface has an interrupt line connected to the interrupt controller.

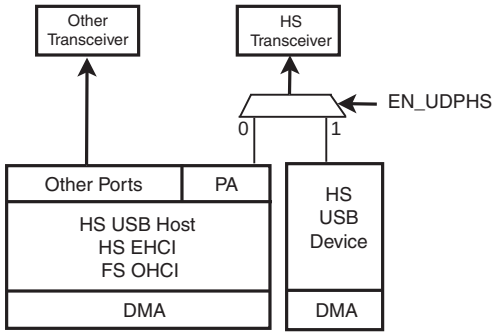
Handling USB host interrupts requires programming the interrupt controller before configuring the UHPHS.

41.6 Functional Description

41.6.1 UTMI Transceivers Sharing

The High Speed USB Host Port A is shared with the High Speed USB Device port and connected to the second UTMI transceiver. The selection between Host Port A and USB device is controlled by the UDPHS enable bit (EN_UDPHS) located in the UDPHS_CTRL register.

Figure 41-4. USB Selection



Downloaded from Arrow.com.

41.6.2 EHCI

The USB Host Port controller is fully compliant with the Enhanced HCI specification. The USB Host Port User Interface (registers description) can be found in the Enhanced HCI Rev 1.0 Specification available on www.usb.org

41.6.3 OHCI

The USB Host Port integrates a root hub and transceivers on downstream ports. It provides several Full-speed half-duplex serial communication ports at a baud rate of 12 Mbps. Up to 127 USB devices (printer, camera, mouse, keyboard, disk, etc.) and the USB hub can be connected to the USB host in the USB “tiered star” topology.

The USB Host Port controller is fully compliant with the Open HCI specification. The USB Host Port User Interface (registers description) can be found in the Open HCI Rev 1.0 Specification available on www.usb.org.

All standard class devices are automatically detected and available to the user’s application. As an example, integrating an HID (Human Interface Device) class driver provides a plug & play feature for all USB keyboards and mouses.

Downloaded from Arrow.com.

41.7 Register Summary

The Enhanced USB Host Controller contains two sets of software-accessible hardware registers: memory-mapped Host Controller Registers and optional PCI configuration registers. Note that the PCI configuration registers are only needed for PCI devices that implement the Host Controller.

- Memory-mapped USB Host Controller Registers—This block of registers is memory-mapped into non-cacheable memory. This memory space must begin on a DWord (32-bit) boundary. This register space is divided into two sections: a set of read-only capability registers and a set of read/write operational registers. The table below describes each register space.
  Note: Host controllers are not required to support exclusive-access mechanisms (such as PCI LOCK) for accesses to the memory-mapped register space. Therefore, if software attempts exclusive-access mechanisms to the host controller memory-mapped register space, the results are undefined.
- PCI Configuration Registers (for PCI devices)—In addition to the normal PCI header, power management, and device-specific registers, two registers are needed in the PCI configuration space to support USB. The normal PCI header and device-specific registers are beyond the scope of this document (the UHPHS_CLASSC register is shown in this document). Note that HCD does not interact with the PCI configuration space. This space is used only by the PCI enumerator to identify the USB Host Controller, and assign the appropriate system resources.
  The table below summarizes the enhanced interface register sets.

Offset	Register Set	Explanation
0 to N-1	Capability Registers	The capability registers specify the limits, restrictions, and capabilities of a host controller implementation. These values are used as parameters to the host controller driver.
N to N+M-1	Operational Registers	The operational registers are used by system software to control and monitor the operational state of the host controller.

Note: Software must not modify reserved bits in Read/Write registers.

Offset	Name	Bit Pos.								
0x00	UHPHS_HCCAPBASE	31:24	HCIVERSION[15:8]							
		23:16	HCIVERSION[7:0]							
		15:8								
		7:0	CAPLENGTH[7:0]							
0x04	UHPHS_HCSPARAMS	31:24								
		23:16	N_DP[3:0]							P_INDICATOR
		15:8	N_CC[3:0]				N_PCC[3:0]			
		7:0				PPC	N_PORTS[3:0]			
0x08	UHPHS_HCCPARAMS	31:24								
		23:16								
		15:8	EECP[7:0]							
		7:0	IST[3:0]					ASPC	PFLF	AC
0x0C ... 0x0F	Reserved									
0x10	UHPHS_USBCMD	31:24								
		23:16	ITC[7:0]							
		15:8					ASPME		ASPMC[1:0]	
		7:0	LHCR	IAAD	ASE	PSE	FLS[1:0]		HCRESET	RS
0x14	UHPHS_USBSTS	31:24								
		23:16								
		15:8	ASS	PSS	RCM	HCHLT				
		7:0			IAA	HSE	FLR	PCD	USBERRINT	USBINT

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x18	UHPHS_USBINTR	31:24								
		23:16								
		15:8								
		7:0			IAAE	HSEE	FLRE	PCIE	USBEIE	USBIE
0x1C	UHPHS_FRINDEX	31:24								
		23:16								
		15:8			FI[13:8]					
		7:0	FI[7:0]							
0x20 ... 0x23	Reserved									
0x24	UHPHS_PERIODICLISTBASE	31:24	BA[19:12]							
		23:16	BA[11:4]							
		15:8	BA[3:0]							
		7:0								
0x28	UHPHS_ASYNCLISTADDR	31:24	LPL[26:19]							
		23:16	LPL[18:11]							
		15:8	LPL[10:3]							
		7:0	LPL[2:0]							
0x2C ... 0x4F	Reserved									
0x50	UHPHS_CONFIGFLAG	31:24								
		23:16								
		15:8								
		7:0								CF
0x54	UHPHS_PORTSC0	31:24								
		23:16		WKOC_E	WKDSCNNT_E	WKCNNT_E	PTC[3:0]			
		15:8	PIC[1:0]		PO	PP	LS[1:0]			PR
		7:0	SUS	FPR	OCC	OCA	PEDC	PED	CSC	CCS
0x58	UHPHS_PORTSC1	31:24								
		23:16		WKOC_E	WKDSCNNT_E	WKCNNT_E	PTC[3:0]			
		15:8	PIC[1:0]		PO	PP	LS[1:0]			PR
		7:0	SUS	FPR	OCC	OCA	PEDC	PED	CSC	CCS
0x5C	UHPHS_PORTSC2	31:24								
		23:16		WKOC_E	WKDSCNNT_E	WKCNNT_E	PTC[3:0]			
		15:8	PIC[1:0]		PO	PP	LS[1:0]			PR
		7:0	SUS	FPR	OCC	OCA	PEDC	PED	CSC	CCS
0x60 ... 0xA7	Reserved									
0xA8	UHPHS_INSNREG06	31:24	AHB_ERR							
		23:16								
		15:8					HBURST[2:0]			Nb_Burst[4]
		7:0	Nb_Burst[3:0]				Nb_Success_Burst[3:0]			
0xAC	UHPHS_INSNREG07	31:24	AHB_ADDR[31:24]							
		23:16	AHB_ADDR[23:16]							
		15:8	AHB_ADDR[15:8]							
		7:0	AHB_ADDR[7:0]							

Downloaded from Arrow.com.

41.7.1 UHPHS Host Controller Capability Register

Name: UHPHS_HCCAPBASE
Offset: 0x00
Reset: 0x01000010
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	HCIVERSION[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	1

Bit	23	22	21	20	19	18	17	16
	HCIVERSION[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CAPLENGTH[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	0	0	0	0

Bits 31:16 – HCIVERSION[15:0] Host Controller Interface Version Number
This is a two-byte field containing a BCD encoding of the EHCI revision number supported by this host controller. The most significant byte of this field represents the major revision and the least significant byte the minor revision.

Bits 7:0 – CAPLENGTH[7:0] Capability Registers Length
This field is used as an offset to add to the register base to find the beginning of the Operational Register Space.

Downloaded from Arrow.com.

41.7.2 UHPHS Host Controller Structural Parameters Register

Name: UHPHS_HCSPARAMS
Offset: 0x04
Reset: 0x00001116
Property: Read-only

These fields define structural parameters: number of downstream ports, etc.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	N_DP[3:0]							P_INDICATOR
Access	R	R	R	R				R
Reset	0	0	0	0				0

Bit	15	14	13	12	11	10	9	8
	N_CC[3:0]				N_PCC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	0	0	0	1

Bit	7	6	5	4	3	2	1	0
				PPC	N_PORTS[3:0]			
Access				R	R	R	R	R
Reset				1	0	1	1	0

Bits 23:20 – N_DP[3:0] Debug Port Number
Optional. This register identifies which of the host controller ports is the debug port. The value is the port number (1-based) of the debug port. A non-zero value in this field indicates the presence of a debug port. The value in this register must not be greater than N_PORTS.

Bit 16 – P_INDICATOR Port Indicators
This bit indicates whether the ports support port indicator control. When this bit is a 1, the port status and control registers include a read/writeable field for controlling the state of the port indicator. See UHPHS Port Status and Control Register for a definition of the port indicator control field.

Bits 15:12 – N_CC[3:0] Number of Companion Controllers
This field indicates the number of companion controllers associated with this USB 2.0 host controller.
A zero in this field indicates there are no companion host controllers. Port-ownership hand-off is not supported. Only high-speed devices are supported on the host controller root ports.
A value larger than zero in this field indicates there are companion USB 1.1 host controller(s). Port-ownership hand-offs are supported. High, Full- and Low-speed devices are supported on the host controller root ports.

Bits 11:8 – N_PCC[3:0] Number of Ports per Companion Controller
This field indicates the number of ports supported per companion host controller. It is used to indicate the port routing configuration to system software.
For example, if N_PORTS has a value of 6 and N_CC has a value of 2, then N_PCC could have a value of 3. The convention is that the first N_PCC ports are assumed to be routed to companion controller 1, the next N_PCC ports to companion controller 2, etc. In the previous example, the N_PCC could have been 4, where the first four are routed to companion controller 1 and the last two are routed to companion controller 2.
The number in this field must be consistent with N_PORTS and N_CC.

Downloaded from Arrow.com.

Bit 4 – PPC Port Power Control
This field indicates whether the host controller implementation includes port power control. A one in this bit indicates the ports have port power switches. A zero in this bit indicates the ports do not have port power switches. The value of this field affects the functionality of the Port Power field in each port status and control register (see UHPHS Port Status and Control Register).

Bits 3:0 – N_PORTS[3:0] Number of Ports
This field specifies the number of physical downstream ports implemented on this host controller. The value of this field determines how many port registers are addressable in the Operational Register Space. Valid values are in the range of 1 to 15.
A zero in this field is undefined.

Downloaded from Arrow.com.

41.7.3 UHPHS Host Controller Capability Parameters Register

Name: UHPHS_HCCPARAMS
Offset: 0x08
Reset: 0x0000A010
Property: Read-only

These fields define capability parameters: Multiple Mode control (time-base bit functionality), addressing capability, etc.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EECP[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	1	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IST[3:0]					ASPC	PFLF	AC
Access	R	R	R	R		R	R	R
Reset	0	0	0	1		0	0	0

Bits 15:8 – EECP[7:0] EHCI Extended Capabilities Pointer
Indicates the existence of a capabilities list. A value of 0 indicates no extended capabilities are implemented. A non-zero value in this register indicates the offset in the PCI configuration space of the first EHCI extended capability. The pointer value must be 64 or greater if implemented to maintain the consistency of the PCI header defined for this class of device.

Bits 7:4 – IST[3:0] Isochronous Scheduling Threshold
Indicates, relative to the current position of the executing host controller, where software can reliably update the isochronous schedule. When bit [7] is 0, the value of the least significant three bits indicates the number of microframes a host controller can hold a set of isochronous data structures (one or more) before flushing the state. When bit [7] is set to 1, the host software assumes the host controller may cache an isochronous data structure for an entire frame.

Bit 2 – ASPC Asynchronous Schedule Park Capability
The park capability can be disabled or enabled and set to a specific level by using the Asynchronous Schedule Park Mode Enable and Asynchronous Schedule Park Mode Count fields in the UHPHS_USBCMD register.

Value	Description
0	Host controller does not supports the park feature for high-speed queue.
1	Host controller supports the park feature for high-speed queue heads in the Asynchronous Schedule.

Bit 1 – PFLF Programmable Frame List Flag

Value	Description
0	System software must use a frame list length of 1024 elements with this host controller. The UHPHS_USBCMD register Frame List Size field is a read-only register and must be set to 0.
1	System software can specify and use a smaller frame list and configure the host controller via the UHPHS_USBCMD register Frame List Size field. The frame list must always be aligned on a 4-Kbyte page boundary. This requirement ensures that the frame list is always physically contiguous.

Downloaded from Arrow.com.

Bit 0 – AC 64-bit Addressing Capability
This field documents the addressing range capability of this implementation. The value of this field determines whether software should use 32-bit or 64-bit data structures.
This information is not tightly coupled with the UHPHS_USBBASE address register mapping control. The 64-bit Addressing Capability bit indicates whether the host controller can generate 64-bit addresses as a master. The UHPHS_USBBASE register indicates the host controller only needs to decode 32-bit addresses as a slave.

Value	Description
0	Data structures using 32-bit address memory pointers
1	Data structures using 64-bit address memory pointers

Downloaded from Arrow.com.

41.7.4 UHPHS USB Command Register

Name: UHPHS_USBCMD
Offset: 0x10
Reset: 0x00080B00
Property: Read/Write

The Command Register indicates the command to be executed by the serial bus host controller. Writing to the register causes a command to be executed.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	ITC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8
					ASPME		ASPMC[1:0]	
Access					R-R/W		R-R/W	R-R/W
Reset					1		1	1

Bit	7	6	5	4	3	2	1	0
	LHCR	IAAD	ASE	PSE	FLS[1:0]		HCRESET	RS
Access	R/W	R/W	R/W	R/W	R-R/W	R-R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – ITC[7:0] Interrupt Threshold Control
This field is used by system software to select the maximum rate at which the host controller will issue interrupts. The only valid values are defined below. If software writes an invalid value to this register, the results are undefined.

Value	Maximum Interrupt Interval
0	Reserved
1	1 microframe
2	2 microframes
4	4 microframes
8	8 microframes (1 ms)
16	16 microframes (2 ms)
32	32 microframes (4 ms)
64	64 microframes (8 ms)

Any other value in this register yields undefined results.
Software modifications to this field while HCHLT=0 results in undefined behavior.

Bit 11 – ASPME Asynchronous Schedule Park Mode Enable (optional)
If the Asynchronous Park Capability bit in the UHPHS_HCCPARAMS register is set to 1, then this bit is set to 1 and is Read/Write. Otherwise the bit must be 0 and is read-only.

Value	Description
0	Park mode is enabled.
1	Park mode is disabled.

Downloaded from Arrow.com.

Bits 9:8 – ASPMC[1:0] Asynchronous Schedule Park Mode Count (optional)
If the Asynchronous Park Capability bit in the UHPHS_HCCPARAMS register is set to 1, then this field defaults to 3 and is read/write. Otherwise it defaults to 0 and is read-only. It contains a count of the number of successive transactions the host controller is allowed to execute from a high-speed queue head on the Asynchronous schedule before continuing traversal of the Asynchronous schedule. Valid values are 1 to 3. Software must not write a 0 to this bit when Park Mode Enable is set to 1 as this will result in undefined behavior.

Bit 7 – LHCR Light Host Controller Reset (optional)
This control bit is not required. If implemented, it allows the driver to reset the EHCI controller without affecting the state of the ports or the relationship to the companion host controllers. For example, the UHPHS_PORTSC registers should not be reset to their default values and the CF bit setting should not go to 0 (retaining port ownership relationships).
A host software read of this bit as 0 indicates the Light Host Controller Reset has completed and it is safe for host software to re-initialize the host controller. A host software read of this bit as 1 indicates the Light Host Controller Reset has not yet completed.
If not implemented, a read of this field will always return a 0.

Bit 6 – IAAD Interrupt on Async Advance Doorbell
This bit is used as a doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule. Software must write a 1 to this bit to ring the doorbell.
When the host controller has evicted all appropriate cached schedule state, it sets the Interrupt on Async Advance status bit in the UHPHS_USBSTS register. If the Interrupt on Async Advance Enable bit in the UHPHS_USBINTR register is set to 1, then the host controller will assert an interrupt at the next interrupt threshold.
The host controller sets this bit to 0 after it has set the Interrupt on Async Advance status bit in the UHPHS_USBSTS register to 1.
Software should not write a 1 to this bit when the asynchronous schedule is disabled. Doing so will yield undefined results.

Bit 5 – ASE Asynchronous Schedule Enable
This bit controls whether the host controller skips processing the Asynchronous Schedule.

Value	Description
0	Do not process the Asynchronous Schedule.
1	Use the UHPHS_ASYNCLISTADDR register to access the Asynchronous Schedule.

Bit 4 – PSE Periodic Schedule Enable
This bit controls whether the host controller skips processing the Periodic Schedule.

Value	Description
0	Do not process the Periodic Schedule.
1	Use the UHPHS_PERIODICLISTBASE register to access the Periodic Schedule.

Bits 3:2 – FLS[1:0] Frame List Size
This field is read-only with one exception: it is read/write if the Programmable Frame List flag, in the UHPHS_HCCPARAMS register, is set to 1. This field specifies the size of the frame list. The size of the frame list controls which bits in the Frame Index Register should be used for the Frame List Current index.

Value	Description
0	1024 elements (4096 bytes).
1	512 elements (2048 bytes).
2	256 elements (1024 bytes), for resource-constrained environments.
3	Reserved.

Bit 1 – HCRESET Host Controller Reset
This control bit is used by software to reset the host controller. The effects of this on Root Hub registers are similar to a Chip Hardware Reset.
When software writes a 1 to this bit, the Host Controller resets its internal pipelines, timers, counters, state machines, etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports.

Downloaded from Arrow.com.

PCI Configuration registers are not affected by this reset. All operational registers, including port registers and port state machines, are set to their initial values. Port ownership reverts to the companion host controller(s) with side effects. Software must reinitialize the host controller in order to return the host controller to an operational state.
This bit is set to 0 by the Host Controller when the reset process is complete. Software cannot terminate the reset process early by writing a 0 to this register.
Software must not set this bit to 1 when HCHLT in the UHPHS_USBSTS register is 0. Attempting to reset an actively running host controller results in undefined behavior.

Bit 0 – RS Run/Stop
The Host Controller must halt within 16 microframes after software clears the bit RS. The HCHLT bit in the status register indicates when the Host Controller has finished its pending pipelined transactions and has entered the stopped state. Software must not write 1 to this field unless the host controller is in the halted state (i.e., HCHLT in the UHPHS_USBSTS register is 1). Doing so yields undefined results.

Value	Description
0	Host Controller completes the current and any actively pipelined transactions on the USB and then halts.
1	Host Controller proceeds with execution of the schedule.

Downloaded from Arrow.com.

41.7.5 UHPHS USB Status Register

Name: UHPHS_USBSTS
Offset: 0x14
Reset: 0x00001000
Property: Read/Write

This register indicates pending interrupts and various states of the Host Controller. The status resulting from a transaction on the serial bus is not indicated in this register. Software sets a bit to 0 in this register by writing a 1 to it.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ASS	PSS	RCM	HCHLT				
Access	R	R	R	R				
Reset	0	0	0	1				

Bit	7	6	5	4	3	2	1	0
			IAA	HSE	FLR	PCD	USBERRINT	USBINT
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 15 – ASS Asynchronous Schedule Status
The bit reports the current real status of the Asynchronous Schedule.
The Host Controller is not required to immediately disable or enable the Asynchronous Schedule when software transitions the Asynchronous Schedule Enable bit in the UHPHS_USBCMD register. When this bit and the Asynchronous Schedule Enable bit are the same value, the Asynchronous Schedule is either enabled or disabled.

Value	Description
0	Asynchronous Schedule is disabled.
1	Asynchronous Schedule is enabled.

Bit 14 – PSS Periodic Schedule Status
The bit reports the current real status of the Periodic Schedule. If this bit is set to 0, then the status of the Periodic Schedule is disabled. If this bit is set to 1, then the status of the Periodic Schedule is enabled. The Host Controller is not required to immediately disable or enable the Periodic Schedule when software transitions the Periodic Schedule Enable bit in the UHPHS_USBCMD register. When this bit and the Periodic Schedule Enable bit are the same value, the Periodic Schedule is either enabled or disabled.

Bit 13 – RCM Reclamation
This is a read-only status bit used to detect any empty asynchronous schedule.

Bit 12 – HCHLT HCHalted
This bit is 0 whenever the Run/Stop bit is 1. The Host Controller sets this bit to 1 after it has stopped executing as a result of the Run/Stop bit being set to 0, either by software or by the Host Controller hardware (e.g. internal error).

Bit 5 – IAA Interrupt on Async Advance (Cleared on write)
System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing 1 to the Interrupt on the Async Advance Doorbell bit in the UHPHS_USBCMD register. This status bit indicates the assertion of that interrupt source.

Downloaded from Arrow.com.

Bit 4 – HSE Host System Error (Cleared on write)
The Host Controller sets this bit to 1 when a serious error occurs during a host system access involving the Host Controller module. In a PCI system, conditions that set this bit to 1 include PCI Parity error, PCI Master Abort, and PCI Target Abort. When this error occurs, the Host Controller clears the Run/Stop bit in the Command register to prevent further execution of the scheduled TDs.

Bit 3 – FLR Frame List Rollover (Cleared on write)
The Host Controller sets this bit to 1 when the Frame List Index (see UHPHS USB Frame Index Register) rolls over from its maximum value to 0. The exact value at which the rollover occurs depends on the frame list size. For example, if the frame list size (as programmed in the Frame List Size field of the UHPHS_USBCMD register) is 1024, the Frame Index Register rolls over every time FRINDEX[13] toggles. Similarly, if the size is 512, the Host Controller sets this bit to 1 every time FRINDEX[12] toggles.

Bit 2 – PCD Port Change Detect (Cleared on write)
The Host Controller sets this bit to 1 when any port for which the Port Owner bit is set to 0 (see UHPHS Port Status and Control Register) has a change bit transition from 0 to 1 or a Force Port Resume bit transition from 0 to 1 as a result of a J-K transition detected on a suspended port. This bit will also be set as a result of the Connect Status Change being set to 1 after system software has relinquished ownership of a connected port by writing 1 to a port's Port Owner bit.
This bit is allowed to be maintained in the Auxiliary power well. Alternatively, it is also acceptable that on a D3 to D0 transition of the EHCI HC device, this bit is loaded with the OR of all of the PORTSC change bits (including: Force Port Resume, Overcurrent Change, Enable/Disable Change and Connect Status Change).

Bit 1 – USBERRINT USB Error Interrupt (Cleared on write)
The Host Controller sets this bit to 1 when completion of a USB transaction results in an error condition (e.g., error counter underflow). If the TD on which the error interrupt occurred also had its IOC bit set, both this bit and USBINT bit are set.

Bit 0 – USBINT USB Interrupt (Cleared on write)
The Host Controller sets this bit to 1 on the completion of a USB transaction, which results in the retirement of a Transfer Descriptor that had its IOC bit set.
The Host Controller also sets this bit to 1 when a short packet is detected (the actual number of bytes received was less than the expected number of bytes).

Downloaded from Arrow.com.

41.7.6 UHPHS USB Interrupt Enable Register

Name: UHPHS_USBINTR
Offset: 0x18
Reset: 0x00000000
Property: Read/Write

This register enables and disables reporting of the corresponding interrupt to the software. When a bit is set and the corresponding interrupt is active, an interrupt is generated to the host. Interrupt sources that are disabled in this register still appear in the UHPHS_USBSTS to allow the software to poll for events.

For all bits, 1=Enabled, 0=Disabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			IAAE	HSEE	FLRE	PCIE	USBEIE	USBIE
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – IAAE Interrupt on Async Advance Enable
The interrupt is acknowledged by software clearing the Interrupt on Async Advance bit UHPHS_USBSTS.

Bit 4 – HSEE Host System Error Enable
The interrupt is acknowledged by software clearing the Host System Error bit in UHPHS_USBSTS.

Bit 3 – FLRE Frame List Rollover Enable
The interrupt is acknowledged by software clearing the Frame List Rollover in UHPHS_USBSTS.

Bit 2 – PCIE Port Change Interrupt Enable
The interrupt is acknowledged by software clearing the Port Change Detect bit in UHPHS_USBSTS.

Bit 1 – USBEIE USB Error Interrupt Enable
The interrupt is acknowledged by software clearing the USBERRINT in UHPHS_USBSTS.

Bit 0 – USBIE USB Interrupt Enable
The interrupt is acknowledged by software clearing the USBINT in UHPHS_USBSTS.

Downloaded from Arrow.com.

41.7.7 UHPHS USB Frame Index Register

Name: UHPHS_FRINDEX
Offset: 0x1C
Reset: 0x00000000
Property: Read/Write

This register is used by the host controller to index into the periodic frame list. The register updates every 125 µs (once each microframe). Bits [N:3] are used to select a particular entry in the Periodic Frame List during periodic schedule execution. The number of bits used for the index depends on the size of the frame list as set by system software in the Frame List Size field in the UHPHS_USBCMD register).

This register must be written as a DWord. Byte writes produce undefined results. This register cannot be written unless the Host Controller is in the Halted state as indicated by the HCHLT bit (UHPHS_USBSTS register). A write to this register while the Run/Stop bit is set to 1 (UHPHS_USBCMD register) produces undefined results. Writes to this register also affect the SOF value.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			FI[13:8]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FI[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 13:0 – FI[13:0] Frame Index
The value in this register increments at the end of each time frame (e.g., microframe). Bits [N:3] are used for the Frame List current index. This means that each location of the frame list is accessed eight times (frames or microframes) before moving to the next index. The following illustrates values of N based on the value of FLS (Frame List Size) in the UHPHS_USBCMD register.

UHPHS_USBCMD.FLS	Number Elements	N
0	1024	12
1	512	11
2	256	10
3	Reserved	–

The SOF frame number value for the bus SOF token is derived or alternatively managed from this register. The value of FRINDEX must be 125 µs (1 microframe) ahead of the SOF token value. The SOF value may be implemented as an 11-bit shadow register. For this discussion, this shadow register is 11 bits and is named SOFV. SOFV updates every eight microframes (1 millisecond). An example implementation to achieve this behavior is to increment SOFV each time the FRINDEX[2:0] increments from 0 to 1.
Software must use the value of FRINDEX to derive the current microframe number, both for high-speed isochronous scheduling purposes and to provide the “get microframe number” function required for client drivers. Therefore, the

Downloaded from Arrow.com.

value of FRINDEX and the value of SOFV must be kept consistent if chip is reset or software writes to FRINDEX. Writes to FRINDEX must also write-through FRINDEX[13:3] to SOFV[10:0]. In order to keep the update as simple as possible, software should never write a FRINDEX value where the three least significant bits are 7 or 0.

Downloaded from Arrow.com.

41.7.8 UHPHS Periodic Frame List Base Address Register

Name: UHPHS_PERIODICLISTBASE
Offset: 0x24
Reset: 0x00000000
Property: Read/Write

This 32-bit register contains the beginning address of the Periodic Frame List in the system memory. If the host controller is in 64-bit mode (as indicated by a 1 in the 64-bit Addressing Capability field in the UHPHS_HCCSPARAMS register), then the most significant 32 bits of every control data structure address comes from the UHPHS_CTRLDSSEGMENT register. System software loads this register prior to starting the schedule execution by the Host Controller. The memory structure referenced by this physical memory pointer is assumed to be 4-Kbyte aligned. The contents of this register are combined with the Frame Index Register (UHPHS_FRINDEX) to enable the Host Controller to step through the Periodic Frame List in sequence. This register must be written as a DWord. Byte writes produce undefined results.

Bit	31	30	29	28	27	26	25	24
	BA[19:12]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BA[11:4]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BA[3:0]							
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 31:12 – BA[19:0] Base Address (Low)
These bits correspond to memory address signals [31:12], respectively.

Downloaded from Arrow.com.

41.7.9 UHPHS Asynchronous List Address Register

Name: UHPHS_ASYNCLISTADDR
Offset: 0x28
Reset: 0x00000000
Property: Read/Write

This 32-bit register contains the address of the next asynchronous queue head to be executed. If the host controller is in 64-bit mode (as indicated by a 1 in the 64-bit Addressing Capability field in the UHPHS_HCCPARAMS register), then the most significant 32 bits of every control data structure address comes from the UHPHS_CTRLDSSEGMENT register. Bits [4:0] of this register cannot be modified by system software and will always return a zero when read. The memory structure referenced by this physical memory pointer is assumed to be 32-byte (cache line) aligned. This register must be written as a DWord. Byte writes produce undefined results.

Bit	31	30	29	28	27	26	25	24
	LPL[26:19]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LPL[18:11]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LPL[10:3]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LPL[2:0]							
Access	R/W	R/W	R/W					
Reset	0	0	0					

Bits 31:5 – LPL[26:0] Link Pointer Low
These bits correspond to memory address signals [31:5], respectively. This field may only reference a Queue Head (QH).

Downloaded from Arrow.com.

41.7.10 UHPHS Configure Flag Register

Name: UHPHS_CONFIGFLAG
Offset: 0x50
Reset: 0x00000000
Property: Read/Write

This register is in the auxiliary power well. It is only reset by hardware when the auxiliary power is initially applied or in response to a host controller reset.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								CF
Access								R/W
Reset								0

Bit 0 – CF Configure Flag
Host software sets this bit as the last action in its process of configuring the Host Controller. This bit controls the default port-routing control logic. Bit values and side-effects are listed below.

Value	Description
0	Port routing control logic default-routes each port to an implementation-dependent classic host controller (default value).
1	Port routing control logic default-routes all ports to this host controller.

Downloaded from Arrow.com.

41.7.11 UHPHS Port Status and Control Register

Name: UHPHS_PORTSCx
Offset: 0x54 + x*0x04 [x=0..2]
Reset: 0x00002000
Property: Read/Write

The number of port registers is documented in the UHPHS_HCSPARAMS register. Software uses this information as an input parameter to determine how many ports need to be serviced. All ports have the structure defined below.

This register is in the auxiliary power well. It is only reset by hardware when the auxiliary power is initially applied or in response to a host controller reset. The initial conditions of a port are:

- No device connected
- Port disabled

If the port has port power control, software cannot change the state of the port until after it applies power to the port by setting port power to a 1. Software must not attempt to change the state of the port until after power is stable on the port. The host is required to have power stable to the port within 20 milliseconds of the 0 to 1 transition.

Note:

1. When a device is attached, the port state transitions to the connected state and system software will process this as with any status change notification.
2. If a port is being used as the Debug Port, then the port may report device connected and enabled when the Configured Flag is set to 0.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		WKOC_E	WKDSCNNT_E	WKCNNT_E	PTC[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PIC[1:0]		PO	PP	LS[1:0]			PR
Access	R/W	R/W	R/W	R-R/W	R	R		R/W
Reset	0	0	1	0	0	0		0

Bit	7	6	5	4	3	2	1	0
	SUS	FPR	OCC	OCA	PEDC	PED	CSC	CCS
Access	R/W	R/W	R/W	R	R/W	R/W	R/W	R
Reset	0	0	0	0	0	0	0	0

Bit 22 – WKOC_E Wake on Overcurrent Enable
This field is 0 if Port Power is 0.

Value	Description
0	Disables the port to be sensitive to overcurrent conditions as wake-up events.
1	Enables the port to be sensitive to overcurrent conditions as wake-up events.

Bit 21 – WKDSCNNT_E Wake on Disconnect Enable
This field is 0 if Port Power is 0.

Value	Description
0	Disables the port to be sensitive to device disconnects as wake-up events.
1	Enables the port to be sensitive to device disconnects as wake-up events.

Downloaded from Arrow.com.

Bit 20 – WKCNNT_E Wake on Connect Enable
This field is 0 if Port Power is 0.

Value	Description
0	Disables the port to be sensitive to device connects as wake-up events.
1	Enables the port to be sensitive to device connects as wake-up events.

Bits 19:16 – PTC[3:0] Port Test Control
When this field is set to 0, the port is NOT operating in a test mode. A non-zero value indicates that it is operating in test mode and the specific test mode is indicated by the specific value.
Test mode bits are encoded as follows (6 to 15 are reserved):

Value	Test Mode
0	Test mode not enabled
1	Test J_STATE
2	Test K_STATE
3	Test SE0_NAK
4	Test Packet
5	Test FORCE_ENABLE

Refer to the USB Specification Revision 2.0, Chapter 7, for details on each test mode.

Bits 15:14 – PIC[1:0] Port Indicator Control
Writing to these bits has no effect if the P_INDICATOR bit in the UHPHS_HCSPARAMS register is set to 0. If the P_INDICATOR bit is set to 1, then the bits are encoded as follows:

Value	Meaning
0	Port indicators are off
1	Amber
2	Green
3	Undefined

Refer to the USB Specification Revision 2.0 for a description of how these bits are to be used.
This field is 0 if Port Power is 0.

Bit 13 – PO Port Owner
System software uses this field to release ownership of the port to a selected host controller (in the event that the attached device is not a high-speed device). Software writes 1 to this bit when the attached device is not a high-speed device. A 1 in this bit means that a companion host controller owns and controls the port.

Value	Description
0	This bit unconditionally goes to a 0 when the bit UHPHS_CONFIGFLAG.CF makes a 0 to 1 transition.
1	This bit unconditionally goes to 1 whenever the bit UHPHS_CONFIGFLAG.CF=0.

Bit 12 – PP Port Power
The function of this bit depends on the value of the Port Power Control (PPC) field in the UHPHS_HCSPARAMS register. When host controller has port power control switches (PPC=0), PP is in read-only mode:

Value	Description
1	Each port is hard-wired to power.

When host controller has port power control switches (PPC=1), PP is in read/write mode:

Downloaded from Arrow.com.

Value	Description
0	Host port power switch is OFF. When power is not available on a port (i.e., PP at 0), the port is non-functional and does not report attaches, detaches, etc.
1	Host port power switch is ON. When power is not available on a port (i.e., PP at 0), the port is non-functional and does not report attaches, detaches, etc.

When an overcurrent condition is detected on a powered port and PPC is set to 1, the PP bit in each affected port may be transitioned by the host controller from 1 to 0 (removing power from the port).

Bits 11:10 – LS[1:0] Line Status
These bits reflect the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines. These bits are used for detection of low-speed USB devices prior to the port reset and enable sequence. This field is valid only when the port enable bit is 0 and the current connect status bit is set to 1.
This value of this field is undefined if Port Power is 0.

Value	Name	Description
0	SE0	Not a low-speed device, perform EHCI reset
1	K-STATE	Low-speed device, release ownership of port
2	J-STATE	Not a low-speed device, perform EHCI reset
3	Undefined	Not a low-speed device, perform EHCI reset

Bit 8 – PR Port Reset
When software writes a 1 to this bit (from 0), the bus reset sequence as defined in the USB Specification Revision 2.0 is started. Software writes a 0 to this bit to terminate the bus reset sequence. Software must keep this bit set to 1 long enough to ensure the reset sequence, as specified in the USB Specification Revision 2.0, completes.
Note: When software writes this bit to 1, it must also write 0 to the Port Enable bit.

When software writes a 0 to this bit, there may be a delay before the bit status changes to 0. The bit status will not read as 0 until after the reset has completed. If the port is in High-Speed mode after reset is complete, the host controller will automatically enable this port (e.g., set the Port Enable bit to 1). A host controller must terminate the reset and stabilize the state of the port within 2 milliseconds of software transitioning this bit from 1 to 0. For example: if the port detects that the attached device is high-speed during reset, then the host controller must have the port in the enabled state within 2 ms of software writing this bit to 0.
The HCHLT bit in the UHPHS_USBSTS register should be set to 0 before software attempts to use this bit. The host controller may hold Port Reset asserted to 1 when the HCHLT bit is 1.
This field is 0 if Port Power is 0.

Value	Description
0	Port is not in Reset.
1	Port is in Reset.

Bit 7 – SUS Suspend

Value	Description
0	Port not in suspend state.
1	Port in suspend state.

Downloaded from Arrow.com.

Note:
Port Enabled Bit and Suspend bit of this register define the port states as follows:

Bits [Port Enabled, Suspend]	Port State
0X	Disable
10	Enable
11	Suspend

When in suspend state, downstream propagation of data is blocked on this port, except for port reset. The blocking occurs at the end of the current transaction, if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB.

A write of 0 to this bit is ignored by the host controller. The host controller will unconditionally set this bit to 0 when:

- Software sets the Force Port Resume bit to 0 (from 1).
- Software sets the Port Reset bit to 1 (from 0).

If host software sets this bit to 1 when the port is not enabled (i.e., Port Enabled bit set to 0), the results are undefined.

This field is 0 if Port Power is set to 0.

Bit 6 – FPR Force Port Resume
This functionality defined for manipulating this bit depends on the value of the Suspend bit. For example, if the port is not suspended (Suspend and Enabled bits are set to 1) and software transitions this bit to 1, then the effects on the bus are undefined.
Software sets this bit to a 1 to drive resume signaling. The Host Controller sets this bit to 1 if a J-to-K transition is detected while the port is in the Suspend state. When this bit transitions to 1 because a J-to-K transition is detected, the Port Change Detect bit in the UHPHS_USBSTS register is also set to 1. If software sets this bit to 1, the host controller must not set the Port Change Detect bit.
Note that when the EHCI controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification Revision 2.0. The resume signaling (Full-speed 'K') is driven on the port as long as this bit remains set to 1. Software must appropriately time the Resume and set this bit to 0 when the appropriate amount of time has elapsed. Writing a 0 (from 1) causes the port to return to High-Speed mode (forcing the bus below the port into a high-speed idle). This bit will remain set to 1 until the port has switched to the high-speed idle. The host controller must complete this transition within 2 milliseconds of software setting this bit to 0.
This field is 0 if Port Power is 0.

Value	Description
0	No resume (K-state) detected/driven on port.
1	Resume detected/driven on port.

Bit 5 – OCC Overcurrent Change (Cleared on write)
Software clears this bit by writing 1.

Value	Description
0	No change to Overcurrent Active.
1	Changes to Overcurrent Active.

Bit 4 – OCA Overcurrent Active
This bit will automatically transition from 1 to 0 when the overcurrent condition is removed.

Value	Description
0	This port does not have an overcurrent condition.
1	This port currently has an overcurrent condition.

Downloaded from Arrow.com.

Bit 3 – PEDC Port Enable/Disable Change (Cleared on write)
For the root hub, this bit gets set to 1 only when a port is disabled due to the appropriate conditions existing at the EOF2 point (refer to Chapter 11 of the USB Specification for the definition of a Port Error). Software clears this bit by writing a 1 to it.
This field is 0 if Port Power bit is 0.

Value	Description
0	No change in port enabled/disabled status.
1	Port enabled/disabled status has changed.

Bit 2 – PED Port Enabled/Disabled
Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a 1 to this field. The host controller will only set this bit to 1 when the reset sequence determines that the attached device is a high-speed device.
Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by host software. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events.
When the port is disabled (0b), downstream propagation of data is blocked on this port, except for reset.
This field is 0 if Port Power bit is 0.

Value	Description
0	Disable.
1	Enable.

Bit 1 – CSC Connect Status Change (Cleared on write)
Indicates a change has occurred in the port's Current Connect Status. The host controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be "setting" an already-set bit (i.e., the bit remains set). Software sets this bit to 0 by writing a 1 to it.
This field is 0 if Port Power bit is 0.

Value	Description
0	No change.
1	Change in Current Connect Status.

Bit 0 – CCS Current Connect Status
This value reflects the current state of the port, and may not correspond directly to the event that caused the CSC bit to be 1.
This bit is 0 if Port Power is 0.

Value	Description
0	No device is present.
1	Device is present on port.

Downloaded from Arrow.com.

41.7.12 EHCI: REG06 - AHB Error Status

Name: UHPHS_INSNREG06
Offset: 0xA8
Reset: 0x00000000
Property: Read/Write

Control and Status Register, used to read the UTMI registers from the signals below.

Bit	31	30	29	28	27	26	25	24
	AHB_ERR							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					HBURST[2:0]			Nb_Burst[4]
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	Nb_Burst[3:0]				Nb_Success_Burst[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 31 – AHB_ERR AHB Error
System bus error was encountered and erroneous burst characteristics are captured. To clear this field the application must write a 0.
EHCI:

- When no error, 0 is written to INSNREG06[8:4].
- When INCR4 and an error occurs, 4 is written to INSNREG06[8:4].
- When INCR8 and an error occurs, 8 is written to INSNREG06[8:4].
- When INCR16 and an error occurs, 16 is written to INSNREG06[8:4].
- Other values except 4, 8, and 16 are not written to INSNREG06[8:4].

OHCI:

- When no error, 0 is written to INSNREG06[8:4].
- When INCR4 and error occurs, 4 is written to INSNREG06[8:4].
- Other values except 4 are not written to INSNREG06[8:4].

Bits 11:9 – HBURST[2:0] Burst Value
Value of the control phase at which the AHB error occurred.
This field applies to enabled incremental bursts only.

Bits 8:4 – Nb_Burst[4:0] Number of Bursts
Number of beats expected in the burst at which the AHB error occurred. Valid values are 0 to 16.
This field applies to enabled incremental bursts only.

Bits 3:0 – Nb_Success_Burst[3:0] Number of Successful Bursts
Number of successfully completed beats in the current burst before the AHB error occurred.
This field applies to enabled incremental bursts only.

Downloaded from Arrow.com.

41.7.13 EHCI: REG07 - AHB Master Error Address

Name: UHPHS_INSNREG07
Offset: 0xAC
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	AHB_ADDR[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	AHB_ADDR[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	AHB_ADDR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	AHB_ADDR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – AHB_ADDR[31:0] AHB Address
System bus address of the control phase at which the system bus error occurred.

Downloaded from Arrow.com.

42. Audio Class D Amplifier (CLASSD)

42.1 Description

The Audio Class D Amplifier (CLASSD) is a digital input, Pulse Width Modulated (PWM) output stereo Class D amplifier. It features a high-quality interpolation filter embedding a digitally-controlled gain, an equalizer and a de-emphasis filter.

On its input side, the CLASSD is compatible with most common audio data rates. On the output side, its PWM output can drive either:

- high-impedance single-ended or differential output loads (Audio DAC application) or,
- external MOSFETs through an integrated non-overlapping circuit (Class D power amplifier application).

Note:

- CLASSD is stereo but depending on the available I/O at the product level, only one (right or left) channel may be accessed. Refer to the product pin description table.

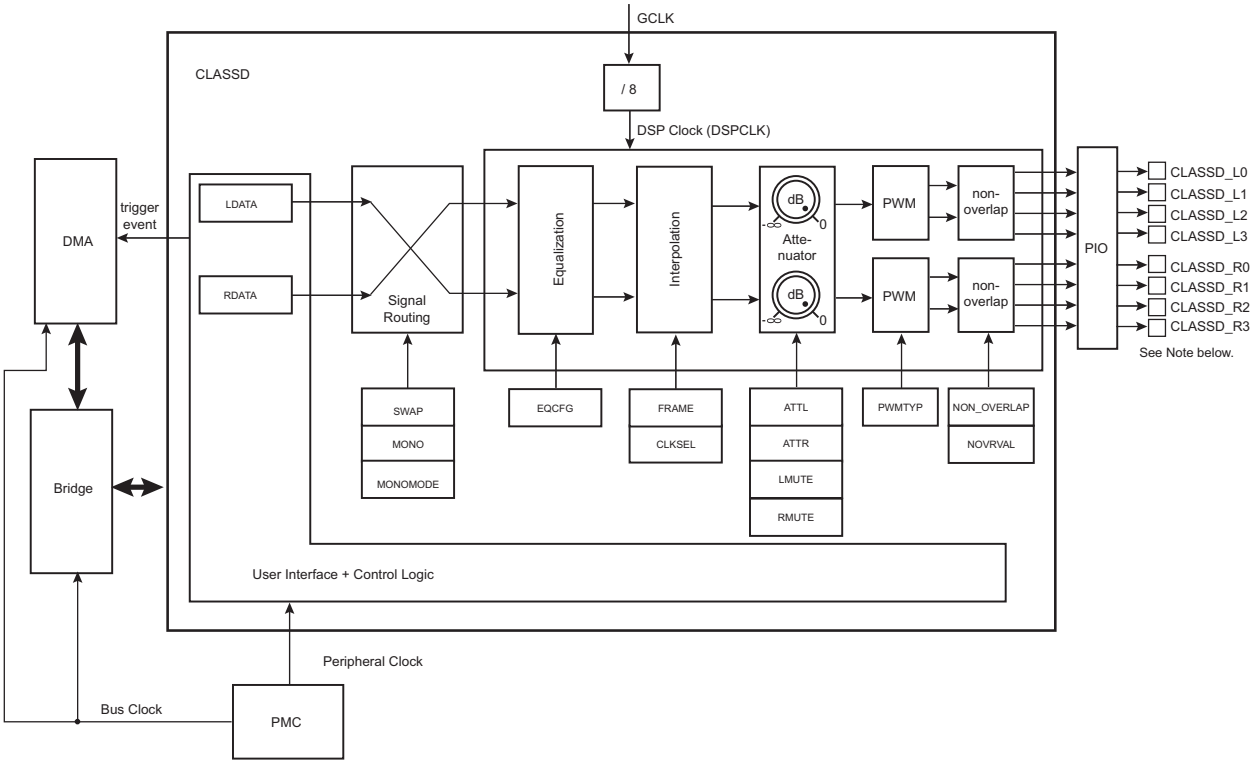
42.2 Embedded Characteristics

- PWM Class D Amplifier
- 16-bit Audio Data
- DSP Clocks: 12.288 and 11.2896 MHz
- Input Sampling Rates: 8, 16, 32, 48, 96, 22.05, 44.1, 88.2 kHz
- 3-band Equalizer
- De-emphasis Filter
- Digital Volume Control
- Differential or Single-ended Outputs
- Non-overlapping Circuit to Control External MOSFETs
- Supports DMA

Downloaded from Arrow.com.

42.3 Block Diagram

Figure 42-1. CLASSD Block Diagram



Note:
CLASSD is stereo but depending on the available I/O at the product level, only one (right or left) channel may be accessed.
Refer to the product pin description table.

42.4 Pin Name List

Table 42-1. Output Pins Assignment Versus Application Use Cases

Pin	External MOS Driver (NON_OVERLAP = 1)		Direct Load (NON_OVERLAP = 0)		Type
	Full H-Bridge (PWMTYP = 1)	Half H-Bridge (PWMTYP = 0)	Differential Load (PWMTYP = 1)	Single-Ended Load (PWMTYP = 0)	
	Use Case 1	Use Case 2	Use Cases 3A & 3B	Use Cases 4A & 4B	
CLASSD_L0	gate_pmos_leftp	gate_pmos_left	leftp	left	Output
CLASSD_L1	gate_nmos_leftp	gate_nmos_left	Not used (fixed to 0)	Not used (fixed to 0)	Output
CLASSD_L2	gate_pmos_leftn	Not used (fixed to 1)	leftn	Not used (fixed to 0)	Output
CLASSD_L3	gate_nmos_leftn	Not used (fixed to 1)	Not used (fixed to 0)	Not used (fixed to 0)	Output
CLASSD_R0	gate_pmos_rightp	gate_pmos_right	rightp	right	Output
CLASSD_R1	gate_nmos_rightp	gate_nmos_right	Not used (fixed to 0)	Not used (fixed to 0)	Output
CLASSD_R2	gate_pmos_rightn	Not used (fixed to 1)	rightn	Not used (fixed to 0)	Output
CLASSD_R3	gate_nmos_rightn	Not used (fixed to 1)	Not used (fixed to 0)	Not used (fixed to 0)	Output

Downloaded from Arrow.com.

42.5 Product Dependencies

42.5.1 I/O Lines

The pins used for interfacing the compliant external devices may be multiplexed with PIO lines. The programmer must first program the PIO controllers to assign the CLASSD pins to their peripheral functions.

42.5.2 Power Management

The CLASSD is clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the CLASSD Peripheral Clock and provide a generic clock (GCLK).

The fields NOVRVAL, NON_OVERLAP and PWMTYP in CLASSD_MR, and DSPCLKFREQ and FREQ in CLASSD_INTPMR, must be configured prior to applying the GCLK.

42.5.3 Interrupt

The CLASSD has an interrupt line connected to the interrupt controller. Handling the CLASSD interrupt requires programming the interrupt controller before configuring the CLASSD.

42.6 Functional Description

42.6.1 Interpolator

42.6.1.1 Clock Configuration

The interpolator accepts input sampling frequencies (f_s) and the input DSP clock (DSPCLK) that can be configured in the CLASSD Interpolator Mode Register. GCLK must be configured in the PMC according to the desired DSPCLK so that DSPCLK = GCLK / 8.

The following table provides authorized DSPCLK / f_s ratios and associated filter types.

Table 42-2. Authorized DSPCLK / f_s Ratios & Filter Types

f_s	DSPCLK	
	12.288 MHz	11.2896 MHz
8 kHz	2	–
16 kHz	2	–
32 kHz	2	–
48 kHz	1	–
96 kHz	3	–
22.05 kHz	–	1
44.1 kHz	–	1
88.2 kHz	–	3

Note: Each dash (–) indicates a configuration that is not authorized and that raises the CFGERR flag in CLASSD_INTSR.

42.6.1.2 CLASSD Frequency Response

Interpolation is performed with a combination of Infinite Impulse Response (IIR) and Cascaded Integrator-Comb (CIC) filters. Given the input configuration, the coefficients of the filters are redefined to optimize their transfer function to optimize the audio bandwidth. The different types of filters are defined in section Clock Configuration.

Downloaded from Arrow.com.

Figure 42-2. Type 1 Frequency Response

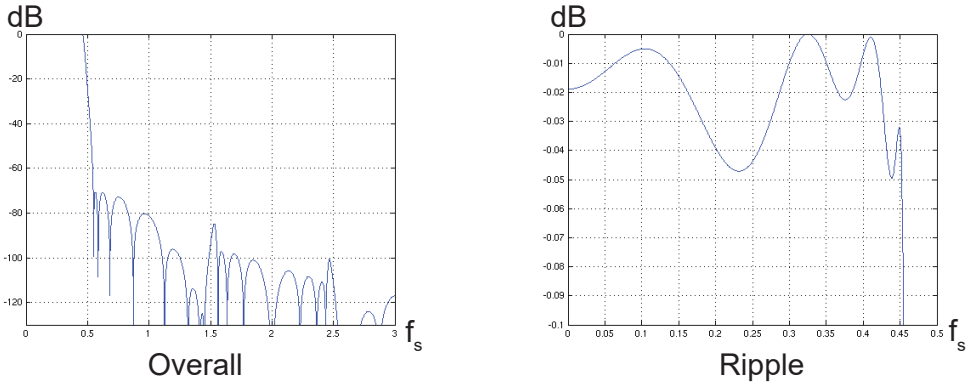


Figure 42-3. Type 2 Frequency Response

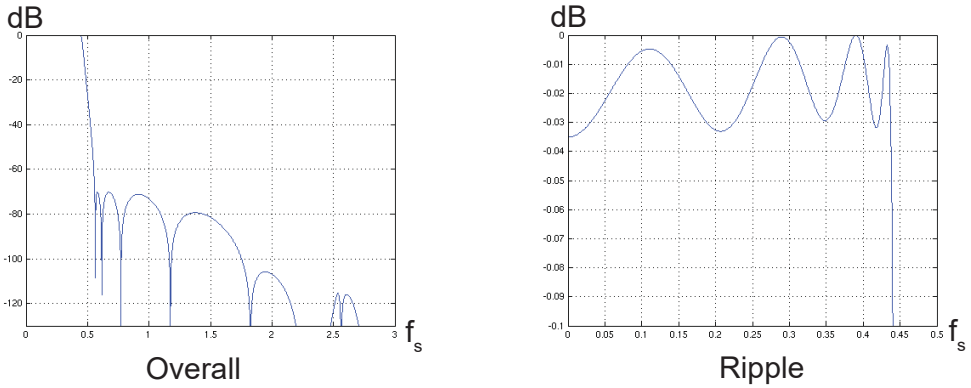
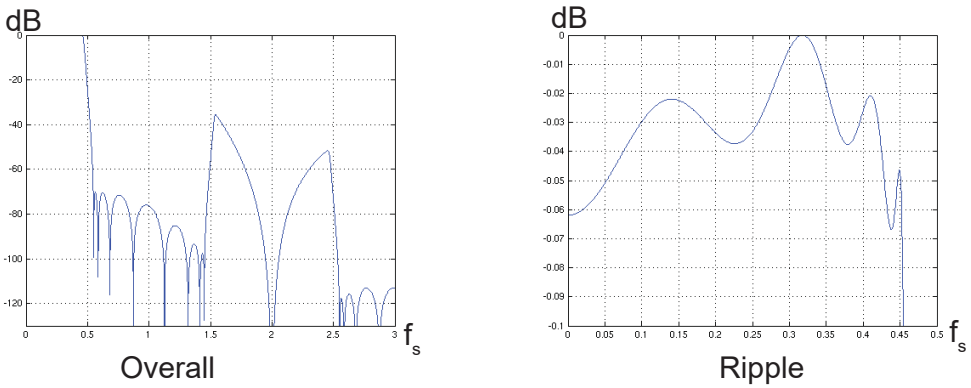


Figure 42-4. Type 3 Frequency Response



42.6.2 Equalizer

The CLASSD offers 12 pre-programmed equalization filters.

A zero-cross detection system is used to modify the equalizer on-the-fly with minimum disturbance on the output signal.

Programming of the equalization filter is detailed in section CLASSD Interpolator Mode Register.

The following figures show the frequency response of the equalizer function implemented in the D/A channels.

Downloaded from Arrow.com.

Figure 42-5. Bass Filters Response

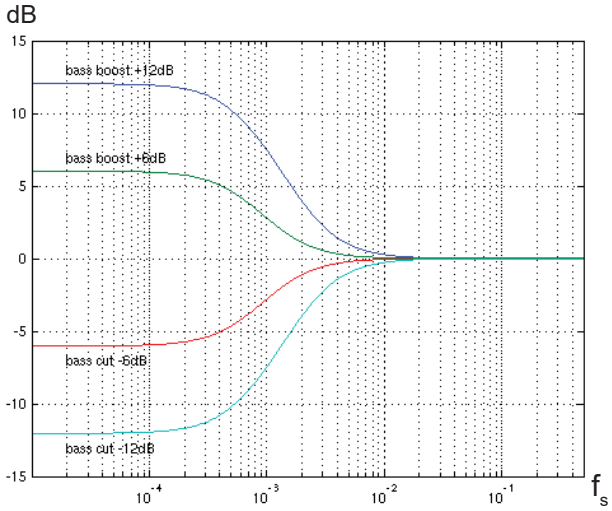
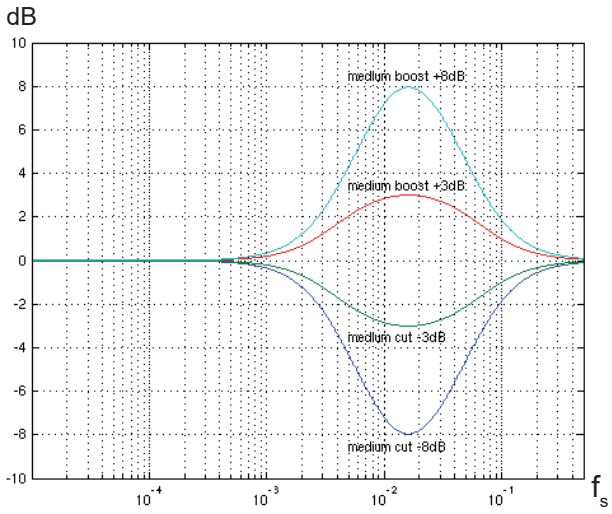
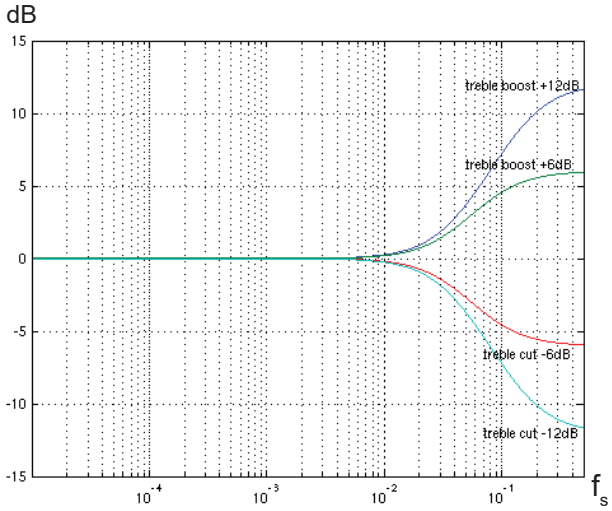


Figure 42-6. Medium Filters Response



Downloaded from Arrow.com.

Figure 42-7. Treble Filters Response



42.6.3 De-emphasis Filter Frequency Response

The CLASSD includes a de-emphasis filter which can be enabled for 32, 44.1 or 48 kHz sampling frequencies.

The response and the error generated by the digital approximation of the filter are illustrated in the following figures.

Figure 42-8. De-emphasis Filter: Frequency Response & Error (f_s = 32 kHz)

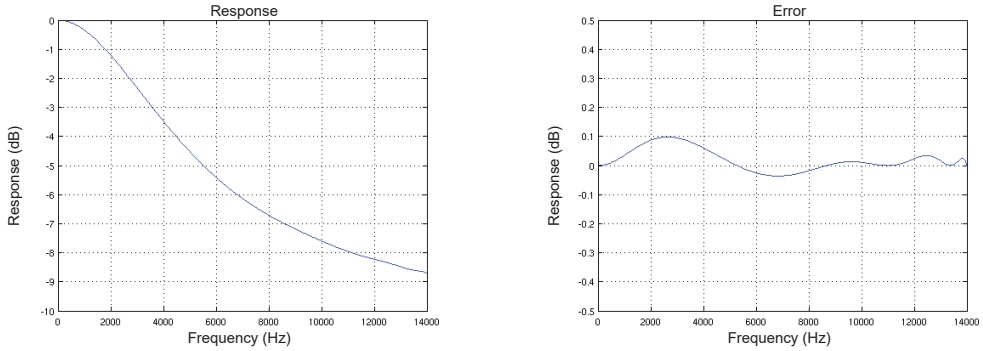
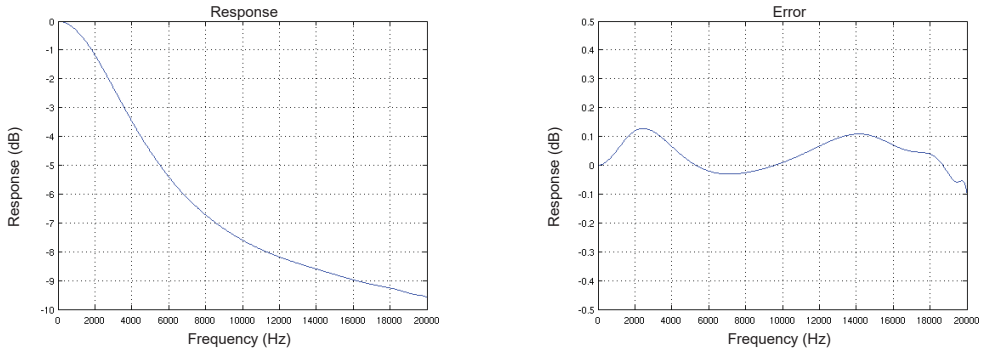
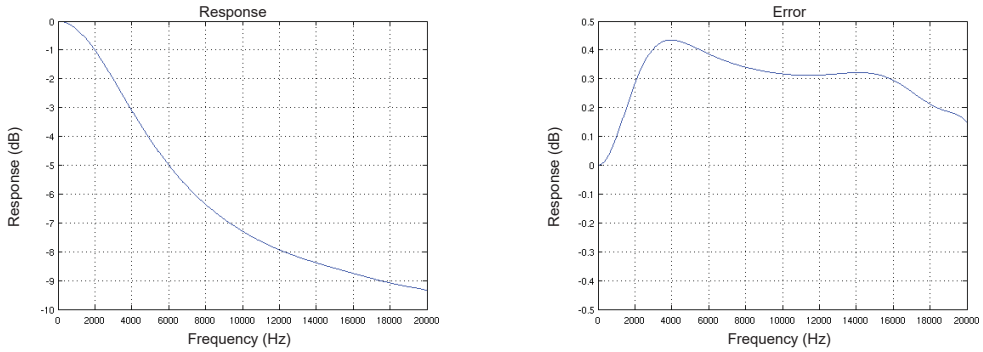


Figure 42-9. De-emphasis Filter: Frequency Response & Error (f_s = 44.1 kHz)



Downloaded from Arrow.com.

Figure 42-10. De-emphasis Filter: Frequency Response & Error (f_s = 48 kHz)



42.6.4 Attenuator and Recommended Input Levels

The CLASSD features a digital attenuator with an attenuation range of 0–77 dB and a step size of 1 dB. When attenuation greater than 77 dB is programmed, the attenuator mutes the channel.

To avoid saturations in the PWM stage, it is recommended to avoid input levels greater than 1 dB below the digital full scale (-1 dBFS). This can done by programming a minimum attenuation of 1 dB.

42.6.5 Pulse Width Modulator (PWM)

The CLASSD Pulse Width Modulator generates fixed frequency pulse width modulated output signals. For the 44.1 kS/s and 48 kS/s standard audio sample rates, the PWM output frequency is set to 16 × f_s: 705.6 kHz and 768 kHz respectively. For 8, 16, 24 and 96 kS/s, the 16× (interpolation) ratio is adapted to keep the output frequency at 768 kHz. In the same way, the output frequency is 705.6 kHz for the 22.05 and 88.2 kS/s cases.

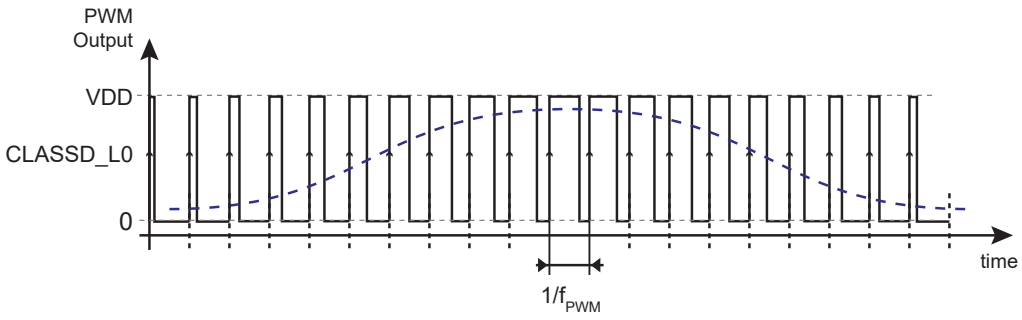
The CLASSD functions either as a DAC loaded by a medium-to-high resistive load (e.g., 1 kΩ to 100 kΩ) or as a Class D power amplifier controller driving an external power stage. Depending on the value of CLASSD_MR.NON_OVERLAP, the CLASSD drives:

- Single-ended or differential resistive loads (NON_OVERLAP = 0)
- Full or Half MOSFET H-bridges (NON_OVERLAP = 1)

When driving an external power stage (NON_OVERLAP = 1), the CLASSD generates the signals to control complementary MOSFET pairs (PMOS and NMOS) with a non-overlapping delay between the NMOS and PMOS controls to avoid short circuit current. The non-overlapping delay can be adjusted in the CLASSD_MR.NOVRVAL field.

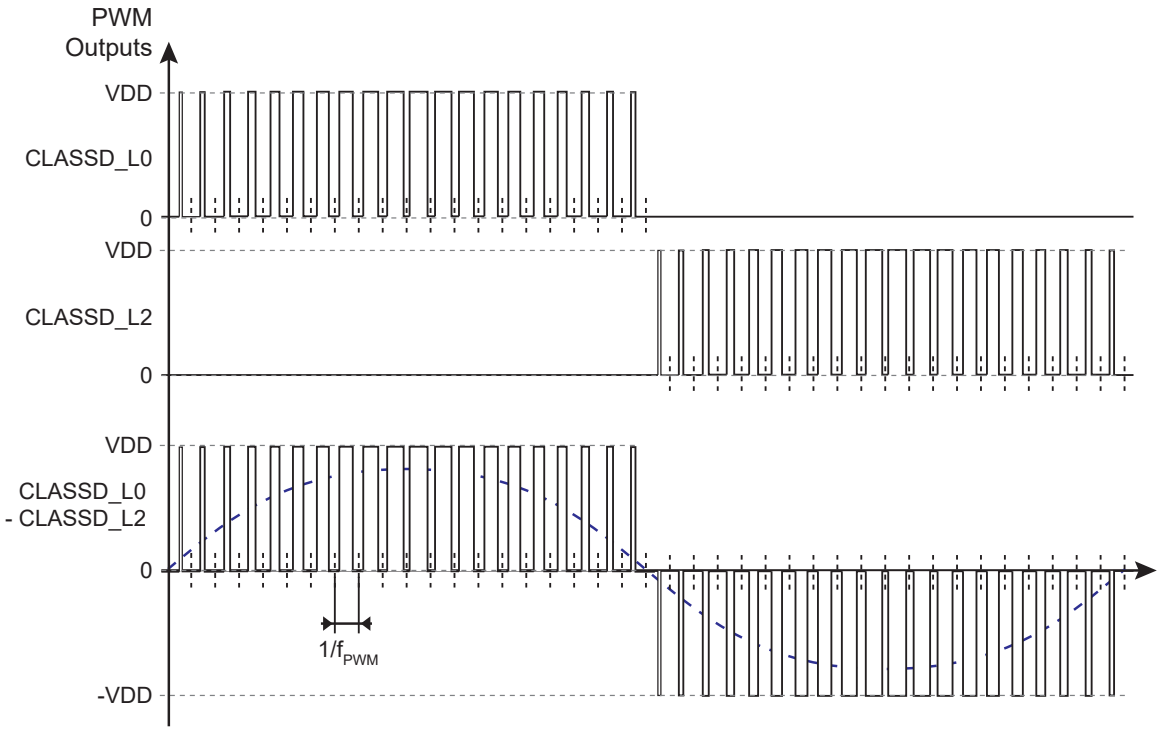
The CLASSD can have a single-ended or a differential output. A specific pulse width modulation type is associated to each case. For single-ended output (CLASSD_MR.PWMTYP = 0), the PWM acts only on the falling edge of the PWM waveform (trailing edge PWM). For differential output (CLASSD_MR.PWMTYP = 1), both the rising and the falling edges of the PWM waveform are modulated (symmetric PWM). Modulation principles are illustrated in the following figures for both types of PWM. In particular, when describing a null input, if PWMTYP = 0 (trailing edge PWM), the output waveform is a square wave with 50% duty cycle. With the same input and PWMTYP = 1, the differential output waveform is zero. This difference removes the classical L-C low-pass filter when PWMTYP = 1.

Figure 42-11. Output Waveform Modulation Principle for PWMTYP = 0



Downloaded from Arrow.com.

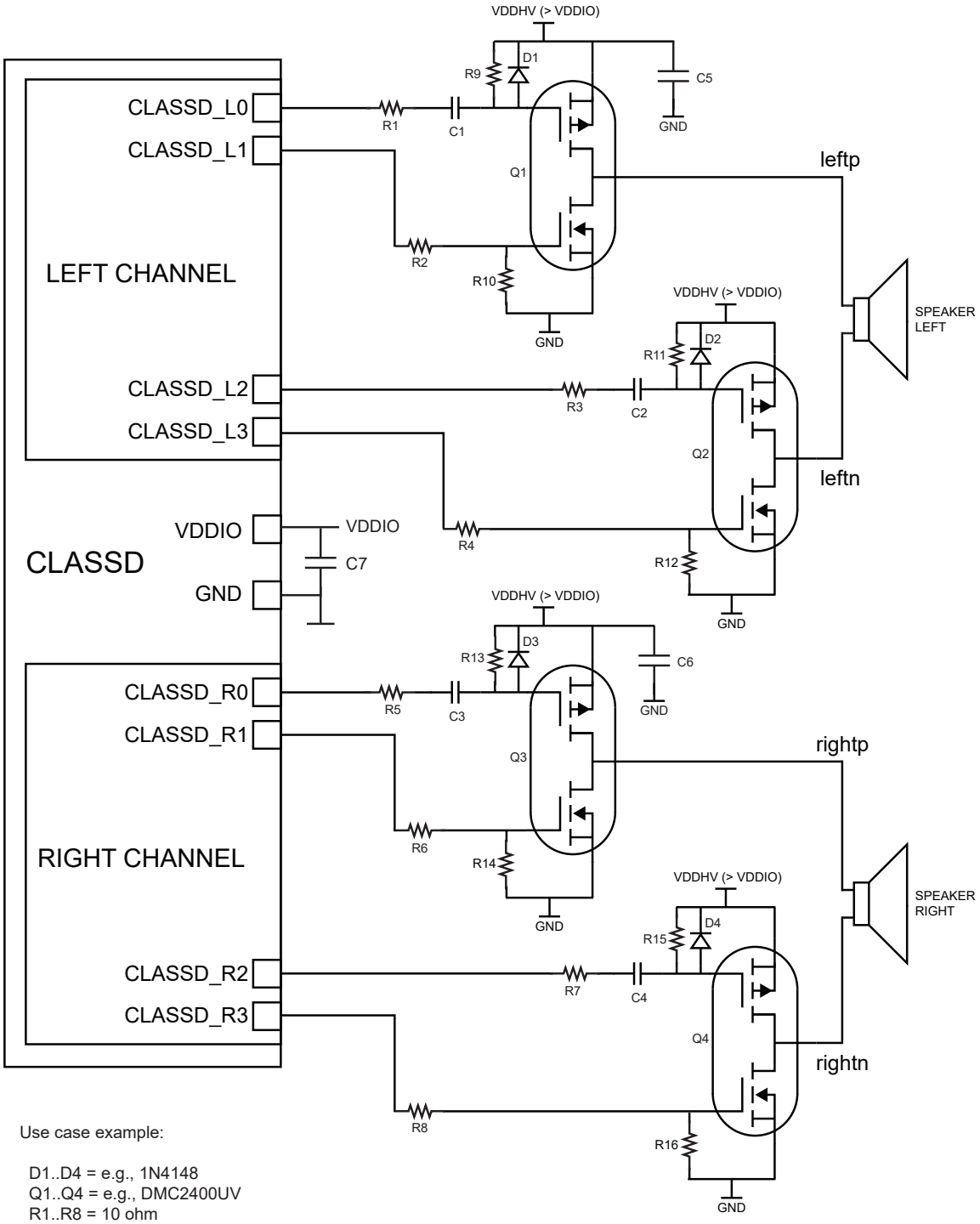
Figure 42-12. Output Waveform Modulation Principle for PWMTYP = 1 (Only Left Channel Pins Shown)



Downloaded from Arrow.com.

42.6.6 Application Schematics For Use Case Examples

Figure 42-13. Use Case 1: Stereo Class D Amplifier With External Differential Power Stage



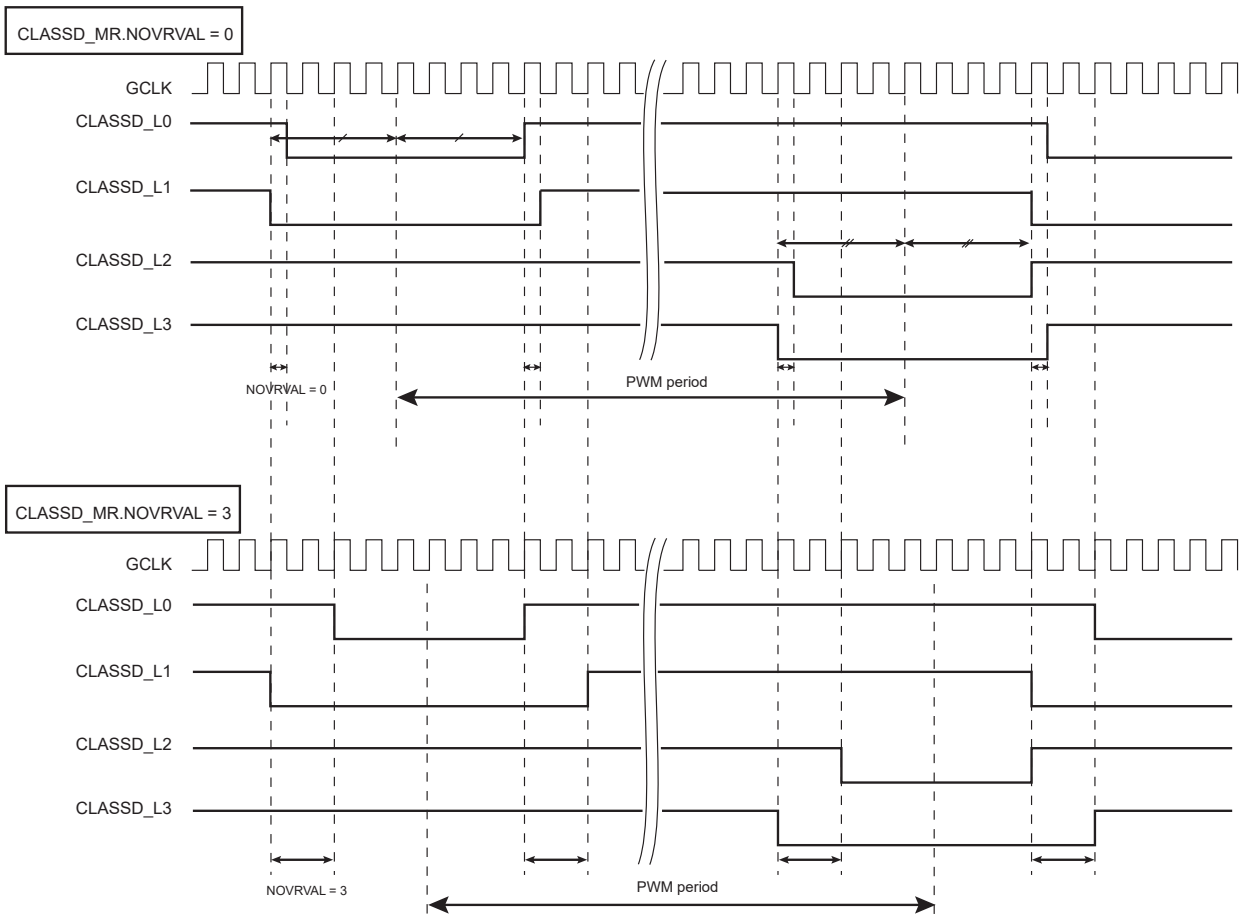
Use case example:

D1..D4 = e.g., 1N4148
Q1..Q4 = e.g., DMC2400UV
R1..R8 = 10 ohm
R9..R16 = 10 kohm
C1..C4 = 10 nF
C5..C6 = 10 µF
C7 = 1 µF

Downloaded from Arrow.com.

Figure 42-14. Use Case 1: Waveforms

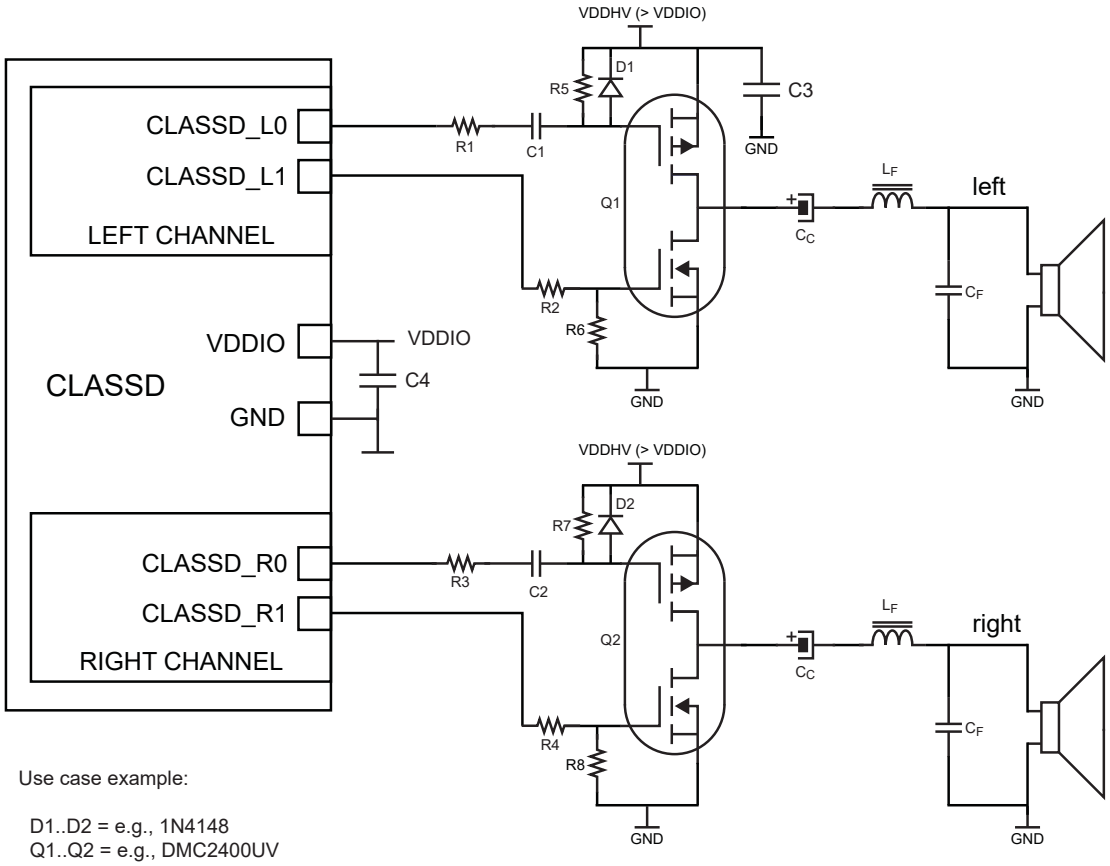
CLASSD_MR.PWMTYP = 1, CLASSD_MR.NON_OVERLAP = 1



In Use Case 1, the external power stages are made of complementary low-cost MOSFETs. In addition to the R_{DSON} and drain breakdown voltage characteristics, the choice of these components is driven by a low gate threshold voltage, a low input capacitance, a low total gate charge and a fast turn-on time characteristics. Series resistance (10 Ω) added to the gates of the MOSFETs are optional and may be adjusted to optimize the gate drive. They help to limit the output current peaks driven by the I/Os into the MOSFET gates in some cases. The 10k resistors ensure an OFF condition when not driven and the capacitor / diode network (C1..C2 / D1..D2) shifts the PMOS drive from the typical V_{DDIO} level (3.3V) to a higher supply voltage (e.g., a 5V power domain).

Downloaded from Arrow.com.

Figure 42-15. Use Case 2: Stereo Class D Amplifier With External Single-ended Power Stage



Use case example:

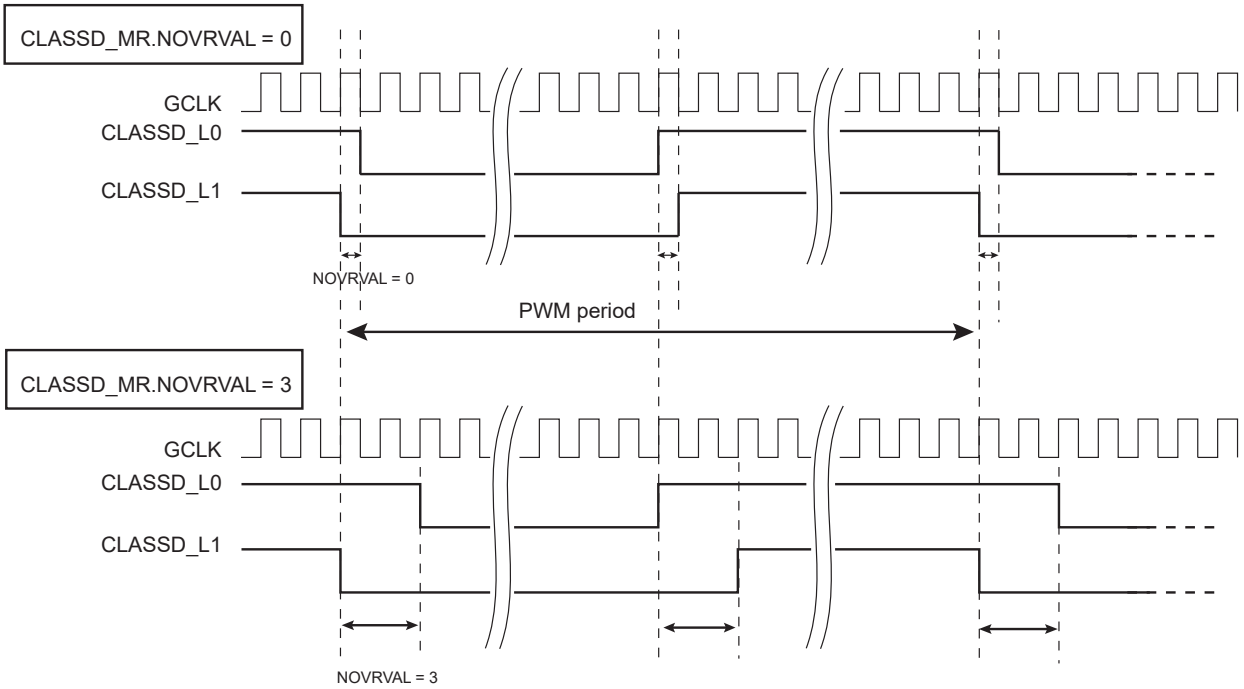
D1..D2 = e.g., 1N4148
Q1..Q2 = e.g., DMC2400UV
R1..R4 = 10 ohm
R5..R8 = 10 kohm
C1..C2 = 10 nF
C3 = 10 µF
C4 = 1 µF

In the Use Case 2 application schematic, the drive network of the MOSFETs gates follows the principles described in Use Case 1.

Downloaded from Arrow.com.

Figure 42-16. Use Case 2: Waveforms

CLASSD_MR.PWMTYP = 0, CLASSD_MR.NON_OVERLAP = 1



A coupling capacitor (C_C) and an L-C low-pass filter (L_F, C_F) are added to the output of the power stage to remove both the DC and the high frequency components of the PWM signal. C_C with the resistive part of the speaker (R_{SPK}) forms a C-R high pass filter with a corner frequency of $f_{HP} = 1 / (2 \times PI \times C_C \times R_{SPK})$.

L_F, C_F and R_{SPK} form a second-order low-pass filter of corner frequency $f_C = 1 / (2 \times PI \times \text{sqrt}(L_F \times C_F))$ and of quality factor $Q = R_{SPK} \times \text{sqrt}(C_F / L_F)$. As a numerical example, consider the case f_{HP} = 200 Hz, f_C = 30 kHz, Q = 0.707 (maximum flat response) with R_{SPK} = 8 Ω. This leads to C_C = 100 µF, L_F = 60 µH, C_F = 470 nF.

Downloaded from Arrow.com.

Figure 42-17. Use Case 3A: Stereo Audio DAC With Active Differential-to-Single Low-Pass Filter

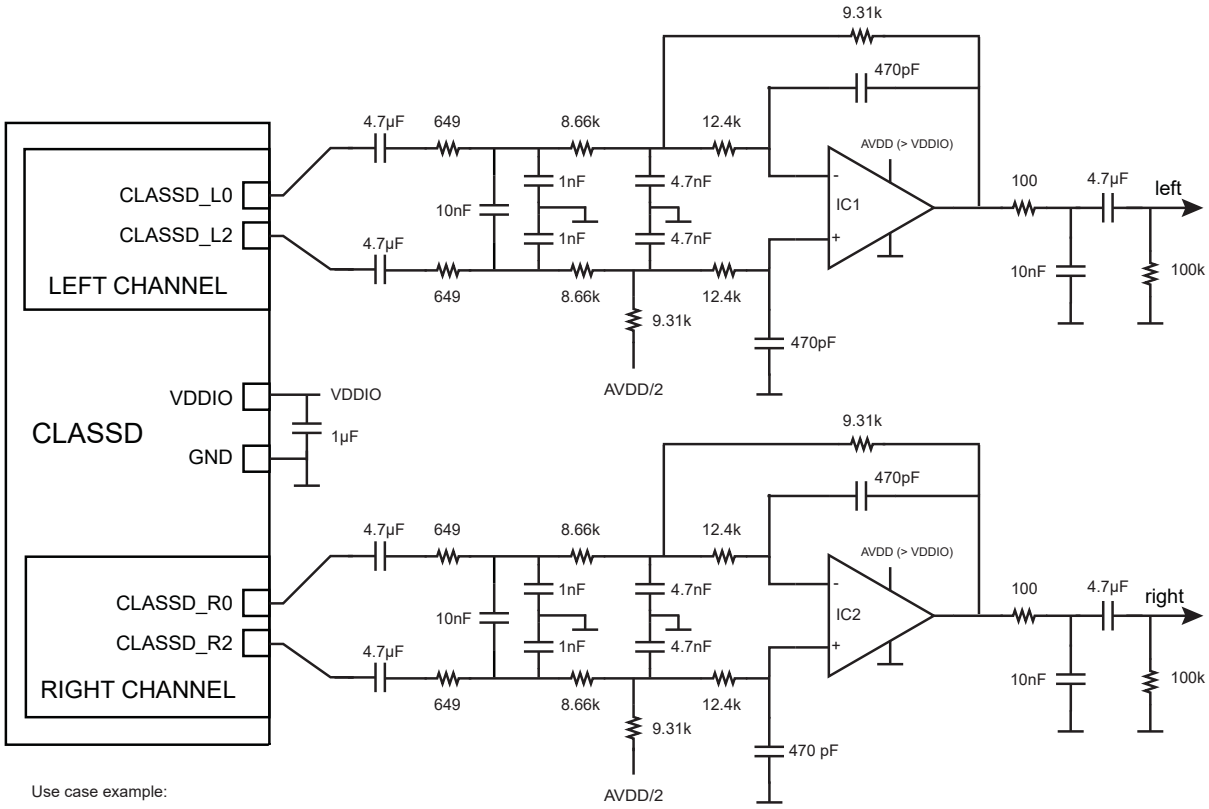
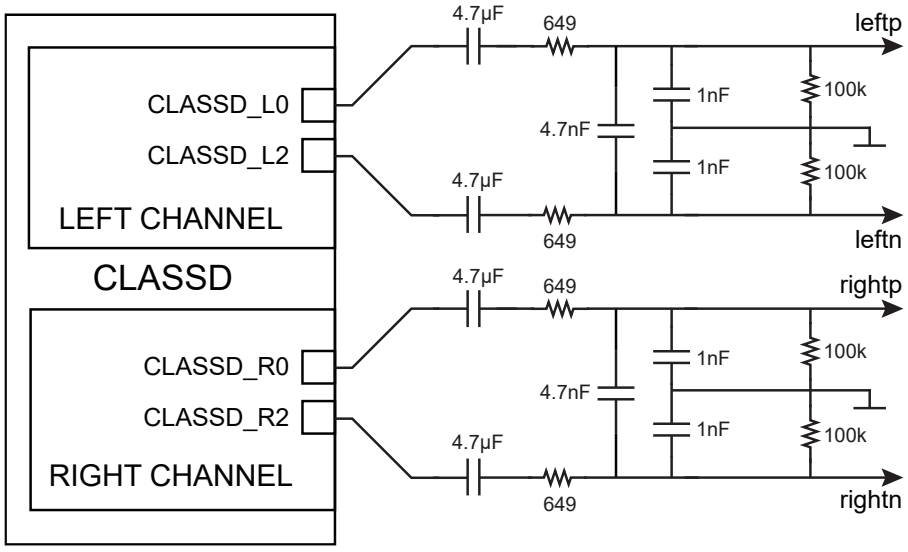
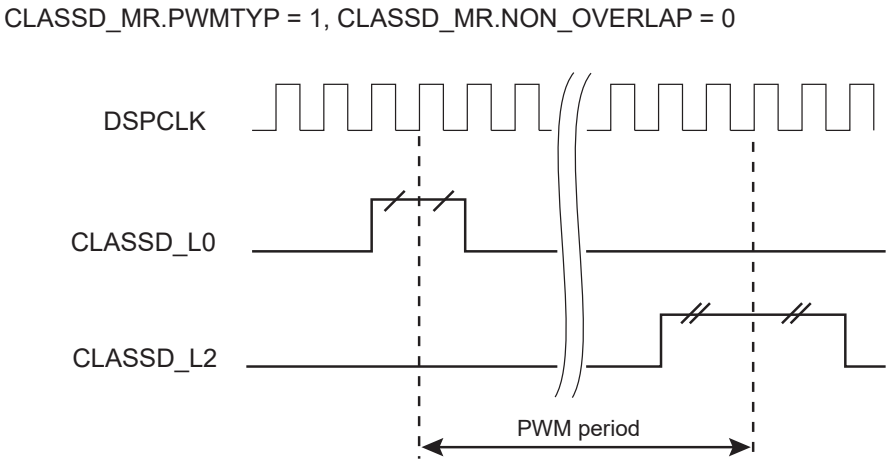


Figure 42-18. Use Case 3B: Stereo Audio DAC With Simple Passive Low-Pass Filter and Differential Outputs



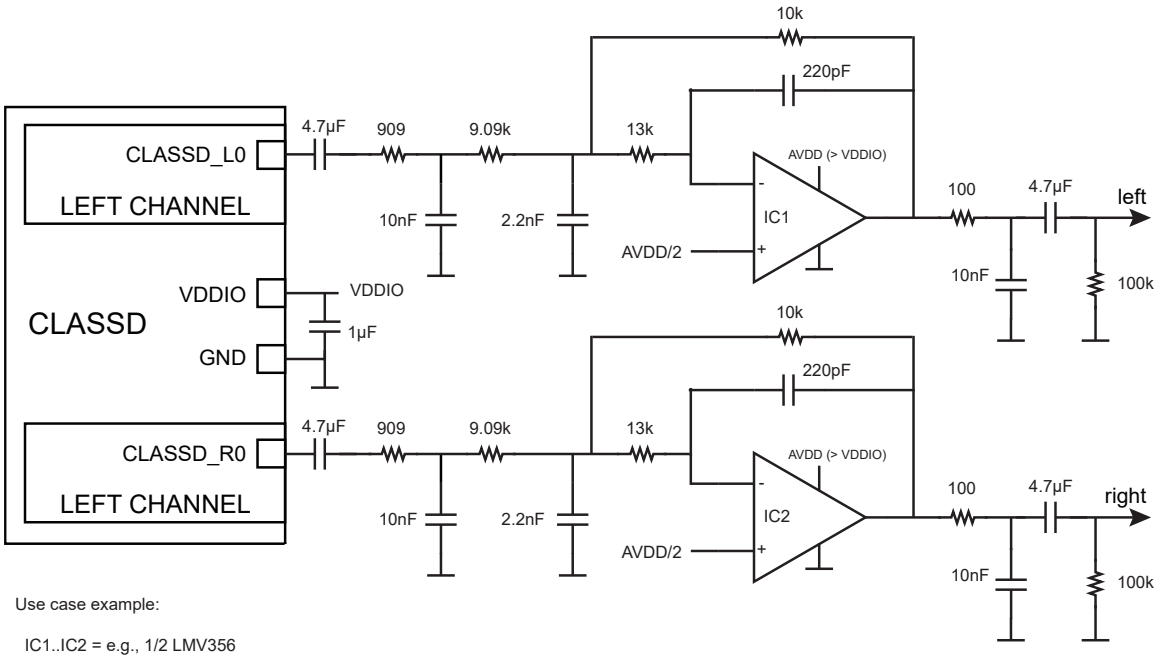
Downloaded from Arrow.com.

Figure 42-19. Use Cases 3A and 3B: Waveforms



In Use Case 3A, the CLASSD is used as an audio DAC. In this case, the differential outputs of the CLASSD are used. The application schematic suggested in figure "Use Case 3A: Stereo Audio DAC With Active differential to Single Low-Pass Filter" above implements a third order 10 kHz low-pass Butterworth filter and makes the differential to single-ended conversion. Note that in this schematic, the AVDD/2 point needs to be fed at low impedance (e.g., a buffered voltage). A simpler schematic (Use Case 3B) may also be possible, as shown in figure "Use Case 3B: Stereo Audio DAC With Simple Passive Low-Pass Filter and Differential Outputs" above, at the cost of higher out-of-band noise and differential outputs which may be acceptable in some applications.

Figure 42-20. Use Case 4A: Stereo Audio DAC With Active Low-Pass Filter and Single-ended Outputs



Downloaded from Arrow.com.

Figure 42-21. Use Case 4B: Stereo Audio DAC With Passive Low-Pass Filter and Single-ended Outputs

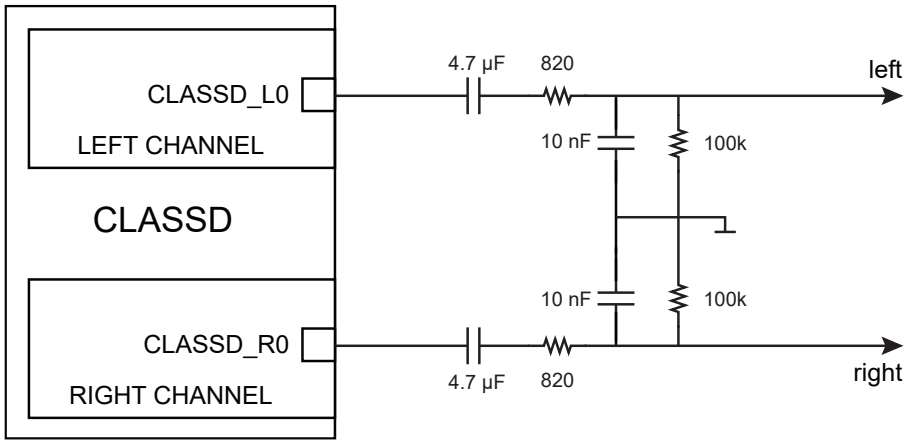
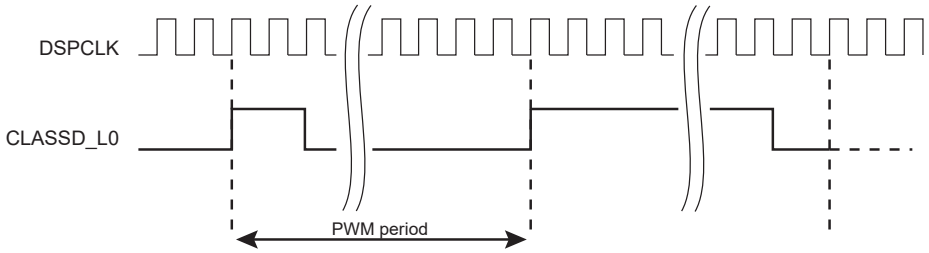


Figure 42-22. Use Cases 4A and 4B: Waveforms

CLASSD_MR.PWMTYP = 0, CLASSD_MR.NON_OVERLAP = 0



In Use Case 4A, the CLASSD is used as an audio DAC with active low-pass filter. In this case, the single-ended outputs of the CLASSD are selected (PWMTYP = 0, trailing edge PWM) which leaves more I/Os to the application. A third-order 30 kHz low-pass Butterworth filter is shown in figure "Use Case 4A: Stereo Audio DAC With Active Low-Pass Filter and Single-ended Outputs". The AVDD/2 point can be fed at relatively high impedance as no current is drawn from this point (a simple resistive divider properly decoupled is acceptable). A reduced complexity schematic is presented in figure "Use Case 4B: Stereo Audio DAC With Passive Low-Pass Filter and Single-ended Outputs" above for less constrained applications.

42.6.7 Register Write Protection

To prevent any single software error from corrupting CLASSD behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the CLASSD Write Protection Mode Register (CLASSD_WPMR).

The following registers can be write-protected:

- CLASSD Mode Register
- CLASSD Interpolator Mode Register

Downloaded from Arrow.com.

42.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CLASSD_CR	31:24								
		23:16								
		15:8								
		7:0								SWRST
0x04	CLASSD_MR	31:24								
		23:16			NOVRVAL[1:0]					NON_OVERLAP
		15:8								PWMTYP
		7:0			RMUTE	REN			LMUTE	LEN
0x08	CLASSD_INTPMR	31:24		MONOMODE[1:0]		MONO	EQCFG[3:0]			
		23:16		FRAME[2:0]			SWAP	DEEMP		DSPCLKFREQ
		15:8		ATTR[6:0]						
		7:0		ATTL[6:0]						
0x0C	CLASSD_INTSR	31:24								
		23:16								
		15:8								
		7:0								CFGERR
0x10	CLASSD_THR	31:24	RDATA[15:8]							
		23:16	RDATA[7:0]							
		15:8	LDATA[15:8]							
		7:0	LDATA[7:0]							
0x14	CLASSD_IER	31:24								
		23:16								
		15:8								
		7:0								DATRDY
0x18	CLASSD_IDR	31:24								
		23:16								
		15:8								
		7:0								DATRDY
0x1C	CLASSD_IMR	31:24								
		23:16								
		15:8								
		7:0								DATRDY
0x20	CLASSD_ISR	31:24								
		23:16								
		15:8								
		7:0								DATRDY
0x24 ... 0xE3	Reserved									
0xE4	CLASSD_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN

Downloaded from Arrow.com.

42.7.1 CLASSD Control Register

Name: CLASSD_CR
Offset: 0x00
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								W
Reset								–

Bit 0 – SWRST Software Reset

Value	Description
0	No effect.
1	Resets CLASSD, simulating a hardware reset.

Downloaded from Arrow.com.

42.7.2 CLASSD Mode Register

Name: CLASSD_MR
Offset: 0x04
Reset: 0x00010022
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the CLASSD Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			NOVRVAL[1:0]					NON_OVERLAP
Access			R/W	R/W				R/W
Reset			0	0				1

Bit	15	14	13	12	11	10	9	8
								PWMTYP
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
			RMUTE	REN			LMUTE	LEN
Access			R/W	R/W			R/W	R/W
Reset			1	0			1	0

Bits 21:20 – NOVRVAL[1:0] Non-Overlapping Value
This field has no effect when NON_OVERLAP = 0.

Value	Name	Description
0	5NS	Non-overlapping time is 5 ns
1	10NS	Non-overlapping time is 10 ns
2	15NS	Non-overlapping time is 15 ns
3	20NS	Non-overlapping time is 20 ns

Bit 16 – NON_OVERLAP Non-Overlapping Enable

Value	Description
0	Non-overlapping circuit is disabled.
1	Non-overlapping circuit is enabled.

Bit 8 – PWMTYP PWM Modulation Type
0 (TRAILING_EDGE): The signal is single-ended.
If NON_OVERLAP is cleared, the signal is sent to CLASSD_L0 and CLASSD_R0 (see figure Use Case 4A or figure Use Case 4B).
If NON_OVERLAP is set, the signal is sent to CLASSD_L0/L1 and CLASSD_R0/R1 (see figure Use Case 2).
1 (UNIFORM): The signal is differential.
If NON_OVERLAP is cleared, the signal is sent to CLASSD_L0/L2 and CLASSD_R0/R2 (see figure Use Case 3A or figure Use Case 3B).
If NON_OVERLAP is set, the signal is sent to CLASSD_L0/L1/L2/L3 and CLASSD_R0/R1/R2/R3 (see figure Use Case 1).

Bit 5 – RMUTE Right Channel Mute

Value	Description
0	Right channel is unmuted.

Downloaded from Arrow.com.

Value	Description
1	Right channel is muted.

Bit 4 – REN Right Channel Enable

Value	Description
0	Right channel is disabled.
1	Right channel is enabled.

Bit 1 – LMUTE Left Channel Mute

Value	Description
0	Left channel is unmuted.
1	Left channel is muted.

Bit 0 – LEN Left Channel Enable

Value	Description
0	Left channel is disabled.
1	Left channel is enabled.

Downloaded from Arrow.com.

42.7.3 CLASSD Interpolator Mode Register

Name: CLASSD_INTPMR
Offset: 0x08
Reset: 0x00304E4E
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the CLASSD Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
		MONOMODE[1:0]		MONO	EQCFG[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		FRAME[2:0]			SWAP	DEEMP		DSPCLKFREQ
Access		R/W	R/W	R/W	R/W	R/W		R/W
Reset		0	1	1	0	0		0

Bit	15	14	13	12	11	10	9	8
		ATTR[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		1	0	0	1	1	1	0

Bit	7	6	5	4	3	2	1	0
		ATTL[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		1	0	0	1	1	1	0

Bits 30:29 – MONOMODE[1:0] Mono Mode Selection
Defines which signal is sent to both channels when the MONO bit is set.

Value	Name	Description
0	MONOMIX	(left + right) / 2 is sent to both channels
1	MONOSAT	(left + right) is sent to both channels. If the sum is too high, the result is saturated.
2	MONOLEFT	THR[15:0] is sent to both the left and the right channels
3	MONORIGHT	THR[31:16] is sent to both the left and the right channels

Bit 28 – MONO Mono Signal
0 (DISABLED): The signal is sent stereo to the left and right channels.
1 (ENABLED): The same signal is sent to both the left and the right channels. The sent signal is defined by the MONOMODE field value.

Bits 27:24 – EQCFG[3:0] Equalization Selection
EQCFG field values 13–15 = flat response

Value	Name	Description
0	FLAT	Flat response
1	BBOOST12	Bass boost +12 dB
2	BBOOST6	Bass boost +6 dB
3	BCUT12	Bass cut -12 dB
4	BCUT6	Bass cut -6 dB
5	MBOOST3	Medium boost +3 dB
6	MBOOST8	Medium boost +8 dB
7	MCUT3	Medium cut -3 dB
8	MCUT8	Medium cut -8 dB
9	TBOOST12	Treble boost +12 dB
10	TBOOST6	Treble boost +6 dB
11	TCUT12	Treble cut -12 dB

Downloaded from Arrow.com.

Value	Name	Description
12	TCUT6	Treble cut -6 dB

Bits 22:20 – FRAME[2:0] CLASSD Incoming Data Sampling Frequency

Value	Name	Description
0	FRAME_8K	8 kHz
1	FRAME_16K	16 kHz
2	FRAME_32K	32 kHz
3	FRAME_48K	48 kHz
4	FRAME_96K	96 kHz
5	FRAME_22K	22.05 kHz
6	FRAME_44K	44.1 kHz
7	FRAME_88K	88.2 kHz

Bit 19 – SWAP Swap Left and Right Channels
0 (LEFT_ON_LSB): Left channel is on CLASSD_THR[15:0], right channel is on CLASSD_THR[31:16].
1 (RIGHT_ON_LSB): Right channel is on CLASSD_THR[15:0], left channel is on CLASSD_THR[31:16].

Bit 18 – DEEMP Enable De-emphasis Filter
0 (DISABLED): De-emphasis filter is disabled.
1 (ENABLED): De-emphasis filter is enabled.

Bit 16 – DSPCLKFREQ DSP Clock Frequency
0 (12M288): DSP Clock (DSPCLK) is 12.288 MHz.
1 (11M2896): DSP Clock (DSPCLK) is 11.2896 MHz.

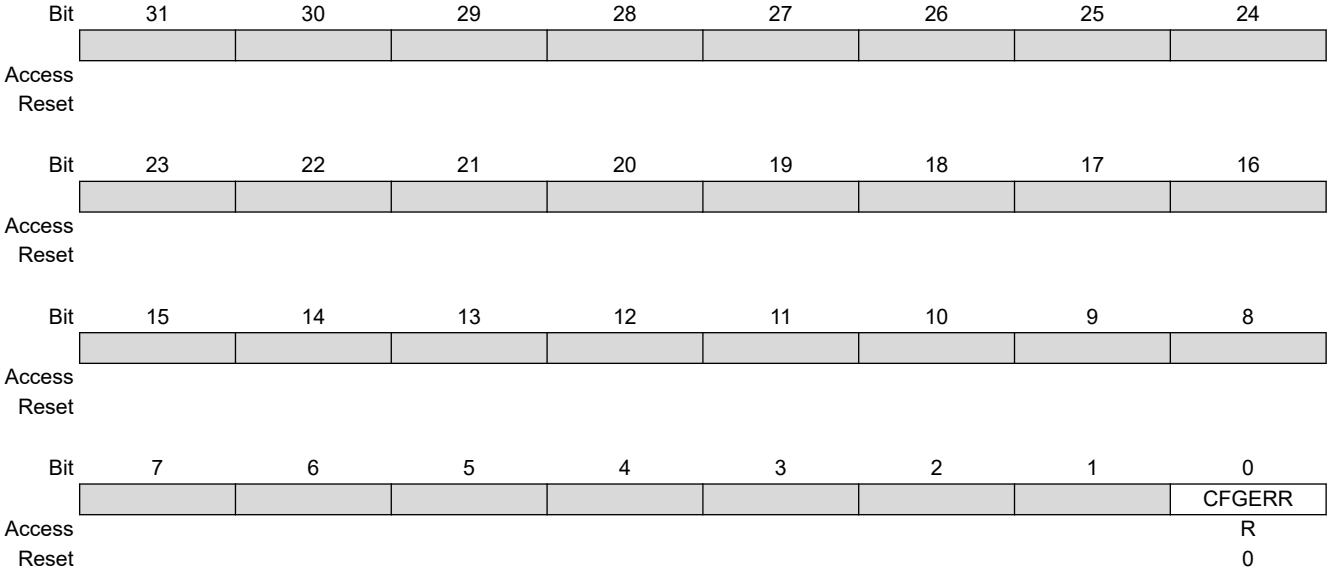
Bits 14:8 – ATTR[6:0] Right Channel Attenuation
Right channel attenuation is defined as follows:
– if ATTR ≤ 77 the attenuation is -ATTR dB
– else the right signal is muted

Bits 6:0 – ATTL[6:0] Left Channel Attenuation
Left channel attenuation is defined as follows:
– if ATTL ≤ 77 the attenuation is -ATTL dB
– else the left signal is muted

Downloaded from Arrow.com.

42.7.4 CLASSD Interpolator Status Register

Name: CLASSD_INTSR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only



Bit 0 – CFGERR Configuration Error

Value	Description
0	The frame and clock configurations are correct.
1	The frame and clock configurations are incorrect (see Clock Configuration for information about allowed configurations).

Downloaded from Arrow.com.

42.7.5 CLASSD Transmit Holding Register

Name: CLASSD_THR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	RDATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RDATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LDATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LDATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – RDATA[15:0] Right Channel Data

Bits 15:0 – LDATA[15:0] Left Channel Data

Downloaded from Arrow.com.

42.7.6 CLASSD Interrupt Enable Register

Name: CLASSD_IER
Offset: 0x14
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								W
Reset								–

Bit 0 – DATRDY Data Ready

Value	Description
0	No effect.
1	Enables the interrupt when CLASSD is ready to receive new data to convert.

Downloaded from Arrow.com.

42.7.7 CLASSD Interrupt Disable Register

Name: CLASSD_IDR
Offset: 0x18
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								W
Reset								–

Bit 0 – DATRDY Data Ready

Value	Description
0	No effect.
1	Disables the interrupt when CLASSD is ready to receive new data to convert.

Downloaded from Arrow.com.

42.7.8 CLASSD Interrupt Mask Register

Name: CLASSD_IMR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								R
Reset								0

Bit 0 – DATRDY Data Ready

Value	Description
0	The interrupt is disabled.
1	The interrupt is enabled.

Downloaded from Arrow.com.

42.7.9 CLASSD Interrupt Status Register

Name: CLASSD_ISR
Offset: 0x20
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								R
Reset								0

Bit 0 – DATRDY Data Ready

Value	Description
0	CLASSD has not been ready to convert a value since the last read of CLASSD_ISR.
1	CLASSD has been ready to convert a value since the last read of CLASSD_ISR.

Downloaded from Arrow.com.

42.7.10 CLASSD Write Protection Mode Register

Name: CLASSD_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x434C44	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 0 – WPEN Write Protection Enable

See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x434C44 (“CLD” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x434C44 (“CLD” in ASCII).

Downloaded from Arrow.com.

43. Inter-IC Sound Multi-Channel Controller (I2SMCC)

43.1 Description

The Inter-IC Sound Controller (I2SMCC) provides a 5-wire, bidirectional, synchronous, digital audio link to external audio devices: I2SMCC_DIN, I2SMCC_DOUT, I2SMCC_WS, I2SMCC_CK, and I2SMCC_MCK pins.

The I2SMCC complies with the Inter-IC Sound (I^2S) bus specification and supports a Time Division Multiplexed (TDM) interface with external multi-channel audio codecs.

The I2SMCC consists of a receiver, a transmitter and a common clock generator that can be enabled separately to provide Master, Slave or Controller modes with receiver and/or transmitter active.

DMA Controller channels, separate for the receiver and for the transmitter, allow a continuous high bit rate data transfer without processor intervention to the following:

- Audio CODECs in Master, Slave, or Controller mode
- Stereo DAC or ADC through a dedicated I^2S serial interface
- Multi-channel or multiple stereo DACs or ADCs, using the TDM format

The I2SMCC uses a single DMA Controller channel for all audio channels.

The 8- and 16-bit compact stereo formats reduce the required DMA Controller bandwidth by transferring the left and right samples within the same data word.

In Master mode, the I2SMCC can produce a 16 f_s to 1024 f_s master clock that provides an over-sampling clock to an external audio codec or digital signal processor (DSP).

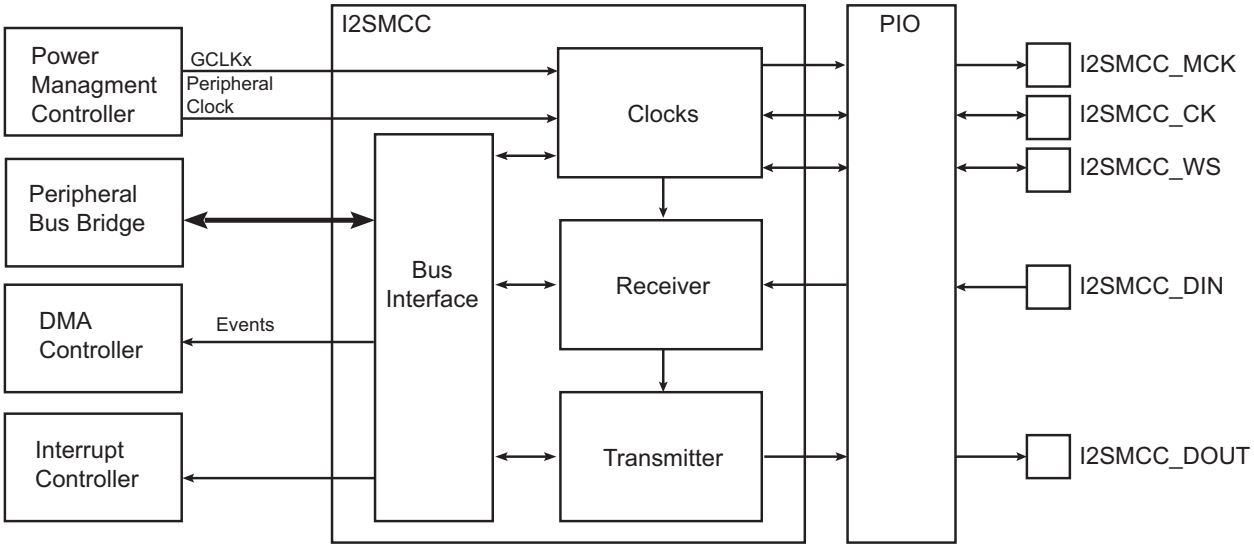
43.2 Embedded Characteristics

- Compliant with Inter-IC Sound (I^2S) Bus Specification
- Master, Slave, and Controller Modes
  - Slave: Data Received/Transmitted
  - Master: Data Received/Transmitted And Clocks Generated
  - Controller: Clocks Generated
- Individual Enable and Disable of Receiver, Transmitter and Clocks
- Configurable Clock Generator Common to Receiver and Transmitter
  - Suitable for a Wide Range of Sample Frequencies (f_s), Including 32 kHz, 44.1 kHz, 48 kHz, 88.2 kHz, 96 kHz, and 192 kHz
  - 16 f_s to 1024 f_s Master Clock Generated for External Oversampling Data Converters
- Support for Multiple Data Formats
  - 32-, 24-, 20-, 18-, 16-, and 8-bit Mono or Stereo Format
  - 16- and 8-bit Compact Stereo Format, with Left and Right Samples Packed in the Same Word to Reduce Data Transfers
- Support for Multiple Data Frame Formats
  - 2-channel I^2S with Word Select
  - 1- to 8-channel Time Division Multiplexed (TDM) with Frame Synchronization
- DMA Controller Interfaces the Receiver and Transmitter to Reduce Processor Overhead
- Smart Holding Registers Management to Avoid Audio Channel Mix After Overrun or Underrun

Downloaded from Arrow.com.

43.3 Block Diagram

Figure 43-1. I2SMCC Block Diagram



43.4 I/O Lines Description

Table 43-1. I/O Lines Description

Pin Name	Pin Description	Type
I2SMCC_MCK	Master Clock	Output
I2SMCC_CK	Serial Clock	Input/Output
I2SMCC_WS	I^2S Word Select or TDM Frame Synchronization	Input/Output
I2SMCC_DIN	Serial Data Input	Input
I2SMCC_DOUT	Serial Data Output	Output

43.5 Product Dependencies

To use the I2SMCC, other parts of the system must be configured correctly, as described below.

43.5.1 I/O Lines

The I2SMCC pins may be multiplexed with I/O Controller lines. The user must first program the PIO Controller to assign the required I2SMCC pins to their peripheral function. If the I2SMCC I/O lines are not used by the application, they can be used for other purposes by the PIO Controller. The user must enable the I2SMCC inputs and outputs that are used.

43.5.2 Power Management

If the processor enters a Sleep mode that disables clocks used by the I2SMCC, the I2SMCC stops functioning and resumes operation after the system wakes up from Sleep mode.

43.5.3 Clocks

The I2SMCC runs from the peripheral clock and the generic clock (GCLK), both generated by the Power Management Controller (PMC). Prior to using the I2SMCC, the user must first program the PMC. The I2S master and serial clock can be generated either from the peripheral clock or the generic clock.

Downloaded from Arrow.com.

In a similar way, the I2SMCC must be disabled before removing its clock source to avoid freezing it in an undefined state.

43.5.4 DMA Controller

The I2SMCC interfaces to the DMA Controller. Using the I2SMCC DMA functionality requires the DMA Controller to be programmed first.

43.5.5 Interrupt Sources

The I2SMCC interrupt line is connected to the Interrupt Controller. Using the I2SMCC interrupt requires the Interrupt Controller to be programmed first.

43.6 Functional Description

43.6.1 Initialization

The I2SMCC features a receiver, a transmitter and a clock generator for Master and Controller modes. Receiver and transmitter share the same serial clock and word select.

Before enabling the I2SMCC, the selected configuration must be written to the I2SMCC Mode Register A (I2SMCC_MRA). If I2SMCC_MRA.FORMAT is configured in one of the TDM formats, then the I2SMCC_MRA.NBCHAN and I2SMCC_MRA.TDMFS fields must also be written.

Once the I2SMCC_MRA has been written, the I2SMCC clock generator, receiver, and transmitter can be enabled by writing a ‘1’ to the CKEN, RXEN, and TXEN bits in the Control Register (I2SMCC_CR). The clock generator can be enabled alone in Controller mode to output clocks to the I2SMCC_MCK, I2SMCC_CK, and I2SMCC_WS pins. The clock generator must also be enabled if the receiver or the transmitter is enabled.

The clock generator, receiver, and transmitter can be disabled independently by writing a ‘1’ to I2SMCC_CR.CXDIS, I2SMCC_CR.RXDIS and/or I2SMCC_CR.TXDIS, respectively. Once requested to stop, they stop only when the transmission of the pending frame transmission is completed.

43.6.2 Basic Operation

The receiver can be operated by reading the Receiver Holding Register (I2SMCC_RHR), whenever the Receive Left x Ready (RXLRDYx) bit or the Receive Right Ready (RXRRDYx) bit in the Interrupt Status Register A (I2SMCC_ISRA) is set. Successive values read from I2SMCC_RHR correspond to the samples from the first left audio channel to the last left audio channel enabled then from the first right audio channel to the last right audio channel enabled, or from channels 0 to I2SMCC_MRA.NBCHAN in TDM mode for the successive frames.

The transmitter can be operated by writing to the Transmitter Holding Register (I2SMCC_THR), whenever the Transmit Left x Ready (TXLRDYx) bit or the Transmit Right x Ready (TXRRDYx) bit in the I2SMCC_ISRA is set. Successive values written to I2SMCC_THR correspond to the samples from the first left audio channel to the last left audio channel enabled, then from the first right audio channel to the last right audio channel enabled, or from channels 0 to I2SMCC_MRA.NBCHAN in TDM mode for the successive frames.

The RXLRDYx, RXRRDYx, TXLRDYx and TXRRDYx bits can be polled by reading the I2SMCC_ISRA.

The I2SMCC processor load can be reduced by enabling interrupt-driven operation. The RXLRDYx, RXRRDYx, TXLRDYx and/or TXRRDYx interrupt requests can be enabled by writing a ‘1’ to the corresponding bit in the Interrupt Enable Register A (I2SMCC_IERA). The interrupt service routine associated to the I2SMCC interrupt request is executed when at least one of the RXLRDYx, RXRRDYx, TXLRDYx and TXRRDYx status bits is set.

43.6.3 Master, Controller and Slave Modes

In Master and Controller modes, the I2SMCC provides the master clock, the serial clock and the word select. I2SMCC_MCK, I2SMCC_CK, and I2SMCC_WS pins are outputs.

In Controller mode, the I2SMCC receiver and transmitter are disabled. Only the clocks are enabled and used by an external receiver and/or transmitter.

In Slave mode, the I2SMCC receives the serial clock and the word select from an external master. I2SMCC_CK and I2SMCC_WS pins are inputs.

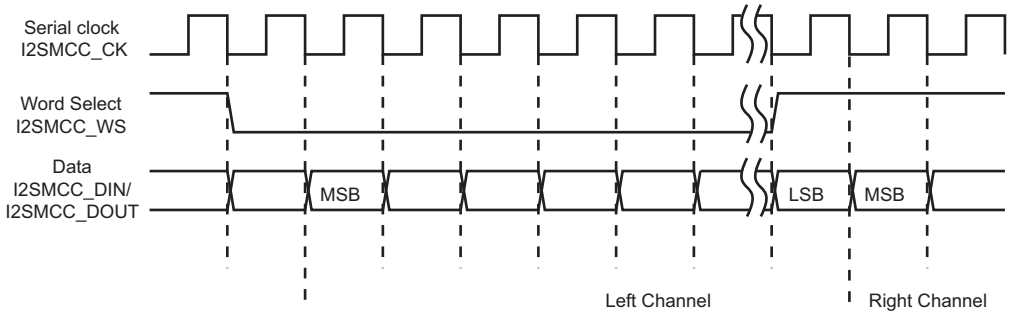
Downloaded from Arrow.com.

The mode is selected by writing the MODE field in the I2SMCC_MRA. Since the MODE field changes the direction of the I2SMCC_WS and I2SSCK pins, the I2SMCC_MRA must only be written when the I2SMCC is stopped in order to avoid unwanted glitches on the I2SMCC_WS and I2SMCC_CK pins.

43.6.4 I²S Reception and Transmission Sequence

As specified in the I²S protocol, data bits are left-justified in the word select time slot, with the MSB transmitted first, starting one clock period after the transition on the word select line.

Figure 43-2. I²S Reception and Transmission Sequence



Data bits are sent on the falling edge of the serial clock and sampled on the rising edge of the serial clock. The word select line indicates the channel in transmission, a low level for the left channel and a high level for the right channel.

The length of transmitted words can be chosen among 8, 16, 18, 20, 24, and 32 bits by writing I2SMCC_MRA.DATALENGTH.

If the time slot allows for more data bits than written in I2SMCC_MRA.DATALENGTH, zeroes are appended to the transmitted data word or extra received bits are discarded.

The slot length is defined in the following table.

Table 43-2. Slot Length (I²S format)

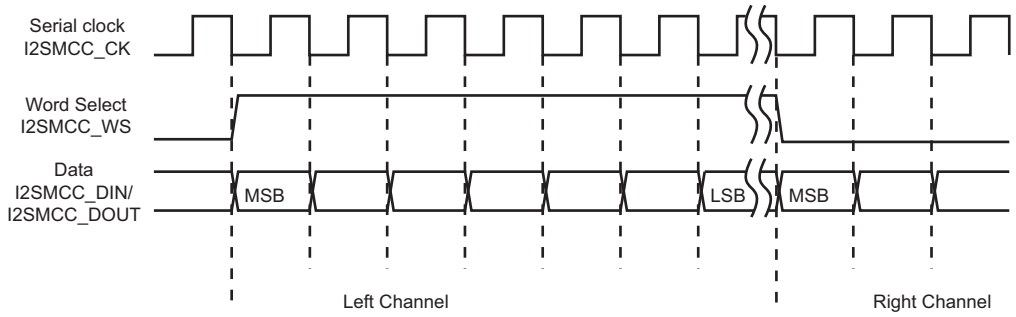
I2SMCC_MRA.DATALENGTH	Word Length	Slot Length
0	32 bits	32
1	24 bits	32 if I2SMCC_MRA.IWS = 0 24 if I2SMCC_MRA.IWS = 1
2	20 bits	
3	18 bits	
4	16 bits	16
5	16 bits compact stereo	
6	8 bits	8
7	8 bits compact stereo	

43.6.5 Left-Justified Reception and Transmission Sequence

As specified in the I²S protocol, data bits are left-justified in the word select time slot, with the MSB transmitted first, starting at the same clock period as the transition on the word select line.

Downloaded from Arrow.com.

Figure 43-3. Left-Justified Reception and Transmission Sequence



Data bits are sent on the falling edge of the serial clock and sampled on the rising edge of the serial clock. The word select line indicates the channel in transmission, a low level for the right channel and a high level for the left channel.

The length of transmitted words can be chosen among 8, 16, 18, 20, 24, and 32 bits by writing I2SMCC_MRA.DATALENGTH.

If the time slot allows for more data bits than written in I2SMCC_MRA.DATALENGTH, zeroes are appended to the transmitted data word or extra received bits are discarded.

The Slot Length is defined in the following table.

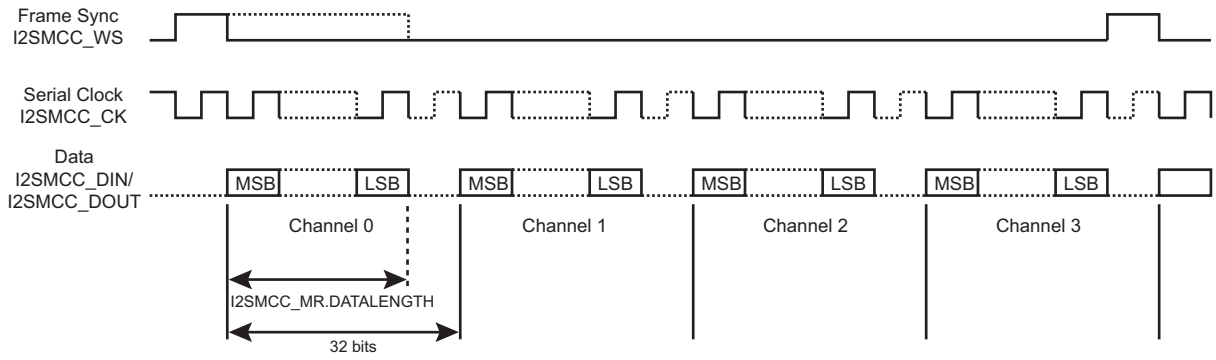
Table 43-3. Slot Length (Left-Justified format)

I2SMCC_MRA.DATALENGTH	Word Length	Slot Length
0	32 bits	32
1	24 bits	32 if I2SMCC_MRA.IWS = 0 24 if I2SMCC_MRA.IWS = 1
2	20 bits	
3	18 bits	
4	16 bits	16
5	16 bits compact stereo	
6	8 bits	8
7	8 bits compact stereo	

43.6.6 TDM Reception and Transmission Sequence

In Time Division Multiplexed (TDM) format, one to eight data words are sent or received within each frame. As specified in the I^2S protocol, data bits are left-justified in the channel time slot, with the MSB transmitted first, starting one clock period after the transition on the word select line. Each time slot is 32 bits long.

Figure 43-4. TDM Reception and Transmission Sequence



Downloaded from Arrow.com.

Data bits are sent on the falling edge of the serial clock and sampled on the rising edge of the serial clock. The I2SMCC_WS pin provides a frame synchronization signal, starting one I2SMCC_CK period before the MSB of channel 0.

The TDM format is selected by writing a '2' to I2SMCC_MRA.FORMAT.

The Frame Synchronization pulse can be either one I2SMCC_CK period, 16-bit I2SMCC_CK period (half time slot) or one 32-bit time slot. This selection is done by writing the I2SMCC_MRA.TDMFS bit.

The number of channels is selected by writing the I2SMCC_MRA.NBCHAN field.

The Frame Synchronization pulse set to 32-bit time slot with the number of channel set to 1 configuration is not supported.

The length of transmitted words can be chosen among 8, 16, 18, 20, 24, and 32 bits by writing the I2SMCC_MRA.DATALENGTH field.

If the time slot allows for more data bits than programmed in the I2SMCC_MRA.DATALENGTH field, zeroes are appended to the transmitted data word or extra received bits are discarded.

43.6.7 Serial Clock and Word Select Generation

The generation of clocks in the I2SMCC is described in the following figure.

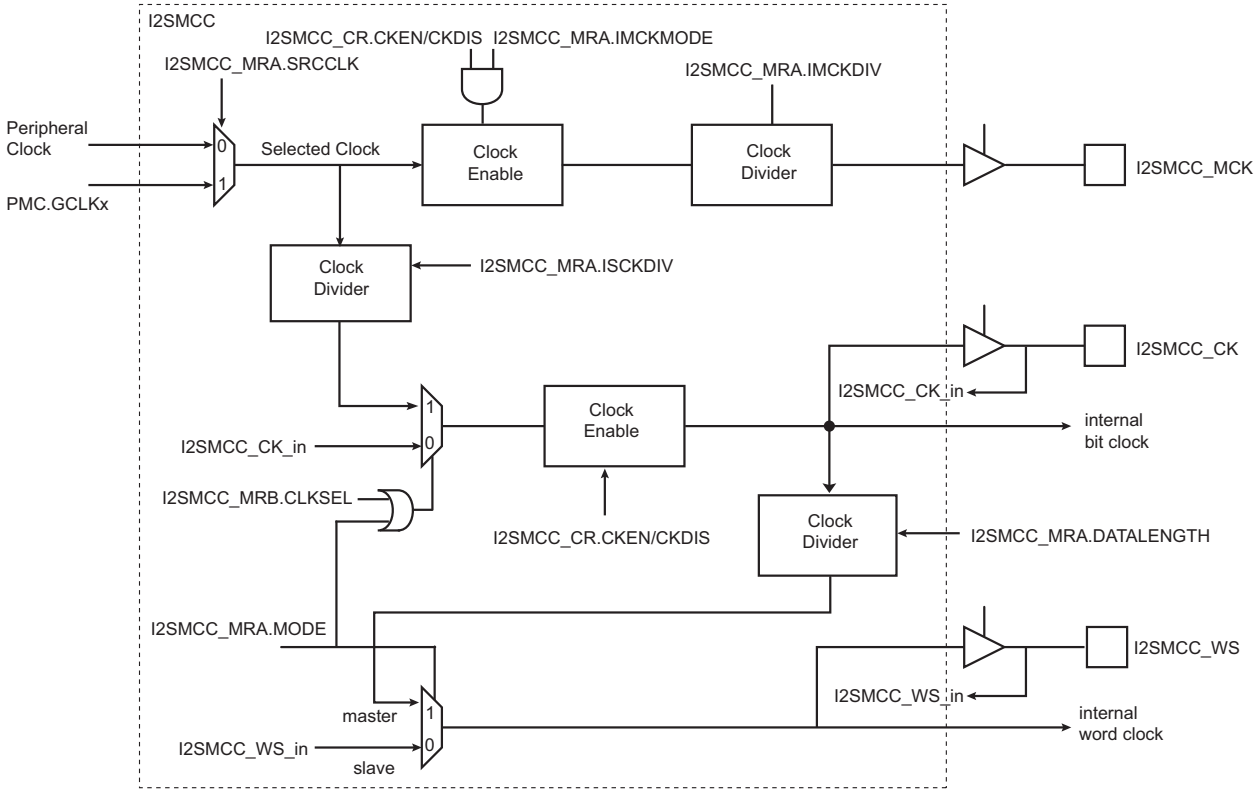
In Slave mode, the word select clock is driven by an external master and the serial clock is driven by either an external master or the internal clock circuitry. I2SMCC_CK and I2SMCC_WS pins are inputs.

In Master mode, the user configures the master clock, serial clock, and word select clock through the I2SMCC_MRA. I2SMCC_MCK, I2SMCC_CK, and I2SMCC_WS pins are outputs, MCK and SCK frequencies are configured independently using IMCKDIV and ISCKDIV bits of I2SMCC_MRA, respectively.

If a master clock output is not required, the MCK clock is used as I2SMCC_CK by clearing I2SMCC_MRA.IMCKMODE.

The I2SMCC_WS pin is used as word select in I^2S format and as frame synchronization in TDM format, as described in I2S Reception and Transmission Sequence and TDM Reception and Transmission Sequence, respectively.

Figure 43-5. I2SMCC Clock Generation



Downloaded from Arrow.com.

43.6.8 Mono

When the Transmit Mono bit (TXMONO) in I2SMCC_MRA is set, data written to the left channel is duplicated to the right output channel. In TDM mode, with more than two channels numbered from 0, data written to the even-numbered channels is duplicated to the following odd-numbered channel. When TXMONO is set, TXRRDYx bits of I2SMCC_ISRA are always read as ‘0’ and the behavior of TXRUNFx bits is unchanged.

When the Receive Mono bit (RXMONO) in I2SMCC_MRA is set, data received from the left channel is duplicated to the right input channel. In TDM mode, with more than two channels numbered from 0, data received from the even-numbered channels is duplicated to the following odd-numbered channel. When RXMONO is set, the behavior of RXRRDYx and RXROVFx bits is unchanged.

43.6.9 Holding Registers

The I2SMCC user interface includes two common holding registers—the Receive Holding Register (I2SMCC_RHR) and the Transmit Holding Register (I2SMCC_THR). The common registers are used to access audio samples for all audio channels.

Each I^2S or TDM channel has its own register. The register depth depending on the configuration is available in the following table.

Table 43-4. Registers Depth

I2SMCC_MRA.FORMAT	I2SMCC_MRA.NBCHAN	Register Depth
0 or 1 (I^2S or LJ)	NA	4 words
2 or 3 (TDM or TDMLJ)	0 or 1	4 words
	2 or 3	2 words
	4, 5, 6 or 7	1 word

43.6.9.1 Common Registers

The Receiver Holding Register (I2SMCC_RHR) and the Transmitter Holding Register (I2SMCC_THR) provide an access to all channels enabled through a single location.

When a new data word is available, the corresponding RXLRDYx bit or RXRRDYx bit in I2SMCC_ISRA is set. Reading I2SMCC_RHR clears this bit. In I^2S or Left-Justified mode, consecutive access to I2SMCC_RHR reads first the left channel then the right channel. In TDM or TDM Left-Justified mode, consecutive access to I2SMCC_RHR reads the TDM channels enabled.

A receive overrun condition occurs if a new data word becomes available for the left channel x or right channel x before the previous data word has been read from I2SMCC_RHR. In this case, the corresponding RXLOVFx or RXROVFx bit in I2SMCC_ISRA is set. Reading I2SMCC_ISRA clears this bit.

When a data can be written in I2SMCC_THR, the corresponding TXLRDYx bit or TXRRDYx bit in I2SMCC_ISRA is set. Writing to I2SMCC_THR clears this bit. In I2S or Left-Justified mode, consecutive access to I2SMCC_THR writes first the left channel then the right channel. In TDM or TDM Left-Justified mode, consecutive access to I2SMCC_THR writes the TDM channels enabled.

A transmit underrun condition occurs if a new data word needs to be transmitted before it has been written to I2SMCC_THR. In this case, the corresponding TXLUNFx or TXRUNFx bit in I2SMCC_ISRA is set. Reading I2SMCC_ISRA clears this bit.

In case of transmit underrun, if the value of I2SMCC_MRA.TXSAME is ‘0’, then a ‘0’ is transmitted. If the value of I2SMCC_MRA.TXSAME is ‘1’, then the previous data word for the current transmit channel number is transmitted.

After a transmit underrun, the data written in I2SMCC_THR is discarded to keep the left and right channels synchronized.

Data words are right-justified in the common registers (I2SMCC_RHR and I2SMCC_THR). For the 16-bit compact stereo data format, the left sample uses bits [15:0] and the right sample uses bits [31:16] of the same data word. For the 8-bit compact stereo data format, the left sample uses bits [7:0] and the right sample uses bits [15:8] of the same data word.

Downloaded from Arrow.com.

43.6.10 DMA Controller Operation

All receiver audio channels and all transmitter audio channels are each assigned to a single DMA Controller channel.

The DMA Controller reads from the I2SMCC_RHR and writes to the I2SMCC_THR for all audio channels successively.

The DMA Controller transfers may use 32-bit word, 16-bit halfword, or 8-bit byte depending on the value of the I2SMCC_MRA.DATALENGTH field.

The DMA chunk size field (DMACHUNK) of the Mode Register B(I2SMCC_MRB) should correspond to the DMA channel configuration. The supported chunk according to the configuration is available in the two tables below

Table 43-5. TX DMA Chunk Configurations

FORMAT	TXMONO	DATALENGTH	NBCHAN	DMACHUNK	Description
0 or 1 (I^2S or LJ)	0 (Stereo)	0, 1, 2, 3, 4 or 6 (32, 24, 20, 18, 16 or 8 bits)	NA	0	1-word chunk
				1	2-word chunk
				2	4-word chunk
				3	8-word chunk
	1 (Mono)			0	1-word chunk
				1	2-word chunk
				2	4-word chunk
				3	
	0 or 1 (Stereo or Mono)	5 or 7 (16 bits compact or 8 bits compact)		0	1-word chunk
				1	2-word chunk
				2	4-word chunk
				3	

Downloaded from Arrow.com.

.....continued

FORMAT	TXMONO	DATALENGTH	NBCHAN	DMACHUNK	Description
2 or 3 (TDM or TDMLJ)	NA	NA	0 (1 channel)	0	1-word chunk
				1	2-word chunk
				2	4-word chunk
				3	
	0 (Stereo)	0, 1, 2, 3, 4 or 6 (32, 24, 20, 18, 16 or 8 bits)	1, 3 or 7 (2, 4 or 8 channels)	0	1-word chunk
				1	2-word chunk
				2	4-word chunk
				3	8-word chunk
			2, 4, 5 or 6 (3, 5, 6 or 7 channels)	0	1-word chunk
				1	2-word chunk
				2	4-word chunk
				3	
	1 (Mono)		1, 3 or 7 (2, 4 or 8 channels)	0	1-word chunk
				1	2-word chunk
				2	4-word chunk
				3	
			2, 4, 5 or 6 (3, 5, 6 or 7 channels)	0	1-word chunk
				1	2-word chunk
				2	
				3	
	0 or 1 (Stereo or Mono)	5 or 7 (16 bits compact or 8 bits compact)	1, 3 or 7 (2, 4 or 8 channels)	0	1-word chunk
				1	2-word chunk
				2	4-word chunk
				3	
			2, 4, 5 or 6 (3, 5, 6 or 7 channels)	0	1-word chunk
				1	2-word chunk
				2	
				3	

Downloaded from Arrow.com.

Table 43-6. RX DMA Chunk Configurations

FORMAT	DATALENGTH	NBCHAN	DMACHUNK	Description
0 or 1 (I^2S or LJ)	0, 1, 2, 3, 4 or 6 (32, 24, 20, 18, 16 or 8 bits)	NA	0	1-word chunk
			1	2-word chunk
			2	4-word chunk
			3	8-word chunk
	5 or 7 (16 bits compact or 8 bits compact)		0	1-word chunk
			1	2-word chunk
			2	4-word chunk
			3	
2 or 3 (TDM or TDMLJ)	NA	0 (1 channel)	0	1-word chunk
			1	2-word chunk
			2	4-word chunk
			3	
	0, 1, 2, 3, 4 or 6 (32, 24, 20, 18, 16 or 8 bits)	1, 3 or 7 (2, 4 or 8 channels)	0	1-word chunk
			1	2-word chunk
			2	4-word chunk
			3	8-word chunk
		2, 4, 5 or 6 (3, 5, 6 or 7 channels)	0	1-word chunk
			1	2-word chunk
			2	4-word chunk
			3	
	5 or 7 (16 bits compact or 8 bits compact)	1, 3 or 7 (2, 4 or 8 channels)	0	1-word chunk
			1	2-word chunk
			2	4-word chunk
			3	
		2, 4, 5 or 6 (3, 5, 6 or 7 channels)	0	1-word chunk
			1	2-word chunk
			2	
			3	

43.6.11 Loop-back Mode

For debug purposes, the I2SMCC can be configured to loop back the transmitter to the receiver. Writing a ’1’ to the I2SMCC_MRA.RXLOOP bit internally connects I2SMCC_DOUTx to I2SMCC_DINx, so that the transmitted data is also received. Writing a ’0’ to I2SMCC_MRA.RXLOOP restores the normal behavior with independent receiver and transmitter. As for other changes to the receiver or transmitter configuration, the I2SMCC receiver and transmitter must be disabled before writing to I2SMCC_MRA to update I2SMCC_MRA.RXLOOP.

Downloaded from Arrow.com.

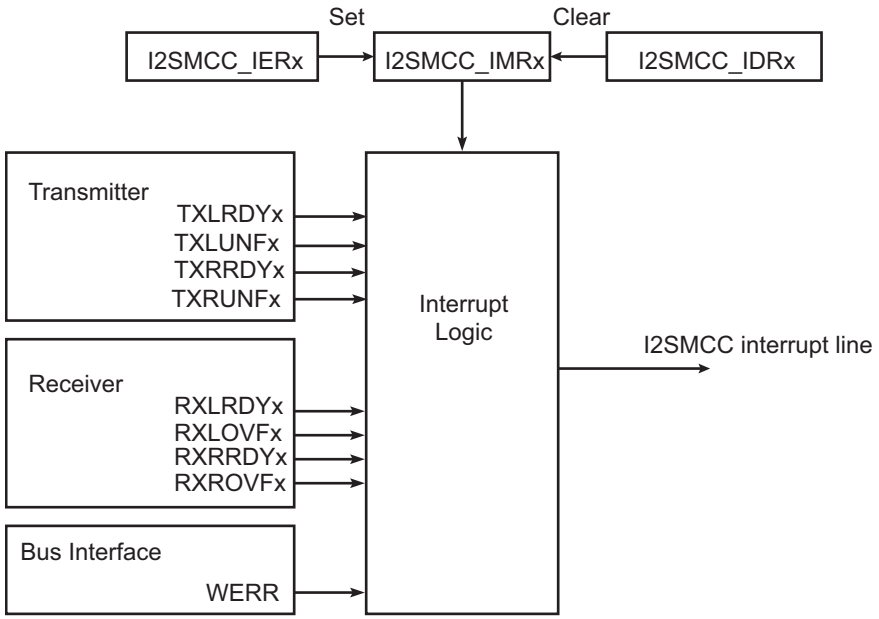
43.6.12 Interrupts

An I2SMCC interrupt request can be triggered whenever one or several of the following bits are set in I2SMCC_ISRA and/or I2SMCC_ISRB:

- Receive Left x Ready (RXLRDYx)
- Receive Right x Ready (RXRRDYx)
- Receive Left x Overrun (RXLOVFx)
- Receive Right x Overrun (RXROVFx)
- Transmit Left x Ready (TXLRDYx)
- Transmit Right x Ready (TXRRDYx)
- Transmit Left x Underrun (TXLUNFx)
- Transmit Right x Underrun (TXRUNFx)
- Write Error (WERR)

The interrupt request is generated if the corresponding bit in the Interrupt Mask registers (I2SMCC_IMRA and I2SMCC_IMRB) is set. Bits in I2SMCC_IMRx are set by writing a ’1’ to the corresponding bit in I2SMCC_IERx and cleared by writing a ’1’ to the corresponding bit in I2SMCC_IDRx. The interrupt request remains active until the corresponding bit in I2SMCC_ISRx is cleared.

Figure 43-6. Interrupt Block Diagram



43.6.13 Register Write Protection

To prevent any single software error from corrupting I2SMCC behavior, certain registers in the address space can be write-protected by setting the Write Protection Configuration Enable (WPCFEN), Write Protection Interrupt Enable (WPITEN) and/or Write Protection Control Enable (WPCTEN) bit(s) in the Write Protection Mode register (I2SMCC_WPMR).

If a write access to the protected registers is detected, the Write Protection Violation Status (WPVS) flag in the Write Protection Status register (I2SMCC_WPSR) is set and the field Write Protection Violation Source (WPVSRC) indicates the register in which the write access has been attempted. An interrupt can be raised if the Write Error (WERR) interrupt is set in I2SMCC_IMRB.

The WPVS flag is automatically reset by reading I2SMCC_WPSR.

The following register can be write-protected with the I2SMCC_WPMR.WPCFEN bit:

- Inter-IC Sound Multi Channel Controller Mode Register A
- Inter-IC Sound Multi Channel Controller Mode Register B

Downloaded from Arrow.com.

The following registers can be write-protected with the I2SMCC_WPMR.WPITEN bit:

- Inter-IC Sound Multi Channel Controller Interrupt Enable Register A
- Inter-IC Sound Multi Channel Controller Interrupt Disable Register A
- Inter-IC Sound Multi Channel Controller Interrupt Enable Register B
- Inter-IC Sound Multi Channel Controller Interrupt Disable Register B

The following register can be write-protected with the I2SMCC_WPMR.WPCTEN bit:

- Inter-IC Sound Multi Channel Controller Control Register

Downloaded from Arrow.com.

43.7 Register Summary

Offset	Name	Bit Pos.								
0x00	I2SMCC_CR	31:24								
		23:16								
		15:8								
		7:0	SWRST		TXDIS	TXEN	CKDIS	CKEN	RXDIS	RXEN
0x04	I2SMCC_MRA	31:24	IWS	IMCKMODE	ISCKDIV[5:0]					
		23:16	TDMFS[1:0]		IMCKDIV[5:0]					
		15:8	NBCHAN[2:0]			SRCCLK	TXSAME	TXMONO	RXLOOP	RXMONO
		7:0	FORMAT[1:0]		ZERO[1:0]		DATALENGTH[2:0]			MODE
0x08	I2SMCC_MRB	31:24								
		23:16								CLKSEL
		15:8							DMACHUNK[1:0]	
		7:0								
0x0C	I2SMCC_SR	31:24								
		23:16								
		15:8								
		7:0				TXEN				RXEN
0x10	I2SMCC_IERA	31:24	RXROVF3	RXLOVF3	RXROVF2	RXLOVF2	RXROVF1	RXLOVF1	RXROVF0	RXLOVF0
		23:16	RXRRDY3	RXLRDY3	RXRRDY2	RXLRDY2	RXRRDY1	RXLRDY1	RXRRDY0	RXLRDY0
		15:8	TXRUNF3	TXLUNF3	TXRUNF2	TXLUNF2	TXRUNF1	TXLUNF1	TXRUNF0	TXLUNF0
		7:0	TXRRDY3	TXLRDY3	TXRRDY2	TXLRDY2	TXRRDY1	TXLRDY1	TXRRDY0	TXLRDY0
0x14	I2SMCC_IDRA	31:24	RXROVF3	RXLOVF3	RXROVF2	RXLOVF2	RXROVF1	RXLOVF1	RXROVF0	RXLOVF0
		23:16	RXRRDY3	RXLRDY3	RXRRDY2	RXLRDY2	RXRRDY1	RXLRDY1	RXRRDY0	RXLRDY0
		15:8	TXRUNF3	TXLUNF3	TXRUNF2	TXLUNF2	TXRUNF1	TXLUNF1	TXRUNF0	TXLUNF0
		7:0	TXRRDY3	TXLRDY3	TXRRDY2	TXLRDY2	TXRRDY1	TXLRDY1	TXRRDY0	TXLRDY0
0x18	I2SMCC_IMRA	31:24	RXROVF3	RXLOVF3	RXROVF2	RXLOVF2	RXROVF1	RXLOVF1	RXROVF0	RXLOVF0
		23:16	RXRRDY3	RXLRDY3	RXRRDY2	RXLRDY2	RXRRDY1	RXLRDY1	RXRRDY0	RXLRDY0
		15:8	TXRUNF3	TXLUNF3	TXRUNF2	TXLUNF2	TXRUNF1	TXLUNF1	TXRUNF0	TXLUNF0
		7:0	TXRRDY3	TXLRDY3	TXRRDY2	TXLRDY2	TXRRDY1	TXLRDY1	TXRRDY0	TXLRDY0
0x1C	I2SMCC_ISRA	31:24	RXROVF3	RXLOVF3	RXROVF2	RXLOVF2	RXROVF1	RXLOVF1	RXROVF0	RXLOVF0
		23:16	RXRRDY3	RXLRDY3	RXRRDY2	RXLRDY2	RXRRDY1	RXLRDY1	RXRRDY0	RXLRDY0
		15:8	TXRUNF3	TXLUNF3	TXRUNF2	TXLUNF2	TXRUNF1	TXLUNF1	TXRUNF0	TXLUNF0
		7:0	TXRRDY3	TXLRDY3	TXRRDY2	TXLRDY2	TXRRDY1	TXLRDY1	TXRRDY0	TXLRDY0
0x20	I2SMCC_IERB	31:24								
		23:16								
		15:8								
		7:0								WERR
0x24	I2SMCC_IDRB	31:24								
		23:16								
		15:8								
		7:0								WERR
0x28	I2SMCC_IMRB	31:24								
		23:16								
		15:8								
		7:0								WERR
0x2C	I2SMCC_ISRB	31:24								
		23:16								
		15:8								
		7:0								WERR
0x30	I2SMCC_RHR	31:24	RHR[31:24]							
		23:16	RHR[23:16]							
		15:8	RHR[15:8]							
		7:0	RHR[7:0]							
0x34	I2SMCC_THR	31:24	THR[31:24]							
		23:16	THR[23:16]							
		15:8	THR[15:8]							
		7:0	THR[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x38 ... 0xE3	Reserved									
0xE4	I2SMCC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCTEN	WPITEN	WPCFEN
0xE8	I2SMCC_WPSR	31:24	WPVSRC[23:16]							
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

Downloaded from Arrow.com.

43.7.1 Inter-IC Sound Multi Channel Controller Control Register

Name: I2SMCC_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCTEN bit is cleared in the Inter-IC Sound Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	SWRST		TXDIS	TXEN	CKDIS	CKEN	RXDIS	RXEN
Access	W		W	W	W	W	W	W
Reset	–		–	–	–	–	–	–

Bit 7 – SWRST Software Reset

Value	Description
0	No effect.
1	Resets all the registers in the I2SMCC. The I2SMCC is disabled after the reset.

Bit 5 – TXDIS Transmitter Disable

Value	Description
0	No effect.
1	Disables the I2SMCC transmitter. Bit I2SMCC_SR.TXEN is cleared when the Transmitter is stopped.

Bit 4 – TXEN Transmitter Enable

Value	Description
0	No effect.
1	Enables the I2SMCC transmitter, if TXDIS is not ‘1’. Bit I2SMCC_SR.TXEN is set when the Transmitter is started.

Bit 3 – CKDIS Clocks Disable

Value	Description
0	No effect.
1	Disables the I2SMCC clock generation.

Bit 2 – CKEN Clocks Enable

Value	Description
0	No effect.
1	Enables the I2SMCC clock generation, if CKDIS is not ‘1’.

Bit 1 – RXDIS Receiver Disable

Downloaded from Arrow.com.

Value	Description
0	No effect.
1	Disables the I2SMCC receiver. Bit I2SMCC_SR.RXEN is cleared when the receiver is stopped.

Bit 0 – RXEN Receiver Enable

Value	Description
0	No effect.
1	Enables the I2SMCC receiver, if RXDIS is not ‘1’. Bit I2SMCC_SR.RXEN is set when the receiver is activated.

Downloaded from Arrow.com.

43.7.2 Inter-IC Sound Multi Channel Controller Mode Register A

Name: I2SMCC_MRA
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPCFEN bit is cleared in the Inter-IC Sound Write Protection Mode Register.

The I2SMCC_MRA must only be written when the I2SMCC is stopped in order to avoid unexpected behavior on the I2SMCC_WS, I2SMCC_CK and I2SMCC_DOUT outputs. The proper sequence is to write to I2SMCC_MRA, then write to I2SMCC_CR to enable the I2SMCC or to disable the I2SMCC before writing a new value to I2SMCC_MRA.

Bit	31	30	29	28	27	26	25	24
	IWS	IMCKMODE	ISCKDIV[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TDMFS[1:0]		IMCKDIV[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NBCHAN[2:0]			SRCCLK	TXSAME	TXMONO	RXLOOP	RXMONO
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FORMAT[1:0]		ZERO[1:0]		DATALENGTH[2:0]			MODE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – IWS I2SMCC_WS Slot Length
See tables Slot Length (I2S format) and Slot Length (Left-Justified format).

Value	Description
0	I2SMCC_WS slot is 32 bits long for DATALENGTH = 18/20/24 bits.
1	I2SMCC_WS slot is 24 bits long for DATALENGTH = 18/20/24 bits.

Bit 30 – IMCKMODE Master Clock Mode

Value	Description
0	No master clock generated.
1	Master clock generated.

Bits 29:24 – ISCKDIV[5:0] Selected Clock to I2SMCC Serial Clock Ratio
I2SMCC_CK Serial clock output frequency is Selected Clock divided by (2 * ISCKDIV). If ISCKDIV is 0, the I2SMCC_CK Serial clock output frequency is equal to the Selected Clock frequency.

Bits 23:22 – TDMFS[1:0] TDM Frame Synchronization

Value	Name	Description
0	SLOT	I2SMCC_WS pulse is high for one time slot at beginning of frame.
1	HALF	I2SMCC_WS pulse is high for half the time slots at beginning of frame.
2	BIT	I2SMCC_WS pulse is high for one bit period at beginning of frame, i.e., one ISCK period.

Bits 21:16 – IMCKDIV[5:0] Selected Clock to I2SMCC Master Clock Ratio
I2SMCC_MCK Master clock output frequency is Selected Clock divided by (2 * IMCKDIV). If IMCKDIV is 0, the I2SMCC_MCK Master clock output frequency is equal to the Selected Clock frequency.

Downloaded from Arrow.com.

Bits 15:13 – NBCHAN[2:0] Number of TDM Channels-1
Must be written with the number of TDM channels minus one.

Bit 12 – SRCCLK Source Clock Selection

Value	Description
0	The Peripheral clock is selected as source clock.
1	The PMC.GCLKx clock is selected as source clock.

Bit 11 – TXSAME Transmit Data when Underrun

Value	Description
0	'0' is transmitted when underrun.
1	Previous sample transmitted when underrun.

Bit 10 – TXMONO Transmit Mono

Value	Description
0	Stereo
1	Mono, with left audio samples duplicated to right audio channel by the I2SMCC.

Bit 9 – RXLOOP Loop-back Test Mode

Value	Description
0	Normal mode
1	I2SMCC_DOUT output of I2SMCC are internally connected to I2SMCC_DIN inputs.

Bit 8 – RXMONO Receive Mono

Value	Description
0	Stereo
1	Mono, with left audio samples duplicated to right audio channel by the I2SMCC.

Bits 7:6 – FORMAT[1:0] Data Format

Value	Name	Description
0	I2S	I^2S format, stereo with I2SMCC_WS low for left channel, and MSB of sample starting one I2SMCC_CK period after I2SMCC_WS edge.
1	LJ	Left-justified format, stereo with I2SMCC_WS high for left channel, and MSB of sample starting on I2SMCC_WS edge.
2	TDM	TDM format, with (NBCHAN + 1) channels, I2SMCC_WS high at beginning of first channel, and MSB of sample starting one I2SMCC_CK period after I2SMCC_WS edge.
3	TDMLJ	TDM format, left-justified, with (NBCHAN + 1) channels, I2SMCC_WS high at beginning of first channel, and MSB of sample starting on I2SMCC_WS edge.

Bits 5:4 – ZERO[1:0] Must always be written to 0

Bits 3:1 – DATALENGTH[2:0] Data Word Length

Value	Name	Description
0	32_BITS	Data length is set to 32 bits.
1	24_BITS	Data length is set to 24 bits.
2	20_BITS	Data length is set to 20 bits.
3	18_BITS	Data length is set to 18 bits.
4	16_BITS	Data length is set to 16 bits.
5	16_BITS_COMPACT	Data length is set to 16-bit compact stereo. Left sample in bits [15:0] and right sample in bits [31:16] of same word.
6	8_BITS	Data length is set to 8 bits.
7	8_BITS_COMPACT	Data length is set to 8-bit compact stereo. Left sample in bits [7:0] and right sample in bits [15:8] of the same word.

Bit 0 – MODE Inter-IC Sound Multi Channel Controller Mode

Downloaded from Arrow.com.

Value	Name	Description
0	SLAVE	I2SMCC_CK and I2SMCC_WS pin inputs used as bit clock and word select/frame synchronization.
1	MASTER	Bit clock and word select/frame synchronization generated by I2SMCC from MCK and output to I2SMCC_CK and I2SMCC_WS pins. MCK is output as master clock on I2SMCC_MCK if I2SMCC_MRA.IMCKMODE is set.

Downloaded from Arrow.com.

43.7.3 Inter-IC Sound Multi Channel Controller Mode Register B

Name: I2SMCC_MRB
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPCFEN bit is cleared in the Inter-IC Sound Write Protection Mode Register.

The I2SMCC_MRB must only be written when the I2SMCC is stopped in order to avoid unexpected behavior on the I2SMCC_WS, I2SMCC_CK and I2SMCC_DOUT outputs. The proper sequence is to write to I2SMCC_MRB, then write to I2SMCC_CR to enable the I2SMCC or to disable the I2SMCC before writing a new value to I2SMCC_MRB.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								CLKSEL
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
							DMACHUNK[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 16 – CLKSEL Serial Clock Selection

Value	Description
0	The I2SMCC_CK clock (provided by the external I2S master) is selected as serial clock.
1	The internal clock (generated from source clock) is selected as serial clock.

Bits 9:8 – DMACHUNK[1:0] DMA Chunk Size

Value	Name	Description
0	1_WORD	Each DMA transfer contains 1 word.
1	2_WORDS	Each DMA transfer contains 2 words.
2	4_WORDS	Each DMA transfer contains 4 words.
3	8_WORDS	Each DMA transfer contains 8 words.

Downloaded from Arrow.com.

43.7.4 Inter-IC Sound Multi Channel Controller Status Register

Name: I2SMCC_SR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				TXEN				RXEN
Access				R				R
Reset				0				0

Bit 4 – TXEN Transmitter Enabled

Value	Description
0	Cleared when the transmitter is disabled, following a I2SMCC_CR.TXDIS or I2SMCC_CR.SWRST request.
1	Set when the transmitter is enabled, following a I2SMCC_CR.TXEN request.

Bit 0 – RXEN Receiver Enabled

Value	Description
0	Cleared when the receiver is disabled, following a RXDIS or SWRST request in I2SMCC_CR.
1	Set when the receiver is enabled, following a RXEN request in I2SMCC_CR.

Downloaded from Arrow.com.

43.7.5 Inter-IC Sound Multi Channel Controller Interrupt Enable Register A

Name: I2SMCC_IERA
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the Inter-IC Sound Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	RXROVF3	RXLOVF3	RXROVF2	RXLOVF2	RXROVF1	RXLOVF1	RXROVF0	RXLOVF0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	RXRRDY3	RXLRDY3	RXRRDY2	RXLRDY2	RXRRDY1	RXLRDY1	RXRRDY0	RXLRDY0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TXRUNF3	TXLUNF3	TXRUNF2	TXLUNF2	TXRUNF1	TXLUNF1	TXRUNF0	TXLUNF0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TXRRDY3	TXLRDY3	TXRRDY2	TXLRDY2	TXRRDY1	TXLRDY1	TXRRDY0	TXLRDY0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 25, 27, 29, 31 – RXROVFx I2S Receive Right x (x=0 only) or TDM Channel [2x]+1 Overrun Interrupt Enable

Bits 24, 26, 28, 30 – RXLOVFx I2S Receive Left x (x=0 only) or TDM Channel 2x Overrun Interrupt Enable

Bits 17, 19, 21, 23 – RXRRDYx I2S Receive Right x (x=0 only) or TDM Channel [2x]+1 Ready Interrupt Enable

Bits 16, 18, 20, 22 – RXLRDYx I2S Receive Left x (x=0 only) or TDM Channel 2x Ready Interrupt Enable

Bits 9, 11, 13, 15 – TXRUNFx I2S Transmit Right x (x=0 only) or TDM Channel [2x]+1 Underrun Interrupt Enable

Bits 8, 10, 12, 14 – TXLUNFx I2S Transmit Left x (x=0 only) or TDM Channel 2x Underrun Interrupt Enable

Bits 1, 3, 5, 7 – TXRRDYx I2S Transmit Right x (x=0 only) or TDM Channel [2x]+1 Ready Interrupt Enable

Bits 0, 2, 4, 6 – TXLRDYx I2S Transmit Left x (x=0 only) or TDM Channel 2x Ready Interrupt Enable

Downloaded from Arrow.com.

43.7.6 Inter-IC Sound Multi Channel Controller Interrupt Disable Register A

Name: I2SMCC_IDRA
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the Inter-IC Sound Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	RXROVF3	RXLOVF3	RXROVF2	RXLOVF2	RXROVF1	RXLOVF1	RXROVF0	RXLOVF0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	RXRRDY3	RXLRDY3	RXRRDY2	RXLRDY2	RXRRDY1	RXLRDY1	RXRRDY0	RXLRDY0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TXRUNF3	TXLUNF3	TXRUNF2	TXLUNF2	TXRUNF1	TXLUNF1	TXRUNF0	TXLUNF0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TXRRDY3	TXLRDY3	TXRRDY2	TXLRDY2	TXRRDY1	TXLRDY1	TXRRDY0	TXLRDY0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 25, 27, 29, 31 – RXROVFx I2S Receive Right x (x=0 only) or TDM Channel [2x]+1 Overrun Interrupt Disable

Bits 24, 26, 28, 30 – RXLOVFx I2S Receive Left x (x=0 only) or TDM Channel 2x Overrun Interrupt Disable

Bits 17, 19, 21, 23 – RXRRDYx I2S Receive Right x (x=0 only) or TDM Channel [2x]+1 Ready Interrupt Disable

Bits 16, 18, 20, 22 – RXLRDYx I2S Receive Left x (x=0 only) or TDM Channel 2x Ready Interrupt Disable

Bits 9, 11, 13, 15 – TXRUNFx I2S Transmit Right x (x=0 only) or TDM Channel [2x]+1 Underrun Interrupt Disable

Bits 8, 10, 12, 14 – TXLUNFx I2S Transmit Left x (x=0 only) or TDM Channel 2x Underrun Interrupt Disable

Bits 1, 3, 5, 7 – TXRRDYx I2S Transmit Right x (x=0 only) or TDM Channel [2x]+1 Ready Interrupt Disable

Bits 0, 2, 4, 6 – TXLRDYx I2S Transmit Left x (x=0 only) or TDM Channel 2x Ready Interrupt Disable

Downloaded from Arrow.com.

43.7.7 Inter-IC Sound Multi Channel Controller Interrupt Mask Register A

Name: I2SMCC_IMRA
Offset: 0x18
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding source of interrupt is disabled.

1: The corresponding source of interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
	RXROVF3	RXLOVF3	RXROVF2	RXLOVF2	RXROVF1	RXLOVF1	RXROVF0	RXLOVF0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RXRRDY3	RXLRDY3	RXRRDY2	RXLRDY2	RXRRDY1	RXLRDY1	RXRRDY0	RXLRDY0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TXRUNF3	TXLUNF3	TXRUNF2	TXLUNF2	TXRUNF1	TXLUNF1	TXRUNF0	TXLUNF0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXRRDY3	TXLRDY3	TXRRDY2	TXLRDY2	TXRRDY1	TXLRDY1	TXRRDY0	TXLRDY0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 25, 27, 29, 31 – RXROVFx I2S Receive Right x (x=0 only) or TDM Channel [2x]+1 Overrun Interrupt Mask

Bits 24, 26, 28, 30 – RXLOVFx I2S Receive Left x (x=0 only) or TDM Channel 2x Overrun Interrupt Mask

Bits 17, 19, 21, 23 – RXRRDYx I2S Receive Right x (x=0 only) or TDM Channel [2x]+1 Ready Interrupt Mask

Bits 16, 18, 20, 22 – RXLRDYx I2S Receive Left x (x=0 only) or TDM Channel 2x Ready Interrupt Mask

Bits 9, 11, 13, 15 – TXRUNFx I2S Transmit Right x (x=0 only) or TDM Channel [2x]+1 Underrun Interrupt Mask

Bits 8, 10, 12, 14 – TXLUNFx I2S Transmit Left x (x=0 only) or TDM Channel 2x Underrun Interrupt Mask

Bits 1, 3, 5, 7 – TXRRDYx I2S Transmit Right x (x=0 only) or TDM Channel [2x]+1 Ready Interrupt Mask

Bits 0, 2, 4, 6 – TXLRDYx I2S Transmit Left x (x=0 only) or TDM Channel 2x Ready Interrupt Mask

Downloaded from Arrow.com.

43.7.8 Inter-IC Sound Multi Channel Controller Interrupt Status Register A

Name: I2SMCC_ISRA
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RXROVF3	RXLOVF3	RXROVF2	RXLOVF2	RXROVF1	RXLOVF1	RXROVF0	RXLOVF0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RXRRDY3	RXLRDY3	RXRRDY2	RXLRDY2	RXRRDY1	RXLRDY1	RXRRDY0	RXLRDY0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TXRUNF3	TXLUNF3	TXRUNF2	TXLUNF2	TXRUNF1	TXLUNF1	TXRUNF0	TXLUNF0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXRRDY3	TXLRDY3	TXRRDY2	TXLRDY2	TXRRDY1	TXLRDY1	TXRRDY0	TXLRDY0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 25, 27, 29, 31 – RXROVFx I2S Receive Right x (x=0 only) or TDM Channel [2x]+1 Overrun Flag (Cleared on read)

Value	Description
0	Cleared when I2SMCC_ISRA is read.
1	Set when an overrun error occurs in I2SMCC_RHR.

Bits 24, 26, 28, 30 – RXLOVFx I2S Receive Left x (x=0 only) or TDM Channel 2x Overrun Flag (Cleared on read)

Value	Description
0	Cleared when I2SMCC_ISRA is read.
1	Set when an overrun error occurs in I2SMCC_RHR.

Bits 17, 19, 21, 23 – RXRRDYx I2S Receive Right x (x=0 only) or TDM Channel [2x]+1 Ready Flag (Cleared by reading I2SMCC_RHR)

Value	Description
0	Cleared when the corresponding data has been read through I2SMCC_RHR.
1	Set when received data is available in I2SMCC_RHR.

Bits 16, 18, 20, 22 – RXLRDYx I2S Receive Left x (x=0 only) or TDM Channel 2x Ready Flag (Cleared by reading I2SMCC_RHR)

Value	Description
0	Cleared when the corresponding data has been read in I2SMCC_RHR.
1	Set when received data is available in I2SMCC_RHR.

Bits 9, 11, 13, 15 – TXRUNFx I2S Transmit Right x (x=0 only) or TDM Channel [2x]+1 Underrun Flag (Cleared on read)

Value	Description
0	Cleared when the I2SMCC_ISRA is read.
1	Set when an underrun error occurs in I2SMCC_THR.

Downloaded from Arrow.com.

Bits 8, 10, 12, 14 – TXLUNFx I2S Transmit Left x (x=0 only) or TDM Channel 2x Underrun (Cleared on read)

Value	Description
0	Cleared when I2SMCC_ISRA is read.
1	Set when an underrun error occurs in I2SMCC_THR.

Bits 1, 3, 5, 7 – TXRRDYx I2S Transmit Right x (x=0 only) or TDM Channel [2x]+1 Ready Flag (Cleared by writing I2SMCC_THR)

Value	Description
0	Cleared the corresponding data has been written in I2SMCC_THR.
1	Set when I2SMCC_THR is empty.

Bits 0, 2, 4, 6 – TXLRDYx I2S Transmit Left x (x=0 only) or TDM Channel 2x Ready Flag (Cleared by writing I2SMCC_THR)

Value	Description
0	Cleared when the corresponding data has been written in I2SMCC_THR.
1	Set when I2SMCC_THR is empty.

Downloaded from Arrow.com.

43.7.9 Inter-IC Sound Multi Channel Controller Interrupt Enable Register B

Name: I2SMCC_IERB
Offset: 0x20
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the Inter-IC Sound Write Protection Mode Register.

The following configuration values are valid for the listed bit of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								WERR
Reset								W
								–

Bit 0 – WERR Write Error Interrupt Enable

Downloaded from Arrow.com.

43.7.10 Inter-IC Sound Multi Channel Controller Interrupt Disable Register B

Name: I2SMCC_IDRB
Offset: 0x24
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the Inter-IC Sound Write Protection Mode Register.

The following configuration values are valid for the listed bit of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								WERR
Reset								W
								–

Bit 0 – WERR Write Error Interrupt Disable

Downloaded from Arrow.com.

43.7.11 Inter-IC Sound Multi Channel Controller Interrupt Mask Register B

Name: I2SMCC_IMRB
Offset: 0x28
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for the listed bit of this register:

0: The corresponding source of interrupt is disabled.

1: The corresponding source of interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								WERR
Access								R
Reset								0

Bit 0 – WERR Write Error Interrupt Mask

Downloaded from Arrow.com.

43.7.12 Inter-IC Sound Multi Channel Controller Interrupt Status Register B

Name: I2SMCC_ISRB
Offset: 0x2C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								WERR
Access								R
Reset								0

Bit 0 – WERR Write Error Flag (Cleared on read)

Value	Description
0	Cleared when the I2SMCC_ISRB is read.
1	Set when a write occurs in a protected register.

Downloaded from Arrow.com.

43.7.13 Inter-IC Sound Multi Channel Controller Receiver Holding Register

Name: I2SMCC_RHR
Offset: 0x30
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RHR[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RHR[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RHR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RHR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RHR[31:0] Receiver Holding Register
Set by hardware to the last received data word. If I2SMCC_MRA.DATALENGTH specifies fewer than 32 bits, data is right justified in the RHR field.

Downloaded from Arrow.com.

43.7.14 Inter-IC Sound Multi Channel Controller Transmitter Holding Register

Name: I2SMCC_THR
Offset: 0x34
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	THR[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	THR[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	THR[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	THR[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – THR[31:0] Transmitter Holding Register
Next data word to be transmitted after the current word if TXLRDYx or TXRRDYx is not set. If I2SMCC_MRA.DATALENGTH specifies fewer than 32 bits, data is right-justified in the THR field.

Downloaded from Arrow.com.

43.7.15 Inter-IC Sound Write Protection Mode Register

Name: I2SMCC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						WPCTEN	WPITEN	WPCFEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x493253	PASSWD	Writing any other value in this field aborts the write operation. Always reads as 0.

Bit 2 – WPCTEN Write Protection Control Enable

Value	Description
0	Disables the write protection of the control if WPKEY matches to 0x493253 (I2S in ASCII).
1	Enables the write protection of the control if WPKEY matches to 0x493253 (I2S in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection of the interruption if WPKEY matches to 0x493253 (I2S in ASCII).
1	Enables the write protection of the interruption if WPKEY matches to 0x493253 (I2S in ASCII).

Bit 0 – WPCFEN Write Protection Configuration Enable

Value	Description
0	Disables the write protection of the configuration if WPKEY matches to 0x493253 (I2S in ASCII).
1	Enables the write protection of the configuration if WPKEY matches to 0x493253 (I2S in ASCII).

Downloaded from Arrow.com.

43.7.16 Inter-IC Sound Write Protection Status Register

Name: I2SMCC_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	WPVSRC[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 31:8 – WPVSRC[23:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the I2SMCC_WPSR.
1	A write protection violation has occurred since the last read of the I2SMCC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

44. Synchronous Serial Controller (SSC)

44.1 Description

The Synchronous Serial Controller (SSC) provides a synchronous communication link with external devices. It supports many serial synchronous communication protocols generally used in audio and telecom applications such as I2S, Short Frame Sync, Long Frame Sync, etc.

The SSC contains an independent receiver and transmitter and a common clock divider. The receiver and the transmitter each interface with three signals: the TD/RD signal for data, the TK/RK signal for the clock and the TF/RF signal for the Frame Sync. The transfers can be programmed to start automatically or on different events detected on the Frame Sync signal.

The SSC high-level of programmability and its use of DMA enable a continuous high bit rate data transfer without processor intervention.

Featuring connection to the DMA, the SSC enables interfacing with low processor overhead to:

- Codecs in Master or Slave mode
- DAC through dedicated serial interface, particularly I2S
- Magnetic card reader

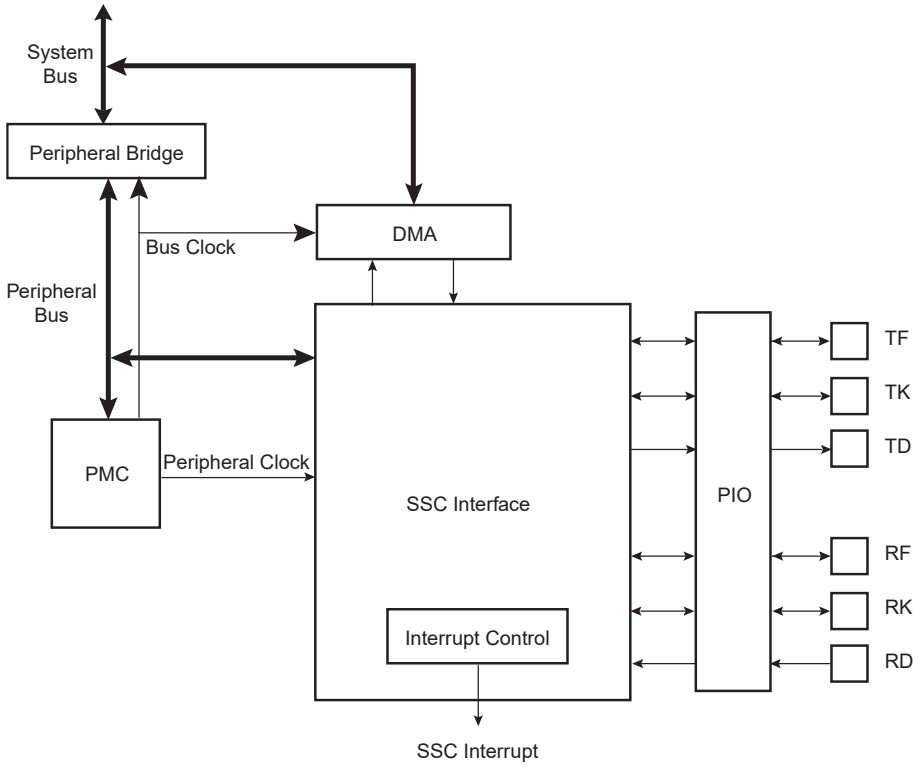
44.2 Embedded Characteristics

- Provides Serial Synchronous Communication Links Used in Audio and Telecom Applications
- Contains an Independent Receiver and Transmitter and a Common Clock Divider
- Interfaced with the DMA Controller (DMAC) to Reduce Processor Overhead
- Offers a Configurable Frame Sync and Data Length
- Receiver and Transmitter Can be Programmed to Start Automatically or on Detection of Different Events on the Frame Sync Signal
- Receiver and Transmitter Include a Data Signal, a Clock Signal and a Frame Sync Signal

Downloaded from Arrow.com.

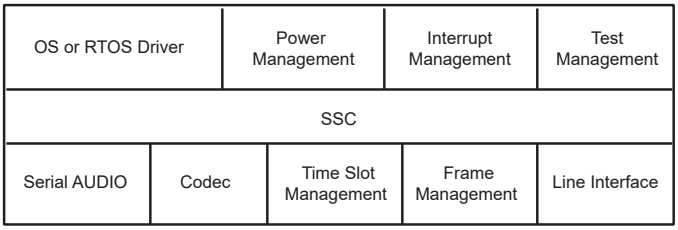
44.3 Block Diagram

Figure 44-1. SSC Block Diagram



44.4 Application Block Diagram

Figure 44-2. Application Block Diagram



44.5 SSC Application Examples

The SSC can support several serial communication modes used in audio or high speed serial links. Some standard applications are shown in the following figures. All serial link applications supported by the SSC are not listed here.

Downloaded from Arrow.com.

Figure 44-3. Audio Application Block Diagram

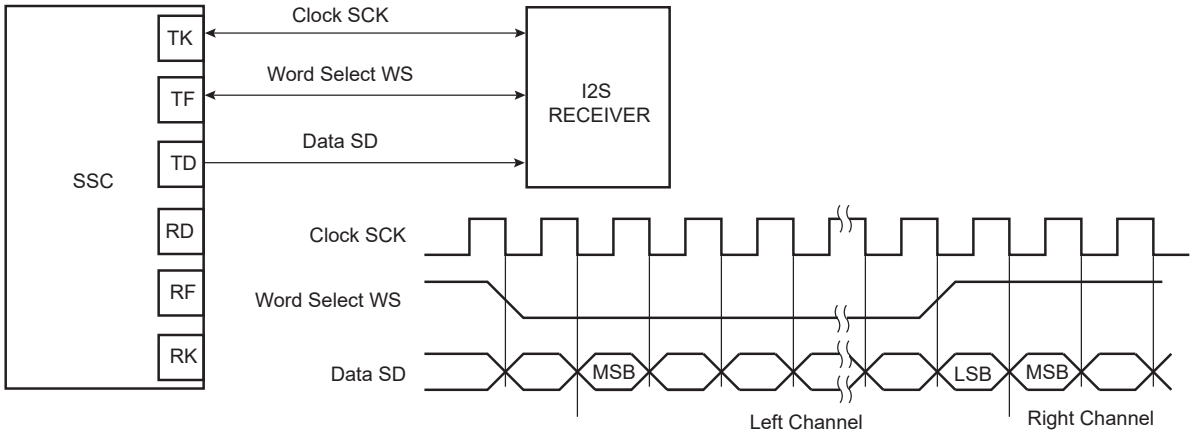
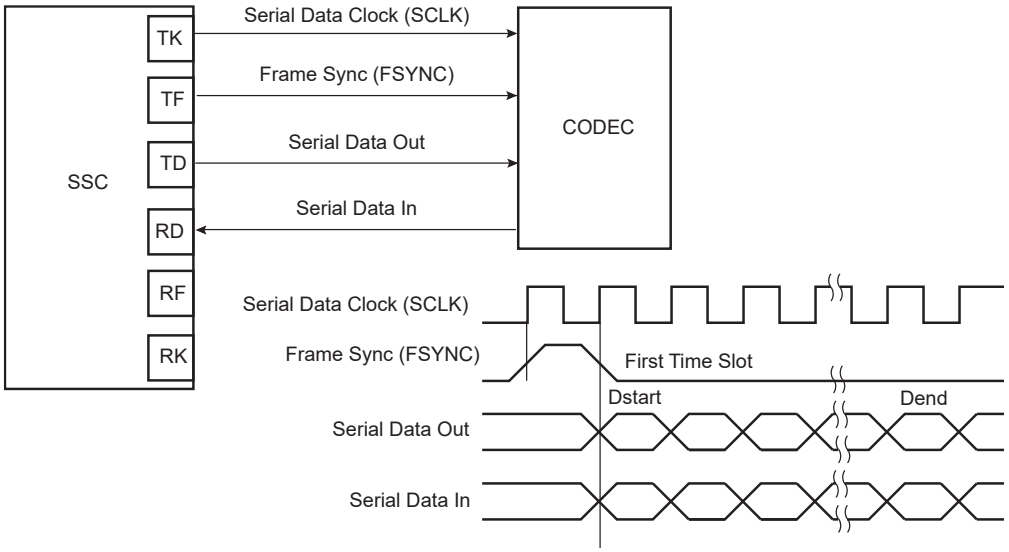
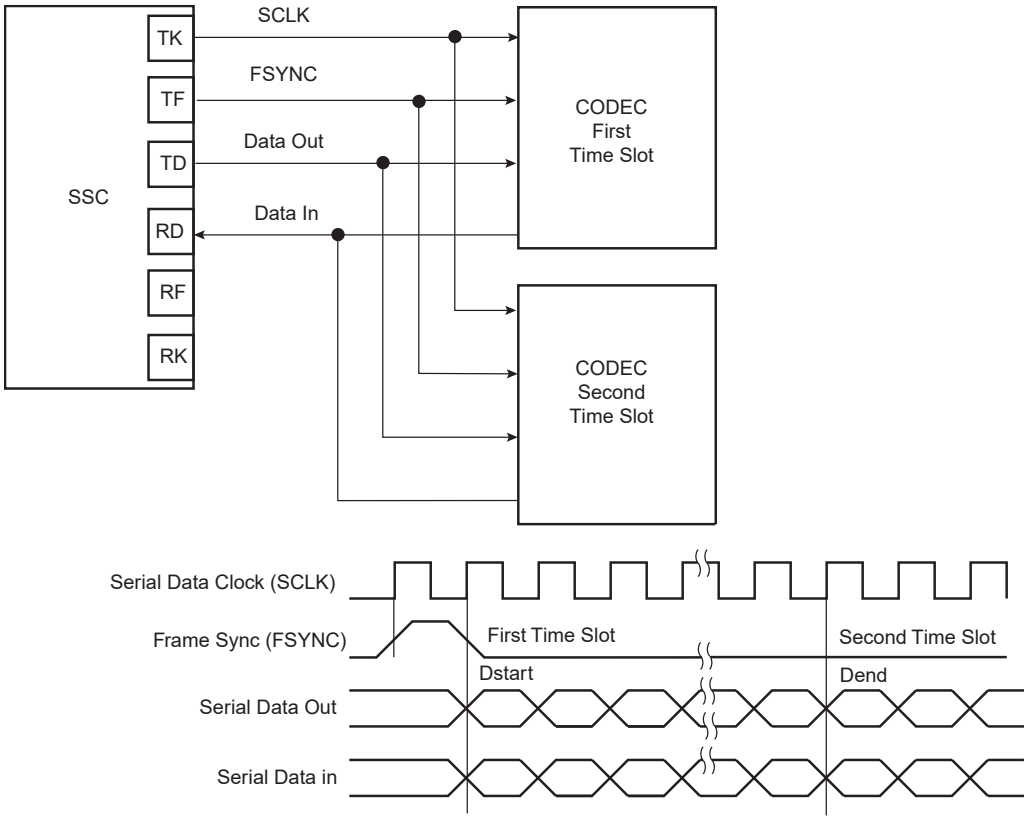


Figure 44-4. Codec Application Block Diagram



Downloaded from Arrow.com.

Figure 44-5. Time Slot Application Block Diagram



44.6 Pin Name List

Table 44-1. I/O Lines Description

Pin Name	Pin Description	Type
RF	Receive Frame Synchronization	Input/Output
RK	Receive Clock	Input/Output
RD	Receive Data	Input
TF	Transmit Frame Synchronization	Input/Output
TK	Transmit Clock	Input/Output
TD	Transmit Data	Output

44.7 Product Dependencies

44.7.1 I/O Lines

The pins used for interfacing the compliant external devices may be multiplexed with PIO lines.

Before using the SSC receiver, the PIO controller must be configured to dedicate the SSC receiver I/O lines to the SSC Peripheral mode.

Before using the SSC transmitter, the PIO controller must be configured to dedicate the SSC transmitter I/O lines to the SSC Peripheral mode.

Downloaded from Arrow.com.

44.7.2 Power Management

The SSC is not continuously clocked. The SSC interface may be clocked through the Power Management Controller (PMC), therefore the programmer must first configure the PMC to enable the SSC clock.

44.7.3 Interrupt

The SSC interface has an interrupt line connected to the interrupt controller. Handling interrupts requires programming the interrupt controller before configuring the SSC.

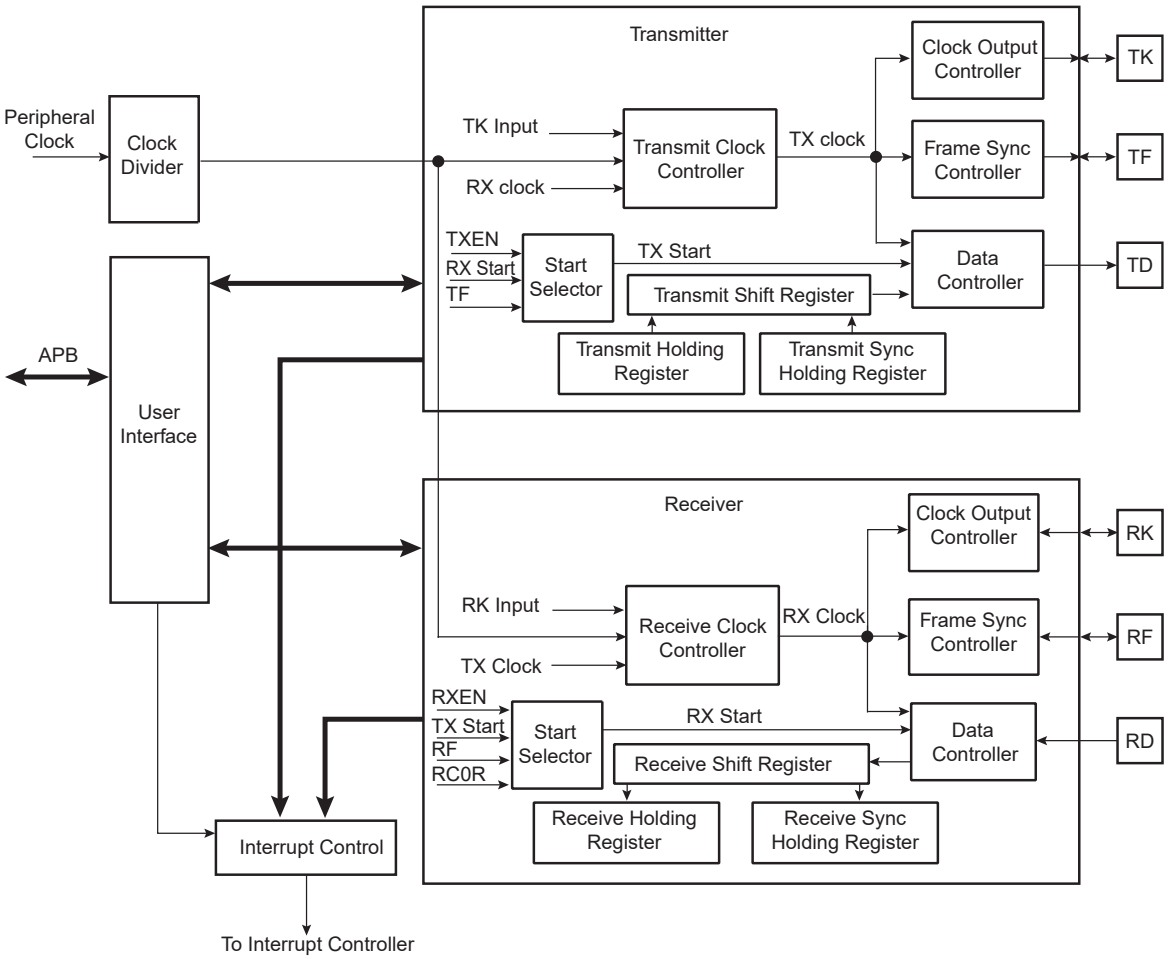
All SSC interrupts can be enabled/disabled configuring the SSC Interrupt Mask Register. Each pending and unmasked SSC interrupt asserts the SSC interrupt line. The SSC interrupt service routine can get the interrupt origin by reading the SSC Interrupt Status Register.

44.8 Functional Description

This section contains the functional description of the following: SSC Functional Block, Clock Management, Data Format, Start, Transmit, Receive and Frame Synchronization.

The receiver and transmitter operate separately. However, they can work synchronously by programming the receiver to use the transmit clock and/or to start a data transfer when transmission starts. Alternatively, this can be done by programming the transmitter to use the receive clock and/or to start a data transfer when reception starts. The transmitter and the receiver can be programmed to operate with the clock signals provided on either the TK or RK pins. This allows the SSC to support many Slave mode data transfers. The maximum clock speed allowed on the TK and RK pins is the peripheral clock divided by 2.

Figure 44-6. SSC Functional Block Diagram



Downloaded from Arrow.com.

44.8.1 Clock Management

The transmit clock can be generated by:

- an external clock received on the TK I/O pad
- the receive clock
- the internal clock divider

The receive clock can be generated by:

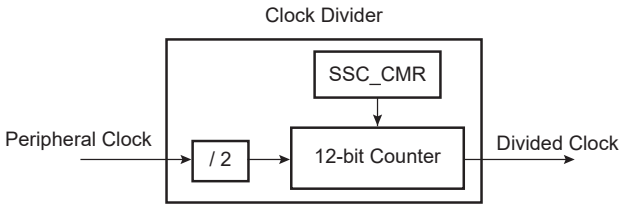
- an external clock received on the RK I/O pad
- the transmit clock
- the internal clock divider

Furthermore, the transmitter block can generate an external clock on the TK I/O pad, and the receive block can generate an external clock on the RK I/O pad.

This allows the SSC to support many Master and Slave mode data transfers.

44.8.1.1 Clock Divider

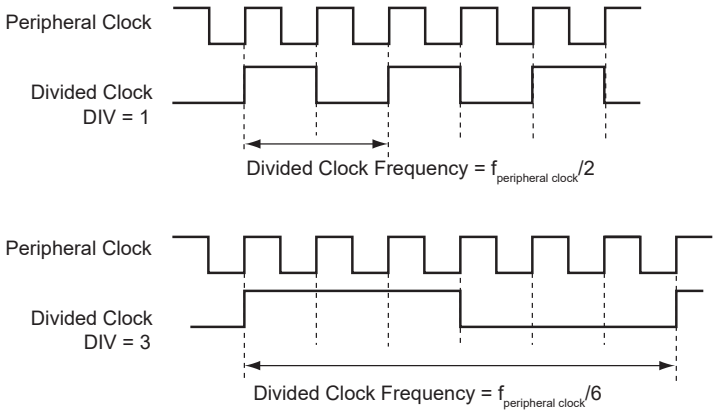
Figure 44-7. Divided Clock Block Diagram



The peripheral clock divider is determined by the 12-bit field DIV counter and comparator (so its maximal value is 4095) in the Clock Mode Register (SSC_CMR), allowing a peripheral clock division by up to 8190. The Divided Clock is provided to both the receiver and the transmitter. When this field is programmed to 0, the Clock Divider is not used and remains inactive.

When DIV is set to a value equal to or greater than 1, the Divided Clock has a frequency of peripheral clock divided by 2 times DIV. Each level of the Divided Clock has a duration of the peripheral clock multiplied by DIV. This ensures a 50% duty cycle for the Divided Clock regardless of whether the DIV value is even or odd.

Figure 44-8. Divided Clock Generation



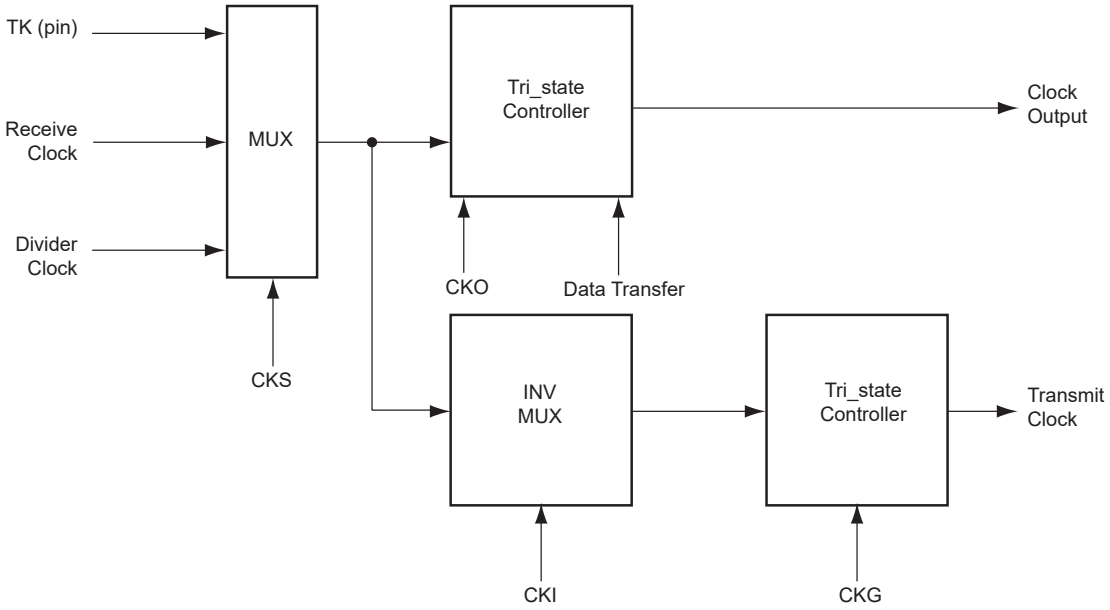
44.8.1.2 Transmit Clock Management

The transmit clock is generated from the receive clock or the divider clock or an external clock scanned on the TK I/O pad. The transmit clock is selected by the CKS field in the Transmit Clock Mode Register (SSC_TCMR). Transmit Clock can be inverted independently by the CKI bits in the SSC_TCMR.

The transmitter can also drive the TK I/O pad continuously or be limited to the current data transfer. The clock output is configured by the SSC_TCMR. The Transmit Clock Inversion (CKI) bits have no effect on the clock outputs. Programming the SSC_TCMR to select TK pin (CKS field) and at the same time Continuous Transmit Clock (CKO field) can lead to unpredictable results.

Downloaded from Arrow.com.

Figure 44-9. Transmit Clock Management

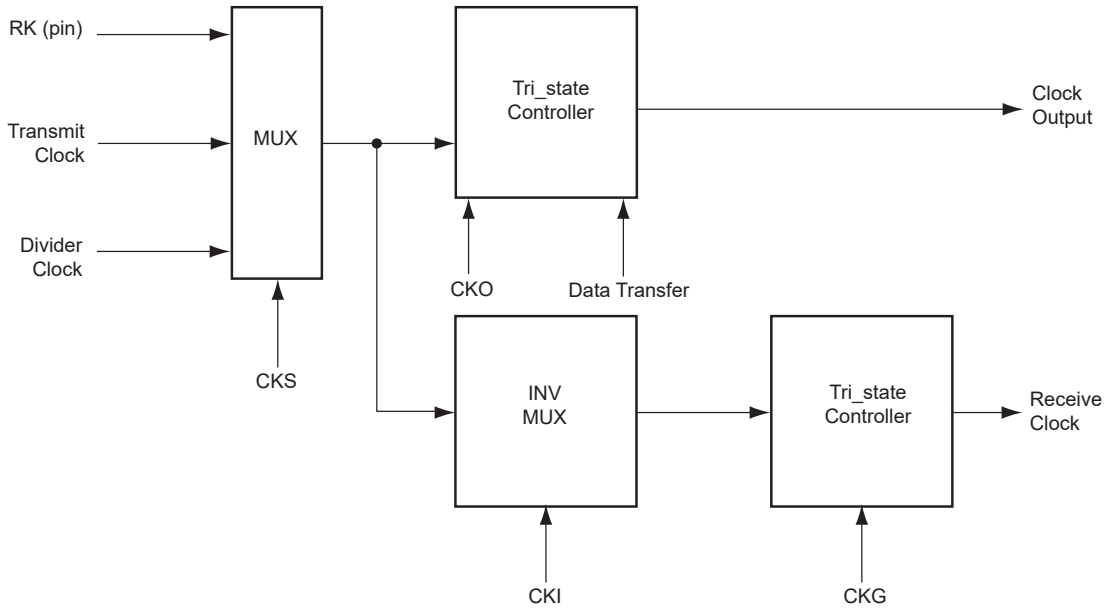


44.8.1.3 Receive Clock Management

The receive clock is generated from the transmit clock or the divider clock or an external clock scanned on the RK I/O pad. The Receive Clock is selected by the CKS field in SSC_RCMR (Receive Clock Mode Register). Receive Clocks can be inverted independently by the CKI bits in SSC_RCMR.

The receiver can also drive the RK I/O pad continuously or be limited to the current data transfer. The clock output is configured by the SSC_RCMR. The Receive Clock Inversion (CKI) bits have no effect on the clock outputs. Programming the SSC_RCMR to select RK pin (CKS field) and at the same time Continuous Receive Clock (CKO field) can lead to unpredictable results.

Figure 44-10. Receive Clock Management



44.8.1.4 Serial Clock Ratio Considerations

The transmitter and the receiver can be programmed to operate with the clock signals provided on either the TK or RK pins. This allows the SSC to support many Slave mode data transfers. In this case, the maximum clock speed allowed on the RK pin is:

Downloaded from Arrow.com.

- Peripheral clock divided by 2 if Receive Frame Synchronization is input
- Peripheral clock divided by 3 if Receive Frame Synchronization is output

In addition, the maximum clock speed allowed on the TK pin is:

- Peripheral clock divided by 6 if Transmit Frame Synchronization is input
- Peripheral clock divided by 2 if Transmit Frame Synchronization is output

These are only theoretical speed limits for first order calculations. Exact speed limits on TK and RK are provided in the "Electrical Characteristics" chapter.

44.8.2 Transmit Operations

A transmit frame is triggered by a start event and can be followed by synchronization data before data transmission.

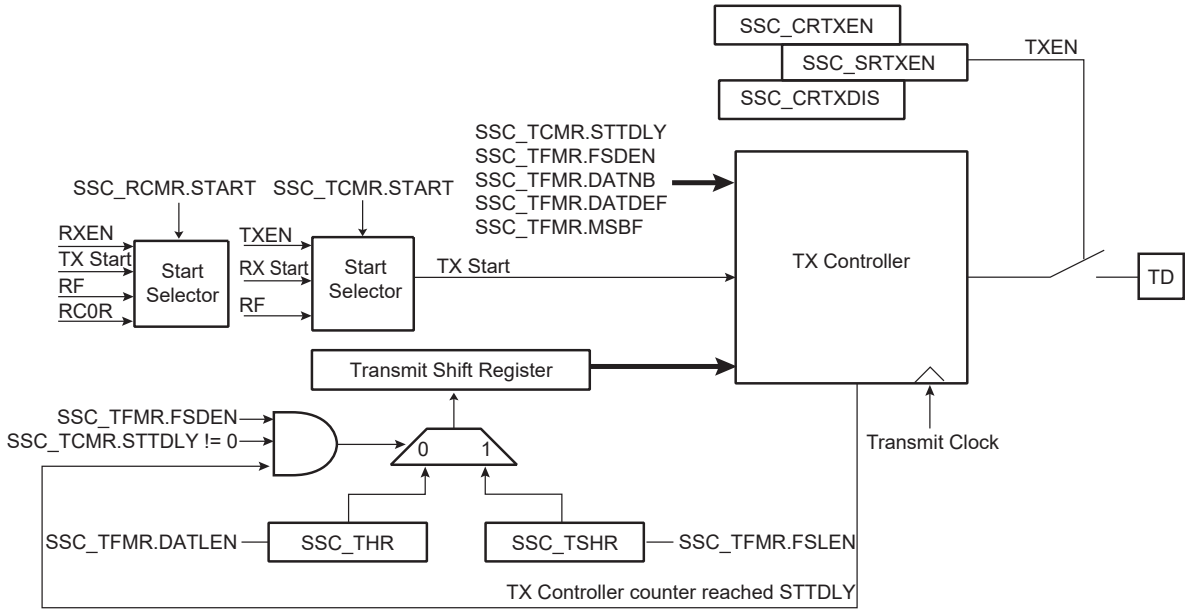
The start event is configured by setting the SSC_TCMR. See Start.

The frame synchronization is configured setting the Transmit Frame Mode Register (SSC_TFMR). See Frame Synchronization.

To transmit data, the transmitter uses a shift register clocked by the transmit clock signal and the start mode selected in the SSC_TCMR. Data is written by the application to the Transmit Holding register (SSC_THR) then transferred to the transmit shift register according to the data format selected.

When both the SSC_THR and the transmit shift register are empty, the status flag TXEMPTY is set in the Status register (SSC_SR). When the Transmit Holding register is transferred in the transmit shift register, the status flag TXRDY is set in the SSC_SR and additional data can be loaded in the Transmit Holding register.

Figure 44-11. Transmit Block Diagram



44.8.3 Receive Operations

A receive frame is triggered by a start event and can be followed by synchronization data before data transmission.

The start event is configured by setting the Receive Clock Mode Register (SSC_RCMR). See Start.

The frame synchronization is configured by setting the Receive Frame Mode Register (SSC_RFMR). See Frame Synchronization.

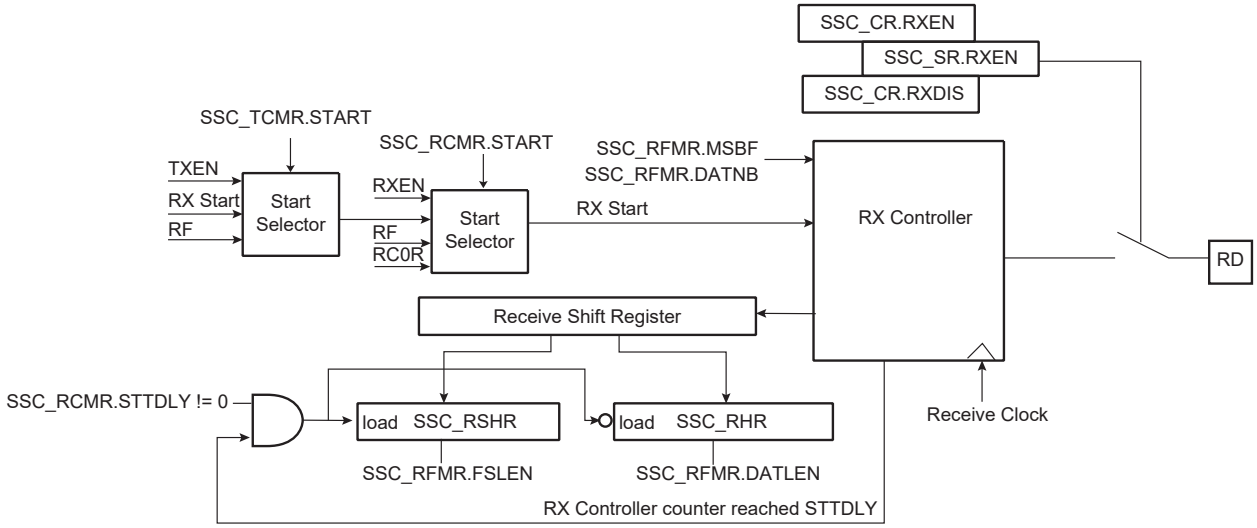
The receiver uses a shift register clocked by the receive clock signal and the start mode selected in the SSC_RCMR. The data is transferred from the shift register depending on the data format selected.

When the receive shift register is full, the SSC transfers the data into the Receive Holding register (SSC_RHR), the status flag RXRDY is set in the SSC_SR and the data can be read in the Receive Holding register. If another transfer

Downloaded from Arrow.com.

occurs before read of the Receive Holding register, the status flag OVRUN is set in the SSC_SR and the receive shift register is transferred in the SSC_RHR. The old unread data is then lost.

Figure 44-12. Receive Block Diagram



44.8.4 Start

The transmitter and receiver can both be programmed to start their operations when an event occurs, respectively in the Transmit Start Selection (START) field of SSC_TCMR and in the Receive Start Selection (START) field of SSC_RCMR.

Under the following conditions the start event is independently programmable:

- Continuous. In this case, the transmission starts as soon as a word is written in SSC_THR and the reception starts as soon as the receiver is enabled.
- Synchronously with the transmitter/receiver
- On detection of a falling/rising edge on TF/RF
- On detection of a low level/high level on TF/RF
- On detection of a level change or an edge on TF/RF

A start can be programmed in the same manner on either side of the Transmit/Receive Clock Register (SSC_RCMR/SSC_TCMR). Thus, the start could be on TF (Transmit) or RF (Receive).

Moreover, the receiver can start when data is detected in the bit stream with the Compare Functions.

Detection on TF/RF input/output is done by the field FSOS of the Transmit/Receive Frame Mode Register (SSC_TFMR/SSC_RFMR).

Downloaded from Arrow.com.

Figure 44-13. Transmit Start Mode

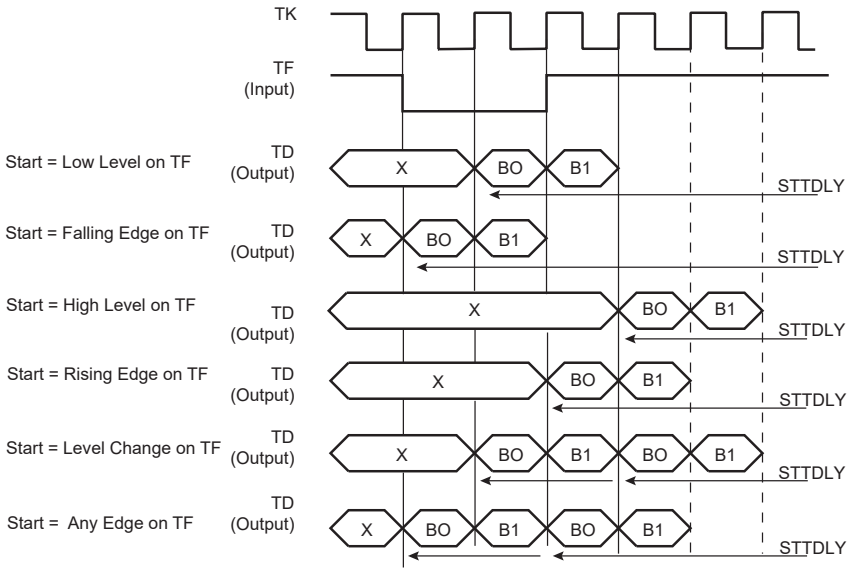
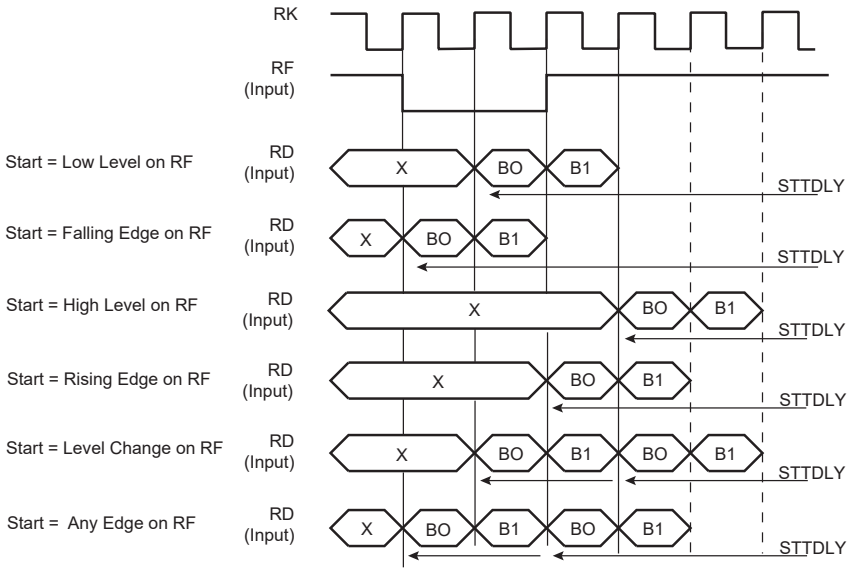


Figure 44-14. Receive Pulse/Edge Start Modes



44.8.5 Frame Synchronization

The Transmit and Receive Frame Sync pins, TF and RF, can be programmed to generate different kinds of Frame Sync signals. The Frame Sync Output Selection (FSOS) field in the Receive Frame Mode Register (SSC_RFMR) and in the Transmit Frame Mode Register (SSC_TFMR) are used to select the required waveform.

- Programmable low or high levels during data transfer are supported.
- Programmable high levels before the start of data transfers or toggling are also supported.

If a pulse waveform is selected, the Frame Sync Length (FSLEN) field in SSC_RFMR and SSC_TFMR programs the length of the pulse, from 1 bit time up to 256 bit times.

The periodicity of the Receive and Transmit Frame Sync pulse output can be programmed through the Period Divider Selection (PERIOD) field in SSC_RCMR and SSC_TCMR.

44.8.5.1 Frame Sync Data

Frame Sync Data transmits or receives a specific tag during the Frame Sync signal.

Downloaded from Arrow.com.

During the Frame Sync signal, the receiver can sample the RD line and store the data in the Receive Sync Holding Register and the transmitter can transfer Transmit Sync Holding Register in the shift register. The data length to be sampled/shifted out during the Frame Sync signal is programmed by the FSLEN field in SSC_RFMR/SSC_TFMR and has a maximum value of 256.

Concerning the Receive Frame Sync Data operation, if the Frame Sync Length is equal to or lower than the delay between the start event and the current data reception, the data sampling operation is performed in the Receive Sync Holding Register through the receive shift register.

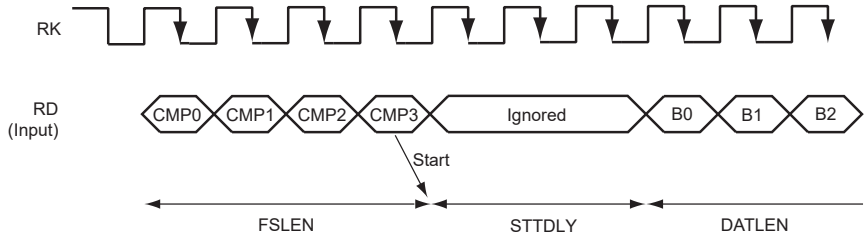
The Transmit Frame Sync Operation is performed by the transmitter only if the bit Frame Sync Data Enable (FSDEN) in SSC_TFMR is set. If the Frame Sync length is equal to or lower than the delay between the start event and the current data transmission, the normal transmission has priority and the data contained in the Transmit Sync Holding Register is transferred in the Transmit Register, then shifted out.

44.8.5.2 Frame Sync Edge Detection

The Frame Sync Edge detection is programmed by the FSEDGE field in SSC_RFMR/SSC_TFMR. This sets the corresponding flags RXSYN/TXSYN in the SSC Status Register (SSC_SR) on Frame Sync Edge detection (signals RF/TF).

44.8.6 Receive Compare Modes

Figure 44-15. Receive Compare Modes



44.8.6.1 Compare Functions

The length of the comparison patterns (Compare 0, Compare 1) and thus the number of bits they are compared to is defined by FSLEN, but with a maximum value of 256 bits. Comparison is always done by comparing the last bits received with the comparison pattern. Compare 0 can be one start event of the receiver. In this case, the receiver compares at each new sample the last bits received at the Compare 0 pattern contained in the Compare 0 Register (SSC_RC0R). When this start event is selected, the user can program the receiver to start a new data transfer either by writing a new Compare 0, or by receiving continuously until Compare 1 occurs. This selection is done with the STOP bit in the SSC_RCMR.

44.8.7 Data Format

The data framing format of both the transmitter and the receiver are programmable through the Transmitter Frame Mode Register (SSC_TFMR) and the Receive Frame Mode Register (SSC_RFMR). In either case, the user can independently select the following parameters:

- Event that starts the data transfer (START)
- Delay in number of bit periods between the start event and the first data bit (STTDLY)
- Length of the data (DATLEN)
- Number of data to be transferred for each start event (DATNB)
- Length of synchronization transferred for each start event (FSLEN)
- Bit sense: most or least significant bit first (MSBF)

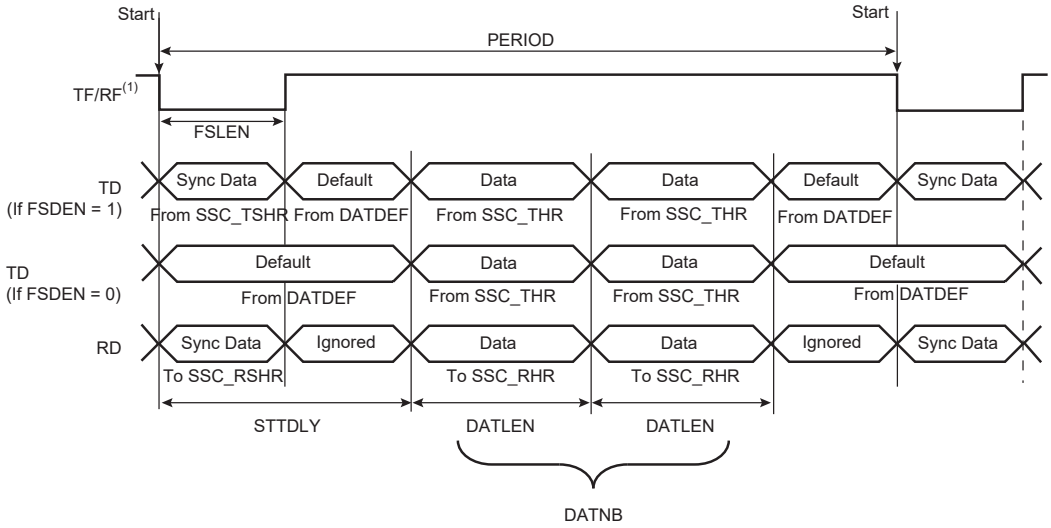
Additionally, the transmitter can be used to transfer synchronization and select the level driven on the TD pin while not in data transfer operation. This is done respectively by the Frame Sync Data Enable (FSDEN) and by the Data Default Value (DATDEF) bits in SSC_TFMR.

Downloaded from Arrow.com.

Table 44-2. Data Frame Registers

Transmitter	Receiver	Field	Length	Comment
SSC_TFMR	SSC_RFMR	DATLEN	Up to 32	Size of word
SSC_TFMR	SSC_RFMR	DATNB	Up to 16	Number of words transmitted in frame
SSC_TFMR	SSC_RFMR	MSBF	–	Most significant bit first
SSC_TFMR	SSC_RFMR	FSLEN	Up to 256	Size of Synchro data register
SSC_TFMR	–	DATDEF	0 or 1	Data default value ended
SSC_TFMR	–	FSDEN	–	Enable send SSC_TSHR
SSC_TCMR	SSC_RCMR	PERIOD	Up to 512	Frame size
SSC_TCMR	SSC_RCMR	STTDLY	Up to 255	Size of transmit start delay

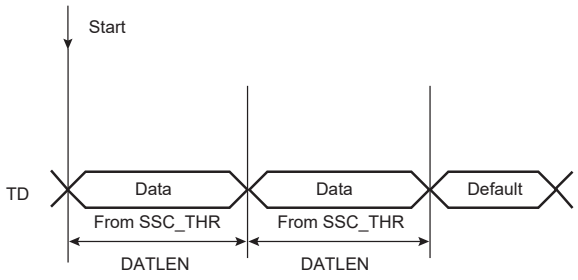
Figure 44-16. Transmit and Receive Frame Format in Edge/Pulse Start Modes



Note: 1. Example of input on falling edge of TF/RF.

In the example illustrated above, the SSC_THR is loaded twice. The FSDEN value has no effect on the transmission. SyncData cannot be output in Continuous mode.

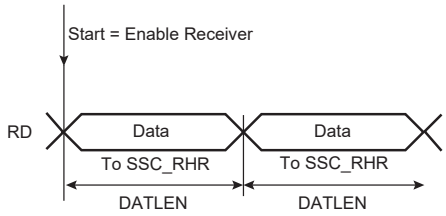
Figure 44-17. Transmit Frame Format in Continuous Mode (STTDLY = 0)



Start: 1. TXEMPTY set to 1
2. Write into the SSC_THR

Downloaded from Arrow.com.

Figure 44-18. Receive Frame Format in Continuous Mode (STTDLY = 0)



44.8.8 Loop Mode

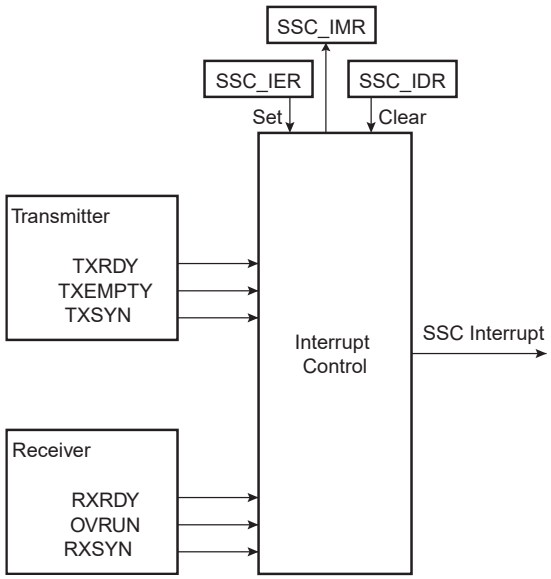
The receiver can be programmed to receive transmissions from the transmitter. This is done by setting the Loop Mode (LOOP) bit in the SSC_RFMR. In this case, RD is connected to TD, RF is connected to TF and RK is connected to TK.

44.8.9 Interrupt

Most bits in the SSC_SR have a corresponding bit in interrupt management registers.

The SSC can be programmed to generate an interrupt when it detects an event. The interrupt is controlled by writing the Interrupt Enable Register (SSC_IER) and Interrupt Disable Register (SSC_IDR). These registers enable and disable, respectively, the corresponding interrupt by setting and clearing the corresponding bit in the Interrupt Mask Register (SSC_IMR), which controls the generation of interrupts by asserting the SSC interrupt line connected to the interrupt controller.

Figure 44-19. Interrupt Block Diagram



44.8.10 Register Write Protection

To prevent any single software error from corrupting SSC behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the SSC Write Protection Mode Register (SSC_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the SSC Write Protection Status Register (SSC_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the SSC_WPSR.

The following registers can be write-protected:

- SSC Clock Mode Register
- SSC Receive Clock Mode Register
- SSC Receive Frame Mode Register

Downloaded from Arrow.com.

- SSC Transmit Clock Mode Register
- SSC Transmit Frame Mode Register
- SSC Receive Compare 0 Register
- SSC Receive Compare 1 Register

Downloaded from Arrow.com.

44.9 Register Summary

Note: Offsets 0x100–0x128 are reserved for PDC registers.

Offset	Name	Bit Pos.								
0x00	SSC_CR	31:24								
		23:16								
		15:8	SWRST						TXDIS	TXEN
		7:0							RXDIS	RXEN
0x04	SSC_CMR	31:24								
		23:16								
		15:8					DIV[11:8]			
		7:0	DIV[7:0]							
0x08 ... 0x0F	Reserved									
0x10	SSC_RCMR	31:24	PERIOD[7:0]							
		23:16	STTDLY[7:0]							
		15:8				STOP	START[3:0]			
		7:0	CKG[1:0]		CKI	CKO[2:0]			CKS[1:0]	
0x14	SSC_RFMR	31:24	FSLEN_EXT[3:0]							FSEDGE
		23:16		FSOS[2:0]			FSLEN[3:0]			
		15:8					DATNB[3:0]			
		7:0	MSBF		LOOP	DATLEN[4:0]				
0x18	SSC_TCMR	31:24	PERIOD[7:0]							
		23:16	STTDLY[7:0]							
		15:8					START[3:0]			
		7:0	CKG[1:0]		CKI	CKO[2:0]			CKS[1:0]	
0x1C	SSC_TFMR	31:24	FSLEN_EXT[3:0]							FSEDGE
		23:16	FSDEN	FSOS[2:0]			FSLEN[3:0]			
		15:8					DATNB[3:0]			
		7:0	MSBF		DATDEF	DATLEN[4:0]				
0x20	SSC_RHR	31:24	RDAT[31:24]							
		23:16	RDAT[23:16]							
		15:8	RDAT[15:8]							
		7:0	RDAT[7:0]							
0x24	SSC_THR	31:24	TDAT[31:24]							
		23:16	TDAT[23:16]							
		15:8	TDAT[15:8]							
		7:0	TDAT[7:0]							
0x28 ... 0x2F	Reserved									
0x30	SSC_RSHR	31:24								
		23:16								
		15:8	RSDAT[15:8]							
		7:0	RSDAT[7:0]							
0x34	SSC_TSHR	31:24								
		23:16								
		15:8	TSDAT[15:8]							
		7:0	TSDAT[7:0]							
0x38	SSC_RC0R	31:24								
		23:16								
		15:8	CP0[15:8]							
		7:0	CP0[7:0]							
0x3C	SSC_RC1R	31:24								
		23:16								
		15:8	CP1[15:8]							
		7:0	CP1[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x40	SSC_SR	31:24								
		23:16							RXEN	TXEN
		15:8					RXSYN	TXSYN	CP1	CP0
		7:0			OVRUN	RXRDY			TXEMPTY	TXRDY
0x44	SSC_IER	31:24								
		23:16								
		15:8					RXSYN	TXSYN	CP1	CP0
		7:0			OVRUN	RXRDY			TXEMPTY	TXRDY
0x48	SSC_IDR	31:24								
		23:16								
		15:8					RXSYN	TXSYN	CP1	CP0
		7:0			OVRUN	RXRDY			TXEMPTY	TXRDY
0x4C	SSC_IMR	31:24								
		23:16								
		15:8					RXSYN	TXSYN	CP1	CP0
		7:0			OVRUN	RXRDY			TXEMPTY	TXRDY
0x50 ... 0xE3	Reserved									
0xE4	SSC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0xE8	SSC_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

Downloaded from Arrow.com.

44.9.1 SSC Control Register

Name: SSC_CR
Offset: 0x0
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SWRST						TXDIS	TXEN
Access	W						W	W
Reset	–						–	–

Bit	7	6	5	4	3	2	1	0
							RXDIS	RXEN
Access							W	W
Reset							–	–

Bit 15 – SWRST Software Reset

Value	Description
0	No effect.
1	Performs a software reset. Has priority on any other bit in SSC_CR.

Bit 9 – TXDIS Transmit Disable

Value	Description
0	No effect.
1	Disables Transmit. If a character is currently being transmitted, disables at end of current character transmission.

Bit 8 – TXEN Transmit Enable

Value	Description
0	No effect.
1	Enables Transmit if TXDIS is not set.

Bit 1 – RXDIS Receive Disable

Value	Description
0	No effect.
1	Disables Receive. If a character is currently being received, disables at end of current character reception.

Bit 0 – RXEN Receive Enable

Value	Description
0	No effect.
1	Enables Receive if RXDIS is not set.

Downloaded from Arrow.com.

44.9.2 SSC Clock Mode Register

Name: SSC_CMR
Offset: 0x4
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SSC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					DIV[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – DIV[11:0] Clock Divider

Value	Description
0	The Clock Divider is not active.
Any other value	The divided clock equals the peripheral clock divided by 2 times DIV. The maximum bit rate is $f_{peripheral\ clock}/2$. The minimum bit rate is $f_{peripheral\ clock}/2 \times 4095 = f_{peripheral\ clock}/8190$.

Downloaded from Arrow.com.

44.9.3 SSC Receive Clock Mode Register

Name: SSC_RCMR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SSC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	PERIOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	STTDLY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				STOP	START[3:0]			
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CKG[1:0]		CKI	CKO[2:0]			CKS[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – PERIOD[7:0] Receive Period Divider Selection
This field selects the divider to apply to the selected Receive Clock in order to generate a new Frame Sync signal. If 0, no PERIOD signal is generated. If not 0, a PERIOD signal is generated each 2 x (PERIOD + 1) Receive Clock.

Bits 23:16 – STTDLY[7:0] Receive Start Delay
If STTDLY is not 0, a delay of STTDLY clock cycles is inserted between the start event and the current start of reception. When the receiver is programmed to start synchronously with the transmitter, the delay is also applied.
Note:
STTDLY must be configured in relation to the receive synchronization data to be stored in SSC_RSHR.

Bit 12 – STOP Receive Stop Selection

Value	Description
0	After completion of a data transfer when starting with a Compare 0, the receiver stops the data transfer and waits for a new compare 0.
1	After starting a receive with a Compare 0, the receiver operates in a continuous mode until a Compare 1 is detected.

Bits 11:8 – START[3:0] Receive Start Selection

Value	Name	Description
0	CONTINUOUS	Continuous, as soon as the receiver is enabled, and immediately after the end of transfer of the previous data.
1	TRANSMIT	Transmit start
2	RF_LOW	Detection of a low level on RF signal
3	RF_HIGH	Detection of a high level on RF signal
4	RF_FALLING	Detection of a falling edge on RF signal
5	RF_RISING	Detection of a rising edge on RF signal
6	RF_LEVEL	Detection of any level change on RF signal
7	RF_EDGE	Detection of any edge on RF signal

Downloaded from Arrow.com.

Value	Name	Description
8	CMP_0	Compare 0

Bits 7:6 – CKG[1:0] Receive Clock Gating Selection

Value	Name	Description
0	CONTINUOUS	None
1	EN_RF_LOW	Receive Clock enabled only if RF Low
2	EN_RF_HIGH	Receive Clock enabled only if RF High

Bit 5 – CKI Receive Clock Inversion

CKI affects only the Receive Clock and not the output clock signal.

Value	Description
0	The data inputs (Data and Frame Sync signals) are sampled on Receive Clock falling edge. The Frame Sync signal output is shifted out on Receive Clock rising edge.
1	The data inputs (Data and Frame Sync signals) are sampled on Receive Clock rising edge. The Frame Sync signal output is shifted out on Receive Clock falling edge.

Bits 4:2 – CKO[2:0] Receive Clock Output Mode Selection

Value	Name	Description
0	NONE	None, RK pin is an input
1	CONTINUOUS	Continuous Receive Clock, RK pin is an output
2	TRANSFER	Receive Clock only during data transfers, RK pin is an output

Bits 1:0 – CKS[1:0] Receive Clock Selection

Value	Name	Description
0	MCK	Divided Clock
1	TK	TK Clock signal
2	RK	RK pin

Downloaded from Arrow.com.

44.9.4 SSC Receive Frame Mode Register

Name: SSC_RFMR
Offset: 0x14
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SSC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	FSLEN_EXT[3:0]							FSEDGE
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit	23	22	21	20	19	18	17	16
		FSOS[2:0]			FSLEN[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					DATNB[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MSBF		LOOP	DATLEN[4:0]				
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bits 31:28 – FSLEN_EXT[3:0] FSLEN Field Extension
Extends FSLEN field. For details, see FSLEN: Receive Frame Sync Length.

Bit 24 – FSEDGE Frame Sync Edge Detection
Determines which edge on Frame Sync will generate the interrupt RXSYN in the SSC Status Register.

Value	Name	Description
0	POSITIVE	Positive Edge Detection
1	NEGATIVE	Negative Edge Detection

Bits 22:20 – FSOS[2:0] Receive Frame Sync Output Selection

Value	Name	Description
0	NONE	None, RF pin is an input
1	NEGATIVE	Negative Pulse, RF pin is an output
2	POSITIVE	Positive Pulse, RF pin is an output
3	LOW	Driven Low during data transfer, RF pin is an output
4	HIGH	Driven High during data transfer, RF pin is an output
5	TOGGLING	Toggling at each start of data transfer, RF pin is an output

Bits 19:16 – FSLEN[3:0] Receive Frame Sync Length
This field defines the number of bits sampled and stored in the Receive Sync Data Register. When this mode is selected by the START field in the Receive Clock Mode Register, it also determines the length of the sampled data to be compared to the Compare 0 or Compare 1 register.
This field is used with FSLEN_EXT to determine the pulse length of the Receive Frame Sync signal.
Pulse length is equal to FSLEN + (FSLEN_EXT × 16) + 1 Receive Clock periods.

Bits 11:8 – DATNB[3:0] Data Number per Frame
This field defines the number of data words to be received after each transfer start, which is equal to (DATNB + 1).

Downloaded from Arrow.com.

Bit 7 – MSBF Most Significant Bit First

Value	Description
0	The lowest significant bit of the data register is sampled first in the bit stream.
1	The most significant bit of the data register is sampled first in the bit stream.

Bit 5 – LOOP Loop Mode

Value	Description
0	Normal operating mode.
1	RD is driven by TD, RF is driven by TF and TK drives RK.

Bits 4:0 – DATLEN[4:0] Data Length

Value	Description
0	Forbidden value (1-bit data length not supported).
Any other value	The bit stream contains DATLEN + 1 data bits.

Downloaded from Arrow.com.

44.9.5 SSC Transmit Clock Mode Register

Name: SSC_TCMR
Offset: 0x18
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SSC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	PERIOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	STTDLY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					START[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CKG[1:0]		CKI	CKO[2:0]			CKS[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – PERIOD[7:0] Transmit Period Divider Selection
This field selects the divider to apply to the selected Transmit Clock to generate a new Frame Sync signal. If 0, no period signal is generated. If not 0, a period signal is generated at each 2 × (PERIOD + 1) Transmit Clock.

Bits 23:16 – STTDLY[7:0] Transmit Start Delay
If STTDLY is not 0, a delay of STTDLY clock cycles is inserted between the start event and the current start of transmission of data. When the transmitter is programmed to start synchronously with the receiver, the delay is also applied.
Note:
If STTDLY is too short with respect to transmit synchronization data (SSC_TSHR), SSC_THR.TDAT is transmitted instead of the end of SSC_TSHR.

Bits 11:8 – START[3:0] Transmit Start Selection

Value	Name	Description
0	CONTINUOUS	Continuous, as soon as a word is written in the SSC_THR (if Transmit is enabled), and immediately after the end of transfer of the previous data
1	RECEIVE	Receive start
2	TF_LOW	Detection of a low level on TF signal
3	TF_HIGH	Detection of a high level on TF signal
4	TF_FALLING	Detection of a falling edge on TF signal
5	TF_RISING	Detection of a rising edge on TF signal
6	TF_LEVEL	Detection of any level change on TF signal
7	TF_EDGE	Detection of any edge on TF signal

Bits 7:6 – CKG[1:0] Transmit Clock Gating Selection

Value	Name	Description
0	CONTINUOUS	None
1	EN_TF_LOW	Transmit Clock enabled only if TF Low

Downloaded from Arrow.com.

Value	Name	Description
2	EN_TF_HIGH	Transmit Clock enabled only if TF High

Bit 5 – CKI Transmit Clock Inversion
CKI affects only the Transmit Clock and not the Output Clock signal.

Value	Description
0	The data outputs (Data and Frame Sync signals) are shifted out on Transmit Clock falling edge. The Frame Sync signal input is sampled on Transmit Clock rising edge.
1	The data outputs (Data and Frame Sync signals) are shifted out on Transmit Clock rising edge. The Frame Sync signal input is sampled on Transmit Clock falling edge.

Bits 4:2 – CKO[2:0] Transmit Clock Output Mode Selection

Value	Name	Description
0	NONE	None, TK pin is an input
1	CONTINUOUS	Continuous Transmit Clock, TK pin is an output
2	TRANSFER	Transmit Clock only during data transfers, TK pin is an output

Bits 1:0 – CKS[1:0] Transmit Clock Selection

Value	Name	Description
0	MCK	Divided Clock
1	RK	RK Clock signal
2	TK	TK pin

Downloaded from Arrow.com.

44.9.6 SSC Transmit Frame Mode Register

Name: SSC_TFMR
Offset: 0x1C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SSC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	FSLEN_EXT[3:0]							FSEDGE
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit	23	22	21	20	19	18	17	16
	FSDEN	FSOS[2:0]			FSLEN[3:0]			
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					DATNB[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MSBF		DATDEF	DATLEN[4:0]				
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bits 31:28 – FSLEN_EXT[3:0] FSLEN Field Extension
Extends FSLEN field. For details, seee FSLEN bit description below.

Bit 24 – FSEDGE Frame Sync Edge Detection
Determines which edge on frame synchronization will generate the interrupt TXSYN (Status Register).

Value	Name	Description
0	POSITIVE	Positive Edge Detection
1	NEGATIVE	Negative Edge Detection

Bit 23 – FSDEN Frame Sync Data Enable

Value	Description
0	The TD line is driven with the default value during the Transmit Frame Sync signal.
1	SSC_TSHR value is shifted out during the transmission of the Transmit Frame Sync signal.

Bits 22:20 – FSOS[2:0] Transmit Frame Sync Output Selection

Value	Name	Description
0	NONE	None, TF pin is an input
1	NEGATIVE	Negative Pulse, TF pin is an output
2	POSITIVE	Positive Pulse, TF pin is an output
3	LOW	Driven Low during data transfer
4	HIGH	Driven High during data transfer
5	TOGGLING	Toggling at each start of data transfer

Bits 19:16 – FSLEN[3:0] Transmit Frame Sync Length
This field defines the length of the Transmit Frame Sync signal and the number of bits shifted out from SSC_TSHR if FSDEN is 1.
This field is used with FSLEN_EXT to determine the pulse length of the Transmit Frame Sync signal.
Pulse length is equal to FSLEN + (FSLEN_EXT × 16) + 1 Transmit Clock period.

Downloaded from Arrow.com.

Bits 11:8 – DATNB[3:0] Data Number per Frame
This field defines the number of data words to be transferred after each transfer start, which is equal to (DATNB + 1).

Bit 7 – MSBF Most Significant Bit First

Value	Description
0	The lowest significant bit of the data register is shifted out first in the bit stream.
1	The most significant bit of the data register is shifted out first in the bit stream.

Bit 5 – DATDEF Data Default Value
This bit defines the level driven on the TD pin while out of transmission. Note that if the pin is defined as multi-drive by the PIO Controller, the pin is enabled only if the SCC TD output is 1.
When the TD pin is configured in Multi-drive (Open-drain) mode by the PIO controller, a 0 is driven if SSC data output equals 0 and the pin is in high-impedance when SSC data output is 1.

Bits 4:0 – DATLEN[4:0] Data Length

Value	Description
0	Forbidden value (1-bit data length not supported).
Any other value	The bit stream contains DATLEN + 1 data bits.

Downloaded from Arrow.com.

44.9.7 SSC Receive Holding Register

Name: SSC_RHR
Offset: 0x20
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RDAT[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RDAT[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RDAT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RDAT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RDAT[31:0] Receive Data
Right-aligned regardless of the number of data bits defined by DATLEN in SSC_RFMR.

Downloaded from Arrow.com.

44.9.8 SSC Transmit Holding Register

Name: SSC_THR
Offset: 0x24
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	TDAT[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	TDAT[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TDAT[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TDAT[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – TDAT[31:0] Transmit Data
Right-aligned regardless of the number of data bits defined by DATLEN in SSC_TFMR.

Downloaded from Arrow.com.

44.9.9 SSC Receive Synchronization Holding Register

Name: SSC_RSHR
Offset: 0x30
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RSDAT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RSDAT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RSDAT[15:0] Receive Synchronization Data

Downloaded from Arrow.com.

44.9.10 SSC Transmit Synchronization Holding Register

Name: SSC_TSHR
Offset: 0x34
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TSDAT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TSDAT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TSDAT[15:0] Transmit Synchronization Data

Downloaded from Arrow.com.

44.9.11 SSC Receive Compare 0 Register

Name: SSC_RC0R
Offset: 0x38
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SSC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CP0[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CP0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CP0[15:0] Receive Compare Data 0

Downloaded from Arrow.com.

44.9.12 SSC Receive Compare 1 Register

Name: SSC_RC1R
Offset: 0x3C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SSC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CP1[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CP1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CP1[15:0] Receive Compare Data 1

Downloaded from Arrow.com.

44.9.13 SSC Status Register

Name: SSC_SR
Offset: 0x40
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							RXEN	TXEN
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
					RXSYN	TXSYN	CP1	CP0
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
			OVRUN	RXRDY			TXEMPTY	TXRDY
Access			R	R			R	R
Reset			0	0			0	0

Bit 17 – RXEN Receive Enable

Value	Description
0	Receive is disabled.
1	Receive is enabled.

Bit 16 – TXEN Transmit Enable

Value	Description
0	Transmit is disabled.
1	Transmit is enabled.

Bit 11 – RXSYN Receive Sync

Value	Description
0	An Rx Sync has not occurred since the last read of the Status Register.
1	An Rx Sync has occurred since the last read of the Status Register.

Bit 10 – TXSYN Transmit Sync

Value	Description
0	A Tx Sync has not occurred since the last read of the Status Register.
1	A Tx Sync has occurred since the last read of the Status Register.

Bit 9 – CP1 Compare 1

Value	Description
0	A compare 1 has not occurred since the last read of the Status Register.
1	A compare 1 has occurred since the last read of the Status Register.

Bit 8 – CP0 Compare 0

Value	Description
0	A compare 0 has not occurred since the last read of the Status Register.

Downloaded from Arrow.com.

Value	Description
1	A compare 0 has occurred since the last read of the Status Register.

Bit 5 – OVRUN Receive Overrun

Value	Description
0	No data has been loaded in SSC_RHR while previous data has not been read since the last read of the Status Register.
1	Data has been loaded in SSC_RHR while previous data has not yet been read since the last read of the Status Register.

Bit 4 – RXRDY Receive Ready

Value	Description
0	SSC_RHR is empty.
1	Data has been received and loaded in SSC_RHR.

Bit 1 – TXEMPTY Transmit Empty

Value	Description
0	Data remains in SSC_THR or is currently transmitted from TSR.
1	Last data written in SSC_THR has been loaded in TSR and last data loaded in TSR has been transmitted.

Bit 0 – TXRDY Transmit Ready

Value	Description
0	Data has been loaded in SSC_THR and is waiting to be loaded in the transmit shift register (TSR).
1	SSC_THR is empty.

Downloaded from Arrow.com.

44.9.14 SSC Interrupt Enable Register

Name: SSC_IER
Offset: 0x44
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					RXSYN	TXSYN	CP1	CP0
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
			OVRUN	RXRDY			TXEMPTY	TXRDY
Access			W	W			W	W
Reset			–	–			–	–

Bit 11 – RXSYN Rx Sync Interrupt Enable

Value	Description
0	No effect.
1	Enables the Rx Sync Interrupt.

Bit 10 – TXSYN Tx Sync Interrupt Enable

Value	Description
0	No effect.
1	Enables the Tx Sync Interrupt.

Bit 9 – CP1 Compare 1 Interrupt Enable

Value	Description
0	No effect.
1	Enables the Compare 1 Interrupt.

Bit 8 – CP0 Compare 0 Interrupt Enable

Value	Description
0	No effect.
1	Enables the Compare 0 Interrupt.

Bit 5 – OVRUN Receive Overrun Interrupt Enable

Value	Description
0	No effect.
1	Enables the Receive Overrun Interrupt.

Bit 4 – RXRDY Receive Ready Interrupt Enable

Value	Description
0	No effect.

Downloaded from Arrow.com.

Value	Description
1	Enables the Receive Ready Interrupt.

Bit 1 – TXEMPTY Transmit Empty Interrupt Enable

Value	Description
0	No effect.
1	Enables the Transmit Empty Interrupt.

Bit 0 – TXRDY Transmit Ready Interrupt Enable

Value	Description
0	No effect.
1	Enables the Transmit Ready Interrupt.

Downloaded from Arrow.com.

44.9.15 SSC Interrupt Disable Register

Name: SSC_IDR
Offset: 0x48
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					RXSYN	TXSYN	CP1	CP0
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
			OVRUN	RXRDY			TXEMPTY	TXRDY
Access			W	W			W	W
Reset			–	–			–	–

Bit 11 – RXSYN Rx Sync Interrupt Enable

Value	Description
0	No effect.
1	Disables the Rx Sync Interrupt.

Bit 10 – TXSYN Tx Sync Interrupt Enable

Value	Description
0	No effect.
1	Disables the Tx Sync Interrupt.

Bit 9 – CP1 Compare 1 Interrupt Disable

Value	Description
0	No effect.
1	Disables the Compare 1 Interrupt.

Bit 8 – CP0 Compare 0 Interrupt Disable

Value	Description
0	No effect.
1	Disables the Compare 0 Interrupt.

Bit 5 – OVRUN Receive Overrun Interrupt Disable

Value	Description
0	No effect.
1	Disables the Receive Overrun Interrupt.

Bit 4 – RXRDY Receive Ready Interrupt Disable

Value	Description
0	No effect.

Downloaded from Arrow.com.

Value	Description
1	Disables the Receive Ready Interrupt.

Bit 1 – TXEMPTY Transmit Empty Interrupt Disable

Value	Description
0	No effect.
1	Disables the Transmit Empty Interrupt.

Bit 0 – TXRDY Transmit Ready Interrupt Disable

Value	Description
0	No effect.
1	Disables the Transmit Ready Interrupt.

Downloaded from Arrow.com.

44.9.16 SSC Interrupt Mask Register

Name: SSC_IMR
Offset: 0x4C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					RXSYN	TXSYN	CP1	CP0
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
			OVRUN	RXRDY			TXEMPTY	TXRDY
Access			R	R			R	R
Reset			0	0			0	0

Bit 11 – RXSYN Rx Sync Interrupt Mask

Value	Description
0	The Rx Sync Interrupt is disabled.
1	The Rx Sync Interrupt is enabled.

Bit 10 – TXSYN Tx Sync Interrupt Mask

Value	Description
0	The Tx Sync Interrupt is disabled.
1	The Tx Sync Interrupt is enabled.

Bit 9 – CP1 Compare 1 Interrupt Mask

Value	Description
0	The Compare 1 Interrupt is disabled.
1	The Compare 1 Interrupt is enabled.

Bit 8 – CP0 Compare 0 Interrupt Mask

Value	Description
0	The Compare 0 Interrupt is disabled.
1	The Compare 0 Interrupt is enabled.

Bit 5 – OVRUN Receive Overrun Interrupt Mask

Value	Description
0	The Receive Overrun Interrupt is disabled.
1	The Receive Overrun Interrupt is enabled.

Bit 4 – RXRDY Receive Ready Interrupt Mask

Value	Description
0	The Receive Ready Interrupt is disabled.

Downloaded from Arrow.com.

Value	Description
1	The Receive Ready Interrupt is enabled.

Bit 1 – TXEMPTY Transmit Empty Interrupt Mask

Value	Description
0	The Transmit Empty Interrupt is disabled.
1	The Transmit Empty Interrupt is enabled.

Bit 0 – TXRDY Transmit Ready Interrupt Mask

Value	Description
0	The Transmit Ready Interrupt is disabled.
1	The Transmit Ready Interrupt is enabled.

Downloaded from Arrow.com.

44.9.17 SSC Write Protection Mode Register

Name: SSC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x535343	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 0 – WPEN Write Protection Enable

See Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x535343 (“SSC” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x535343 (“SSC” in ASCII).

Downloaded from Arrow.com.

44.9.18 SSC Write Protection Status Register

Name: SSC_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protect Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the SSC_WPSR.
1	A write protection violation has occurred since the last read of the SSC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

45. Flexible Serial Communication Controller (FLEXCOM)

45.1 Description

The Flexible Serial Communication Controller (FLEXCOM) offers several serial communication protocols that are managed by the three submodules USART, SPI, and TWI.

The Universal Synchronous Asynchronous Receiver Transceiver (USART) provides one full-duplex universal synchronous asynchronous serial link. Data frame format is widely programmable (data length, parity, number of stop bits) to support a maximum of standards. The receiver implements parity error, framing error and overrun error detection. The receiver timeout enables handling variable-length frames and the transmitter timeguard facilitates communications with slow remote devices. Multidrop communications are also supported through address bit handling in reception and transmission.

The USART features three test modes: Remote Loopback, Local Loopback and Automatic Echo.

The USART supports specific operating modes providing interfaces on RS485, LIN, LON, , with ISO7816 T = 0 or T = 1 smart card slots, and infrared transceivers. The hardware handshaking feature enables an out-of-band flow control by automatic management of the pins RTS and CTS.

The USART supports the connection to the DMA Controller, which enables data transfers to the transmitter and from the receiver. The DMAC provides chained buffer management without any intervention of the processor.

The Serial Peripheral Interface (SPI) circuit is a synchronous serial data link that provides communication with external devices in Master or Slave mode. It also enables communication between processors if an external processor is connected to the system.

The Serial Peripheral Interface is essentially a shift register that serially transmits data bits to other SPIs. During a data transfer, one SPI system acts as the “master” which controls the data flow, while the other devices act as “slaves” which have data shifted into and out by the master. Different CPUs can take turn being masters (multiple master protocol, contrary to single master protocol where one CPU is always the master while all of the others are always slaves). One master can simultaneously shift data into multiple slaves. However, only one slave can drive its output to write data back to the master at any given time.

A slave device is selected when the master asserts its NSS signal. If multiple slave devices exist, the master generates a separate slave select signal for each slave (NPCS).

The SPI system consists of two data lines and two control lines:

- Master Out Slave In (MOSI)—This data line supplies the output data from the master shifted into the input(s) of the slave(s).
- Master In Slave Out (MISO)—This data line supplies the output data from a slave to the input of the master. There may be no more than one slave transmitting data during any particular transfer.
- Serial Clock (SPCK)—This control line is driven by the master and regulates the flow of the data bits. The master can transmit data at a variety of baud rates; there is one SPCK pulse for each bit that is transmitted.
- Slave Select (NSS)—This control line allows slaves to be turned on and off by hardware.

The Two-wire Interface (TWI) interconnects components on a unique two-wire bus, made up of one clock line and one data line with speeds of up to 400 kbits per second in Fast mode and up to 3.4 Mbits per second in High-speed Slave mode only, based on a byte-oriented transfer format. It can be used with any Two-wire Interface bus Serial EEPROM and I^2C-compatible devices, such as a Real-Time Clock (RTC), Dot Matrix/Graphic LCD Controller and temperature sensor. The TWI is programmable as a master or a slave with sequential or single-byte access. Multiple master capability is supported.

Arbitration of the bus is performed internally and puts the TWI in Slave mode automatically if the bus arbitration is lost.

A configurable baud rate generator permits the output data rate to be adapted to a wide range of core clock frequencies.

The following table lists the compatibility level of any Two-wire Interface in Master mode and a full I^2C compatible device.

Downloaded from Arrow.com.

Table 45-1. TWI Compatibility with I²C Standard

I²C Standard	TWI
Standard Mode Speed (100 kHz)	Supported
Fast Mode Speed (400 kHz)	Supported
Fast Mode Plus (1 MHz)	Supported
High-speed Mode (3.4 MHz)	Supported
7- or 10-bit(1) Slave Addressing	Supported
Repeated Start (Sr) Condition	Supported
ACK and NACK Management	Supported
Input Filtering	Supported
Slope Control	Not Supported
Clock Stretching	Supported
Multi Master Capability	Supported

Notes: 1. 10-bit support in Master mode only

45.2 Embedded Characteristics

45.2.1 USART/UART Characteristics

- 16-byte Transmit and Receive FIFOs
- Programmable Baud Rate Generator
- Baud Rate can be Independent of the Processor/Peripheral Clock
- Comparison Function on Received Character
- 5-bit to 9-bit Full-duplex Synchronous or Asynchronous Serial Communications
  - 1, 1.5 or 2 stop bits in Asynchronous mode or 1 or 2 stop bits in Synchronous mode
  - Parity generation and error detection
  - Framing error detection, overrun error detection
  - Digital filter on receive line
  - MSB- or LSB-first
  - Optional break generation and detection
  - By 8 or by 16 oversampling receiver frequency
  - Optional hardware handshaking RTS-CTS
  - Receiver timeout and transmitter timeguard
  - Optional Multidrop mode with address generation and detection
- RS485 with Driver Control Signal
- ISO7816, T = 0 or T = 1 Protocols for Interfacing with Smart Cards
  - NACK handling, error counter with repetition and iteration limit
- IrDA Modulation and Demodulation
  - Communication at up to 115.2 kbit/s
- Up to 16-bit data, Manchester-encoded Frame Support
- LIN Mode
  - Compliant with LIN 1.3 and LIN 2.0 specifications
  - Master or slave
  - Processing of frames with up to 256 data bytes

Downloaded from Arrow.com.

  - Response data length can be configurable or defined automatically by the identifier
  - Self-synchronization in slave node configuration
  - Automatic processing and verification of the “synch break” and the “synch field”
  - “Synch break” detection even when partially superimposed with a data byte
  - Automatic identifier parity calculation/sending and verification
  - Parity sending and verification can be disabled
  - Automatic checksum calculation/sending and verification
  - Checksum sending and verification can be disabled
  - Support both “classic” and “enhanced” checksum types
  - Full LIN error checking and reporting
  - Frame Slot mode: master allocates slots to the scheduled frames automatically
  - Generation of the wakeup signal
- LON Mode
  - Compliant with CEA-709 specification
  - Full-layer 2 implementation
  - Differential Manchester encoding/decoding (CDP)
  - Preamble generation including bit- and byte-sync fields
  - LON timings handling (beta1, beta2, IDT, etc.)
  - CRC generation and checking
  - Automated random number generation
  - Backlog calculation and update
  - Collision detection support
  - Supports both comm_type = 1 and comm_type = 2 modes
  - Clock drift tolerance up to 16%
  - Optimal for node-to-node communication (no embedded digital line filter)
- Test Modes
  - Remote loopback, local loopback, automatic echo
- Supports Connection of:
  - Two DMA Controller (DMAC) channels
  - Offers buffer transfer without processor intervention
- Register Write Protection

45.2.2 SPI Characteristics

- 16-byte Transmit and Receive FIFOs
- Master or Slave Serial Peripheral Bus Interface
  - 8-bit to 16-bit programmable data length per chip select
  - Programmable phase and polarity per chip select
  - Programmable transfer delay between consecutive transfers and delay before SPI clock per chip select
  - Programmable delay between chip selects
- Selectable Mode Fault Detection
- Master Mode Can Drive SPCK up to Peripheral Clock
- Master Mode Bit Rate Can Be Independent of the Processor/Peripheral Clock
- Slave Mode Operates on SPCK, Asynchronously with Core and Bus Clock
- Four Chip Selects with External Decoder Support Allow Communication with up to 15Peripherals
- Communication with Serial External Devices Supported
  - Serial memories, such as DataFlash and 3-wire EEPROMs
  - Serial peripherals, such as ADCs, DACs, LCD controllers, CAN controllers and sensors
  - External coprocessors
- Connection to DMA Channel Capabilities, Optimizing Data Transfers
  - One channel for the receiver

Downloaded from Arrow.com.

  - One channel for the transmitter
- Register Write Protection

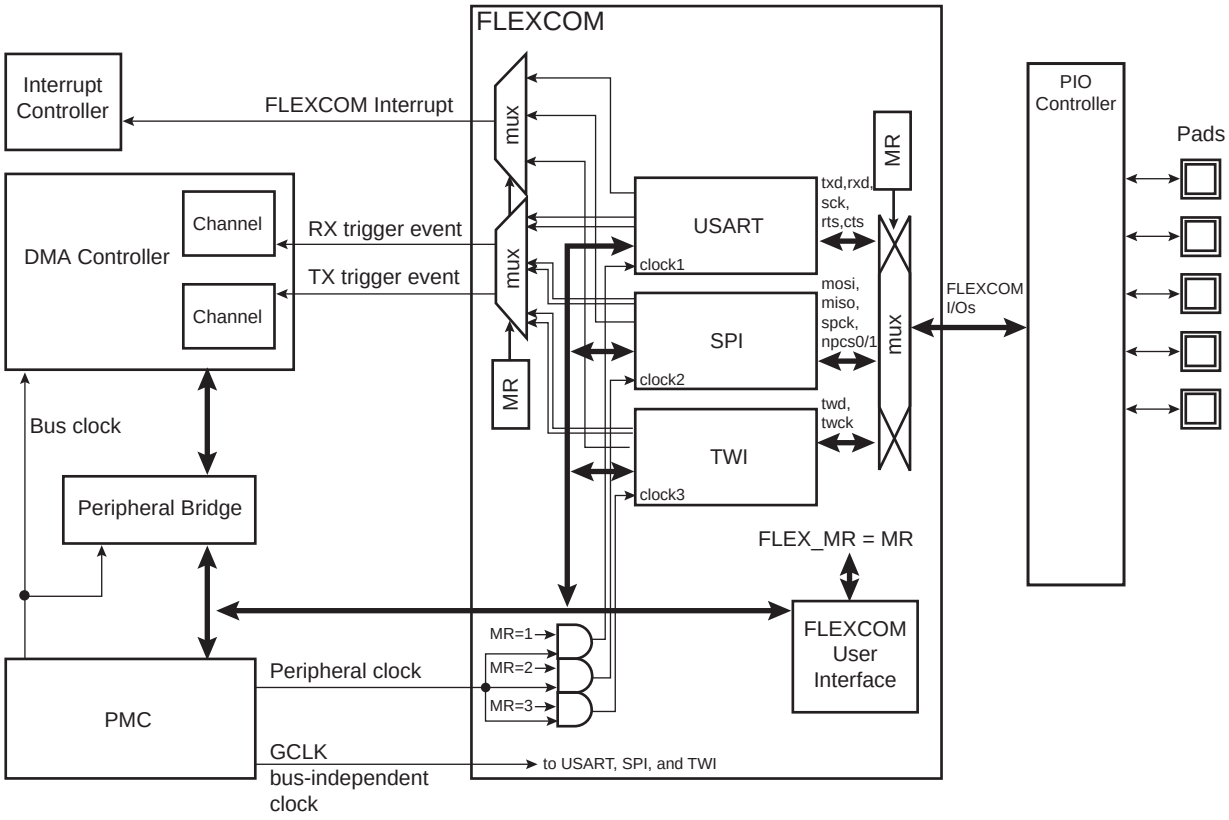
45.2.3 TWI/SMBus Characteristics

- 16-byte Transmit and Receive FIFOs
- Bit Rate: 1 Mbit/s in Fast Mode Plus and 3.4 Mbit/s in High-Speed Mode
- Bit Rate can be Independent of the Processor/Peripheral Clock
- SMBus Support
- Compatible with Two-wire Interface Serial Memory and I^2C Compatible Devices(1)
- Master and Multi-Master Operation (Standard and Fast Mode Only)
- Slave Mode Operation (Standard, Fast and High-Speed Mode)
- One, Two or Three Bytes for Slave Address
- Sequential Read/Write Operations
- General Call Supported in Slave Mode
- Connection to DMA Controller Channels Optimizes Data Transfers
  - One channel for the receiver
  - One channel for the transmitter
- Register Write Protection

(1) See table TWI Compatibility with I^2C Standard for further details.

45.3 Block Diagram

Figure 45-1. FLEXCOM Block Diagram



Downloaded from Arrow.com.

45.4 I/O Lines Description

Table 45-2. I/O Lines Description

Name	Description			Type
	USART/UART	SPI	TWI	
FLEXCOM_IO0	TXD	MOSI	TWD	I/O
FLEXCOM_IO1	RXD	MISO	TWCK	I/O
FLEXCOM_IO2	SCK	SPCK	–	I/O
FLEXCOM_IO3	CTS	NPCS0/NSS	–	I/O
FLEXCOM_IO4	RTS	NPCS1	–	O
FLEXCOM_IO5	–	NPCS2	–	O
FLEXCOM_IO6	–	NPCS3	–	O

45.5 Product Dependencies

45.5.1 I/O Lines

The pins used for interfacing the FLEXCOM are multiplexed with the PIO lines. The programmer must first program the PIO controller to assign the desired FLEXCOM pins to their peripheral function. If I/O lines of the FLEXCOM are not used by the application, they can be used for other purposes by the PIO Controller.

45.5.2 Power Management

The peripheral clock is not continuously provided to the FLEXCOM. The programmer must first enable the FLEXCOM Clock in the Power Management Controller (PMC) before using the USART or SPI or TWI.

45.5.3 Interrupt Sources

The FLEXCOM interrupt line is connected on one of the internal sources of the Interrupt Controller. Using the FLEXCOM interrupt requires the Interrupt Controller to be programmed first.

45.6 Register Accesses

Register accesses support 8-bit, 16-bit and 32-bit accesses, which means that only an 8-bit part of a 32-bit register can be written in one access, for instance. For this the access must be done with the right size at the right address.

8-bit, 16-bit and 32-bit accesses are supported for register accesses. However, a field in a register cannot be partially written (e.g., if a field is bigger than 8 bits, the whole field must be written).

This feature helps avoiding a read-modify-write process if only a small part of the register is to be modified.

45.7 USART Functional Description

45.7.1 Baud Rate Generator

The baud rate generator provides the bit period clock named “baud rate clock” to both the receiver and the transmitter.

Configuring the USCLKS field in FLEX_US_MR selects the baud rate generator clock from one of the following sources:

- the peripheral clock

Downloaded from Arrow.com.

- a division of the peripheral clock, the divider being product dependent, but generally set to 8
- a fully programmable generic clock (GCLK) provided by PMC and independent of processor/peripheral clock
- the external clock, available on the SCK pin

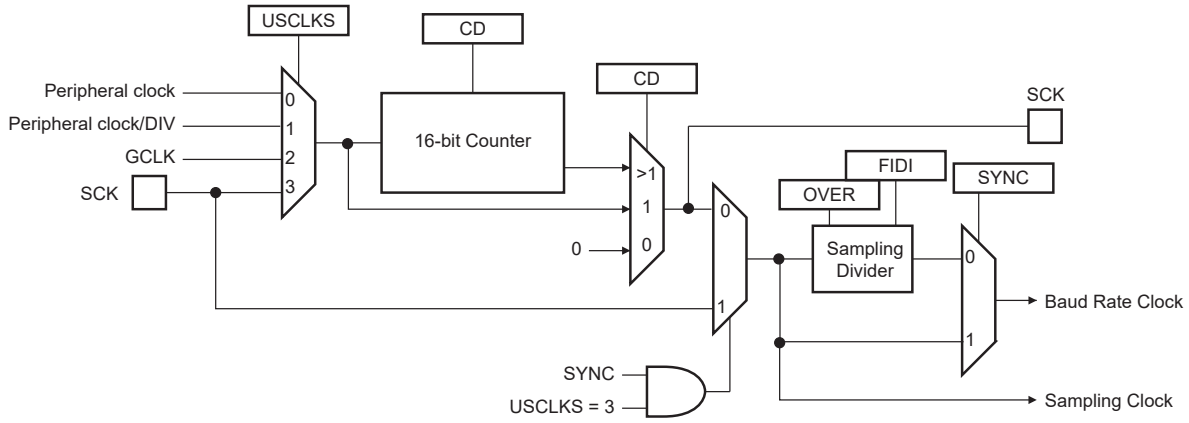
The baud rate generator is based upon a 16-bit divider, which is programmed with the CD field of the Baud Rate Generator Register (FLEX_US_BRGR). If a zero is written to CD, the baud rate generator does not generate any clock. If a one is written to CD, the divider is bypassed and becomes inactive.

If the external SCK clock is selected, the duration of the low and high levels of the signal provided on the SCK pin must be longer than a peripheral clock period. The frequency of the signal provided on SCK must be at least three times lower than peripheral clock .

If GCLK is selected, the baud rate is independent of the processor/peripheral clock and thus processor/peripheral clock frequency can be changed without affecting the USART transfer. The GCLK frequency must be at least three times lower than peripheral clock frequency.

If GCLK is selected (USCLKS = 2) and the SCK pin is driven (CLKO = 1), the CD field must be greater than 1.

Figure 45-2. Baud Rate Generator



45.7.1.1 Baud Rate in Asynchronous Mode

If the USART is programmed to operate in Asynchronous mode, the selected clock is first divided by CD, which is field-programmed in FLEX_US_BRGR. The resulting clock is provided to the receiver as a sampling clock and then divided by 16 or 8, depending on the programming of FLEX_US_MR.OVER.

If OVER is set, the receiver sampling is eight times higher than the baud rate clock. If OVER is cleared, the sampling is performed at 16 times the baud rate clock.

The baud rate is calculated as per the following formula:

$$\text{Baud rate} = \frac{\text{Selected Clock}}{(8(2-\text{OVER})\text{CD})}$$

This gives a maximum baud rate of peripheral clock divided by 8, assuming that peripheral clock is the highest possible clock and that the OVER bit is set.

45.7.1.1.1 Baud Rate Calculation Example

The following table shows calculations of CD to obtain a baud rate at 38,400 bit/s for different source clock frequencies. It also shows the actual resulting baud rate and the error.

Table 45-3. Baud Rate Example (OVER = 0)

Source Clock (MHz)	Expected Baud Rate (bit/s)	Calculation Result	CD	Actual Baud Rate (bit/s)	Error
3,686,400	38,400	6.00	6	38,400.00	0.00%
4,915,200	38,400	8.00	8	38,400.00	0.00%
5,000,000	38,400	8.14	8	39,062.50	1.70%

Downloaded from Arrow.com.

.....continued

Source Clock (MHz)	Expected Baud Rate (bit/s)	Calculation Result	CD	Actual Baud Rate (bit/s)	Error
7,372,800	38,400	12.00	12	38,400.00	0.00%
8,000,000	38,400	13.02	13	38,461.54	0.16%
12,000,000	38,400	19.53	20	37,500.00	2.40%
12,288,000	38,400	20.00	20	38,400.00	0.00%
14,318,180	38,400	23.30	23	38,908.10	1.31%
14,745,600	38,400	24.00	24	38,400.00	0.00%
18,432,000	38,400	30.00	30	38,400.00	0.00%
24,000,000	38,400	39.06	39	38,461.54	0.16%
24,576,000	38,400	40.00	40	38,400.00	0.00%
25,000,000	38,400	40.69	40	38,109.76	0.76%
32,000,000	38,400	52.08	52	38,461.54	0.16%
32,768,000	38,400	53.33	53	38,641.51	0.63%
33,000,000	38,400	53.71	54	38,194.44	0.54%
40,000,000	38,400	65.10	65	38,461.54	0.16%
50,000,000	38,400	81.38	81	38,580.25	0.47%

The baud rate is calculated with the following formula:

Baud rate = MCK / CD × 16

The baud rate error is calculated with the following formula. It is not recommended to work with an error higher than 5%.

$$\text{Error} = 1 - \left(\frac{\text{Expected Baud Rate}}{\text{Actual Baud Rate}}\right)$$

45.7.1.2 Fractional Baud Rate in Asynchronous Mode

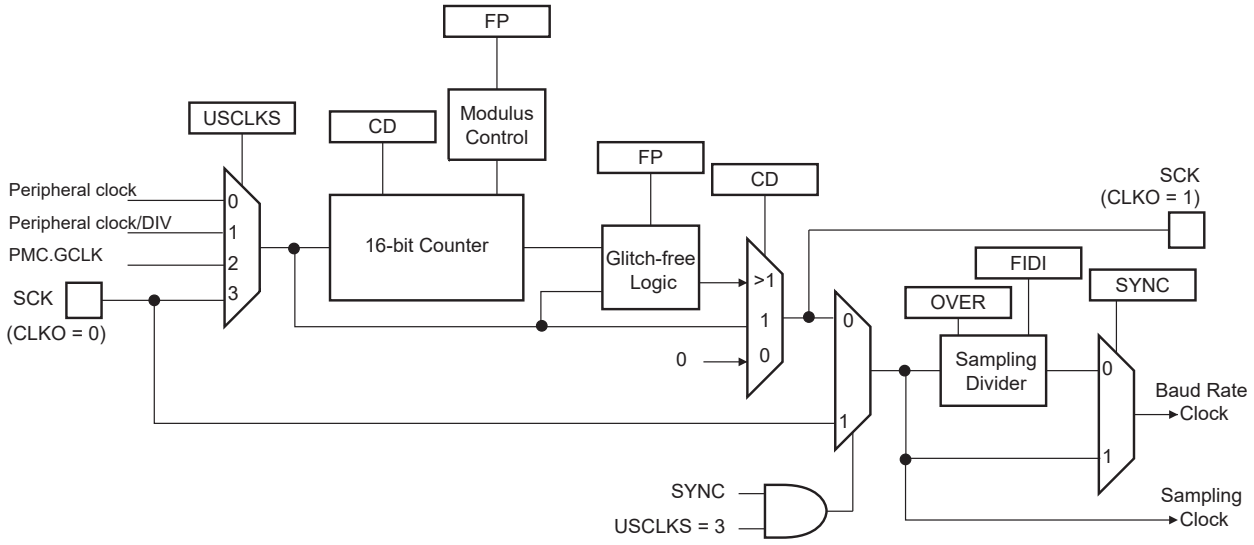
The baud rate generator previously defined is subject to the following limitation: the output frequency changes by only integer multiples of the reference frequency. An approach to this problem is to integrate a fractional N clock generator that has a high resolution. The generator architecture is modified to obtain baud rate changes by a fraction of the reference source clock. This fractional part is programmed with the FP field in FLEX_US_BRGR. If FP is not 0, the fractional part is activated. The resolution is one eighth of the clock divider. The fractional baud rate is calculated using the following formula:

$$\text{Baud rate} = \frac{\text{Selected Clock}}{\left(8(2 - \text{OVER})\left(\text{CD} + \frac{\text{FP}}{8}\right)\right)}$$

The modified architecture is presented in the following figure.

Downloaded from Arrow.com.

Figure 45-3. Fractional Baud Rate Generator



WARNING When the value of field FP is greater than 0, the SCK (oversampling clock) generates nonconstant duty cycles. The SCK high duration is increased by "selected clock" period from time to time. The duty cycle depends on the value of the CD field.

45.7.1.3 Baud Rate in Synchronous Mode

If the USART is programmed to operate in Synchronous mode, the selected clock is simply divided by the CD field in FLEX_US_BRGR:

$$\text{Baud rate} = \frac{\text{Selected Clock}}{\text{CD}}$$

In Synchronous mode, if the external clock is selected (USCLKS = 3) and CLKO = 0 (Slave mode), the clock is provided directly by the signal on the USART SCK pin. No division is active. The value written in FLEX_US_BRGR has no effect. The external clock frequency must be at least three times lower than the system clock. In Synchronous mode master (USCLKS = 0 or 1, CLKO = 1), the receive part limits the SCK maximum frequency to $f_{peripheral\ clock}/3$.

When either the external clock SCK or the internal clock divided (peripheral clock/DIV or GCLK) is selected, the value programmed in CD must be even if the user has to ensure a 50:50 mark/space ratio on the SCK pin. If the peripheral clock is selected, the baud rate generator ensures a 50:50 duty cycle on the SCK pin, even if the value programmed in CD is odd.

45.7.1.4 Baud Rate in ISO 7816 Mode

The ISO7816 specification defines the bit rate with the following formula:

$$B = \frac{\text{Di}}{\text{Fi}} \times f$$

where:

- B is the bit rate
- Di is the bit rate adjustment factor
- Fi is the clock frequency division factor
- f is the ISO7816 clock frequency (Hz)

Di is a binary value encoded on a 4-bit field, named DI, as represented in the following table.

Table 45-4. Binary and Decimal Values for Di

DI field	0001	0010	0011	0100	0101	0110	1000	1001

Downloaded from Arrow.com.

Di (decimal)	1	2	4	8	16	32	12	20

Fi is a binary value encoded on a 4-bit field, named FI, as represented in the following table.

Table 45-5. Binary and Decimal Values for Fi

FI field	0000	0001	0010	0011	0100	0101	0110	1001	1010	1011	1100	1101
Fi (decimal)	372	372	558	744	1116	1488	1860	512	768	1024	1536	2048

The following table shows the resulting Fi/Di Ratio, which is the ratio between the ISO7816 clock and the baud rate clock.

Table 45-6. Possible Values for the Fi/Di Ratio

Fi/Di	372	558	744	1116	1488	1806	512	768	1024	1536	2048
1	372	558	744	1116	1488	1860	512	768	1024	1536	2048
2	186	279	372	558	744	930	256	384	512	768	1024
4	93	139.5	186	279	372	465	128	192	256	384	512
8	46.5	69.75	93	139.5	186	232.5	64	96	128	192	256
16	23.25	34.87	46.5	69.75	93	116.2	32	48	64	96	128
32	11.62	17.43	23.25	34.87	46.5	58.13	16	24	32	48	64
12	31	46.5	62	93	124	155	42.66	64	85.33	128	170.6
20	18.6	27.9	37.2	55.8	74.4	93	25.6	38.4	51.2	76.8	102.4

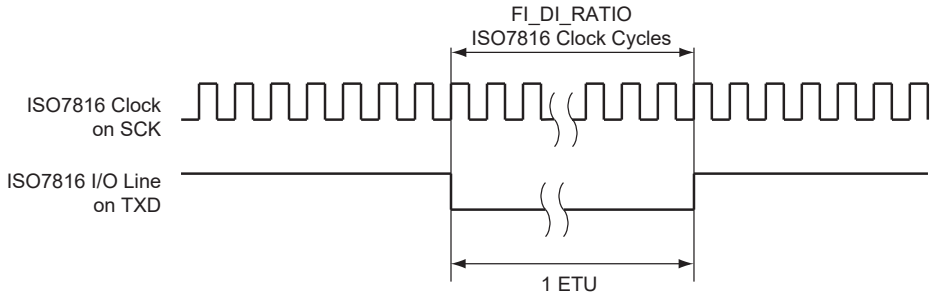
If the USART is configured in ISO7816 mode, the clock selected by the USCLKS field in FLEX_US_MR is first divided by the value programmed in field CD field in FLEX_US_BRGR. The resulting clock can be provided to the SCK pin to feed the smart card clock inputs. This means that FLEX_US_MR.CLKO can be set.

This clock is then divided by the value programmed in the FI_DI_RATIO field in the FI DI Ratio Register (FLEX_US_FIDI). This is performed by the Sampling Divider, which performs a division by up to 65535 in ISO7816 mode. The noninteger values of the Fi/Di Ratio are not supported and the user must program the FI_DI_RATIO field to a value as close as possible to the expected value.

The FI_DI_RATIO field resets to the value 0x174 (372 in decimal) and is the most common divider between the ISO7816 clock and the bit rate (Fi = 372, Di = 1).

The following figure shows the relation between the Elementary Time Unit, corresponding to a bit time, and the ISO 7816 clock.

Figure 45-4. Elementary Time Unit (ETU)



45.7.2 Receiver and Transmitter Control

After reset, the receiver is disabled. The user must enable the receiver by setting the RXEN bit in the USART Control Register (FLEX_US_CR). However, the receiver registers can be programmed before the receiver clock is enabled.

Downloaded from Arrow.com.

After reset, the transmitter is disabled. The user must enable it by setting the TXEN bit in FLEX_US_CR. However, the transmitter registers can be programmed before being enabled.

The receiver and the transmitter can be enabled together or independently.

At any time, the software can perform a reset on the receiver or the transmitter of the USART by setting the corresponding bit, RSTRX and RSTTX respectively, in FLEX_US_CR. The software resets clear the status flag and reset internal state machines but the user interface configuration registers hold the value configured prior to software reset. Regardless of what the receiver or the transmitter is performing, the communication is immediately stopped.

The user can also independently disable the receiver or the transmitter by setting RXDIS and TXDIS respectively in FLEX_US_CR. If the receiver is disabled during a character reception, the USART waits until the end of reception of the current character, then the reception is stopped. If the transmitter is disabled while it is operating, the USART waits the end of transmission of both the current character and character being stored in the USART Transmit Holding Register (FLEX_US_THR). If a timeguard is programmed, it is handled normally.

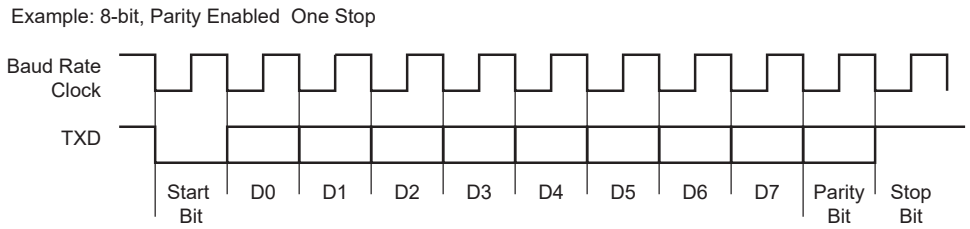
45.7.3 Synchronous and Asynchronous Modes

45.7.3.1 Transmitter Operations

The transmitter performs the same in both Synchronous and Asynchronous operating modes (SYNC = 0 or SYNC = 1). One start bit, up to 9 data bits, 1 optional parity bit and up to 2 stop bits are successively shifted out on the TXD pin at each falling edge of the programmed serial clock.

The number of data bits is selected by the CHRL field and the MODE9 bit in FLEX_US_MR. Nine bits are selected by setting the MODE9 bit regardless of the CHRL field. The parity bit is set according to the PAR field in FLEX_US_MR. The even, odd, space, marked or none parity bit can be configured. The MSBF bit in FLEX_US_MR configures which data bit is sent first. If written to 1, the most significant bit is sent first. If written to 0, the less significant bit is sent first. The number of stop bits is selected by the NBSTOP field in FLEX_US_MR. The 1.5 stop bit is supported in Asynchronous mode only.

Figure 45-5. Character Transmit

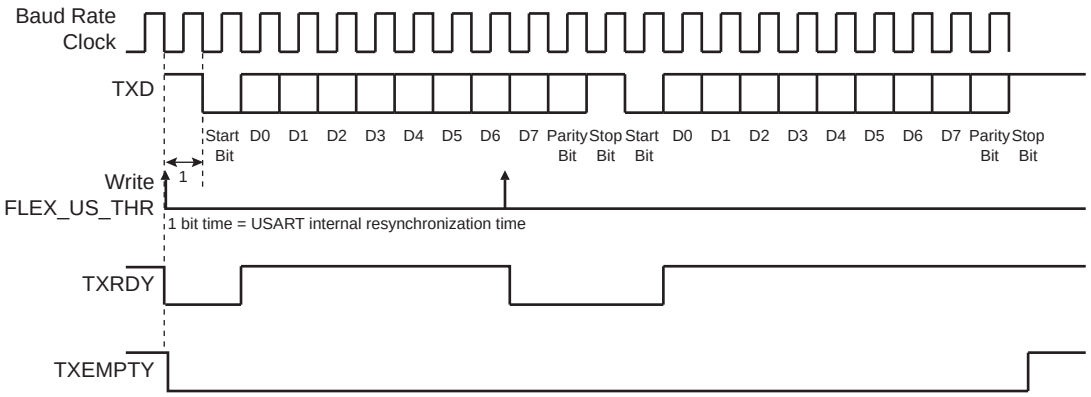


The characters are sent by writing in FLEX_US_THR. The transmitter reports two status bits in the USART Channel Status Register (FLEX_US_CSR): TXRDY (Transmitter Ready), which indicates that FLEX_US_THR is empty and TXEMPTY, which indicates that all the characters written in FLEX_US_THR have been processed. When the current character processing is completed, the last character written in FLEX_US_THR is transferred into the shift register of the transmitter and FLEX_US_THR is emptied, thus TXRDY rises.

Both TXRDY and TXEMPTY bits are low when the transmitter is disabled. Writing a character in FLEX_US_THR while TXRDY is low has no effect and the written character is lost.

Downloaded from Arrow.com.

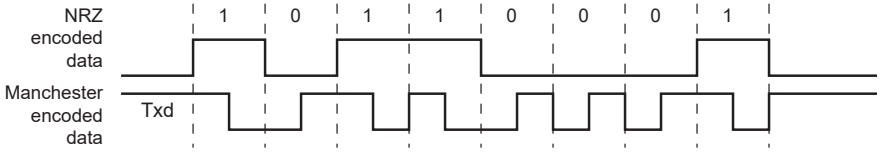
Figure 45-6. Transmitter Status



45.7.3.2 Manchester Encoder

When the Manchester encoder is in use, characters transmitted through the USART are encoded based on biphase Manchester II format. To enable this mode, set the FLEX_US_MR.MAN bit to 1. Depending on polarity configuration, a logic level (zero or one), is transmitted as a coded signal one-to-zero or zero-to-one. Thus, a transition always occurs at the midpoint of each bit time. It consumes more bandwidth than the original NRZ signal (2x) but the receiver has more error control since the expected input must show a change at the center of a bit cell. An example of Manchester encoded sequence is: the byte 0xB1 or 10110001 encodes to 10 01 10 10 01 01 01 10, assuming the default polarity of the encoder. The following figure illustrates this coding scheme.

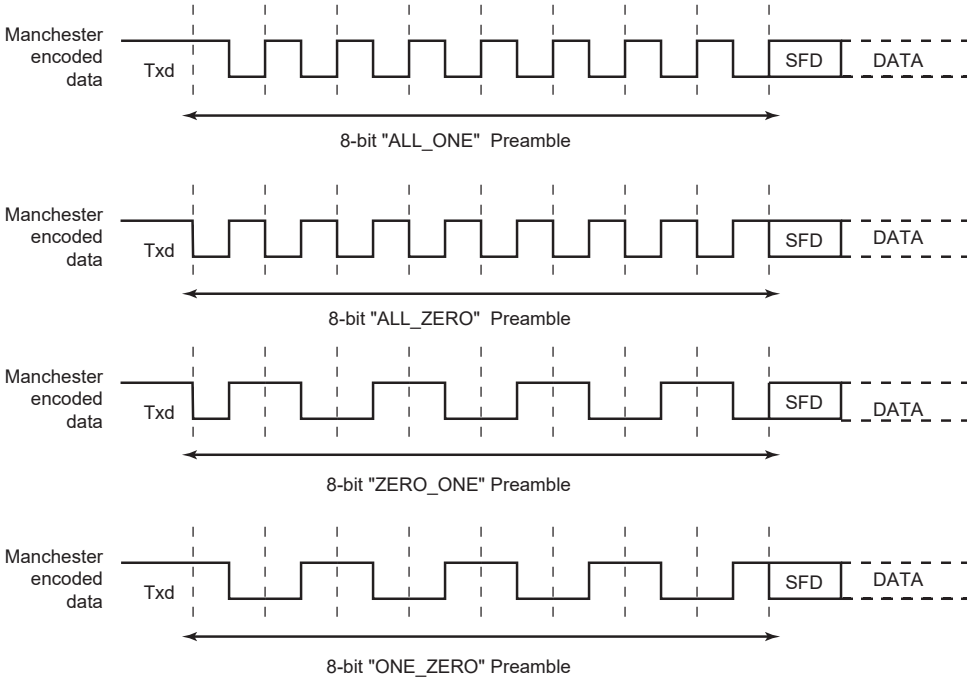
Figure 45-7. NRZ to Manchester Encoding



The Manchester encoded character can also be encapsulated by adding both a configurable preamble and a start frame delimiter pattern. Depending on the configuration, the preamble is a training sequence, composed of a predefined pattern with a programmable length from 1 to 15 bit times. If the preamble length is set to 0, the preamble waveform is not generated prior to any character. The preamble pattern is chosen among the following sequences: ALL_ONE, ALL_ZERO, ONE_ZERO or ZERO_ONE, writing the FLEX_US_MAN.TX_PP field. The TX_PL field is used to configure the preamble length. The following figure illustrates and defines the valid patterns. To improve flexibility, the encoding scheme can be configured using the FLEX_US_MAN.TX_MPOL bit. If the TX_MPOL bit is set to zero (default), a logic zero is encoded with a zero-to-one transition and a logic one is encoded with a one-to-zero transition. If the TX_MPOL bit is set to one, a logic one is encoded with a one-to-zero transition and a logic zero is encoded with a zero-to-one transition.

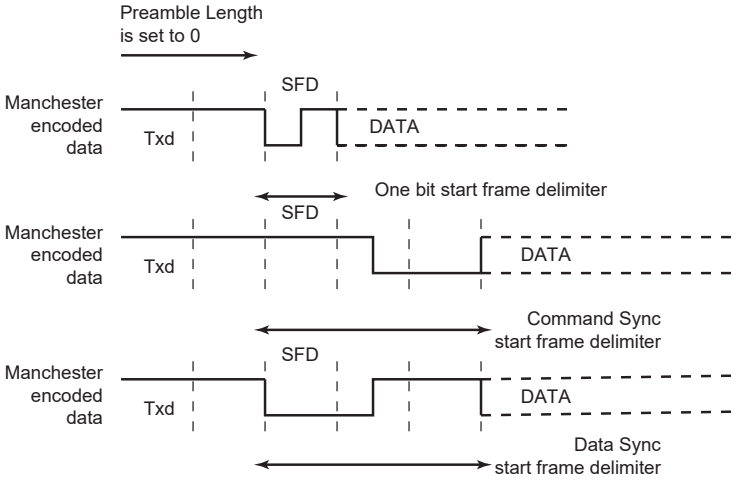
Downloaded from Arrow.com.

Figure 45-8. Preamble Patterns, Default Polarity Assumed



A start frame delimiter is to be configured using the FLEX_US_MR.ONEBIT bit. It consists of a user-defined pattern that indicates the beginning of a valid data. The following figure illustrates these patterns. If the start frame delimiter, also known as the start bit, is one bit, (ONEBIT = 1), a logic zero is Manchester encoded and indicates that a new character is being sent serially on the line. If the start frame delimiter is a synchronization pattern also referred to as sync (ONEBIT = 0), a sequence of three bit times is sent serially on the line to indicate the start of a new character. The sync waveform is in itself an invalid Manchester waveform as the transition occurs at the middle of the second bit time. Two distinct sync patterns are used: the command sync and the data sync. The command sync has a logic one level for one and a half bit times, then a transition to logic zero for the second one and a half bit times. If the FLEX_US_MR.MODSYNC bit is set to 1, the next character is a command. If it is set to 0, the next character is a data. When direct memory access is used, the MODSYNC bit can be immediately updated with a modified character located in memory. To enable this mode, the FLEX_US_MR.VAR_SYNC bit must be set. In this case, the FLEX_US_MR.MODSYNC bit is bypassed and the sync configuration is held in the FLEX_US_THR.TXSYNH bit. The USART character format is modified and includes sync information.

Figure 45-9. Start Frame Delimiter

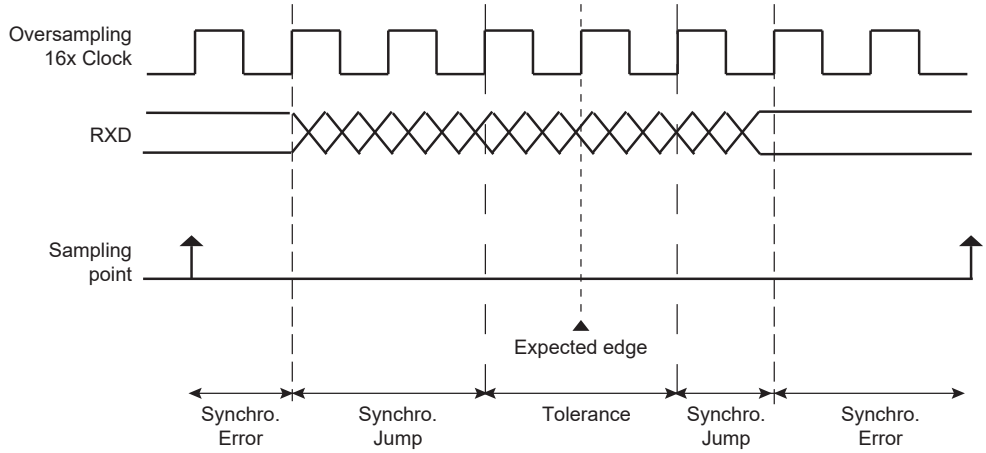


Downloaded from Arrow.com.

45.7.3.2.1 Drift Compensation

Drift compensation is available only in 16X Oversampling mode. An hardware recovery system allows a larger clock drift. To enable the hardware system, the bit in the FLEX_US_MAN register must be set. If the RXD edge is one 16X clock cycle from the expected edge, this is considered as normal jitter and no corrective actions is taken. If the RXD event is between 4 and 2 clock cycles before the expected edge, then the current period is shortened by one clock cycle. If the RXD event is between 2 and 3 clock cycles after the expected edge, then the current period is lengthened by one clock cycle. These intervals are considered to be drift and so corrective actions are automatically taken.

Figure 45-10. Bit Resynchronization



45.7.3.3 Asynchronous Receiver

If the USART is programmed in Asynchronous operating mode (SYNC = 0), the receiver oversamples the RXD input line. The oversampling is either 16 or 8 times the baud rate clock, depending on the FLEX_US_MR.OVER bit.

The receiver samples the RXD line. If the line is sampled during one half of a bit time to 0, a start bit is detected and data, parity and stop bits are successively sampled on the bit rate clock.

If the oversampling is 16 (OVER = 0), a start is detected at the eighth sample to 0. Data bits, parity bit and stop bit are assumed to have a duration corresponding to 16 oversampling clock cycles. If the oversampling is 8 (OVER = 1), a start bit is detected at the fourth sample to 0. Data bits, parity bit and stop bit are assumed to have a duration corresponding to 8 oversampling clock cycles.

The number of data bits, first bit sent and Parity mode are selected by the same fields and bits as the transmitter, i.e., respectively CHRL, MODE9, MSBF and PAR. For the synchronization mechanism only, the number of stop bits has no effect on the receiver as it considers only one stop bit, regardless of the NBSTOP field, so that resynchronization between the receiver and the transmitter can occur. Moreover, as soon as the stop bit is sampled, the receiver starts looking for a new start bit so that resynchronization can also be accomplished when the transmitter is operating with one stop bit.

The following figures illustrate start detection and character reception when USART operates in Asynchronous mode.

Downloaded from Arrow.com.

Figure 45-11. Asynchronous Start Detection

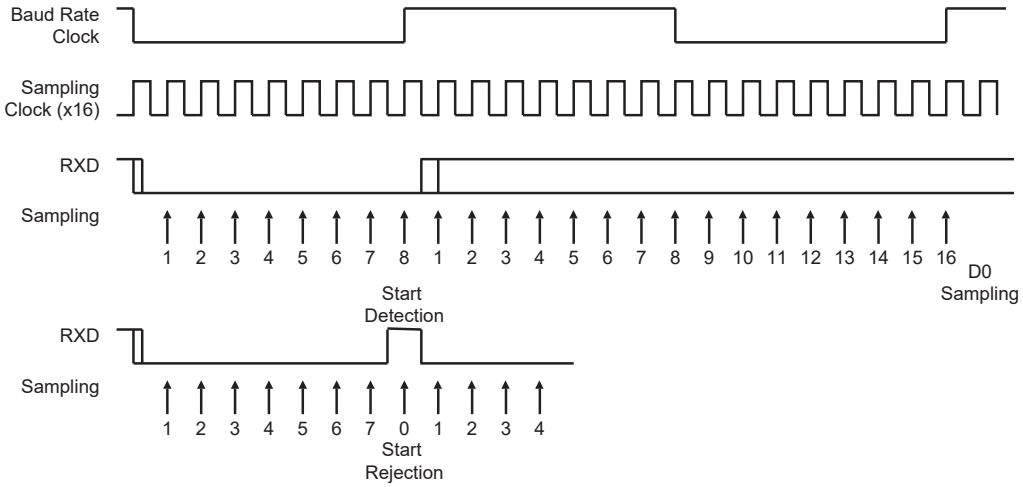
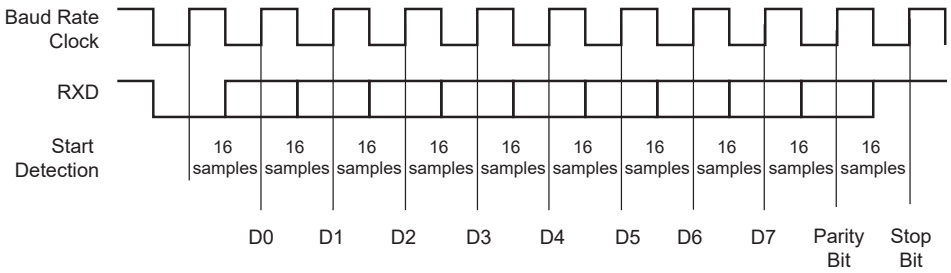


Figure 45-12. Asynchronous Character Reception

Example: 8-bit, Parity Enabled



45.7.3.4 Manchester Decoder

When the FLEX_US_MR.MAN bit is set, the Manchester decoder is enabled. The decoder performs both preamble and start frame delimiter detection. One input line is dedicated to Manchester encoded input data.

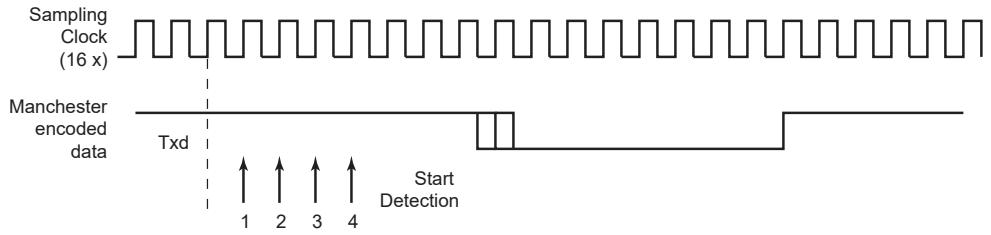
An optional preamble sequence can be defined. Its length is user-defined and totally independent of the transmitter side. Use the FLEX_US_MAN.RX_PL field to configure the length of the preamble sequence. If the length is set to 0, no preamble is detected and the function is disabled. In addition, the polarity of the input stream is programmable with the FLEX_US_MAN.RX_MPOL bit. Depending on the desired application, the preamble pattern matching is to be defined via the FLEX_US_MAN.RX_PP field. See figure Preamble Patterns, Default Polarity Assumed for available preamble patterns.

Unlike preamble, the start frame delimiter is shared between Manchester Encoder and Decoder. So, if ONEBIT bit = 1, only a zero encoded Manchester can be detected as a valid start frame delimiter. If ONEBIT = 0, only a sync pattern is detected as a valid start frame delimiter. Decoder operates by detecting transition on incoming stream. If RXD is sampled during one quarter of a bit time to zero, a start bit is detected. See the following figure. The sample pulse rejection mechanism applies.

The FLEX_US_MAN.RXIDLEV bit informs the USART of the receiver line idle state value (receiver line inactive). The user must define RXIDLEV to ensure reliable synchronization. By default, RXIDLEV is set to one (receiver line is at level 1 when there is no activity).

Downloaded from Arrow.com.

Figure 45-13. Asynchronous Start Bit Detection



The receiver is activated and starts preamble and frame delimiter detection, sampling the data at one quarter and then three quarters. If a valid preamble pattern or start frame delimiter is detected, the receiver continues decoding with the same synchronization. If the stream does not match a valid pattern or a valid start frame delimiter, the receiver resynchronizes on the next valid edge.The minimum time threshold to estimate the bit value is three quarters of a bit time.

If a valid preamble (if used) followed with a valid start frame delimiter is detected, the incoming stream is decoded into NRZ data and passed to USART for processing. The following figure illustrates Manchester pattern mismatch. When incoming data stream is passed to the USART, the receiver is also able to detect Manchester code violation. A code violation is a lack of transition in the middle of a bit cell. In this case, the MANE flag in FLEX_US_CSR is raised. It is cleared by writing a one to FLEX_US_CR.RSTSTA. See figure "Manchester Error Flag" below for an example of Manchester error detection during the data phase.

Figure 45-14. Preamble Pattern Mismatch

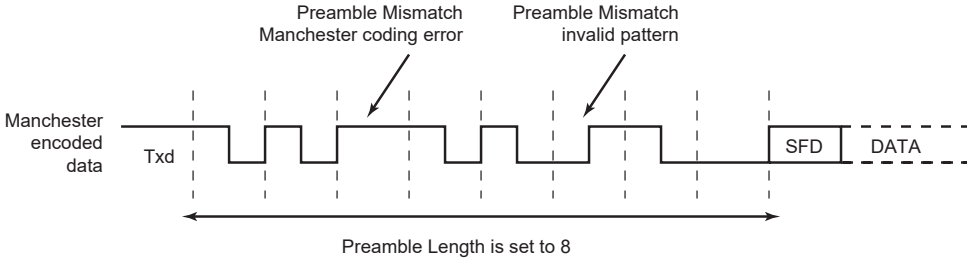
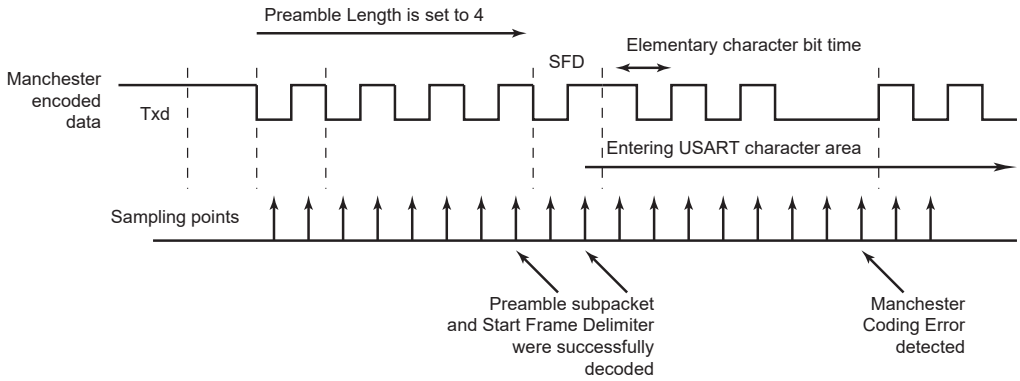


Figure 45-15. Manchester Error Flag



When the start frame delimiter is a sync pattern (ONEBIT = 0), both command and data delimiter are supported. If a valid sync is detected, the received character is written as RXCHR field in the Receive Holding Register (FLEX_US_RHR) and the RXSYNH is updated. RXCHR is set to 1 when the received character is a command, and it is set to 0 if the received character is a data. This mechanism alleviates and simplifies the direct memory access as the character contains its own sync field in the same register.

As the decoder is setup to be used in Unipolar mode, the first bit of the frame has to be a zero-to-one transition.

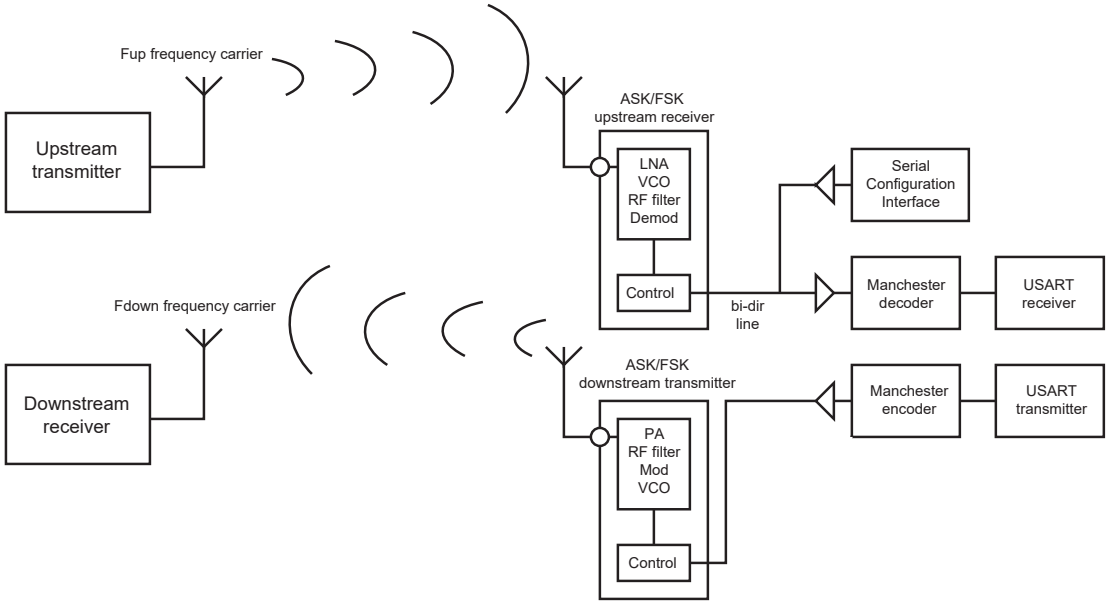
45.7.3.5 Radio Interface: Manchester Encoded USART Application

This section describes low data rate RF transmission systems and their integration with a Manchester encoded USART. These systems are based on transmitter and receiver ICs that support ASK and FSK modulation schemes.

Downloaded from Arrow.com.

The goal is to perform full-duplex radio transmission of characters using two different frequency carriers. See configuration in the following figure.

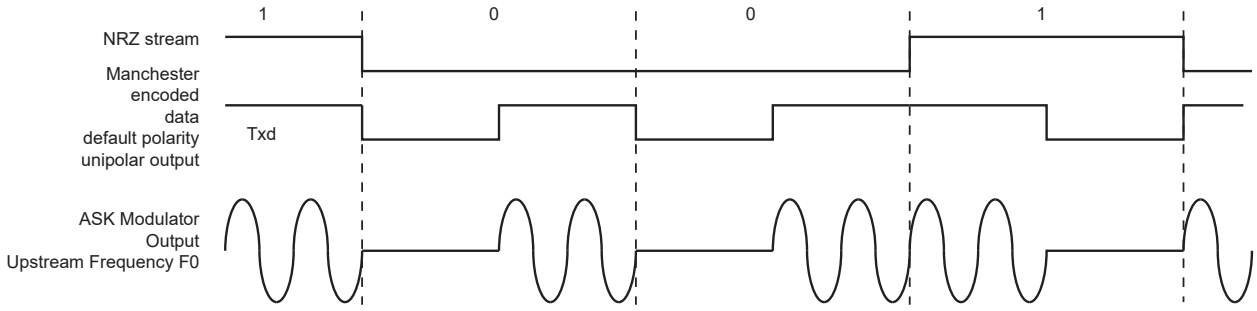
Figure 45-16. Manchester Encoded Characters RF Transmission



The USART peripheral is configured as a Manchester encoder/decoder. Looking at the downstream communication channel, Manchester encoded characters are serially sent to the RF transmitter. This may also include a user defined preamble and a start frame delimiter. Mostly, preamble is used in the RF receiver to distinguish between a valid data from a transmitter and signals due to noise. The Manchester stream is then modulated. See the following figure for an example of ASK modulation scheme. When a logic one is sent to the ASK modulator, the power amplifier, referred to as PA, is enabled and transmits an RF signal at downstream frequency. When a logic zero is transmitted, the RF signal is turned off. If the FSK modulator is activated, two different frequencies are used to transmit data. When a logic 1 is sent, the modulator outputs an RF signal at frequency F0 and switches to F1 if the data sent is a 0. See figure "FSK Modulator Output" below.

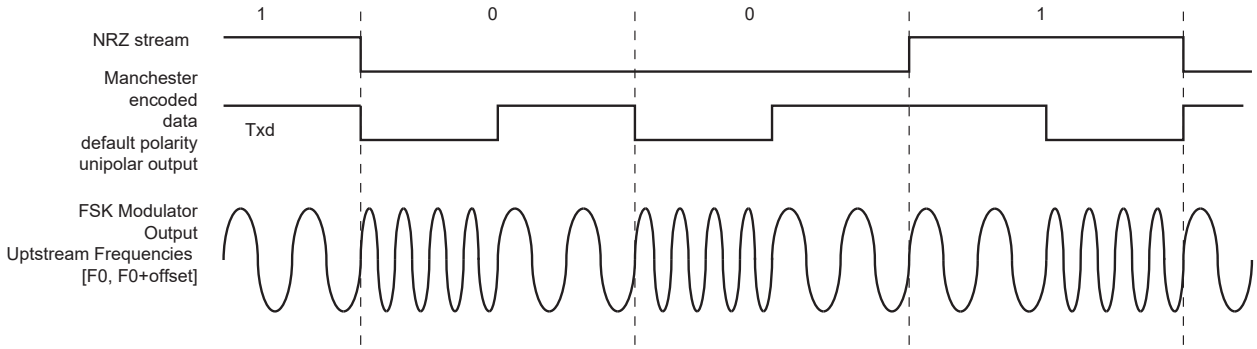
From the receiver side, another carrier frequency is used. The RF receiver performs a bit check operation examining demodulated data stream. If a valid pattern is detected, the receiver switches to Receiving mode. The demodulated stream is sent to the Manchester decoder. Because of bit checking inside RF IC, the data transferred to the microcontroller is reduced by a user-defined number of bits. The Manchester preamble length is to be defined in accordance with the RF IC configuration.

Figure 45-17. ASK Modulator Output



Downloaded from Arrow.com.

Figure 45-18. FSK Modulator Output



45.7.3.6 Synchronous Receiver

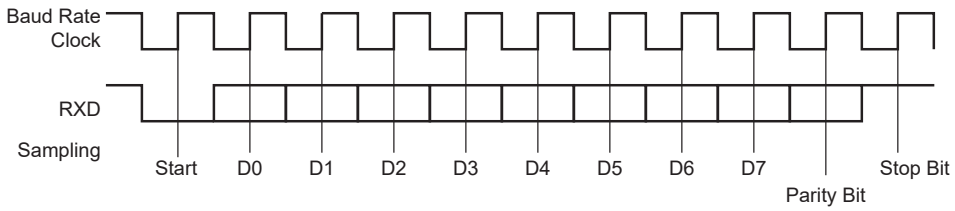
In Synchronous mode (SYNC = 1), the receiver samples the RXD signal on each rising edge of the baud rate clock. If a low level is detected, it is considered as a start. All data bits, the parity bit and the stop bits are sampled and the receiver waits for the next start bit. Synchronous mode operations provide a high-speed transfer capability.

Configuration fields and bits are the same as in Asynchronous mode.

The following figure illustrates a character reception in Synchronous mode.

Figure 45-19. Synchronous Mode Character Reception

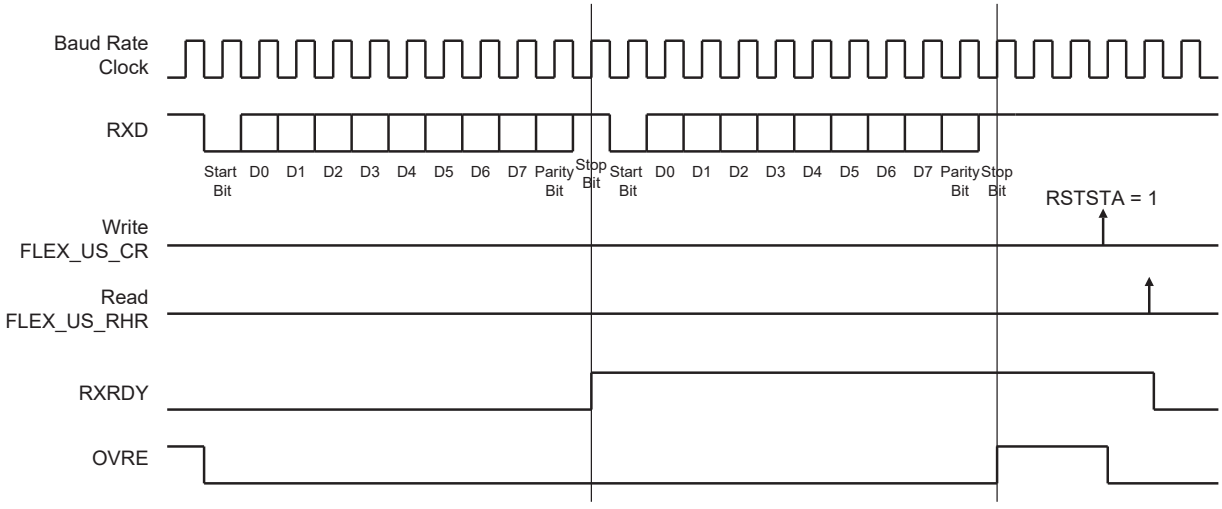
Example: 8-bit, Parity Enabled 1 Stop



45.7.3.7 Receiver Operations

When a character reception is completed, it is transferred to the Receive Holding Register (FLEX_US_RHR) and the FLEX_US_CSR.RXRDY bit is raised. If a character is completed while the RXRDY is set, the Overrun Error (OVRE) bit is set. The last character is transferred into FLEX_US_RHR and overwrites the previous one. The OVRE bit is cleared by writing a one to Reset Status bit FLEX_US_CR.RSTSTA.

Figure 45-20. Receiver Status



Downloaded from Arrow.com.

45.7.3.8 Parity

The USART supports five parity modes that are selected by writing to the FLEX_US_MR.PAR field. The PAR field also enables the Multidrop mode (see section Multidrop Mode). Even and odd parity bit generation and error detection are supported.

If even parity is selected, the parity generator of the transmitter drives the parity bit to 0 if a number of 1s in the character data bit is even, and to 1 if the number of 1s is odd. Accordingly, the receiver parity checker counts the number of received 1s and reports a parity error if the sampled parity bit does not correspond. If odd parity is selected, the parity generator of the transmitter drives the parity bit to 1 if a number of 1s in the character data bit is even, and to 0 if the number of 1s is odd. Accordingly, the receiver parity checker counts the number of received 1s and reports a parity error if the sampled parity bit does not correspond. If the mark parity is used, the parity generator of the transmitter drives the parity bit to 1 for all characters. The receiver parity checker reports an error if the parity bit is sampled to 0. If the space parity is used, the parity generator of the transmitter drives the parity bit to 0 for all characters. The receiver parity checker reports an error if the parity bit is sampled to 1. If parity is disabled, the transmitter does not generate any parity bit and the receiver does not report any parity error.

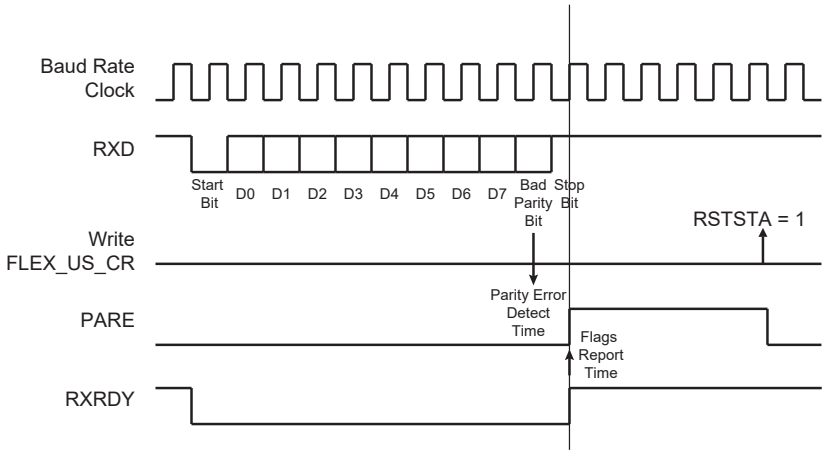
The following table shows an example of the parity bit for the character 0x41 (character ASCII “A”) depending on the configuration of the USART. Because there are two bits set to 1 in the character value, the parity bit is set to 1 when the parity is odd, or configured to 0 when the parity is even.

Table 45-7. Parity Bit Examples

Character	Hexadecimal	Binary	Parity Bit	Parity Mode
A	0x41	0100 0001	1	Odd
A	0x41	0100 0001	0	Even
A	0x41	0100 0001	1	Mark
A	0x41	0100 0001	0	Space
A	0x41	0100 0001	None	None

When the receiver detects a parity error, it sets the Parity Error bit FLEX_US_CSR.PARE. The PARE bit can be cleared by writing a one to the FLEX_US_CR.RSTSTA bit. The following figure illustrates the parity bit status setting and clearing.

Figure 45-21. Parity Error



45.7.3.9 Multidrop Mode

If the value 0x6 or 0x07 is written to the FLEX_US_MR.PAR field, the USART runs in Multidrop mode. This mode differentiates the data characters and the address characters. Data are transmitted with the parity bit to 0 and addresses are transmitted with the parity bit to 1.

Downloaded from Arrow.com.

If the USART is configured in Multidrop mode, the receiver sets the PARE parity error bit when the parity bit is high and the transmitter is able to send a character with the parity bit high when a one is written to the FLEX_US_CR.SENDA bit.

To handle parity error, the PARE bit is cleared by writing a one to the FLEX_US_CR.RSTSTA bit.

The transmitter sends an address byte (parity bit set) when the FLEX_US_CR.SENDA bit is written to 1. In this case, the next byte written to FLEX_US_THR is transmitted as an address. Any character written in FLEX_US_THR when the SENDA command is not written is transmitted normally with parity to 0.

45.7.3.10 Transmitter Timeguard

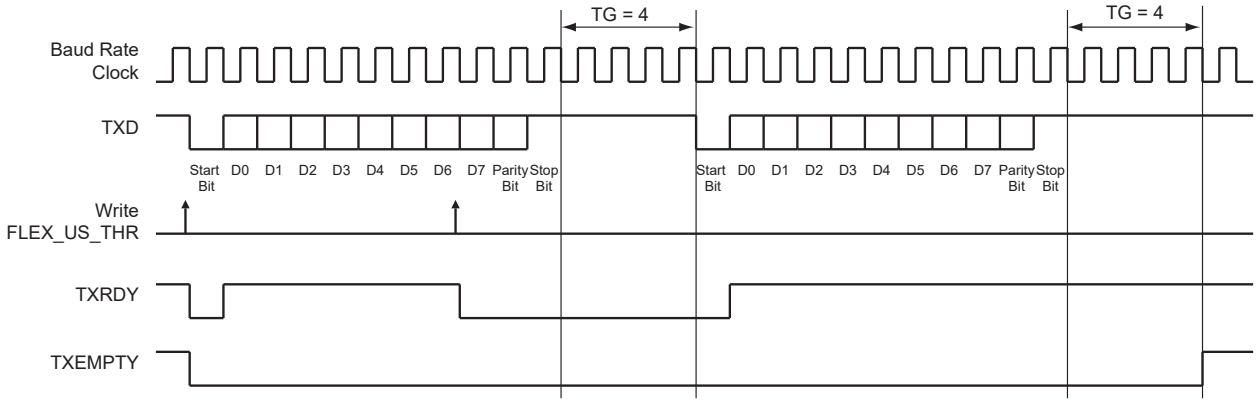
The timeguard feature enables the USART interface with slow remote devices.

The timeguard function enables the transmitter to insert an idle state on the TXD line between two characters. This idle state actually acts as a long stop bit.

The duration of the idle state is programmed in the TG field of the Transmitter Timeguard Register (FLEX_US_TTGR). When this field is written to zero, no timeguard is generated. Otherwise, the transmitter holds a high level on TXD after each transmitted byte during the number of bit periods programmed in TG in addition to the number of stop bits.

As illustrated in the following figure, the behavior of the TXRDY and TXEMPTY status bits is modified by the programming of a timeguard. TXRDY rises only when the start bit of the next character is sent, and thus remains to 0 during the timeguard transmission if a character has been written in FLEX_US_THR. TXEMPTY remains low until the timeguard transmission is completed as the timeguard is part of the current character being transmitted.

Figure 45-22. Timeguard Operations



The following table indicates the maximum length of a timeguard period that the transmitter can handle in relation to the function of the baud rate.

Table 45-8. Maximum Timeguard Length Depending on Baud Rate

Baud Rate (bit/s)	Bit Time (µs)	Timeguard (ms)
1,200	833	212.50
9,600	104	26.56
14,400	69.4	17.71
19,200	52.1	13.28
28,800	34.7	8.85
38,400	26	6.63
56,000	17.9	4.55
57,600	17.4	4.43
115,200	8.7	2.21

Downloaded from Arrow.com.

45.7.3.11 Receiver Timeout

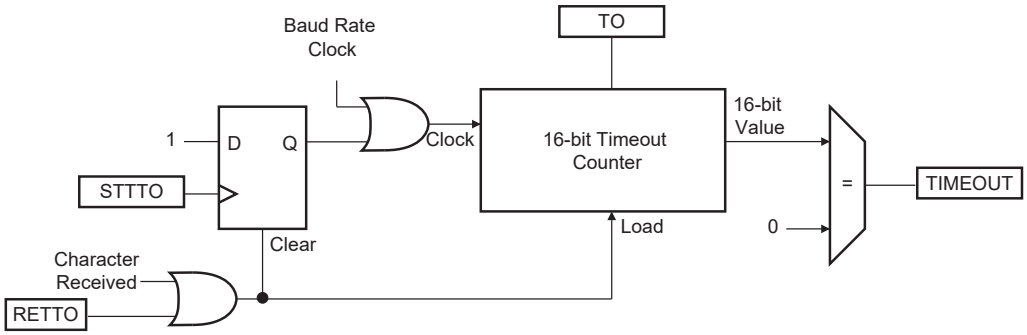
The Receiver Timeout provides support in handling variable-length frames. This feature detects an idle condition on the RXD line. When a timeout is detected, the FLEX_US_CSR.TIMEOUT bit rises and can generate an interrupt, thus indicating to the driver an end of frame.

The timeout delay period (during which the receiver waits for a new character) is programmed in the TO field of the Receiver Timeout Register (FLEX_US_RTOR). If the TO field is written to 0, the Receiver Timeout is disabled and no timeout is detected. The FLEX_US_CSR.TIMEOUT bit remains at 0. Otherwise, the receiver loads a 16-bit counter with the value programmed in TO. This counter is decremented at each bit period and reloaded each time a new character is received. If the counter reaches 0, the FLEX_US_CSR.TIMEOUT bit rises. Then, the user can either:

- Stop the counter clock until a new character is received. This is performed by writing a ‘1’ to FLEX_US_CR.STTTO. In this case, the idle state on RXD before a new character is received does not provide a timeout. This prevents having to handle an interrupt before a character is received and enables waiting for the next idle state on RXD after a frame is received.
- Obtain an interrupt while no character is received. This is performed by writing a ‘1’ to FLEX_US_CR.RETTO. In this case, the counter starts counting down immediately from the value TO. This generates a periodic interrupt so that a user timeout can be handled, for example when no key is pressed on a keyboard.

The following figure shows the block diagram of the Receiver Timeout feature.

Figure 45-23. Receiver Timeout Block Diagram



The following table gives the maximum timeout period for some standard baud rates.

Table 45-9. Maximum Timeout Period

Baud Rate (bit/s)	Bit Time (µs)	Timeout (ms)
600	1,667	109,225
1,200	833	54,613
2,400	417	27,306
4,800	208	13,653
9,600	104	6,827
14,400	69	4,551
19,200	52	3,413
28,800	35	2,276
38,400	26	1,704
56,000	18	1,170
57,600	17	1,138
200,000	5	328

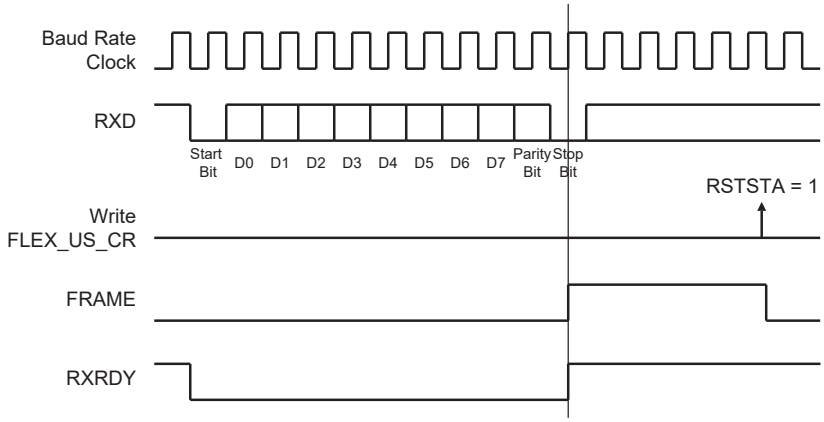
Downloaded from Arrow.com.

45.7.3.12 Framing Error

The receiver is capable of detecting framing errors. A framing error happens when the stop bit of a received character is detected at level 0. This can occur if the receiver and the transmitter are fully desynchronized.

A framing error is reported on the FLEX_US_CSR.FRAME bit. The FRAME bit is asserted in the middle of the stop bit as soon as the framing error is detected. It is cleared by writing a one to the FLEX_US_CR.RSTSTA bit.

Figure 45-24. Framing Error Status



45.7.3.13 Transmit Break

The user can request the transmitter to generate a break condition on the TXD line. A break condition drives the TXD line low during at least one complete character. It appears the same as a 0x00 character sent with the parity and the stop bits to 0. However, the transmitter holds the TXD line at least during one character until the user requests the break condition to be removed.

A break is transmitted by setting the FLEX_US_CR.STTBRK bit. This can be done at any time, either while the transmitter is empty (no character in either the shift register or in FLEX_US_THR) or when a character is being transmitted. If a break is requested while a character is being shifted out, the character is first completed before the TXD line is held low.

Once the Start Break command is requested, further Start Break commands are ignored until the end of the break is completed.

The break condition is removed by setting the FLEX_US_CR.STPBRK bit. If the Stop Break command is requested before the end of the minimum break duration (one character, including start, data, parity and stop bits), the transmitter ensures that the break condition completes.

The transmitter considers the break as though it is a character, i.e., the Start Break and Stop Break commands are processed only if the FLEX_US_CSR.TXRDY bit = 1 and the start of the break condition clears the TXRDY and TXEMPTY bits as if a character was processed.

Setting both the FLEX_US_CR.STTBRK and FLEX_US_CR.STPBRK bits can lead to an unpredictable result. All Stop Break commands requested without a previous Start Break command are ignored. A byte written into the Transmit Holding register while a break is pending, but not started, is ignored.

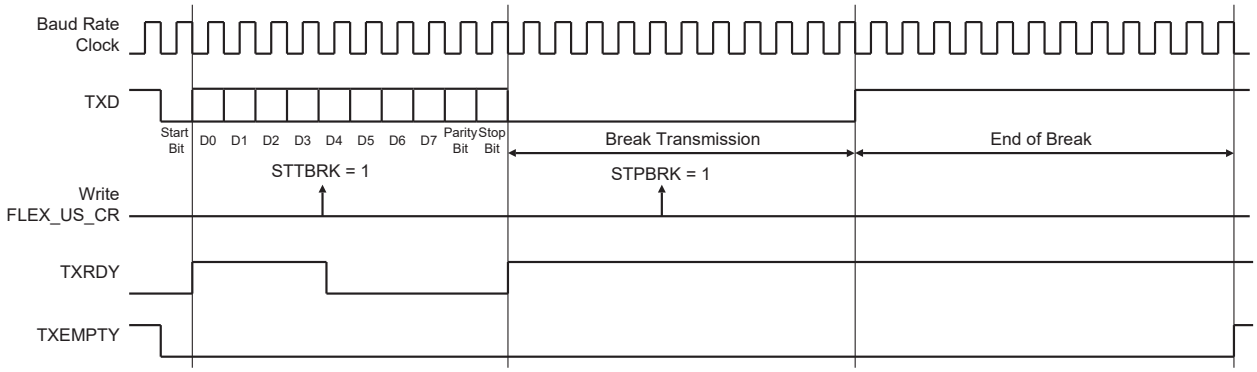
After the break condition, the transmitter returns the TXD line to 1 for a minimum of 12 bit times. Thus, the transmitter ensures that the remote receiver detects correctly the end of break and the start of the next character. If the timeguard is programmed with a value higher than 12, the TXD line is held high for the timeguard period.

After holding the TXD line for this period, the transmitter resumes normal operations.

The following figure illustrates the effect of both the Start Break (STTBRK) and Stop Break (STPBRK) commands on the TXD line.

Downloaded from Arrow.com.

Figure 45-25. Break Transmission



45.7.3.14 Receive Break

The receiver detects a break condition when all data, parity and stop bits are low. This corresponds to detecting a framing error with data to 0x00, but FRAME remains low.

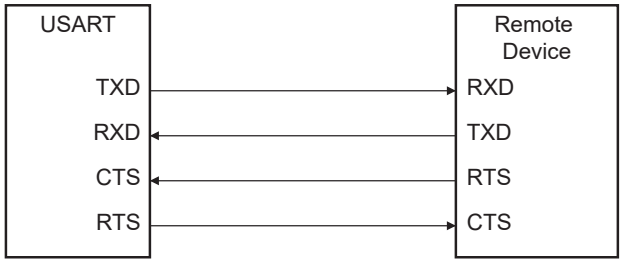
When the low stop bit is detected, the receiver asserts the FLEX_US_CSR.RXBRK bit. FLEX_US_CSR.RXBRK may be cleared by setting the FLEX_US_CR.RSTSTA bit.

An end of receive break is detected by a high level for at least 2/16ths of a bit period in Asynchronous operating mode or one sample at high level in Synchronous operating mode. The end of break detection also asserts the RXBRK bit.

45.7.3.15 Hardware Handshaking

The USART features a hardware handshaking out-of-band flow control. The RTS and CTS pins are used to connect with the remote device, as shown in the following figure.

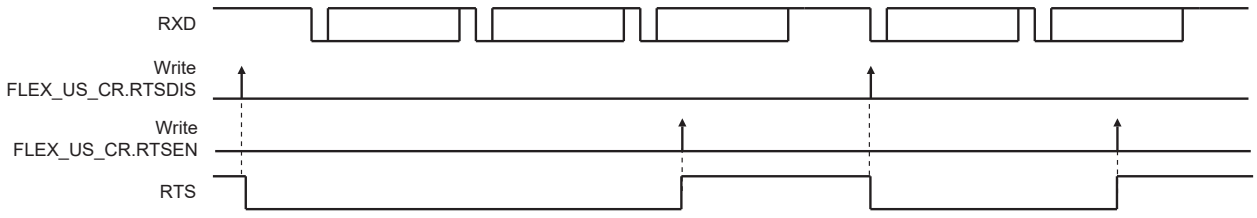
Figure 45-26. Connection with a Remote Device for Hardware Handshaking



Setting the USART to operate with hardware handshaking is performed by writing the FLEX_US_MR.USART_MODE field to the value 0x2.

The USART behavior when hardware handshaking is enabled is the same as the behavior in standard Synchronous or Asynchronous mode, except that the receiver drives the RTS pin as described below and the level on the CTS pin modifies the behavior of the transmitter as described below. Using this mode requires using the DMAC channel for reception. The transmitter can handle hardware handshaking in any case.

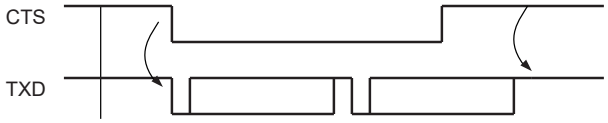
Figure 45-27. RTS Line Software Control when FLEX_US_MR.USART_MODE = 2



The following figure shows how the transmitter operates if hardware handshaking is enabled. The CTS pin disables the transmitter. If a character is being processed, the transmitter is disabled only after the completion of the current character and transmission of the next character happens as soon as the pin CTS falls.

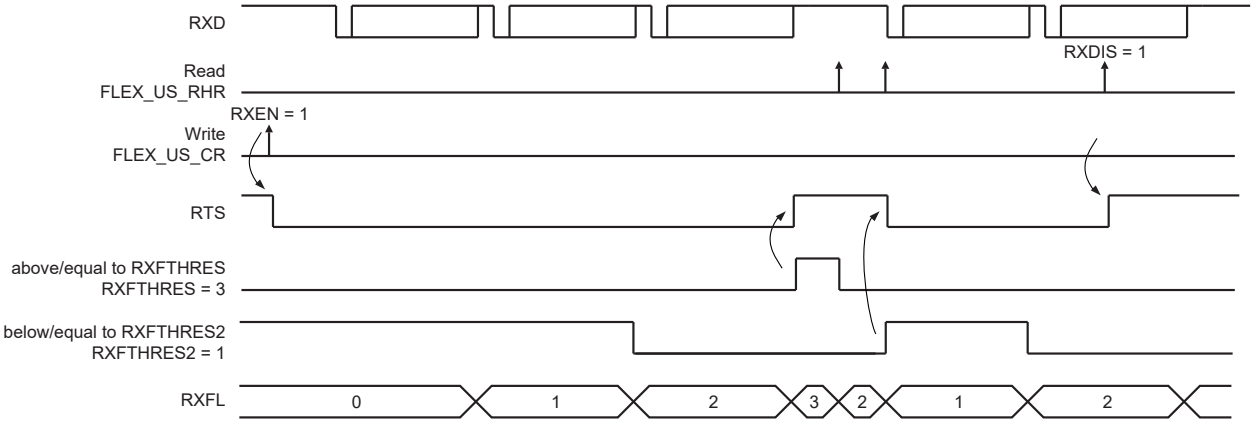
Downloaded from Arrow.com.

Figure 45-28. Transmitter Behavior when Operating with Hardware Handshaking



If USART FIFOs are enabled (bit FLEX_US_CR.FIFOEN), the RTS pin can be controlled by the USART Receive FIFO thresholds. The RTS pin control through Receive FIFO thresholds can be activated with the FLEX_US_FMR.FRTSC bit. Once activated, the RTS pin will be controlled by Receive FIFO thresholds, set to level 1 each time RXFTHRES is reached and set to level '0' each time RXFTHRES2 is reached (and RXFTHRES is not reached).

Figure 45-29. Receiver Behavior When FIFO Enabled and FRTSC Set to '1'



Note: In this mode, RXFTHRES must be > RXFTHRES2.

45.7.4 ISO7816 Mode

The USART features an ISO7816-compatible operating mode. This mode permits interfacing with smart cards and Security Access Modules (SAM) communicating through an ISO7816 link. Both T = 0 and T = 1 protocols defined by the ISO7816 specification are supported.

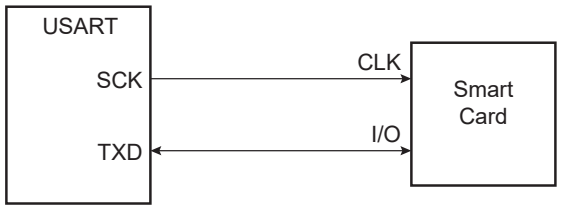
Setting the USART in ISO7816 mode is performed by writing the FLEX_US_MR.USART_MODE field to the value 0x4 for protocol T = 0 and to the value 0x6 for protocol T = 1.

45.7.4.1 ISO7816 Mode Overview

The ISO7816 is a half-duplex communication on only one bidirectional line. The baud rate is determined by a division of the clock provided to the remote device (see figure in section Baud Rate Generator).

The USART connects to a smart card as shown in the following figure. The TXD line becomes bidirectional and the baud rate generator feeds the ISO7816 clock on the SCK pin. As the TXD pin becomes bidirectional, its output remains driven by the output of the transmitter but only when the transmitter is active while its input is directed to the input of the receiver. The USART is considered as the master of the communication as it generates the clock.

Figure 45-30. Connection of a Smart Card to the USART



When operating in ISO7816, either in T = 0 or T = 1 modes, the character format is partially predefined. The configuration is forced to 8 data bits, and 1 or 2 stop bits, regardless of the values programmed in the CHRL, MODE9 and CHMODE fields. MSBF can be used to transmit LSB or MSB first. The bit INVDATA can be used to transmit in Normal or Inverse mode.

Downloaded from Arrow.com.

The USART cannot operate concurrently in both Receiver and Transmitter modes as the communication is unidirectional at a time. It has to be configured according to the required mode by enabling or disabling either the receiver or the transmitter as desired. Enabling both the receiver and the transmitter at the same time in ISO7816 mode may lead to unpredictable results.

The ISO7816 specification defines an inverse transmission format. Data bits of the character must be transmitted on the I/O line at their negative value.

45.7.4.2 Protocol T = 0

In T = 0 protocol, a character is made up of 1 start bit, 8 data bits, 1 parity bit and 1 guard time, which lasts two bit times. The transmitter shifts out the bits and does not drive the I/O line during the guard time.

If no parity error is detected, the I/O line remains at 1 during the guard time and the transmitter can continue with the transmission of the next character, as shown in the following figure.

If a parity error is detected by the receiver, it drives the I/O line to 0 during the guard time, as shown in figure "T = 0 Protocol with Parity Error" below. This error bit is also named NACK, for Non Acknowledge. In this case, the character lasts 1 bit time more, as the guard time length is the same and is added to the error bit time which lasts 1 bit time.

When the USART is the receiver and it detects an error, it does not load the erroneous character in the Receive Holding Register (FLEX_US_RHR). It appropriately sets the PARE bit in the Status Register (FLEX_US_CSR) so that the software can handle the error.

Figure 45-31. T = 0 Protocol without Parity Error

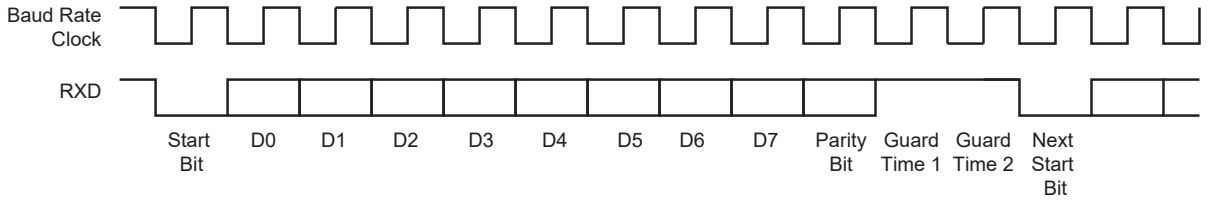
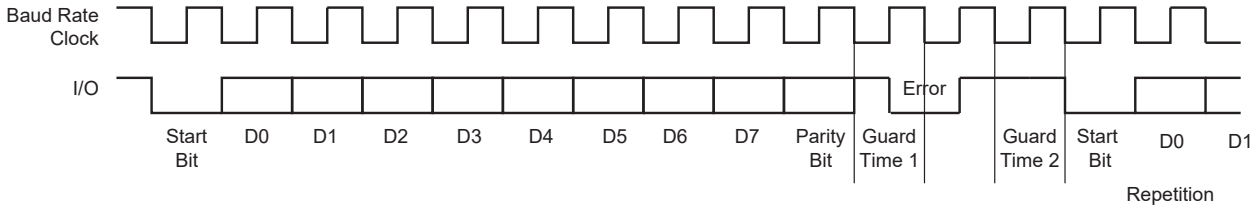


Figure 45-32. T = 0 Protocol with Parity Error



45.7.4.2.1 Receive Error Counter

The USART receiver also records the total number of errors. This can be read in the Number of Error (FLEX_US_NER) register. The NB_ERRORS field can record up to 255 errors. Reading FLEX_US_NER automatically clears the NB_ERRORS field.

45.7.4.2.2 Receive NACK Inhibit

The USART can be configured to inhibit an error. This is done by writing a ‘1’ to FLEX_US_MR.INACK. In this case, no error signal is driven on the I/O line even if a parity bit is detected.

Moreover, if INACK = 1, the erroneous received character is stored in the Receive Holding register as if no error occurred, and the RXRDY bit rises.

45.7.4.2.3 Transmit Character Repetition

When the USART is transmitting a character and gets a NACK, it can automatically repeat the character before moving on to the next one. Repetition is enabled by writing the FLEX_US_MR.MAX_ITERATION field at a value higher than 0. Each character can be transmitted up to eight times: the first transmission plus seven repetitions.

If MAX_ITERATION does not equal zero, the USART repeats the character as many times as the value loaded in MAX_ITERATION.

Downloaded from Arrow.com.

When the USART repetition number reaches MAX_ITERATION, and the last repeated character is not acknowledged, the FLEX_US_CSR.ITER bit is set. If the repetition of the character is acknowledged by the receiver, the repetitions are stopped and the iteration counter is cleared.

The FLEX_US_CSR.ITER bit can be cleared by writing the FLEX_US_CR.RSTIT bit to 1.

45.7.4.2.4 Disable Successive Receive NACK

The receiver can limit the number of successive NACKs sent back to the remote transmitter. This is programmed by setting the FLEX_US_MR.DSNACK bit. The maximum number of NACKs transmitted is programmed in the MAX_ITERATION field. As soon as MAX_ITERATION is reached, no error signal is driven on the I/O line and the FLEX_US_CSR.ITER bit is set.

45.7.4.3 Protocol T = 1

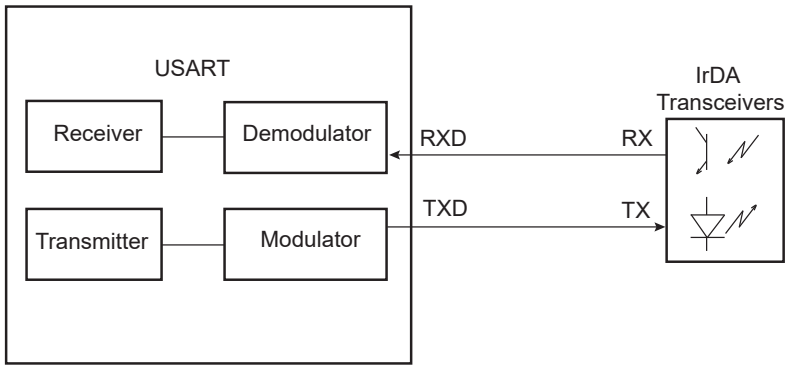
When operating in ISO7816 protocol T = 1, the transmission is similar to an asynchronous format with only one stop bit. The parity is generated when transmitting and checked when receiving. Parity error detection sets the FLEX_US_CSR.PARE bit.

45.7.5 IrDA Mode

The USART features an IrDA mode supplying half-duplex point-to-point wireless communication. It embeds the modulator and demodulator which allows a glueless connection to the infrared transceivers, as shown in the following figure. The modulator and demodulator are compliant with the IrDA specification version 1.1 and support data transfer speeds ranging from 2.4 kbit/s to 115.2 kbit/s.

The USART IrDA mode is enabled by setting the FLEX_US_MR.USART_MODE field to the value 0x8. The IrDA Filter Register (FLEX_US_IF) allows configuring the demodulator filter. The USART transmitter and receiver operate in a normal Asynchronous mode and all parameters are accessible. Note that the modulator and the demodulator are activated.

Figure 45-33. Connection to IrDA Transceivers



The receiver and the transmitter must be enabled or disabled according to the direction of the transmission to be managed.

To receive IrDA signals, the following needs to be done:

- Disable TX and Enable RX
- Configure the TXD pin as PIO and set it as an output to 0 (to avoid LED transmission). Disable the internal pullup (better for power consumption).
- Receive data

45.7.5.1 IrDA Modulation

For baud rates up to and including 115.2 kbit/s, the RZI modulation scheme is used. “0” is represented by a light pulse of 3/16th of a bit time. Some examples of signal pulse duration are shown in the following table.

Table 45-10. IrDA Pulse Duration

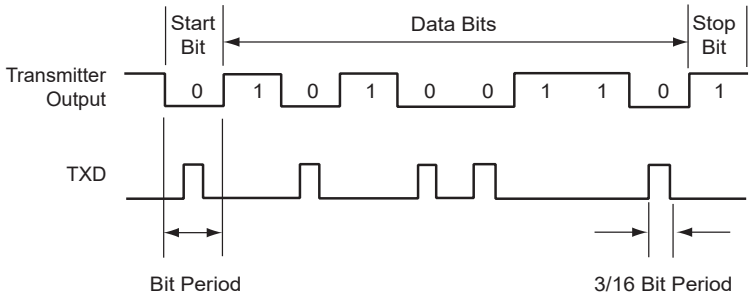
Baud Rate	Pulse Duration (3/16)
2.4 kbit/s	78.13 µs

Downloaded from Arrow.com.

.....continued	
Baud Rate	**Pulse Duration (3/16)**
9.6 kbit/s	19.53 µs
19.2 kbit/s	9.77 µs
38.4 kbit/s	4.88 µs
57.6 kbit/s	3.26 µs
115.2 kbit/s	1.63 µs

The following figure shows an example of character transmission.

Figure 45-34. IrDA Modulation



45.7.5.2 IrDA Baud Rate

The following table gives some examples of CD values, baud rate error and pulse duration. Note that the requirement on the maximum acceptable error of ±1.87% must be met.

Table 45-11. IrDA Baud Rate Error

Peripheral Clock	Baud Rate (bit/s)	CD	Baud Rate Error	Pulse Time (µs)
3,686,400	115,200	2	0.00%	1.63
20,000,000	115,200	11	1.38%	1.63
32,768,000	115,200	18	1.25%	1.63
40,000,000	115,200	22	1.38%	1.63
3,686,400	57,600	4	0.00%	3.26
20,000,000	57,600	22	1.38%	3.26
32,768,000	57,600	36	1.25%	3.26
40,000,000	57,600	43	0.93%	3.26
3,686,400	38,400	6	0.00%	4.88
20,000,000	38,400	33	1.38%	4.88
32,768,000	38,400	53	0.63%	4.88
40,000,000	38,400	65	0.16%	4.88
3,686,400	19,200	12	0.00%	9.77
20,000,000	19,200	65	0.16%	9.77
32,768,000	19,200	107	0.31%	9.77
40,000,000	19,200	130	0.16%	9.77

Downloaded from Arrow.com.

.....continued

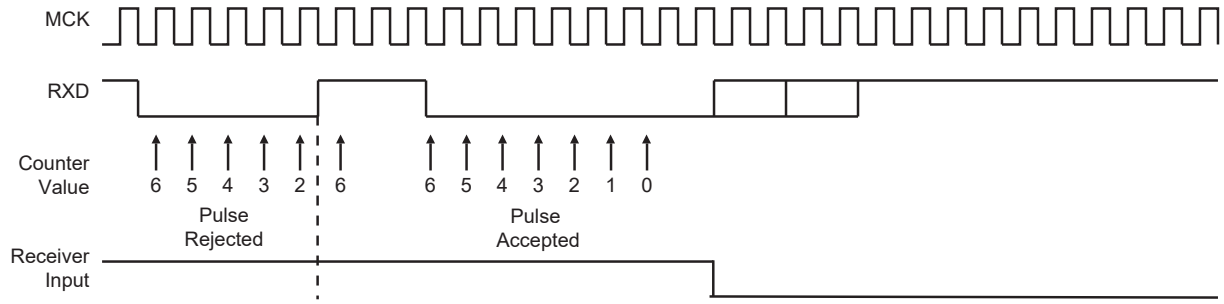
Peripheral Clock	Baud Rate (bit/s)	CD	Baud Rate Error	Pulse Time (µs)
3,686,400	9,600	24	0.00%	19.53
20,000,000	9,600	130	0.16%	19.53
32,768,000	9,600	213	0.16%	19.53
40,000,000	9,600	260	0.16%	19.53
3,686,400	2,400	96	0.00%	78.13
20,000,000	2,400	521	0.03%	78.13
32,768,000	2,400	853	0.04%	78.13

45.7.5.3 IrDA Demodulator

The demodulator is based on the IrDA Receive filter comprised of an 8-bit down counter which is loaded with the value programmed in FLEX_US_IF. When a falling edge is detected on the RXD pin, the Filter Counter starts counting down at the peripheral clock speed. If a rising edge is detected on the RXD pin, the counter stops and is reloaded with FLEX_US_IF. If no rising edge is detected when the counter reaches 0, the input of the receiver is driven low during one bit time.

The following figure illustrates the operations of the IrDA demodulator.

Figure 45-35. IrDA Demodulator Operations



The programmed value in the FLEX_US_IF register must always meet the following criteria:

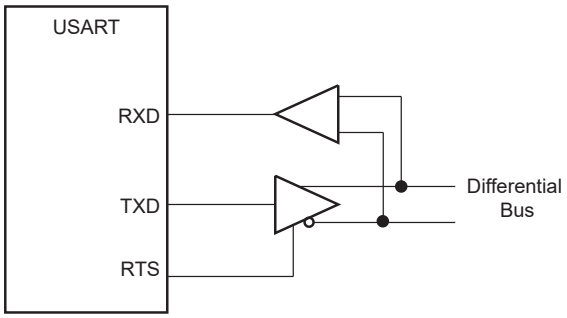
$t_{peripheral\ clock} \times (IRDA_FILTER + 3) < 1.41\ \mu s$

As the IrDA mode uses the same logic as the ISO7816, note that the FLEX_US_FIDI.FI_DI_RATIO field must be set to a value higher than 0 to make sure IrDA communications operate correctly.

45.7.6 RS485 Mode

The USART features the RS485 mode to enable line driver control. While operating in RS485 mode, the USART behaves as though in Asynchronous or Synchronous mode and configuration of all the parameters is possible. The difference is that the RTS pin is driven high when the transmitter is operating. The behavior of the RTS pin is controlled by the TXEMPTY bit. A typical connection of the USART to an RS485 bus is shown in the following figure.

Figure 45-36. Typical Connection to an RS485 Bus

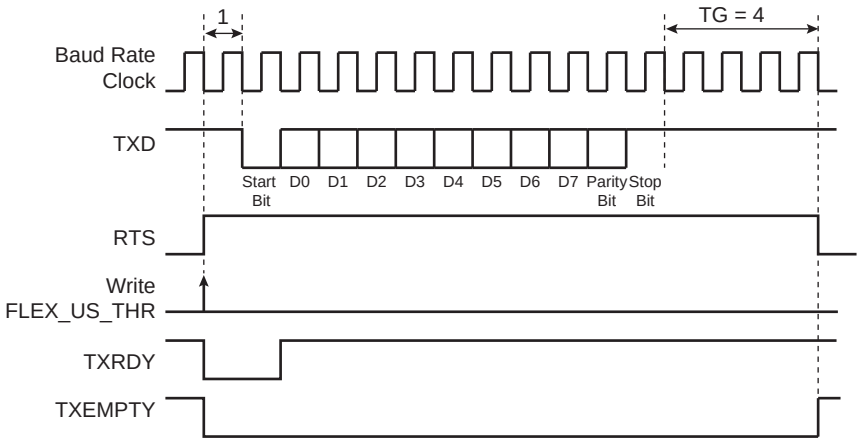


Downloaded from Arrow.com.

The USART is set in RS485 mode by writing the value 0x1 to the FLEX_US_MR.USART_MODE field.

The RTS pin is at a level inverse to the TXEMPTY bit. Significantly, the RTS pin remains high when a timeguard is programmed, so that the line can remain driven after the last character completion. The following figure gives an example of the RTS waveform during a character transmission when the timeguard is enabled.

Figure 45-37. Example of RTS Drive with Timeguard



45.7.7 USART Comparison Function on Received Character

The CMP flag in FLEX_US_CSR is set when the received character matches the conditions programmed in FLEX_US_CMPR. The CMP flag is set as soon as FLEX_US_RHR is loaded with the new received character. The CMP flag is cleared by writing a one to FLEX_US_CR.RSTSTA.

FLEX_US_CMPR can be programmed to provide different comparison methods:

- If VAL1 equals VAL2, then the comparison is performed on a single value and the flag is set to 1 if the received character equals VAL1.
- If VAL1 is strictly lower than VAL2, then any value between VAL1 and VAL2 sets the CMP flag.
- If VAL1 is strictly higher than VAL2, then the flag CMP is set to 1 if any received character equals VAL1 or VAL2.

When the FLEX_US_CMPR.CMPMODE bit is set to FLAG_ONLY (value 0), all received data are loaded in FLEX_US_RHR and the CMP flag provides the status of the comparison result.

By programming the START_CONDITION.CMPMODE bit (value 1), the comparison function result triggers the start of the loading of FLEX_US_RHR (see the following figure). The trigger condition exists as soon as the received character value matches the condition defined by the programming of VAL1, VAL2 and CMPPAR in FLEX_US_CMPR. The comparison trigger event is restarted by writing a 1 to the FLEX_US_CR.REQCLR bit.

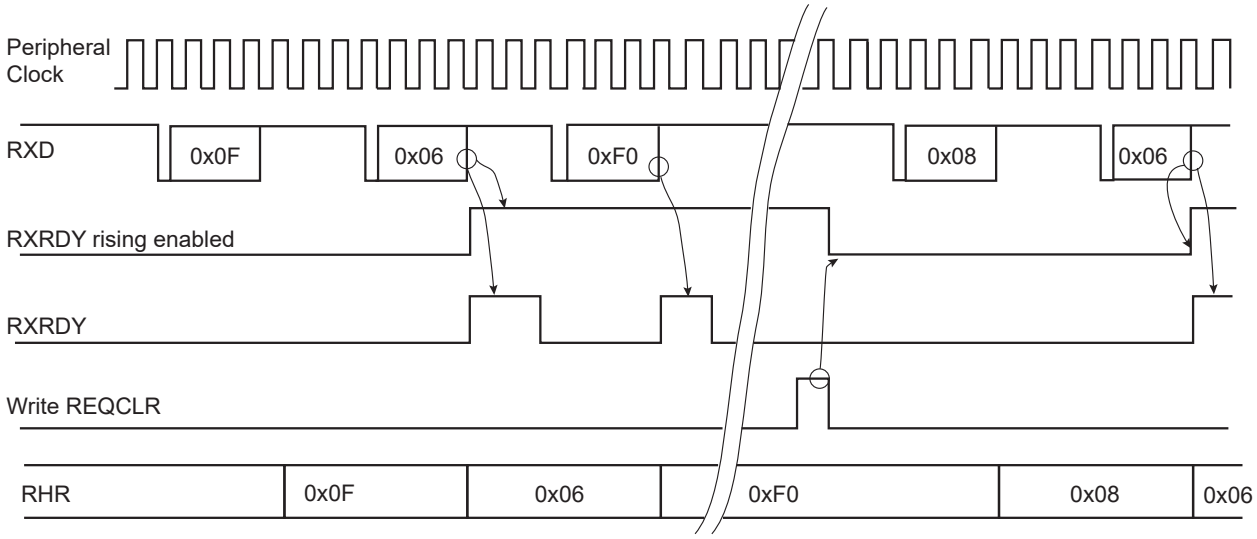
By setting the CMPMODE bit to FILTER (value 2), the comparison result triggers the start of the US_RHR loading. The trigger condition exists as soon as the received address byte value matches the conditions defined by VAL1, VAL2 in US_CMPR. The comparison trigger event is restarted automatically after the reception of the data byte. This comparison mode is only available when FLEX_US_MR.USART_MODE is set to DATA16BIT _SLAVE_MODE (value 13).

The value programmed in the VAL1 and VAL2 fields must not exceed the maximum value of the received character (see CHRL field in register 45.10.5 FLEX_US_MR).

Downloaded from Arrow.com.

Figure 45-38. Receive Holding Register Management

CMPMODE = 1, VAL1 = VAL2 = 0x06



45.7.8 LIN Mode

The LIN mode provides master node and slave node connectivity on a LIN bus.

The LIN (Local Interconnect Network) is a serial communication protocol which efficiently supports the control of mechatronic nodes in distributed automotive applications.

The main properties of the LIN bus are:

- Single master/multiple slaves concept
- Low-cost silicon implementation based on common UART/SCI interface hardware, an equivalent in software, or as a pure state machine.
- Self synchronization without quartz or ceramic resonator in the slave nodes
- Deterministic signal transmission
- Low cost single-wire implementation
- Speed up to 20 kbit/s

LIN provides cost efficient bus communication where the bandwidth and versatility of CAN are not required.

The LIN mode enables processing LIN frames with a minimum of action from the microprocessor.

45.7.8.1 Modes of Operation

The USART can act either as a LIN master node or as a LIN slave node.

The node configuration is chosen by setting the USART_MODE field in the USART Mode Register (FLEX_US_MR):

- LIN master node (USART_MODE = 0xA)
- LIN slave node (USART_MODE = 0xB)

In order to avoid unpredictable behavior, any change of the LIN node configuration must be followed by a software reset of the transmitter and of the receiver (except the initial node configuration after a hardware reset). (See section 45.7.2 Receiver and Transmitter Control.)

45.7.8.2 Baud Rate Configuration

See section Baud Rate in Asynchronous Mode.

- LIN master node: The baud rate is configured in FLEX_US_BRGR.
- LIN slave node: The initial baud rate is configured in FLEX_US_BRGR. This configuration is automatically copied in the LIN Baud Rate Register (FLEX_US_LINBRR) when writing FLEX_US_BRGR. After the synchronization procedure, the baud rate is updated in FLEX_US_LINBRR.

Downloaded from Arrow.com.

45.7.8.3 Receiver and Transmitter Control

See section 45.7.2 Receiver and Transmitter Control.

45.7.8.4 Character Transmission

See section Transmitter Operations.

45.7.8.5 Character Reception

See section Receiver Operations.

45.7.8.6 Header Transmission (Master Node Configuration)

All the LIN Frames start with a header which is sent by the master node and consists of a Synch Break Field, Synch Field and Identifier Field.

So in master node configuration, the frame handling starts with the sending of the header.

The header is transmitted as soon as the identifier is written in the LIN Identifier Register (FLEX_US_LINIR). At this moment the flag TXRDY falls.

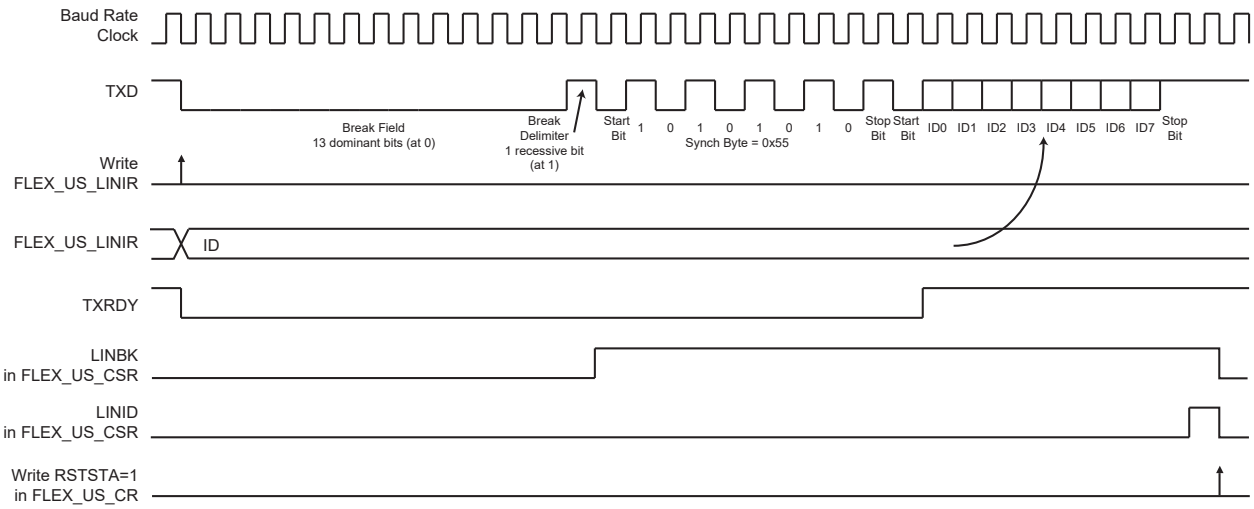
The Break Field, the Synch Field and the Identifier Field are sent automatically one after the other.

The Break Field consists of 13 dominant bits and 1 recessive bit, the Synch Field is the character 0x55 and the Identifier corresponds to the character written in the LIN Identifier Register (FLEX_US_LINIR). The Identifier parity bits can be automatically computed and sent (see section Identifier Parity).

The flag TXRDY rises when the identifier character is transferred into the shift register of the transmitter.

As soon as the Synch Break Field is transmitted, the FLEX_US_CSR.LINBK flag bit is set. Likewise, as soon as the Identifier Field is sent, the FLEX_US_CSR.LINID flag bit is set. These flags are reset by writing a one to the FLEX_US_CR.RSTSTA bit.

Figure 45-39. Header Transmission



45.7.8.7 Header Reception (Slave Node Configuration)

All the LIN frames start with a header which is sent by the master node and consists of a Synch Break Field, Synch Field and Identifier Field.

In slave node configuration, the frame handling starts with the reception of the header.

The USART uses a break detection threshold of 11 nominal bit times at the actual baud rate. At any time, if 11 consecutive recessive bits are detected on the bus, the USART detects a Break Field. As long as a Break Field has not been detected, the USART stays idle and the received data are not taken in account.

When a Break Field has been detected, the FLEX_US_CSR.LINBK flag is set and the USART expects the Synch Field character to be 0x55. This field is used to update the actual baud rate in order to remain synchronized (see section Slave Node Synchronization). If the received Synch character is not 0x55, an Inconsistent Synch Field error is generated (see section LIN Errors).

Downloaded from Arrow.com.

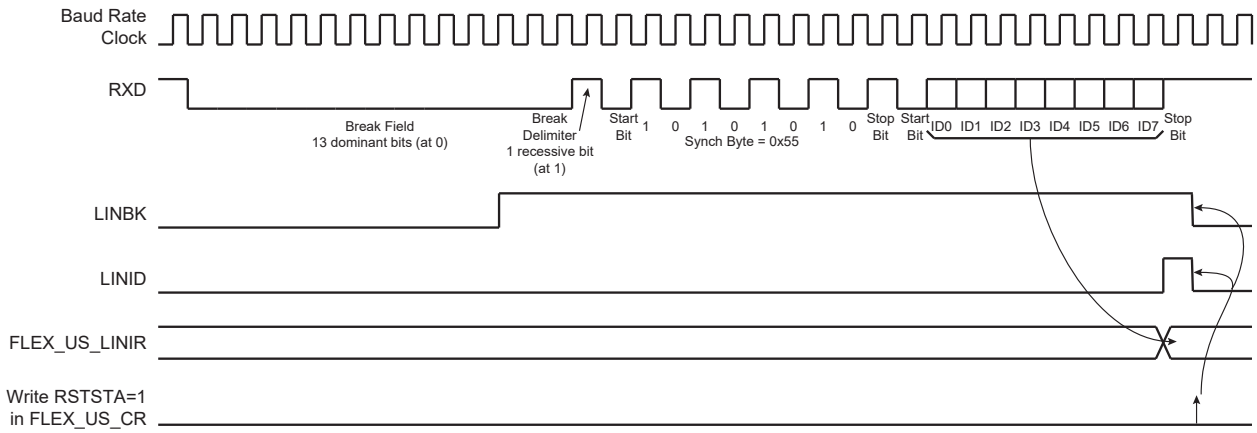
After receiving the Synch Field, the USART expects to receive the Identifier Field.

When the Identifier Field has been received, the FLEX_US_CSR.LINID flag bit is set. At this moment, the IDCHR field in the LIN Identifier Register (FLEX_US_LINIR) is updated with the received character. The Identifier parity bits can be automatically computed and checked (see section Identifier Parity).

If the header is not entirely received within the time given by the maximum length of the header $t_{Header_Maximum}$, the FLEX_US_CSR.LINHTE error flag bit is set.

The flag bits LINID, LINBK and LINHTE are reset by writing a one to the FLEX_US_CR.RSTSTA bit.

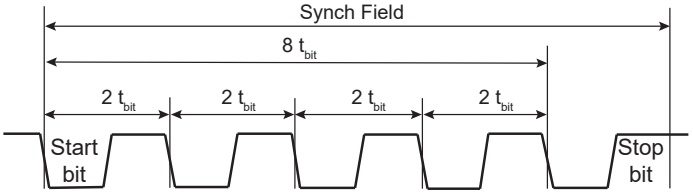
Figure 45-40. Header Reception



45.7.8.8 Slave Node Synchronization

The synchronization is done only in slave node configuration. The procedure is based on time measurement between the falling edges of the Synch Field. The falling edges are available in distances of 2, 4, 6 and 8 bit times.

Figure 45-41. Synch Field



The time measurement is made by a 19-bit counter driven by the sampling clock (see section Baud Rate Generator).

When the start bit of the Synch Field is detected, the counter is reset. Then during the next eight t_{bit} of the Synch Field, the counter is incremented. At the end of these eight t_{bit}, the counter is stopped. At this moment, the 16 most significant bits of the counter (value divided by 8) give the new clock divider (LINCD) and the 3 least significant bits of this value (the remainder) give the new fractional part (LINFP).

Once the Synch Field has been entirely received, the clock divider (LINCD) and the fractional part (LINFP) are updated in the LIN Baud Rate Register (FLEX_US_LINBRR) with the computed values, if the Synchronization is not disabled by the SYNCDIS bit in the LIN Mode Register (FLEX_US_LINMR).

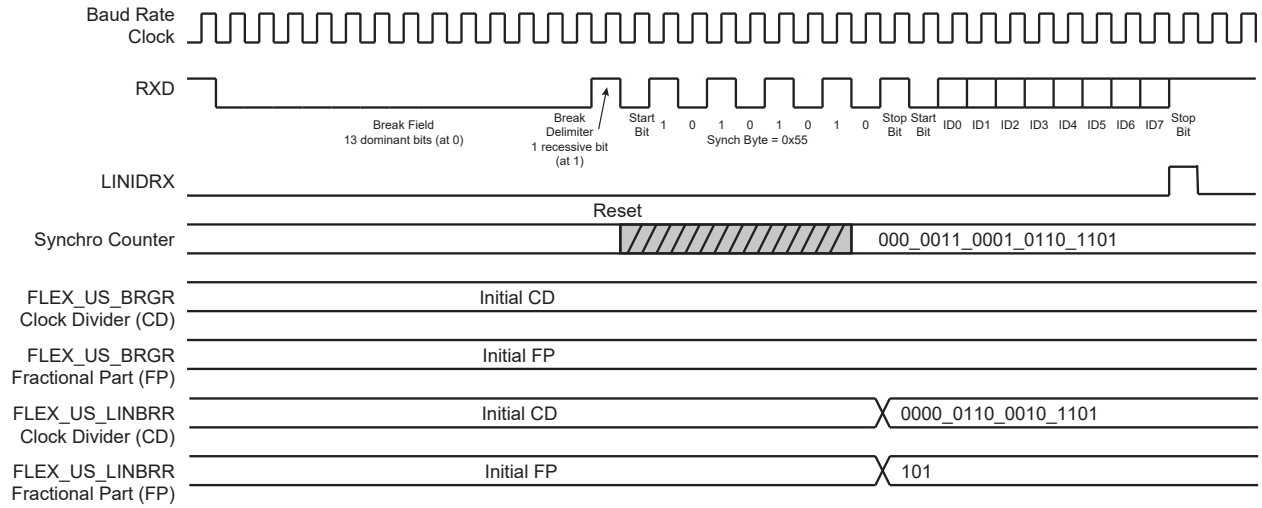
After reception of the Synch Field:

- If it appears that the computed baud rate deviation compared to the initial baud rate is superior to the maximum tolerance FTol_Unsynch (±15%), then the clock divider (LINCD) and the fractional part (LINFP) are not updated, and the FLEX_US_CSR.LINSTE error flag bit is set.
- If it appears that the sampled Synch character is not equal to 0x55, then the clock divider (LINCD) and the fractional part (LINFP) are not updated, and the FLEX_US_CSR.LINISFE error flag bit is set.

Flags LINSTE and LINISFE are reset by writing a one to the FLEX_US_CR.RSTSTA bit.

Downloaded from Arrow.com.

Figure 45-42. Slave Node Synchronization



The synchronization accuracy depends on several parameters:

- The nominal clock frequency (f_{Nom}) (the theoretical slave node clock frequency)
- The baud rate
- The oversampling (OVER = 0 => 16X or OVER = 1 => 8X)

The following formula is used to compute the deviation of the slave bit rate relative to the master bit rate after synchronization (f_{SLAVE} is the real slave node clock frequency).

$$\text{Baud rate deviation} = \left(100 \times \frac{[\alpha \times 8 \times (2 - \text{Over}) + \beta] \times \text{Baud rate}}{8 \times f_{SLAVE}}\right)\%$$

$$\text{Baud rate deviation} = \left(100 \times \frac{[\alpha \times 8 \times (2 - \text{Over}) + \beta] \times \text{Baud rate}}{8 \times \left(\frac{f_{TOL_UNSYNCH}}{100}\right) \times f_{Nom}}\right)\%$$

$-0.5 \leq \alpha \leq +0.5 \quad -1 < \beta < +1$

$f_{TOL_UNSYNCH}$ is the deviation of the real slave node clock from the nominal clock frequency. The LIN Standard imposes that it must not exceed ±15%. The LIN Standard imposes also that for communication between two nodes, their bit rate must not differ by more than ±2%. This means that the baud rate deviation must not exceed ±1%.

Therefore, a minimum value for the nominal clock frequency can be computed as follows:

$$f_{Nom}(\text{min}) = \left(100 \times \frac{[0.5 \times 8 \times (2 - \text{Over}) + 1] \times \text{Baud rate}}{8 \times \left(\frac{-15}{100} + 1\right) \times 1\%}\right)\text{Hz}$$

Examples:

- Baud rate = 20 kbit/s, OVER = 0 (Oversampling 16X) => f_{Nom}(min) = 2.64 MHz
- Baud rate = 20 kbit/s, OVER = 1 (Oversampling 8X) => f_{Nom}(min) = 1.47 MHz
- Baud rate = 1 kbit/s, OVER = 0 (Oversampling 16X) => f_{Nom}(min) = 132 kHz
- Baud rate = 1 kbit/s, OVER = 1 (Oversampling 8X) => f_{Nom}(min) = 74 kHz

45.7.8.9 Identifier Parity

A protected identifier consists of two subfields: the identifier and the identifier parity. Bits 0 to 5 are assigned to the identifier, and bits 6 and 7 are assigned to the parity.

The USART interface can generate/check these parity bits, but this feature can also be disabled. The user can choose between two modes via the FLEX_US_LINMR.PARDIS bit:

- PARDIS = 0:

Downloaded from Arrow.com.

  - During header transmission, the parity bits are computed and sent with the six least significant bits of the IDCHR field of the LIN Identifier Register (FLEX_US_LINIR). Bits 6 and 7 of this register are discarded.
  - During header reception, the parity bits of the identifier are checked. If the parity bits are wrong, an Identifier Parity error occurs (see section Parity). Only the six least significant bits of the IDCHR field are updated with the received Identifier. Bits 6 and 7 are stuck to 0.
- PARDIS = 1:
  - During header transmission, all the bits of the IDCHR field of the LIN Identifier Register (FLEX_US_LINIR) are sent on the bus.
  - During header reception, all the bits of the IDCHR field are updated with the received Identifier.

45.7.8.10 Node Action

Depending on the identifier, the node is affected—or not—by the LIN response. Consequently, after sending or receiving the identifier, the USART must be configured. There are three possible configurations:

- PUBLISH: the node sends the response.
- SUBSCRIBE: the node receives the response.
- IGNORE: the node is not concerned by the response, it does not send and does not receive the response.

This configuration is made by the LIN Node Action (NACT) field in USART LIN Mode Register (FLEX_US_LINMR).

Example: a LIN cluster that contains a master and two slaves:

- Data transfer from the master to slave 1 and to slave 2:

NACT(master) = PUBLISH

NACT(slave 1) = SUBSCRIBE

NACT(slave 2) = SUBSCRIBE

- Data transfer from the master to slave 1 only:

NACT(master) = PUBLISH

NACT(slave 1) = SUBSCRIBE

NACT(slave 2) = IGNORE

- Data transfer from slave 1 to the master:

NACT(master) = SUBSCRIBE

NACT(slave 1) = PUBLISH

NACT(slave 2) = IGNORE

- Data transfer from slave 1 to slave 2:

NACT(master) = IGNORE

NACT(slave 1) = PUBLISH

NACT(slave 2) = SUBSCRIBE

- Data transfer from slave 2 to the master and to slave 1:

NACT(master) = SUBSCRIBE

NACT(slave 1) = SUBSCRIBE

NACT(slave 2) = PUBLISH

45.7.8.11 Response Data Length

The LIN response data length is the number of data fields (bytes) of the response excluding the checksum.

The response data length can either be configured by the user or be defined automatically by bits 4 and 5 of the Identifier (compatibility to LIN Specification 1.1). The user can choose between these two modes by the FLEX_US_LINMR.DLM bit:

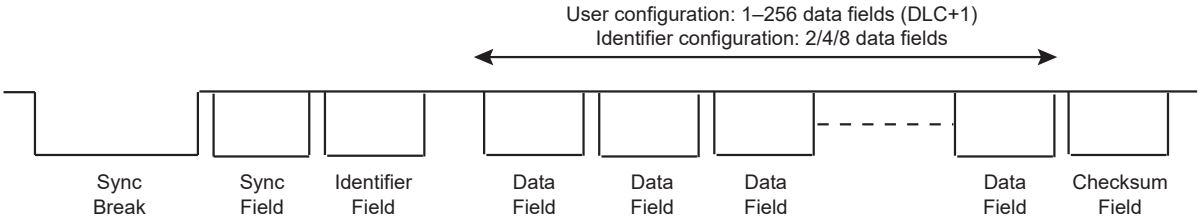
Downloaded from Arrow.com.

- DLM = 0: The response data length is configured by the user via the FLEX_US_LINMR.DLC field. The response data length is equal to (DLC + 1) bytes. DLC can be programmed from 0 to 255, so the response can contain from 1 data byte up to 256 data bytes.
- DLM = 1: The response data length is defined by the Identifier (IDCHR in FLEX_US_LINIR) according to the table below. The FLEX_US_LINMR.DLC field is discarded. The response can contain 2 or 4 or 8 data bytes.

Table 45-12. Response Data Length if DLM = 1

IDCHR[5]	IDCHR[4]	Response Data Length (bytes)
0	0	2
0	1	2
1	0	4
1	1	8

Figure 45-43. Response Data Length



45.7.8.12 Checksum

The last field of a frame is the checksum. The checksum contains the inverted 8- bit sum with carry, over all data bytes or all data bytes and the protected identifier. Checksum calculation over the data bytes only is called classic checksum and it is used for communication with LIN 1.3 slaves. Checksum calculation over the data bytes and the protected identifier byte is called enhanced checksum and it is used for communication with LIN 2.0 slaves.

The USART can be configured to:

- Send/Check an Enhanced checksum automatically (CHKDIS = 0 & CHKTYP = 0)
- Send/Check a Classic checksum automatically (CHKDIS = 0 & CHKTYP = 1)
- Not send/check a checksum (CHKDIS = 1)

This configuration is made by the Checksum Type (CHKTYP) and Checksum Disable (CHKDIS) bits of FLEX_US_LINMR.

If the checksum feature is disabled, the user can send it manually all the same, by considering the checksum as a normal data byte and by adding 1 to the response data length (see section Response Data Length).

45.7.8.13 Frame Slot Mode

This mode is useful only for master nodes. It respects the following rule: each frame slot shall be longer than or equal to $t_{Frame_Maximum}$.

If the Frame Slot mode is enabled (FSDIS = 0) and a frame transfer has been completed, the TXRDY flag is set again only after $t_{Frame_Maximum}$ delay, from the start of frame. So the master node cannot send a new header if the frame slot duration of the previous frame is inferior to $t_{Frame_Maximum}$.

If the Frame Slot mode is disabled (FSDIS = 1) and a frame transfer has been completed, the TXRDY flag is set again immediately.

The $t_{Frame_Maximum}$ is calculated as follows:

If the Checksum is sent (CHKDIS = 0):

- $t_{Header_Nominal} = 34 \times t_{bit}$
- $t_{Response_Nominal} = 10 \times (NData + 1) \times t_{bit}$
- $t_{Frame_Maximum} = 1.4 \times (t_{Header_Nominal} + t_{Response_Nominal} + 1)$(1)
- $t_{Frame_Maximum} = 1.4 \times (34 + 10 \times (DLC + 1 + 1) + 1) \times t_{bit}$

Downloaded from Arrow.com.

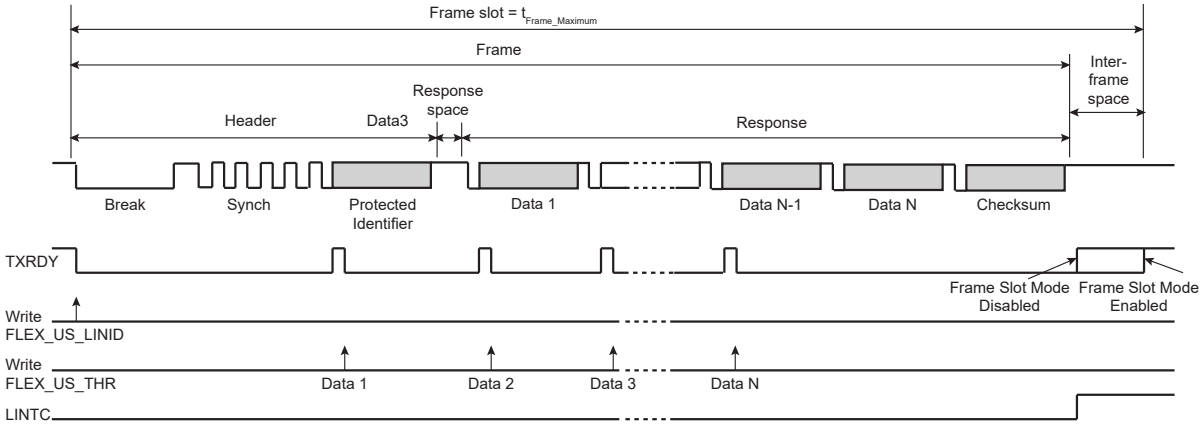
- $t_{Frame_Maximum}$ = (77 + 14 × DLC) × t_{bit}

If the Checksum is not sent (CHKDIS = 1):

- $t_{Header_Nominal}$ = 34 × t_{bit}
- $t_{Response_Nominal}$ = 10 × NData × t_{bit}
- $t_{Frame_Maximum}$ = 1.4 × ($t_{Header_Nominal}$ + $t_{Response_Nominal}$ + 1)(1)
- $t_{Frame_Maximum}$ = 1.4 × (34 + 10 × (DLC + 1) + 1) × t_{bit}
- $t_{Frame_Maximum}$ = (63 + 14 × DLC) × t_{bit}

Note: 1. The term “+1” leads to an integer result for $t_{Frame_Maximum}$ (LIN Specification 1.3).

Figure 45-44. Frame Slot Mode



45.7.8.14 LIN Errors

45.7.8.14.1 Bit Error

This error is generated in master of slave node configuration, when the USART is transmitting and if the transmitted value on the Tx line is different from the value sampled on the Rx line. If a bit error is detected, the transmission is aborted at the next byte border.

This error is reported by the FLEX_US_CSR.LINBE flag.

45.7.8.14.2 Inconsistent Synch Field Error

This error is generated in slave node configuration, if the Synch Field character received is other than 0x55.

This error is reported by the FLEX_US_CSR.LINISFE flag.

45.7.8.14.3 Identifier Parity Error

This error is generated in slave node configuration, if the parity of the identifier is wrong. This error can be generated only if the parity feature is enabled (PARDIS = 0).

This error is reported by the FLEX_US_CSR.LINIPE flag.

45.7.8.14.4 Checksum Error

This error is generated in master of slave node configuration, if the received checksum is wrong. This flag can be set to 1 only if the checksum feature is enabled (CHKDIS = 0).

This error is reported by the FLEX_US_CSR.LINCE flag.

45.7.8.14.5 Slave Not Responding Error

This error is generated in master of slave node configuration, when the USART expects a response from another node (NACT = SUBSCRIBE) but no valid message appears on the bus within the time given by the maximum length of the message frame, $t_{Frame_Maximum}$ (see section Frame Slot Mode). This error is disabled if the USART does not expect any message (NACT = PUBLISH or NACT = IGNORE).

This error is reported by the FLEX_US_CSR.LINSNRE.

Downloaded from Arrow.com.

45.7.8.14.6 Synch Tolerance Error

This error is generated in slave node configuration if, after the clock synchronization procedure, it appears that the computed baud rate deviation compared to the initial baud rate is superior to the maximum tolerance FTol_Unsynch (±15%).

This error is reported by the FLEX_US_CSR.LINSTE flag.

45.7.8.14.7 Header Timeout Error

This error is generated in slave node configuration, if the Header is not entirely received within the time given by the maximum length of the Header, $t_{Header_Maximum}$.

This error is reported by the FLEX_US_CSR.LINHTE flag.

45.7.8.15 LIN Frame Handling

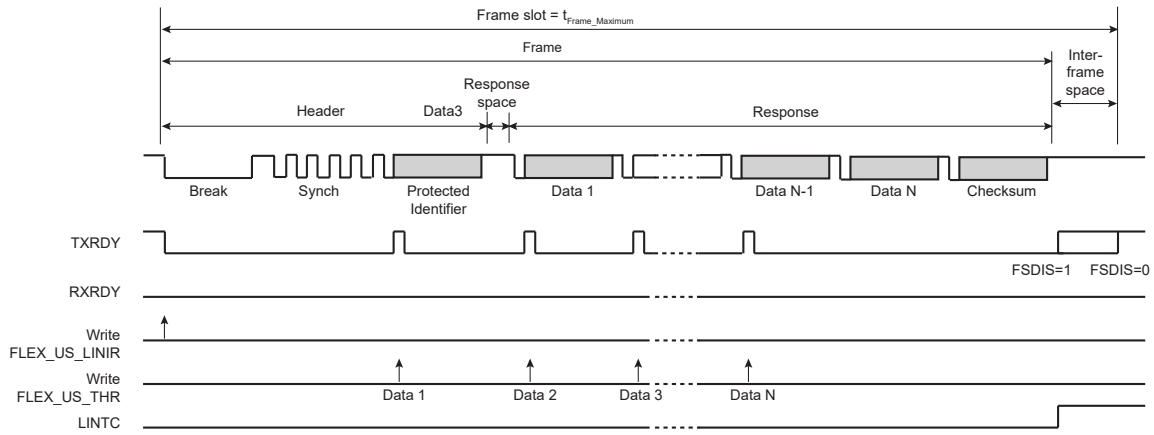
45.7.8.15.1 Master Node Configuration

- Write FLEX_US_CR.TXEN and FLEX_US_CR.RXEN to enable both the transmitter and the receiver.
- Write FLEX_US_MR.USART_MODE to select the LIN mode and the master node configuration.
- Write FLEX_US_BRGR.CD and FLEX_US_BRGR.FP to configure the baud rate.
- Write NACT, PARDIS, CHKDIS, CHKTYPE, DLCM, FSDIS and DLC in FLEX_US_LINMR to configure the frame transfer.
- Check that FLEX_US_CSR. TXRDY is set to 1.
- Write FLEX_US_LINIR.IDCHR to send the header.

What comes next depends on the NACT configuration:

- Case 1: NACT = PUBLISH, the USART sends the response.
  - Wait until FLEX_US_CSR. TXRDY rises.
  - Write FLEX_US_THR.TCHR to send a byte.
  - If all the data have not been written, repeat the two previous steps.
  - Wait until FLEX_US_CSR.LINTC rises.
  - Check the LIN errors.
- Case 2: NACT = SUBSCRIBE, the USART receives the response.
  - Wait until FLEX_US_CSR.RXRDY rises.
  - Read FLEX_US_RHR.RCHR.
  - If all the data have not been read, repeat the two previous steps.
  - Wait until FLEX_US_CSR.LINTC rises.
  - Check the LIN errors.
- Case 3: NACT = IGNORE, the USART is not concerned by the response.
  - Wait until FLEX_US_CSR.LINTC rises.
  - Check the LIN errors.

Figure 45-45. Master Node Configuration, NACT = PUBLISH



Downloaded from Arrow.com.

Figure 45-46. Master Node Configuration, NACT = SUBSCRIBE

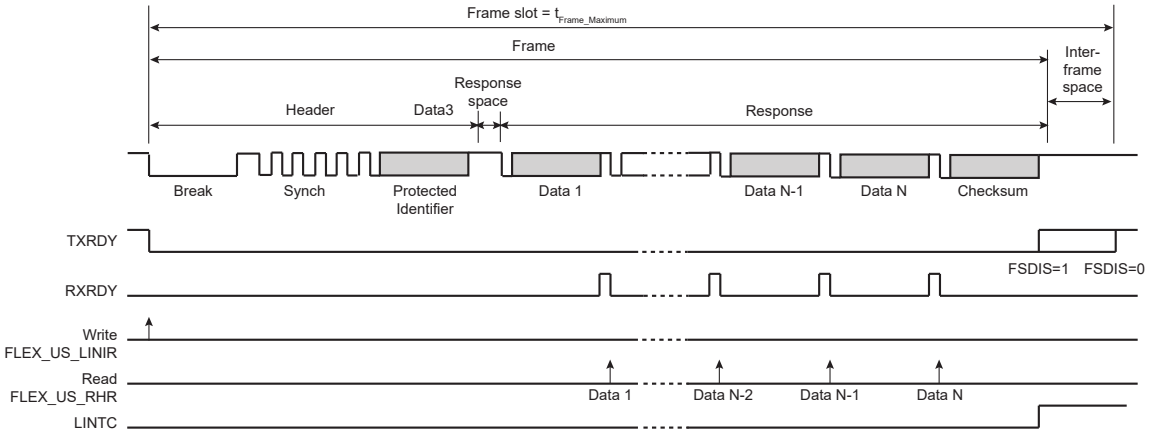
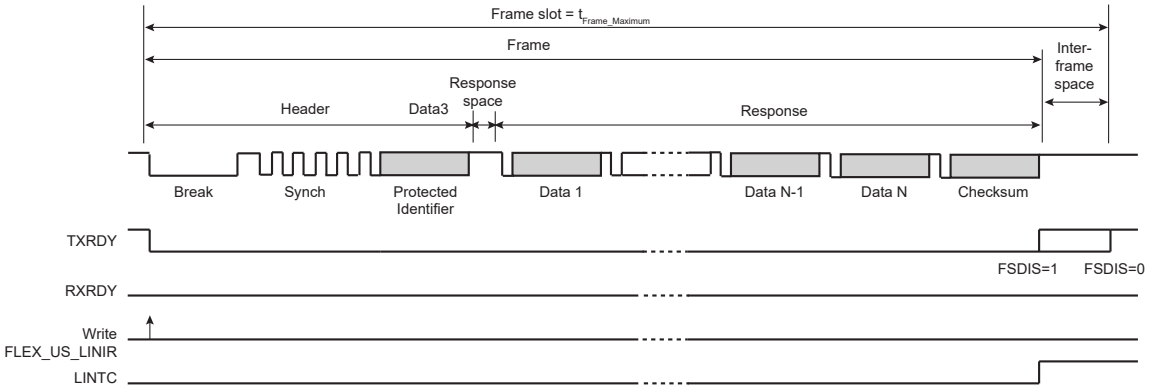


Figure 45-47. Master Node Configuration, NACT = IGNORE



45.7.8.15.2 Slave Node Configuration

- Write FLEX_US_CR.TXEN and FLEX_US_CR.RXEN to enable both the transmitter and the receiver.
- Write FLEX_US_MR.USART_MODE to select the LIN mode and the slave node configuration.
- Write FLEX_US_BRGR.CD and FLEX_US_BRGR.FP to configure the baud rate.
- Wait until FLEX_US_CSR.LINID rises.
- Check LINISFE and LINPE errors.
- Read FLEX_US_RHR.IDCHR.
- Write NACT, PARDIS, CHKDIS, CHKTYPE, DLCM and DLC in FLEX_US_LINMR to configure the frame transfer.

IMPORTANT: If the NACT configuration for this frame is PUBLISH, FLEX_US_LINMR must be written with NACT = PUBLISH even if this field is already correctly configured, in order to set the TXREADY flag and the corresponding write transfer request.

What comes next depends on the NACT configuration:

- Case 1: NACT = PUBLISH, the LIN controller sends the response.
  - Wait until FLEX_US_CSR.TXRDY rises.
  - Write FLEX_US_THR.TCHR to send a byte.
  - If all the data have not been written, repeat the two previous steps.
  - Wait until FLEX_US_CSR. LINTC rises.
  - Check the LIN errors.
- Case 2: NACT = SUBSCRIBE, the USART receives the response.
  - Wait until FLEX_US_CSR.RXRDY rises.
  - Read FLEX_US_RHR.RCHR.

Downloaded from Arrow.com.

– If all the data have not been read, repeat the two previous steps.
– Wait until FLEX_US_CSR.LINTC rises.
– Check the LIN errors.

• Case 3: NACT = IGNORE, the USART is not concerned by the response.
– Wait until FLEX_US_CSR.LINTC rises.
– Check the LIN errors.

Figure 45-48. Slave Node Configuration, NACT = PUBLISH

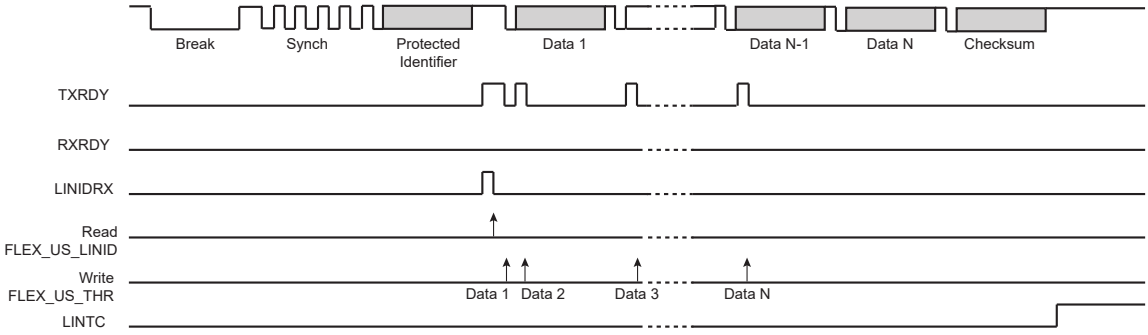


Figure 45-49. Slave Node Configuration, NACT = SUBSCRIBE

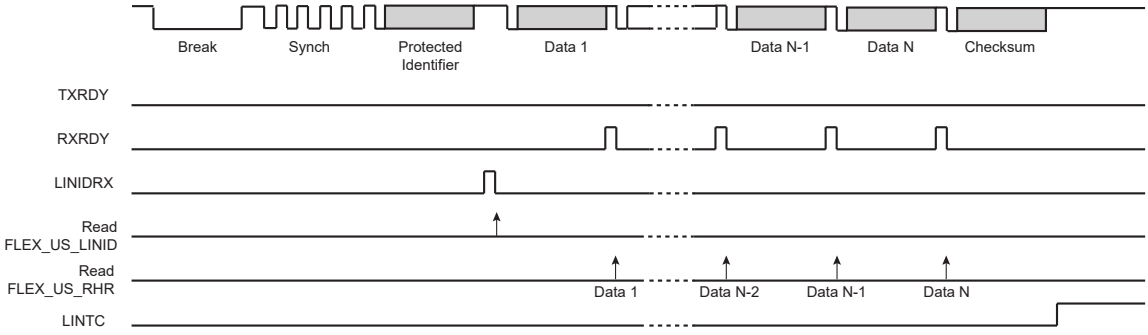
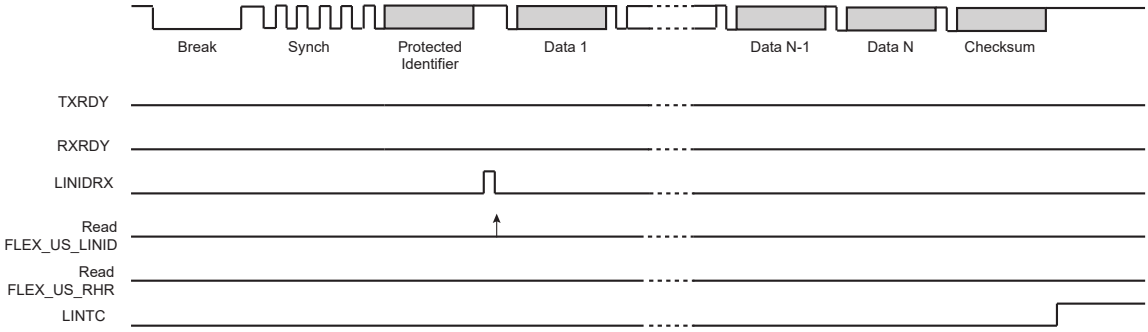


Figure 45-50. Slave Node Configuration, NACT = IGNORE



45.7.8.16 LIN Frame Handling with the DMA

The USART can be used in association with the DMA in order to transfer data directly into/from the on- and off-chip memories without any processor intervention.

The DMA uses the trigger flags, TXRDY and RXRDY, to write or read into the USART. The DMA always writes in the Transmit Holding Register (FLEX_US_THR) and it always reads in the Receive Holding Register (FLEX_US_RHR). The size of the data written or read by the DMA in the USART is always a byte.

45.7.8.16.1 Master Node Configuration

The user can choose between two DMA modes by configuring the FLEX_US_LINMR.PDCM bit:

• PDCM = 1: The LIN configuration is stored in the WRITE buffer and it is written by the DMA in the Transmit Holding register FLEX_US_THR (instead of the LIN Mode register FLEX_US_LINMR). Because the DMA transfer size is limited to a byte, the transfer is split into two accesses. During the first access, the NACT,

Downloaded from Arrow.com.

PARDIS, CHKDIS, CHKTYP, DLM and FSDIS bits are written. During the second access, the 8-bit DLC field is written.

- PDCM = 0: The LIN configuration is not stored in the WRITE buffer and it must be written by the user in FLEX_US_LINMR.

The WRITE buffer also contains the Identifier and the data, if the USART sends the response (NACT = PUBLISH).

The READ buffer contains the data if the USART receives the response (NACT = SUBSCRIBE).

Figure 45-51. Master Node with DMA (PDCM = 1)

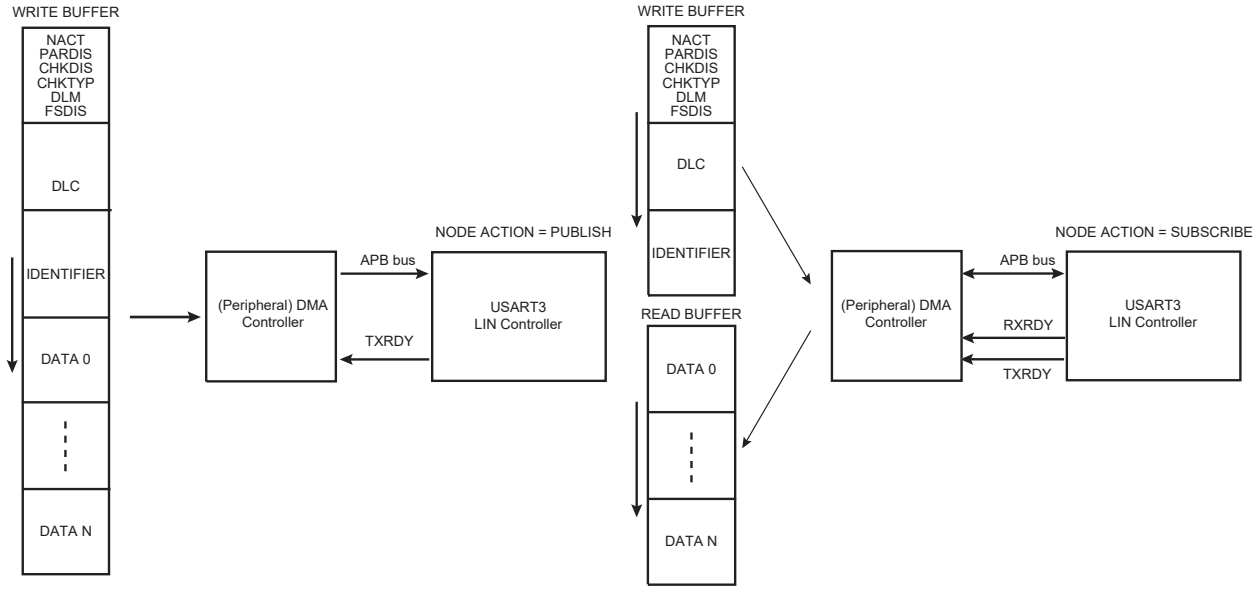
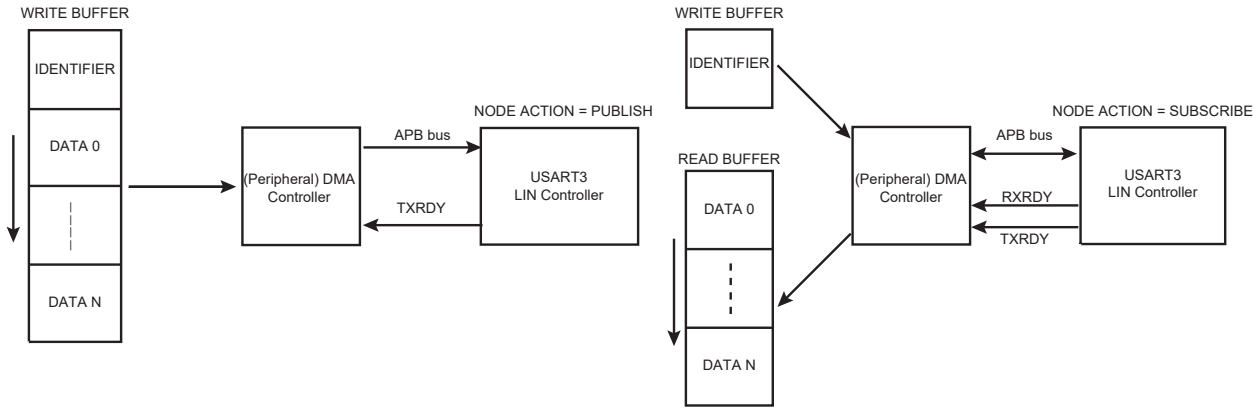


Figure 45-52. Master Node with DMA (PDCM = 0)



45.7.8.16.2 Slave Node Configuration

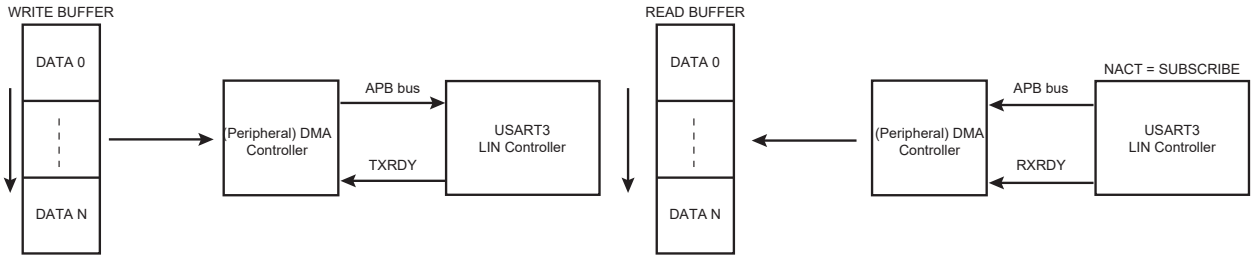
In this configuration, the DMA transfers only the data. The identifier must be read by the user in the LIN Identifier Register (FLEX_US_LINIR). The LIN mode must be written by the user in FLEX_US_LINMR.

The WRITE buffer contains the data if the USART sends the response (NACT = PUBLISH).

The READ buffer contains the data if the USART receives the response (NACT = SUBSCRIBE).

Downloaded from Arrow.com.

Figure 45-53. Slave Node with DMA



45.7.8.17 Wakeup Request

Any node in a sleeping LIN cluster may request a wakeup.

In the LIN 2.0 specification, the wakeup request is issued by forcing the bus to the dominant state from 250 µs to 5 ms. For this, it is necessary to send the character 0xF0 in order to impose five successive dominant bits. Whatever the baud rate is, this character respects the specified timings.

- Baud rate min = 1 kbit/s -> t_{bit} = 1 ms -> 5 t_{bit} = 5 ms
- Baud rate max = 20 kbit/s -> t_{bit} = 50 µs -> 5 t_{bit} = 250 µs

In the LIN 1.3 specification, the wakeup request should be generated with the character 0x80 in order to impose eight successive dominant bits.

Using the FLEX_US_LINMR.WKUPTYP bit, the user can choose to send either a LIN 2.0 wakeup request (WKUPTYP = 0) or a LIN 1.3 wakeup request (WKUPTYP = 1).

A wakeup request is transmitted by writing the FLEX_US_CR.LINWKUP bit to 1. Once the transfer is completed, the LINTC flag is asserted in the Status Register (FLEX_US_CSR). It is cleared by writing a one to the FLEX_US_CR.RSTSTA bit.

45.7.8.18 Bus Idle Timeout

If the LIN bus is inactive for a certain duration, the slave nodes shall automatically enter in Sleep mode. In the LIN 2.0 specification, this timeout is defined as 4 seconds. In the LIN 1.3 specification, it is defined as 25,000 t_{bit}.

In slave Node configuration, the receiver timeout detects an idle condition on the RXD line. When a timeout is detected, the FLEX_US_CSR.TIMEOUT bit rises and can generate an interrupt, thus indicating to the driver to go into Sleep mode.

The timeout delay period (during which the receiver waits for a new character) is programmed in the FLEX_US_RTOR.TO field. If a zero is written to the TO field, the Receiver Timeout is disabled and no timeout is detected. The FLEX_US_CSR.TIMEOUT bit remains at 0. Otherwise, the receiver loads a 17-bit counter with the value programmed in TO. This counter is decremented at each bit period and reloaded each time a new character is received. If the counter reaches 0, the FLEX_US_CSR.TIMEOUT bit rises.

If STTTO is performed, the counter clock is stopped until a first character is received.

If RETTO is performed, the counter starts counting down immediately from the value TO.

Table 45-13. Receiver Timeout Programming

LIN Specification	Baud Rate	Timeout period	TO
2.0	1,000 bit/s	4s	4,000
	2,400 bit/s		9,600
	9,600 bit/s		38,400
	19,200 bit/s		76,800
	20,000 bit/s		80,000
1.3	–	25,000 t_{bit}	25,000

Downloaded from Arrow.com.

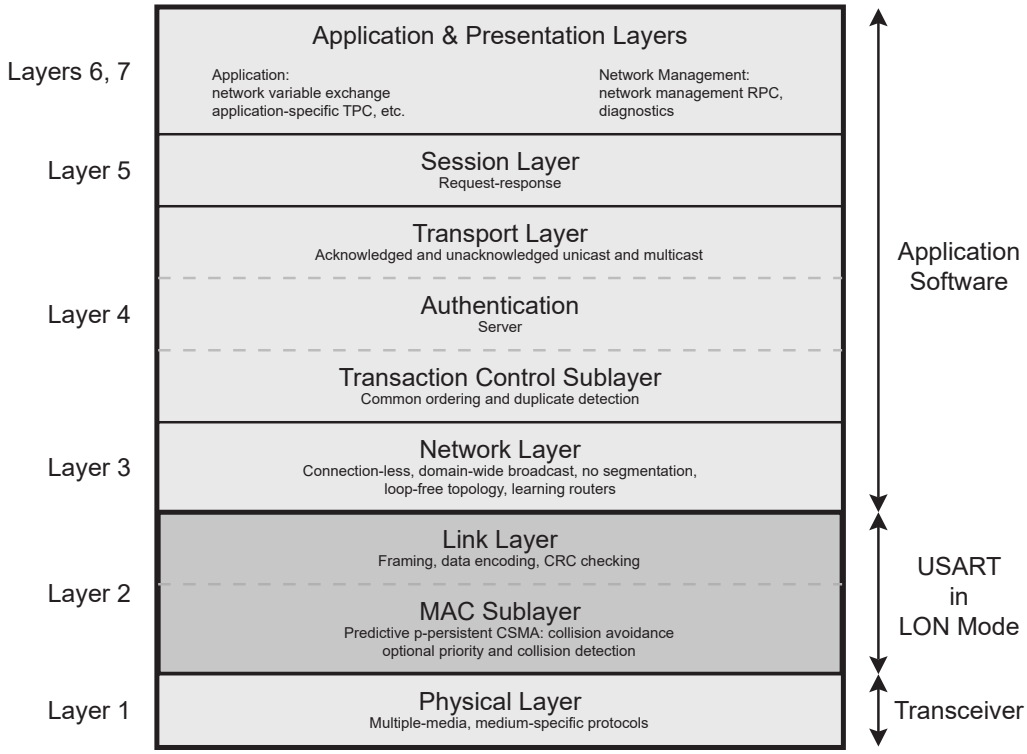
45.7.9 LON Mode

The LON mode provides connectivity to the local operating network (LON).

The LON standard covers all seven layers of the OSI (Open Systems Interconnect) reference model from the physical interfaces such as wired, power line, RF, and IP to the application layer and all layers in between.

The LON mode enables the transmission and reception of Physical Protocol Data Unit (PPDU) frames with minimum intervention from the microprocessor.

Figure 45-54. LON Protocol Layering



The USART configured in LON mode is a full-layer 2 implementation including standard timings handling, framing (transmit and receive PPDU frames), backlog estimation and other features. At the frame encoding/decoding level, differential Manchester encoding is used (also known as CDP). When configured in LON mode, there is no embedded digital line filter, thus the optimal usage is node-to-node communication.

45.7.9.1 Mode of Operation

To configure the USART to act as a LON node, the USART_MODE field of the USART Mode Register (FLEX_US_MR) must be set to 0x9.

To avoid unpredictable behavior, any change of the LON node configuration must be preceded by a software reset of the transmitter and the receiver (except the initial node configuration after a hardware reset) and followed by a transmitter/receiver enable. See section Receiver and Transmitter Control.

45.7.9.2 Receiver and Transmitter Control

See section Receiver and Transmitter Control.

45.7.9.3 Character Transmission

A LON frame is made up of a preamble, a data field (up to 256 bytes) and a 16-bit CRC field. The preamble and CRC fields are automatically generated and the LON node starts the transmission algorithm on a write to LON_L2HDR. See section Sending a Frame.

45.7.9.4 Character Reception

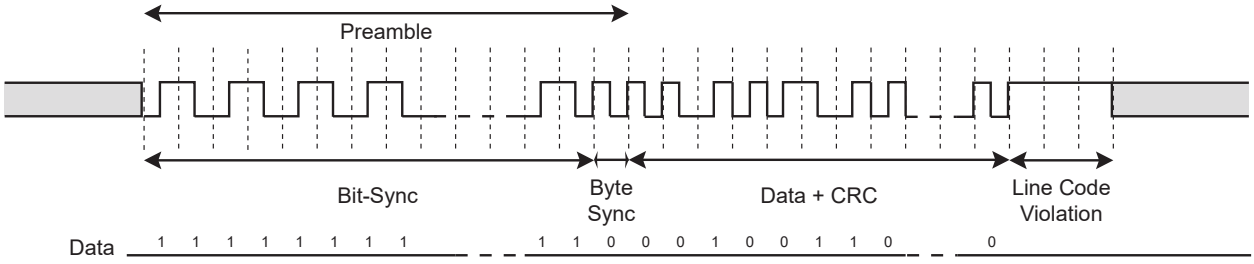
When receiving a LON frame, the Receive Holding Register (FLEX_US_RHR) is updated upon completed character reception and the RXRDY bit in the Status register rises. If a character is completed while the RXRDY bit is set, the

Downloaded from Arrow.com.

OVRE (Overrun Error) bit is set. The LON preamble field is only used for synchronization, therefore only the Data and CRC fields are transmitted to the Receive Holding Register (FLEX_US_RHR). See section Sending a Frame.

45.7.9.5 LON Frame

Figure 45-55. LON Framing



45.7.9.5.1 Encoding / Decoding

The USART configured in LON mode encodes transmitted data and decodes received data using differential Manchester encoding. In differential Manchester encoding, a '1' bit is indicated by making the first half of the signal equal to the last half of the previous bit's signal (no transition at the start of the bit-time). A '0' bit is indicated by making the first half of the signal opposite to the last half of the previous bit's signal (a zero bit is indicated by a transition at the beginning of the bit-time). As is the case with normal Manchester encoding, missing transition at the middle of bit-time represents a Manchester code violation.

The FLEX_US_MAN.RXIDLEV bit informs the USART of the receiver line idle state value (receiver line inactive) thus ensuring higher reliability of preamble synchronization. By default, RXIDLEV is set (receiver line is at level 1 when there is no activity).

Differential Manchester encoding is polarity-insensitive.

Figure 45-56. LON PPDU

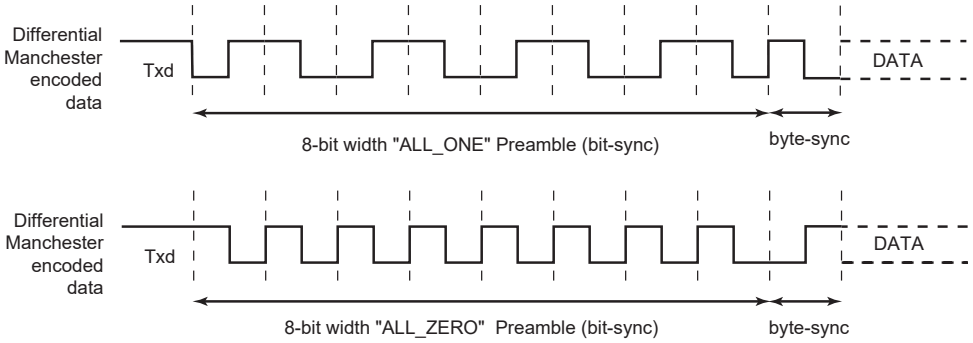
45.7.9.5.2 Preamble Transmission

Each LON frame begins with a preamble of variable length which consists of a bit-sync field and a byte-sync field. The LONPL field of the USART LON Preamble Register (FLEX_US_LONPR) defines the preamble length. Note that a preamble length of '0' is not allowed.

The LON implementation allows two different preamble patterns ALL_ONE and ALL_ZERO which can be configured through the TX_PL field of the USART Manchester Configuration Register (FLEX_US_MAN). The following figure illustrates and defines the valid patterns.

Other preamble patterns are not supported.

Figure 45-57. Preamble Patterns



45.7.9.5.3 Preamble Reception

LON received frames begin with a preamble of variable length. The receiving algorithm does not check the preamble length, although a minimum of length of 4 bits is required for the receiving algorithm to consider the received preamble as valid.

Downloaded from Arrow.com.

As is the case with LON preamble transmission, two preamble patterns (ALL_ONE and ALL_ZERO) are allowed and can be configured through the RX_PL field of the USART Manchester Configuration Register (FLEX_US_MAN). The above figure illustrates and defines the valid patterns.

Other preamble patterns are not supported.

45.7.9.5.4 Header Transmission

Each LON frame, after sending the preamble, starts with the frame header also called L2HDR according to CEA-709 specification. This header consist of the priority bit, the alternative path bit and the backlog increment. It is the first data to be sent.

In LON mode, the transmitting algorithm starts when FLEX_US_LONL2HDR is written (it is the first data to send).

45.7.9.5.5 Header Reception

Each LON frame, after receiving the preamble, receives the frame header also called L2HDR according to CEA-709 specification. This header consists of the priority bit, the alternative path bit, and the backlog increment.

The frame header is the first received data and the RXRDY bit rises as soon as the frame header as been received and stored in the Receive Holding Register (FLEX_US_RHR).

45.7.9.5.6 Data

Data are sent/received serially after the preamble transmission/reception. Data can be either sent/received MSB first or LSB first depending on the MSBF bit value in the USART Mode Register (FLEX_US_MR).

45.7.9.5.7 CRC

The two last bytes of LON frames are dedicated to CRC.

When transmitting, the CRC of the frame is automatically generated and sent when expected.

When receiving frames, the CRC is automatically checked and the FLEX_US_CSR.LCRCE flag is set if the calculated CRC does not match the received one. Note that the two received CRC bytes are seen as two additional data from the user point of view.

45.7.9.5.8 End Of Frame

The USART configured in LON mode terminates the frame with a three t_{bit} long Manchester code violation. After sending the last CRC bit, it maintains the data transitionless during three bit periods.

45.7.9.6 LON Operating Modes

45.7.9.6.1 Transmitting/Receiving Modules

According to the LON node configuration and LON network state, the transmitting module is activated if a transmission request has been made and access to the LON bus granted. It returns to idle state once the transmission ends.

According to the LON node configuration and LON network state, the receiving module is activated if a valid preamble is detected and the transmitting module is not activated.

45.7.9.6.2 comm_type

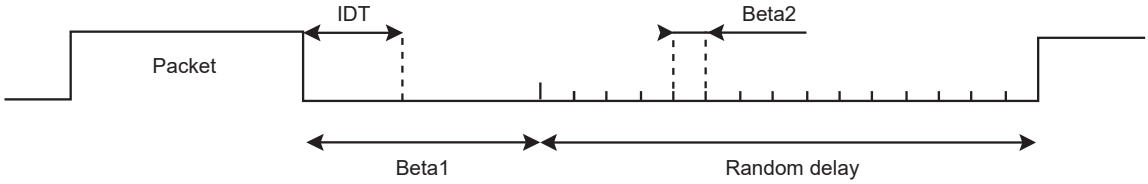
In CEA-709 standard 2, communication configurations are defined and configurable through the comm_type variable. The comm_type variable value can be set in the USART LON Mode Register (FLEX_US_LONMR) through the COMMT bit. The selection of the comm_type determines the MAC behavior in the following ways:

- comm_type = 1:
  - An indeterminate time is defined during the Beta 1 period in which all transitions on the channel are ignored (see figure below).
  - The MAC sublayer ignores collisions occurring during the first 25% of the transmitted preamble. Optionally (according to the FLEX_US_LONMR.CDTAIL bit), it ignores collisions reported following the transmission of the CRC but prior to the end of transmission.
  - If a collision is detected during preamble transmission, the MAC sublayer can terminate the packet if so configured according to the FLEX_US_LONMR.TCOL bit. Collisions detected after the preamble have been sent do not terminate transmission.
- comm_type = 2:
  - No indeterminate time is defined at the MAC sublayer.

Downloaded from Arrow.com.

– The MAC sublayer always terminates the packet upon notification of a collision.

Figure 45-58. LON Indeterminate Time



45.7.9.6.3 Collision Detection

As an option of the CEA-709 standard collision detection is supported through an active low Collision Detect (CD) input from the transceiver.

The Collision Detection source can be either external (see section I/O Lines Description) or internal. The collision detection source selection is defined through the LCDS bit in the USART LON Mode Register (FLEX_US_LONMR).

The Collision Detection feature can be activated through the COLDET bit of the USART LON Mode Register. If the collision detection feature is enabled and CD signal goes low for at least half t_{bit} period then a collision is detected and reported as defined in section comm_type.

45.7.9.6.4 Collision Detection Mode

As defined in section comm_type, if comm_type = 1 the LON node can be configured to either not terminate transmission upon collision notification during preamble transmission or terminate transmission.

The FLEX_US_LONMR.TCOL bit allows to decide whether to terminate transmission or not upon collision notification during preamble transmission.

45.7.9.6.5 Collision Detection after CRC

As defined in section comm_type, if comm_type = 1 the LON node can be configured to ignore collisions reported after the CRC has been sent but prior to the end of the frame.

The FLEX_US_LONMR.CDTAIL bit can be used to decide whether such collision notifications must be considered or not.

45.7.9.6.6 Random Number Generation

The Predictive p-persistent CSMA algorithm defined in the CEA-709.1 Standard is based on a random number generation.

This random number is automatically generated by an internal algorithm.

In addition, a USART IC DIFF Register (FLEX_US_ICDIFF) is available to avoid that two same chips with the same software generate the same random number after reset. The value of this register is used by the internal algorithm to generate the random number. Therefore, putting a different value here for each chip ensures that the random number generated after a reset at the same time will not be the same. It is recommended to put the chip ID code here.

45.7.9.7 LON Node Backlog Estimation

As defined in CEA-709, the LON node maintains its own backlog estimation. The node backlog estimation is initially set to one, will always be greater than 1 and will never exceed 63. If the node backlog estimation exceeds the maximum backlog value, the backlog value is set to 63 and a backlog overflow error flag is set (LBLOVFE flag).

The node backlog estimation is incremented each time a frame is sent or received successfully. The increment to the backlog is encoded into the link layer header, and represents the number of messages that the packet shall cause to be generated upon reception.

The backlog decrements under one of the following conditions:

- On waiting to transmit: If Wbase randomizing slots go by without channel activity.
- On receive: If a packet is received with a backlog increment of ‘0’.
- On transmit: If a packet is transmitted with a backlog increment of ‘0’.
- On idle: If a packet cycle time expires without channel activity.

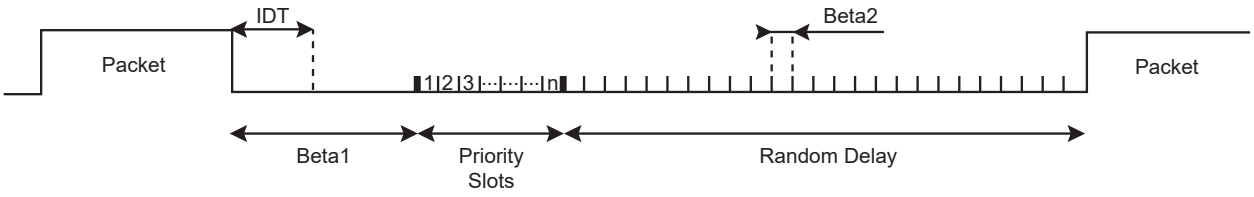
45.7.9.7.1 Optional Collision Detection Feature And Backlog Estimation

Each time a frame is transmitted and a collision occurred, the backlog is incremented by 1. In this case, the backlog increment encoded in the link layer is ignored.

Downloaded from Arrow.com.

45.7.9.8 LON Timings

Figure 45-59. LON Timings



45.7.9.8.1 Beta2

A node wishing to transmit generates a random delay T. This delay is an integer number of randomizing slots of duration Beta2.

The beta2 length (in t_{bit}) is configurable through FLEX_US_FIDI. Note that a length of '0' is not allowed.

45.7.9.8.2 Beta1 Tx/Rx

Beta1 is the period immediately following the end of a packet cycle (see the above figure). A node attempting to transmit monitors the state of the channel, and if it detects no transmission during the Beta1 period, it determines the channel to be idle.

The Beta1 value is different depending on the previous packet type (received packet or transmitted packet).

Beta1Rx and Beta1Tx length can be configured respectively through the USART LON Beta1 Rx Register (FLEX_US_LONB1RX) and the USART LON Beta1 Tx Register (FLEX_US_LONB1TX). Note that a length of '0' is not allowed.

45.7.9.8.3 Pcycle Timer

The packet cycle timer is reset to its initial value whenever the backlog is changed. It is started (begins counting down at its current value) whenever the MAC layer becomes idle. An idle MAC layer is defined as:

- Not receiving
- Not transmitting,
- Not waiting to transmit,
- Not timing Beta1,
- Not waiting for priority slots, and not waiting for the first Wbase randomizing window to complete.

On transition from idle to either transmit or receive, the packet cycle timer is halted.

The pcycle timer value can be configured in FLEX_US_TTGR. Note that '0' value is not allowed.

45.7.9.8.4 Wbase

The wbase timer represents the base windows size. Its duration, derived from Beta2, equals 16 Beta2 slots.

45.7.9.8.5 Priority Slots

On a channel by channel basis, the protocol supports optional priority. Priority slots, if any, follow immediately after the Beta1 period that follows the transmission of a packet (see the above figure). The number of priority slots per channel ranges from 0 to 127.

The number of priority slots in the LON network configuration is defined through the PSNB field of the USART LON Priority Register (FLEX_US_LONPRIO). And the priority slot affected to the LON node, if any, is defined through the FLEX_US_LONPRIO.NPS field.

45.7.9.8.6 Indeterminate Time

See section comm_type.

Like Beta1, the IDT value is different depending on whether the previous frame was transmitted or received.

IDTRx and IDTTx can be configured respectively through the USART LON IDT Rx Register (FLEX_US_LONIDTRX) and the USART LON IDT Tx Register (FLEX_US_LONIDTTX).

45.7.9.8.7 End of Frame Condition

The USART configured in LON mode terminates the frame with a three t_{bit} long Manchester code violation. After sending the last CRC bit, it maintains the data transitionless during three bit periods.

Downloaded from Arrow.com.

While receiving data, the USART configured in LON mode detects an end of frame condition after a t_{eof} transitionless Manchester code violation. The EOFS field in the USART LON Mode Register can configure t_{eof}.

45.7.9.9 LON Errors

All these flags can be read in the LON Channel Status Register (FLEX_US_CSR) and generate interrupts if configured in the LON Interrupt Enable Register (FLEX_US_IER).

These flags can be reset by writing a one to the FLEX_US_CR.RSTSTA bit.

45.7.9.9.1 Underrun Error

If the USART is in LON mode and if a character is sent while the Transmit Holding Register (FLEX_US_THR) is empty, the UNRE bit flag is set.

45.7.9.9.2 Collision Detection

The LCOL flag is set whenever a valid collision has been detected and the LON node is configured to report it (see section Collision Detection).

45.7.9.9.3 LON Frame Early Termination

The LFET flag is set whenever a LON frame has been terminated early due to collision detection.

45.7.9.9.4 Reception Error

The LCRCE flag is set if the received frame has an erroneous CRC and the flag LSFE is set if the received frame is too short (LON frames must be at least 8 bytes long).

These flags can be read in FLEX_US_CSR.

45.7.9.9.5 Backlog Overflow

The LBLOVFE flag is set if the LON node backlog estimation goes over 63, which is the maximum backlog value.

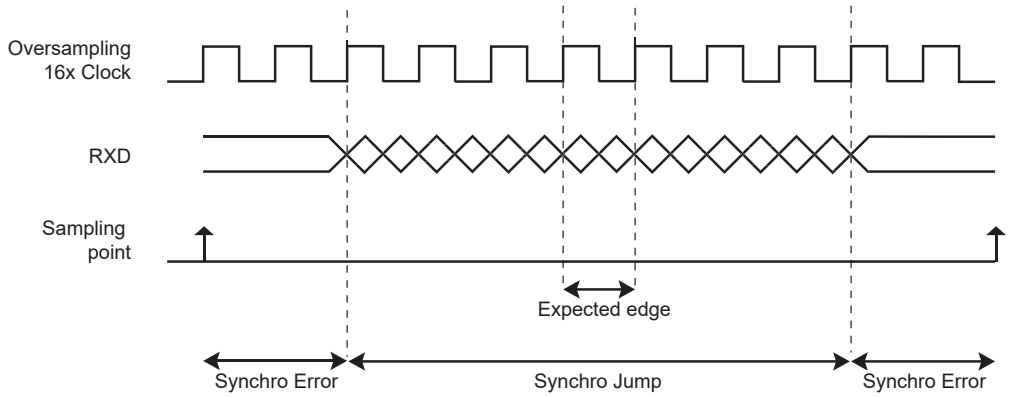
45.7.9.10 Drift Compensation

It may happen that while receiving a frame, the baud rate used by the sender is not exactly the one expected, due to sender clock drifting for instance. In such case, the hardware drift compensation algorithm is used to recover up to 16% clock drift (expected baud rate ±16% is supported).

Drift compensation is available only in 16X Oversampling mode. To enable the hardware system, the DRIFT bit of the FLEX_US_MAN register must be set. If the RXD edge is between one and three 16X clock cycles away from the expected edge, then the period is shortened or lengthened accordingly to center the RXD edge.

Drift compensation hardware feature allows up to 16% clock drift to be handled, provided system clock is fast enough compared to the selected baud rate.

Figure 45-60. Bit Resynchronization



45.7.9.11 LON Frame Handling

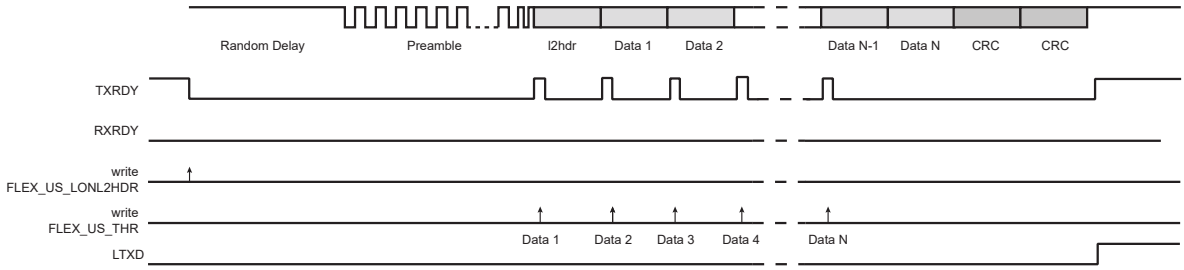
45.7.9.11.1 Sending a Frame

1. Write FLEX_US_CR.TXEN and FLEX_US_CR.RXEN to enable both the transmitter and the receiver.
2. Write FLEX_US_MR.USART_MODE to select the LON mode configuration.
3. Write FLEX_US_BRGR.CD and FLEX_US_BRGR.FP to configure the baud rate.

Downloaded from Arrow.com.

4. Write COMMT, COLDET, TCOL, CDTAIL, RDMNBM and DMAM in FLEX_US_LONMR to configure the LON operating mode.
5. Write BETA2, BETA1TX, BETA1RX, PCYCLE, PSNB, NPS, IDTTX and ITDRX respectively in FLEX_US_FIDI, FLEX_US_LONB1TX, FLEX_US_LONB1RX, FLEX_US_TTGR, FLEX_US_LONPRIO, FLEX_US_LONIDTTX and FLEX_US_LONIDTRX to set the LON network configuration.
6. Write FLEX_US_MAN.TX_PL to select the preamble pattern to use.
7. Write LONPL and LONDL in FLEX_US_LONPR and FLEX_US_LONDL to set the frame transfer.
8. Check that FLEX_US_CSR.TXRDY is set to 1.
9. Write FLEX_US_LONL2HDR to send the header.
10. Wait until FLEX_US_CSR.TXRDY rises.
11. Write FLEX_US_THR.TCHR to send a byte.
12. If all the data have not been written, repeat the two previous steps.
13. Wait until FLEX_US_CSR.LTXD rises.
14. Check the LON errors.

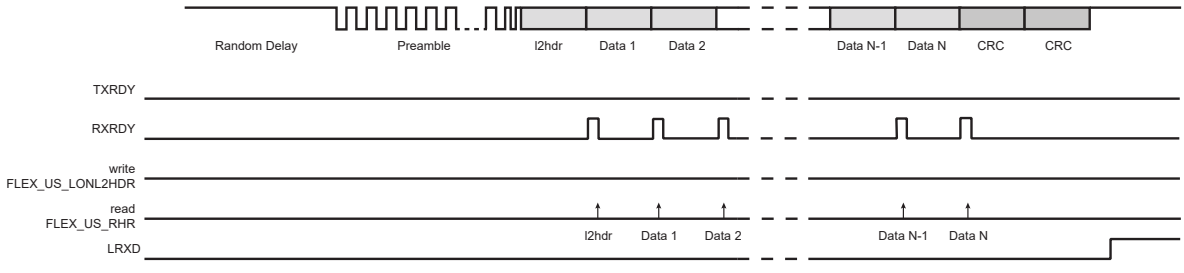
Figure 45-61. Tx Frame



45.7.9.11.2 Receiving a Frame

1. Write FLEX_US_CR.TXEN and FLEX_US_CR.RXEN to enable both the transmitter and the receiver.
2. Write FLEX_US_MR.USART_MODE to select the LON mode configuration.
3. Write FLEX_US_BRGR.CD and FLEX_US_BRGR.FP to configure the baud rate.
4. Write COMMT, COLDET, TCOL, CDTAIL, RDMNBM and DMAM in FLEX_US_LONMR to configure the LON operating mode.
5. Write BETA2, BETA1TX, BETA1RX, PCYCLE, PSNB, NPS, IDTTX and ITDRX respectively in FLEX_US_FIDI, FLEX_US_LONB1TX, FLEX_US_LONB1RX, FLEX_US_TTGR, FLEX_US_LONPRIO, FLEX_US_LONIDTTX and FLEX_US_LONIDTRX to set the LON network configuration.
6. Write FLEX_US_MAN.RXIDLEV and FLEX_US_MAN.RX_PL to indicate the receiver line value and select the preamble pattern to use.
7. Wait until FLEX_US_CSR.RXRDY rises.
8. Read FLEX_US_RHR.RCHR.
9. If all the data and the two CRC bytes have not been read, repeat the two previous steps.
10. Wait until FLEX_US_CSR.LRXD rises.
11. Check the LON errors.

Figure 45-62. Rx Frame



Downloaded from Arrow.com.

45.7.9.12 LON Frame Handling with the Peripheral DMA Controller

The USART can be used in association with the DMA Controller in order to transfer data directly into/from the on- and off-chip memories without any processor intervention.

The DMA uses the trigger flags, TXRDY and RXRDY, to write or read into the USART. The DMA always writes in the Transmit Holding Register (FLEX_US_THR) and it always reads in the Receive Holding Register (FLEX_US_RHR). The size of the data written or read by the DMA in the USART is always a byte.

45.7.9.12.1 Configuration

The user can choose between two DMA modes by the DMAM bit in the LON Mode register

(FLEX_US_LONMR):

- DMAM = 1: The LON frame data length (DATAL) is stored in the WRITE buffer and it is written by the DMA in the Transmit Holding register FLEX_US_THR (instead of the LON Data Length register FLEX_US_LONDL).
- DMAM = 0: The LON frame data length (DATAL) is not stored in the WRITE buffer and it must be written by the user in the LON Data Length Register (FLEX_US_LONDL).

In both DMA modes, L2HDR is considered as a data and its value must be stored in the WRITE buffer as the first data to write.

Figure 45-63. DMAM = 1

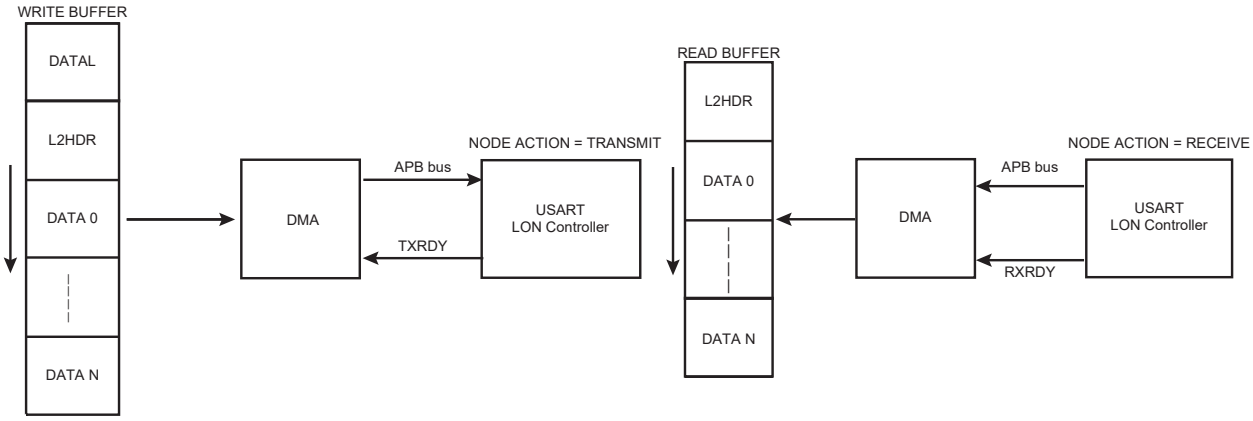
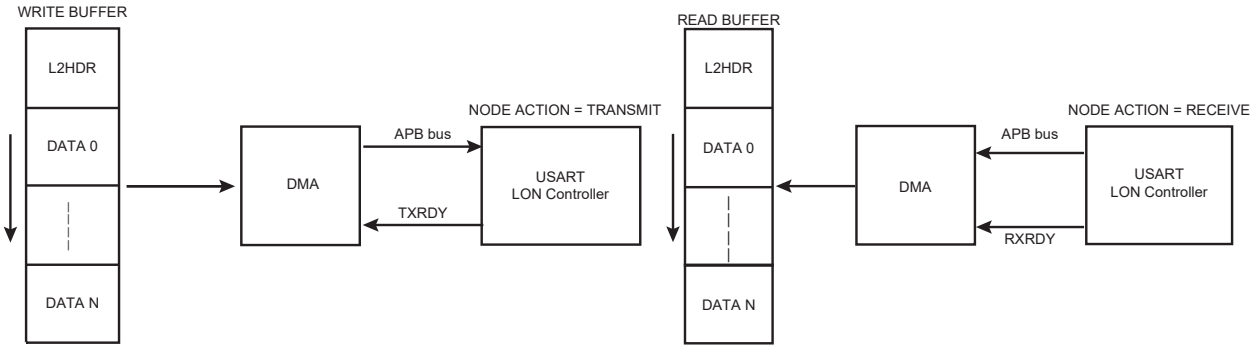


Figure 45-64. DMAM = 0



45.7.9.12.2 DMA and Collision Detection

As explained in section comm_type, depending on LON configuration the transmission may be terminated early upon collision notification which means that the DMA transfer may be stopped before its end.

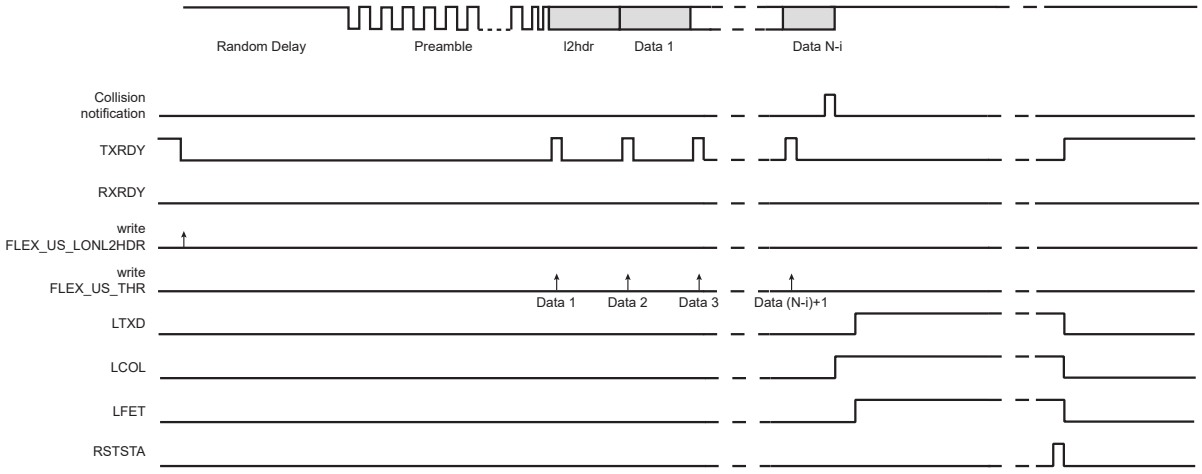
In case of early end of transmission due to collision detection, the USART in LON mode acts as follows:

- Send the end of frame trigger.
- Hold down TXRDY, thus avoiding any additional DMA transfer.
- Set the LTXD, LCOL and LFET flags in FLEX_US_CSR.
- Wait for the application to reconfigure the DMA.

Downloaded from Arrow.com.

- Wait until the LCOL and LFET flags are cleared through the FLEX_US_CR.RSTSTA bit (it will release the TXRDY signal).

Figure 45-65. DMA, Collision and Early Frame Termination



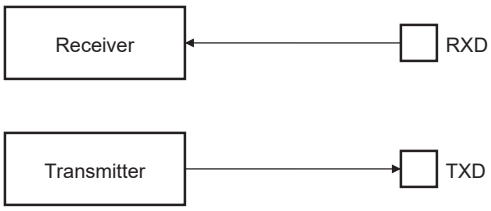
45.7.10 Test Modes

The USART can be programmed to operate in three different test modes. The internal loopback capability allows on-board diagnostics. In Loopback mode, the USART interface pins are disconnected or not and reconfigured for loopback internally or externally.

45.7.10.1 Normal Mode

Normal mode connects the RXD pin on the receiver input and the transmitter output on the TXD pin.

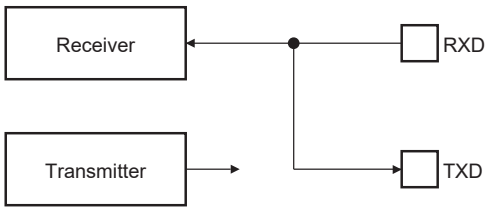
Figure 45-66. Normal Mode Configuration



45.7.10.2 Automatic Echo Mode

Automatic Echo mode allows bit-by-bit retransmission. When a bit is received on the RXD pin, it is sent to the TXD pin, as shown in the following figure. Programming the transmitter has no effect on the TXD pin. The RXD pin is still connected to the receiver input, thus the receiver remains active.

Figure 45-67. Automatic Echo Mode Configuration

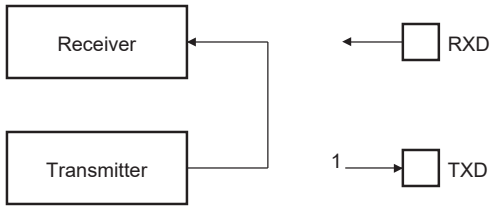


45.7.10.3 Local Loopback Mode

Local Loopback mode connects the output of the transmitter directly to the input of the receiver, as shown in the following figure. The TXD and RXD pins are not used. The RXD pin has no effect on the receiver and the TXD pin is continuously driven high, as in idle state.

Downloaded from Arrow.com.

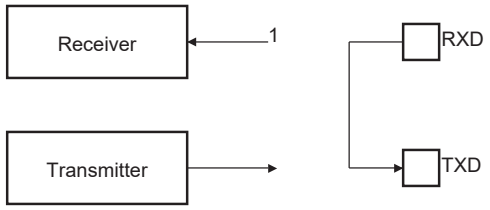
Figure 45-68. Local Loopback Mode Configuration



45.7.10.4 Remote Loopback Mode

Remote Loopback mode directly connects the RXD pin to the TXD pin, as shown in the following figure. The transmitter and the receiver are disabled and have no effect. This mode allows bit-by-bit retransmission.

Figure 45-69. Remote Loopback Mode Configuration



45.7.11 USART FIFOs

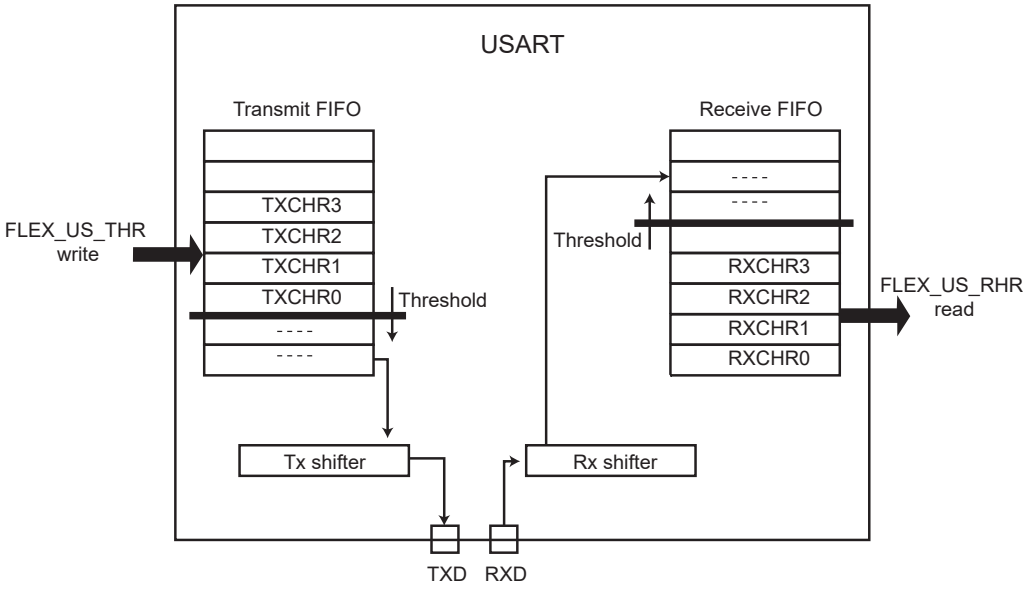
45.7.11.1 Overview

The USART includes two FIFOs which can be enabled/disabled using FLEX_US_CR.FIFOEN/FIFODIS. Both the transmitter and the receiver must be disabled before enabling or disabling the FIFOs, using the FLEX_US_CR.TXDIS/RXDIS bits.

Writing FLEX_US_CR.FIFOEN to ‘1’ enables a 16-data Transmit FIFO and a 16-data Receive FIFO.

It is possible to write or to read single or multiple data in the same access to FLEX_US_THR/RHR. See sections USART Single Data Mode and USART Multiple Data Mode.

Figure 45-70. FIFOs Block Diagram



45.7.11.2 Sending Data with FIFO Enabled

When the Transmit FIFO is enabled, write access to FLEX_US_THR loads the Transmit FIFO.

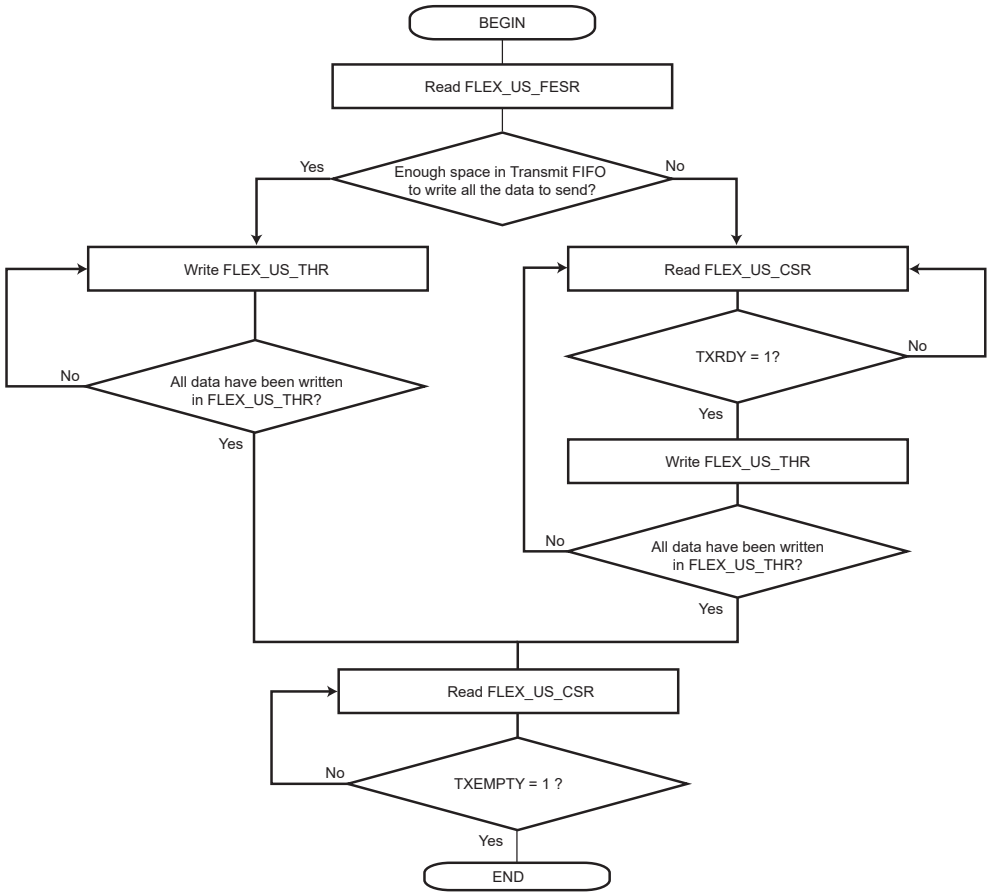
Downloaded from Arrow.com.

The FIFO level is provided in FLEX_US_FLR.TXFL. If the FIFO can accept the number of data to be transmitted, there is no need to monitor FLEX_US_CSR.TXRDY and the data can be successively written in FLEX_US_THR.

If the FIFO cannot accept the data due to insufficient space, wait for the TXRDY flag to be set before writing the data in FLEX_US_THR.

When the space in the FIFO allows only a portion of the data to be written, the TXRDY flag must be monitored before writing the remaining data.

Figure 45-71. Sending Data with FIFO Enabled



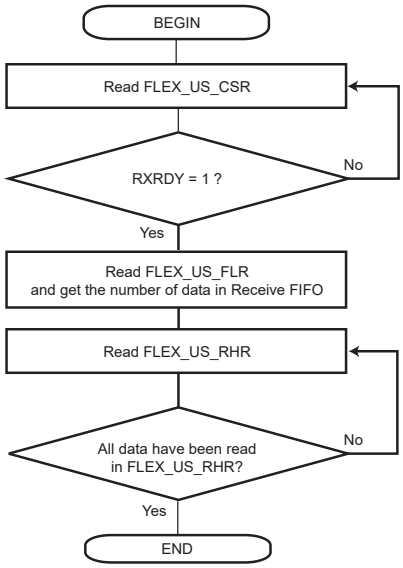
45.7.11.3 Receiving Data with FIFO Enabled

When the Receive FIFO is enabled, FLEX_US_RHR access reads the FIFO.

When data are present in the Receive FIFO (RXRDY flag set to ‘1’), the exact number of data can be checked with FLEX_US_FLR.RXFL. All the data can be read successively in FLEX_US_RHR without checking the RXRDY flag between each access.

Downloaded from Arrow.com.

Figure 45-72. Receiving Data with FIFO Enabled



45.7.11.4 Clearing/Flushing FIFOs

Each FIFO can be cleared/flushed using FLEX_US_CR.TXFCLR/RXFCLR.

45.7.11.5 TXEMPTY, TXRDY and RXRDY Behavior

FLEX_US_CSR.TXEMPTY, FLEX_US_CSR.TXRDY and FLEX_US_CSR.RXRDY flags display a specific behavior when FIFOs are enabled.

The TXEMPTY flag is cleared as long as there are characters in the Transmit FIFO or in the internal shift register. TXEMPTY is set when there are no characters in the Transmit FIFO and in the internal shift register.

TXRDY indicates if a data can be written in the Transmit FIFO. Thus the TXRDY flag is set as long as the Transmit FIFO can accept new data. See figure TXRDY in Single Data Mode and TXRDYM = 0.

RXRDY indicates if an unread data is present in the Receive FIFO. Thus the RXRDY flag is set as soon as one unread data is in the Receive FIFO. See figure RXRDY in Single Data Mode and RXRDYM = 0 below.

TXRDY and RXRDY behavior can be modified using the TXRDYM and RXRDYM fields in the USART FIFO Mode Register (FLEX_US_FMR) to reduce the number of accesses to FLEX_US_RHR/THR. However, for some configurations, the following constraints apply:

- If FLEX_US_MR.MODE9 is set, FLEX_US_FMR.TXRDYM/RXRDYM must be cleared.
- If FLEX_US_MR.USART_MODE is set to either LON_MODE, LIN_MASTER or LIN_SLAVE, FLEX_US_FMR.TXRDYM/RXRDYM must be cleared.

See USART FIFO Mode Register for the FIFO configuration.

Downloaded from Arrow.com.

Figure 45-73. TXRDY in Single Data Mode and TXRDYM = 0

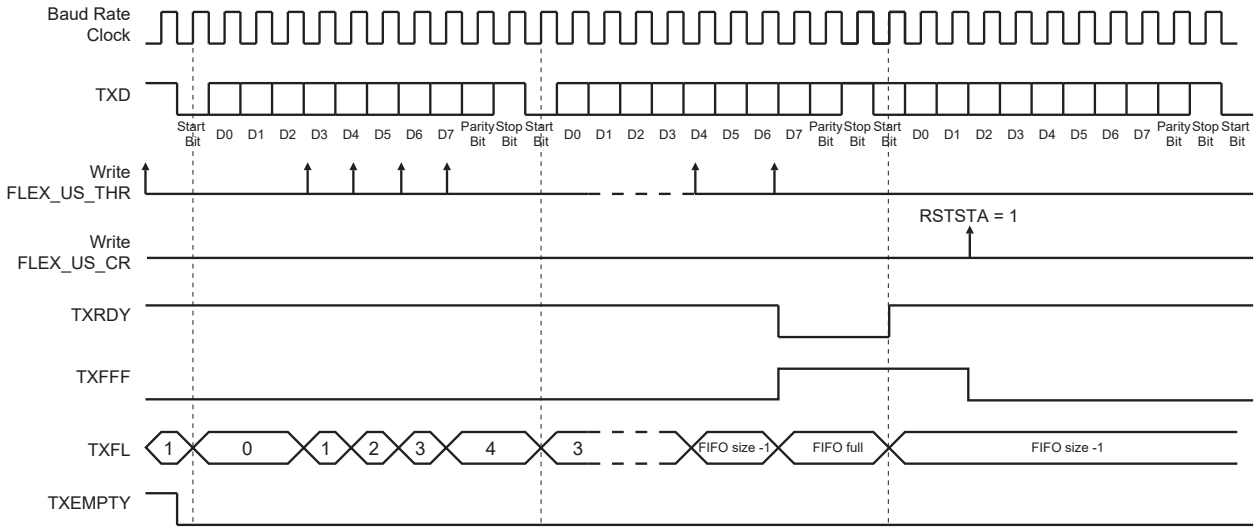
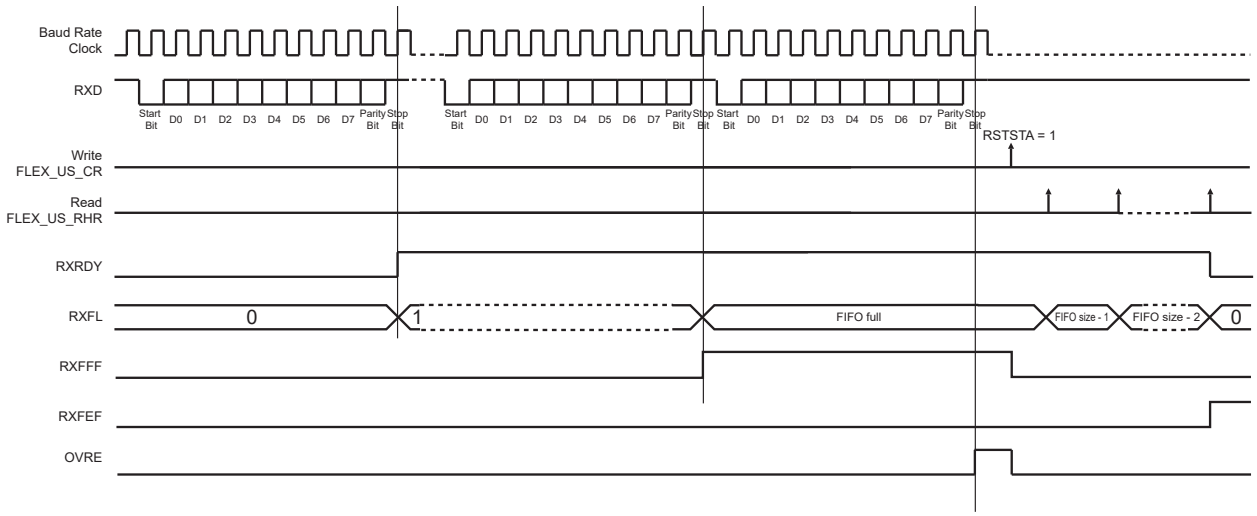


Figure 45-74. RXRDY in Single Data Mode and RXRDYM = 0



45.7.11.6 USART Single Data Mode

In Single Data mode, only one data is written every time FLEX_US_THR is accessed, and only one data is read every time FLEX_US_RHR is accessed.

When FLEX_US_FMR.TXRDYM = 0, the Transmit FIFO operates in Single Data mode.

When FLEX_US_FMR.RXRDYM = 0, the Receive FIFO operates in Single Data mode.

If FLEX_US_MR.MODE9 is set, or if FLEX_US_MR.USART_MODE is set to either LON_MODE, LIN_MASTER or LIN_SLAVE, the FIFOs must operate in Single Data mode.

See USART Receive Holding Register (FLEX_US_RHR) and USART Transmit Holding Register (FLEX_US_THR).

45.7.11.6.1 DMA

The DMA transfer type must be configured in bytes or halfwords when FIFOs operate in Single Data mode (the same applies when FIFOs are disabled).

45.7.11.7 USART Multiple Data Mode

Multiple Data mode minimizes the number of accesses by concatenating the data to send/read in one access.

When FLEX_US_FMR.TXRDYM > 0, the Transmit FIFO operates in Multiple Data mode.

When FLEX_US_FMR.RXRDYM > 0, the Receive FIFO operates in Multiple Data mode.

Downloaded from Arrow.com.

However, Multiple Data mode cannot be used for the following configurations:

- If FLEX_US_MR.MODE9 is set
- If FLEX_US_MR.USART_MODE is set to either LON_MODE, LIN_MASTER or LIN_SLAVE

In Multiple Data mode, it is possible to write/read up to four data in one FLEX_US_THR/FLEX_US_RHR access.

The number of data to write/read is defined by the size of the register access. If the access is a byte-size register access, only one data is written/read, if the access is a halfword size register access, then two data are written/read and, finally, if the access is a word-size register access, four data are written/read.

Written/read data are always right-aligned, as described in sections USART Receive Holding Register (FIFO Multi Data) and USART Transmit Holding Register (FIFO Multi Data).

As an example, if the Transmit FIFO is empty and there are six data to send, either of the following write accesses may be performed:

- six FLEX_US_THR-byte write accesses
- three FLEX_US_THR-halfword write accesses
- one FLEX_US_THR word write access and one FLEX_US_THR halfword write access

With a Receive FIFO containing six data, any of the following read accesses may be performed:

- six FLEX_US_RHR-byte read accesses
- three FLEX_US_RHR-halfword read accesses
- one FLEX_US_RHR-word read access and one FLEX_US_RHR-halfword read access

45.7.11.7.1 TXRDY and RXRDY Configuration

In Multiple Data mode, it is possible to write one or more data in the same FLEX_US_THR/FLEX_US_RHR access. The TXRDY flag indicates if one or more data can be written in the FIFO depending on the configuration of FLEX_US_FMR.TXRDYM/RXRDYM.

As an example, if four data are written each time in FLEX_US_THR, it is useful to configure the TXRDYM field to the value ‘2’ so that the TXRDY flag is at ‘1’ only when at least four data can be written in the Transmit FIFO.

In the same way, if four data are read each time in FLEX_US_RHR, it is useful to configure the RXRDYM field to the value ‘2’ so that the RXRDY flag is at ‘1’ only when at least four unread data are in the Receive FIFO.

45.7.11.7.2 DMA

When FIFOs operate in Multiple Data mode, the DMA transfer type must be configured in byte, halfword or word depending on the FLEX_US_FMR.TXRDYM/RXRDYM settings.

45.7.11.8 Transmit FIFO Lock

- LIN Mode:

If a frame is aborted using the Abort LIN Transmission bit (FLEX_US_CR.LINABT), a lock is set on the Transmit FIFO, preventing any new frame from being sent until it is cleared. This allows clearing the FIFO if needed, resetting DMA channels, etc., without any risk.

- LON Mode:

If a frame is terminated early due to collision, a lock is set on the Transmit FIFO preventing any new frame from being sent until it is cleared. This allows clearing the FIFO if needed, resetting DMA channels, etc., without any risk.

The TXFLOCK bit in the USART FIFO Event Status Register (FLEX_US_FESR) is used to check the state of the Transmit FIFO lock.

The Transmit FIFO lock can be cleared by setting FLEX_US_CR.TXFLCLR to ‘1’.

45.7.11.9 FIFO Pointer Error

A FIFO overflow is reported in FLEX_US_FESR.

If the Transmit FIFO is full and a write access is performed on FLEX_US_THR, it generates a Transmit FIFO pointer error and sets FLEX_US_FESR.TXFPTEF.

In Multiple Data mode, if the number of data written in FLEX_US_THR (according to the register access size) is greater than the free space in the Transmit FIFO, a Transmit FIFO pointer error is generated and FLEX_US_FESR.TXFPTEF is set.

Downloaded from Arrow.com.

A FIFO underflow is reported in FLEX_US_FESR.

In Multiple Data mode, if the number of data read in FLEX_US_RHR (according to the register access size) is greater than the number of unread data in the Receive FIFO, a Receive FIFO pointer error is generated and FLEX_US_FESR.RXFPTEF is set.

No pointer error occurs if the FIFO state/level is checked before writing/reading in FLEX_US_THR/FLEX_US_RHR. The FIFO state/level can be checked either with TXRDY, RXRDY, TXFL or RXFL. When a pointer error occurs, other FIFO flags may not behave as expected; their states should be ignored.

If a Transmit pointer error occurs, a transmitter reset must be performed using FLEX_US_CR.RSTTX. If a Receive pointer error occurs, a receiver reset must be performed using FLEX_US_CR.RSTRX.

45.7.11.10 FIFO Thresholds

Each Transmit and Receive FIFO includes a threshold feature used to set a flag and an interrupt when a FIFO threshold is crossed. Thresholds are defined as a number of data in the FIFO, and the FIFO state (TXFL or RXFL) represents the number of data currently in the FIFO.

The Transmit FIFO threshold can be set using the field FLEX_US_FMR.TXFTHRES. Each time the Transmit FIFO level goes from ‘above threshold’ to ‘equal to or below threshold’, the flag FLEX_US_FESR.TXFTHF is set. The application is warned that the Transmit FIFO has reached the defined threshold and that it can be reloaded.

The Receive FIFO threshold can be set using the field FLEX_US_FMR.RXFTHRES. Each time the Receive FIFO level goes from ‘below threshold’ to ‘equal to or above threshold’, the flag FLEX_US_FESR.RXFTHF is set. The application is warned that the Receive FIFO has reached the defined threshold and that it can be read to prevent an underflow.

The Receive FIFO threshold 2 can be set using the field FLEX_US_FMR.RXFTHRES2. Each time the Receive FIFO level goes from ‘above threshold 2’ to ‘equal to or below threshold 2’, the flag FLEX_US_FESR.RXFTHF2 is set. The application is warned that the Receive FIFO has reached the defined threshold and that it can be read to prevent an underflow.

The TXFTHF, RXFTHF and RXTHF2 flags can be configured to generate an interrupt using FLEX_US_FIER and FLEX_US_FIDR.

45.7.11.11 FIFO Flags

FIFOs come with a set of flags which can be configured to generate interrupts through FLEX_US_FIER and FLEX_US_FIDR.

FIFO flags state can be read in FLEX_US_FESR. They are cleared by writing FLEX_US_CR.RSTSTA to ‘1’.

45.7.12 16-bit Data Protocol Support

When configuring 0xC in the USART_MODE field, the transmitter sends a 16-bit data frame and the receiver expects an 8-bit data frame. The number of stop bits is defined in the NBSTOP field.

When configuring 0xD in the USART_MODE field, the transmitter sends an 8-bit frame whereas the receiver expects a 16-bit frame.

The FIFO mode must be enabled by setting FLEX_US_CR.FIFOEN to ‘1’.

A 16-bit frame starts as soon as two 8-bit characters are written in the FLEX_US_THR (assuming 0xC is written in the USART_MODE field).

45.7.13 USART Register Write Protection

The FLEXCOM operating mode (FLEX_MR.OPMODE) must be set to FLEX_MR_OPMODE_USART to enable access to the write protection registers.

To prevent any single software error from corrupting USART behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the USART Write Protection Mode Register (FLEX_US_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the USART Write Protection Status Register (FLEX_US_WPSR) is set and the Write Protection Violation Source (WPVSRC) field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading FLEX_US_WPSR.

Downloaded from Arrow.com.

The following registers can be write-protected when WPEN is set:

- USART Mode Register
- USART Baud Rate Generator Register
- USART Receiver Timeout Register
- USART Transmitter Timeguard Register
- USART FI DI RATIO Register
- USART IrDA FILTER Register
- USART Manchester Configuration Register
- USART LON Mode Register
- USART LON Beta1 Tx Register
- USART LON Beta1 Rx Register
- USART LON Priority Register
- USART LON IDT Tx Register
- USART LON IDT Rx Register
- USART IC DIFF Register
- USART Comparison Register

The following register(s) can be write-protected when WPITEN is set:

- USART Interrupt Enable Register
- USART Interrupt Disable Register

The following register(s) can be write-protected when WPCREN is set:

- USART Control Register

45.8 SPI Functional Description

45.8.1 Modes of Operation

The SPI operates in Master mode or in Slave mode.

- The SPI operates in Master mode by writing a 1 to the MSTR bit in the SPI Mode Register (FLEX_SPI_MR):
  - The pins NPCS0 to NPCS3 are all configured as outputs.
  - The SPCK pin is driven.
  - The MISO line is wired on the receiver input.
  - The MOSI line is driven as an output by the transmitter.
- The SPI operates in Slave mode if the MSTR bit in FLEX_SPI_MR is written to 0:
  - The MISO line is driven by the transmitter output.
  - The MOSI line is wired on the receiver input.
  - The SPCK pin is driven by the transmitter to synchronize the receiver.
  - The NPCS0 pin becomes an input, and is used as a slave select signal (NSS).
  - Pins NPCS1 to NPCS3 are not driven and can be used for other purposes.

The data transfers are identically programmable for both modes of operation. The bit rate generator is activated only in Master mode.

45.8.2 Data Transfer

Four combinations of polarity and phase are available for data transfers. The clock polarity is programmed with the CPOL bit in the SPI Chip Select Register (FLEX_SPI_CSR). The clock phase is programmed with the NCPHA bit. These two parameters determine the edges of the clock signal on which data are driven and sampled. Each of the two parameters has two possible states, resulting in four possible combinations that are incompatible with one another. Consequently, a master/slave pair must use the same parameter pair values to communicate. If multiple slaves are connected and require different configurations, the master must reconfigure itself each time it needs to communicate with a different slave.

Downloaded from Arrow.com.

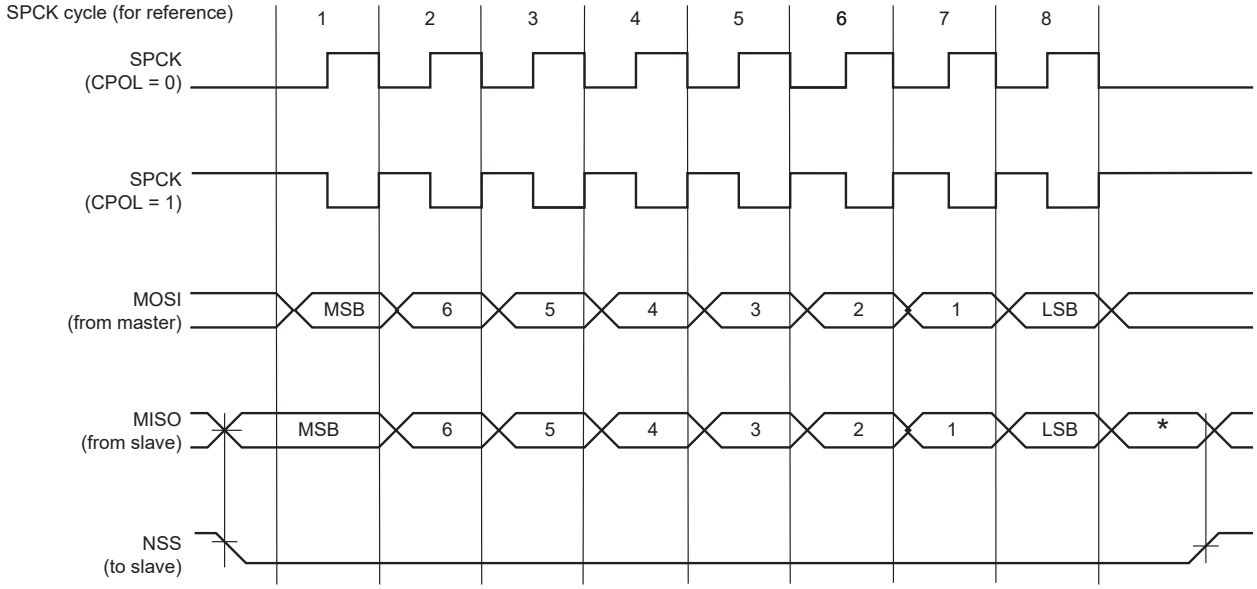
The following table shows the four modes and corresponding parameter settings.

Table 45-14. SPI Bus Protocol Mode

SPI Mode	CPOL	NCPHA	Shift SPCK Edge	Capture SPCK Edge	SPCK Inactive Level
0	0	1	Falling	Rising	Low
1	0	0	Rising	Falling	Low
2	1	1	Rising	Falling	High
3	1	0	Falling	Rising	High

The following figures show examples of data transfers.

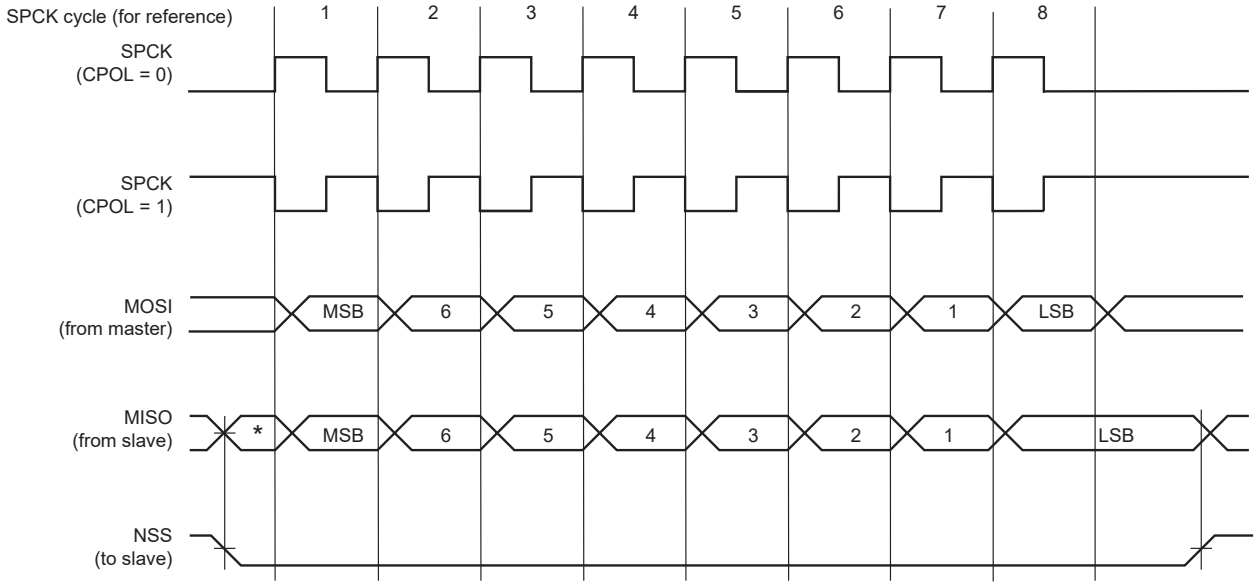
Figure 45-75. SPI Transfer Format (NCPHA = 1, 8 bits per transfer)



* Not defined.

Downloaded from Arrow.com.

Figure 45-76. SPI Transfer Format (NCPHA = 0, 8 bits per transfer)



* Not defined.

45.8.3 Master Mode Operations

When configured in Master mode, the SPI operates on the clock generated by the internal programmable bit rate generator. It fully controls the data transfers to and from the slave(s) connected to the SPI bus. The SPI drives the chip select line to the slave and the serial clock signal (SPCK).

The SPI features two holding registers, the Transmit Data Register (FLEX_SPI_TDR) and the Receive Data Register (FLEX_SPI_RDR), and a single shift register. The holding registers maintain the data flow at a constant rate.

After enabling the SPI, a data transfer starts when the processor writes to FLEX_SPI_TDR. The written data are immediately transferred in the shift register and the transfer on the SPI bus starts. While the data in the shift register is shifted on the MOSI line, the MISO line is sampled and shifted in the shift register. Data cannot be loaded in FLEX_SPI_RDR without transmitting data. If there is no data to transmit, a dummy data can be used (FLEX_SPI_TDR filled with ones). When the WDRBT bit is set, a new data cannot be transmitted if FLEX_SPI_RDR has not been read. If Receiving mode is not required, for example when communicating with a slave receiver only (such as an LCD), the receive status flags in the SPI Status Register (FLEX_SPI_SR) can be discarded.

Before writing the TDR, the FLEX_SPI_MR.PCS field must be set in order to select a slave.

If new data are written in FLEX_SPI_TDR during the transfer, it is kept in FLEX_SPI_TDR until the current transfer is completed. Then, the received data are transferred from the shift register to FLEX_SPI_RDR, the data in FLEX_SPI_TDR is loaded in the shift register and a new transfer starts.

As soon as the FLEX_SPI_TDR is written, the Transmit Data Register Empty (TDRE) flag in FLEX_SPI_SR is cleared. When the data written in FLEX_SPI_TDR is loaded into the shift register, the FLEX_SPI_SR.TDRE flag is set. The TDRE bit is used to trigger the Transmit DMA channel (see figure below).

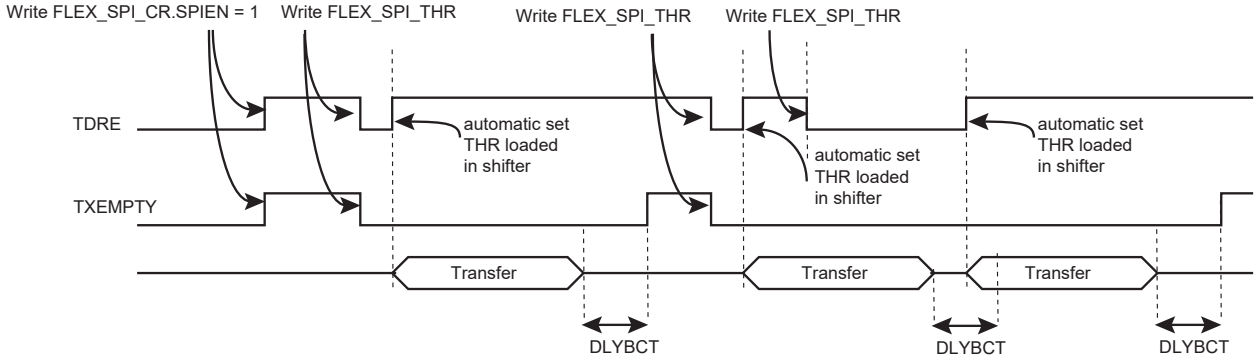
The end of transfer is indicated by FLEX_SPI_SR.TXEMPTY. If a transfer delay (DLYBCT) is greater than 0 for the last transfer, TXEMPTY is set after the completion of this delay. The peripheral clock can be switched off at this time.

Note:

1. When the SPI is enabled, the TDRE and TXEMPTY flags are set.
2. The TXEMPTY flag alone cannot be used to detect the end of the buffer DMA transfer.

Downloaded from Arrow.com.

Figure 45-77. TDRE and TXEMPTY Flag Behavior



The transfer of received data from the shift register to FLEX_SPI_RDR is indicated by the Receive Data Register Full (RDRF) bit in FLEX_SPI_SR. When the received data are read, the RDRF bit is cleared.

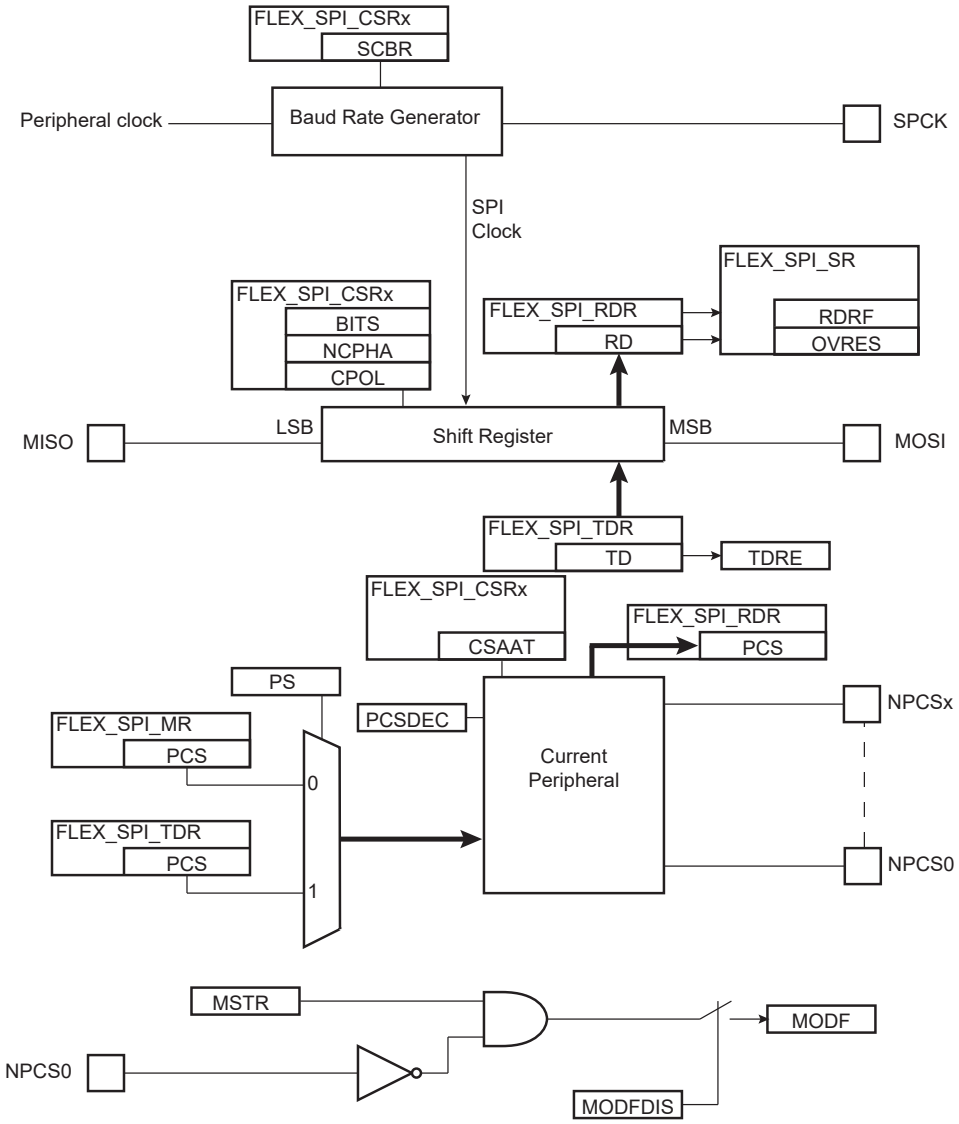
If FLEX_SPI_RDR has not been read before new data are received, the Overrun Error bit (OVRES) in FLEX_SPI_SR is set. As long as this flag is set, data are loaded in FLEX_SPI_RDR. The user has to read the status register to clear the OVRES bit.

The following figures show, respectively, a block diagram of the SPI when operating in Master mode and a flow chart describing how transfers are handled.

Downloaded from Arrow.com.

45.8.3.1 Master Mode Block Diagram

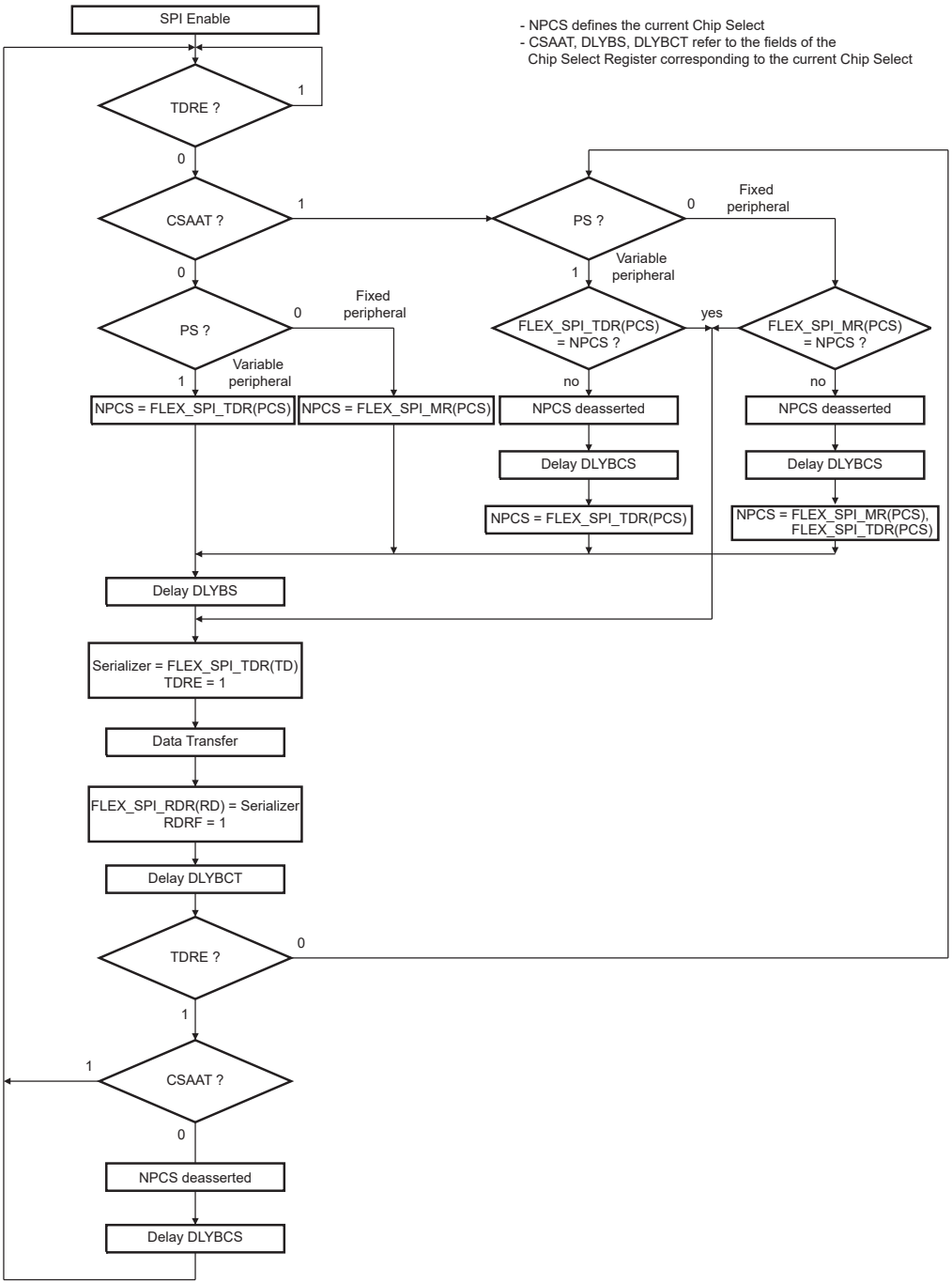
Figure 45-78. Master Mode Block Diagram



Downloaded from Arrow.com.

45.8.3.2 Master Mode Flowchart

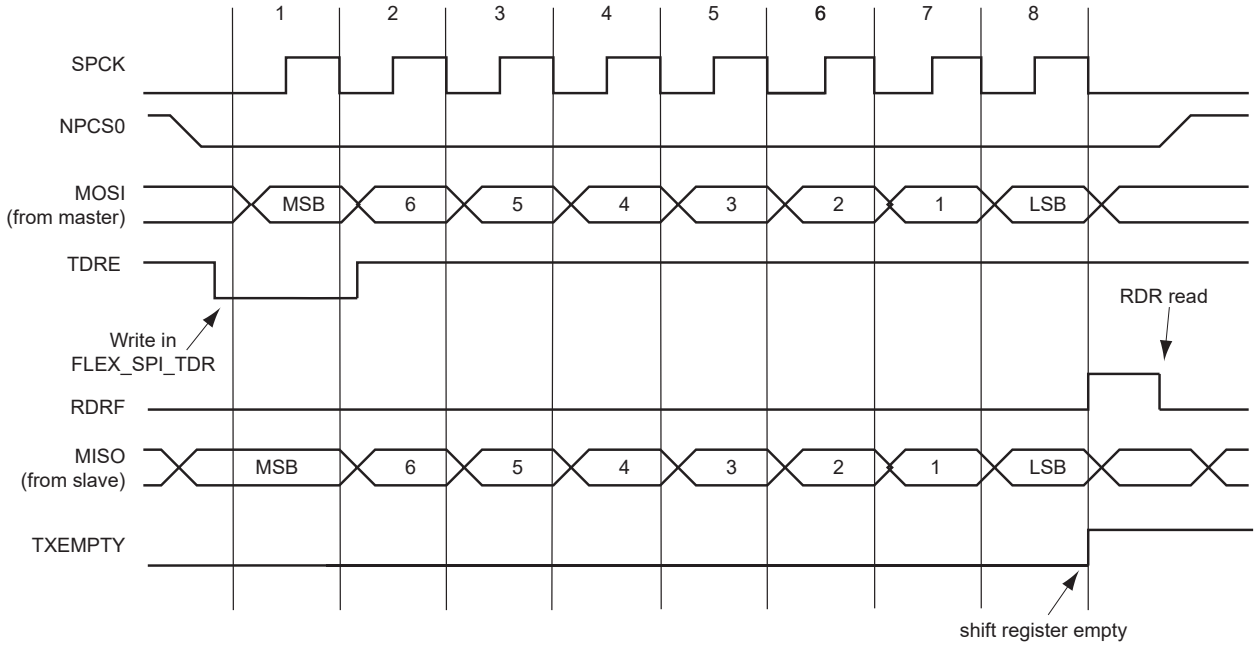
Figure 45-79. Master Mode



The following figure shows the behavior of Transmit Data Register Empty (TDRE), Receive Data Register (RDRF) and Transmission Register Empty (TXEMPTY) status flags within FLEX_SPI_SR during an 8-bit data transfer in Fixed mode without the DMAC involved.

Downloaded from Arrow.com.

Figure 45-80. Status Register Flags Behavior



45.8.3.3 Clock Generation

The SPI bit rate clock is generated by dividing a source clock which can be the peripheral clock or a programmable clock from the GCLK. The divider can be a value between 1 and 255.

If the SCBR field is programmed to 1 and the clock source is GCLK, the operating bit rate is peripheral clock (refer to the section “Electrical Characteristics” for the SPCK maximum frequency). Triggering a transfer while SCBR is at 0 can lead to unpredictable results.

At reset, SCBR is 0 and the user has to program it to a valid value before performing the first transfer.

The divisor can be defined independently for each chip select, as it has to be programmed in the FLEX_SPI_CSR.SCBR field. This allows the SPI to automatically adapt the bit rate for each interfaced peripheral without reprogramming.

If GCLK is selected as source clock (FLEX_SPI_MR.BRSRCCLK = 1), the bit rate is independent of the processor/bus clock. Thus, the processor clock can be changed while SPI is enabled. The processor clock frequency changes must be performed only by programming the PMC_MCKR.PRES field (refer to the section “Power Management Controller” (PMC)). Any other method to modify the processor/bus clock frequency (PLL multiplier, etc.) is forbidden when SPI is enabled.

The peripheral clock frequency must be at least three times higher than GCLK.

45.8.3.4 Transfer Delays

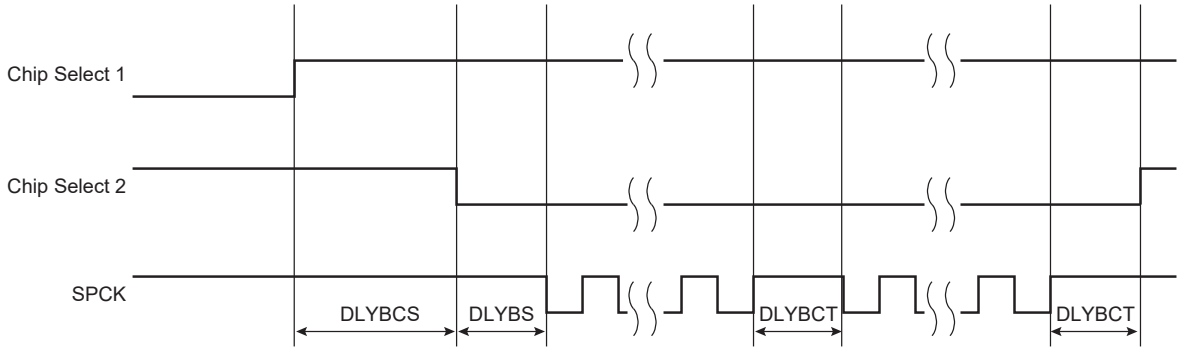
The figure below shows a chip select transfer change and consecutive transfers on the same chip select. Three delays can be programmed to modify the transfer waveforms:

- The delay between the chip selects. It is programmable only once for all chip selects by writing the FLEX_SPI_MR.DLYBCS field. The SPI slave device deactivation delay is managed through DLYBCS. If there is only one SPI slave device connected to the master, the DLYBCS field does not need to be configured. If several slave devices are connected to a master, DLYBCS must be configured depending on the highest deactivation delay. Refer to the SPI slave device electrical characteristics.
- The delay before SPCK, independently programmable for each chip select by writing the DLYBS field. The SPI slave device activation delay is managed through DLYBS. Refer to the SPI slave device electrical characteristics to define DLYBS.
- The delay between consecutive transfers, independently programmable for each chip select by writing the DLYBCT field. The time required by the SPI slave device to process received data is managed through DLYBCT. This time depends on the SPI slave system activity.

Downloaded from Arrow.com.

These delays allow the SPI to be adapted to the interfaced peripherals and their speed and bus release time.

Figure 45-81. Programmable Delays



45.8.3.5 Peripheral Selection

The serial peripherals are selected through the assertion of the NPCS0 to NPCS3 signals. By default, all NPCS signals are high before and after each transfer.

- Fixed Peripheral Select Mode: SPI exchanges data with only one peripheral.
  Fixed Peripheral Select mode is enabled by writing the FLEX_SPI_MR.PS bit to zero. In this case, the current peripheral is defined by the FLEX_SPI_MR.PCS field, and the FLEX_SPI_TDR. PCS field has no effect.
- Variable Peripheral Select Mode: Data can be exchanged with more than one peripheral without having to reprogram FLEX_SPI_MR.PCS.
  Variable Peripheral Select Mode is enabled by setting the FLEX_SPI_MR.PS bit to one. The FLEX_SPI_TDR.PCS field is used to select the current peripheral. This means that the peripheral selection can be defined for each new data. The value must be written in a single access to FLEX_SPI_TDR in the following format:

  [xxxxxxx(7-bit) + LASTXFER(1-bit)(1)+ xxxx(4-bit) + PCS (4-bit) + TD (8 to 16-bit data)]
  with LASTXFER at 0 or 1 depending on the CSAAT bit, and PCS equal to the chip select to assert, as defined in section SPI Transmit Data Register (FLEX_SPI_TDR).
  Note: 1. Optional

  The CSAAT, LASTXFER and CSNAAT bits are discussed in section Peripheral Deselection with DMA.

  If LASTXFER is used, the command must be issued after writing the last character. Instead of LASTXFER, the user can use the SPIDIS command. After the end of the DMA transfer, it is necessary to wait for the TXEMPTY flag and then write SPIDIS into the SPI Control Register (FLEX_SPI_CR). This does not change the configuration register values). The NPCS is disabled after the last character transfer. Then, another DMA transfer can be started if the FLEX_SPI_CR.SPIEN bit has previously been written.

45.8.3.6 SPI Direct Access Memory Controller (DMAC)

In both Fixed and Variable modes, the Direct Memory Access Controller (DMAC) can be used to reduce processor overhead.

The fixed peripheral selection allows buffer transfers with a single peripheral. Using the DMAC is an optimal means, as the size of the data transfer between the memory and the SPI is either 8 bits or 16 bits. However, if the peripheral selection is modified, FLEX_SPI_MR must be reprogrammed.

The variable peripheral selection allows buffer transfers with multiple peripherals without reprogramming FLEX_SPI_MR. Data written in FLEX_SPI_TDR is 32 bits wide and defines the real data to be transmitted and the destination peripheral. Using the DMAC in this mode requires 32-bit wide buffers, with the data in the LSBs and the PCS and LASTXFER fields in the MSBs. However, the SPI still controls the number of bits (8 to 16) to be transferred through MISO and MOSI lines with the chip select configuration registers. This is not the optimal means in terms of memory size for the buffers, but it provides a very effective means to exchange data with several peripherals without any intervention of the processor.

Downloaded from Arrow.com.

45.8.3.7 Peripheral Chip Select Decoding

The user can program the SPI to operate with up to 15 slave peripherals by decoding the four chip select lines, NPCS0 to NPCS3 with an external decoder/demultiplexer (see the following figure). This can be enabled by setting the FLEX_SPI_MR.PCSDEC bit.

When operating without decoding, the SPI makes sure that in any case only one chip select line is activated, i.e., one NPCS line driven low at a time. If two bits are defined low in a PCS field, only the lowest numbered chip select is driven low.

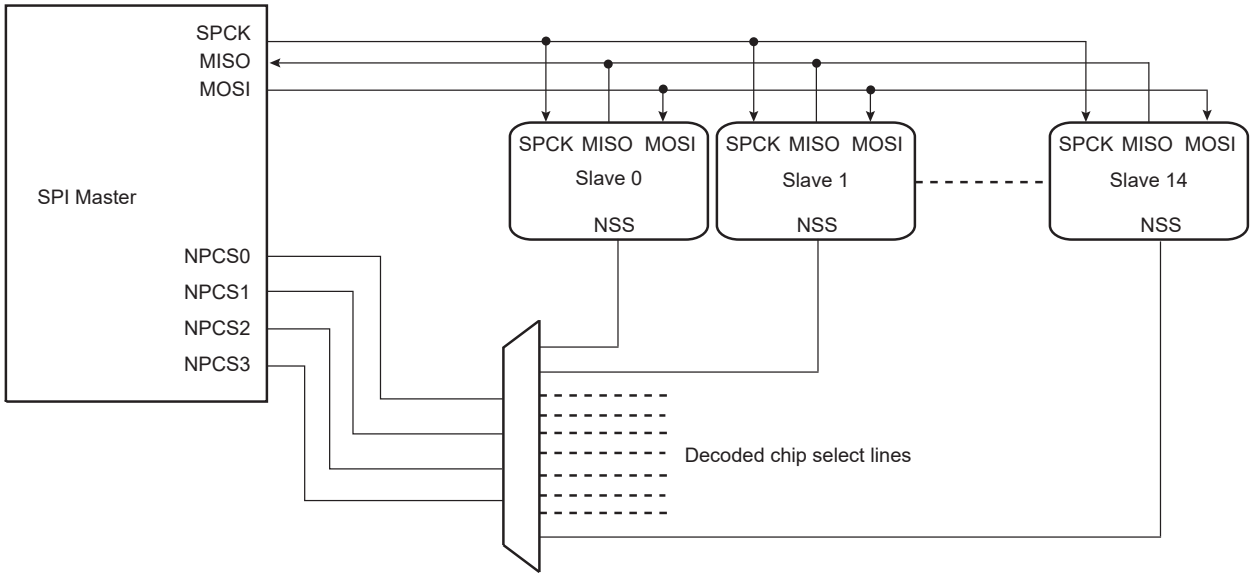
When operating with decoding, the SPI directly outputs the value defined by the PCS field on the NPCS lines of either FLEX_SPI_MR or FLEX_SPI_TDR (depending on PS).

As the SPI sets a default value of 0xF on the chip select lines (i.e., all chip select lines at 1) when not processing any transfer, only 15 peripherals can be decoded.

The SPI has only four Chip Select registers. As a result, when external decoding is activated, each NPCS chip select defines the characteristics of up to four peripherals. As an example, FLEX_SPI_CRS0 defines the characteristics of the externally decoded peripherals 0 to 3, corresponding to the PCS values 0x0 to 0x3. Consequently, the user has to make sure to connect compatible peripherals on the decoded chip select lines 0 to 3, 4 to 7, 8 to 11 and 12 to 14. The following figure shows this type of implementation.

If the CSAAT bit is used, with or without the DMAC, the mode fault detection for NPCS0 line must be disabled. This is not needed for all other chip select lines since mode fault detection is only on NPCS0.

Figure 45-82. Chip Select Decoding Application Block Diagram: Single Master/Multiple Slave Implementation



45.8.3.8 Peripheral Deselection without DMA

During a transfer of more than one data on a Chip Select without the DMA, FLEX_SPI_TDR is loaded by the processor, the TDRE flag rises as soon as the content of FLEX_SPI_TDR is transferred into the internal shift register. When this flag is detected high, FLEX_SPI_TDR can be reloaded. If this reload by the processor occurs before the end of the current transfer, and if the next transfer is performed on the same chip select as the current transfer, the Chip Select is not deasserted between the two transfers. But depending on the application software handling the SPI status register flags (by interrupt or polling method) or servicing other interrupts or other tasks, the processor may not reload FLEX_SPI_TDR in time to keep the chip select active (low). A null DLYBCT value (delay between consecutive transfers) in FLEX_SPI_CSR, gives even less time for the processor to reload FLEX_SPI_TDR. With some SPI slave peripherals, if the chip select line must remain active (low) during a full set of transfers, communication errors can occur.

To facilitate interfacing with such devices, the Chip Select registers [CSR0...CSR3] can be programmed with the Chip Select Active After Transfer (CSAAT) bit to 1. This allows the chip select lines to remain in their current state (low = active) until a transfer to another chip select is required. Even if FLEX_SPI_TDR is not reloaded, the chip select

Downloaded from Arrow.com.

remains active. To de-assert the chip select line at the end of the transfer, the Last Transfer (LASTXFER) bit in FLEX_SPI_CR must be set after writing the last data to transmit into FLEX_SPI_TDR.

45.8.3.9 Peripheral Deselection with DMA

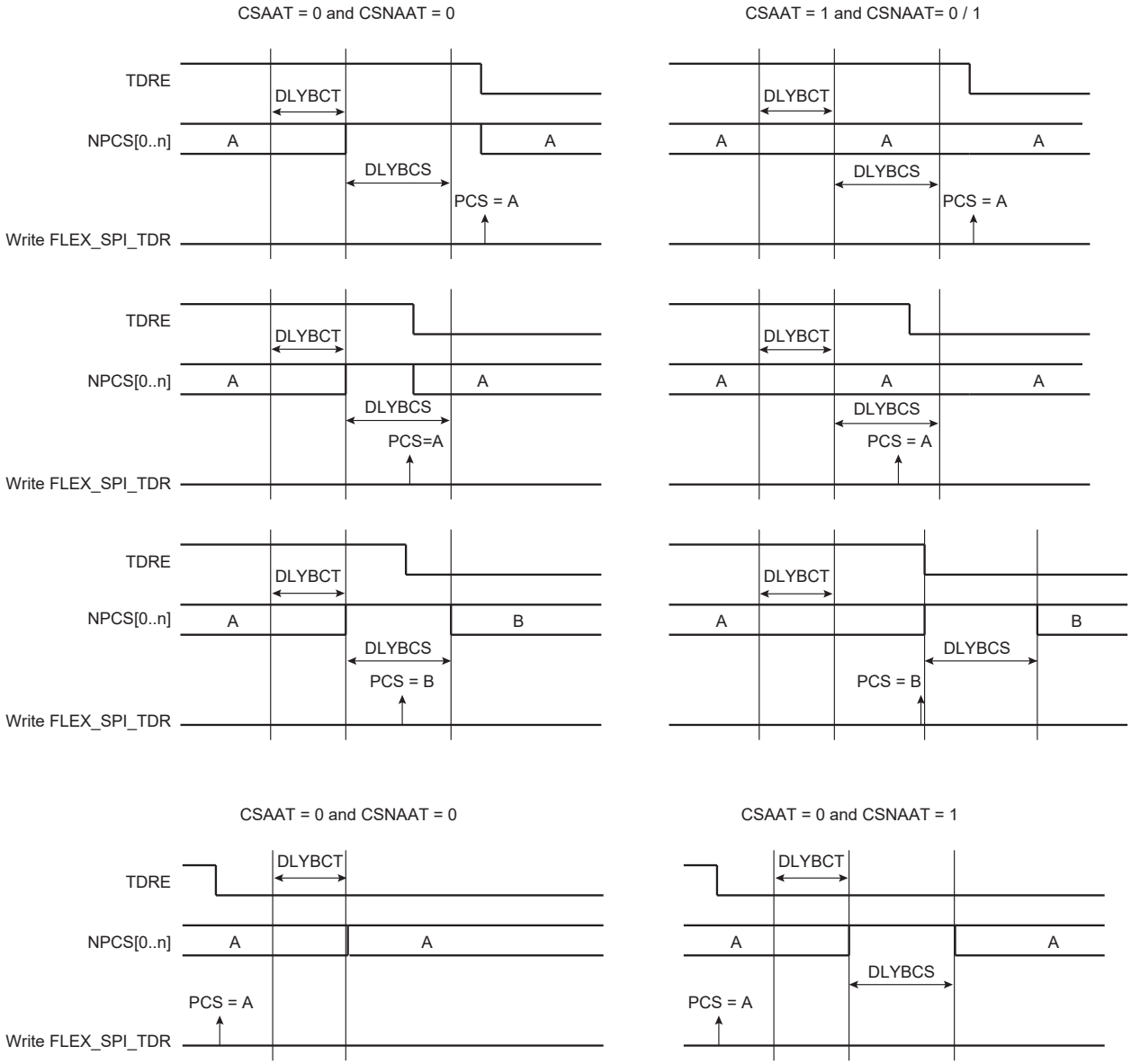
DMA provides faster reloads of FLEX_SPI_TDR compared to software. However, depending on the system activity, it is not guaranteed that FLEX_SPI_TDR is written with the next data before the end of the current transfer. Consequently, a data can be lost by the deassertion of the NPCS line for SPI slave peripherals requiring the chip select line to remain active between two transfers. The only way to guarantee a safe transfer in this case is the use of the CSAAT and LASTXFER bits.

When the CSAAT bit is cleared, the NPCS does not rise in all cases between two transfers on the same peripheral. During a transfer on a Chip Select, the TDRE flag rises as soon as the content of FLEX_SPI_TDR is transferred into the internal shift register. When this flag is detected, FLEX_SPI_TDR can be reloaded. If this reload occurs before the end of the current transfer and if the next transfer is performed on the same chip select as the current transfer, the Chip Select is not deasserted between the two transfers. This can lead to difficulties to interface with some serial peripherals requiring the chip select to be deasserted after each transfer. To facilitate interfacing with such devices, FLEX_SPI_CSR can be programmed with the Chip Select Not Active After Transfer (CSNAAT) bit to 1. This allows the chip select lines to be deasserted systematically during a time “DLYBCS” (the value of the CSNAAT bit is processed only if the CSAAT bit is cleared for the same chip select).

The following figure shows different peripheral deselection cases and the effect of the CSAAT and CSNAAT bits.

Downloaded from Arrow.com.

Figure 45-83. Peripheral Deselection



45.8.3.10 Mode Fault Detection

The SPI has the capability to operate in multi-master environment. Consequently, the NPCS0/NSS line must be monitored. If one of the masters on the SPI bus is currently transmitting, the NPCS0/NSS line is low and the SPI must not transmit a data. A mode fault is detected when the SPI is programmed in Master mode and a low level is driven by an external master on the NPCS0/NSS signal. In multi-master environment, NPCS0, MOSI, MISO and SPCK pins must be configured in open drain (through the PIO controller). When a mode fault is detected, the FLEX_SPI_SR.MODF bit is set until FLEX_SPI_SR is read and the SPI is automatically disabled until it is re-enabled by writing the FLEX_SPI_CR.SPIEN bit to 1.

By default, the mode fault detection is enabled. The user can disable it by setting the FLEX_SPI_MR.MODFDIS bit.

45.8.4 SPI Slave Mode

When operating in Slave mode, the SPI processes data bits on the clock provided on the SPI clock pin (SPCK).

The SPI waits until NSS goes active before receiving the serial clock from an external master. When NSS falls, the clock is validated and the data are loaded in FLEX_SPI_RDR according to the configuration value of the FLEX_SPI_CSR0.BITS field. These bits are processed following a phase and a polarity defined respectively by the

Downloaded from Arrow.com.

FLEX_SPI_CSR0.NCPHA and FLEX_SPI_CSR0.CPOL bits. Note that the BITS field, CPOL bit and NCPHA bit of the other Chip Select registers have no effect when the SPI is programmed in Slave mode.

The bits are shifted out on the MISO line and sampled on the MOSI line.

Note: For more information on the BITS field, see also the note below the FLEX_SPI_CSRx register bitmap in section SPI Chip Select Register

When all bits are processed, the received data are transferred in FLEX_SPI_RDR and the RDRF bit rises. If FLEX_SPI_RDR has not been read before new data are received, the Overrun Error bit (OVRES) in FLEX_SPI_SR is set. As long as this flag is set, data are loaded in FLEX_SPI_RDR. The user must read FLEX_SPI_SR to clear the OVRES bit.

When a transfer starts, the data shifted out is the data present in the shift register. If no data has been written in FLEX_SPI_TDR, the last data received is transferred. If no data has been received since the last reset, all bits are transmitted low, as the shift register resets to 0.

When a first data is written in FLEX_SPI_TDR, it is transferred immediately in the shift register and the TDRE flag rises. If new data is written, it remains in FLEX_SPI_TDR until a transfer occurs, i.e., NSS falls and there is a valid clock on the SPCK pin. When the transfer occurs, the last data written in FLEX_SPI_TDR is transferred in the shift register and the TDRE flag rises. This enables frequent updates of critical variables with single transfers.

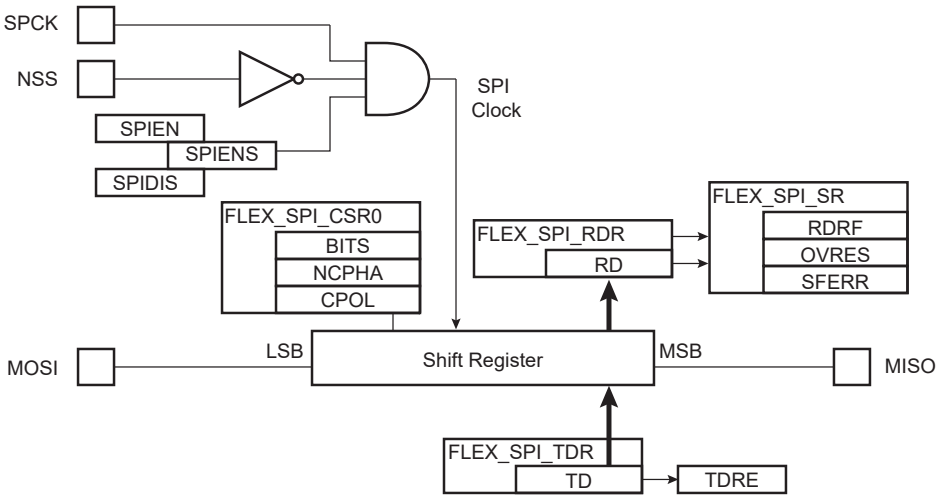
Then, a new data is loaded in the shift register from FLEX_SPI_TDR. If no character is ready to be transmitted, i.e., no character has been written in FLEX_SPI_TDR since the last load from FLEX_SPI_TDR to the shift register, FLEX_SPI_TDR is retransmitted. In this case the Underrun Error Status Flag (UNDES) is set in FLEX_SPI_SR.

If NSS rises between two characters, it must be kept high for two MCK clock periods or more and the next SPCK capture edge must not occur less than four MCK periods after NSS rise.

In slave mode, if the NSS line rises and the received character length does not match the configuration defined in FLEX_SPI_CSR0.BITS, the flag SFERR is set in FLEX_SPI_SR.

The following figure shows a block diagram of the SPI when operating in Slave mode.

Figure 45-84. Slave Mode Functional Block Diagram



45.8.5 SPI Comparison Function on Received Character

The comparison is only relevant for SPI Slave mode (MSTR = 0 in FLEX_US_MR).

In Active mode, the CMP flag in FLEX_SPI_SR is raised. It is set when the received character matches the conditions programmed in the SPI Comparison Register (FLEX_SPI_CMPR). The CMP flag is set as soon as FLEX_SPI_RDR is loaded with the new received character. The CMP flag is cleared by reading FLEX_SPI_SR.

The SPI Comparison Register can be programmed to provide different comparison methods. These are listed below:

- If VAL1 equals VAL2, then the comparison is performed on a single value and the flag is set to 1 if the received character equals VAL1.

Downloaded from Arrow.com.

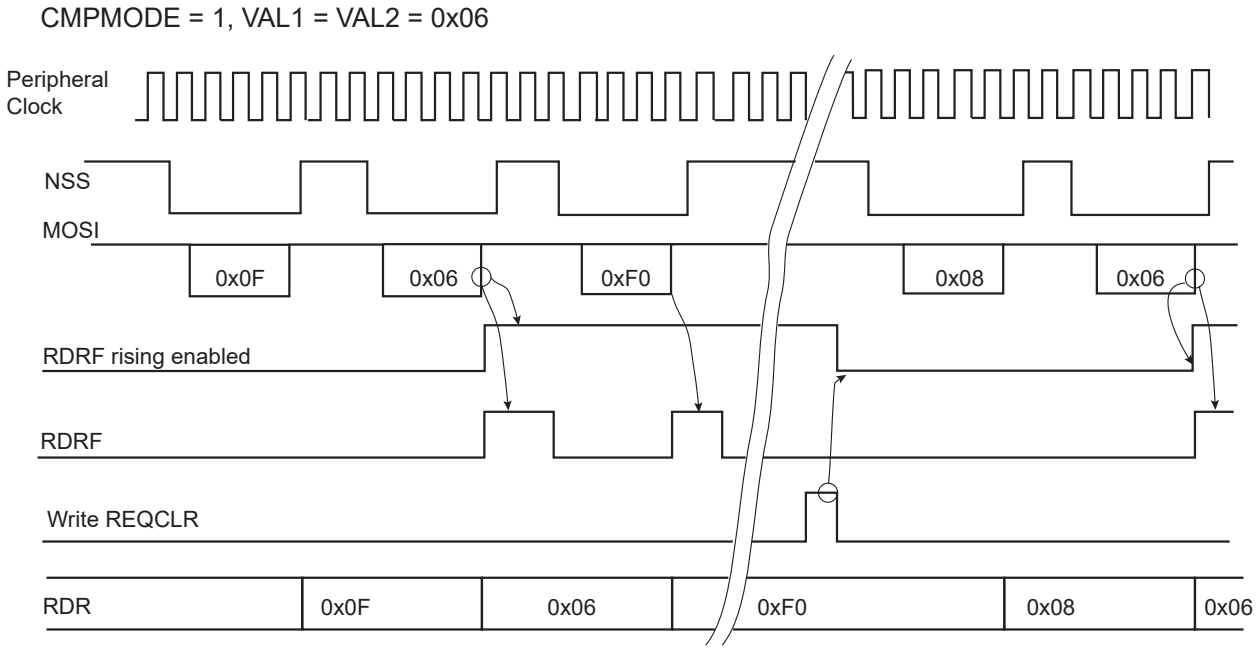
- If VAL1 is strictly lower than VAL2, then any value between VAL1 and VAL2 sets the CMP flag.
- If VAL1 is strictly higher than VAL2, then the flag CMP is set to 1 if any received character equals VAL1 or VAL2.

When FLEX_SPI_MR.CMPMODE is cleared, all received data is loaded in FLEX_SPI_RDR and the CMP flag provides the status of the comparison result.

By setting the CMPMODE bit, the comparison result triggers the start of FLEX_SPI_RDR loading (see the figure below). The trigger condition exists as soon as the received character value matches the conditions defined by VAL1 and VAL2 in FLEX_SPI_CMPR. The comparison trigger event is restarted by writing a 1 to the FLEX_SPI_CR.REQCLR bit.

The value programmed in VAL1 and VAL2 fields must not exceed the maximum value of the received character (see BITS field in SPI Chip Select Register (FLEX_SPI_CSR)).

Figure 45-85. Receive Data Register Management



45.8.6 SPI FIFOs

45.8.6.1 Overview

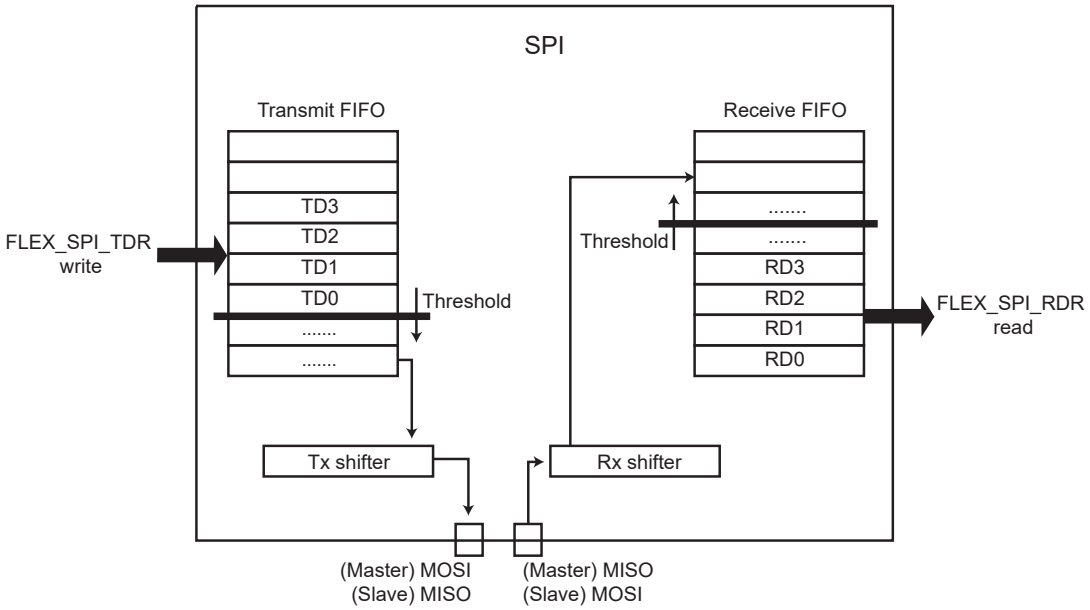
The SPI includes two FIFOs which can be enabled/disabled using the FLEX_SPI_CR.FIFOEN/FIFODIS. The SPI module must be disabled before enabling or disabling the SPI FIFOs (FLEX_SPI_CR.SPIDIS).

Writing FLEX_SPI_CR.FIFOEN to ‘1’ enables a FIFO_DEPTH-data Transmit FIFO and a FIFO_DEPTH-data Receive FIFO.

It is possible to write or to read single or multiple data in the same access to FLEX_SPI_TDR/RDR. See sections SPI Single Data Mode and SPI Multiple Data Mode.

Downloaded from Arrow.com.

Figure 45-86. FIFOs Block Diagram



45.8.6.2 Sending Data with FIFO Enabled

When the Transmit FIFO is enabled, write access to FLEX_SPI_TDR loads the Transmit FIFO.

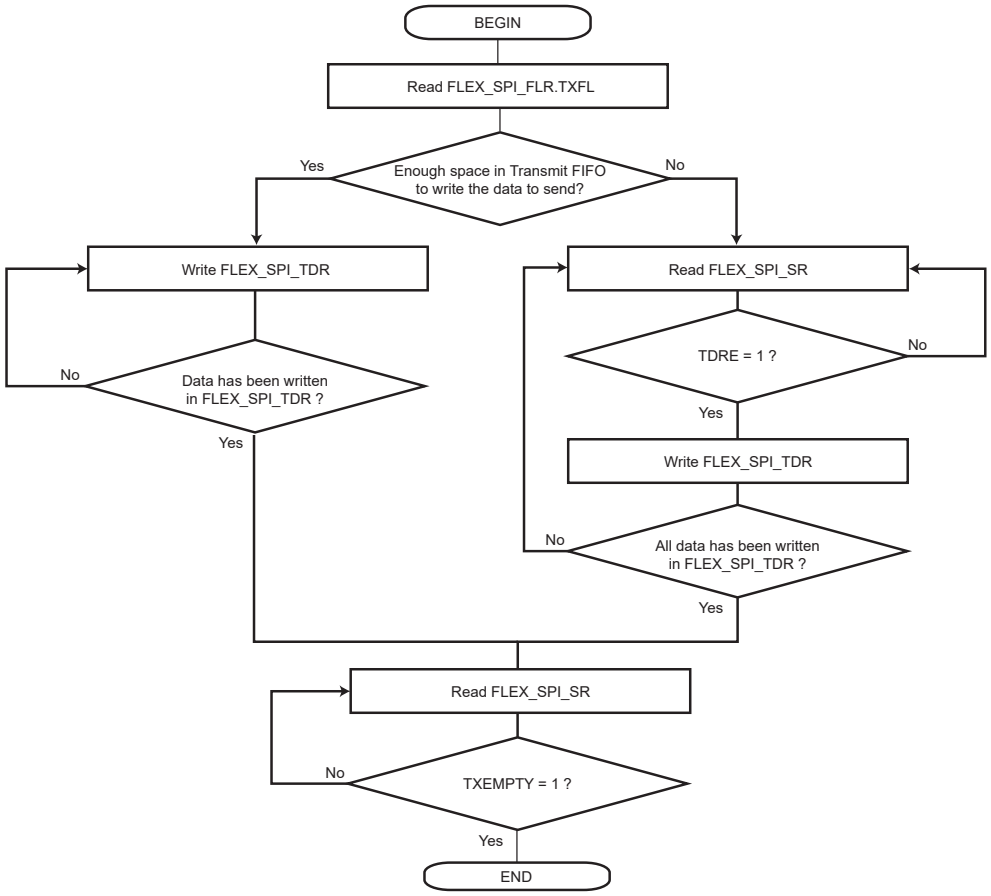
The FIFO level is provided in FLEX_SPI_FLR.TXFL. If the FIFO can accept the number of data to be transmitted, there is no need to monitor FLEX_SPI_SR.TDRE and the data can be successively written in FLEX_SPI_TDR.

If the FIFO cannot accept the data due to insufficient space, wait for the TDRE flag to be set before writing the data in FLEX_SPI_TDR.

When the space in the FIFO allows only a portion of the data to be written, the TDRE flag must be monitored before writing the remaining data.

Downloaded from Arrow.com.

Figure 45-87. Sending Data with FIFO Enabled



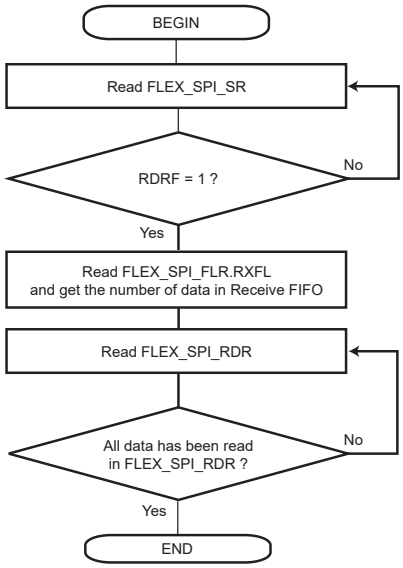
45.8.6.3 Receiving Data with FIFO Enabled

When the Receive FIFO is enabled, FLEX_SPI_RDR access reads the FIFO.

When data are present in the Receive FIFO (RDRF flag set to ‘1’), the exact number of data can be checked with FLEX_SPI_FLR.RXFL. All the data can be read successively in FLEX_SPI_RDR without checking the RDRF flag between each access.

Downloaded from Arrow.com.

Figure 45-88. Receiving Data with FIFO Enabled



45.8.6.4 Clearing/Flushing FIFOs

Each FIFO can be cleared/flushed using FLEX_SPI_CR.TXFCLR/RXFCLR.

45.8.6.5 TXEMPTY, TDRE and RDRF Behavior

FLEX_SPI_SR.TXEMPTY, FLEX_SPI_SR.TDRE and FLEX_SPI_SR.RDRF flags display a specific behavior when FIFOs are enabled.

The TXEMPTY flag is cleared as long as there are characters in the Transmit FIFO or in the internal shift register. TXEMPTY is set when there are no characters in the Transmit FIFO and in the internal shift register.

TDRE indicates if a data can be written in the Transmit FIFO. Thus the TDRE flag is set as long as the Transmit FIFO can accept new data. See figure TDRE in Single Data Mode and TXRDYM = 0.

RDRF indicates if an unread data is present in the Receive FIFO. Thus the RDRF flag is set as soon as one unread data is in the Receive FIFO. See figure RDRF in Single Data Mode and RXRDYM = 0.

TDRE and RDRF behavior can be modified using the TXRDYM and RXRDYM fields in the SPI FIFO Mode Register (FLEX_SPI_FMR) to reduce the number of accesses to FLEX_SPI_TDR/RDR. However, for some configurations, the following constraints apply:

- When the Variable Peripheral Select mode is used (FLEX_SPI_MR.PS=1), TXRDYM/RXRDYM must be cleared.
- In Master mode (FLEX_SPI_MR.MSTR=1), RXRDYM must be cleared.

As an example, in Master mode, the Transmit FIFO can be loaded with multiple data in the same access by configuring TXRDYM>0.

See SPI FIFO Mode Register (FLEX_SPI_FMR) for the FIFO configuration.

Downloaded from Arrow.com.

Figure 45-89. TDRE in Single Data Mode and TXRDYM = 0

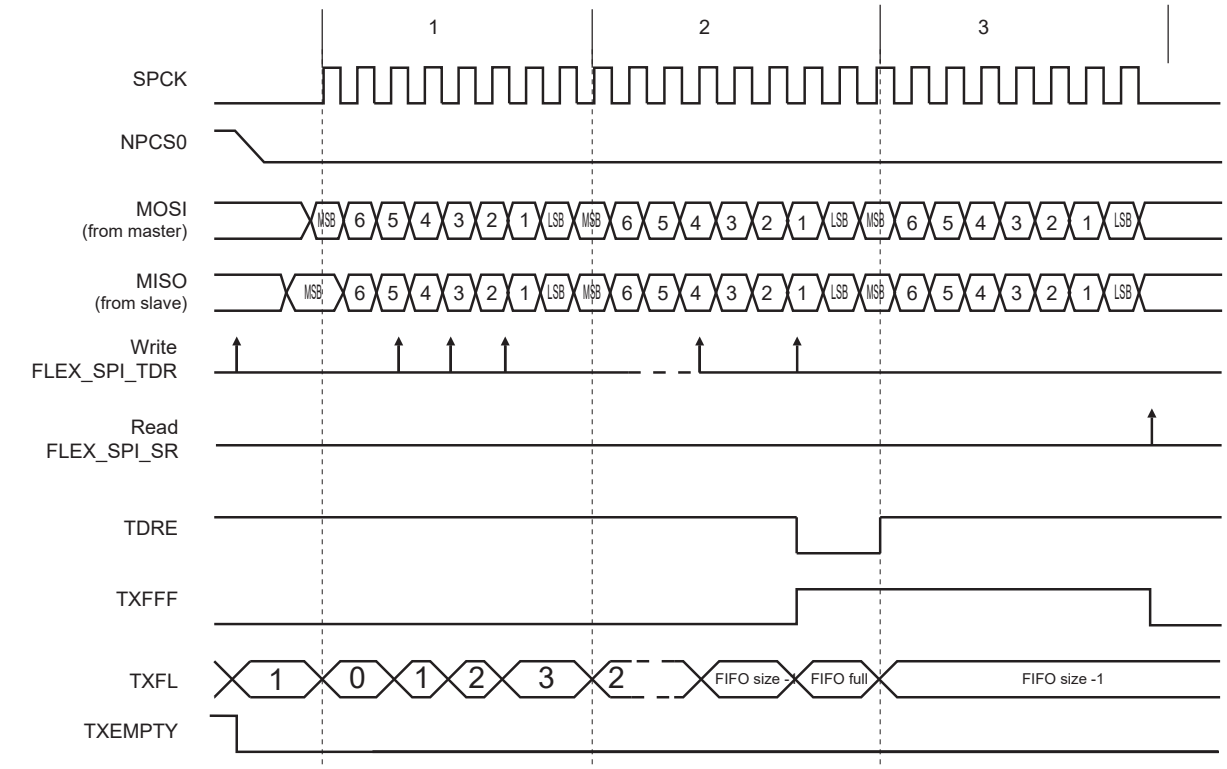
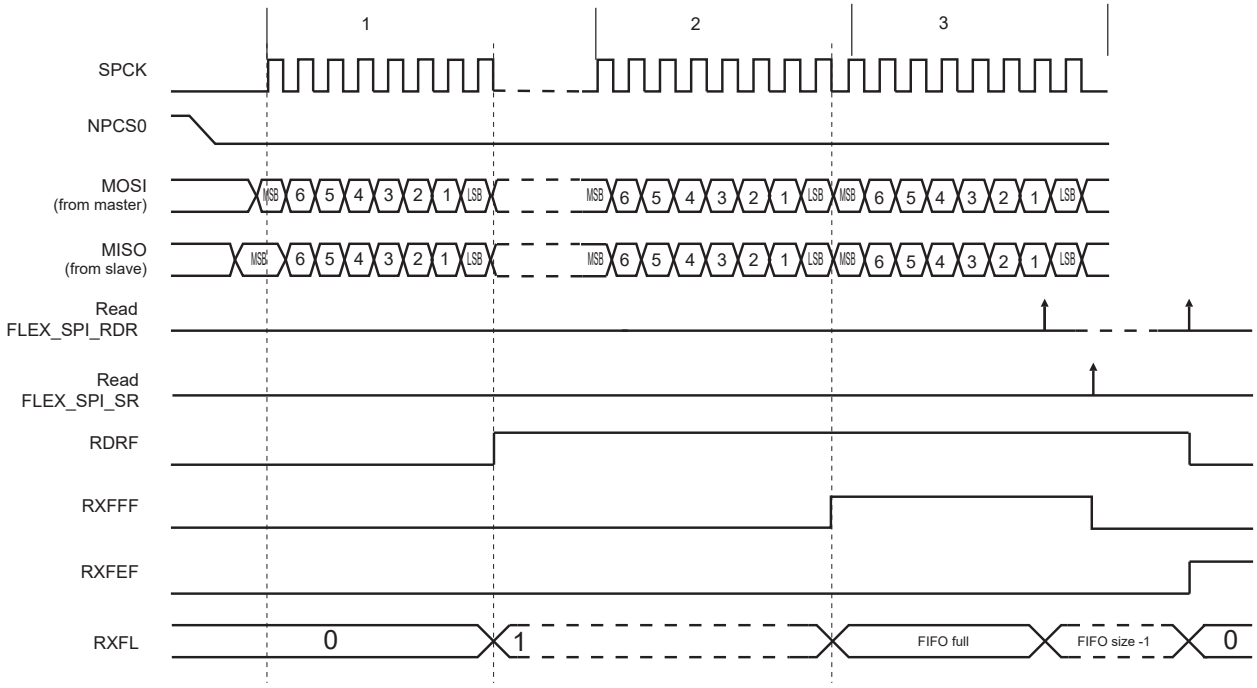


Figure 45-90. RDRF in Single Data Mode and RXRDYM = 0



45.8.6.6 SPI Single Data Mode

In Single Data mode, only one data is written every time FLEX_SPI_TDR is accessed, and only one data is read every time FLEX_SPI_RDR is accessed.

When FLEX_SPI_FMR.TXRDYM = 0, the Transmit FIFO operates in Single Data mode.

Downloaded from Arrow.com.

When FLEX_SPI_FMR.RXRDYM = 0, the Receive FIFO operates in Single Data mode.

If Master mode is used (FLEX_SPI_MR.MSTR=1), the Receive FIFO must operate in Single Data mode.

If Variable Peripheral Select mode is used (FLEX_SPI_MR.PS=1), the Transmit FIFO must operate in Single Data mode.

See sections SPI Transmit Data Register and SPI Receive Data Register.

45.8.6.6.1 DMA

When FIFOs operate in Single Data mode, the DMA transfer type must be configured either in bytes, halfwords or words depending on FLEX_SPI_MR.PS bit value and FLEX_SPI_CSRx.BITS field value.

The same applies when FIFOs are disabled.

45.8.6.7 SPI Multiple Data Mode

Multiple Data mode minimizes the number of accesses by concatenating the data to send/read in one access.

When FLEX_SPI_FMR.TXRDYM > 0, the Transmit FIFO operates in Multiple Data mode.

When FLEX_SPI_FMR.RXRDYM > 0, the Receive FIFO operates in Multiple Data mode.

Multiple data can be read from the Receive FIFO only in Slave mode (FLEX_SPI_MR.MSTR=0).

The Transmit FIFO can be loaded with multiple data in the same access by configuring TXRDYM>0 and when FLEX_SPI_MR.PS=0.

In Multiple Data mode, up to two data can be written in one FLEX_SPI_TDR write access. It is also possible to read up to four data in one FLEX_SPI_RDR access if FLEX_SPI_CSRx.BITS is configured to ‘0’ (8-bit data size) and up to two data if FLEX_SPI_CSRx.BITS is configured to a value other than ‘0’ (more than 8-bit data size).

The number of data to write/read is defined by the size of the register access. If the access is a byte-size register access, only one data is written/read. If the access is a halfword size register access, then up to two data are read and only one data is written. Lastly, if the access is a word-size register access, then up to four data are read and up to two data are written.

Written/read data are always right-aligned, as described in sections SPI Receive Data Register (FIFO Multiple Data, 8-bit), SPI Receive Data Register (FIFO Multiple Data, 16-bit) and SPI Transmit Data Register (FIFO Multiple Data, 8- to 16-bit).

As an example, if the Transmit FIFO is empty and there are six data to send, either of the following write accesses may be performed:

- six FLEX_SPI_TDR-byte write accesses
- three FLEX_SPI_TDR-halfword write accesses

With a Receive FIFO containing six data, any of the following read accesses may be performed:

- six FLEX_SPI_RDR-byte read accesses
- three FLEX_SPI_RDR-halfword read accesses
- one FLEX_SPI_RDR-word read access and one FLEX_SPI_RDR-halfword read access

45.8.6.7.1 TDRE and RDRF Configuration

In Multiple Data mode, it is possible to write one or more data in the same FLEX_SPI_TDR/RDR access. The TDRE flag indicates if one or more data can be written in the FIFO depending on the configuration of FLEX_SPI_FMR.TXRDYM/RXRDYM.

As an example, if two data are written each time in FLEX_SPI_TDR, it is useful to configure the TXRDYM field to the value ‘1’ so that the TDRE flag is at ‘1’ only when at least two data can be written in the Transmit FIFO.

Similarly, if four data are read each time in FLEX_SPI_RDR, it is useful to configure the RXRDYM field to the value ‘2’ so that the RDRF flag is at ‘1’ only when at least four unread data are in the Receive FIFO.

45.8.6.7.2 DMA

It is mandatory to configure DMA channel size (byte, halfword or word) according to FLEX_SPI_FMR.TXRDYM/RXRDYM configuration. See section SPI Multiple Data Mode for constraints.

Downloaded from Arrow.com.

45.8.6.8 FIFO Pointer Error

A FIFO overflow is reported in FLEX_SPI_SR.

If the Transmit FIFO is full and a write access is performed on FLEX_SPI_TDR, it generates a Transmit FIFO pointer error and sets FLEX_SPI_SR.TXFPTEF.

In Multiple Data mode, if the number of data written in FLEX_SPI_TDR (according to the register access size) is greater than the free space in the Transmit FIFO, a Transmit FIFO pointer error is generated and FLEX_SPI_SR.TXFPTEF is set.

A FIFO underflow is reported in FLEX_SPI_SR.

In Multiple Data mode, if the number of data read in FLEX_SPI_RDR (according to the register access size) is greater than the number of unread data in the Receive FIFO, a Receive FIFO pointer error is generated and FLEX_SPI_SR.RXFPTEF is set.

No pointer error occurs if the FIFO state/level is checked before writing/reading in FLEX_SPI_TDR/SPI_RDR. The FIFO state/level can be checked either with TXRDY, RXRDY, TXFL or RXFL. When a pointer error occurs, other FIFO flags may not behave as expected; their states should be ignored.

If a pointer error occurs, a software reset must be performed using FLEX_SPI_CR.SWRST (configuration will be lost).

45.8.6.9 FIFO Thresholds

Each Transmit and Receive FIFO includes a threshold feature used to set a flag and an interrupt when a FIFO threshold is crossed. Thresholds are defined as a number of data in the FIFO, and the FIFO state (TXFL or RXFL) represents the number of data currently in the FIFO.

The Transmit FIFO threshold can be set using the field FLEX_SPI_FMR.TXFTHRES. Each time the Transmit FIFO level goes from ‘above threshold’ to ‘equal to or below threshold’, the flag FLEX_SPI_SR.TXFTHF is set. The application is warned that the Transmit FIFO has reached the defined threshold and that it can be reloaded.

The Receive FIFO threshold can be set using the field FLEX_SPI_FMR.RXFTHRES. Each time the Receive FIFO level goes from ‘below threshold’ to ‘equal to or above threshold’, the flag FLEX_SPI_SR.RXFTHF is set. The application is warned that the Receive FIFO has reached the defined threshold and that it can be read to prevent an underflow.

The TXFTHF and RXFTHF flags can be configured to generate an interrupt using FLEX_SPI_IER and FLEX_SPI_IDR.

45.8.6.10 FIFO Flags

FIFOs come with a set of flags which can be configured to generate interrupts through FLEX_SPI_IER and FLEX_SPI_IDR.

FIFO flags state can be read in FLEX_SPI_SR. They are cleared when FLEX_SPI_SR is read.

45.8.7 SPI Register Write Protection

The FLEXCOM operating mode (FLEX_MR.OPMODE) must be set to FLEX_MR_OPMODE_SPI to enable access to the write protection registers.

To prevent any single software error from corrupting SPI behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the SPI Write Protection Mode Register (FLEX_SPI_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the SPI Write Protection Status Register (FLEX_SPI_WPSR) is set and the Write Protection Violation Source (WPVSRC) field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading FLEX_SPI_WPSR.

The following registers can be write-protected when WPEN is set:

- SPI Mode Register
- SPI Chip Select Register
- SPI Comparison Register

The following registers can be write-protected when WPITEN is set:

Downloaded from Arrow.com.

- SPI Interrupt Enable Register
- SPI Interrupt Disable Register

The following register can be write-protected when WPCREN is set:

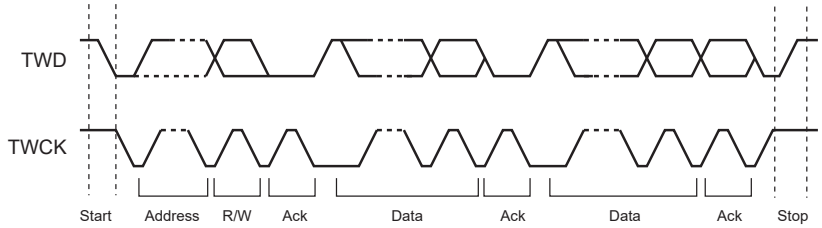
- SPI Control Register

45.9 TWI Functional Description

45.9.1 Transfer Format

The data put on the TWD line must be 8 bits long. Data are transferred MSB first; each byte must be followed by an acknowledgement. The number of bytes per transfer is unlimited (see figure below).

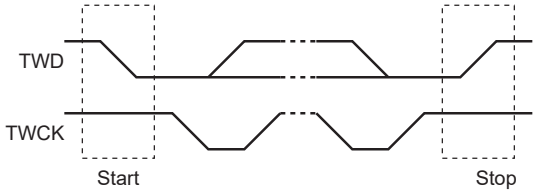
Figure 45-91. Transfer Format



Each transfer begins with a START condition and terminates with a STOP condition (see figure below).

- A high-to-low transition on the TWD line while TWCK is high defines the START condition.
- A low-to-high transition on the TWD line while TWCK is high defines a STOP condition.

Figure 45-92. START and STOP Conditions



45.9.2 Modes of Operation

The TWI has different modes of operation:

- Master Transmitter mode (Standard, Fast mode, Fast mode Plus)
- Master Receiver mode (Standard, Fast mode, Fast mode Plus)
- Multi-master Transmitter mode (Standard, Fast mode, Fast mode Plus)
- Multi-master Receiver mode (Standard, Fast mode, Fast mode Plus)
- Slave Transmitter mode (Standard, Fast mode, Fast mode Plus and High-speed mode)
- Slave Receiver mode (Standard, Fast mode, Fast mode Plus and High-speed mode)

These modes are described in the following sections.

45.9.3 Master Mode

45.9.3.1 Definition

The master is the device that starts a transfer, generates a clock and stops it.

45.9.3.2 Programming Master Mode

The following fields must be programmed before entering Master mode:

1. DADR (+ IADRSZ + IADR if a 10-bit device is addressed): The device address is used to access slave devices in Read or Write mode.

Downloaded from Arrow.com.

2. CWGR + CKDIV + CHDIV + CLDIV: Clock waveform.
3. SVDIS: Disables Slave mode.
4. MSEN: Enables Master mode.
   Note: If the TWI is already in Master mode, the device address (DADR) can be configured without disabling the Master mode.

45.9.3.3 Transfer Speed/Bit Rate

The TWI speed is defined in FLEX_TWI_CWGR. The TWI bit rate can be based either on the peripheral clock if the BRSRCCLK bit value is 0 or on a programmable clock source provided by the GCLK if the BRSRCCLK bit value is 1.

If BRSRCCLK = 1, the bit rate is independent of the processor/peripheral clock and thus processor/peripheral clock frequency can be changed without affecting the TWI transfer rate.

The GCLK frequency must be at least three times lower than the peripheral clock frequency.

45.9.3.4 Master Transmitter Mode

After the master initiates a START condition when writing into the Transmit Holding register FLEX_TWI_THR, it sends a 7-bit slave address, configured in the Master Mode Register (DADR in FLEX_TWI_MMR), to notify the slave device. The bit following the slave address indicates the transfer direction, 0 in this case (FLEX_TWI_MMR.MREAD = 0).

The TWI transfers require the slave to acknowledge each received byte. During the acknowledge clock pulse (ninth pulse), the master releases the data line (HIGH), enabling the slave to pull it down in order to generate the acknowledge. If the slave does not acknowledge the byte, then the Not Acknowledge flag (NACK) is set in the TWI Status Register (FLEX_TWI_SR) of the master and a STOP condition is sent. Alternatively, if the FLEX_TWI_MMR.NOAP bit is set, no stop condition will be sent and a START or STOP condition must be triggered manually through the FLEX_TWI_CR.START or FLEX_TWI_CR.STOP bit once the software is ready for the transmission of the condition. The NACK flag must be cleared by reading the TWI Status Register (FLEX_TWI_SR) before the next write into the TWI Transmit Holding Register (FLEX_TWI_THR). As with the other status bits, an interrupt can be generated if enabled in the interrupt enable Register (FLEX_TWI_IER). If the slave acknowledges the byte, the data written in FLEX_TWI_THR is then shifted in the internal shifter and transferred. When an acknowledge is detected, the TXRDY bit is set until a new write in FLEX_TWI_THR.

TXRDY is used as transmit ready for the DMA transmit channel.

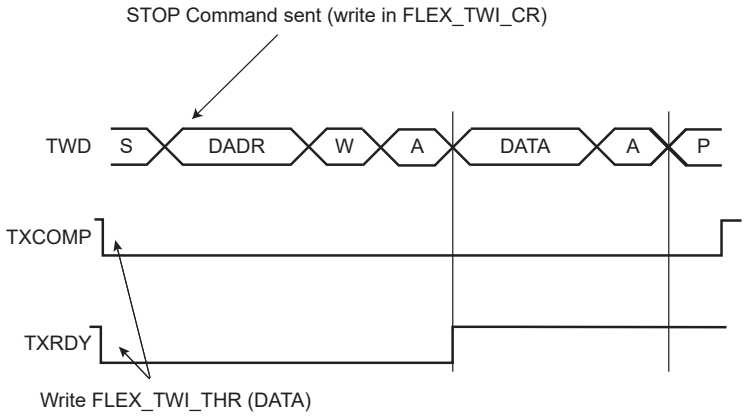
Note: To clear the TXRDY flag in Master mode, write the FLEX_TWI_CR.MSDIS bit to 1, then write the FLEX_TWI_CR.MSEN bit to 1.

While no new data is written in FLEX_TWI_THR, the serial clock line is tied low. When new data is written in FLEX_TWI_THR, the SCL is released and the data is sent. To generate a STOP event, the STOP command must be performed by writing in the STOP field of the TWI Control Register (FLEX_TWI_CR).

After a master write transfer, the Serial Clock line is stretched (tied low) while no new data is written in FLEX_TWI_THR or until a STOP command is performed.

See the following figures.

Figure 45-93. Master Write with One Data Byte



Downloaded from Arrow.com.

Figure 45-94. Master Write with Multiple Data Bytes

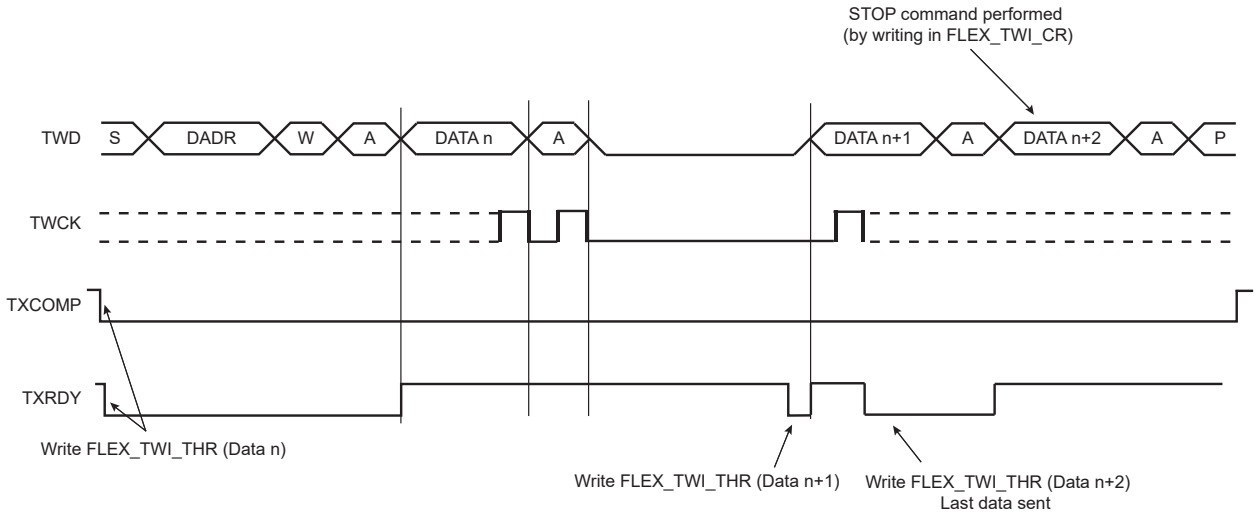
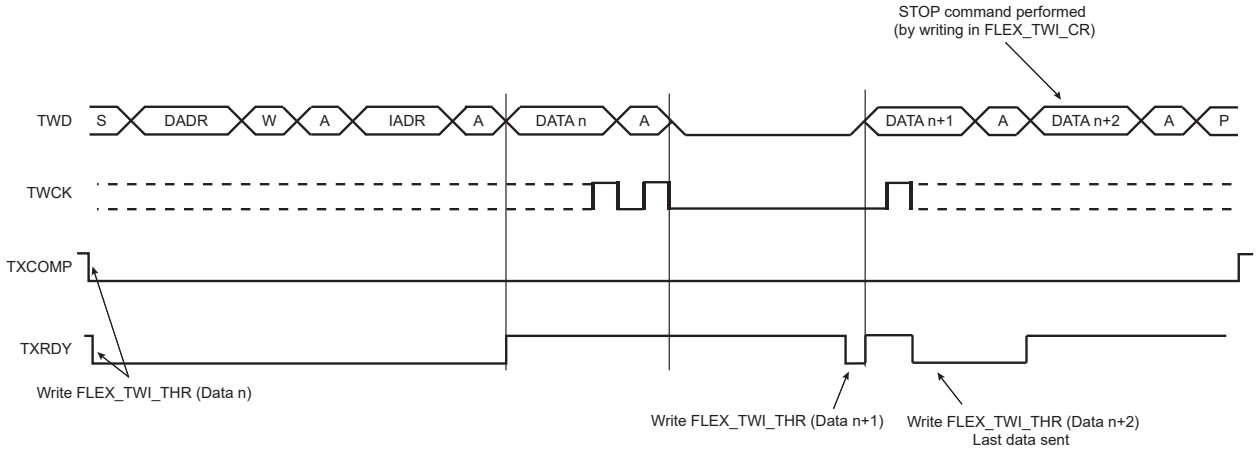


Figure 45-95. Master Write with One Byte Internal Address and Multiple Data Bytes



45.9.3.5 Master Receiver Mode

The read sequence begins by setting the START bit. After the start condition has been sent, the master sends a 7-bit slave address to notify the slave device. The bit following the slave address indicates the transfer direction, 1 in this case (FLEX_TWI_MMR.MREAD = 1). During the acknowledge clock pulse (9th pulse), the master releases the data line (HIGH), enabling the slave to pull it down in order to generate the acknowledge. The master polls the data line during this clock pulse and sets the FLEX_TWI_SR.NACK bit if the slave does not acknowledge the byte.

If an acknowledge is received, the master is then ready to receive data from the slave. After data has been received, the master sends an acknowledge condition to notify the slave that the data has been received except for the last data (see figure "Master Read with One Data Byte" below). When the FLEX_TWI_SR.RXRDY bit is set, a character has been received in the Receive Holding Register (FLEX_TWI_RHR). The RXRDY bit is reset when reading FLEX_TWI_RHR.

When a single data byte read is performed, with or without internal address (IADR), the START and STOP bits must be set at the same time. See figure "Master Read with One Data Byte" below. When a multiple data byte read is performed, with or without internal address (IADR), the STOP bit must be set after the next-to-last data received (same condition applies for START bit to generate a repeated start). See figure "Master Read with Multiple Data Bytes" below. For internal address usage, see section Internal Address.

If FLEX_TWI_RHR is full (RXRDY high) and the master is receiving data, the serial clock line will be tied low before receiving the last bit of the data and until FLEX_TWI_RHR is read. Once FLEX_TWI_RHR is read, the master will stop stretching the serial clock line and end the data reception. See figure "Master Read Clock Stretching with Multiple Data Bytes" below.

Downloaded from Arrow.com.

⚠WARNING When receiving multiple bytes in Master Read mode, if the next-to-last access is not read (the RXRDY flag remains high), the last access will not be completed until FLEX_TWI_RHR is read. The last access stops on the next-to-last bit (clock stretching). When FLEX_TWI_RHR is read there is only half a bit period to send the STOP bit (or START bit) command, else another read access might occur (spurious access).

A possible workaround is to set the STOP bit (or START bit) before reading FLEX_TWI_RHR on the next-to-last access (within IT handler).

Figure 45-96. Master Read with One Data Byte

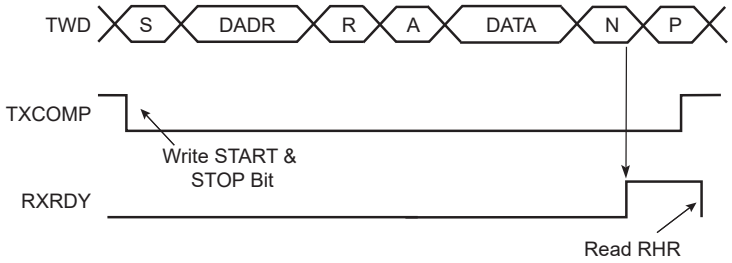


Figure 45-97. Master Read with Multiple Data Bytes

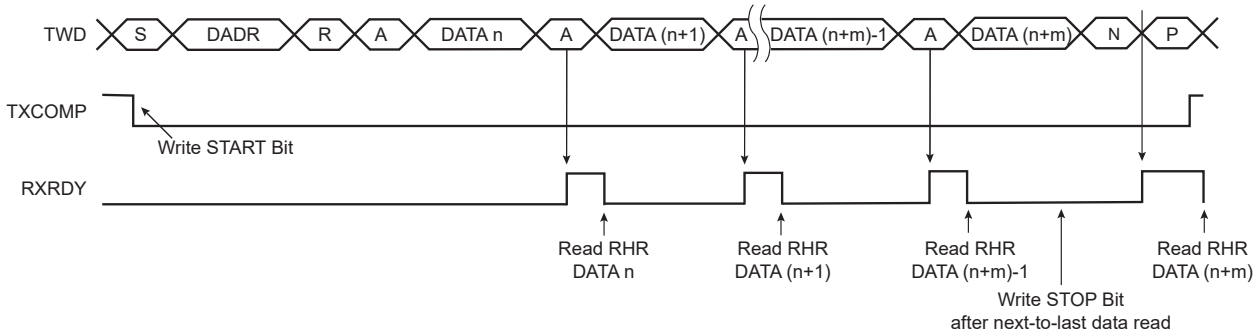
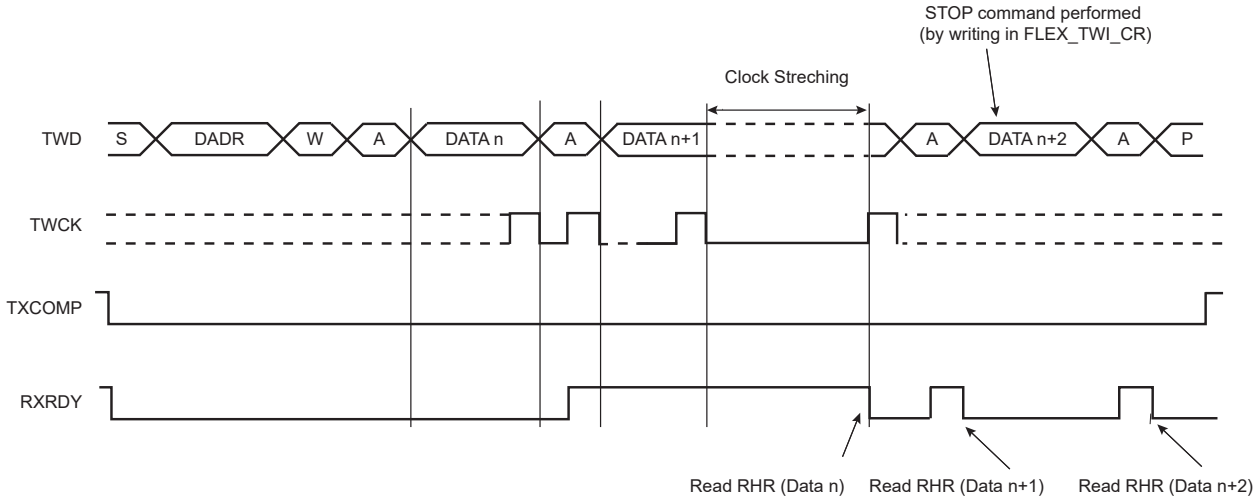


Figure 45-98. Master Read Clock Stretching with Multiple Data Bytes



RXRDY is used as receive ready trigger event for the DMA receive channel.

45.9.3.6 Internal Address

The TWI interface can perform transfers with 7-bit slave address devices and with 10-bit slave address devices.

45.9.3.6.1 7-bit Slave Addressing

When addressing 7-bit slave devices, the internal address bytes are used to perform random address (read or write) accesses to reach one or more data bytes, e.g., within a memory page location in a serial memory. When performing

Downloaded from Arrow.com.

read operations with an internal address, the TWI performs a write operation to set the internal address into the slave device, and then switch to Master Receiver mode. Note that the second start condition (after sending the IADR) is sometimes called “repeated start” (Sr) in I^2C fully-compatible devices. See figure Master Read with One, Two or Three Bytes Internal Address and One Data Byte.

See figures Master Write with One, Two or Three Bytes Internal Address and One Data Byte and Internal Address Usage for the master write operation with internal address.

The three internal address bytes are configurable through the Master Mode Register (FLEX_TWI_MMR).

If the slave device supports only a 7-bit address, i.e., no internal address, IADRSZ must be configured to 0.

The abbreviations listed below are used in the following figures:

S	Start
Sr	Repeated Start
P	Stop
W	Write
R	Read
A	Acknowledge
N	Not Acknowledge
DADR	Device Address
IADR	Internal Address

Figure 45-99. Master Write with One, Two or Three Bytes Internal Address and One Data Byte

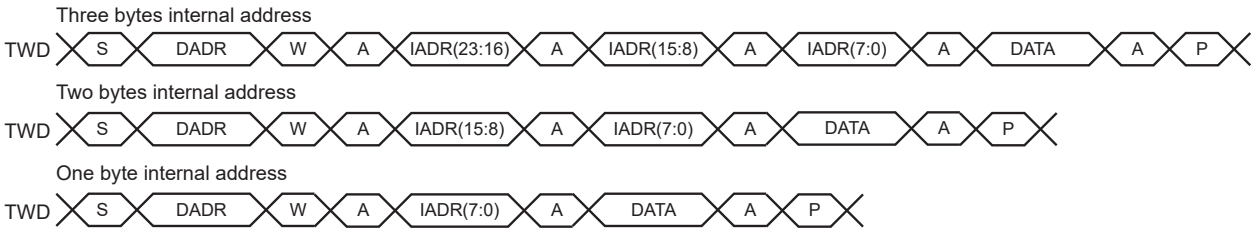
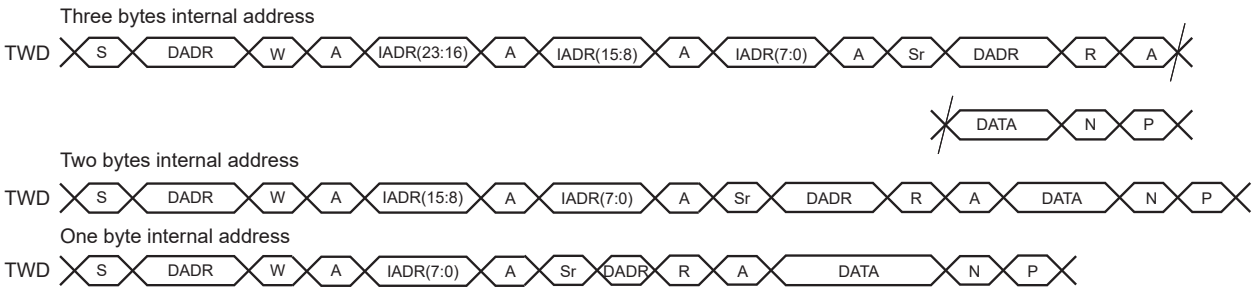


Figure 45-100. Master Read with One, Two or Three Bytes Internal Address and One Data Byte



45.9.3.6.2 10-bit Slave Addressing

For a slave address higher than seven bits, the user must configure the address size (IADRSZ) and set the other slave address bits in the Internal Address Register (FLEX_TWI_IADR). The two remaining internal address bytes, IADR[15:8] and IADR[23:16], can be used the same way as in 7-bit slave addressing.

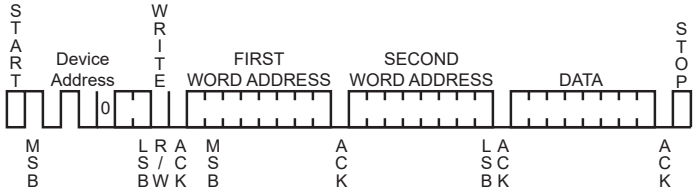
Example: Address a 10-bit device (10-bit device address is b1 b2 b3 b4 b5 b6 b7 b8 b9 b10)

1. Program IADRSZ = 1
2. Program DADR with 1 1 1 1 0 b1 b2 (b1 is the MSB of the 10-bit address, b2, etc.)
3. Program FLEX_TWI_IADR with b3 b4 b5 b6 b7 b8 b9 b10 (b10 is the LSB of the 10-bit address)

Downloaded from Arrow.com.

The following figure shows a byte write to a TWI EEPROM. This demonstrates the use of internal addresses to access the device.

Figure 45-101. Internal Address Usage



45.9.3.7 Repeated Start

In addition to Internal Address mode, repeated start (Sr) can be generated manually by writing the START bit at the end of a transfer instead of the STOP bit. In such case the parameters of the next transfer (direction, SADR, etc.) will need to be set before writing the START bit at the end of the previous transfer.

See section Read/Write Flowcharts.

45.9.3.8 Bus Clear Command

The TWI interface can perform a Bus Clear Command:

1. Configure the Master mode (DADR, CKDIV, etc).
2. Start the transfer by setting the FLEX_TWI_CR.CLEAR bit.

Note: If an alternative command is used (ACMEN bit = 1), the DATAL field must be cleared.

45.9.3.9 SMBus Mode

SMBus mode is enabled when the FLEX_TWI_CR.SMBEN bit is written to one. SMBus mode operation is similar to I²C operation with the following exceptions:

1. Only 7-bit addressing can be used.
2. The SMBus standard describes a set of timeout values to ensure progress and throughput on the bus. These timeout values must be programmed into FLEX_TWI_SMBTR.
3. Transmissions can optionally include a CRC byte, called Packet Error Check (PEC).
4. A set of addresses has been reserved for protocol handling, such as alert response address (ARA) and host header (HH) address. Address matching on these addresses can be enabled by configuring FLEX_TWI_CR appropriately.

45.9.3.9.1 Packet Error Checking

Each SMBus transfer can optionally end with a CRC byte, called the PEC byte. Writing the FLEX_TWI_CR.PECEN bit to one enables automatic PEC handling in the current transfer. Transfers with and without PEC can freely be intermixed in the same system, since some slaves may not support PEC. The PEC LFSR is always updated on every bit transmitted or received, so that PEC handling on combined transfers will be correct.

In Master Transmitter mode, the master calculates a PEC value and transmits it to the slave after all data bytes have been transmitted. Upon reception of this PEC byte, the slave will compare it to the PEC value it has computed itself. If the values match, the data was received correctly, and the slave will return an ACK to the master. If the PEC values differ, data was corrupted, and the slave will return a NACK value. Some slaves may not be able to check the received PEC in time to return a NACK if an error occurred. In this case, the slave should always return an ACK after the PEC byte, and some other mechanism must be implemented to verify that the transmission was received correctly.

In Master Receiver mode, the slave calculates a PEC value and transmits it to the master after all data bytes have been transmitted. Upon reception of this PEC byte, the master will compare it to the PEC value it has computed itself. If the values match, the data was received correctly. If the PEC values differ, data was corrupted, and the FLEX_TWI_SR.PECERR bit is set. In Master Receiver mode, the PEC byte is always followed by a NACK transmitted by the master, since it is the last byte in the transfer.

In combined transfers, the PECRQ bit should only be set in the last of the combined transfers. If Alternative Command mode is enabled, only the NPEC bit should be set.

Consider the following transfer:

Downloaded from Arrow.com.

S, ADR+W, COMMAND_BYTE, ACK, SR, ADR+R, DATA_BYTE, ACK, PEC_BYTE, NACK, P

See section Read/Write Flowcharts for detailed flowcharts.

45.9.3.9.2 Timeouts

The FLEX_TWI_SMBTR.TLOWS/TLOWM fields configure the SMBus timeout values. If a timeout occurs, the master transmits a STOP condition and leaves the bus. Furthermore, the FLEX_TWI_SR.TOUT bit is set.

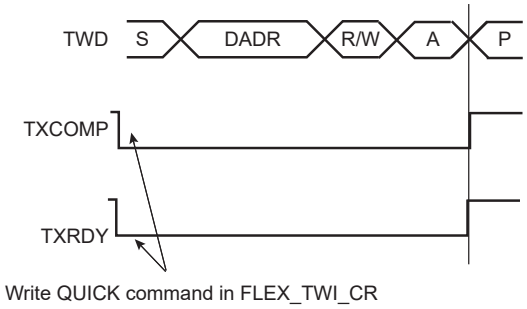
45.9.3.10 SMBus Quick Command (Master Mode Only)

The TWI interface can perform a quick command:

1. Configure the Master mode (DADR, CKDIV, etc).
2. Write the FLEX_TWI_MMR.MREAD bit at the value of the one-bit command to be sent.
3. Start the transfer by setting the FLEX_TWI_CR.QUICK bit.

Note: If an alternative command is used (ACMEN bit = 1), the DATAL field must be cleared.

Figure 45-102. SMBus Quick Command



45.9.3.11 Alternative Command

Another way to configure the transfer is to enable the Alternative Command mode with the ACMEN bit of the TWI Control Register.

In this mode, the transfer is configured through the TWI Alternative Command Register. It is possible to define a simple read or write transfer or a combined transfer with a repeated start.

In order to set a simple transfer, the DATAL field and the DIR field of the TWI Alternative Command Register must be filled accordingly and the NDATAL field must be cleared. To begin the transfer, either set the START bit in the TWI Control Register in case of a read transfer, or write the TWI Transmit Holding Register in case of a write transfer.

For a combined transfer linked by a repeated start, the NDATAL field must be filled with the length of the second transfer and NDIR with the corresponding direction.

The PEC and NPEC bits are used to set a PEC field. In the case of a single transfer with PEC, the PEC bit must be set. In the case of a combined transfer, the NPEC bit must be set.

Note: If the Alternative Command mode is used, the TWIHS_MMR.IADRSZ field must be set to 0.

See Read/Write Flowcharts for detailed flowcharts.

45.9.3.12 Handling Errors in Alternative Command

In case of NACK generated by a slave device or SMBus timeout error, the TWI stops immediately the frame, but the DMA transfer may still be active. To prevent a new frame to be restarted with the remaining DMA data (transmit), the TWI prevents any start of frame until the FLEX_TWI_SR.LOCK flag is cleared.

The FLEX_TWI_SR.LOCK bit indicates the state of the TWI (locked or not locked).

When the TWI is locked, no transfer can begin until the LOCK is cleared using the FLEX_TWI_CR.LOCKCLR bit and until the error flags are cleared reading FLEX_TWI_SR.

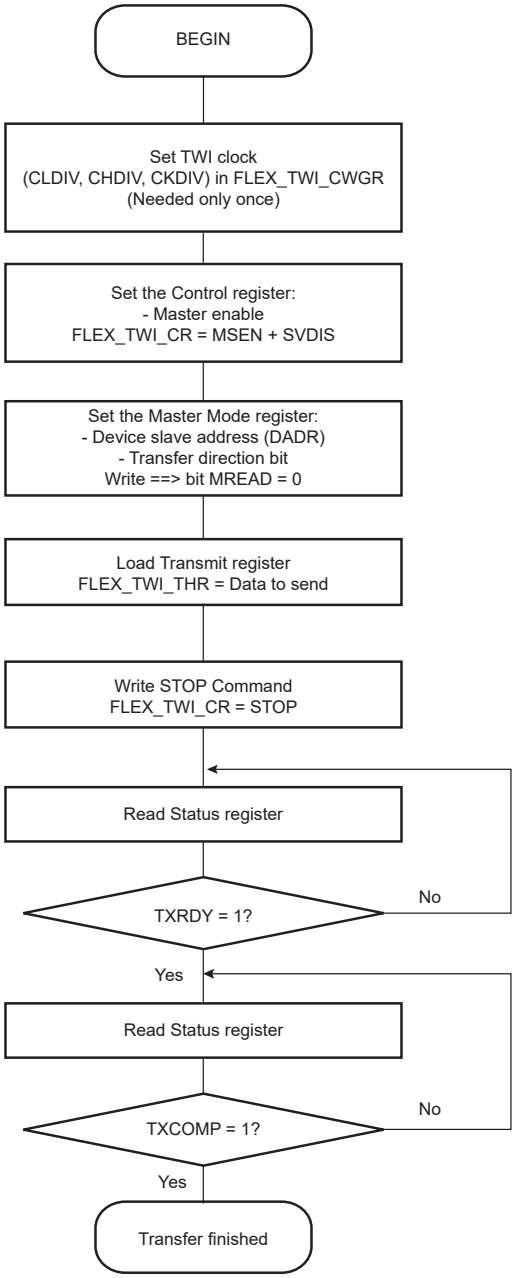
In case of error, FLEX_TWI_THR may have been loaded with a new data. The FLEX_TWI_CR.THRCLR bit can be used to flush FLEX_TWI_THR. If the THRCLR bit is set, the TXRDY and TXCOMP flags are set.

Downloaded from Arrow.com.

45.9.3.13 Read/Write Flowcharts

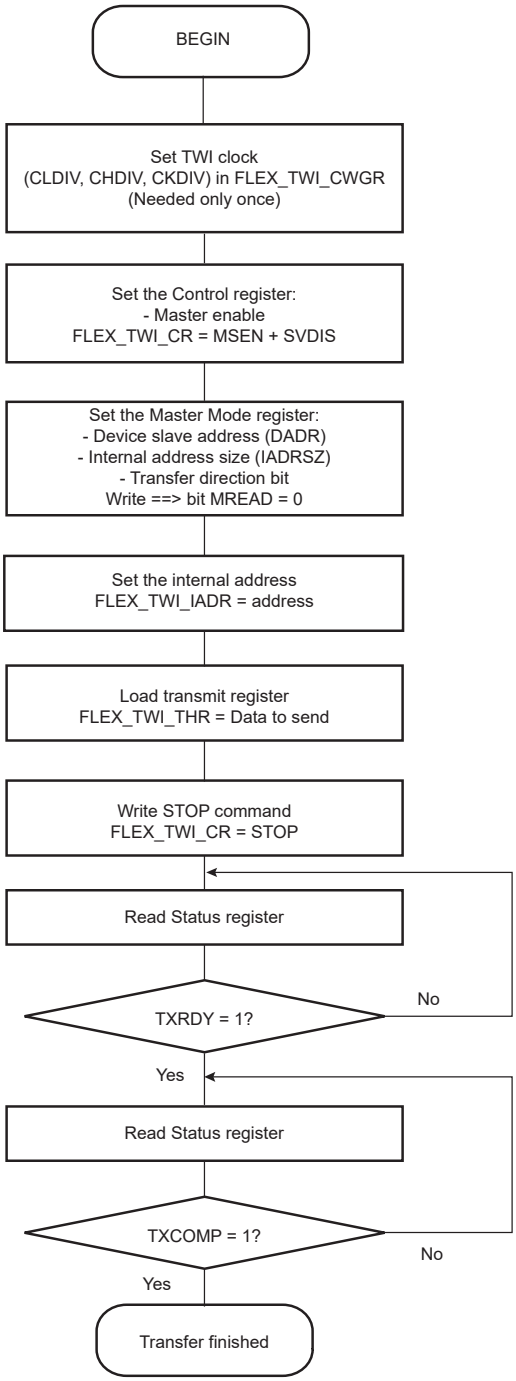
The flowcharts shown in this section provide examples for read and write operations. A polling or interrupt method can be used to check the status bits. The interrupt method requires that the Interrupt Enable Register (FLEX_TWI_IER) be configured first.

Figure 45-103. TWI Write Operation with Single Data Byte without Internal Address



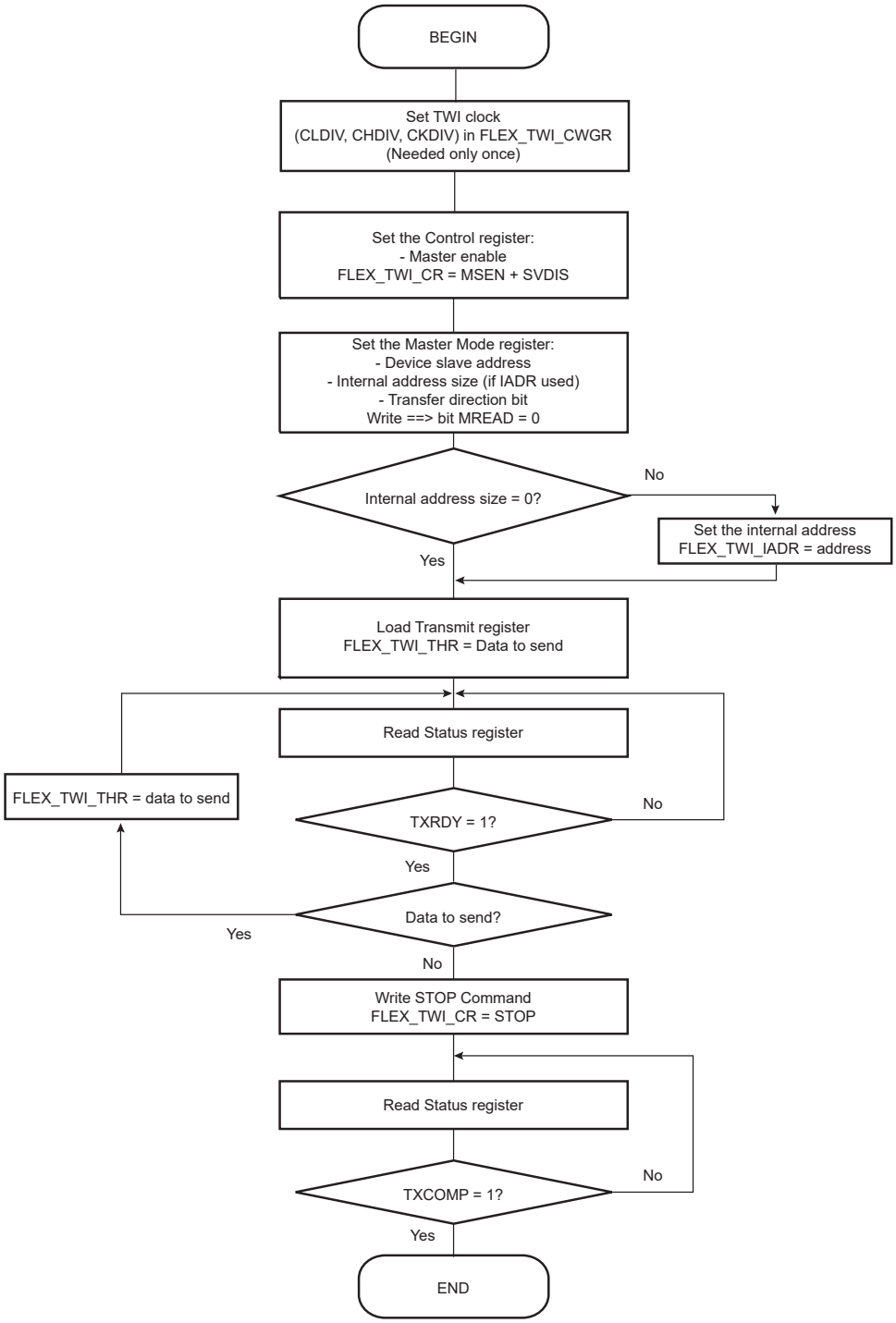
Downloaded from Arrow.com.

Figure 45-104. TWI Write Operation with Single Data Byte and Internal Address



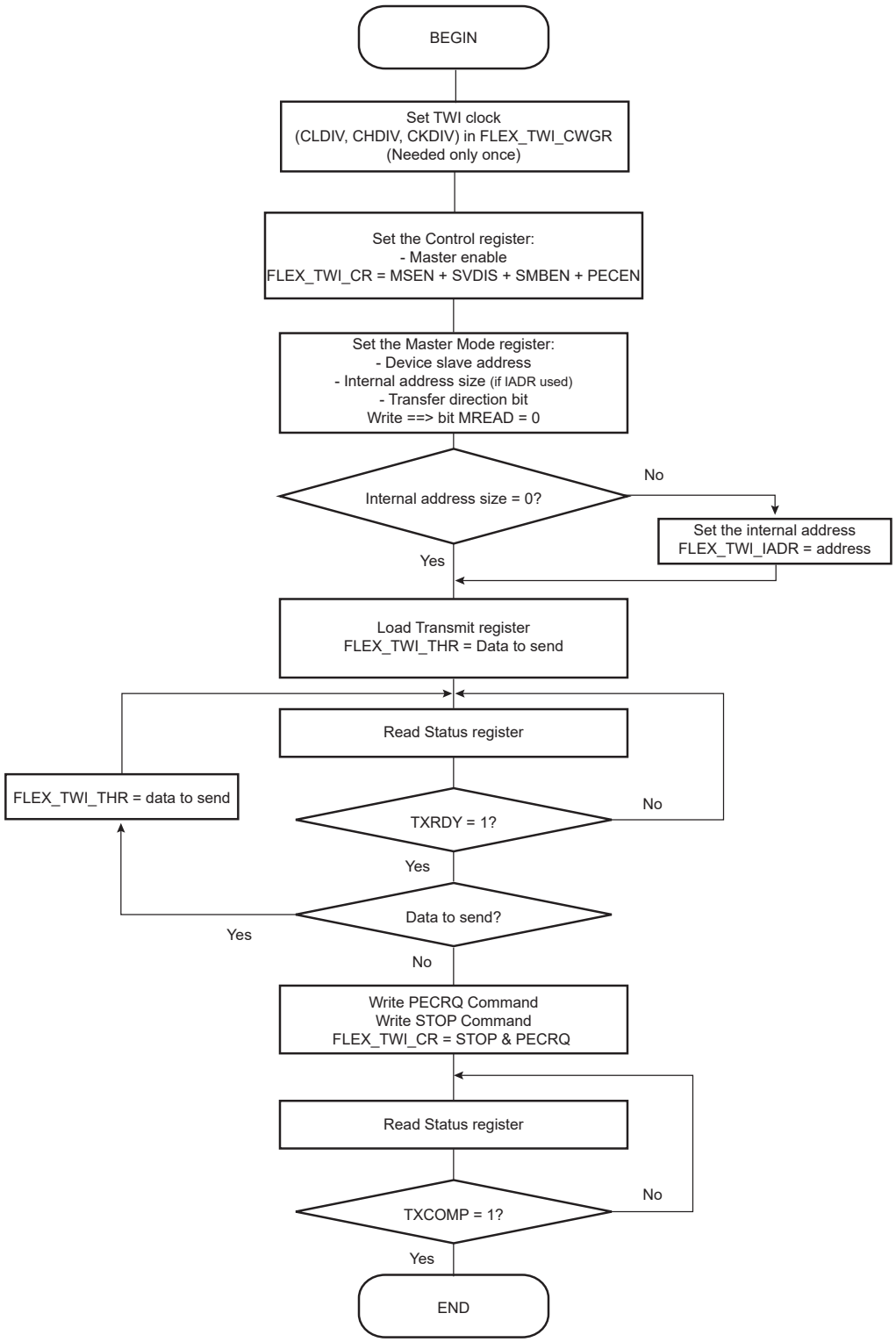
Downloaded from Arrow.com.

Figure 45-105. TWI Write Operation with Multiple Data Bytes with or without Internal Address



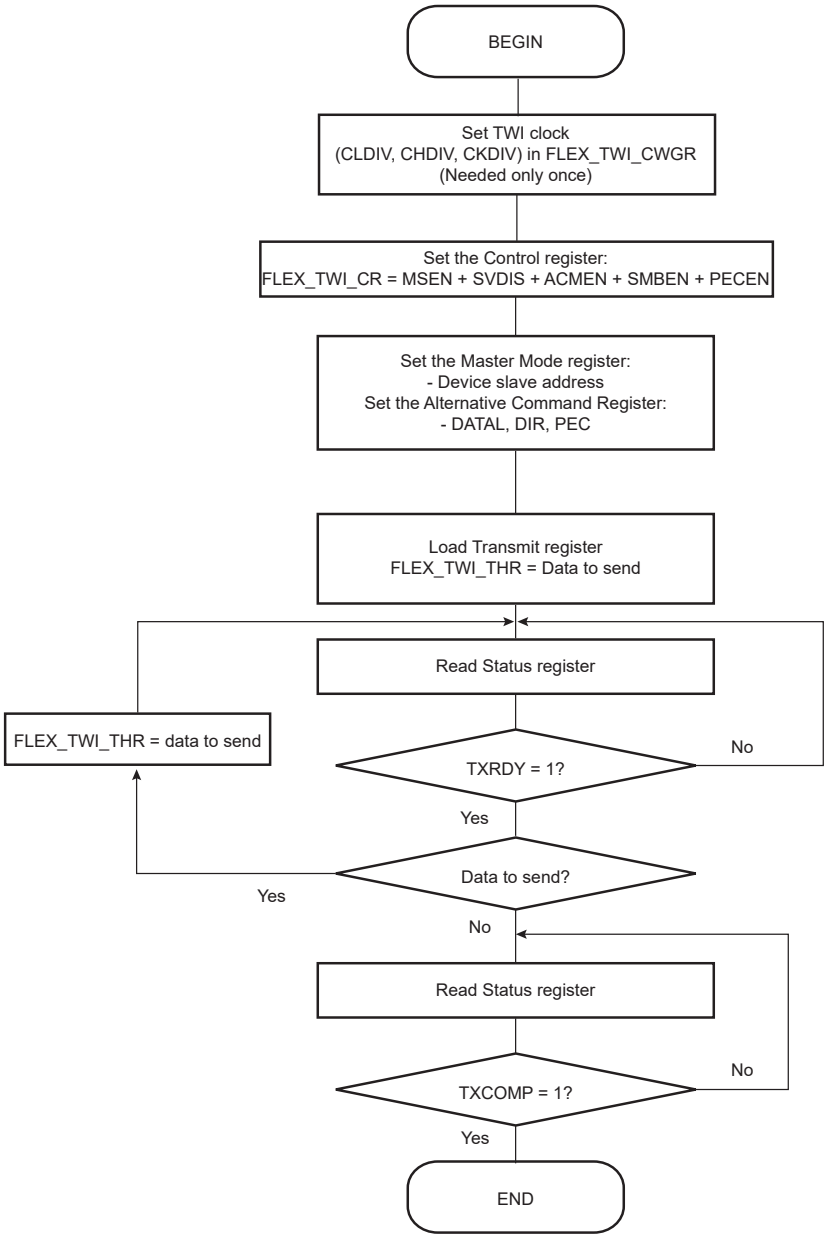
Downloaded from Arrow.com.

Figure 45-106. SMBus Write Operation with Multiple Data Bytes with or without Internal Address and PEC Sending



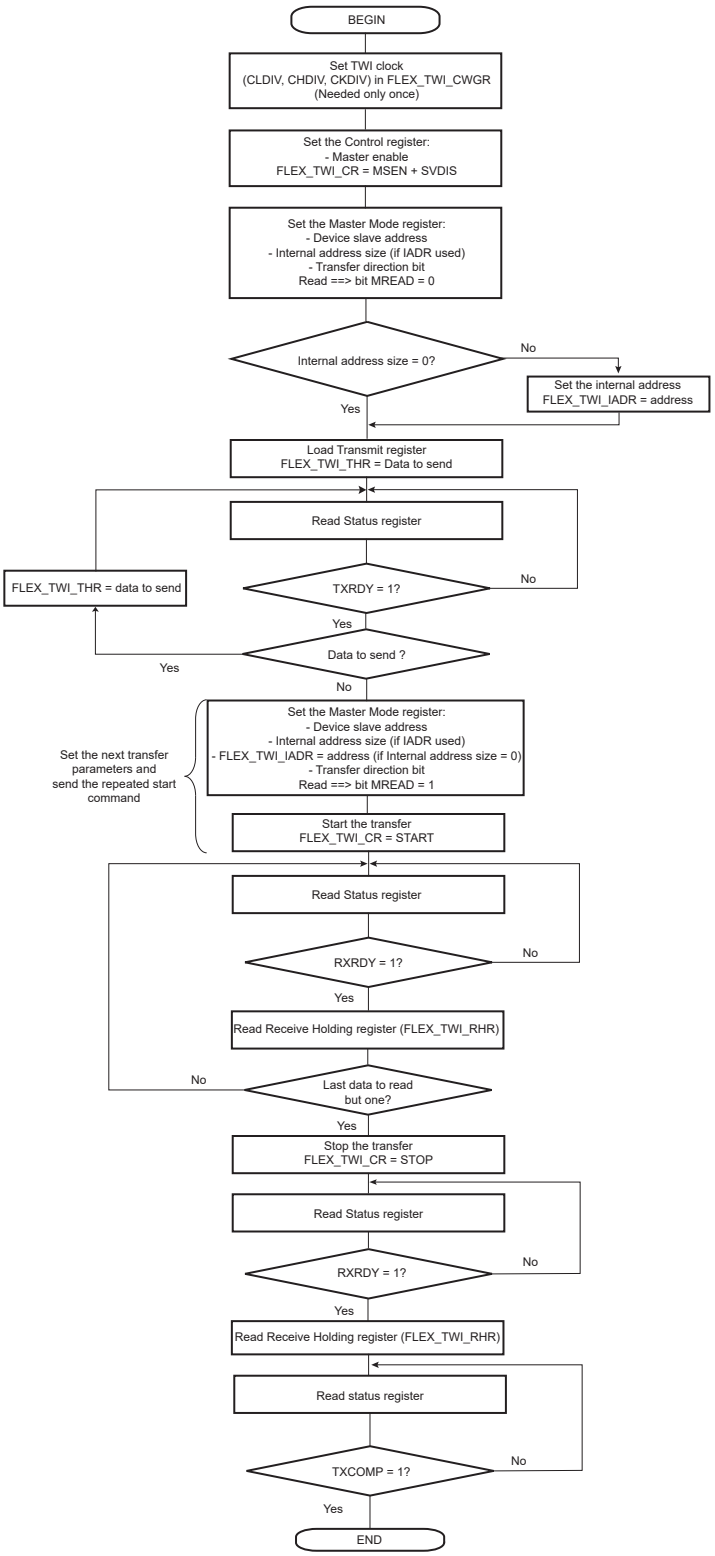
Downloaded from Arrow.com.

Figure 45-107. SMBus Write Operation with Multiple Data Bytes with PEC and Alternative Command Mode



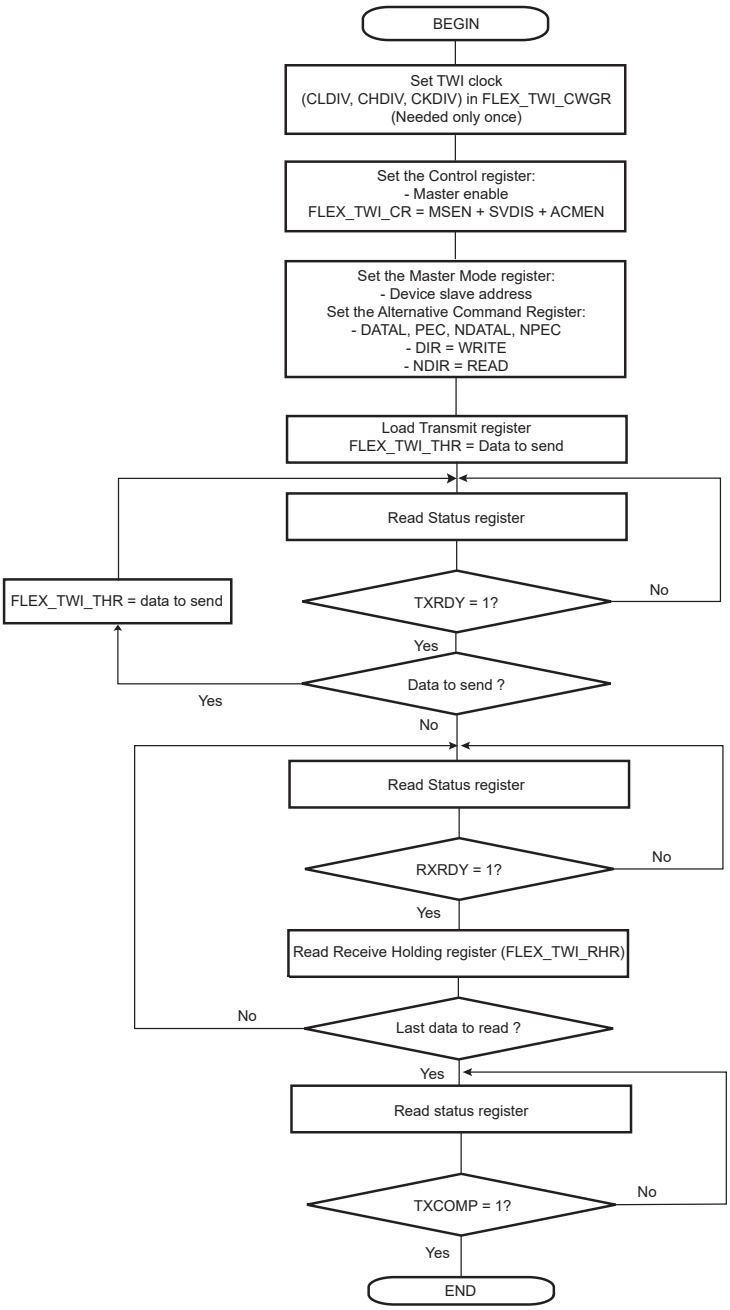
Downloaded from Arrow.com.

Figure 45-108. TWI Write Operation with Multiple Data Bytes and Read Operation with Multiple Data Bytes (Sr)



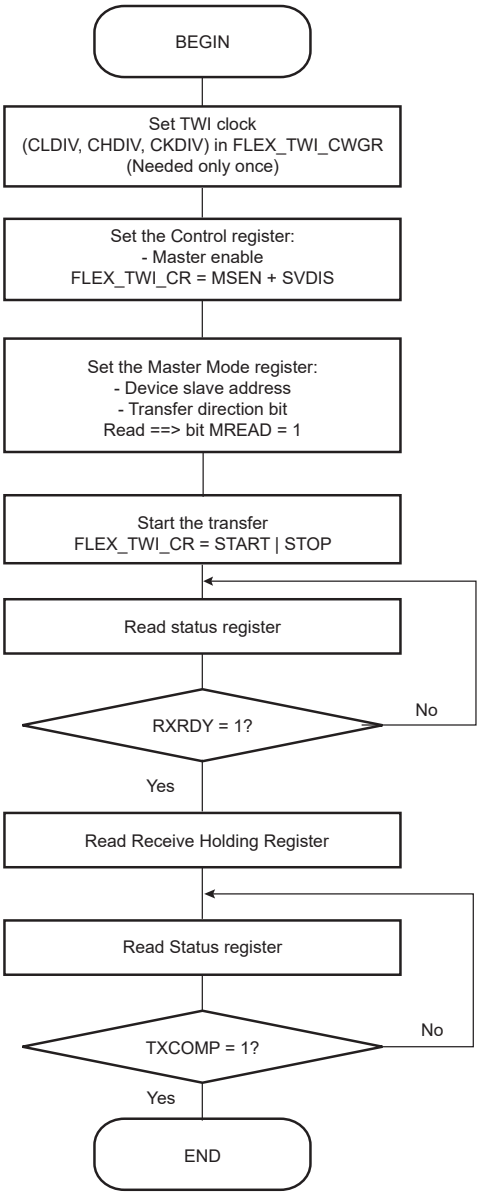
Downloaded from Arrow.com.

Figure 45-109. TWI Write Operation with Multiple Data Bytes + Read Operation and Alternative Command Mode + PEC



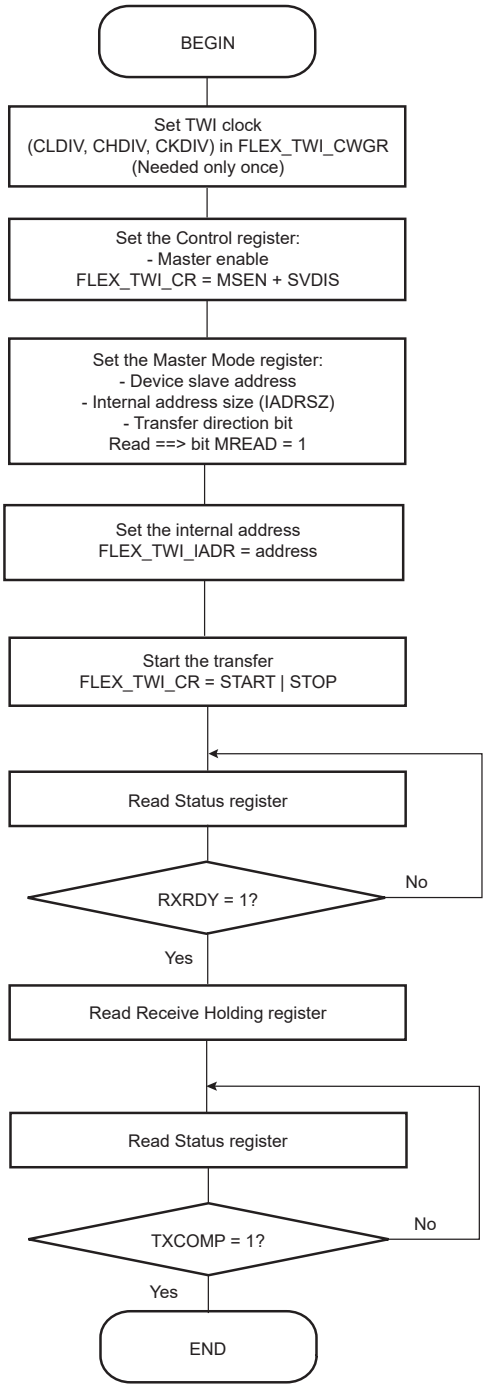
Downloaded from Arrow.com.

Figure 45-110. TWI Read Operation with Single Data Byte without Internal Address



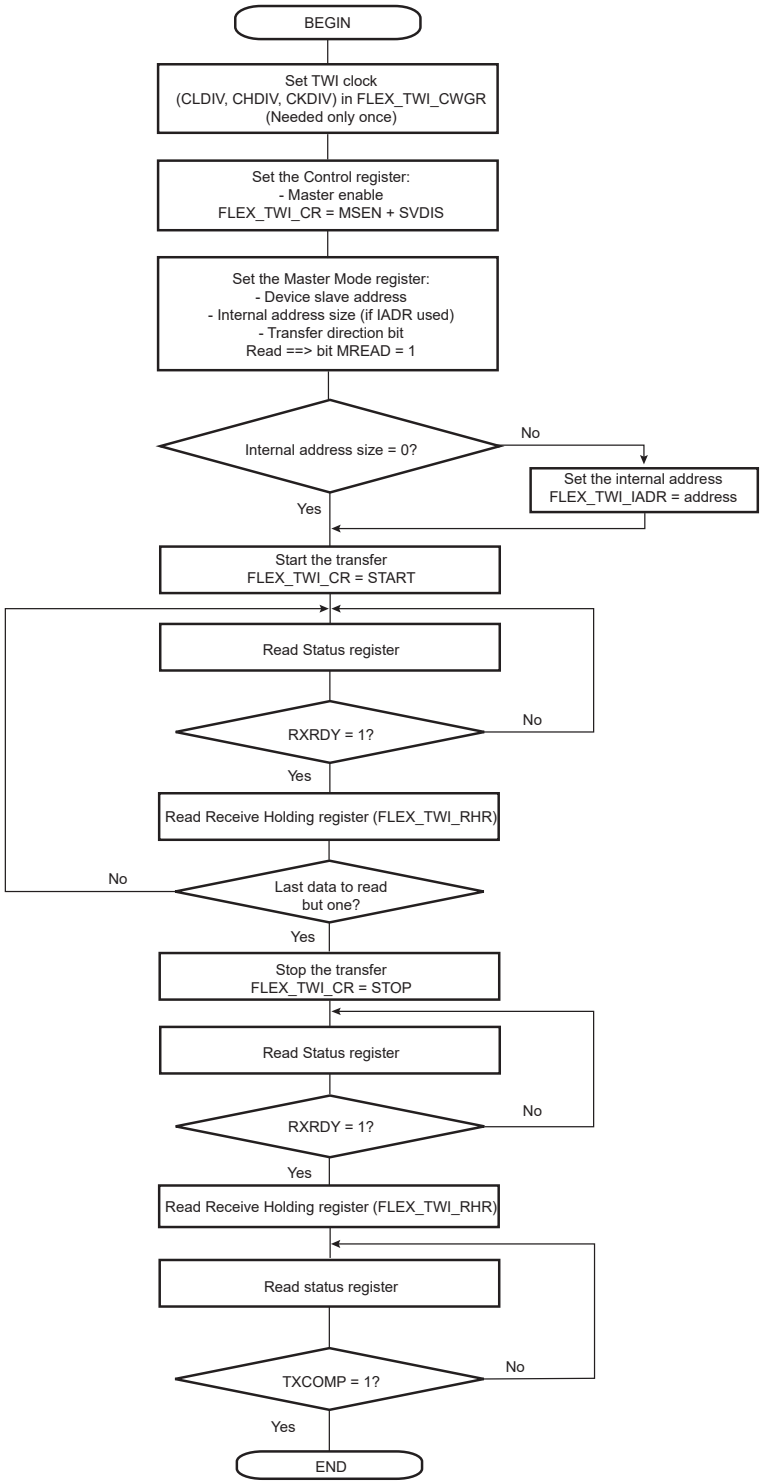
Downloaded from Arrow.com.

Figure 45-111. TWI Read Operation with Single Data Byte and Internal Address



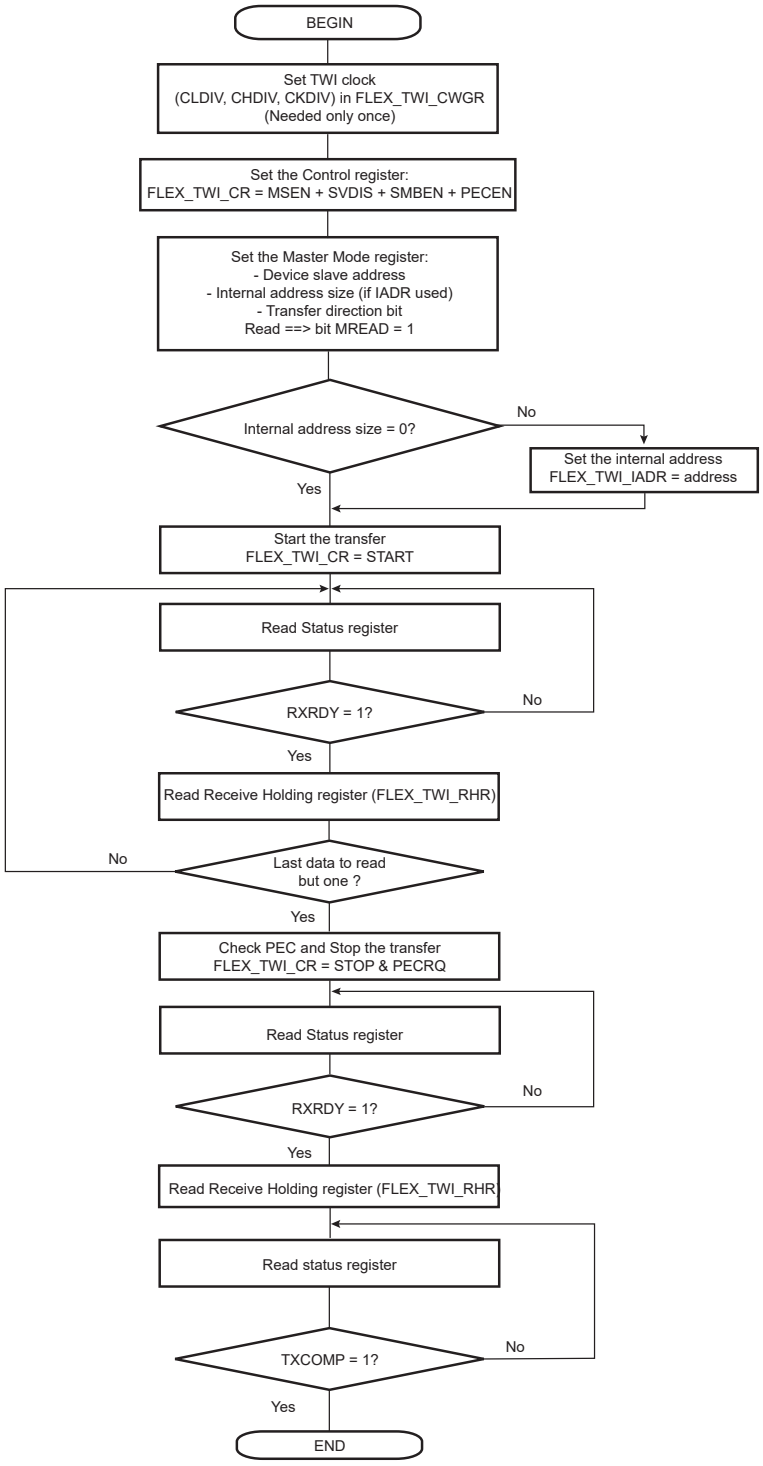
Downloaded from Arrow.com.

Figure 45-112. TWI Read Operation with Multiple Data Bytes with or without Internal Address



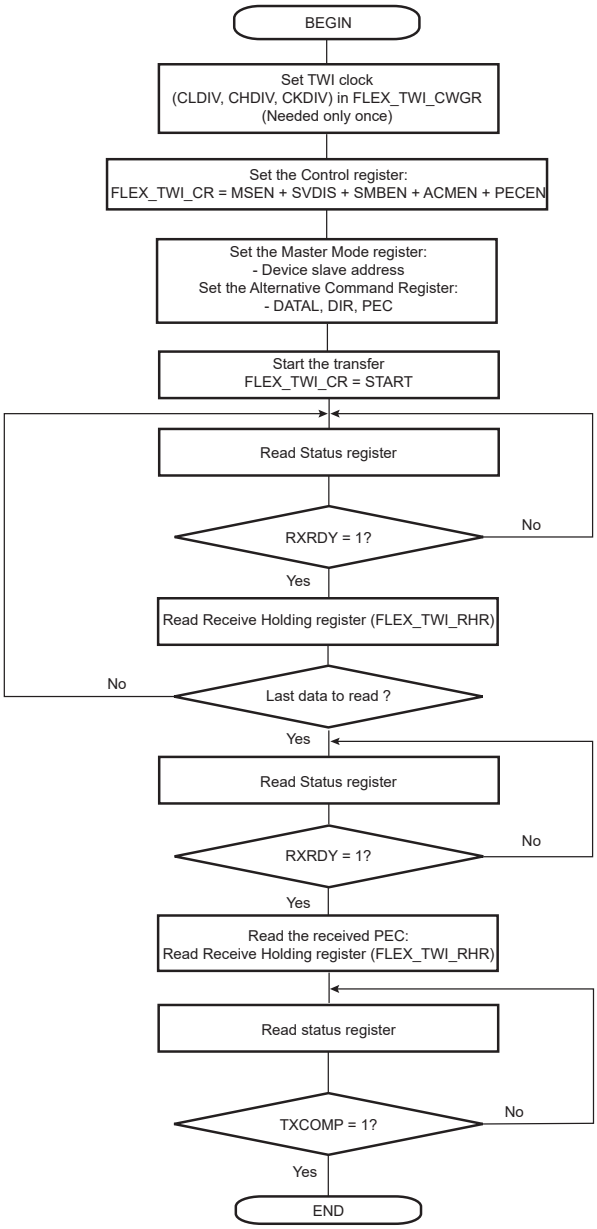
Downloaded from Arrow.com.

Figure 45-113. TWI Read Operation with Multiple Data Bytes with or without Internal Address with PEC



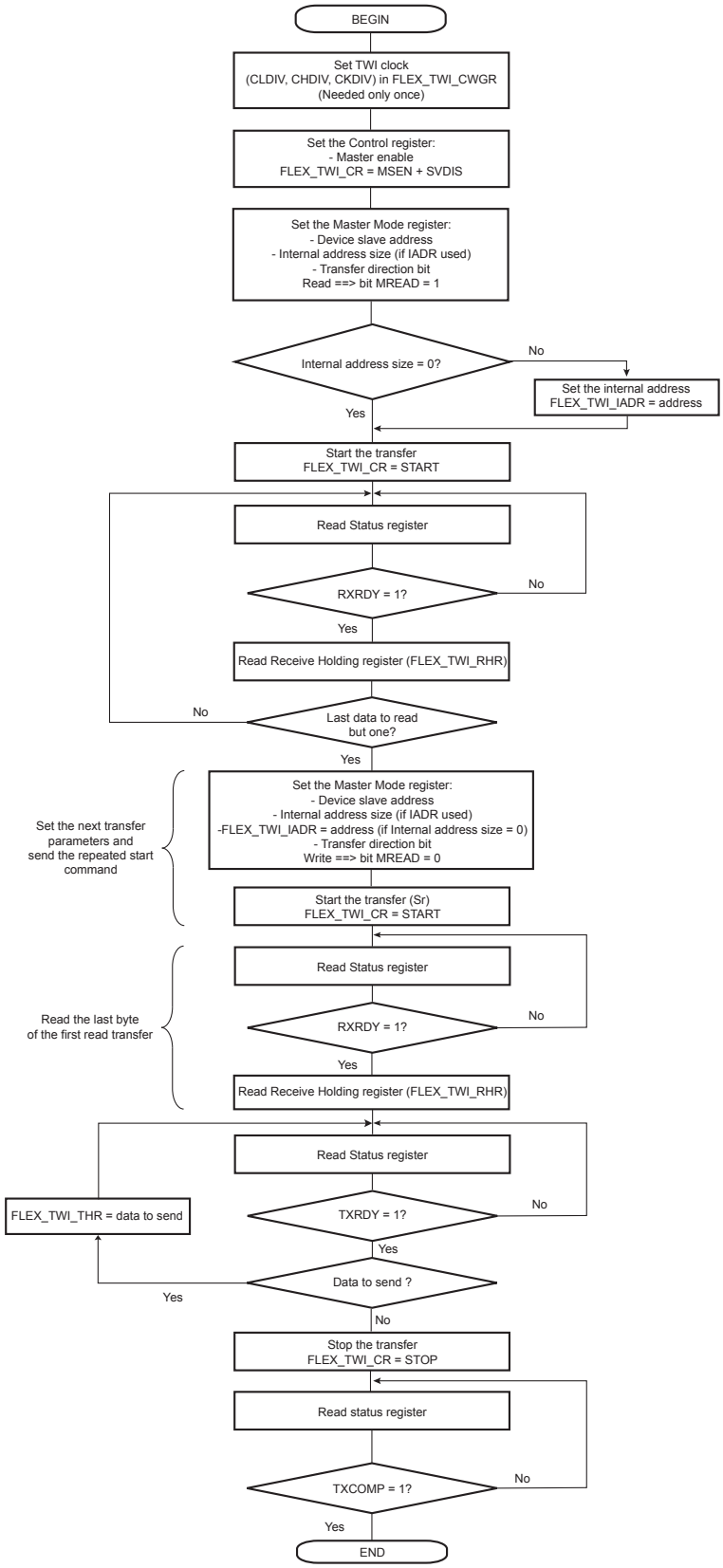
Downloaded from Arrow.com.

Figure 45-114. TWI Read Operation with Multiple Data Bytes with Alternative Command Mode with PEC



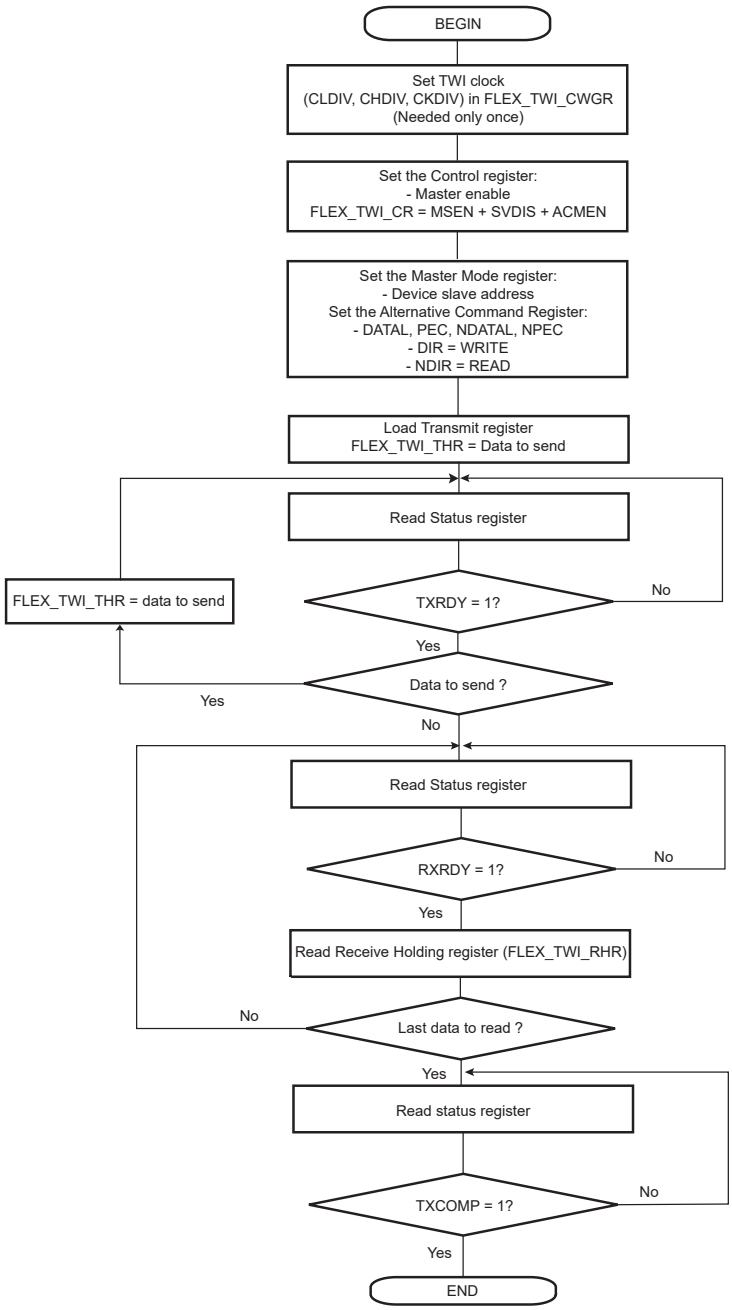
Downloaded from Arrow.com.

Figure 45-115. TWI Read Operation with Multiple Data Bytes + Write Operation with Multiple Data Bytes (Sr)



Downloaded from Arrow.com.

Figure 45-116. TWI Read Operation with Multiple Data Bytes + Write with Alternative Command Mode with PEC



45.9.4 Multi-Master Mode

45.9.4.1 Definition

In Multi-Master mode, more than one master may handle the bus at the same time without data corruption by using arbitration.

Arbitration starts as soon as two or more masters place information on the bus at the same time, and stops (arbitration is lost) for the master that intends to send a logical one while the other master sends a logical zero.

As soon as arbitration is lost by a master, it stops sending data and listens to the bus in order to detect a STOP. When the STOP is detected, the master that has lost arbitration may put its data on the bus by respecting arbitration.

Downloaded from Arrow.com.

Arbitration is illustrated in figure "Arbitration Cases" below.

45.9.4.2 Different Multi-Master Modes

Two Multi-Master modes may be distinguished:

- TWI as Master Only—TWI is considered as a master only and will never be addressed.
- TWI as Master or Slave—TWI may be either a master or a slave and may be addressed.

Note: Arbitration in supported in both Multi-Master modes.

45.9.4.2.1 TWI as Master Only

In this mode, the TWI is considered as a master only (MSEN is always at one) and must be driven like a master with the ARBLST (ARBitration Lost) flag in addition.

If arbitration is lost (ARBLST = 1), the user must reinitiate the data transfer.

If the user starts a transfer (ex.: DADR + START + W + Write in THR) and if the bus is busy, the TWI automatically waits for a STOP condition on the bus to initiate the transfer (see figure "User Sends Data While the Bus is Busy" below).

Note: The state of the bus (busy or free) is not indicated in the user interface.

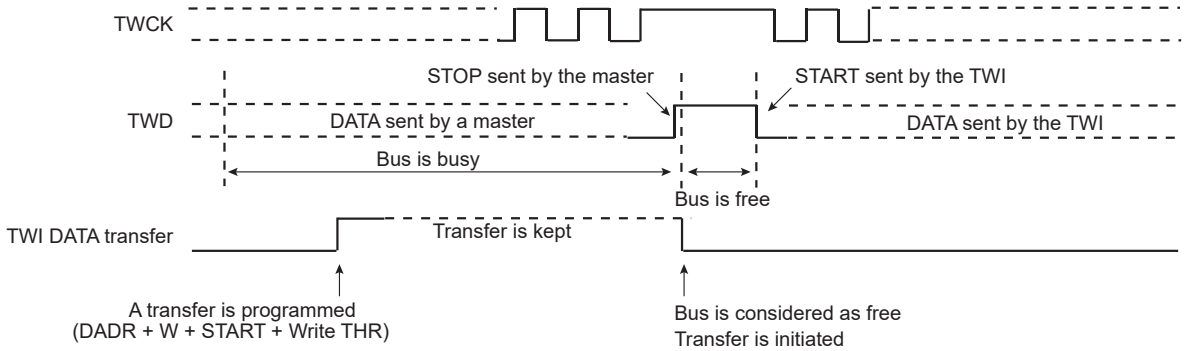
45.9.4.2.2 TWI as Master or Slave

The automatic reversal from master to slave is not supported in case of a lost arbitration.

Then, in the case where TWI may be either a master or a slave, the user must manage the pseudo Multi-Master mode described in the steps below:

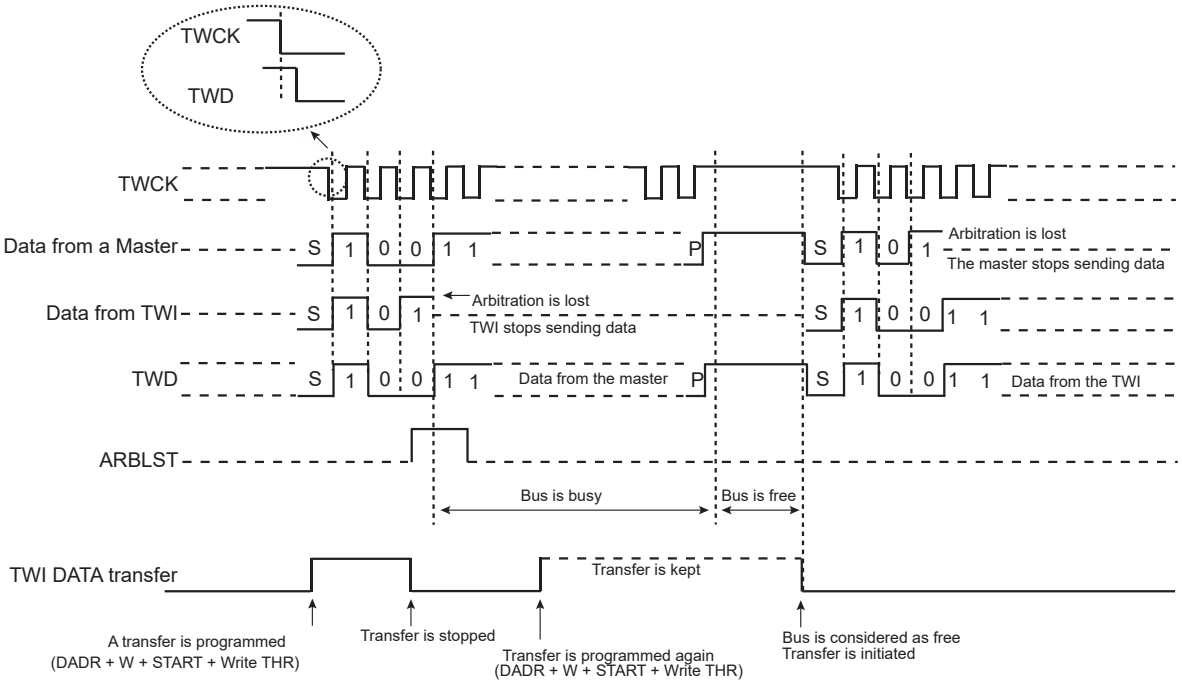
1. Program the TWI in Slave mode (SADR + MSDIS + SVEN) and perform a slave access (if TWI is addressed).
2. If the TWI has to be set in Master mode, wait until TXCOMP flag is at 1.
3. Program the Master mode (DADR + SVDIS + MSEN) and start the transfer (ex: START + Write in THR).
4. As soon as the Master mode is enabled, the TWI scans the bus in order to detect if it is busy or free. When the bus is considered as free, the TWI initiates the transfer.
5. As soon as the transfer is initiated and until a STOP condition is sent, the arbitration becomes relevant and the user must monitor the ARBLST flag.
6. If the arbitration is lost (ARBLST is = 1), the user must program the TWI in Slave mode in case the master that won the arbitration needs to access the TWI.
7. If the TWI has to be set in Slave mode, wait until TXCOMP flag is at 1 and then program the Slave mode.
   Note: In case the arbitration is lost and the TWI is addressed, the TWI will not acknowledge even if it is programmed in Slave mode as soon as ARBLST = 1. Then the master must repeat SADR.

Figure 45-117. User Sends Data While the Bus is Busy



Downloaded from Arrow.com.

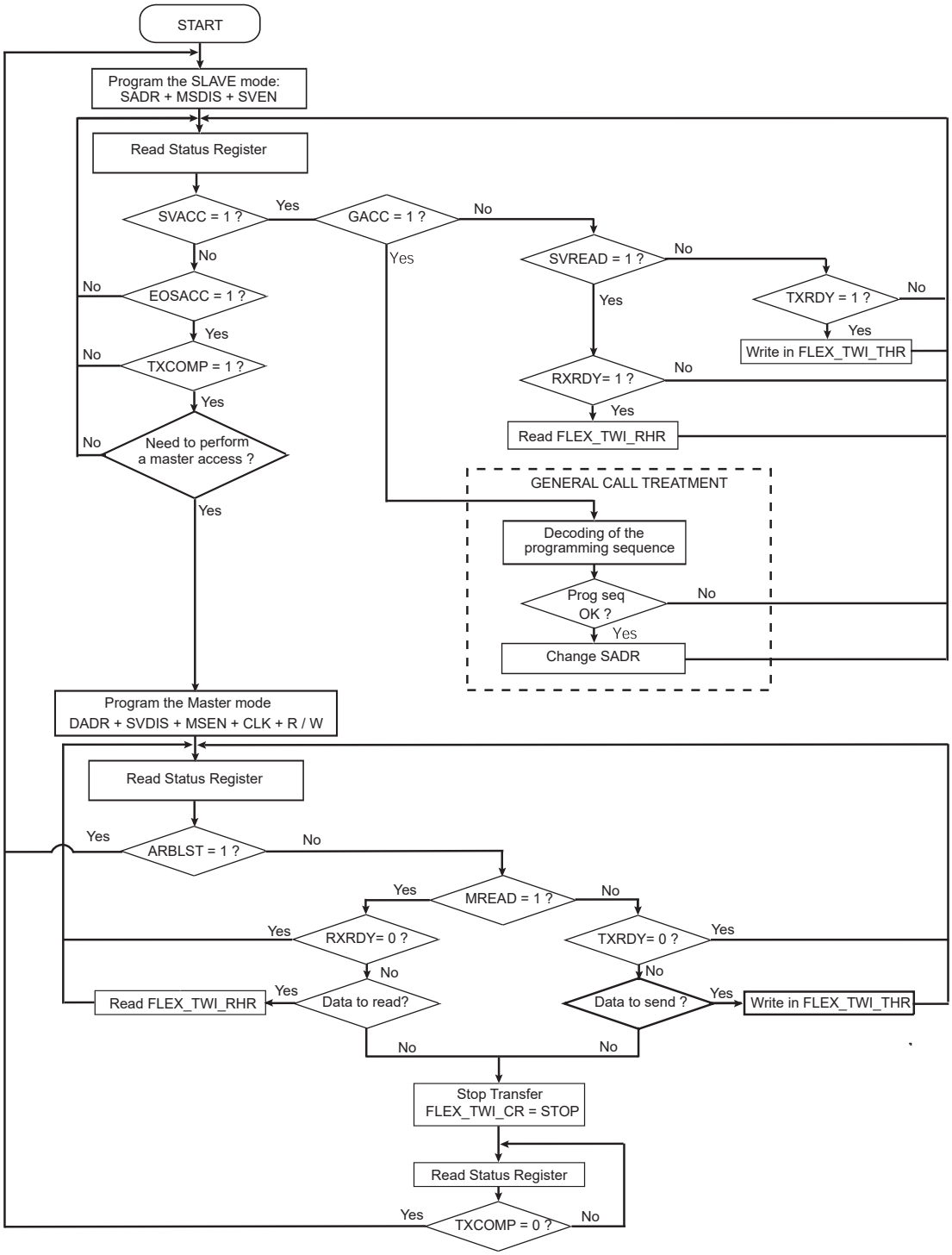
Figure 45-118. Arbitration Cases



The flowchart shown in the following figure gives an example of read and write operations in Multi-Master mode.

Downloaded from Arrow.com.

Figure 45-119. Multi-Master Mode



45.9.5 Slave Mode

45.9.5.1 Definition

Slave mode is defined as a mode where the device receives the clock and the address from another device called the master.

Downloaded from Arrow.com.

In this mode, the device never initiates and never completes the transmission (START, REPEATED_START and STOP conditions are always provided by the master).

45.9.5.2 Programming Slave Mode

The following fields must be programmed before entering Slave mode:

1. FLEX_TWI_SMR.SADR: The slave device address is used in order to be accessed by master devices in Read or Write mode.
2. (Optional) FLEX_TWI_SMR.MASK can be set to mask some SADR address bits and thus allow multiple address matching.
3. FLEX_TWI_CR.MSDIS: Disables the Master mode.
4. FLEX_TWI_CR.SVEN: Enables the Slave mode.

As the device receives the clock, values written in FLEX_TWI_CWGR are not processed.

45.9.5.3 Receiving Data

After a START or repeated START condition is detected, and if the address sent by the master matches the slave address programmed in the SADR (Slave Address) field, the SVACC (Slave Access) flag is set and SVREAD (Slave Read) indicates the direction of the transfer.

SVACC remains high until a STOP condition or a repeated START is detected. When such a condition is detected, EOSACC (End Of Slave Access) flag is set.

45.9.5.3.1 Read Sequence

In the case of a read sequence (SVREAD is high), the TWI transfers data written in FLEX_TWI_THR (TWI Transmit Holding Register) until a STOP condition or a REPEATED_START + an address different from SADR is detected. Note that at the end of the read sequence TXCOMP (Transmission Complete) flag is set and SVACC is reset.

As soon as data is written in FLEX_TWI_THR, the TXRDY (Transmit Holding Register Ready) flag is reset, and it is set when the internal shifter is empty and the sent data acknowledged or not. If the data is not acknowledged, the NACK flag is set.

Note that a STOP or a repeated START always follows a NACK.

See figure "Read Access Ordered by a Master" below.

Note: To clear the TXRDY flag in Slave mode, write the FLEX_TWI_CR.SVDIS bit to 1, then write the FLEX_TWI_CR.SVEN bit to 1.

45.9.5.3.2 Write Sequence

In the case of a write sequence (SVREAD is low), the RXRDY (Receive Holding Register Ready) flag is set as soon as a character has been received in FLEX_TWI_RHR (TWI Receive Holding Register). RXRDY is reset when reading FLEX_TWI_RHR.

TWI continues receiving data until a STOP condition or a REPEATED_START + an address different from SADR is detected. Note that at the end of the write sequence TXCOMP flag is set and SVACC reset.

See figure "Write Access Ordered by a Master" below.

45.9.5.3.3 Clock Stretching Sequence

If FLEX_TWI_THR or FLEX_TWI_RHR is not written/read in time, the TWI performs a clock stretching.

Clock stretching information is given by the SCLWS (Clock Wait State) bit.

See figures "Clock Stretching in Read Mode" and "Clock Stretching in Write Mode" below.

Note: Clock stretching can be disabled by configuring the FLEX_TWI_SMR.SCLWSDIS bit. In that case, UNRE and OVRE flags will indicate underrun (when FLEX_TWI_THR is not filled on time) or overrun (when FLEX_TWI_RHR is not read on time).

45.9.5.3.4 General Call

In the case where a GENERAL CALL is performed, the GACC (General Call Access) flag is set.

After GACC is set, it is up to the user to interpret the meaning of the GENERAL CALL and to decode the new address programming sequence.

See figure "Master Performs a General Call" below.

Downloaded from Arrow.com.

45.9.5.4 Data Transfer

45.9.5.4.1 Read Operation

The Read mode is defined as a data requirement from the master.

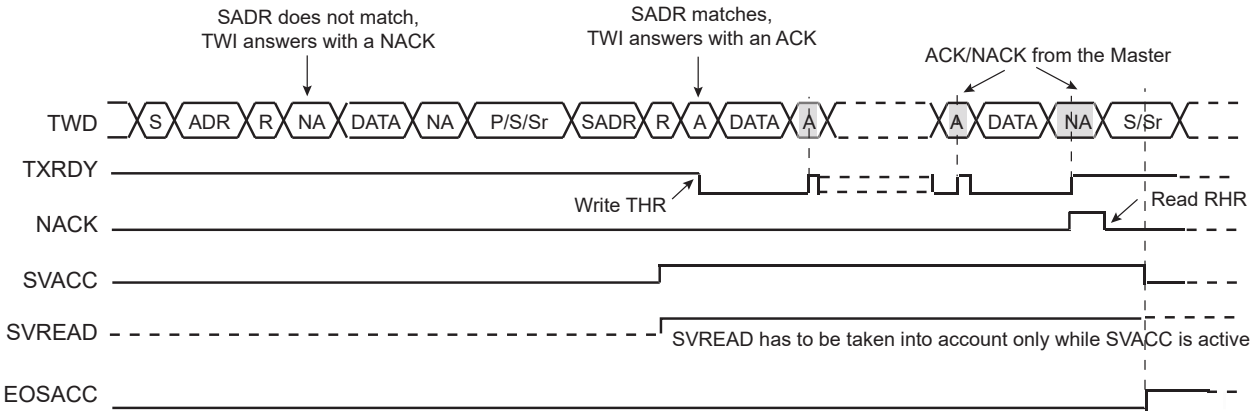
After a START or a REPEATED START condition is detected, the decoding of the address starts. If the slave address (SADR) is decoded, SVACC is set and SVREAD indicates the direction of the transfer.

Until a STOP or REPEATED START condition is detected, TWI continues sending data loaded in FLEX_TWI_THR.

If a STOP condition or a REPEATED START + an address different from SADR is detected, SVACC is reset.

The following figure describes the read operation.

Figure 45-120. Read Access Ordered by a Master



Note:

1. When SVACC is low, the state of SVREAD becomes irrelevant.
2. TXRDY is reset when data has been transmitted from FLEX_TWI_THR to the internal shifter and set when this data has been acknowledged or non acknowledged.

45.9.5.4.2 Write Operation

The Write mode is defined as a data transmission from the master.

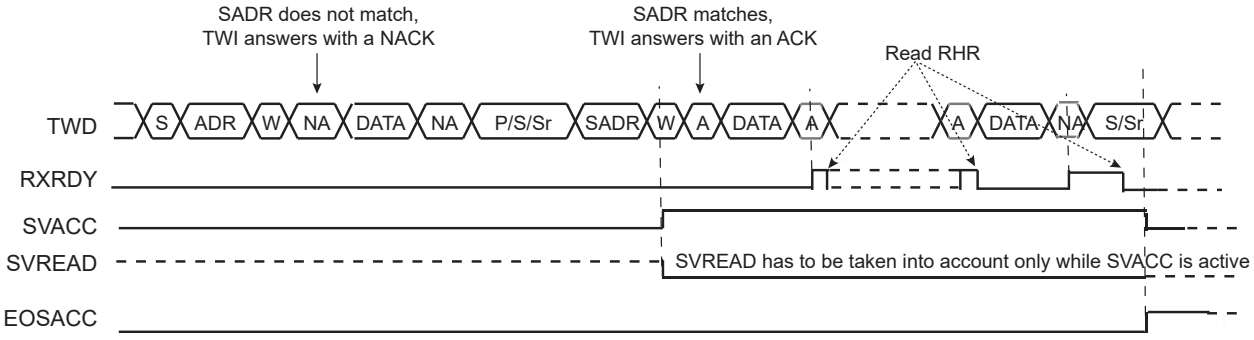
After a START or a REPEATED START, the decoding of the address starts. If the slave address is decoded, SVACC is set and SVREAD indicates the direction of the transfer (SVREAD is low in this case).

Until a STOP or REPEATED START condition is detected, TWI stores the received data in FLEX_TWI_RHR.

If a STOP condition or a REPEATED START + an address different from SADR is detected, SVACC is reset.

The following figure describes the write operation.

Figure 45-121. Write Access Ordered by a Master



Note:

1. When SVACC is low, the state of SVREAD becomes irrelevant.
2. RXRDY is set when data has been transmitted from the internal shifter to FLEX_TWI_RHR, and reset when this data is read.

Downloaded from Arrow.com.

45.9.5.4.3 General Call

The general call is performed in order to change the address of the slave.

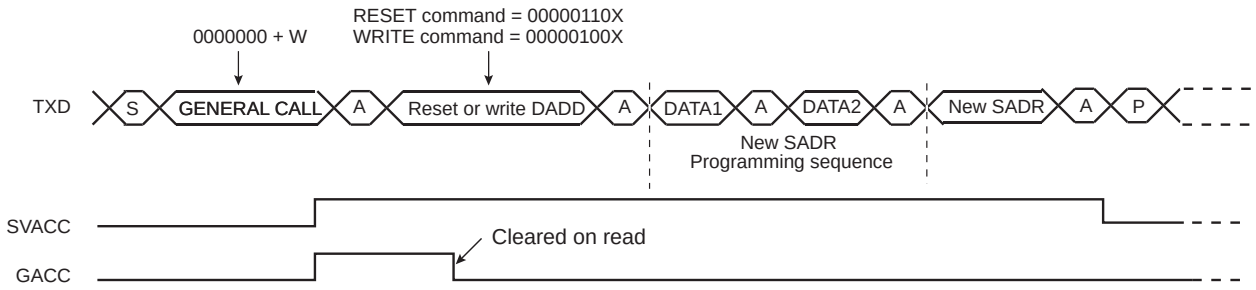
If a GENERAL CALL is detected, GACC is set.

After the detection of general call, it is up to the user to decode the commands which follow.

In case of a WRITE command, the user has to decode the programming sequence and program a new SADR if the programming sequence matches.

The following figure describes the general call access.

Figure 45-122. Master Performs a General Call



Note: This method enables to create a user-specific programming sequence by choosing the number of programming bytes. The programming sequence has to be provided to the master.

45.9.5.4.4 Clock Stretching

In both Read and Write modes, it may happen that the FLEX_TWI_THR/FLEX_TWI_RHR buffer is not filled/emptied before the transmission/reception of a new character. In this case, to avoid sending/receiving undesired data, a clock stretching mechanism is implemented.

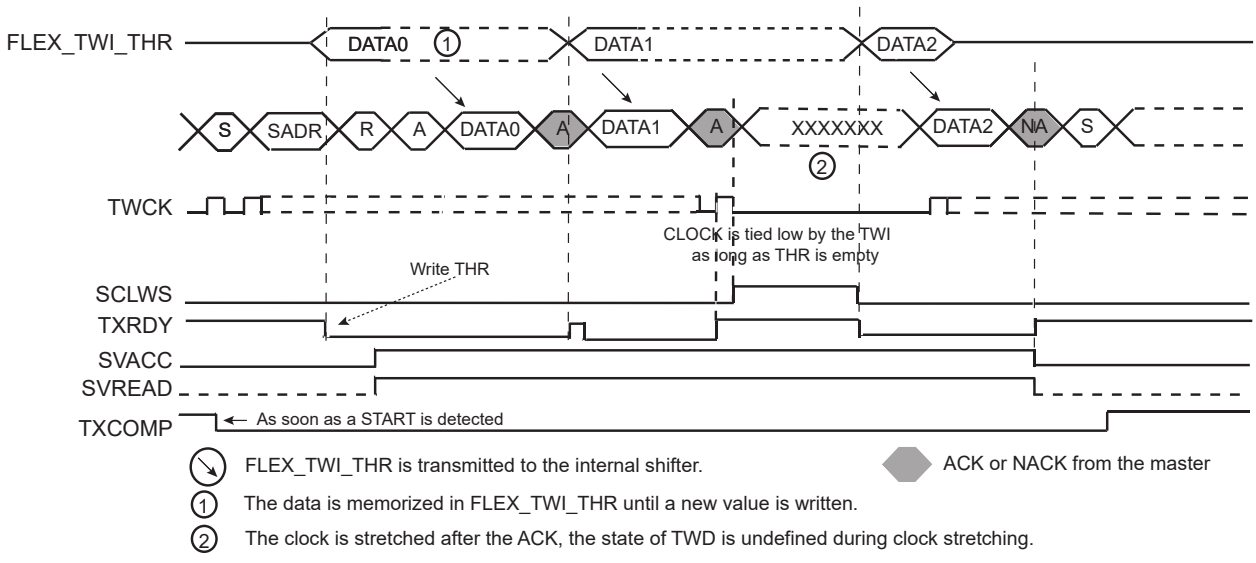
Note: Clock stretching can be disabled by setting the FLEX_TWI_SMR.SCLWSDIS bit. In that case, the UNRE and OVRE flags indicate an underrun (when FLEX_TWI_THR is not filled on time) or an overrun (when FLEX_TWI_RHR is not read on time).

— Clock Stretching in Read Mode

The clock is tied low if the internal shifter is empty and if a STOP or REPEATED START condition was not detected. It is tied low until the internal shifter is loaded.

The following figure describes clock stretching in Read mode.

Figure 45-123. Clock Stretching in Read Mode



Downloaded from Arrow.com.

Note:

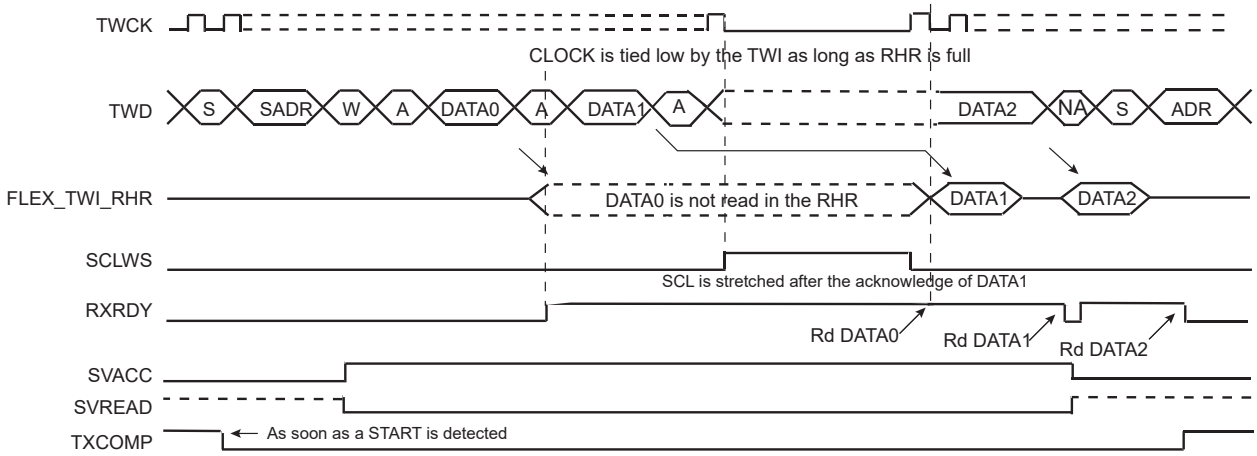
1. TXRDY is reset when data has been written in FLEX_TWI_THR to the internal shifter, and set when this data has been acknowledged or non acknowledged.
2. At the end of the read sequence, TXCOMP is set after a STOP or after a REPEATED_START + an address different from SADR.
3. SCLWS is automatically set when the clock stretching mechanism is started.

— Clock Stretching in Write Mode

The clock is tied low if the internal shifter and FLEX_TWI_RHR are full. If a STOP or REPEATED_START condition was not detected, it is tied low until FLEX_TWI_RHR is read.

The following figure describes the clock stretching in Write mode.

Figure 45-124. Clock Stretching in Write Mode



Note:

1. At the end of the read sequence, TXCOMP is set after a STOP or after a REPEATED_START + an address different from SADR.
2. SCLWS is automatically set when the clock stretching mechanism is started and automatically reset when the mechanism is finished.

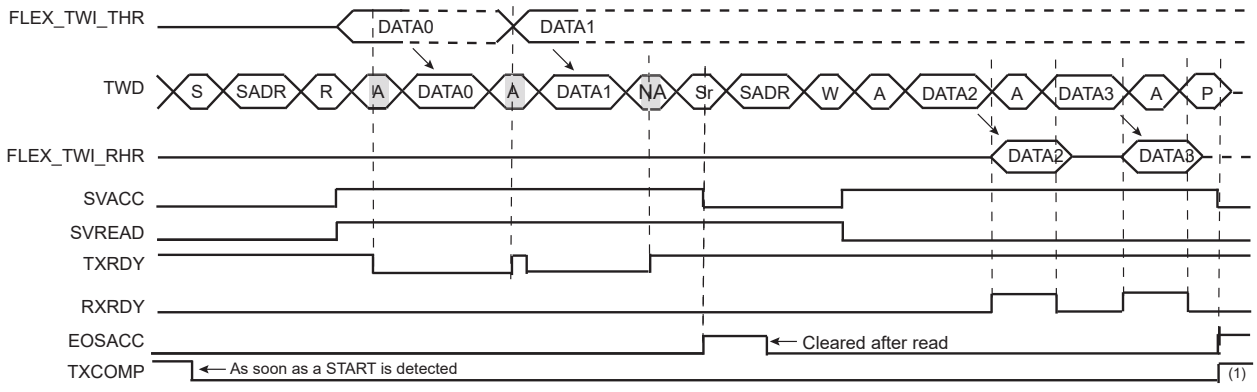
45.9.5.4.5 Reversal after a Repeated Start

— Reversal of Read to Write

The master initiates the communication by a read command and finishes it by a write command.

The following figure describes the repeated start and the reversal from Read mode to Write mode.

Figure 45-125. Repeated Start and Reversal from Read Mode to Write Mode



Note:
1. TXCOMP is only set at the end of the transmission because after the repeated start, SADR is detected again.

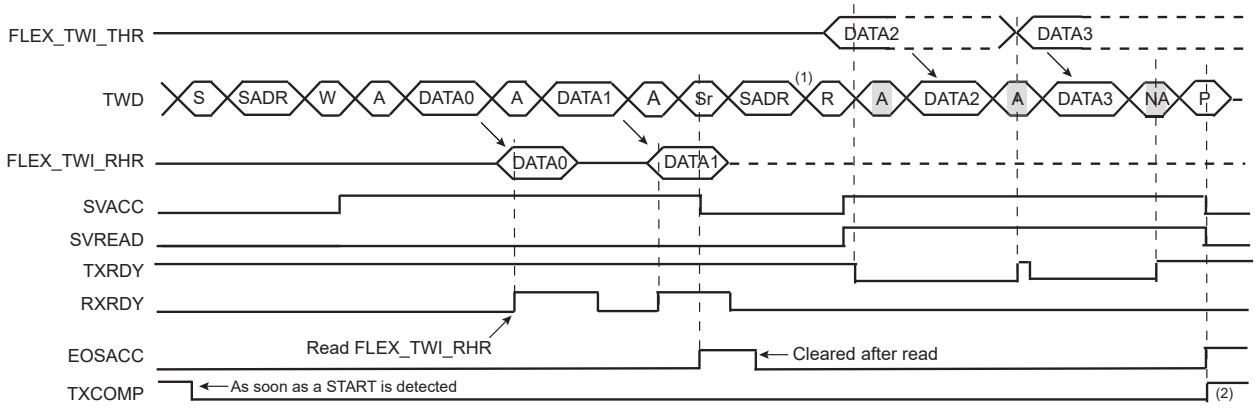
Downloaded from Arrow.com.

— Reversal of Write to Read

The master initiates the communication by a write command and finishes it by a read command.

The following figure describes the repeated start and the reversal from Write mode to Read mode.

Figure 45-126. Repeated Start and Reversal from Write Mode to Read Mode



Notes:
1. In this case, if FLEX_TWI_THR has not been written at the end of the read command, the clock is automatically stretched before the ACK.
2. TXCOMP is only set at the end of the transmission because after the repeated start, SADR is detected again.

45.9.5.4.6 SMBus Mode

SMBus mode is enabled when the FLEX_TWI_CR.SMEN bit is written to one. SMBus mode operation is similar to I²C operation with the following exceptions:

1. Only 7-bit addressing can be used.
2. The SMBus standard describes a set of timeout values to ensure progress and throughput on the bus. These timeout values must be programmed into FLEX_TWI_SMBTR.
3. Transmissions can optionally include a CRC byte, called Packet Error Check (PEC).
4. A set of addresses have been reserved for protocol handling, such as alert response address (ARA) and host header (HH) address. Address matching on these addresses can be enabled by configuring FLEX_TWI_CR appropriately.

— Packet Error Checking

Each SMBus transfer can optionally end with a CRC byte, called the PEC byte. Writing the FLEX_TWI_CR.PECEN bit to one will send/check the FLEX_TWI_ACR.PEC field in the current transfer. The PEC generator is always updated on every bit transmitted or received, so that PEC handling on following linked transfers will be correct.

In Slave Receiver mode, the master calculates a PEC value and transmits it to the slave after all data bytes have been transmitted. Upon reception of this PEC byte, the slave will compare it to the PEC value it has computed itself. If the values match, the data was received correctly, and the slave will return an ACK to the master. If the PEC values differ, data was corrupted, and the slave will return a NACK value. The FLEX_TWI_SR.PECERR bit is set automatically if a PEC error occurred.

In Slave Transmitter mode, the slave calculates a PEC value and transmits it to the master after all data bytes have been transmitted. Upon reception of this PEC byte, the master will compare it to the PEC value it has computed itself. If the values match, the data was received correctly. If the PEC values differ, data was corrupted, and the master must take appropriate action.

See section Slave Read/Write Flowcharts for detailed flowcharts.

— Timeouts

The TWI SMBus Timing Register (FLEX_TWI_SMBTR) configures the SMBus timeout values. If a timeout occurs, the slave will leave the bus. Furthermore, the FLEX_TWI_SR.TOUT bit is set.

45.9.5.5 High-Speed Slave Mode

High-speed mode is enabled when the FLEX_TWI_CR.HSEN bit is written to one. Furthermore, the analog pad filter must be enabled, the FLEX_TWI_FILTR.PADFEN bit must be written to one and the FLEX_TWI_FILTR.FILT bit must be cleared. TWI High-speed mode operation is similar to TWI operation with the following exceptions:

Downloaded from Arrow.com.

1. A master code is received first at normal speed before entering High-speed mode period.
2. When TWI High-speed mode is active, clock stretching is only allowed after acknowledge (ACK), not-acknowledge (NACK), START (S) or repeated START (Sr) (asa consequence, OVF may happen).

TWI High-speed mode allows transfers of up to 3.4 Mbit/s.

The TWI slave in High-speed mode requires that the peripheral clock runs at a minimum of 14 MHz if slave clock stretching is enabled (SCLWSDIS bit at '0'). If slave clock stretching is disabled (SCLWSDIS bit at '1'), the peripheral clock must run at a minimum of 11 MHz (assuming the system has no latency).

Note:

1. When slave clock stretching is disabled, FLEX_TWI_RHR must always be read before receiving the next data (MASTER write frame). It is strongly recommended to use either the polling method on the FLEX_TWI_SR.RXRDY flag, or the DMA. If the receive is managed by an interrupt, the TWI interrupt priority must be set to the right level and its latency minimized to avoid receive overrun.
2. When slave clock stretching is disabled, FLEX_TWI_THR must be filled with the first data to send before the beginning of the frame (MASTER read frame). It is strongly recommended to use either the polling method on the FLEX_TWI_SR.TXRDY flag, or the DMA. If the transmit is managed by an interrupt, the TWI interrupt priority must be set to the right level and its latency minimized to avoid transmit underrun.

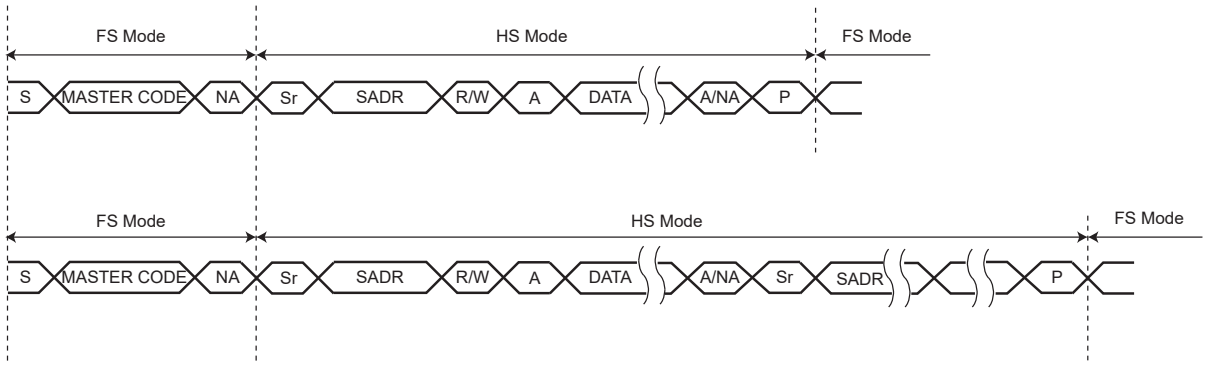
45.9.5.5.1 Read/Write Operation

A TWI high-speed frame always begins with the following sequence:

1. START condition (S)
2. Master Code (0000 1XXX)
3. Not-acknowledge (NACK)

When the TWI is programmed in Slave mode and TWI High-speed mode is activated, master code matching is activated and internal timings are set to match the TWI High-speed mode requirements.

Figure 45-127. High-Speed Mode Read/Write



45.9.5.5.2 Usage

TWI High-speed mode usage is the same as the standard TWI (see section Read/Write Flowcharts).

45.9.5.6 Alternative Command

In Slave mode, the Alternative Command mode is used when the SMBus mode is enabled to send or check the PEC byte.

The Alternative Command mode is enabled by setting the ACMEN bit of the TWIHS Control Register, and the transfer is configured in TWIHS_ACR.

For a combined transfer with PEC, only the NPEC bit in TWIHS_ACR must be set as the PEC byte is sent once at the end of the frame.

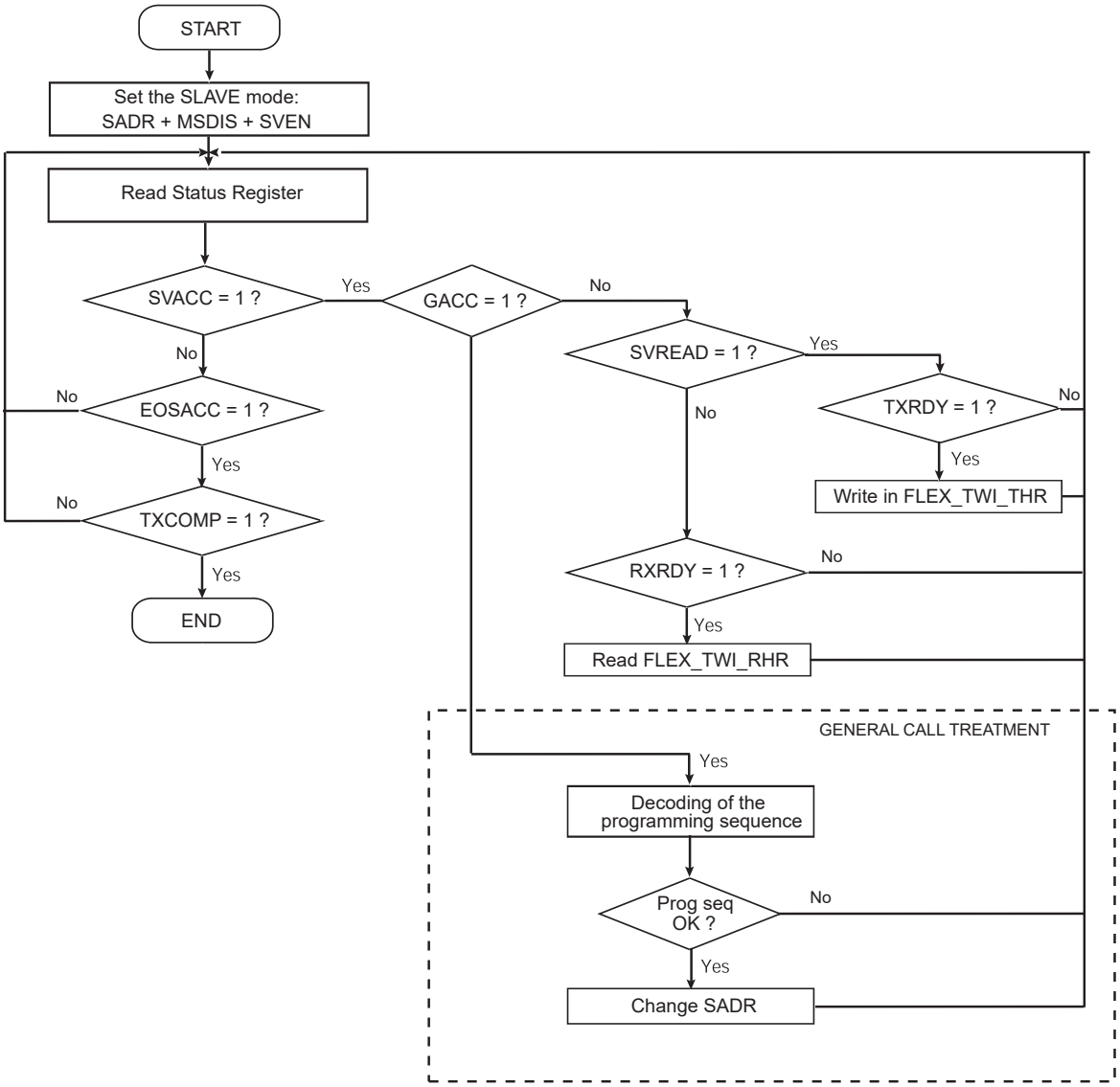
See section Slave Read/Write Flowcharts for detailed flowcharts.

45.9.5.7 Slave Read/Write Flowcharts

The flowchart shown in the following figure gives an example of read and write operations in Slave mode. A polling or interrupt method can be used to check the status bits. The interrupt method requires that the Interrupt Enable Register (FLEX_TWI_IER) be configured first.

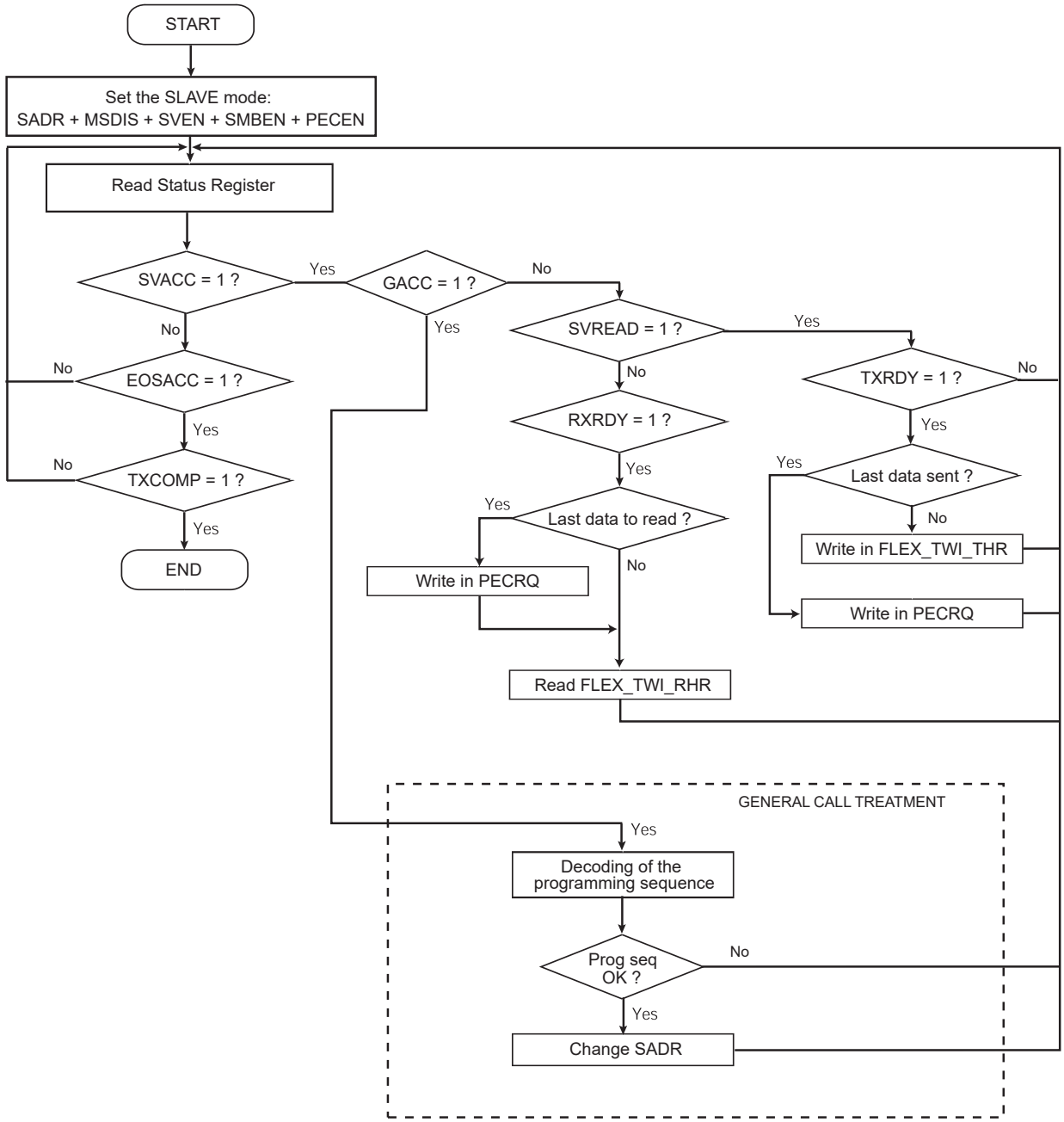
Downloaded from Arrow.com.

Figure 45-128. Read/Write in Slave Mode



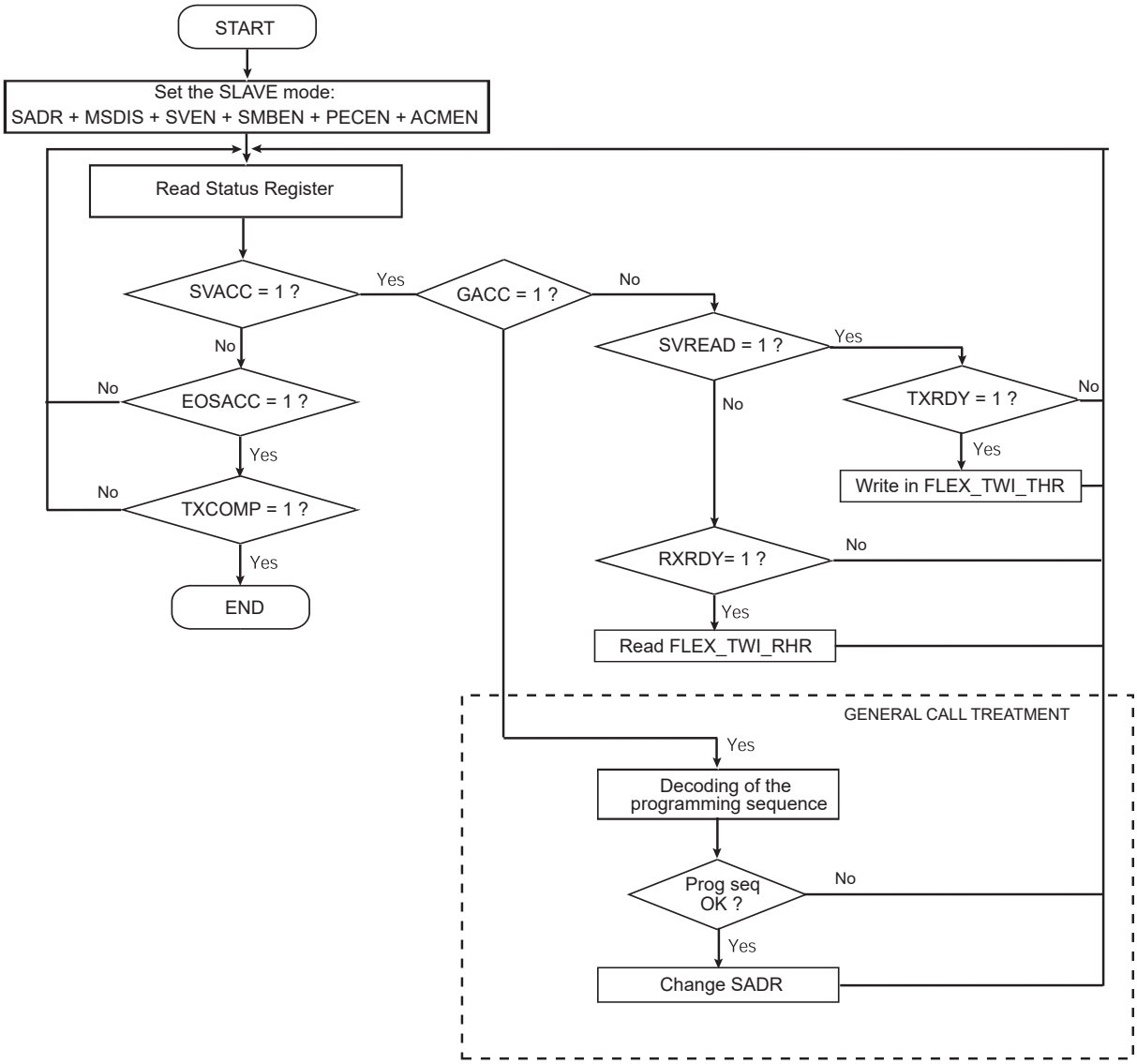
Downloaded from Arrow.com.

Figure 45-129. Read/Write in Slave Mode with SMBus PEC



Downloaded from Arrow.com.

Figure 45-130. Read/Write in Slave Mode with SMBus PEC and Alternative Command Mode



45.9.6 TWI FIFOs

45.9.6.1 Overview

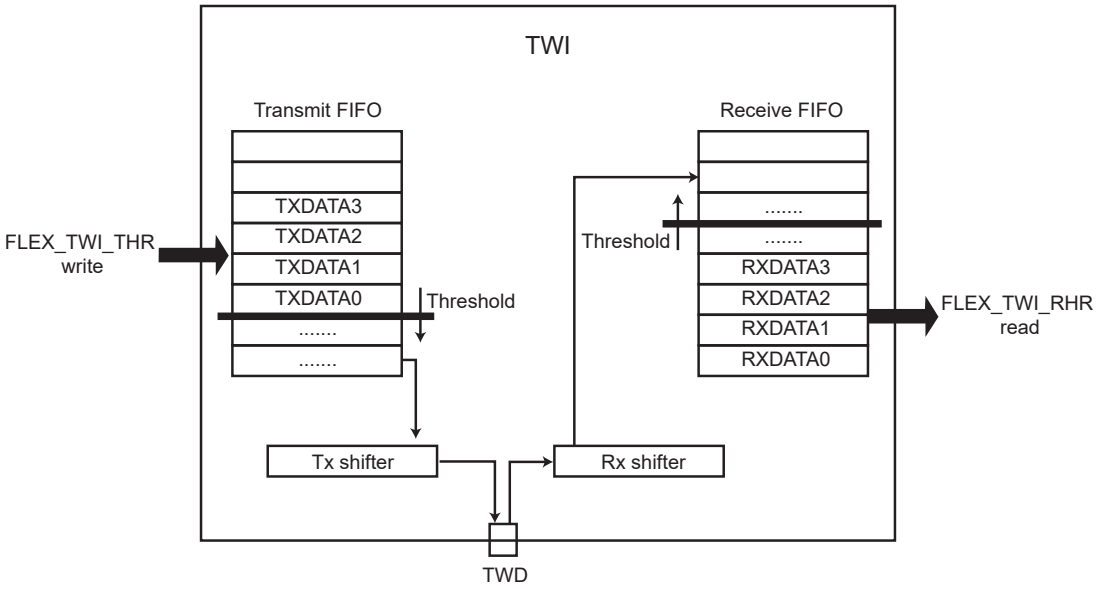
The TWI includes two FIFOs which can be enabled/disabled using FLEX_TWI_CR.FIFOEN/FIFODIS. Both Master and Slave modes must be disabled before enabling or disabling the FIFOs (FLEX_TWI_CR.MSDIS/SVDIS).

Writing FLEX_TWI_CR.FIFOEN to ‘1’ enables a 16-data Transmit FIFO and a 16-data Receive FIFO.

It is possible to write or to read single or multiple data in the same access to FLEX_TWI_THR/RHR, depending on FLEX_TWI_FMR.TXRDYM/RXRDYM settings.

Downloaded from Arrow.com.

Figure 45-131. FIFOs Block Diagram



45.9.6.2 Sending Data with FIFO Enabled

When the Transmit FIFO is enabled, write access to FLEX_TWI_THR loads the Transmit FIFO.

The Transmit FIFO level is provided in FLEX_TWI_FLR.TXFL. If the FIFO can accept the number of data to be transmitted, there is no need to monitor FLEX_TWI_SR.TXRDY and the data can be successively written in FLEX_TWI_THR.

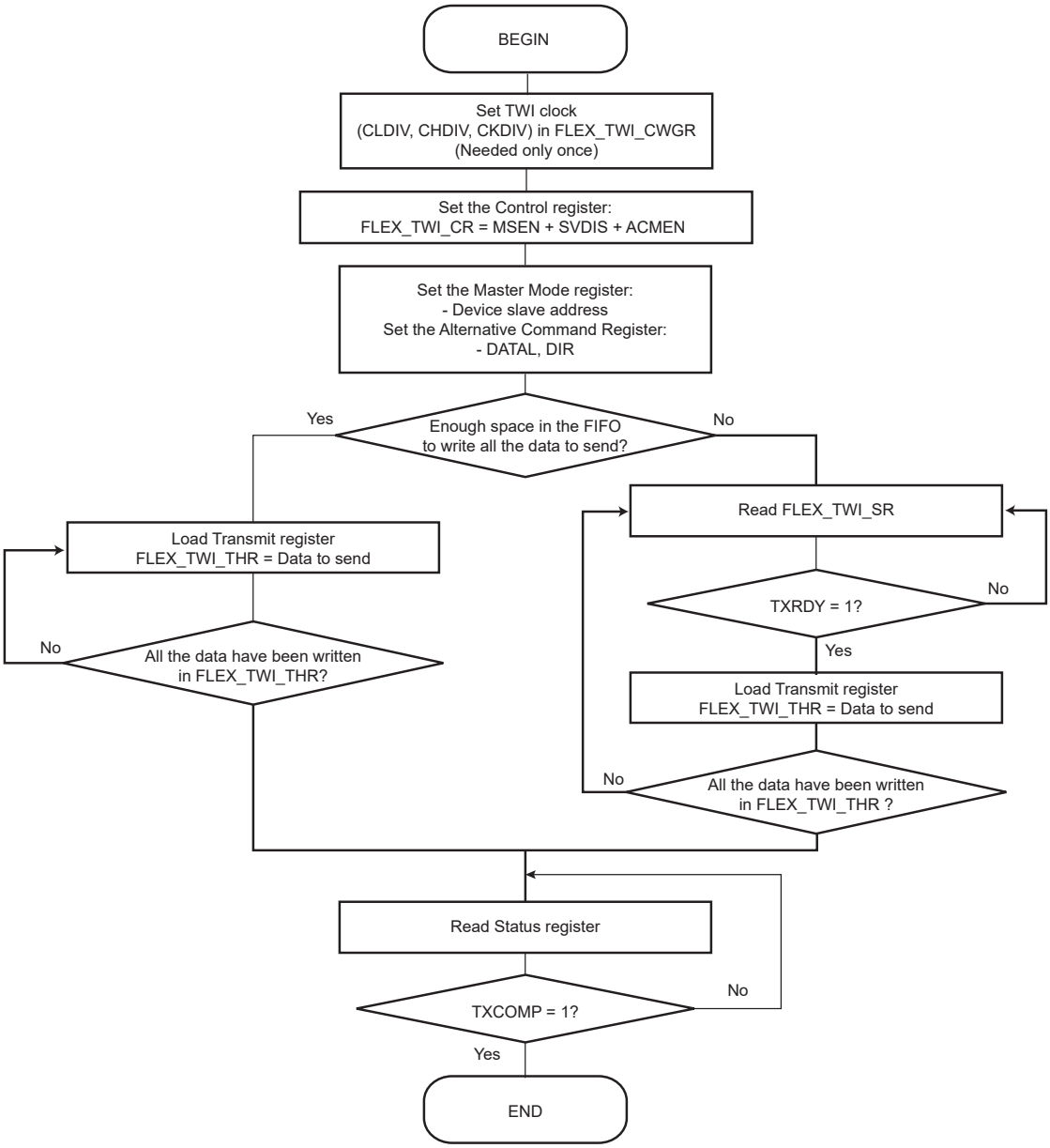
If the FIFO cannot accept the data due to insufficient space, wait for the TXRDY flag to be set before writing the data in FLEX_TWI_THR.

When the space in the FIFO allows only a portion of the data to be written, the TXRDY flag must be monitored before writing the remaining data.

See figures Sending Data with FIFO Enabled in Master Mode and Sending/Receiving Data with FIFO Enabled in Slave Mode.

Downloaded from Arrow.com.

Figure 45-132. Sending Data with FIFO Enabled in Master Mode



45.9.6.3 Receiving Data with FIFO Enabled

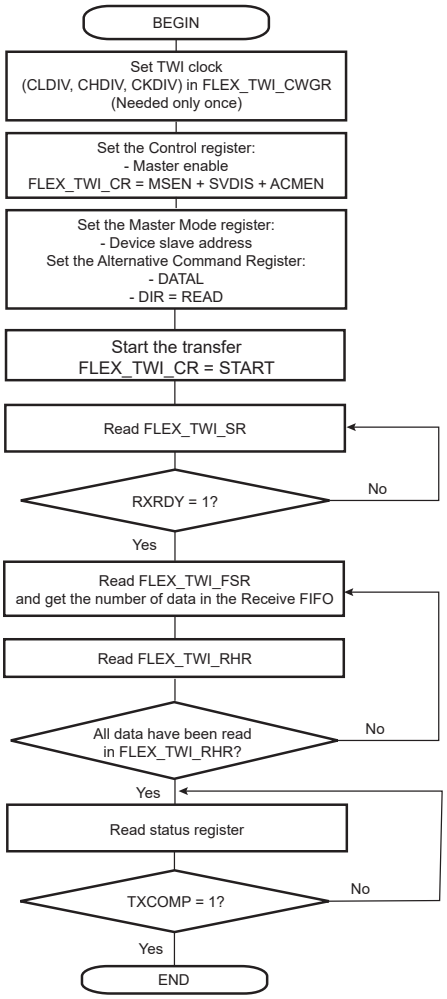
When the Receive FIFO is enabled, FLEX_TWI_RHR access reads the FIFO.

When data are present in the Receive FIFO (RXRDY flag set to ‘1’), the exact number of data can be checked with FLEX_TWI_FLR.RXFL. All the data can be read successively in FLEX_TWI_RHR without checking the FLEX_TWI_SR.RXRDY flag between each access.

See figures Receiving Data with FIFO Enabled in Master Mode and Sending/Receiving Data with FIFO Enabled in Slave Mode.

Downloaded from Arrow.com.

Figure 45-133. Receiving Data with FIFO Enabled in Master Mode

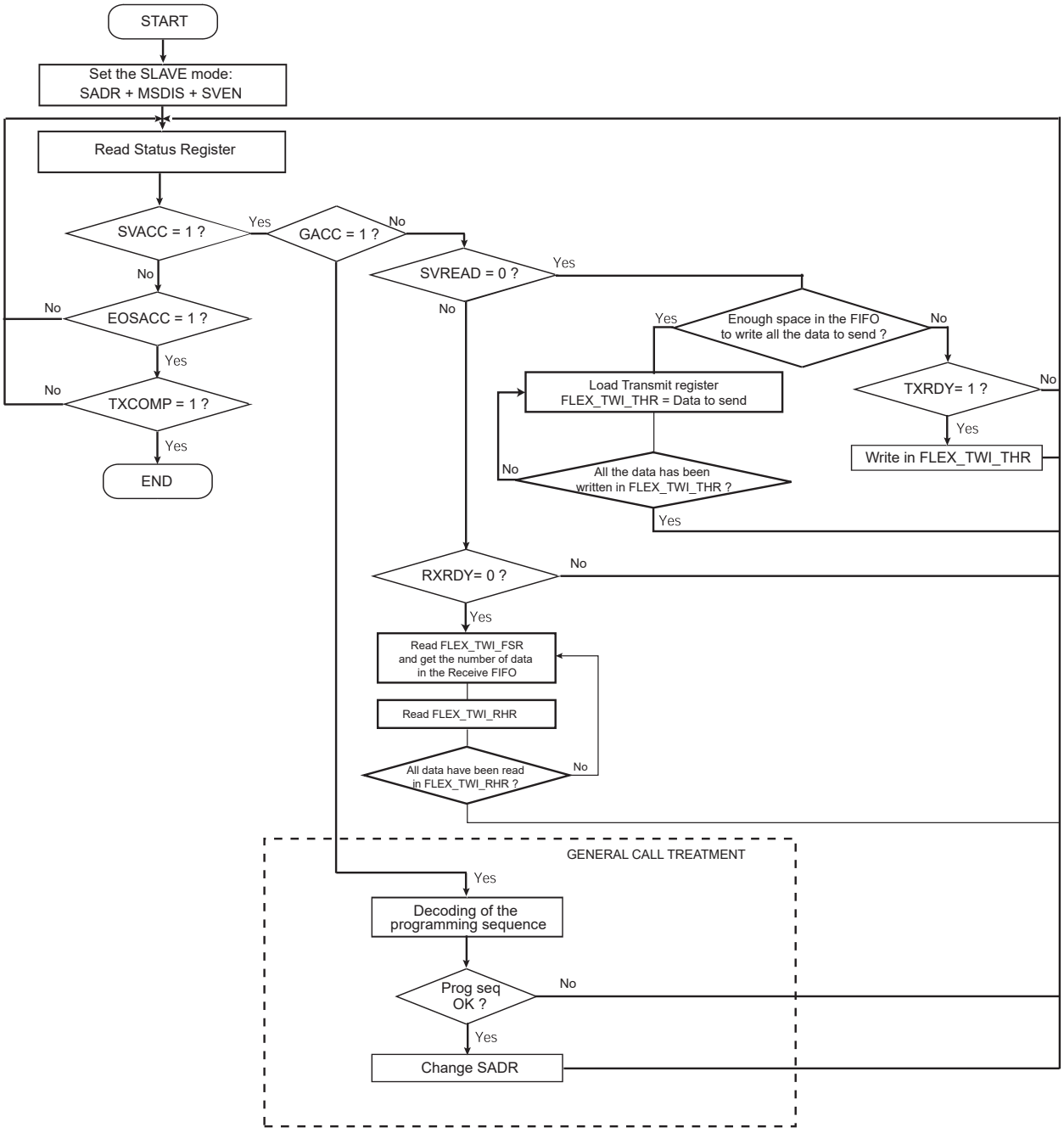


45.9.6.4 Sending/Receiving with FIFO Enabled in Slave Mode

See sections Sending Data with FIFO Enabled and Receiving Data with FIFO Enabled for details.

Downloaded from Arrow.com.

Figure 45-134. Sending/Receiving Data with FIFO Enabled in Slave Mode



45.9.6.5 Clearing/Flushing FIFOs

Each FIFO can be cleared/flushed using FLEX_TWI_CR.TXFCLR/RXFCLR.

45.9.6.6 TXRDY and RXRDY Behavior

FLEX_TWI_SR.TXRDY/RXRDY flags display a specific behavior when FIFOs are enabled.

TXRDY indicates if a data can be written in the Transmit FIFO. Thus the TXRDY flag is set as long as the Transmit FIFO can accept new data. See figure TXRDY Behavior when TXRDYM = 0 in Master mode.

RXRDY indicates if an unread data is present in the Receive FIFO. Thus the RXRDY flag is set as soon as one unread data is in the Receive FIFO. Refer to figure RXRDY Behavior when RXRDYM = 0 in Master and Slave modes.

Downloaded from Arrow.com.

TXRDY and RXRDY behavior can be modified using the TXRDYM and RXRDYM fields in the TWI FIFO Mode Register (FLEX_TWI_FMR) to reduce the number of accesses to FLEX_TWI_THR/RHR.

As an example, in Master mode, the Transmit FIFO can be loaded with multiple data in the same access by configuring TXRDYM>0.

See TWI FIFO Mode Register for the FIFO configuration.

Figure 45-135. TXRDY Behavior when TXRDYM = 0 in Master Mode

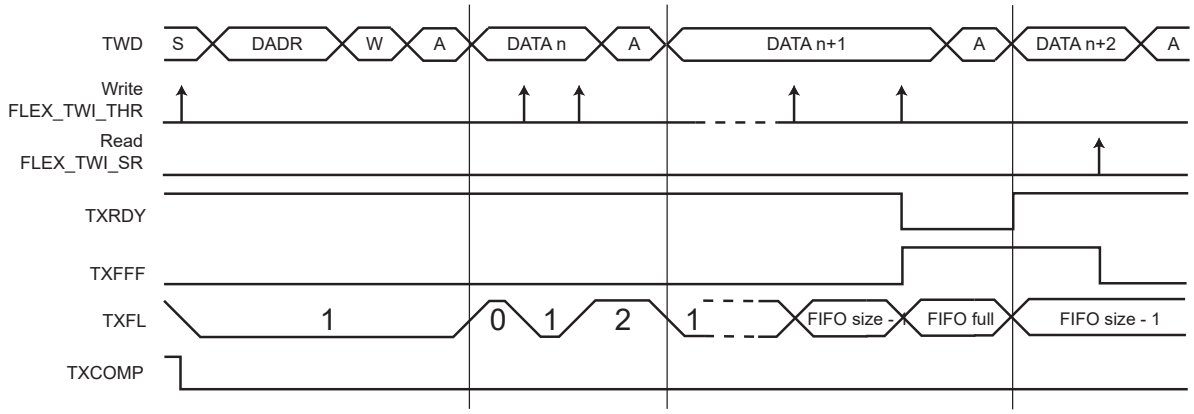


Figure 45-136. RXRDY Behavior when RXRDYM = 0 in Master and Slave Modes

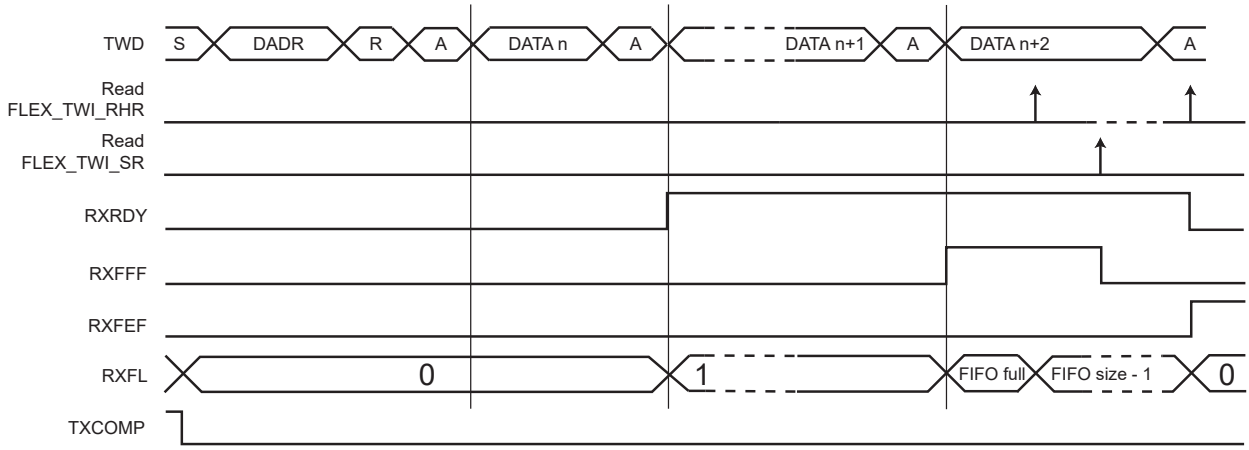
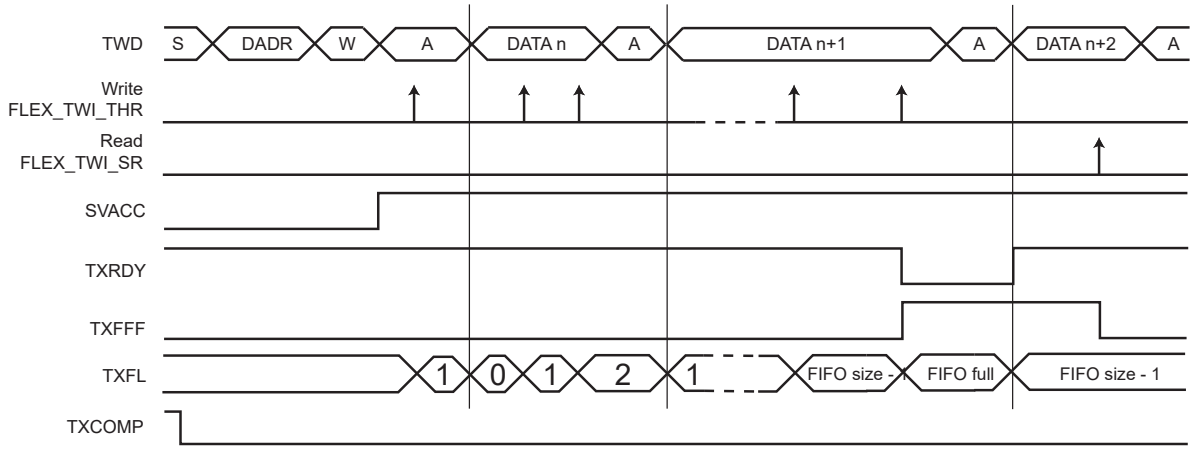


Figure 45-137. TXRDY Behavior when TXRDYM = 0 in Slave Mode



Downloaded from Arrow.com.

45.9.6.7 TWI Single Data Mode

In Single Data mode, only one data is written every time FLEX_TWI_THR is accessed, and only one data is read every time FLEX_TWI_RHR is accessed.

When FLEX_TWI_FMR.TXRDYM = 0, the Transmit FIFO operates in Single Data mode.

When FLEX_TWI_FMR.RXRDYM = 0, the Receive FIFO operates in Single Data mode.

See sections TWI Transmit Holding Register and TWI Receive Holding Register.

45.9.6.8 TWI Multiple Data Mode

Multiple Data mode minimizes the number of accesses by concatenating the data to send/read in one access.

When FLEX_TWI_FMR.TXRDYM > 0, the Transmit FIFO operates in Multiple Data mode.

When FLEX_TWI_FMR.RXRDYM > 0, the Receive FIFO operates in Multiple Data mode.

In Multiple Data mode, it is possible to write/read up to four data in one FLEX_TWI_THR/FLEX_TWI_RHR register access.

The number of data to write/read is defined by the size of the register access. If the access is a byte-size register access, only one data is written/read. If the access is a halfword size register access, then up to two data are read and only one data is written. Lastly, if the access is a wordsize register access, then up to four data are read and up to two data are written.

Written/Read data are always right-aligned, as described in sections TWI Receive Holding Register (FIFO Enabled) and TWI Transmit Holding Register (FIFO Enabled).

As an example, if the Transmit FIFO is empty and there are six data to send, either of the following write accesses may be performed:

- six FLEX_TWI_THR-byte write accesses
- three FLEX_TWI_THR-halfword write accesses
- one FLEX_TWI_THR-word write access and one FLEX_TWI_THR halfword write access

With a Receive FIFO containing six data, any of the following read accesses may be performed:

- six FLEX_TWI_RHR-byte read accesses
- three FLEX_TWI_RHR-halfword read accesses
- one FLEX_TWI_RHR-word read access and one FLEX_TWI_RHR-halfword read access

45.9.6.8.1 TXRDY and RXRDY Configuration

In Multiple Data mode, it is possible to write one or more data in the same FLEX_TWI_THR/FLEX_TWI_RHR access. The TXRDY flag indicates if one or more data can be written in the FIFO depending on the configuration of FLEX_TWI_FMR.TXRDYM/RXRDYM.

As an example, if two data are written each time in FLEX_TWI_THR, it is useful to configure the TXRDYM field to the value ‘1’ so that the TXRDY flag is at ‘1’ only when at least two data can be written in the Transmit FIFO.

In the same way, if four data are read each time in FLEX_TWI_RHR, it is useful to configure the RXRDYM field to the value ‘2’ so that the RXRDY flag is at ‘1’ only when at least four unread data are in the Receive FIFO.

45.9.6.8.2 DMA

When FIFOs operate in Multiple Data mode, the DMA transfer type must be configured in byte, halfword or word depending on the FLEX_TWI_FMR.TXRDYM/RXRDYM settings.

45.9.6.9 Transmit FIFO Lock

If a frame is terminated early due to a not-acknowledge error (NACK flag), SMBus timeout error (TOUT flag) or master code acknowledge error (MACK flag), a lock is set on the Transmit FIFO preventing any new frame from being sent until it is cleared. This allows clearing the FIFO if needed, resetting DMA channels, etc., without any risk.

FLEX_TWI_SR.LOCK is used to check the state of the Transmit FIFO lock.

The Transmit FIFO lock can be cleared by setting FLEX_TWI_CR.TXFLCLR to ‘1’.

45.9.6.10 FIFO Pointer Error

A FIFO overflow is reported in FLEX_TWI_FSR.

Downloaded from Arrow.com.

If the Transmit FIFO is full and a write access is performed on FLEX_TWI_THR, it generates a Transmit FIFO pointer error and sets FLEX_TWI_FSR.TXFPTEF.

In Multiple Data mode, if the number of data written in FLEX_TWI_THR (according to the register access size) is greater than the free space in the Transmit FIFO, a Transmit FIFO pointer error is generated and FLEX_TWI_FSR.TXFPTEF is set.

A FIFO underflow is reported in FLEX_TWI_FSR.

In Multiple Data mode, if the number of data read in FLEX_TWI_RHR (according to the register access size) is greater than the number of unread data in the Receive FIFO, a Receive FIFO pointer error is generated and FLEX_TWI_FSR.RXFPTEF is set.

No pointer error occurs if the FIFO state/level is checked before writing/reading in FLEX_TWI_THR/FLEX_TWI_RHR. The FIFO state/level can be checked either with TXRDY, RXRDY, TXFL or RXFL. When a pointer error occurs, other FIFO flags may not behave as expected; their states should be ignored.

If a Transmit or Receive pointer error occurs, a software reset must be performed using FLEX_TWI_CR.SWRST. Note that issuing a software while transmitting might leave a slave in an unknown state holding the TWD line. In such case, a Bus Clear Command will allow to make the slave release the TWD line (the first frame sent afterwards might not be received properly by the slave).

45.9.6.11 FIFO Thresholds

Each Transmit and Receive FIFO includes a threshold feature used to set a flag and an interrupt when a FIFO threshold is crossed. Thresholds are defined as a number of data in the FIFO, and the FIFO state (TXFL or RXFL) represents the number of data currently in the FIFO.

The Transmit FIFO threshold can be set using the field FLEX_TWI_FMR.TXFTHRES. Each time the Transmit FIFO level goes from ‘above threshold’ to ‘equal to or below threshold’, the flag FLEX_TWI_FESR.TXFTHF is set. The application is warned that the Transmit FIFO has reached the defined threshold and that it can be reloaded.

The Receive FIFO threshold can be set using the field FLEX_TWI_FMR.RXFTHRES. Each time the Receive FIFO level goes from ‘below threshold’ to ‘equal to or above threshold’, the flag FLEX_TWI_FESR.RXFTHF is set. The application is warned that the Receive FIFO has reached the defined threshold and that it can be read to prevent an underflow.

The TXFTHF and RXFTHF flags can be configured to generate an interrupt using FLEX_TWI_FIER and FLEX_TWI_FIDR.

45.9.6.12 FIFO Flags

FIFOs come with a set of flags which can be configured to generate interrupts through FLEX_TWI_FIER and FLEX_TWI_FIDR.

FIFO flags state can be read in FLEX_TWI_FSR. They are cleared when FLEX_TWI_FSR is read.

45.9.7 TWI Comparison Function on Received Character

The TWI has the capability to extend the address matching on up to three slave addresses. The FLEX_TWI_SMR.SADR1EN/SADR2EN/SADR3EN bits enable address matching on additional addresses which can be configured through the FLEX_TWI_SWMR.SADR1/SADR2/SADR3 fields. The DATAMEN bit has no effect.

The SVACC bit is set when there is a comparison match with the received slave address.

45.9.8 Sniffer Mode

The Slave Sniffer mode of a TWI can be enabled to ease the analysis/debug of a TWI bus activity. The TWI bus to be analyzed can be mastered by one TWI master embedded in the product or mastered by an I2C master outside the product.

In this mode, the TWI reports all (or part of) the TWI bus activity without impacting the TWI bus (no bus drive is performed). Depending on the MASK field value, only some specific transfers can be logged instead of the whole activity.

The peripheral TWIn+1 can be configured to analyze the peripheral TWIn (n being the instance index of the TWI) in a full transparent mode via predefined internal connections between TWI instances.

Downloaded from Arrow.com.

The predefined internal connections provides the capability to use the TWD and TWCK pins for alternate functions while the TWI peripheral is configured in Sniffer Slave mode and the selected TWI bus to analyze is carried on these internal links.

The following fields must be programmed before entering Slave mode:

1. FLEX_TWI_SMR.SADR: Use the slave device address to indicate which frame(s) to log.
2. FLEX_TWI_SMR.MASK: Indicate which SADR bits should be masked and thus which transfers should be logged (set to 0x7F to log the whole TWI bus activity; all SADR bits are masked in this case). General Call accesses will always match.
3. FLEX_TWI_SMR.BSEL: Select the TWI bus to analyze (see Figure 45-138).
4. FLEX_TWI_SMR.SNIFF: Set to '1' to enable Slave Sniffer mode.
5. FLEX_TWI_CR.MSDIS: Disable Master mode.
6. FLEX_TWI_CR.SVEN: Enable Slave mode.

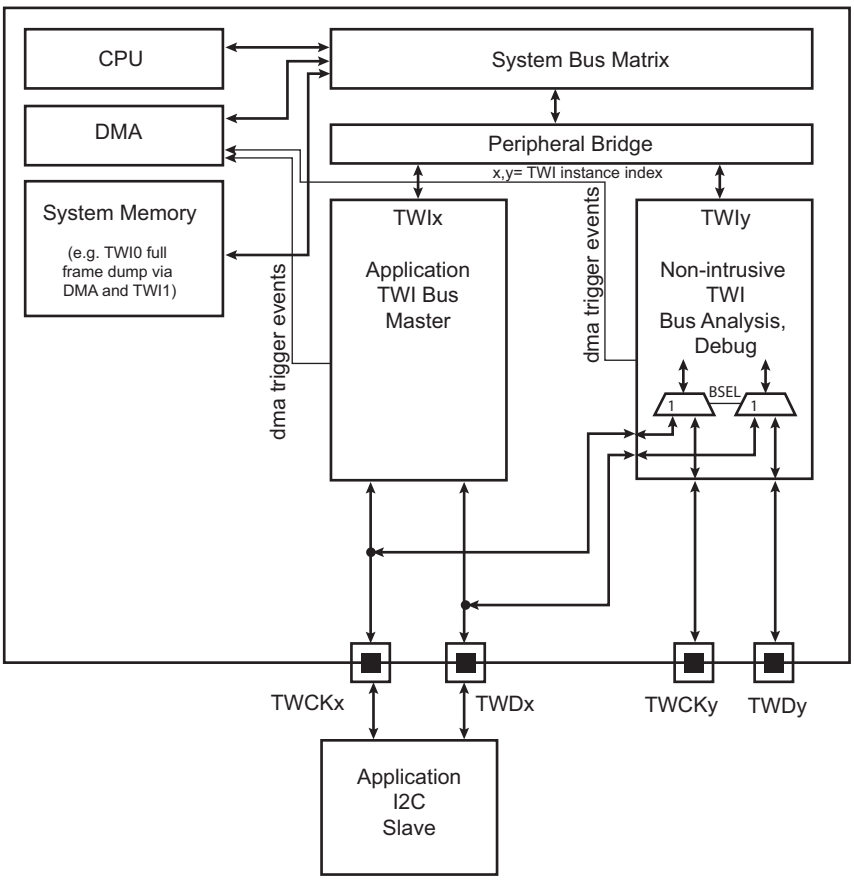
As the device receives the clock, values written in FLEX_TWI_CWGR are not relevant.

Once configured in Slave Sniffer mode, the FLEX_TWI_SR.RXRDY bit indicates when a transfer has been logged in FLEX_TWI_RHR. An interrupt can be generated if configured. The FLEX_TWI_SR.OVRE flag indicates if an overrun error occurred if the application is not fast enough to read FLEX_TWI_RHR.

In Slave Sniffer mode, FLEX_TWI_RHR logs data as follows:

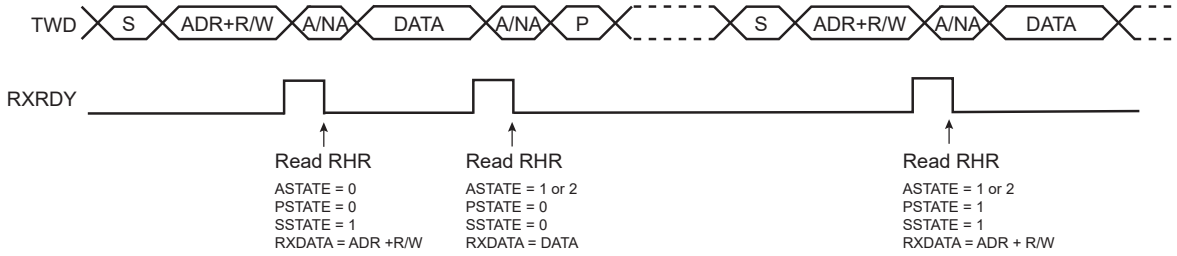
- The RXDATA field reports the sniffed 8-bit data field.
- The SSTATE field indicates if a START condition has been detected before the 8-bit data field.
- The PSTATE field indicates if a STOP condition has been detected after the previously sniffed 8-bit data field.
- The ASTATE field indicates which acknowledge condition has been detected after the previously sniffed 8-bit data field.

Figure 45-138. Sniffer Mode Application Overview



Downloaded from Arrow.com.

Figure 45-139. Slave Sniffer Mode Log



45.9.9 TWI Register Write Protection

The FLEXCOM operating mode (FLEX_MR.OPMODE) must be set to FLEX_MR_OPMODE_TWI to enable access to the write protection registers.

To prevent any single software error from corrupting TWI behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the TWI Write Protection Mode Register (FLEX_TWI_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the TWI Write Protection Status Register (FLEX_TWI_WPSR) is set and the Write Protection Violation Source (WPVSRC) field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading FLEX_TWI_WPSR.

The following register(s) can be write-protected when WPEN is set:

- TWI Slave Mode Register
- TWI Clock Waveform Generator Register
- TWI SMBus Timing Register
- TWI FIFO Mode Register

The following register(s) can be write-protected when WPITEN is set:

- TWI Interrupt Enable Register
- TWI Interrupt Disable Register

The following register(s) can be write-protected when WPCREN is set:

- TWI Control Register

Downloaded from Arrow.com.

45.10 Register Summary

Offset	Name	Bit Pos.								
0x00	FLEX_MR	31:24								
		23:16								
		15:8								
		7:0							OPMODE[1:0]	
0x04 ... 0x0F	Reserved									
0x10	FLEX_RHR	31:24								
		23:16								
		15:8	RXDATA[15:8]							
		7:0	RXDATA[7:0]							
0x14 ... 0x1F	Reserved									
0x20	FLEX_THR	31:24								
		23:16								
		15:8	TXDATA[15:8]							
		7:0	TXDATA[7:0]							
0x24 ... 0x01FF	Reserved									
0x0200	FLEX_US_CR	31:24	FIFODIS	FIFOEN		REQCLR		TXFLCLR	RXFCLR	TXFCLR
		23:16			LINWKUP	LINABT	RTSDIS	RTSEN		
		15:8	RETTO	RSTNACK	RSTIT	SENDA	STTTO	STPBRK	STTBRK	RSTSTA
		7:0	TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX		
0x0204	FLEX_US_MR	31:24	ONEBIT	MODSYNC	MAN	FILTER		MAX_ITERATION[2:0]		
		23:16	INVDATA	VAR_SYNC	DSNACK	INACK	OVER	CLKO	MODE9	MSBF
		15:8	CHMODE[1:0]		NBSTOP[1:0]		PAR[2:0]			SYNC
		7:0	CHRL[1:0]		USCLKS[1:0]		USART_MODE[3:0]			
0x0208	FLEX_US_IER	31:24								MANE
		23:16		CMP			CTSIC			
		15:8			NACK			ITER	TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY
0x0208	FLEX_US_IER (LIN_MODE)	31:24	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
		23:16								
		15:8	LINTC	LINID	LINBK				TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE				TXRDY	RXRDY
0x0208	FLEX_US_IER (LON_MODE)	31:24				LBLOVFE	LRXD	LFET	LCOL	LTXD
		23:16								
		15:8						UNRE	TXEMPTY	
		7:0	LCRCE	LSFE	OVRE				TXRDY	RXRDY
0x020C	FLEX_US_IDR	31:24								MANE
		23:16		CMP			CTSIC			
		15:8			NACK			ITER	TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY
0x020C	FLEX_US_IDR (LIN_MODE)	31:24	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
		23:16								
		15:8	LINTC	LINID	LINBK				TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE				TXRDY	RXRDY
0x020C	FLEX_US_IDR (LON_MODE)	31:24				LBLOVFE	LRXD	LFET	LCOL	LTXD
		23:16								
		15:8						UNRE	TXEMPTY	
		7:0	LCRCE	LSFE	OVRE				TXRDY	RXRDY

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x0210	FLEX_US_IMR	31:24								MANE
		23:16		CMP			CTSIC			
		15:8			NACK			ITER	TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY
0x0210	FLEX_US_IMR (LIN_MODE)	31:24	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
		23:16								
		15:8	LINTC	LINID	LINBK				TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE				TXRDY	RXRDY
0x0210	FLEX_US_IMR (LON_MODE)	31:24				LBLOVFE	LRXD	LFET	LCOL	LTXD
		23:16								
		15:8						UNRE	TXEMPTY	
		7:0	LCRCE	LSFE	OVRE				TXRDY	RXRDY
0x0214	FLEX_US_CSR	31:24								MANE
		23:16	CTS	CMP			CTSIC			
		15:8			NACK			ITER	TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY
0x0214	FLEX_US_CSR (LIN_MODE)	31:24	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
		23:16	LINBLS							
		15:8	LINTC	LINID	LINBK				TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE				TXRDY	RXRDY
0x0214	FLEX_US_CSR (LON_MODE)	31:24				LBLOVFE	LRXD	LFET	LCOL	LTXD
		23:16								
		15:8						UNRE	TXEMPTY	
		7:0	LCRCE	LSFE	OVRE				TXRDY	RXRDY
0x0218	FLEX_US_RHR	31:24								
		23:16								
		15:8	RXSYNH							RXCHR[8]
		7:0	RXCHR[7:0]							
0x0218	FLEX_US_RHR (FIFO_MULTI_DATA)	31:24	RXCHR3[7:0]							
		23:16	RXCHR2[7:0]							
		15:8	RXCHR1[7:0]							
		7:0	RXCHR0[7:0]							
0x021C	FLEX_US_THR	31:24								
		23:16								
		15:8	TXSYNH							TXCHR[8]
		7:0	TXCHR[7:0]							
0x021C	FLEX_US_THR (FIFO_MULTI_DATA)	31:24	TXCHR3[7:0]							
		23:16	TXCHR2[7:0]							
		15:8	TXCHR1[7:0]							
		7:0	TXCHR0[7:0]							
0x0220	FLEX_US_BRGR	31:24								
		23:16						FP[2:0]		
		15:8	CD[15:8]							
		7:0	CD[7:0]							
0x0224	FLEX_US_RTOR	31:24								
		23:16								TO[16]
		15:8	TO[15:8]							
		7:0	TO[7:0]							
0x0228	FLEX_US_TTGR	31:24								
		23:16								
		15:8								
		7:0	TG[7:0]							
0x0228	FLEX_US_TTGR (LON_MODE)	31:24								
		23:16	PCYCLE[23:16]							
		15:8	PCYCLE[15:8]							
		7:0	PCYCLE[7:0]							
0x022C ... 0x023F	Reserved									

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x0240	FLEX_US_FIDI	31:24								
		23:16								
		15:8	FI_DI_RATIO[15:8]							
		7:0	FI_DI_RATIO[7:0]							
0x0240	FLEX_US_FIDI (LON_MODE)	31:24								
		23:16	BETA2[23:16]							
		15:8	BETA2[15:8]							
		7:0	BETA2[7:0]							
0x0244	FLEX_US_NER	31:24								
		23:16								
		15:8								
		7:0	NB_ERRORS[7:0]							
0x0248 ... 0x024B	Reserved									
0x024C	FLEX_US_IF	31:24								
		23:16								
		15:8								
		7:0	IRDA_FILTER[7:0]							
0x0250	FLEX_US_MAN	31:24	RXIDLEV	DRIFT	ONE	RX_MPOL			RX_PP[1:0]	
		23:16					RX_PL[3:0]			
		15:8				TX_MPOL			TX_PP[1:0]	
		7:0					TX_PL[3:0]			
0x0254	FLEX_US_LINMR	31:24								
		23:16							SYNCDIS	PDCM
		15:8	DLC[7:0]							
		7:0	WKUPTYP	FSDIS	DLM	CHKTYP	CHKDIS	PARDIS	NACT[1:0]	
0x0258	FLEX_US_LINIR	31:24								
		23:16								
		15:8								
		7:0	IDCHR[7:0]							
0x025C	FLEX_US_LINBRR	31:24								
		23:16						LINFP[2:0]		
		15:8	LINCD[15:8]							
		7:0	LINCD[7:0]							
0x0260	FLEX_US_LONMR	31:24								
		23:16	EOFS[7:0]							
		15:8								
		7:0			LCDS	DMAM	CDTAIL	TCOL	COLDET	COMMT
0x0264	FLEX_US_LONPR	31:24								
		23:16								
		15:8			LONPL[13:8]					
		7:0	LONPL[7:0]							
0x0268	FLEX_US_LONDL	31:24								
		23:16								
		15:8								
		7:0	LONDL[7:0]							
0x026C	FLEX_US_LONL2HDR	31:24								
		23:16								
		15:8								
		7:0	PB	ALTP	BLI[5:0]					
0x0270	FLEX_US_LONBL	31:24								
		23:16								
		15:8								
		7:0			LONBL[5:0]					
0x0274	FLEX_US_LONB1TX	31:24								
		23:16	BETA1TX[23:16]							
		15:8	BETA1TX[15:8]							
		7:0	BETA1TX[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x0278	FLEX_US_LONB1RX	31:24								
		23:16	BETA1RX[23:16]							
		15:8	BETA1RX[15:8]							
		7:0	BETA1RX[7:0]							
0x027C	FLEX_US_LONPRIO	31:24								
		23:16								
		15:8		NPS[6:0]						
		7:0		PSNB[6:0]						
0x0280	FLEX_US_IDTTX	31:24								
		23:16	IDTTX[23:16]							
		15:8	IDTTX[15:8]							
		7:0	IDTTX[7:0]							
0x0284	FLEX_US_IDTRX	31:24								
		23:16	IDTRX[23:16]							
		15:8	IDTRX[15:8]							
		7:0	IDTRX[7:0]							
0x0288	FLEX_US_ICDIFF	31:24								
		23:16								
		15:8								
		7:0					ICDIFF[3:0]			
0x028C ... 0x028F	Reserved									
0x0290	FLEX_US_CMPR	31:24								VAL2[8]
		23:16	VAL2[7:0]							
		15:8		CMPPAR	CMPMODE[1:0]					VAL1[8]
		7:0	VAL1[7:0]							
0x0294 ... 0x029F	Reserved									
0x02A0	FLEX_US_FMR	31:24			RXFTHRES2[5:0]					
		23:16			RXFTHRES[5:0]					
		15:8			TXFTHRES[5:0]					
		7:0	FRTSC		RXRDYM[1:0]				TXRDYM[1:0]	
0x02A4	FLEX_US_FLR	31:24								
		23:16			RXFL[5:0]					
		15:8								
		7:0			TXFL[5:0]					
0x02A8	FLEX_US_FIER	31:24								
		23:16								
		15:8							RXFTHF2	
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x02AC	FLEX_US_FIDR	31:24								
		23:16								
		15:8							RXFTHF2	
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x02B0	FLEX_US_FIMR	31:24								
		23:16								
		15:8							RXFTHF2	
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x02B4	FLEX_US_FESR	31:24								
		23:16								
		15:8							RXFTHF2	TXFLOCK
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x02B8 ... 0x02E3	Reserved									

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x02E4	FLEX_US_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPEN
0x02E8	FLEX_US_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS
0x02EC ... 0x03FF	Reserved									
0x0400	FLEX_SPI_CR	31:24	FIFODIS	FIFOEN						LASTXFER
		23:16							RXFCLR	TXFCLR
		15:8				REQCLR				
		7:0	SWRST						SPIDIS	SPIEN
0x0404	FLEX_SPI_MR	31:24	DLYBCS[7:0]							
		23:16					PCS[3:0]			
		15:8				CMPMODE				
		7:0	LLB		WDRBT	MODFDIS	BRSRCCLK	PCSDEC	PS	MSTR
0x0408	FLEX_SPI_RDR	31:24								
		23:16					PCS[3:0]			
		15:8	RD[15:8]							
		7:0	RD[7:0]							
0x0408	FLEX_SPI_RDR (FIFO_MULTI_DATA_8)	31:24	RD3[7:0]							
		23:16	RD2[7:0]							
		15:8	RD1[7:0]							
		7:0	RD0[7:0]							
0x0408	FLEX_SPI_RDR (FIFO_MULTI_DATA_16)	31:24	RD1[15:8]							
		23:16	RD1[7:0]							
		15:8	RD0[15:8]							
		7:0	RD0[7:0]							
0x040C	FLEX_SPI_TDR	31:24								LASTXFER
		23:16					PCS[3:0]			
		15:8	TD[15:8]							
		7:0	TD[7:0]							
0x040C	FLEX_SPI_TDR (FIFO_MULTI_DATA)	31:24	TD1[15:8]							
		23:16	TD1[7:0]							
		15:8	TD0[15:8]							
		7:0	TD0[7:0]							
0x0410	FLEX_SPI_SR	31:24	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
		23:16								SPIENS
		15:8				SFERR	CMP	UNDES	TXEMPTY	NSSR
		7:0					OVRES	MODF	TDRE	RDRF
0x0414	FLEX_SPI_IER	31:24	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
		23:16								
		15:8					CMP	UNDES	TXEMPTY	NSSR
		7:0					OVRES	MODF	TDRE	RDRF
0x0418	FLEX_SPI_IDR	31:24	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
		23:16								
		15:8					CMP	UNDES	TXEMPTY	NSSR
		7:0					OVRES	MODF	TDRE	RDRF
0x041C	FLEX_SPI_IMR	31:24	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
		23:16								
		15:8					CMP	UNDES	TXEMPTY	NSSR
		7:0					OVRES	MODF	TDRE	RDRF
0x0420 ... 0x042F	Reserved									

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x0430	FLEX_SPI_CSR0	31:24	DLYBCT[7:0]							
		23:16	DLYBS[7:0]							
		15:8	SCBR[7:0]							
		7:0	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL
0x0434	FLEX_SPI_CSR1	31:24	DLYBCT[7:0]							
		23:16	DLYBS[7:0]							
		15:8	SCBR[7:0]							
		7:0	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL
0x0438	FLEX_SPI_CSR2	31:24	DLYBCT[7:0]							
		23:16	DLYBS[7:0]							
		15:8	SCBR[7:0]							
		7:0	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL
0x043C	FLEX_SPI_CSR3	31:24	DLYBCT[7:0]							
		23:16	DLYBS[7:0]							
		15:8	SCBR[7:0]							
		7:0	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL
0x0440	FLEX_SPI_FMR	31:24			RXFTHRES[5:0]					
		23:16			TXFTHRES[5:0]					
		15:8								
		7:0			RXRDYM[1:0]				TXRDYM[1:0]	
0x0444	FLEX_SPI_FLR	31:24								
		23:16			RXFL[5:0]					
		15:8								
		7:0			TXFL[5:0]					
0x0448	FLEX_SPI_CMPR	31:24	VAL2[15:8]							
		23:16	VAL2[7:0]							
		15:8	VAL1[15:8]							
		7:0	VAL1[7:0]							
0x044C ... 0x04E3	Reserved									
0x04E4	FLEX_SPI_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPEN
0x04E8	FLEX_SPI_WPSR	31:24								
		23:16								
		15:8	WPVSRC[7:0]							
		7:0								WPVS
0x04EC ... 0x05FF	Reserved									
0x0600	FLEX_TWI_CR	31:24			FIFODIS	FIFOEN		LOCKCLR		THRCLR
		23:16							ACMDIS	ACMEN
		15:8	CLEAR	PECRQ	PECDIS	PECEN	SMBDIS	SMBEN	HSDIS	HSEN
		7:0	SWRST	QUICK	SVDIS	SVEN	MSDIS	MSEN	STOP	START
0x0600	FLEX_TWI_CR (FIFO_ENABLED)	31:24			FIFODIS	FIFOEN		TXFLCLR	RXFCLR	TXFCLR
		23:16							ACMDIS	ACMEN
		15:8	CLEAR	PECRQ	PECDIS	PECEN	SMBDIS	SMBEN	HSDIS	HSEN
		7:0	SWRST	QUICK	SVDIS	SVEN	MSDIS	MSEN	STOP	START
0x0604	FLEX_TWI_MMR	31:24								NOAP
		23:16		DADR[6:0]						
		15:8				MREAD			IADRSZ[1:0]	
		7:0								
0x0608	FLEX_TWI_SMR	31:24								
		23:16		SADR[6:0]						
		15:8		MASK[6:0]						
		7:0	SNIFF	SCLWSDIS	BSEL	SADAT	SMHH	SMDA		NACKEN

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x060C	FLEX_TWI_IADR	31:24								
		23:16	IADR[23:16]							
		15:8	IADR[15:8]							
		7:0	IADR[7:0]							
0x0610	FLEX_TWI_CWGR	31:24			HOLD[5:0]					
		23:16				BRSRCCLK		CKDIV[2:0]		
		15:8	CHDIV[7:0]							
		7:0	CLDIV[7:0]							
0x0614 ... 0x061F	Reserved									
0x0620	FLEX_TWI_SR	31:24						SR	SDA	SCL
		23:16	LOCK		SMBHHM	SMBDAM	PECERR	TOUT		MCACK
		15:8					EOSACC	SCLWS	ARBLST	NACK
		7:0	UNRE	OVRE	GACC	SVACC	SVREAD	TXRDY	RXRDY	TXCOMP
0x0620	FLEX_TWI_SR (FIFO_ENABLED)	31:24						SR	SDA	SCL
		23:16	TXFLOCK		SMBHHM	SMBDAM	PECERR	TOUT		MCACK
		15:8					EOSACC	SCLWS	ARBLST	NACK
		7:0	UNRE	OVRE	GACC	SVACC	SVREAD	TXRDY	RXRDY	TXCOMP
0x0624	FLEX_TWI_IER	31:24								
		23:16			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
		15:8	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK
		7:0	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
0x0628	FLEX_TWI_IDR	31:24								
		23:16			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
		15:8	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK
		7:0	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
0x062C	FLEX_TWI_IMR	31:24								
		23:16			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
		15:8	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK
		7:0	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
0x0630	FLEX_TWI_RHR	31:24								
		23:16								
		15:8				ASTATE[1:0]		PSTATE	SSTATE[1:0]	
		7:0	RXDATA[7:0]							
0x0630	FLEX_TWI_RHR (FIFO_ENABLED)	31:24	RXDATA3[7:0]							
		23:16	RXDATA2[7:0]							
		15:8	RXDATA1[7:0]							
		7:0	RXDATA0[7:0]							
0x0634	FLEX_TWI_THR	31:24								
		23:16								
		15:8								
		7:0	TXDATA[7:0]							
0x0634	FLEX_TWI_THR (FIFO_ENABLED)	31:24	TXDATA3[7:0]							
		23:16	TXDATA2[7:0]							
		15:8	TXDATA1[7:0]							
		7:0	TXDATA0[7:0]							
0x0638	FLEX_TWI_SMBTR	31:24	THMAX[7:0]							
		23:16	TLOWM[7:0]							
		15:8	TLOWS[7:0]							
		7:0					PRESC[3:0]			
0x063C ... 0x063F	Reserved									
0x0640	FLEX_TWI_ACR	31:24							NPEC	NDIR
		23:16	NDATAL[7:0]							
		15:8							PEC	DIR
		7:0	DATAL[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x0644	FLEX_TWI_FILTR	31:24								
		23:16								
		15:8						THRES[2:0]		
		7:0							PADFEN	FILT
0x0648 ... 0x064F	Reserved									
0x0650	FLEX_TWI_FMR	31:24			RXFTHRES[5:0]					
		23:16			TXFTHRES[5:0]					
		15:8								
		7:0			RXRDYM[1:0]				TXRDYM[1:0]	
0x0654	FLEX_TWI_FLR	31:24								
		23:16			RXFL[5:0]					
		15:8								
		7:0			TXFL[5:0]					
0x0658 ... 0x065F	Reserved									
0x0660	FLEX_TWI_FSR	31:24								
		23:16								
		15:8								
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x0664	FLEX_TWI_FIER	31:24								
		23:16								
		15:8								
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x0668	FLEX_TWI_FIDR	31:24								
		23:16								
		15:8								
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x066C	FLEX_TWI_FIMR	31:24								
		23:16								
		15:8								
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x0670 ... 0x06E3	Reserved									
0x06E4	FLEX_TWI_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPEN
0x06E8	FLEX_TWI_WPSR	31:24	WPVSRC[23:16]							
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

Downloaded from Arrow.com.

45.10.1 FLEXCOM Mode Register

Name: FLEX_MR
Offset: 0x000
Reset: 0x00000001
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							OPMODE[1:0]	
Access							R/W	R/W
Reset							0	1

Bits 1:0 – OPMODE[1:0] FLEXCOM Operating Mode

Value	Name	Description
0	NO_COM	No communication
1	USART	All UART-related protocols are selected (RS232, RS485, IrDA, ISO7816, LIN, LON) SPI/TWI-related registers are not accessible and have no impact on IOs.
2	SPI	SPI operating mode is selected. USART/TWI related registers are not accessible and have no impact on IOs.
3	TWI	All TWI-related protocols are selected (TWI, SMBus). USART/SPI-related registers are not accessible and have no impact on IOs.

Downloaded from Arrow.com.

45.10.2 FLEXCOM Receive Holding Register

Name: FLEX_RHR
Offset: 0x010
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RXDATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RXDATA[15:0] Receive Data
This register is a mirror of:
- USART Receive Holding Register (FLEX_US_RHR) if FLEX_MR.OPMODE field equals 1
- SPI Receive Data Register (FLEX_SPI_RDR) if FLEX_MR.OPMODE field equals 2
- TWI Transmit Holding Register (FLEX_TWI_RHR) if FLEX_MR.OPMODE field equals 3

Downloaded from Arrow.com.

45.10.3 FLEXCOM Transmit Holding Register

Name: FLEX_THR
Offset: 0x020
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TXDATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXDATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TXDATA[15:0] Transmit Data
This register is a mirror of:
- USART Transmit Holding Register (FLEX_US_THR) if FLEX_MR.OPMODE field equals 1
- SPI Transmit Data Register (FLEX_SPI_TDR) if FLEX_MR.OPMODE field equals 2
- TWI Transmit Holding Register (FLEX_TWI_THR) if FLEX_MR.OPMODE field equals 3

Downloaded from Arrow.com.

45.10.4 USART Control Register

Name: FLEX_US_CR
Offset: 0x200
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	FIFODIS	FIFOEN		REQCLR		TXFLCLR	RXFCLR	TXFCLR
Access	W	W		W		W	W	W
Reset	–	–		–		–	–	–

Bit	23	22	21	20	19	18	17	16
			LINWKUP	LINABT	RTSDIS	RTSEN		
Access			W	W	W	W		
Reset			–	–	–	–		

Bit	15	14	13	12	11	10	9	8
	RETTO	RSTNACK	RSTIT	SENDA	STTTO	STPBRK	STTBRK	RSTSTA
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX		
Access	W	W	W	W	W	W		
Reset	–	–	–	–	–	–		

Bit 31 – FIFODIS FIFO Disable

Value	Description
0	No effect.
1	Disables the Transmit and Receive FIFOs.

Bit 30 – FIFOEN FIFO Enable

Value	Description
0	No effect.
1	Enables the Transmit and Receive FIFOs.

Bit 28 – REQCLR Request to Clear the Comparison Trigger

Value	Description
0	No effect.
1	Restarts the comparison trigger to enable FLEX_US_RHR loading.

Bit 26 – TXFLCLR Transmit FIFO Lock CLEAR

Value	Description
0	No effect.
1	Clears the Transmit FIFO Lock.

Bit 25 – RXFCLR Receive FIFO Clear

Value	Description
0	No effect.
1	Empties the Receive FIFO.

Bit 24 – TXFCLR Transmit FIFO Clear

Downloaded from Arrow.com.

Value	Description
0	No effect.
1	Empties the Transmit FIFO.

Bit 21 – LINWKUP Send LIN Wakeup Signal

Value	Description
0	No effect:
1	Sends a wakeup signal on the LIN bus.

Bit 20 – LINABT Abort LIN Transmission

Value	Description
0	No effect.
1	Aborts the current LIN transmission.

Bit 19 – RTSDIS Request to Send Disable

Value	Description
0	No effect.
1	Drives the RTS pin to 0 if FLEX_US_MR.USART_MODE field = 2, else drives the RTS pin to 1 if FLEX_US_MR.USART_MODE field = 0.

Bit 18 – RTSEN Request to Send Enable

Value	Description
0	No effect.
1	Drives the RTS pin to 1 if FLEX_US_MR.USART_MODE field = 2, else drives the RTS pin to 0 if FLEX_US_MR.USART_MODE field = 0.

Bit 15 – RETTO Start Timeout Immediately

Value	Description
0	No effect
1	Immediately restarts timeout period.

Bit 14 – RSTNACK Reset Non Acknowledge

Value	Description
0	No effect
1	Resets FLEX_US_CSR.NACK.

Bit 13 – RSTIT Reset Iterations

Value	Description
0	No effect.
1	Resets FLEX_US_CSR.ITER. No effect if the ISO7816 is not enabled.

Bit 12 – SENDA Send Address

Value	Description
0	No effect.
1	In Multidrop mode only, the next character written to FLEX_US_THR is sent with the address bit set.

Bit 11 – STTTO Clear TIMEOUT Flag and Start Timeout After Next Character Received

Value	Description
0	No effect.
1	Starts waiting for a character before clocking the timeout counter. Immediately disables a timeout period in progress. Resets the FLEX_US_CSR.TIMEOUT status bit.

Bit 10 – STPBRK Stop Break

Value	Description
0	No effect.
1	Stops transmission of the break after a minimum of one character length and transmits a high level during 12-bit periods. No effect if no break is being transmitted.

Downloaded from Arrow.com.

Bit 9 – STTBRK Start Break

Value	Description
0	No effect.
1	Starts transmission of a break after the characters present in FLEX_US_THR and the Transmit Shift Register have been transmitted. No effect if a break is already being transmitted.

Bit 8 – RSTSTA Reset Status Bits

Value	Description
0	No effect.
1	Resets the PARE, FRAME, OVRE, MANE, LINBE, LINISFE, LINIPE, LINCE, LINSNRE, LINSTE, LINHTE, LINID, LINTC, LINBK, CMP and RXBRK in FLEX_US_CSR status bits, as well as the TXFEF, TXFFF, TXFTHF, RXFEF, RXFFF, RXFTHF, TXFPTEF, RXFPTEF in FLEX_US_FESR status bits.

Bit 7 – TXDIS Transmitter Disable

Value	Description
0	No effect.
1	Disables the transmitter.

Bit 6 – TXEN Transmitter Enable

Value	Description
0	No effect.
1	Enables the transmitter if TXDIS is 0.

Bit 5 – RXDIS Receiver Disable

Value	Description
0	No effect.
1	Disables the receiver.

Bit 4 – RXEN Receiver Enable

Value	Description
0	No effect.
1	Enables the receiver, if RXDIS is 0.

Bit 3 – RSTTX Reset Transmitter

Value	Description
0	No effect.
1	Resets the transmitter.

Bit 2 – RSTRX Reset Receiver

Value	Description
0	No effect.
1	Resets the receiver.

Downloaded from Arrow.com.

45.10.5 USART Mode Register

Name: FLEX_US_MR
Offset: 0x204
Reset: 0xC0000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	ONEBIT	MODSYNC	MAN	FILTER		MAX_ITERATION[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	–	–	–	–		–	–	–

Bit	23	22	21	20	19	18	17	16
	INVDATA	VAR_SYNC	DSNACK	INACK	OVER	CLKO	MODE9	MSBF
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	CHMODE[1:0]		NBSTOP[1:0]		PAR[2:0]			SYNC
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	CHRL[1:0]		USCLKS[1:0]		USART_MODE[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bit 31 – ONEBIT Start Frame Delimiter Selector

Value	Description
0	Start frame delimiter is COMMAND or DATA SYNC.
1	Start frame delimiter is one bit.

Bit 30 – MODSYNC Manchester Synchronization Mode

Value	Description
0	The Manchester start bit is a 0 to 1 transition
1	The Manchester start bit is a 1 to 0 transition.

Bit 29 – MAN Manchester Encoder/Decoder Enable

Value	Description
0	Manchester encoder/decoder are disabled.
1	Manchester encoder/decoder are enabled.

Bit 28 – FILTER Receive Line Filter

Value	Description
0	The USART does not filter the receive line.
1	The USART filters the receive line using a three-sample filter (1/16-bit clock) (2 over 3 majority).

Bits 26:24 – MAX_ITERATION[2:0] Maximum Number of Automatic Iterations

Value	Description
0–7	Defines the maximum number of iterations in mode ISO7816, protocol T = 0.

Bit 23 – INVDATA Inverted Data

Value	Description
0	The data field transmitted on TXD line is the same as the one written in FLEX_US_THR or the content read in FLEX_US_RHR is the same as RXD line. Normal mode of operation.

Downloaded from Arrow.com.

Value	Description
1	The data field transmitted on TXD line is inverted (voltage polarity only) compared to the value written in FLEX_US_THR or the content read in FLEX_US_RHR is inverted compared to what is received on RXD line (or ISO7816 IO line). Inverted mode of operation, useful for contactless card application. To be used with configuration bit MSBF.

Bit 22 – VAR_SYNC Variable Synchronization of Command/Data Sync Start Frame Delimiter

Value	Description
0	User defined configuration of command or data sync field depending on MODSYNC value.
1	The sync field is updated when a character is written into FLEX_US_THR.

Bit 21 – DSNACK Disable Successive NACK
The MAX_ITERATION field must be cleared if DSNACK is cleared.

Value	Description
0	NACK is sent on the ISO line as soon as a parity error occurs in the received character (unless INACK is set).
1	Successive parity errors are counted up to the value specified in the MAX_ITERATION field. These parity errors generate a NACK on the ISO line. As soon as this value is reached, no additional NACK is sent on the ISO line. The flag ITER is asserted.

Bit 20 – INACK Inhibit Non Acknowledge

Value	Description
0	The NACK is generated.
1	The NACK is not generated.

Bit 19 – OVER Oversampling Mode

Value	Description
0	16x Oversampling.
1	8x Oversampling.

Bit 18 – CLKO Clock Output Select

Value	Description
0	The USART does not drive the SCK pin (Synchronous Slave mode or Asynchronous mode with external baud rate clock source).
1	The USART drives the SCK pin if USCLKS does not select the external clock SCK (USART Synchronous Master mode).

Bit 17 – MODE9 9-bit Character Length

Value	Description
0	CHRL defines character length.
1	9-bit character length.

Bit 16 – MSBF Bit Order

Value	Description
0	Least significant bit is sent/received first.
1	Most significant bit is sent/received first.

Bits 15:14 – CHMODE[1:0] Channel Mode

Value	Name	Description
0	NORMAL	Normal mode
1	AUTOMATIC	Automatic Echo. Receiver input is connected to the TXD pin.
2	LOCAL_LOOPBACK	Local Loopback. Transmitter output is connected to the Receiver Input.
3	REMOTE_LOOPBACK	Remote Loopback. RXD pin is internally connected to the TXD pin.

Bits 13:12 – NBSTOP[1:0] Number of Stop Bits

Value	Name	Description
0	1_BIT	1 stop bit

Downloaded from Arrow.com.

Value	Name	Description
1	1_5_BIT	1.5 stop bit (SYNC = 0) or reserved (SYNC = 1)
2	2_BIT	2 stop bits

Bits 11:9 – PAR[2:0] Parity Type

Value	Name	Description
0	EVEN	Even parity
1	ODD	Odd parity
2	SPACE	Parity forced to 0 (Space)
3	MARK	Parity forced to 1 (Mark)
4	NO	No parity
6	MULTIDROP	Multidrop mode

Bit 8 – SYNC Synchronous Mode Select

Value	Description
0	USART operates in Asynchronous mode (UART).
1	USART operates in Synchronous mode.

Bits 7:6 – CHRL[1:0] Character Length

Value	Name	Description
0	5_BIT	Character length is 5 bits
1	6_BIT	Character length is 6 bits
2	7_BIT	Character length is 7 bits
3	8_BIT	Character length is 8 bits

Bits 5:4 – USCLKS[1:0] Clock Selection

Value	Name	Description
0	MCK	Peripheral clock is selected
1	DIV	Peripheral clock divided (DIV = 8) is selected
2	GCLK	PMC generic clock is selected. If the SCK pin is driven (CLKO = 1), the CD field must be greater than 1.
3	SCK	External pin SCK is selected

Bits 3:0 – USART_MODE[3:0] USART Mode of Operation

Value	Name	Description
0x0	NORMAL	Normal mode
0x1	RS485	RS485
0x2	HW_HANDSHAKING	Hardware handshaking
0x3	MODEM	Modem
0x4	IS07816_T_0	IS07816 Protocol: T = 0
0x6	IS07816_T_1	IS07816 Protocol: T = 1
0x8	IRDA	IrDA
0x9	LON	LON
0xA	LIN_MASTER	LIN Master mode
0xB	LIN_SLAVE	LIN Slave mode
0xC	DATA16BIT_MASTER	16-bit data master
0xD	DATA16BIT_SLAVE	16-bit data slave
0xE	SPI_MASTER	SPI Master mode (CLKO must be written to 1 and USCLKS = 0, 1 or 2)
0xF	SPI_SLAVE	SPI Slave mode

Downloaded from Arrow.com.

45.10.6 USART Interrupt Enable Register

Name: FLEX_US_IER
Offset: 0x208
Reset: –
Property: Write-only

For LIN-specific configurations, see USART Interrupt Enable Register (LIN_MODE).

For LON-specific configurations, see USART Interrupt Enable Register (LON_MODE).

This register can only be written if the WPITEN bit is cleared in the USART Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
								MANE
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
		CMP			CTSIC			
Access		W			W			
Reset		–			–			

Bit	15	14	13	12	11	10	9	8
			NACK			ITER	TXEMPTY	TIMEOUT
Access			W			W	W	W
Reset			–			–	–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY
Access	W	W	W			W	W	W
Reset	–	–	–			–	–	–

Bit 24 – MANE Manchester Error Interrupt Enable

Bit 22 – CMP Comparison Interrupt Enable

Bit 19 – CTSIC Clear to Send Input Change Interrupt Enable

Bit 13 – NACK Non Acknowledge Interrupt Enable

Bit 10 – ITER Max number of Repetitions Reached Interrupt Enable

Bit 9 – TXEMPTY TXEMPTY Interrupt Enable

Bit 8 – TIMEOUT Timeout Interrupt Enable

Bit 7 – PARE Parity Error Interrupt Enable

Bit 6 – FRAME Framing Error Interrupt Enable

Bit 5 – OVRE Overrun Error Interrupt Enable

Downloaded from Arrow.com.

Bit 2 – RXBRK Receiver Break Interrupt Enable

Bit 1 – TXRDY TXRDY Interrupt Enable

Bit 0 – RXRDY RXRDY Interrupt Enable

Downloaded from Arrow.com.

45.10.7 USART Interrupt Enable Register (LIN_MODE)

Name: FLEX_US_IER (LIN_MODE)
Offset: 0x208
Reset: –
Property: Write-only

This configuration is relevant only if USART_MODE = 0xA or 0xB in the USART Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
Access	W	W	W	W	W	W	W	
Reset	–	–	–	–	–	–	–	

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LINTC	LINID	LINBK				TXEMPTY	TIMEOUT
Access	W	W	W				W	W
Reset	–	–	–				–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE				TXRDY	RXDRY
Access	W	W	W				W	W
Reset	–	–	–				–	–

Bit 31 – LINHTE LIN Header Timeout Error Interrupt Enable

Bit 30 – LINSTE LIN Synch Tolerance Error Interrupt Enable

Bit 29 – LINSNRE LIN Slave Not Responding Error Interrupt Enable

Bit 28 – LINCE LIN Checksum Error Interrupt Enable

Bit 27 – LINIPE LIN Identifier Parity Interrupt Enable

Bit 26 – LINISFE LIN Inconsistent Synch Field Error Interrupt Enable

Bit 25 – LINBE LIN Bus Error Interrupt Enable

Bit 15 – LINTC LIN Transfer Completed Interrupt Enable

Bit 14 – LINID LIN Identifier Sent or LIN Identifier Received Interrupt Enable

Bit 13 – LINBK LIN Break Sent or LIN Break Received Interrupt Enable

Bit 9 – TXEMPTY TXEMPTY Interrupt Enable

Bit 8 – TIMEOUT Timeout Interrupt Enable

Downloaded from Arrow.com.

Bit 7 – PARE Parity Error Interrupt Enable

Bit 6 – FRAME Framing Error Interrupt Enable

Bit 5 – OVRE Overrun Error Interrupt Enable

Bit 1 – TXRDY TXRDY Interrupt Enable

Bit 0 – RXRDY RXRDY Interrupt Enable

Downloaded from Arrow.com.

45.10.8 USART Interrupt Enable Register (LON_MODE)

Name: FLEX_US_IER (LON_MODE)
Offset: 0x208
Reset: –
Property: Write-only

This register is relevant only if USART_MODE = 0x9 in the USART Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
				LBLOVFE	LRXD	LFET	LCOL	LTXD
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						UNRE	TXEMPTY	
Access						W	W	
Reset						–	–	

Bit	7	6	5	4	3	2	1	0
	LCRCE	LSFE	OVRE				TXRDY	RXRDY
Access	W	W	W				W	W
Reset	–	–	–				–	–

Bit 28 – LBLOVFE LON Backlog Overflow Error Interrupt Enable

Bit 27 – LRXD LON Reception Done Interrupt Enable

Bit 26 – LFET LON Frame Early Termination Interrupt Enable

Bit 25 – LCOL LON Collision Interrupt Enable

Bit 24 – LTXD LON Transmission Done Interrupt Enable

Bit 10 – UNRE Underrun Error Interrupt Enable

Bit 9 – TXEMPTY TXEMPTY Interrupt Enable

Bit 7 – LCRCE LON CRC Error Interrupt Enable

Bit 6 – LSFE LON Short Frame Error Interrupt Enable

Bit 5 – OVRE Overrun Error Interrupt Enable

Bit 1 – TXRDY TXRDY Interrupt Enable

Bit 0 – RXRDY RXRDY Interrupt Enable

Downloaded from Arrow.com.

45.10.9 USART Interrupt Disable Register

Name: FLEX_US_IDR
Offset: 0x20C
Reset: –
Property: Write-only

For LIN-specific configurations, see USART Interrupt Disable Register (LIN_MODE).

For LON-specific configurations, see USART Interrupt Disable Register (LON_MODE).

This register can only be written if the WPITEN bit is cleared in the USART Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
								MANE
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
		CMP			CTSIC			
Access		W			W			
Reset		–			–			

Bit	15	14	13	12	11	10	9	8
			NACK			ITER	TXEMPTY	TIMEOUT
Access			W			W	W	W
Reset			–			–	–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY
Access	W	W	W			W	W	W
Reset	–	–	–			–	–	–

Bit 24 – MANE Manchester Error Interrupt Disable

Bit 22 – CMP Comparison Interrupt Disable

Bit 19 – CTSIC Clear to Send Input Change Interrupt Disable

Bit 13 – NACK Non Acknowledge Interrupt Disable

Bit 10 – ITER Max Number of Repetitions Reached Interrupt Disable

Bit 9 – TXEMPTY TXEMPTY Interrupt Disable

Bit 8 – TIMEOUT Timeout Interrupt Disable

Bit 7 – PARE Parity Error Interrupt Disable

Bit 6 – FRAME Framing Error Interrupt Disable

Bit 5 – OVRE Overrun Error Interrupt Disable

Downloaded from Arrow.com.

Bit 2 – RXBRK Receiver Break Interrupt Disable

Bit 1 – TXRDY TXRDY Interrupt Disable

Bit 0 – RXRDY RXRDY Interrupt Disable

Downloaded from Arrow.com.

45.10.10 USART Interrupt Disable Register (LIN_MODE)

Name: FLEX_US_IDR (LIN_MODE)
Offset: 0x20C
Reset: –
Property: Write-only

This configuration is relevant only if USART_MODE = 0xA or 0xB in the USART Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
Access	W	W	W	W	W	W	W	
Reset	–	–	–	–	–	–	–	

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LINTC	LINID	LINBK				TXEMPTY	TIMEOUT
Access	W	W	W				W	W
Reset	–	–	–				–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE				TXRDY	RXDY
Access	W	W	W				W	W
Reset	–	–	–				–	–

Bit 31 – LINHTE LIN Header Timeout Error Interrupt Disable

Bit 30 – LINSTE LIN Synch Tolerance Error Interrupt Disable

Bit 29 – LINSNRE LIN Slave Not Responding Error Interrupt Disable

Bit 28 – LINCE LIN Checksum Error Interrupt Disable

Bit 27 – LINIPE LIN Identifier Parity Interrupt Disable

Bit 26 – LINISFE LIN Inconsistent Synch Field Error Interrupt Disable

Bit 25 – LINBE LIN Bus Error Interrupt Disable

Bit 15 – LINTC LIN Transfer Completed Interrupt Disable

Bit 14 – LINID LIN Identifier Sent or LIN Identifier Received Interrupt Disable

Bit 13 – LINBK LIN Break Sent or LIN Break Received Interrupt Disable

Bit 9 – TXEMPTY TXEMPTY Interrupt Disable

Bit 8 – TIMEOUT Timeout Interrupt Disable

Downloaded from Arrow.com.

Bit 7 – PARE Parity Error Interrupt Disable

Bit 6 – FRAME Framing Error Interrupt Disable

Bit 5 – OVRE Overrun Error Interrupt Disable

Bit 1 – TXRDY TXRDY Interrupt Disable

Bit 0 – RXRDY RXRDY Interrupt Disable

Downloaded from Arrow.com.

45.10.11 USART Interrupt Disable Register (LON_MODE)

Name: FLEX_US_IDR (LON_MODE)
Offset: 0x20C
Reset: –
Property: Write-only

This configuration is relevant only if USART_MODE = 0x9 in the USART Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
				LBLOVFE	LRXD	LFET	LCOL	LTXD
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						UNRE	TXEMPTY	
Access						W	W	
Reset						–	–	

Bit	7	6	5	4	3	2	1	0
	LCRCE	LSFE	OVRE				TXRDY	RXRDY
Access	W	W	W				W	W
Reset	–	–	–				–	–

Bit 28 – LBLOVFE LON Backlog Overflow Error Interrupt Disable

Bit 27 – LRXD LON Reception Done Interrupt Disable

Bit 26 – LFET LON Frame Early Termination Interrupt Disable

Bit 25 – LCOL LON Collision Interrupt Disable

Bit 24 – LTXD LON Transmission Done Interrupt Disable

Bit 10 – UNRE Underrun Error Interrupt Disable

Bit 9 – TXEMPTY TXEMPTY Interrupt Disable

Bit 7 – LCRCE LON CRC Error Interrupt Disable

Bit 6 – LSFE LON Short Frame Error Interrupt Disable

Bit 5 – OVRE Overrun Error Interrupt Disable

Bit 1 – TXRDY TXRDY Interrupt Disable

Bit 0 – RXRDY RXRDY Interrupt Disable

Downloaded from Arrow.com.

45.10.12 USART Interrupt Mask Register

Name: FLEX_US_IMR
Offset: 0x210
Reset: 0x00000000
Property: Read-only

For LIN-specific configurations, see USART Interrupt Mask Register (LIN_MODE).

For LON-specific configurations, see USART Interrupt Mask Register (LON_MODE).

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
								MANE
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
		CMP			CTSIC			
Access		R			R			
Reset		0			0			

Bit	15	14	13	12	11	10	9	8
			NACK			ITER	TXEMPTY	TIMEOUT
Access			R			R	R	R
Reset			0			0	0	0

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY
Access	R	R	R			R	R	R
Reset	0	0	0			0	0	0

Bit 24 – MANE Manchester Error Interrupt Mask

Bit 22 – CMP Comparison Interrupt Mask

Bit 19 – CTSIC Clear to Send Input Change Interrupt Mask

Bit 13 – NACK Non Acknowledge Interrupt Mask

Bit 10 – ITER Max Number of Repetitions Reached Interrupt Mask

Bit 9 – TXEMPTY TXEMPTY Interrupt Mask

Bit 8 – TIMEOUT Timeout Interrupt Mask

Bit 7 – PARE Parity Error Interrupt Mask

Bit 6 – FRAME Framing Error Interrupt Mask

Bit 5 – OVRE Overrun Error Interrupt Mask

Bit 2 – RXBRK Receiver Break Interrupt Mask

Downloaded from Arrow.com.

Bit 1 – TXRDY TXRDY Interrupt Mask

Bit 0 – RXRDY RXRDY Interrupt Mask

Downloaded from Arrow.com.

45.10.13 USART Interrupt Mask Register (LIN_MODE)

Name: FLEX_US_IMR (LIN_MODE)
Offset: 0x210
Reset: 0x00000000
Property: Read-only

This configuration is relevant only if USART_MODE = 0xA or 0xB in the USART Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
Access	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LINTC	LINID	LINBK				TXEMPTY	TIMEOUT
Access	R	R	R				R	R
Reset	0	0	0				0	0

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE				TXRDY	RXRDY
Access	R	R	R				R	R
Reset	0	0	0				0	0

Bit 31 – LINHTE LIN Header Timeout Error Interrupt Mask

Bit 30 – LINSTE LIN Synch Tolerance Error Interrupt Mask

Bit 29 – LINSNRE LIN Slave Not Responding Error Interrupt Mask

Bit 28 – LINCE LIN Checksum Error Interrupt Mask

Bit 27 – LINIPE LIN Identifier Parity Interrupt Mask

Bit 26 – LINISFE LIN Inconsistent Synch Field Error Interrupt Mask

Bit 25 – LINBE LIN Bus Error Interrupt Mask

Bit 15 – LINTC LIN Transfer Completed Interrupt Mask

Bit 14 – LINID LIN Identifier Sent or LIN Identifier Received Interrupt Mask

Bit 13 – LINBK LIN Break Sent or LIN Break Received Interrupt Mask

Bit 9 – TXEMPTY TXEMPTY Interrupt Mask

Bit 8 – TIMEOUT Timeout Interrupt Mask

Downloaded from Arrow.com.

Bit 7 – PARE Parity Error Interrupt Mask

Bit 6 – FRAME Framing Error Interrupt Mask

Bit 5 – OVRE Overrun Error Interrupt Mask

Bit 1 – TXRDY TXRDY Interrupt Mask

Bit 0 – RXRDY RXRDY Interrupt Mask

Downloaded from Arrow.com.

45.10.14 USART Interrupt Mask Register (LON_MODE)

Name: FLEX_US_IMR (LON_MODE)
Offset: 0x210
Reset: 0x00000000
Property: Read-only

This configuration is relevant only if USART_MODE = 0x9 in the USART Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
				LBLOVFE	LRXD	LFET	LCOL	LTXD
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						UNRE	TXEMPTY	
Access						R	R	
Reset						0	0	

Bit	7	6	5	4	3	2	1	0
	LCRCE	LSFE	OVRE				TXRDY	RXRDY
Access	R	R	R				R	R
Reset	0	0	0				0	0

Bit 28 – LBLOVFE LON Backlog Overflow Error Interrupt Mask

Bit 27 – LRXD LON Reception Done Interrupt Mask

Bit 26 – LFET LON Frame Early Termination Interrupt Mask

Bit 25 – LCOL LON Collision Interrupt Mask

Bit 24 – LTXD LON Transmission Done Interrupt Mask

Bit 10 – UNRE Underrun Error Interrupt Mask

Bit 9 – TXEMPTY TXEMPTY Interrupt Mask

Bit 7 – LCRCE LON CRC Error Interrupt Mask

Bit 6 – LSFE LON Short Frame Error Interrupt Mask

Bit 5 – OVRE Overrun Error Interrupt Mask

Bit 1 – TXRDY TXRDY Interrupt Mask

Bit 0 – RXRDY RXRDY Interrupt Mask

Downloaded from Arrow.com.

45.10.15 USART Channel Status Register

Name: FLEX_US_CSR
Offset: 0x214
Reset: 0x00000000
Property: Read-only

For LIN-specific configurations, see USART Channel Status Register (LIN_MODE).

For LON-specific configurations, see USART Channel Status Register (LON_MODE).

Bit	31	30	29	28	27	26	25	24
								MANE
Access								R
Reset								–

Bit	23	22	21	20	19	18	17	16
	CTS	CMP			CTSIC			
Access	R	R			R			
Reset	–	–			–			

Bit	15	14	13	12	11	10	9	8
			NACK			ITER	TXEMPTY	TIMEOUT
Access			R			R	R	R
Reset			–			–	–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY
Access	R	R	R			R	R	R
Reset	–	–	–			–	–	–

Bit 24 – MANE Manchester Error

Value	Description
0	No Manchester error has been detected since the last RSTSTA command was issued.
1	At least one Manchester error has been detected since the last RSTSTA command was issued.

Bit 23 – CTS Image of CTS Input

Value	Description
0	CTS input is driven low.
1	CTS input is driven high.

Bit 22 – CMP Comparison Status

Value	Description
0	No received character matched the comparison criteria programmed in VAL1, VAL2 fields and CMPPAR bit in since the last RSTSTA command was issued.
1	A received character matched the comparison criteria since the last RSTSTA command was issued.

Bit 19 – CTSIC Clear to Send Input Change Flag

Value	Description
0	No input change has been detected on the CTS pin since the last read of FLEX_US_CSR.
1	At least one input change has been detected on the CTS pin since the last read of FLEX_US_CSR.

Bit 13 – NACK Non Acknowledge Interrupt

Value	Description
0	Non acknowledge has not been detected since the last RSTNACK.
1	At least one non acknowledge has been detected since the last RSTNACK.

Downloaded from Arrow.com.

Bit 10 – ITER Max Number of Repetitions Reached

Value	Description
0	Maximum number of repetitions has not been reached since the last RSTIT command was issued.
1	Maximum number of repetitions has been reached since the last RSTIT command was issued.

Bit 9 – TXEMPTY Transmitter Empty (cleared by writing FLEX_US_THR)

Value	Description
0	There are characters in either FLEX_US_THR or the Transmit Shift Register, or the transmitter is disabled.
1	There are no characters in FLEX_US_THR, nor in the Transmit Shift Register.

Bit 8 – TIMEOUT Receiver Timeout

Value	Description
0	There has not been a timeout since the last Start Timeout command (FLEX_US_CR.STTTO) or the Timeout Register is 0.
1	There has been a timeout since the last Start Timeout command (FLEX_US_CR.STTTO).

Bit 7 – PARE Parity Error

Value	Description
0	No parity error has been detected since the last RSTSTA command was issued.
1	At least one parity error has been detected since the last RSTSTA command was issued.

Bit 6 – FRAME Framing Error

Value	Description
0	No stop bit has been detected low since the last RSTSTA command was issued.
1	At least one stop bit has been detected low since the last RSTSTA command was issued.

Bit 5 – OVRE Overrun Error

Value	Description
0	No overrun error has occurred since the last RSTSTA command was issued.
1	At least one overrun error has occurred since the last RSTSTA command was issued.

Bit 2 – RXBRK Break Received/End of Break

Value	Description
0	No break received or end of break detected since the last RSTSTA command was issued.
1	Break received or end of break detected since the last RSTSTA command was issued.

Bit 1 – TXRDY Transmitter Ready (cleared by writing FLEX_US_THR)
When FIFOs are disabled:
0: A character in FLEX_US_THR is waiting to be transferred to the Transmit Shift Register, or an STTBRK command has been requested, or the transmitter is disabled. As soon as the transmitter is enabled, TXRDY becomes 1.
1: There is no character in FLEX_US_THR.
When FIFOs are enabled:
0: Transmit FIFO is full and cannot accept more data.
1: Transmit FIFO is not full; one or more data can be written according to TXRDYM field configuration.
TXRDY behavior with FIFO enabled is illustrated in 45.7.11.5 TXEMPTY, TXRDY and RXRDY Behavior.

Bit 0 – RXRDY Receiver Ready (cleared by reading FLEX_US_RHR)
When FIFOs are disabled:
0: No complete character has been received since the last read of FLEX_US_RHR or the receiver is disabled. If characters were received when the receiver was disabled, RXRDY changes to 1 when the receiver is enabled.
1: At least one complete character has been received and FLEX_US_RHR has not yet been read.
When FIFOs are enabled:
0: Receive FIFO is empty; no data to read
1: At least one unread data is in the Receive FIFO
RXRDY behavior with FIFO enabled is illustrated in 45.7.11.5 TXEMPTY, TXRDY and RXRDY Behavior.

Downloaded from Arrow.com.

45.10.16 USART Channel Status Register (LIN_MODE)

Name: FLEX_US_CSR (LIN_MODE)
Offset: 0x214
Reset: –
Property: Read-only

This configuration is relevant only if USART_MODE = 0xA or 0xB in the USART Mode Register.

Bit	31	30	29	28	27	26	25	24
	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
Access	R	R	R	R	R	R	R	
Reset	–	–	–	–	–	–	–	

Bit	23	22	21	20	19	18	17	16
	LINBLS							
Access	R							
Reset	–							

Bit	15	14	13	12	11	10	9	8
	LINTC	LINID	LINBK				TXEMPTY	TIMEOUT
Access	R	R	R				R	R
Reset	–	–	–				–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE				TXRDY	RXRDY
Access	R	R	R				R	R
Reset	–	–	–				–	–

Bit 31 – LINHTE LIN Header Timeout Error

Value	Description
0	No LIN header timeout error has been detected since the last RSTSTA command was issued.
1	A LIN header timeout error has been detected since the last RSTSTA command was issued.

Bit 30 – LINSTE LIN Synch Tolerance Error

Value	Description
0	No LIN synch tolerance error has been detected since the last RSTSTA command was issued.
1	A LIN synch tolerance error has been detected since the last RSTSTA command was issued.

Bit 29 – LINSNRE LIN Slave Not Responding Error

Value	Description
0	No LIN slave not responding error has been detected since the last RSTSTA command was issued.
1	A LIN slave not responding error has been detected since the last RSTSTA command was issued.

Bit 28 – LINCE LIN Checksum Error

Value	Description
0	No LIN checksum error has been detected since the last RSTSTA command was issued.
1	A LIN checksum error has been detected since the last RSTSTA command was issued.

Bit 27 – LINIPE LIN Identifier Parity Error

Value	Description
0	No LIN identifier parity error has been detected since the last RSTSTA command was issued.
1	A LIN identifier parity error has been detected since the last RSTSTA command was issued.

Bit 26 – LINISFE LIN Inconsistent Synch Field Error

Downloaded from Arrow.com.

Value	Description
0	No LIN inconsistent synch field error has been detected since the last RSTSTA
1	The USART is configured as a slave node and a LIN Inconsistent synch field error has been detected since the last RSTSTA command was issued.

Bit 25 – LINBE LIN Bit Error

Value	Description
0	No bit error has been detected since the last RSTSTA command was issued.
1	A bit error has been detected since the last RSTSTA command was issued.

Bit 23 – LINBLS LIN Bus Line Status

Value	Description
0	LIN bus line is set to 0.
1	LIN bus line is set to 1.

Bit 15 – LINTC LIN Transfer Completed

Value	Description
0	The USART is idle or a LIN transfer is ongoing.
1	A LIN transfer has been completed since the last RSTSTA command was issued.

Bit 14 – LINID LIN Identifier Sent or LIN Identifier Received
If USART operates in LIN Master mode (USART_MODE = 0xA):
0: No LIN identifier has been sent since the last RSTSTA command was issued.
1: At least one LIN identifier has been sent since the last RSTSTA command was issued.
If USART operates in LIN Slave mode (USART_MODE = 0xB):
0: No LIN identifier has been received since the last RSTSTA command was issued.
1: At least one LIN identifier has been received since the last RSTSTA.

Bit 13 – LINBK LIN Break Sent or LIN Break Received
Applicable if USART operates in LIN Master mode (USART_MODE = 0xA):
0: No LIN break has been sent since the last RSTSTA command was issued.
1: At least one LIN break has been sent since the last RSTSTA.
If USART operates in LIN Slave mode (USART_MODE = 0xB):
0: No LIN break has received sent since the last RSTSTA command was issued.
1: At least one LIN break has been received since the last RSTSTA command was issued.

Bit 9 – TXEMPTY Transmitter Empty (cleared by writing FLEX_US_THR)

Value	Description
0	There are characters in either FLEX_US_THR or the Transmit Shift Register, or the transmitter is disabled.
1	There are no characters in FLEX_US_THR, nor in the Transmit Shift Register.

Bit 8 – TIMEOUT Receiver Timeout

Value	Description
0	There has not been a timeout since the last start timeout command (FLEX_US_CR.STTTO) or the Timeout Register is 0.
1	There has been a timeout since the last start timeout command (FLEX_US_CR.STTTO).

Bit 7 – PARE Parity Error

Value	Description
0	No parity error has been detected since the last RSTSTA command was issued.
1	At least one parity error has been detected since the last RSTSTA command was issued.

Bit 6 – FRAME Framing Error

Value	Description
0	No stop bit has been detected low since the last RSTSTA command was issued.
1	At least one stop bit has been detected low since the last RSTSTA command was issued.

Downloaded from Arrow.com.

Bit 5 – OVRE Overrun Error

Value	Description
0	No overrun error has occurred since the last RSTSTA command was issued.
1	At least one overrun error has occurred since the last RSTSTA command was issued.

Bit 1 – TXRDY Transmitter Ready (cleared by writing FLEX_US_THR)

Value	Description
0	A character in FLEX_US_THR is waiting to be transferred to the Transmit Shift Register, or the transmitter is disabled. As soon as the transmitter is enabled, TXRDY becomes 1.
1	There is no character in FLEX_US_THR.

Bit 0 – RXRDY Receiver Ready (cleared by reading FLEX_US_RHR)

Value	Description
0	No complete character has been received since the last read of FLEX_US_RHR or the receiver is disabled. If characters were being received when the receiver was disabled, RXRDY changes to 1 when the receiver is enabled.
1	At least one complete character has been received and FLEX_US_RHR has not yet been read.

Downloaded from Arrow.com.

45.10.17 USART Channel Status Register (LON_MODE)

Name: FLEX_US_CSR (LON_MODE)
Offset: 0x214
Reset: –
Property: Read-only

This configuration is relevant only if USART_MODE = 0x9 in the USART Mode Register.

Bit	31	30	29	28	27	26	25	24
				LBLOVFE	LRXD	LFET	LCOL	LTXD
Access				R	R	R	R	R
Reset				–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						UNRE	TXEMPTY	
Access						R	R	
Reset						–	–	

Bit	7	6	5	4	3	2	1	0
	LCRCE	LSFE	OVRE				TXRDY	RXRDY
Access	R	R	R				R	R
Reset	–	–	–				–	–

Bit 28 – LBLOVFE LON Backlog Overflow Error

Value	Description
0	No backlog overflow error occurred since the last RSTSTA command was issued.
1	At least one backlog error overflow occurred since the last RSTSTA command was issued.

Bit 27 – LRXD LON Reception End Flag

Value	Description
0	Reception on going or no reception occurred since the last RSTSTA command was issued.
1	At least one reception has been performed since the last RSTSTA command was issued.

Bit 26 – LFET LON Frame Early Termination

Value	Description
0	No frame has been terminated early due to collision detection since the last RSTSTA command was issued.
1	At least one transmission has been terminated due to collision detection since the last RSTSTA command was issued. (This stops the DMA until reset with RSTSTA bit).

Bit 25 – LCOL LON Collision Detected Flag

Value	Description
0	No collision occurred while transmitting since the last RSTSTA command was issued.
1	At least one collision occurred while transmitting since the last RSTSTA command was issued.

Bit 24 – LTXD LON Transmission End Flag

Value	Description
0	Transmission on going or no transmission occurred since the last RSTSTA command was issued.
1	At least one transmission has been performed since the last RSTSTA command was issued.

Bit 10 – UNRE Underrun Error

Downloaded from Arrow.com.

Value	Description
0	No LON underrun error has occurred since the last RSTSTA command was issued.
1	At least one LON underrun error has occurred since the last RSTSTA command was issued.

Bit 9 – TXEMPTY Transmitter Empty (cleared by writing FLEX_US_THR)

Value	Description
0	There are characters in either FLEX_US_THR or the Transmit Shift Register, or the transmitter is disabled.
1	There are no characters in FLEX_US_THR, nor in the Transmit Shift Register.

Bit 7 – LCRCE LON CRC Error

Value	Description
0	No CRC error has been detected since the last RSTSTA command was issued.
1	At least one CRC error has been detected since the last RSTSTA command was issued.

Bit 6 – LSFE LON Short Frame Error

Value	Description
0	No short frame received since the last RSTSTA command was issued.
1	At least one short frame received since the last RSTSTA command was issued.

Bit 5 – OVRE Overrun Error

Value	Description
0	No overrun error has occurred since the last RSTSTA command was issued.
1	At least one overrun error has occurred since the last RSTSTA command was issued.

Bit 1 – TXRDY Transmitter Ready (cleared by writing FLEX_US_THR)

Value	Description
0	A character in FLEX_US_THR is waiting to be transferred to the Transmit Shift Register, or the transmitter is disabled. As soon as the transmitter is enabled, TXRDY becomes 1.
1	There is no character in FLEX_US_THR.

Bit 0 – RXRDY Receiver Ready (cleared by reading FLEX_US_RHR)

Value	Description
0	No complete character has been received since the last read of FLEX_US_RHR or the receiver is disabled. If characters were being received when the receiver was disabled, RXRDY changes to 1 when the receiver is enabled.
1	At least one complete character has been received and FLEX_US_RHR has not yet been read.

Downloaded from Arrow.com.

45.10.18 USART Receive Holding Register

Name: FLEX_US_RHR
Offset: 0x218
Reset: 0x00000000
Property: Read-only

If FIFO is enabled (FLEX_US_CR.FIFOEN bit) and FLEX_US_FMR.RXRDYM = 0, see 45.7.11.6 USART Single Data Mode for details.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RXSYNH							RXCHR[8]
Access	R							R
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	RXCHR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 15 – RXSYNH Received Sync

Value	Description
0	Last character received is a data.
1	Last character received is a command.

Bits 8:0 – RXCHR[8:0] Received Character
Last character received if RXRDY is set.

Downloaded from Arrow.com.

45.10.19 USART Receive Holding Register (FIFO Multi Data)

Name: FLEX_US_RHR (FIFO_MULTI_DATA)
Offset: 0x218
Reset: 0x00000000
Property: Read-only

If FIFO is enabled (FLEX_US_CR.FIFOEN bit) and FLEX_US_FMR.RXRDYM > 0, see 45.7.11.7 USART Multiple Data Mode for details.

Bit	31	30	29	28	27	26	25	24
	RXCHR3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RXCHR2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RXCHR1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXCHR0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0:7, 8:15, 16:23, 24:31 – RXCHRx Received Character
First unread character in the Receive FIFO if RXRDY is set.

Downloaded from Arrow.com.

45.10.20 USART Transmit Holding Register

Name: FLEX_US_THR
Offset: 0x21C
Reset: –
Property: Write-only

If FIFO is enabled (FLEX_US_CR.FIFOEN bit) and FLEX_US_FMR.TXRDY = 0, see 45.7.11.6 USART Single Data Mode for details.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TXSYNH							TXCHR[8]
Access	W							W
Reset	–							–

Bit	7	6	5	4	3	2	1	0
	TXCHR[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 15 – TXSYNH Sync Field to be Transmitted

Value	Description
0	The next character sent is encoded as a data. Start frame delimiter is DATA SYNC.
1	The next character sent is encoded as a command. Start frame delimiter is COMMAND SYNC.

Bits 8:0 – TXCHR[8:0] Character to be Transmitted
The next character to be transmitted after the current character if TXRDY is not set.

Downloaded from Arrow.com.

45.10.21 USART Transmit Holding Register (FIFO Multi Data)

Name: FLEX_US_THR (FIFO_MULTI_DATA)
Offset: 0x21C
Reset: –
Property: Write-only

If FIFO is enabled (FLEX_US_CR.FIFOEN bit) and FLEX_US_FMR.TXRDY > 0, see 45.7.11.7 USART Multiple Data Mode for details.

Bit	31	30	29	28	27	26	25	24
	TXCHR3[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	TXCHR2[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TXCHR1[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TXCHR0[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0:7, 8:15, 16:23, 24:31 – TXCHRx Character to be Transmitted
Next character to be transmitted.

Downloaded from Arrow.com.

45.10.22 USART Baud Rate Generator Register

Name: FLEX_US_BRGR
Offset: 0x220
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						FP[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	CD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 18:16 – FP[2:0] Fractional Part

WARNING When the value of field FP is greater than 0, the SCK (oversampling clock) generates nonconstant duty cycles. The SCK high duration is increased by “selected clock” period from time to time. The duty cycle depends on the value of the CD field.

Value	Description
0	Fractional divider is disabled.
1–7	Baud rate resolution, defined by FP × 1/8.

Bits 15:0 – CD[15:0] Clock Divider

CD	USART_MODE ≠ ISO7816			USART_MODE = ISO7816
	SYNC = 0		SYNC = 1	
	OVER = 0	OVER = 1		
0	Baud Rate Clock Disabled			
1 to 65535	CD = Selected Clock / (16 × Baud Rate)	CD = Selected Clock / (8 × Baud Rate)	CD = Selected Clock / Baud Rate	CD = Selected Clock / (FI_DI_RATIO × Baud Rate)

Downloaded from Arrow.com.

45.10.23 USART Receiver Timeout Register

Name: FLEX_US_RTOR
Offset: 0x224
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								TO[16]
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	TO[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TO[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 16:0 – TO[16:0] Timeout Value
The TO field size is limited to 8 bits if the ISO7816 logic is not implemented on some instances of FLEXCOM. The ISO7816 logic is implemented if it is possible to write FLEX_US_MR.MAX_ITERATIONS=1 (a read operation must be performed after the write operation to check that MAX_ITERATIONS equals 1).

Value	Description
0	The receiver timeout is disabled.
1–131071	The receiver timeout is enabled and the timeout delay is TO × bit period.

Downloaded from Arrow.com.

45.10.24 USART Transmitter Timeguard Register

Name: FLEX_US_TTGR
Offset: 0x228
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

For LON-specific configurations, see USART Transmitter Timeguard Register (LON_MODE).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	TG[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – TG[7:0] Timeguard Value

Value	Description
0	The transmitter timeguard is disabled.
1-255	The transmitter timeguard is enabled and TG is timeguard delay / bit period.

Downloaded from Arrow.com.

45.10.25 USART Transmitter Timeguard Register (LON_MODE)

Name: FLEX_US_TTGR (LON_MODE)
Offset: 0x228
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	PCYCLE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PCYCLE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PCYCLE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – PCYCLE[23:0] LON PCYCLE Length

Value	Description
1-16777215	LON PCYCLE length in t_{bit}.

Downloaded from Arrow.com.

45.10.26 USART FI DI RATIO Register

Name: FLEX_US_FIDI
Offset: 0x240
Reset: 0x174
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

For LON-specific configurations, see USART Transmitter Timeguard Register (LON_MODE).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	FI_DI_RATIO[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
	FI_DI_RATIO[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	1	0	1	0	0

Bits 15:0 – FI_DI_RATIO[15:0] FI Over DI Ratio Value

Value	Description
0	If ISO7816 mode is selected, the baud rate generator generates no signal.
1-2	Do not use.
3-65535	If ISO7816 mode is selected, the baud rate is the clock provided on SCK divided by FI_DI_RATIO.

Downloaded from Arrow.com.

45.10.27 USART FI DI RATIO Register (LON_MODE)

Name: FLEX_US_FIDI (LON_MODE)
Offset: 0x240
Reset: 0x174
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	BETA2[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BETA2[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
	BETA2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	1	0	1	0	0

Bits 23:0 – BETA2[23:0] LON BETA2 Length

Value	Description
1-16777215	LON BETA2 length in t_{bit}.

Downloaded from Arrow.com.

45.10.28 USART Number of Errors Register

Name: FLEX_US_NER
Offset: 0x244
Reset: 0x00000000
Property: Read-only

This register is relevant only if USART_MODE = 0x4 or 0x6 in the USART Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	NB_ERRORS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bits 7:0 – NB_ERRORS[7:0] Number of Errors
Total number of errors that occurred during an ISO7816 transfer. This register automatically clears when read.

Downloaded from Arrow.com.

45.10.29 USART IrDA FILTER Register

Name: FLEX_US_IF
Offset: 0x24C
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x8 in the USART Mode Register.

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	IRDA_FILTER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – IRDA_FILTER[7:0] IrDA Filter
The IRDA_FILTER value must be defined to meet the following criteria:
$t_{peripheral\ clock} \times (IRDA_FILTER + 3) < 1.41\ \mu s$

Downloaded from Arrow.com.

45.10.30 USART Manchester Configuration Register

Name: FLEX_US_MAN
Offset: 0x250
Reset: 0xB0011004
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	RXIDLEV	DRIFT	ONE	RX_MPOL			RX_PP[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	1	0	1	1			0	0

Bit	23	22	21	20	19	18	17	16
					RX_PL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	1

Bit	15	14	13	12	11	10	9	8
				TX_MPOL			TX_PP[1:0]	
Access				R/W			R/W	R/W
Reset				1			0	0

Bit	7	6	5	4	3	2	1	0
					TX_PL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	1	0	0

Bit 31 – RXIDLEV Receiver Idle Value

Value	Description
0	Receiver line idle value is 0.
1	Receiver line idle value is 1.

Bit 30 – DRIFT Drift Compensation

Value	Description
0	The USART cannot recover from an important clock drift.
1	The USART can recover from clock drift. The 16X Clock mode must be enabled.

Bit 29 – ONE Must Be Set to 1
Bit 29 must always be set to 1 when programming the FLEX_US_MAN register.

Bit 28 – RX_MPOL Receiver Manchester Polarity

Value	Description
0	Logic zero is coded as a zero-to-one transition, Logic one is coded as a one-to-zero transition.
1	Logic zero is coded as a one-to-zero transition, Logic one is coded as a zero-to-one transition.

Bits 25:24 – RX_PP[1:0] Receiver Preamble Pattern detected
The following values assume that RX_MPOL field is not set:

Value	Name	Description
0	ALL_ONE	The preamble is composed of ‘1’s.
1	ALL_ZERO	The preamble is composed of ‘0’s.
2	ZERO_ONE	The preamble is composed of ‘01’s.
3	ONE_ZERO	The preamble is composed of ‘10’s.

Bits 19:16 – RX_PL[3:0] Receiver Preamble Length

Downloaded from Arrow.com.

Value	Description
0	The receiver preamble pattern detection is disabled.
1–15	The detected preamble length is RX_PL × Bit Period.

Bit 12 – TX_MPOL Transmitter Manchester Polarity

Value	Description
0	Logic zero is coded as a zero-to-one transition, Logic one is coded as a one-to-zero transition.
1	Logic zero is coded as a one-to-zero transition, Logic one is coded as a zero-to-one transition.

Bits 9:8 – TX_PP[1:0] Transmitter Preamble Pattern
The following values assume that TX_MPOL field is not set:

Value	Name	Description
0	ALL_ONE	The preamble is composed of ‘1’s.
1	ALL_ZERO	The preamble is composed of ‘0’s.
2	ZERO_ONE	The preamble is composed of ‘01’s.
3	ONE_ZERO	The preamble is composed of ‘10’s.

Bits 3:0 – TX_PL[3:0] Transmitter Preamble Length

Value	Description
0	The transmitter preamble pattern generation is disabled.
1–15	The preamble length is TX_PL × Bit Period.

Downloaded from Arrow.com.

45.10.31 USART LIN Mode Register

Name: FLEX_US_LINMR
Offset: 0x254
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0xA or 0xB in the USART Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							SYNCDIS	PDCM
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	DLC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WKUPTYP	FSDIS	DLM	CHKTYP	CHKDIS	PARDIS	NACT[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 17 – SYNCDIS Synchronization Disable

Value	Description
0	The synchronization procedure is performed in LIN slave node configuration.
1	The synchronization procedure is not performed in LIN slave node configuration.

Bit 16 – PDCM DMA Mode

Value	Description
0	The LIN mode register FLEX_US_LINMR is not written by the DMA.
1	The LIN mode register FLEX_US_LINMR (excepting that flag) is written by the DMA.

Bits 15:8 – DLC[7:0] Data Length Control

Value	Description
0-255	Defines the response data length if DLM = 0, in that case the response data length is equal to DLC+1 bytes.

Bit 7 – WKUPTYP Wake-up Signal Type

Value	Description
0	Setting the LINWKUP bit in the control register sends a LIN 2.0 wake-up signal.
1	Setting the LINWKUP bit in the control register sends a LIN 1.3 wake-up signal.

Bit 6 – FSDIS Frame Slot Mode Disable

Value	Description
0	The Frame Slot mode is enabled.
1	The Frame Slot mode is disabled.

Bit 5 – DLM Data Length Mode

Value	Description
0	The response data length is defined by the DLC field of this register.

Downloaded from Arrow.com.

Value	Description
1	The response data length is defined by the bits 5 and 6 of the identifier (FLEX_US_LINIR.IDCHR).

Bit 4 – CHKTYP Checksum Type

Value	Description
0	LIN 2.0 “enhanced” checksum
1	LIN 1.3 “classic” checksum

Bit 3 – CHKDIS Checksum Disable

Value	Description
0	In master node configuration, the checksum is computed and sent automatically. In slave node configuration, the checksum is checked automatically.
1	Whatever the node configuration is, the checksum is not computed/sent and it is not checked.

Bit 2 – PARDIS Parity Disable

Value	Description
0	In master node configuration, the identifier parity is computed and sent automatically. In master node and slave node configuration, the parity is checked automatically.
1	Whatever the node configuration is, the Identifier parity is not computed/sent and it is not checked.

Bits 1:0 – NACT[1:0] LIN Node Action

Values which are not listed in the table must be considered as “reserved”.

Value	Name	Description
0	PUBLISH	The USART transmits the response.
1	SUBSCRIBE	The USART receives the response.
2	IGNORE	The USART does not transmit and does not receive the response.

Downloaded from Arrow.com.

45.10.32 USART LIN Identifier Register

Name: FLEX_US_LINIR
Offset: 0x258
Reset: 0x00000000
Property: Read/Write

Write is possible only in LIN master node configuration.

This register is relevant only if USART_MODE = 0xA or 0xB in USART Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	IDCHR[7:0]							
Access	R/W-R	R/W-R	R/W-R	R/W-R	R/W-R	R/W-R	R/W-R	R/W-R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – IDCHR[7:0] Identifier Character
If USART_MODE = 0xA (master node configuration):

- IDCHR is Read/Write and its value is the identifier character to be transmitted.

If USART_MODE = 0xB (slave node configuration):

- IDCHR is Read-only and its value is the last identifier character that has been received.

Downloaded from Arrow.com.

45.10.33 USART LIN Baud Rate Register

Name: FLEX_US_LINBRR
Offset: 0x25C
Reset: 0x00000000
Property: Read-only

This register is relevant only if USART_MODE = 0xA or 0xB in USART Mode Register.

Returns the baud rate value after the synchronization process completion.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						LINFP[2:0]		
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	LINCD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LINCD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 18:16 – LINFP[2:0] Fractional Part after Synchronization

Bits 15:0 – LINCD[15:0] Clock Divider after Synchronization

Downloaded from Arrow.com.

45.10.34 USART LON Mode Register

Name: FLEX_US_LONMR
Offset: 0x260
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 in the USART Mode Register.

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	EOFS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			LCDS	DMAM	CDTAIL	TCOL	COLDET	COMMT
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 23:16 – EOFS[7:0] End of Frame Condition Size

Value	Description
0–255	Defines the minimum transitionless time for the IP to detect a LON end of frame condition. t_{eof} = (EOFS+1) × t_{clock} × 8 × (2- OVER)

Bit 5 – LCDS LON Collision Detection Source

Value	Description
0	LON collision detection source is external.
1	LON collision detection source is internal.

Bit 4 – DMAM LON DMA Mode

Value	Description
0	The LON data length register FLEX_US_LONDL is not written by the DMA.
1	The LON data length register FLEX_US_LONDL is written by the DMA.

Bit 3 – CDTAIL LON Collision Detection on Frame Tail

Value	Description
0	Detect collisions after CRC has been sent but prior end of transmission in LON comm_type = 1 mode.
1	Ignore collisions after CRC has been sent but prior end of transmission in LON comm_type = 1 mode.

Bit 2 – TCOL Terminate Frame upon Collision Notification

Value	Description
0	Do not terminate the frame in LON comm_type = 1 mode upon collision detection.
1	Terminate the frame in LON comm_type = 1 mode upon collision detection if possible.

Bit 1 – COLDET LON Collision Detection Feature

Downloaded from Arrow.com.

Value	Description
0	LON collision detection feature disabled.
1	LON collision detection feature enabled.

Bit 0 – COMMT LON comm_type Parameter Value

Value	Description
0	LON comm_type = 1 mode.
1	LON comm_type = 2 mode.

Downloaded from Arrow.com.

45.10.35 USART LON Preamble Register

Name: FLEX_US_LONPR
Offset: 0x264
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 in the USART Mode Register.

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			LONPL[13:8]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LONPL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 13:0 – LONPL[13:0] LON Preamble Length

Value	Description
1–16383	LON preamble length in t_{bit} (without byte-sync).

Downloaded from Arrow.com.

45.10.36 USART LON Data Length Register

Name: FLEX_US_LONDL
Offset: 0x268
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 in the USART Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	LONDL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – LONDL[7:0] LON Data Length

Value	Description
0–255	LON data length is LONDL + 1 byte.

Downloaded from Arrow.com.

45.10.37 USART LON L2HDR Register

Name: FLEX_US_LONL2HDR
Offset: 0x26C
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 in the USART Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PB	ALTP	BLI[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – PB LON Priority Bit

Value	Description
0	LON priority bit reset.
1	LON priority bit set.

Bit 6 – ALTP LON Alternate Path Bit

Value	Description
0	LON alternate path bit reset.
1	LON alternate path bit set.

Bits 5:0 – BLI[5:0] LON Backlog Increment

Value	Description
0-63	LON backlog increment to be generated as a result of delivering the LON frame.

Downloaded from Arrow.com.

45.10.38 USART LON Backlog Register

Name: FLEX_US_LONBL
Offset: 0x270
Reset: 0x00000000
Property: Read-only

This register is relevant only if USART_MODE = 0x9 in the USART Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			LONBL[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:0 – LONBL[5:0] LON Node Backlog Value

Value	Description
1-63	LON node backlog value.

Downloaded from Arrow.com.

45.10.39 USART LON Beta1 Tx Register

Name: FLEX_US_LONB1TX
Offset: 0x274
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 in the USART Mode Register.

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	BETA1TX[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BETA1TX[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BETA1TX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – BETA1TX[23:0] LON Beta1 Length after Transmission

Value	Description
1–16777215	LON beta1 length after transmission in t_{bit}.

Downloaded from Arrow.com.

45.10.40 USART LON Beta1 Rx Register

Name: FLEX_US_LONB1RX
Offset: 0x278
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 in the USART Mode Register.

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	BETA1RX[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BETA1RX[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BETA1RX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – BETA1RX[23:0] LON Beta1 Length after Reception

Value	Description
1–16777215	LON beta1 length after reception in t_{bit}.

Downloaded from Arrow.com.

45.10.41 USART LON Priority Register

Name: FLEX_US_LONPRIO
Offset: 0x27C
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 in the USART Mode Register.

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		NPS[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		PSNB[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 14:8 – NPS[6:0] LON Node Priority Slot

Value	Description
0-127	Node priority slot.

Bits 6:0 – PSNB[6:0] LON Priority Slot Number

Value	Description
0-127	Number of priority slots in the LON network configuration.

Downloaded from Arrow.com.

45.10.42 USART LON IDT Tx Register

Name: FLEX_US_IDTTX
Offset: 0x280
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 in the USART Mode Register.

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	IDTTX[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IDTTX[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IDTTX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – IDTTX[23:0] LON Indeterminate Time after Transmission (comm_type = 1 mode only)

Value	Description
0–16777215	LON indeterminate time after transmission in t_{bit}.

Downloaded from Arrow.com.

45.10.43 USART LON IDT Rx Register

Name: FLEX_US_IDTRX
Offset: 0x284
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 in the USART Mode Register.

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	IDTRX[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IDTRX[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IDTRX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – IDTRX[23:0] LON Indeterminate Time after Reception (comm_type = 1 mode only)

Value	Description
0–16777215	LON indeterminate time after reception in t_{bit}.

Downloaded from Arrow.com.

45.10.44 USART IC DIFF Register

Name: FLEX_US_ICDIFF
Offset: 0x288
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 in the USART Mode Register.

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					ICDIFF[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – ICDIFF[3:0] IC Differentiator Number

Downloaded from Arrow.com.

45.10.45 USART Comparison Register

Name: FLEX_US_CMPR
Offset: 0x290
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								VAL2[8]
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
	VAL2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		CMPPAR	CMPMODE[1:0]					VAL1[8]
Access		R/W	R/W	R/W				R/W
Reset		0	0	0				0

Bit	7	6	5	4	3	2	1	0
	VAL1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 24:16 – VAL2[8:0] Second Comparison Value for Received Character

Value	Description
0-511	The received character must be lower than or equal to the value of VAL2 and higher than or equal to VAL1 to set the FLEX_US_CSR.CMP flag.

Bit 14 – CMPPAR Compare Parity

Value	Description
0	The parity is not checked and a bad parity cannot prevent from waking up the system.
1	The parity is checked and a matching condition on data can be cancelled by an error on parity bit, so no wakeup is performed.

Bits 13:12 – CMPMODE[1:0] Comparison Mode

Value	Name	Description
0	FLAG_ONLY	Any character is received and comparison function drives CMP flag.
1	START_CONDITION	Comparison condition must be met to start reception.
2	FILTER	Comparison must be met to receive the current data only

Bits 8:0 – VAL1[8:0] First Comparison Value for Received Character

Value	Description
0-511	The received character must be higher than or equal to the value of VAL1 and lower than or equal to VAL2 to set the FLEX_US_CSR.CMP flag.

Downloaded from Arrow.com.

45.10.46 USART FIFO Mode Register

Name: FLEX_US_FMR
Offset: 0x2A0
Reset: 0x00000000
Property: Read/Write

This register reads '0' if the FIFO is disabled (see FLEX_US_CR to enable/disable the internal FIFO).

Bit	31	30	29	28	27	26	25	24
			RXFTHRES2[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			RXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			TXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FRTSC		RXRDYM[1:0]				TXRDYM[1:0]	
Access	R/W		R/W	R/W			R/W	R/W
Reset	0		0	0			0	0

Bits 29:24 – RXFTHRES2[5:0] Receive FIFO Threshold 2

Value	Description
0-16	Defines the Receive FIFO threshold 2 value (number of data). The FLEX_US_FESR.RXFTHF2 flag will be set when Receive FIFO goes from "above" threshold state to "equal to or below" threshold state.

Bits 21:16 – RXFTHRES[5:0] Receive FIFO Threshold

Value	Description
0-16	Defines the Receive FIFO threshold value (number of data). The FLEX_US_FESR.RXFTHF flag will be set when Receive FIFO goes from "below" threshold state to "equal to or above" threshold state.

Bits 13:8 – TXFTHRES[5:0] Transmit FIFO Threshold

Value	Description
0-16	Defines the Transmit FIFO threshold value (number of data). The FLEX_US_FESR.TXFTHF flag will be set when Transmit FIFO goes from "above" threshold state to "equal to or below" threshold state.

Bit 7 – FRTSC FIFO RTS Pin Control enable (Hardware Handshaking mode only)
See Hardware Handshaking for details.

Value	Description
0	RTS pin is not controlled by Receive FIFO thresholds.
1	RTS pin is controlled by Receive FIFO thresholds.

Bits 5:4 – RXRDYM[1:0] Receiver Ready Mode
If FIFOs are enabled, the FLEX_US_CSR.RXRDY flag behaves as follows.

Value	Name	Description
0	ONE_DATA	RXRDY will be at level '1' when at least one unread data is in the Receive FIFO.
1	TWO_DATA	RXRDY will be at level '1' when at least two unread data are in the Receive FIFO.
2	FOUR_DATA	RXRDY will be at level '1' when at least four unread data are in the Receive FIFO.

Downloaded from Arrow.com.

Bits 1:0 – TXRDYM[1:0] Transmitter Ready Mode
If FIFOs are enabled, the FLEX_US_CSR.TXRDY flag behaves as follows.

Value	Name	Description
0	ONE_DATA	TXRDY will be at level ‘1’ when at least one data can be written in the Transmit FIFO
1	TWO_DATA	TXRDY will be at level ‘1’ when at least two data can be written in the Transmit FIFO
2	FOUR_DATA	TXRDY will be at level ‘1’ when at least four data can be written in the Transmit FIFO

Downloaded from Arrow.com.

45.10.47 USART FIFO Level Register

Name: FLEX_US_FLR
Offset: 0x2A4
Reset: 0x00000000
Property: Read-only

This register reads '0' if the FIFO is disabled (see FLEX_US_CR to enable/disable the internal FIFO).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			RXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bits 21:16 – RXFL[5:0] Receive FIFO Level

Value	Description
0	There is no unread data in the Receive FIFO.
1-16	Indicates the number of unread data in the Receive FIFO.

Bits 5:0 – TXFL[5:0] Transmit FIFO Level

Value	Description
0	There is no data in the Transmit FIFO.
1-16	Indicates the number of data in the Transmit FIFO.

Downloaded from Arrow.com.

45.10.48 USART FIFO Interrupt Enable Register

Name: FLEX_US_FIER
Offset: 0x2A8
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							RXFTHF2	
Access							W	
Reset							–	

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 9 – RXFTHF2 RXFTHF2 Interrupt Enable

Bit 7 – RXFPTEF RXFPTEF Interrupt Enable

Bit 6 – TXFPTEF TXFPTEF Interrupt Enable

Bit 5 – RXFTHF RXFTHF Interrupt Enable

Bit 4 – RXFFF RXFFF Interrupt Enable

Bit 3 – RXFEF RXFEF Interrupt Enable

Bit 2 – TXFTHF TXFTHF Interrupt Enable

Bit 1 – TXFFF TXFFF Interrupt Enable

Bit 0 – TXFEF TXFEF Interrupt Enable

Downloaded from Arrow.com.

45.10.49 USART FIFO Interrupt Disable Register

Name: FLEX_US_FIDR
Offset: 0x2AC
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							RXFTHF2	
Access							W	
Reset							–	

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 9 – RXFTHF2 RXFTHF2 Interrupt Disable

Bit 7 – RXFPTEF RXFPTEF Interrupt Disable

Bit 6 – TXFPTEF TXFPTEF Interrupt Disable

Bit 5 – RXFTHF RXFTHF Interrupt Disable

Bit 4 – RXFFF RXFFF Interrupt Disable

Bit 3 – RXFEF RXFEF Interrupt Disable

Bit 2 – TXFTHF TXFTHF Interrupt Disable

Bit 1 – TXFFF TXFFF Interrupt Disable

Bit 0 – TXFEF TXFEF Interrupt Disable

Downloaded from Arrow.com.

45.10.50 USART FIFO Interrupt Mask Register

Name: FLEX_US_FIMR
Offset: 0x2B0
Reset: 0x00000000
Property: Read-only

This register reads '0' if the FIFO is disabled (see FLEX_US_CR to enable/disable the internal FIFO).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							RXFTHF2	
Access							R	
Reset							0	

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 9 – RXFTHF2 RXFTHF2 Interrupt Mask

Bit 7 – RXFPTEF RXFPTEF Interrupt Mask

Bit 6 – TXFPTEF TXFPTEF Interrupt Mask

Bit 5 – RXFTHF RXFTHF Interrupt Mask

Bit 4 – RXFFF RXFFF Interrupt Mask

Bit 3 – RXFEF RXFEF Interrupt Mask

Bit 2 – TXFTHF TXFTHF Interrupt Mask

Bit 1 – TXFFF TXFFF Interrupt Mask

Bit 0 – TXFEF TXFEF Interrupt Mask

Downloaded from Arrow.com.

45.10.51 USART FIFO Event Status Register

Name: FLEX_US_FESR
Offset: 0x2B4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							RXFTHF2	TXFLOCK
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 9 – RXFTHF2 Receive FIFO Threshold Flag 2 (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Number of unread data in Receive FIFO is above RXFTHRES threshold.
1	Number of unread data in Receive FIFO has reached RXFTHRES2 threshold since the last RSTSTA command was issued.

Bit 8 – TXFLOCK Transmit FIFO Lock

Value	Description
0	The Transmit FIFO is not locked.
1	The Transmit FIFO is locked.

Bit 7 – RXFPTEF Receive FIFO Pointer Error Flag
See 45.7.11.9 FIFO Pointer Error for details.

Value	Description
0	No Receive FIFO pointer occurred.
1	Receive FIFO pointer error occurred. Receiver must be reset.

Bit 6 – TXFPTEF Transmit FIFO Pointer Error Flag
See 45.7.11.9 FIFO Pointer Error for details.

Value	Description
0	No Transmit FIFO pointer occurred.
1	Transmit FIFO pointer error occurred. Transceiver must be reset.

Bit 5 – RXFTHF Receive FIFO Threshold Flag (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Number of unread data in Receive FIFO is below RXFTHRES threshold.
1	Number of unread data in Receive FIFO has reached RXFTHRES threshold since the last RSTSTA command was issued.

Downloaded from Arrow.com.

Bit 4 – RXFFF Receive FIFO Full Flag (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Receive FIFO is not empty.
1	Receive FIFO has been filled since the last RSTSTA command was issued.

Bit 3 – RXFEF Receive FIFO Empty Flag (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Receive FIFO is not empty.
1	Receive FIFO has been emptied since the last RSTSTA command was issued.

Bit 2 – TXFTHF Transmit FIFO Threshold Flag (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Number of data in Transmit FIFO is above TXFTHRES threshold.
1	Number of data in Transmit FIFO has reached TXFTHRES threshold since the last RSTSTA command was issued.

Bit 1 – TXFFF Transmit FIFO Full Flag (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Transmit FIFO is not full.
1	Transmit FIFO has been filled since the last RSTSTA command was issued.

Bit 0 – TXFEF Transmit FIFO Empty Flag (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Transmit FIFO is not empty.
1	Transmit FIFO has been emptied since the last RSTSTA command was issued.

Downloaded from Arrow.com.

45.10.52 USART Write Protection Mode Register

Name: FLEX_US_WPMR
Offset: 0x2E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						WPCREN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x555341	PASSWD	Writing any other value in this field aborts the write operation of bits WPEN, WPITEN and WPCREN. Always reads as 0.

Bit 2 – WPCREN Write Protection Control Enable
See USART Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x555341 (“USA” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x555341 (“USA” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable
See USART Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x555341 (“USA” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x555341 (“USA” in ASCII).

Bit 0 – WPEN Write Protection Enable
See USART Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on configuration registers if WPKEY corresponds to 0x555341 (“USA” in ASCII).
1	Enables the write protection on configuration registers if WPKEY corresponds to 0x555341 (“USA” in ASCII).

Downloaded from Arrow.com.

45.10.53 USART Write Protection Status Register

Name: FLEX_US_WPSR
Offset: 0x2E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of FLEX_US_WPSR.
1	A write protection violation has occurred since the last read of FLEX_US_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

45.10.54 SPI Control Register

Name: FLEX_SPI_CR
Offset: 0x400
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the SPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	FIFODIS	FIFOEN						LASTXFER
Access	W	W						W
Reset	–	–						–

Bit	23	22	21	20	19	18	17	16
							RXFCLR	TXFCLR
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
				REQCLR				
Access				W				
Reset				–				

Bit	7	6	5	4	3	2	1	0
	SWRST						SPIDIS	SPIEN
Access	W						W	W
Reset	–						–	–

Bit 31 – FIFODIS FIFO Disable

Value	Description
0	No effect.
1	Disables the Transmit and Receive FIFOs

Bit 30 – FIFOEN FIFO Enable

Value	Description
0	No effect.
1	Enables the Transmit and Receive FIFOs

Bit 24 – LASTXFER Last Transfer
See Peripheral Selection for more details.

Value	Description
0	No effect.
1	The current NPCS will be de-asserted after the character written in TD has been transferred. When CSAAT is set, the communication with the current serial peripheral can be closed by raising the corresponding NPCS line as soon as TD transfer is completed.

Bit 17 – RXFCLR Receive FIFO Clear

Value	Description
0	No effect.
1	Empties the Receive FIFO.

Bit 16 – TXFCLR Transmit FIFO Clear

Value	Description
0	No effect.
1	Empties the Transmit FIFO.

Downloaded from Arrow.com.

Bit 12 – REQCLR Request to Clear the Comparison Trigger

Value	Description
0	No effect.
1	Restarts the comparison trigger to enable FLEX_SPI_RDR.

Bit 7 – SWRST SPI Software Reset

The SPI is in Slave mode after software reset.

Value	Description
0	No effect.
1	Resets the SPI. A software-triggered hardware reset of the SPI interface is performed.

Bit 1 – SPIDIS SPI Disable

If a transfer is in progress when SPIDIS is set, the SPI completes the transmission of the shifter register and does not start any new transfer, even if the FLEX_US_THR is loaded.

All pins are set in Input mode after completion of the transmission in progress, if any.

Value	Description
0	No effect.
1	Disables the SPI.

Bit 0 – SPIEN SPI Enable

Value	Description
0	No effect.
1	Enables the SPI to transfer and receive data.

Downloaded from Arrow.com.

45.10.55 SPI Mode Register

Name: FLEX_SPI_MR
Offset: 0x404
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	DLYBCS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
					PCS[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
				CMPMODE				
Access				R/W				
Reset				0				

Bit	7	6	5	4	3	2	1	0
	LLB		WDRBT	MODFDIS	BRSRCCLK	PCSDEC	PS	MSTR
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bits 31:24 – DLYBCS[7:0] Delay Between Chip Selects
This field defines the delay between the inactivation and the activation of NPCS. The DLYBCS time guarantees nonoverlapping chip selects and solves bus contentions in case of peripherals having long data float times.
If DLYBCS is ≤ 6, six peripheral clock periods are inserted by default.
Otherwise, the following equations determine the delay:
If FLEX_SPI_MR.BRSRCCLK = 0: DLYBCS = Delay Between Chip Selects × $f_{peripheral\ clock}$
If FLEX_SPI_MR.BRSRCCLK = 1: DLYBCS = Delay Between Chip Selects × f_{GCLK}

Bits 19:16 – PCS[3:0] Peripheral Chip Select
This field is only used if fixed peripheral select is active (PS = 0).
If PCSDEC = 0:
PCS = xxx0 NPCS[3:0] = 1110
PCS = xx01 NPCS[3:0] = 1101
PCS = x011 NPCS[3:0] = 1011
PCS = 0111 NPCS[3:0] = 0111
PCS = 1111 forbidden (no peripheral is selected)
(x = don't care)
If PCSDEC = 1:
NPCS[3:0] output signals = PCS

Bit 12 – CMPMODE Comparison Mode

Value	Name	Description
0	FLAG_ONLY	Any character is received and comparison function drives CMP flag.
1	START_CONDITION	Comparison condition must be met to start reception of all incoming characters until REQCLR is set.

Downloaded from Arrow.com.

Bit 7 – LLB Local Loopback Enable
LLB controls the local loopback on the data shift register for testing in Master mode only (MISO is internally connected on MOSI).

Value	Description
0	Local loopback path disabled.
1	Local loopback path enabled.

Bit 5 – WDRBT Wait Data Read Before Transfer

Value	Description
0	No Effect. In Master mode, a transfer can be initiated regardless of the FLEX_SPI_RDR state.
1	In Master mode, a transfer can start only if FLEX_SPI_RDR is empty, i.e., does not contain any unread data. This mode prevents overrun error in reception.

Bit 4 – MODFDIS Mode Fault Detection

Value	Description
0	Mode fault detection is enabled.
1	Mode fault detection is disabled.

Bit 3 – BRSRCCLK Bit Rate Source Clock
If the bit BRSRCCLK = 1, the FLEX_US_CSRx.SCBR field must be programmed with a value greater than 1.

Value	Name	Description
0	PERIPH_CLK	The peripheral clock is the source clock for the bit rate generation.
1	GCLK	GCLK is the source clock for the bit rate generation, thus the bit rate can be independent of the core/peripheral clock.

Bit 2 – PCSDEC Chip Select Decode
When PCSDEC equals one, up to 15 Chip Select signals can be generated with the four NPCS lines using an external 4- to 16-bit decoder. The Chip Select registers define the characteristics of the 15 chip selects, with the following rules:
FLEX_SPI_CSR0 defines peripheral chip select signals 0 to 3.
FLEX_SPI_CSR1 defines peripheral chip select signals 4 to 7.
FLEX_SPI_CSR2 defines peripheral chip select signals 8 to 11.
FLEX_SPI_CSR3 defines peripheral chip select signals 12 to 14.

Value	Description
0	The chip selects are directly connected to a peripheral device.
1	The four NPCS chip select lines are connected to a 4- to 16-bit decoder.

Bit 1 – PS Peripheral Select

Value	Description
0	Fixed Peripheral Select
1	Variable Peripheral Select

Bit 0 – MSTR Master/Slave Mode

Value	Description
0	SPI is in Slave mode.
1	SPI is in Master mode.

Downloaded from Arrow.com.

45.10.56 SPI Receive Data Register

Name: FLEX_SPI_RDR
Offset: 0x408
Reset: 0x00000000
Property: Read-only

If FIFO is enabled (FLEX_SPI_CR.FIFOEN) and FLEX_SPI_FMR.RXRDYM = 0, see SPI Single Data Mode for details.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					PCS[3:0]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 19:16 – PCS[3:0] Peripheral Chip Select
In Master mode only, these bits indicate the value on the NPCS pins at the end of a transfer. Otherwise, these bits are read as zero.
Note: When using Variable Peripheral Select mode (FLEX_SPI_MR.PS = 1), it is mandatory to set the FLEX_SPI_MR.WDRBT bit to 1 if the PCS field must be processed in FLEX_SPI_RDR.

Bits 15:0 – RD[15:0] Receive Data
Data received by the SPI Interface is stored in this register in a right-justified format. Unused bits are read as zero.

Downloaded from Arrow.com.

45.10.57 SPI Receive Data Register (FIFO Multiple Data, 8-bit)

Name: FLEX_SPI_RDR (FIFO_MULTI_DATA_8)
Offset: 0x408
Reset: 0x00000000
Property: Read-only

If FIFO is enabled (FLEX_SPI_CR.FIFOEN) and FLEX_SPI_FMR.RXRDYM > 0, see SPI Multiple Data Mode for details.

Bit	31	30	29	28	27	26	25	24
	RD3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RD2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RD1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RD0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0:7, 8:15, 16:23, 24:31 – RDx Receive Data
First unread data in the Receive FIFO. Data received by the SPI Interface is stored in this register in a right-justified format. Unused bits are read as zero.

Downloaded from Arrow.com.

45.10.58 SPI Receive Data Register (FIFO Multiple Data, 16-bit)

Name: FLEX_SPI_RDR (FIFO_MULTI_DATA_16)
Offset: 0x408
Reset: 0x00000000
Property: Read-only

If FIFO is enabled (FLEX_SPI_CR.FIFOEN) and FLEX_SPI_FMR.RXRDYM > 0, see SPI Multiple Data Mode for details.

Bit	31	30	29	28	27	26	25	24
	RD1[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RD1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RD0[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RD0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0:15, 16:31 – RDx Receive Data
First unread data in the Receive FIFO. Data received by the SPI Interface is stored in this register in a right-justified format. Unused bits are read as zero.

Downloaded from Arrow.com.

45.10.59 SPI Transmit Data Register

Name: FLEX_SPI_TDR
Offset: 0x40C
Reset: –
Property: Write-only

If FIFO is enabled (FLEX_SPI_CR.FIFOEN) and FLEX_SPI_FMR.TXRDYM = 0, see 45.8.6.6 SPI Single Data Mode for details.

Bit	31	30	29	28	27	26	25	24
								LASTXFER
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
					PCS[3:0]			
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 24 – LASTXFER Last Transfer
This field is only used if variable peripheral select is active (FLEX_SPI_MR.PS = 1).

Value	Description
0	No effect.
1	The current NPCS is de-asserted after the transfer of the character written in TD. When FLEX_SPI_CSRx.CSAAT is set, the communication with the current serial peripheral can be closed by raising the corresponding NPCS line as soon as TD transfer is completed.

Bits 19:16 – PCS[3:0] Peripheral Chip Select
This field is only used if variable peripheral select is active (FLEX_SPI_MR.PS = 1).
If FLEX_SPI_MR.PCSDEC = 0:
PCS = xxx0 NPCS[3:0] = 1110
PCS = xx01 NPCS[3:0] = 1101
PCS = x011 NPCS[3:0] = 1011
PCS = 0111 NPCS[3:0] = 0111
PCS = 1111 forbidden (no peripheral is selected)
(x = don't care)
If FLEX_SPI_MR.PCSDEC = 1:
NPCS[3:0] output signals = PCS

Bits 15:0 – TD[15:0] Transmit Data
Data to be transmitted by the SPI Interface is stored in this register. Information to be transmitted must be written to the transmit data register in a right-justified format.

Downloaded from Arrow.com.

45.10.60 SPI Transmit Data Register (FIFO Multiple Data, 8- to 16-bit)

Name: FLEX_SPI_TDR (FIFO_MULTI_DATA)
Offset: 0x40C
Reset: –
Property: Write-only

If FIFO is enabled (FLEX_SPI_CR.FIFOEN) and FLEX_SPI_FMR.TXRDYM > 0, see Section 1.8.7.7 “SPI Multiple Data Mode” for details.

Bit	31	30	29	28	27	26	25	24
	TD1[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	TD1[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TD0[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TD0[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0:15, 16:31 – TDx Transmit Data
Next data to write in the Transmit FIFO. Information to be transmitted must be written to this register in a right-justified format.

Downloaded from Arrow.com.

45.10.61 SPI Status Register

Name: FLEX_SPI_SR
Offset: 0x410
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
								SPIENS
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
				SFERR	CMP	UNDES	TXEMPTY	NSSR
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					OVRES	MODF	TDRE	RDRF
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – RXFPTEF Receive FIFO Pointer Error Flag
See 45.8.6.8 FIFO Pointer Error for details.
This bit reads ‘0’ if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	No Receive FIFO pointer occurred
1	Receive FIFO pointer error occurred. Receiver must be reset

Bit 30 – TXFPTEF Transmit FIFO Pointer Error Flag
See 45.8.6.8 FIFO Pointer Error for details.
This bit reads ‘0’ if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	No Transmit FIFO pointer occurred
1	Transmit FIFO pointer error occurred. Transceiver must be reset

Bit 29 – RXFTHF Receive FIFO Threshold Flag
This bit reads ‘0’ if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	Number of unread data in Receive FIFO is below RXFTHRES threshold or RXFTH flag has been cleared.
1	Number of unread data in Receive FIFO has reached RXFTHRES threshold (changing states from “below threshold” to “equal to or above threshold”).

Bit 28 – RXFFF Receive FIFO Full Flag
This bit reads ‘0’ if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	Receive FIFO is not empty or RXFE flag has been cleared.
1	Receive FIFO has been filled (changing states from “not full” to “full”).

Downloaded from Arrow.com.

Bit 27 – RXFEF Receive FIFO Empty Flag
This bit reads '0' if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	Receive FIFO is not empty or RXFE flag has been cleared.
1	Receive FIFO has been emptied (changing states from "not empty" to "empty").

Bit 26 – TXFTHF Transmit FIFO Threshold Flag (cleared on read)
This bit reads '0' if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	Number of data in Transmit FIFO is above TXFTHRES threshold.
1	Number of data in Transmit FIFO has reached TXFTHRES threshold since the last read of FLEX_SPI_SR.

Bit 25 – TXFFF Transmit FIFO Full Flag (cleared on read)
This bit reads '0' if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	Transmit FIFO is not full or TXFF flag has been cleared.
1	Transmit FIFO has been filled since the last read of FLEX_SPI_SR.

Bit 24 – TXFEF Transmit FIFO Empty Flag (cleared on read)
This bit reads '0' if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	Transmit FIFO is not empty.
1	Transmit FIFO has been emptied since the last read of FLEX_SPI_SR.

Bit 16 – SPIENS SPI Enable Status

Value	Description
0	SPI is disabled.
1	SPI is enabled.

Bit 12 – SFERR Slave Frame Error (cleared on read)

Value	Description
0	There is no frame error detected for a slave access since the last read of FLEX_SPI_SR.
1	In Slave mode, the chip select raised while the character defined in FLEX_SPI_CSR0.BITS was not complete.

Bit 11 – CMP Comparison Status (cleared on read)

Value	Description
0	No received character matched the comparison criteria programmed in VAL1 and VAL2 fields in FLEX_SPI_CMPR since the last read of FLEX_SPI_SR.
1	A received character matched the comparison criteria since the last read of FLEX_SPI_SR.

Bit 10 – UNDES Underrun Error Status (Slave mode only) (cleared on read)

Value	Description
0	No underrun has been detected since the last read of FLEX_SPI_SR.
1	A transfer starts whereas no data has been loaded in FLEX_SPI_TDR, cleared when FLEX_SPI_SR is read.

Bit 9 – TXEMPTY Transmission Registers Empty (cleared by writing FLEX_SPI_TDR)

Value	Description
0	As soon as data is written in FLEX_SPI_TDR.
1	FLEX_SPI_TDR and internal shift register are empty. If a transfer delay has been defined, TXEMPTY is set after the end of this delay.

Bit 8 – NSSR NSS Rising (cleared on read)

Value	Description
0	No rising edge detected on NSS pin since the last read of FLEX_SPI_SR.

Downloaded from Arrow.com.

Value	Description
1	A rising edge occurred on NSS pin since the last read of FLEX_SPI_SR.

Bit 3 – OVRES Overrun Error Status (cleared on read)
An overrun occurs when FLEX_SPI_RDR is loaded at least twice from the shift register since the last read of FLEX_SPI_RDR.

Value	Description
0	No overrun has been detected since the last read of FLEX_SPI_SR.
1	An overrun has occurred since the last read of FLEX_SPI_SR.

Bit 2 – MODF Mode Fault Error (cleared on read)

Value	Description
0	No mode fault has been detected since the last read of FLEX_SPI_SR.
1	A mode fault occurred since the last read of FLEX_SPI_SR.

Bit 1 – TDRE Transmit Data Register Empty (cleared by writing FLEX_SPI_TDR)
When FIFOs are disabled:
0: Data has been written to FLEX_SPI_TDR and not yet transferred to the internal shift register.
1: The last data written to FLEX_SPI_TDR has been transferred to the internal shift register.
TDRE is cleared when the SPI is disabled or at reset. Enabling the SPI sets the TDRE flag.
When FIFOs are enabled:
0: Transmit FIFO cannot accept more data.
1: Transmit FIFO can accept data; one or more data can be written according to TXRDYM field configuration.
TDRE behavior with FIFOs enabled is illustrated in 45.8.6.5 TXEMPTY, TDRE and RDRF Behavior.

Bit 0 – RDRF Receive Data Register Full (cleared by reading FLEX_SPI_RDR)
When FIFOs are disabled:
0: No data has been received since the last read of FLEX_SPI_RDR.
1: Data has been received and the received data has been transferred from the internal shift register to FLEX_SPI_RDR since the last read of FLEX_SPI_RDR.
When FIFOs are enabled:
0: Receive FIFO is empty; no data to read.
1: At least one unread data is in the Receive FIFO.
RDRF behavior with FIFOs enabled is illustrated in 45.8.6.5 TXEMPTY, TDRE and RDRF Behavior.

Downloaded from Arrow.com.

45.10.62 SPI Interrupt Enable Register

Name: FLEX_SPI_IER
Offset: 0x414
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the SPI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					CMP	UNDES	TXEMPTY	NSSR
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
					OVRES	MODF	TDRE	RDRF
Access					W	W	W	W
Reset					–	–	–	–

Bit 31 – RXFPTEF RXFPTEF Interrupt Enable

Bit 30 – TXFPTEF TXFPTEF Interrupt Enable

Bit 29 – RXFTHF RXFTHF Interrupt Enable

Bit 28 – RXFFF RXFFF Interrupt Enable

Bit 27 – RXFEF RXFEF Interrupt Enable

Bit 26 – TXFTHF TXFTHF Interrupt Enable

Bit 25 – TXFFF TXFFF Interrupt Enable

Bit 24 – TXFEF TXFEF Interrupt Enable

Bit 11 – CMP Comparison Interrupt Enable

Bit 10 – UNDES Underrun Error Interrupt Enable

Bit 9 – TXEMPTY Transmission Registers Empty Enable

Bit 8 – NSSR NSS Rising Interrupt Enable

Downloaded from Arrow.com.

Bit 3 – OVRES Overrun Error Interrupt Enable

Bit 2 – MODF Mode Fault Error Interrupt Enable

Bit 1 – TDRE SPI Transmit Data Register Empty Interrupt Enable

Bit 0 – RDRF Receive Data Register Full Interrupt Enable

Downloaded from Arrow.com.

45.10.63 SPI Interrupt Disable Register

Name: FLEX_SPI_IDR
Offset: 0x418
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the SPI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					CMP	UNDES	TXEMPTY	NSSR
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
					OVRES	MODF	TDRE	RDRF
Access					W	W	W	W
Reset					–	–	–	–

Bit 31 – RXFPTEF RXFPTEF Interrupt Disable

Bit 30 – TXFPTEF TXFPTEF Interrupt Disable

Bit 29 – RXFTHF RXFTHF Interrupt Disable

Bit 28 – RXFFF RXFFF Interrupt Disable

Bit 27 – RXFEF RXFEF Interrupt Disable

Bit 26 – TXFTHF TXFTHF Interrupt Disable

Bit 25 – TXFFF TXFFF Interrupt Disable

Bit 24 – TXFEF TXFEF Interrupt Disable

Bit 11 – CMP Comparison Interrupt Disable

Bit 10 – UNDES Underrun Error Interrupt Disable

Bit 9 – TXEMPTY Transmission Registers Empty Disable

Bit 8 – NSSR NSS Rising Interrupt Disable

Downloaded from Arrow.com.

Bit 3 – OVRES Overrun Error Interrupt Disable

Bit 2 – MODF Mode Fault Error Interrupt Disable

Bit 1 – TDRE SPI Transmit Data Register Empty Interrupt Disable

Bit 0 – RDRF Receive Data Register Full Interrupt Disable

Downloaded from Arrow.com.

45.10.64 SPI Interrupt Mask Register

Name: FLEX_SPI_IMR
Offset: 0x41C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					CMP	UNDES	TXEMPTY	NSSR
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
					OVRES	MODF	TDRE	RDRF
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – RXFPTEF RXFPTEF Interrupt Mask

Bit 30 – TXFPTEF TXFPTEF Interrupt Mask

Bit 29 – RXFTHF RXFTHF Interrupt Mask

Bit 28 – RXFFF RXFFF Interrupt Mask

Bit 27 – RXFEF RXFEF Interrupt Mask

Bit 26 – TXFTHF TXFTHF Interrupt Mask

Bit 25 – TXFFF TXFFF Interrupt Mask

Bit 24 – TXFEF TXFEF Interrupt Mask

Bit 11 – CMP Comparison Interrupt Mask

Bit 10 – UNDES Underrun Error Interrupt Mask

Bit 9 – TXEMPTY Transmission Registers Empty Mask

Bit 8 – NSSR NSS Rising Interrupt Mask

Downloaded from Arrow.com.

Bit 3 – OVRES Overrun Error Interrupt Mask

Bit 2 – MODF Mode Fault Error Interrupt Mask

Bit 1 – TDRE SPI Transmit Data Register Empty Interrupt Mask

Bit 0 – RDRF Receive Data Register Full Interrupt Mask

Downloaded from Arrow.com.

45.10.65 SPI Chip Select Register

Name: FLEX_SPI_CSRx
Offset: 0x0430 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: R/W

This register can only be written if the WPEN bit is cleared in the SPI Write Protection Mode Register.

FLEX_SPI_CSRx must be written even if the user wants to use the default reset values. The BITS field is not updated with the translated value unless the register is written.

Bit	31	30	29	28	27	26	25	24
	DLYBCT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DLYBS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SCBR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – DLYBCT[7:0] Delay Between Consecutive Transfers
This field defines the delay between two consecutive transfers with the same peripheral without removing the chip select. The delay is always inserted after each transfer and before removing the chip select if needed.
When DLYBCT = 0, no delay between consecutive transfers is inserted and the clock keeps its duty cycle over the character transfers.
Otherwise, the following equations determine the delay:
If FLEX_SPI_MR.BRSRCCLK = 0: DLYBCT = Delay Between Consecutive Transfers × $f_{peripheral\ clock}$ / 32
If FLEX_SPI_MR.BRSRCCLK = 1: DLYBCT = Delay Between Consecutive Transfers × f_{GCLK} / 32

Bits 23:16 – DLYBS[7:0] Delay Before SPCK
This field defines the delay from NPCS falling edge (activation) to the first valid SPCK transition.
When DLYBS = 0, the delay is half the SPCK clock period.
Otherwise, the following equations determine the delay:
If FLEX_SPI_MR.BRSRCCLK = 0: DLYBS = Delay Before SPCK × $f_{peripheral\ clock}$
If FLEX_SPI_MR.BRSRCCLK = 1: DLYBS = Delay Before SPCK × f_{GCLK}

Bits 15:8 – SCBR[7:0] Serial Clock Bit Rate
In Master mode, the SPI Interface uses a modulus counter to derive the SPCK bit rate from the clock defined by the bit BRSRCCLK. The bit rate is selected by writing a value from 1 to 255 in the SCBR field. The following equations determine the SPCK bit rate:
If FLEX_SPI_MR.BRSRCCLK = 0: SCBR = $f_{peripheral\ clock}$ / SPCK Bit Rate
If FLEX_SPI_MR.BRSRCCLK = 1: SCBR = f_{GCLK} / SPCK Bit Rate
Programming the SCBR field to 0 is forbidden. Triggering a transfer while SCBR is at 0 can lead to unpredictable results.
If BRSRCCLK = 1 in FLEX_SPI_MR, SCBR must be programmed with a value greater than 1.
At reset, SCBR is 0 and the user has to program it at a valid value before performing the first transfer.

Downloaded from Arrow.com.

Note: If one of the FLEX_SPI_CSRx.SCBR fields is set to 1, the other FLEX_SPI_CSRx.SCBR fields must be set to 1 as well, if they are used to process transfers. If they are not used to transfer data, they can be set at any value.

Bits 7:4 – BITS[3:0] Bits Per Transfer
See Note.
The BITS field determines the number of data bits transferred. Reserved values should not be used.

Value	Name	Description
0	8_BIT	8 bits for transfer
1	9_BIT	9 bits for transfer
2	10_BIT	10 bits for transfer
3	11_BIT	11 bits for transfer
4	12_BIT	12 bits for transfer
5	13_BIT	13 bits for transfer
6	14_BIT	14 bits for transfer
7	15_BIT	15 bits for transfer
8	16_BIT	16 bits for transfer
9–15	Reserved	

Bit 3 – CSAAT Chip Select Active After Transfer

Value	Description
0	The Peripheral Chip Select Line rises as soon as the last transfer is achieved.
1	The Peripheral Chip Select does not rise after the last transfer is achieved. It remains active until a new transfer is requested on a different chip select.

Bit 2 – CSNAAT Chip Select Not Active After Transfer (Ignored if CSAAT = 1)

If FLEX_SPI_MR.BRSRCCLK = 0: $\frac{\text{DLYBCS}}{\text{f}_{\text{peripheral clock}}}$(if DLYBCS ≠ 0)

If FLEX_SPI_MR.BRSRCCLK = 1: $\frac{\text{DLYBCS}}{\text{f}_{GCLK}}$

If DLYBCS < 6, a minimum of six periods is introduced.

Value	Description
0	The Peripheral Chip Select does not rise between two transfers if the FLEX_SPI_TDR is reloaded before the end of the first transfer and if the two transfers occur on the same Chip Select.
1	The Peripheral Chip Select rises systematically after each transfer performed on the same slave. It remains inactive after the end of transfer for a minimal duration of:

Bit 1 – NCPHA Clock Phase
NCPHA determines which edge of SPCK causes data to change and which edge causes data to be captured. NCPHA is used with CPOL to produce the required clock/data relationship between master and slave devices.

Value	Description
0	Data are changed on the leading edge of SPCK and captured on the following edge of SPCK.
1	Data are captured on the leading edge of SPCK and changed on the following edge of SPCK.

Bit 0 – CPOL Clock Polarity
CPOL is used to determine the inactive state value of the serial clock (SPCK). It is used with NCPHA to produce the required clock/data relationship between master and slave devices.

Value	Description
0	The inactive state value of SPCK is logic level zero.
1	The inactive state value of SPCK is logic level one.

Downloaded from Arrow.com.

45.10.66 SPI FIFO Mode Register

Name: FLEX_SPI_FMR
Offset: 0x440
Reset: 0x00000000
Property: Read/Write

This register reads '0' if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO)

Bit	31	30	29	28	27	26	25	24
			RXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			TXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RXRDYM[1:0]				TXRDYM[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 29:24 – RXFTHRES[5:0] Receive FIFO Threshold

Value	Description
0-16	Defines the Receive FIFO threshold value (number of data). The FLEX_SPI_SR.RXFTH flag will be set when Receive FIFO goes from "below" threshold state to "equal to or above" threshold state.

Bits 21:16 – TXFTHRES[5:0] Transmit FIFO Threshold

Value	Description
0-16	Defines the Transmit FIFO threshold value (number of data). The FLEX_SPI_SR.TXFTH flag will be set when Transmit FIFO goes from "above" threshold state to "equal to or below" threshold state.

Bits 5:4 – RXRDYM[1:0] Receive Data Register Full Mode
If FIFOs are enabled, the FLEX_SPI_SR.RDRF flag behaves as follows.

Value	Name	Description
0	ONE_DATA	RDRF will be at level '1' when at least one unread data is in the Receive FIFO.
1	TWO_DATA	RDRF will be at level '1' when at least two unread data are in the Receive FIFO. Cannot be used when FLEX_SPI_MR.MSTR =1, or if FLEX_SPI_MR.PS =1.
2	FOUR_DATA	RDRF will be at level '1' when at least four unread data are in the Receive FIFO. Cannot be used when FLEX_SPI_CSRx.BITS is greater than 0, or if FLEX_SPI_MR.MSTR =1, or if FLEX_SPI_MR.PS =1.

Bits 1:0 – TXRDYM[1:0] Transmit Data Register Empty Mode
If FIFOs are enabled, the FLEX_SPI_SR.TDRE flag behaves as follows.

Value	Name	Description
0	ONE_DATA	TDRE will be at level '1' when at least one data can be written in the Transmit FIFO.
1	TWO_DATA	TDRE will be at level '1' when at least two data can be written in the Transmit FIFO. Cannot be used if FLEX_SPI_MR.PS =1.

Downloaded from Arrow.com.

45.10.67 SPI FIFO Level Register

Name: FLEX_SPI_FLR
Offset: 0x444
Reset: 0x00000000
Property: Read-only

This register reads '0' if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			RXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bits 21:16 – RXFL[5:0] Receive FIFO Level

Value	Description
0	There is no unread data in the Receive FIFO.
1-16	Indicates the number of unread data in the Receive FIFO.

Bits 5:0 – TXFL[5:0] Transmit FIFO Level

Value	Description
0	There is no data in the Transmit FIFO.
1-16	Indicates the number of data in the Transmit FIFO.

Downloaded from Arrow.com.

45.10.68 SPI Comparison Register

Name: FLEX_SPI_CMPR
Offset: 0x448
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	VAL2[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	VAL2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	VAL1[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VAL1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – VAL2[15:0] Second Comparison Value for Received Character

Value	Description
0-65535	The received character must be lower than or equal to the value of VAL2 and higher than or equal to VAL1 to set the FLEX_SPI_CSR.CMP flag.

Bits 15:0 – VAL1[15:0] First Comparison Value for Received Character

Value	Description
0-65535	The received character must be higher than or equal to the value of VAL1 and lower than or equal to VAL2 to set the FLEX_SPI_SR.CMP flag.

Downloaded from Arrow.com.

45.10.69 SPI Write Protection Mode Register

Name: FLEX_SPI_WPMR
Offset: 0x4E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						WPCREN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x535049	PASSWD	Writing any other value in this field aborts the write operation of bits WPEN, WPITEN and WPCREN. Always reads as 0.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x535049 (“SPI” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x535049 (“SPI” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x535049 (“SPI” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x535049 (“SPI” in ASCII).

Bit 0 – WPEN Write Protection Enable

See 45.8.7 SPI Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x535049 (“SPI” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x535049 (“SPI” in ASCII).

Downloaded from Arrow.com.

45.10.70 SPI Write Protection Status Register

Name: FLEX_SPI_WPSR
Offset: 0x4E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protect violation has occurred since the last read of FLEX_SPI_WPSR.
1	A write protect violation has occurred since the last read of FLEX_SPI_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

45.10.71 TWI Control Register

Name: FLEX_TWI_CR
Offset: 0x600
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the TWI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
			FIFODIS	FIFOEN		LOCKCLR		THRCLR
Access			W	W		W		W
Reset			–	–		–		–

Bit	23	22	21	20	19	18	17	16
							ACMDIS	ACMEN
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
	CLEAR	PECRQ	PECDIS	PECEN	SMBDIS	SMBEN	HSDIS	HSEN
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	SWRST	QUICK	SVDIS	SVEN	MSDIS	MSEN	STOP	START
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 29 – FIFODIS FIFO Disable

Value	Description
0	No effect.
1	Disable the Transmit and Receive FIFOs

Bit 28 – FIFOEN FIFO Enable

Value	Description
0	No effect.
1	Enable the Transmit and Receive FIFOs

Bit 26 – LOCKCLR Lock Clear

Value	Description
0	No effect.
1	Clear the TWI FSM lock.

Bit 24 – THRCLR Transmit Holding Register Clear

Value	Description
0	No effect.
1	Clear the Transmit Holding Register and set TXRDY, TXCOMP flags.

Bit 17 – ACMDIS Alternative Command Mode Disable

Value	Description
0	No effect.
1	Alternative Command mode disabled.

Bit 16 – ACMEN Alternative Command Mode Enable

Downloaded from Arrow.com.

Value	Description
0	No effect.
1	Alternative Command mode enabled.

Bit 15 – CLEAR Bus CLEAR Command

Value	Description
0	No effect.
1	If Master mode is enabled, send a bus clear command.

Bit 14 – PECRQ PEC Request

Value	Description
0	No effect.
1	A PEC check or transmission is requested.

Bit 13 – PECDIS Packet Error Checking Disable

Value	Description
0	No effect.
1	SMBus PEC (CRC) generation and check disabled.

Bit 12 – PECEN Packet Error Checking Enable

Value	Description
0	No effect.
1	SMBus PEC (CRC) generation and check enabled.

Bit 11 – SMBDIS SMBus Mode Disabled

Value	Description
0	No effect.
1	SMBus mode disabled.

Bit 10 – SMBEN SMBus Mode Enabled

Value	Description
0	No effect.
1	If SMBDIS = 0, SMBus mode enabled.

Bit 9 – HSDIS TWI High-Speed Mode Disabled

Value	Description
0	No effect.
1	High-speed mode disabled.

Bit 8 – HSEN TWI High-Speed Mode Enabled

Value	Description
0	No effect.
1	High-speed mode enabled.

Bit 7 – SWRST Software Reset

Value	Description
0	No effect.
1	Equivalent to a system reset.

Bit 6 – QUICK SMBus Quick Command

Value	Description
0	No effect.
1	If Master mode is enabled, an SMBus Quick Command is sent.

Bit 5 – SVDIS TWI Slave Mode Disabled

Downloaded from Arrow.com.

Value	Description
0	No effect.
1	Slave mode is disabled. The shifter and holding characters (if it contains data) are transmitted in the case of a read operation. In a write operation, the character being transferred must be completely received before disabling.

Bit 4 – SVEN TWI Slave Mode Enabled
Switching from Master to Slave mode is only permitted when TXCOMP = 1.

Value	Description
0	No effect.
1	Enables the Slave mode (SVDIS must be written to 0).

Bit 3 – MSDIS TWI Master Mode Disabled

Value	Description
0	No effect.
1	The Master mode is disabled, all pending data is transmitted. The shifter and holding characters (if it contains data) are transmitted in case of write operation. In read operation, the character being transferred must be completely received before disabling.

Bit 2 – MSEN TWI Master Mode Enabled
Switching from Slave to Master mode is only permitted when TXCOMP = 1.

Value	Description
0	No effect.
1	Enables the Master mode (MSDIS must be written to 0).

Bit 1 – STOP Send a STOP Condition

Value	Description
0	No effect.
1	STOP condition is sent just after completing the current byte transmission in Master Read mode. – In single data byte master read, both START and STOP must be set. – In multiple data bytes master read, the STOP must be set after the last data received but one. – In Master Read mode, if a NACK bit is received, the STOP is automatically performed. – In master data write operation, a STOP condition will be sent after the transmission of the current data is finished.

Bit 0 – START Send a START Condition
This action is necessary when the TWI peripheral needs to read data from a slave. When configured in Master mode with a write operation, a frame is sent as soon as the user writes a character in the Transmit Holding Register (FLEX_TWI_THR).

Value	Description
0	No effect.
1	A frame beginning with a START bit is transmitted according to the features defined in the TWI Master Mode Register (FLEX_TWI_MMR).

Downloaded from Arrow.com.

45.10.72 TWI Control Register (FIFO_ENABLED)

Name: FLEX_TWI_CR (FIFO_ENABLED)
Offset: 0x600
Reset: –
Property: Write-only

If FIFO is enabled (FLEX_US_CR.FIFOEN bit), see 45.9.6.8 TWI Multiple Data Mode for details.

This register can only be written if the WPCREN bit is cleared in the TWI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
			FIFODIS	FIFOEN		TXFLCLR	RXFCLR	TXFCLR
Access			W	W		W	W	W
Reset			–	–		–	–	–

Bit	23	22	21	20	19	18	17	16
							ACMDIS	ACMEN
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
	CLEAR	PECRQ	PECDIS	PECEN	SMBDIS	SMBEN	HSDIS	HSEN
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	SWRST	QUICK	SVDIS	SVEN	MSDIS	MSEN	STOP	START
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 29 – FIFODIS FIFO Disable

Value	Description
0	No effect.
1	Disable the Transmit and Receive FIFOs.

Bit 28 – FIFOEN FIFO Enable

Value	Description
0	No effect.
1	Enable the Transmit and Receive FIFOs.

Bit 26 – TXFLCLR Transmit FIFO Lock CLEAR

Value	Description
0	No effect.
1	Clears the Transmit FIFO Lock.

Bit 25 – RXFCLR Receive FIFO Clear

Value	Description
0	No effect.
1	Empties the Receive FIFO.

Bit 24 – TXFCLR Transmit FIFO Clear

Value	Description
0	No effect.
1	Empties the Transmit FIFO.

Bit 17 – ACMDIS Alternative Command Mode Disable

Downloaded from Arrow.com.

Value	Description
0	No effect.
1	Alternative Command mode disabled.

Bit 16 – ACMEN Alternative Command Mode Enable

Value	Description
0	No effect.
1	Alternative Command mode enabled.

Bit 15 – CLEAR Bus CLEAR Command

Value	Description
0	No effect.
1	If Master mode is enabled, send a bus clear command.

Bit 14 – PECRQ PEC Request

Value	Description
0	No effect.
1	A PEC check or transmission is requested.

Bit 13 – PECDIS Packet Error Checking Disable

Value	Description
0	No effect.
1	SMBus PEC (CRC) generation and check disabled.

Bit 12 – PECEN Packet Error Checking Enable

Value	Description
0	No effect.
1	SMBus PEC (CRC) generation and check enabled.

Bit 11 – SMBDIS SMBus Mode Disabled

Value	Description
0	No effect.
1	SMBus mode disabled.

Bit 10 – SMBEN SMBus Mode Enabled

Value	Description
0	No effect.
1	If SMBDIS = 0, SMBus mode enabled.

Bit 9 – HSDIS TWI High-Speed Mode Disabled

Value	Description
0	No effect.
1	High-speed mode disabled.

Bit 8 – HSEN TWI High-Speed Mode Enabled

Value	Description
0	No effect.
1	High-speed mode enabled.

Bit 7 – SWRST Software Reset

Value	Description
0	No effect.
1	Equivalent to a system reset.

Bit 6 – QUICK SMBus Quick Command

Downloaded from Arrow.com.

Value	Description
0	No effect.
1	If Master mode is enabled, a SMBus Quick Command is sent.

Bit 5 – SVDIS TWI Slave Mode Disabled

Value	Description
0	No effect.
1	Slave mode is disabled. The shifter and holding characters (if it contains data) are transmitted in the case of a read operation. In a write operation, the character being transferred must be completely received before disabling.

Bit 4 – SVEN TWI Slave Mode Enabled
Switching from Master to Slave mode is only permitted when TXCOMP = 1.

Value	Description
0	No effect.
1	Enables the Slave mode (SVDIS must be written to 0).

Bit 3 – MSDIS TWI Master Mode Disabled

Value	Description
0	No effect.
1	The Master mode is disabled, all pending data is transmitted. The shifter and holding characters (if it contains data) are transmitted in the case of a write operation. In a read operation, the character being transferred must be completely received before disabling.

Bit 2 – MSEN TWI Master Mode Enabled
Switching from Slave to Master mode is only permitted when TXCOMP = 1.

Value	Description
0	No effect.
1	Enables the Master mode (MSDIS must be written to 0).

Bit 1 – STOP Send a STOP Condition

Value	Description
0	No effect.
1	STOP condition is sent just after completing the current byte transmission in Master Read mode. – In single data byte master read, both START and STOP must be set. – In multiple data bytes master read, the STOP must be set after the last data received but one. – In Master Read mode, if a NACK bit is received, the STOP is automatically performed. – In master data write operation, a STOP condition will be sent after the transmission of the current data is finished.

Bit 0 – START Send a START Condition
This action is necessary when the TWI peripheral needs to read data from a slave. When configured in Master mode with a write operation, a frame is sent as soon as the user writes a character in the Transmit Holding Register (FLEX_TWI_THR).

Value	Description
0	No effect.
1	A frame beginning with a START bit is transmitted according to the features defined in the TWI Master Mode Register (FLEX_TWI_MMR).

Downloaded from Arrow.com.

45.10.73 TWI Master Mode Register

Name: FLEX_TWI_MMR
Offset: 0x604
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
								NOAP
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
		DADR[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				MREAD			IADRSZ[1:0]	
Access				R/W			R/W	R/W
Reset				0			0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 24 – NOAP No Auto-Stop On NACK Error

Value	Description
0	A stop condition is sent automatically upon Not-Acknowledge error detection.
1	No automatic action is performed upon Not-Acknowledge error detection.

Bits 22:16 – DADR[6:0] Device Address
The device address is used to access slave devices in Read or Write mode. Those bits are only used in Master mode.

Bit 12 – MREAD Master Read Direction

Value	Description
0	Master write direction.
1	Master read direction.

Bits 9:8 – IADRSZ[1:0] Internal Device Address Size

Value	Name	Description
0	NONE	No internal device address
1	1_BYTE	One-byte internal device address
2	2_BYTE	Two-byte internal device address
3	3_BYTE	Three-byte internal device address

Downloaded from Arrow.com.

45.10.74 TWI Slave Mode Register

Name: FLEX_TWI_SMR
Offset: 0x608
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TWI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		SADR[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		MASK[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SNIFF	SCLWSDIS	BSEL	SADAT	SMHH	SMDA		NACKEN
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bits 22:16 – SADR[6:0] Slave Address
The slave device address is used in Slave mode in order to be accessed by master devices in Read or Write mode. SADR must be programmed before enabling the Slave mode or after a general call. Writes at other times have no effect.

Bits 14:8 – MASK[6:0] Slave Address Mask
A mask can be applied on the slave device address in Slave mode in order to allow multiple address answer. For each bit of the MASK field set to one, the corresponding SADR bit will be masked.
If the MASK field is set to 0, no mask is applied to the SADR field.

Bit 7 – SNIFF Slave Sniffer Mode

Value	Description
0	Slave Sniffer mode is disabled.
1	Slave Sniffer mode is enabled.

Bit 6 – SCLWSDIS Clock Wait State Disable

Value	Description
0	No effect.
1	Clock stretching disabled in Slave mode, OVRE and UNRE will indicate overrun and underrun.

Bit 5 – BSEL TWI Bus Selection

Value	Description
0	TWI analyzes the TWCK and TWD pins from its TWI bus.
1	TWI analyzes the TWCK pins TWD from consecutive index TWI peripheral of the product.

Bit 4 – SADAT Slave Address Treated as Data
When Slave Sniffer Mode is enabled, the slave address is always received as data in FLEX_TWI_RHR and SADAT has no effect.

Downloaded from Arrow.com.

Value	Description
0	Slave address is handled normally (will not trig RXRDY flag and will not fill FLEX_TWI_RHR upon reception).
1	Slave address is handled as data field, RXRDY will be set and FLEX_TWI_RHR filled upon slave address reception.

Bit 3 – SMHH SMBus Host Header

Value	Description
0	Acknowledge of the SMBus Host Header disabled.
1	Acknowledge of the SMBus Host Header enabled.

Bit 2 – SMDA SMBus Default Address

Value	Description
0	Acknowledge of the SMBus Default Address disabled.
1	Acknowledge of the SMBus Default Address enabled.

Bit 0 – NACKEN Slave Receiver Data Phase NACK Enable

Value	Description
0	Normal value to be returned in the ACK cycle of the data phase in Slave Receiver mode.
1	NACK value to be returned in the ACK cycle of the data phase in Slave Receiver mode.

Downloaded from Arrow.com.

45.10.75 TWI Internal Address Register

Name: FLEX_TWI_IADR
Offset: 0x60C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	IADR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IADR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IADR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – IADR[23:0] Internal Address
0, 1, 2 or 3 bytes depending on IADRSZ.

Downloaded from Arrow.com.

45.10.76 TWI Clock Waveform Generator Register

Name: FLEX_TWI_CWGR
Offset: 0x610
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TWI Write Protection Mode Register.

FLEX_TWI_CWGR is only used in Master mode.

Bit	31	30	29	28	27	26	25	24
			HOLD[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
				BRSRCCLK		CKDIV[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	15	14	13	12	11	10	9	8
	CHDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CLDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 29:24 – HOLD[5:0] TWD Hold Time Versus TWCK Falling
If High-speed mode is selected TWD is internally modified on the TWCK falling edge to meet the I^2C specified maximum hold time, else if High-speed mode is not configured TWD is kept unchanged after TWCK falling edge for a period of (HOLD + 3) × $t_{peripheral\ clock}$.

Bit 20 – BRSRCCLK Bit Rate Source Clock

Value	Name	Description
0	PERIPH_CLK	The peripheral clock is the source clock for the bit rate generation.
1	GCLK	GCLK is the source clock for the bit rate generation, thus the bit rate can be independent of the core/peripheral clock.

Bits 18:16 – CKDIV[2:0] Clock Divider
The CKDIV is used to increase both SCL high and low periods.

Bits 15:8 – CHDIV[7:0] Clock High Divider
The SCL high period is defined as follows:
If FLEX_TWI_FILTR.FILT = 0

- If BRSRCCLK = 0: CHDIV = $((t_{high}/t_{peripheral\ clock}) - 3)/2^{CKDIV}$
- If BRSRCCLK = 1: CHDIV = $(t_{high}/t_{ext_ck})/2^{CKDIV}$

If FLEX_TWI_FILTR.FILT = 1

- If BRSRCCLK = 0: CHDIV = $((t_{high}/t_{peripheral\ clock}) - 3 - (THRES+1))/2^{CKDIV}$
- If BRSRCCLK = 1: CHDIV = $((t_{high} - (THRES+1) * t_{peripheral\ clock})/t_{ext_ck})/2^{CKDIV}$

Bits 7:0 – CLDIV[7:0] Clock Low Divider
The SCL low period is defined as follows:

Downloaded from Arrow.com.

If FLEX_TWI_FILTR.FILT = 0

- If BRSRCCLK = 0: CLDIV = $((t_{low}/t_{peripheral\ clock}) - 3)/2^{CKDIV}$
- If BRSRCCLK = 1: CLDIV = $(t_{low}/t_{ext_ck})/2^{CKDIV}$

If FLEX_TWI_FILTR.FILT = 1

- If BRSRCCLK = 0: CLDIV = $((t_{low}/t_{peripheral\ clock}) - 3 - (THRES+1))/2^{CKDIV}$
- If BRSRCCLK = 1: CLDIV = $((t_{low} - (THRES+1) * t_{peripheral\ clock})/t_{ext_ck})/2^{CKDIV}$

Downloaded from Arrow.com.

45.10.77 TWI Status Register

Name: FLEX_TWI_SR
Offset: 0x620
Reset: 0x0300F009
Property: Read-only

Bit	31	30	29	28	27	26	25	24
						SR	SDA	SCL
Access						R	R	R
Reset						0	1	1

Bit	23	22	21	20	19	18	17	16
	LOCK		SMBHHM	SMBDAM	PECERR	TOUT		MCACK
Access	R		R	R	R	R		R
Reset	0		0	0	0	0		0

Bit	15	14	13	12	11	10	9	8
					EOSACC	SCLWS	ARBLST	NACK
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UNRE	OVRE	GACC	SVACC	SVREAD	TXRDY	RXRDY	TXCOMP
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	0	0	1

Bit 26 – SR Start Repeated

Value	Description
0	No repeated start has been detected since last FLEX_TWI_SR read.
1	At least one repeated start has been detected since last FLEX_TWI_SR read.

Bit 25 – SDA SDA Line Value

Value	Description
0	SDA line sampled value is ‘0’.
1	SDA line sampled value is ‘1’.

Bit 24 – SCL SCL Line Value

Value	Description
0	SCL line sampled value is ‘0’.
1	SCL line sampled value is ‘1.’

Bit 23 – LOCK TWI Lock Due to Frame Errors

Value	Description
0	The TWI is not locked.
1	The TWI is locked due to frame errors (see 45.9.3.12 Handling Errors in Alternative Command and 45.9.6 TWI FIFOs).

Bit 21 – SMBHHM SMBus Host Header Address Match (cleared on read)

Value	Description
0	No SMBus Host Header Address received.
1	A SMBus Host Header Address was received.

Bit 20 – SMBDAM SMBus Default Address Match (cleared on read)

Downloaded from Arrow.com.

Value	Description
0	No SMBus Default Address received.
1	A SMBus Default Address was received.

Bit 19 – PECERR PEC Error (cleared on read)

Value	Description
0	No SMBus PEC error occurred.
1	A SMBus PEC error occurred.

Bit 18 – TOUT Timeout Error (cleared on read)

Value	Description
0	No SMBus timeout occurred.
1	SMBus timeout occurred.

Bit 16 – MCACK Master Code Acknowledge (cleared on read)
MACK used in Slave mode:

Value	Description
0	No master code has been received.
1	A master code has been received.

Bit 11 – EOSACC End Of Slave Access (cleared on read)
This bit is only used in Slave mode.
EOSACC behavior can be seen in figures Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	A slave access is being performing.
1	The Slave Access is finished. End Of Slave Access is automatically set as soon as SVACC is reset.

Bit 10 – SCLWS Clock Wait State
This bit is only used in Slave mode.
SCLWS behavior can be seen in figures Clock Stretching in Read Mode and Clock Stretching in Write Mode.

Value	Description
0	The clock is not stretched.
1	The clock is stretched. FLEX_TWI_THR / FLEX_TWI_RHR buffer is not filled / emptied before the transmission / reception of a new character.

Bit 9 – ARBLST Arbitration Lost (cleared on read)
This bit is only used in Master mode.

Value	Description
0	Arbitration won.
1	Arbitration lost. Another master of the TWI bus has won the multi-master arbitration. TXCOMP is set at the same time.

Bit 8 – NACK Not Acknowledged (cleared on read)
NACK used in Master mode:
0: Each data byte has been correctly received by the far-end side TWI slave component.
1: A data or address byte has not been acknowledged by the slave component. Set at the same time as TXCOMP.
NACK used in Slave Read mode:
0: Each data byte has been correctly received by the master.
1: In Read mode, a data byte has not been acknowledged by the master. When NACK is set, the user must not fill FLEX_TWI_THR even if TXRDY is set, because it means that the master will stop the data transfer or reinitiate it..
Note that in Slave Write mode, all data are acknowledged by the TWI.

Bit 7 – UNRE Underrun Error (cleared on read)
This bit is only used in Slave mode if clock stretching is disabled.

Value	Description
0	FLEX_TWI_THR has been filled on time.

Downloaded from Arrow.com.

Value	Description
1	FLEX_TWI_THR has not been filled on time.

Bit 6 – OVRE Overrun Error (cleared on read)
This bit is only used in Slave mode if clock stretching is disabled.

Value	Description
0	FLEX_TWI_RHR has not been loaded while RXRDY was set.
1	FLEX_TWI_RHR has been loaded while RXRDY was set. Reset by read in FLEX_TWI_SR when TXCOMP is set.

Bit 5 – GACC General Call Access (cleared on read)
This bit is only used in Slave mode.
GACC behavior can be seen in figure Master Performs a General Call.

Value	Description
0	No general call has been detected.
1	A general call has been detected. After the detection of general call, if need be, the user may acknowledge this access and decode the following bytes and respond according to the value of the bytes.

Bit 4 – SVACC Slave Access
This bit is only used in Slave mode.
SVACC behavior can be seen in figures Read Access Ordered by a Master, Write Access Ordered by a Master, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	TWI is not addressed. SVACC is automatically cleared after a NACK or a STOP condition is detected.
1	Indicates that the address decoding sequence has matched (a master has sent SADR). SVACC remains high until a NACK or a STOP condition is detected.

Bit 3 – SVREAD Slave Read
This bit is only used in Slave mode. When SVACC is low (no slave access has been detected) SVREAD is irrelevant.
SVREAD behavior can be seen in figures Read Access Ordered by a Master, Write Access Ordered by a Master, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	Indicates that a write access is performed by a master.
1	Indicates that a read access is performed by a master.

Bit 2 – TXRDY Transmit Holding Register Ready (cleared by writing FLEX_TWI_THR)
TXRDY used in Master mode:
0: The transmit holding register has not been transferred into the internal shifter. Set to 0 when writing into FLEX_TWI_THR.
1: As soon as a data byte is transferred from FLEX_TWI_THR to internal shifter or if a NACK error is detected, TXRDY is set at the same time as TXCOMP and NACK. TXRDY is also set when MSEN is set (enables TWI).
TXRDY behavior in Master mode can be seen in figures Master Write with One Data Byte,Master Write with Multiple Data Bytes and Master Write with One Byte Internal Address and Multiple Data Bytes.
TXRDY used in Slave mode:
0: As soon as data is written in FLEX_TWI_THR, until this data has been transmitted and acknowledged (ACK or NACK).
1: Indicates that FLEX_TWI_THR is empty and that data has been transmitted and acknowledged.
If TXRDY is high and if a NACK has been detected, the transmission will be stopped. Thus when TRDY = NACK = 1, the user must not fill FLEX_TWI_THR to avoid losing it.
TXRDY behavior in Slave mode can be seen in figures Read Access Ordered by a Master, Clock Stretching in Read Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.
When FIFOs are enabled:
0: Transmit FIFO is full and cannot accept more data.
1: Transmit FIFO is not full; one or more data can be written according to TXRDYM field configuration.

Downloaded from Arrow.com.

TXRDY behavior with FIFOs enabled is illustrated in TXRDY and RXRDY Behavior.

Bit 1 – RXRDY Receive Holding Register Ready (cleared when reading FLEX_TWI_RHR)
When FIFOs are disabled:
0: No character has been received since the last FLEX_TWI_RHR read operation.
1: A byte has been received in FLEX_TWI_RHR since the last read.
RXRDY behavior in Master mode can be seen in figure Master Read with Multiple Data Bytes.
RXRDY behavior in Slave mode can be seen in figures Write Access Ordered by a Master, Clock Stretching in Write Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.
When FIFOs are enabled:
0: Receive FIFO is empty; no data to read.
1: At least one unread data is in the Receive FIFO.
RXRDY behavior with FIFO enabled is illustrated in TXRDY and RXRDY Behavior.

Bit 0 – TXCOMP Transmission Completed (cleared by writing FLEX_TWI_THR)
TXCOMP used in Master mode:
0: During the length of the current frame.
1: When both the holding register and the internal shifter are empty and STOP condition has been sent.
TXCOMP behavior in Master mode can be seen in figures Master Write with One Byte Internal Address and Multiple Data Bytes and Master Read with Multiple Data Bytes.
TXCOMP used in Slave mode:
0: As soon as a Start is detected.
1: After a Stop or a Repeated Start + an address different from SADR is detected.
TXCOMP behavior in Slave mode can be seen in figures Clock Stretching in Read Mode, Clock Stretching in Write Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Downloaded from Arrow.com.

45.10.78 TWI Status Register (FIFO ENABLED)

Name: FLEX_TWI_SR (FIFO_ENABLED)
Offset: 0x620
Reset: 0x0300F009
Property: Read-only

If FIFO is enabled (FLEX_US_CR.FIFOEN bit), see TWI Multiple Data Mode for details.

Bit	31	30	29	28	27	26	25	24
						SR	SDA	SCL
Access						R	R	R
Reset						0	1	1

Bit	23	22	21	20	19	18	17	16
	TXFLOCK		SMBHHM	SMBDAM	PECERR	TOUT		MCACK
Access	R		R	R	R	R		R
Reset	0		0	0	0	0		0

Bit	15	14	13	12	11	10	9	8
					EOSACC	SCLWS	ARBLST	NACK
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UNRE	OVRE	GACC	SVACC	SVREAD	TXRDY	RXRDY	TXCOMP
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	0	0	1

Bit 26 – SR Start Repeated

Value	Description
0	No repeated start has been detected since last FLEX_TWI_SR read.
1	At least one repeated start has been detected since last FLEX_TWI_SR read.

Bit 25 – SDA SDA Line Value

Value	Description
0	SDA line sampled value is ‘0’.
1	SDA line sampled value is ‘1’.

Bit 24 – SCL SCL Line Value

Value	Description
0	SCL line sampled value is ‘0’.
1	SCL line sampled value is ‘1.’

Bit 23 – TXFLOCK Transmit FIFO Lock

Value	Description
0	The Transmit FIFO is not locked.
1	The Transmit FIFO is locked.

Bit 21 – SMBHHM SMBus Host Header Address Match (cleared on read)

Value	Description
0	No SMBus Host Header Address received.
1	A SMBus Host Header Address was received.

Bit 20 – SMBDAM SMBus Default Address Match (cleared on read)

Downloaded from Arrow.com.

Value	Description
0	No SMBus Default Address received.
1	A SMBus Default Address was received.

Bit 19 – PECERR PEC Error (cleared on read)

Value	Description
0	No SMBus PEC error occurred.
1	A SMBus PEC error occurred.

Bit 18 – TOUT Timeout Error (cleared on read)

Value	Description
0	No SMBus timeout occurred.
1	SMBus timeout occurred.

Bit 16 – MCACK Master Code Acknowledge (cleared on read)
MACK used in Slave mode:

Value	Description
0	No master code has been received.
1	A master code has been received.

Bit 11 – EOSACC End Of Slave Access (cleared on read)
This bit is only used in Slave mode.
EOSACC behavior can be seen in figures Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	A Slave Access is being performed.
1	The Slave Access is finished. End Of Slave Access is automatically set as soon as SVACC is reset.

Bit 10 – SCLWS Clock Wait State
This bit is only used in Slave mode.
SCLWS behavior can be seen in figures Clock Stretching in Read Mode and Clock Stretching in Write Mode.

Value	Description
0	The clock is not stretched.
1	The clock is stretched. FLEX_TWI_THR / FLEX_TWI_RHR buffer is not filled / emptied before the transmission / reception of a new character.

Bit 9 – ARBLST Arbitration Lost (cleared on read)
This bit is only used in Master mode.

Value	Description
0	Arbitration won.
1	Arbitration lost. Another master of the TWI bus has won the multi-master arbitration. TXCOMP is set at the same time.

Bit 8 – NACK Not Acknowledged (cleared on read)
NACK used in Master mode:
0: Each data byte has been correctly received by the far-end side TWI slave component.
1: A data or address byte has not been acknowledged by the slave component. Set at the same time as TXCOMP.
NACK used in Slave Read mode:
0: Each data byte has been correctly received by the master.
1: In Read mode, a data byte has not been acknowledged by the master. When NACK is set the user must not fill FLEX_TWI_THR even if TXRDY is set, because it means that the master will stop the data transfer or re initiate it. Note that in Slave Write mode all data are acknowledged by the TWI.

Bit 7 – UNRE Underrun Error (cleared on read)
This bit is only used in Slave mode if clock stretching is disabled.

Value	Description
0	FLEX_TWI_THR has been filled on time.

Downloaded from Arrow.com.

Value	Description
1	FLEX_TWI_THR has not been filled on time.

Bit 6 – OVRE Overrun Error (cleared on read)
This bit is only used in Slave mode if clock stretching is disabled.

Value	Description
0	FLEX_TWI_RHR has not been loaded while RXRDY was set.
1	FLEX_TWI_RHR has been loaded while RXRDY was set. Reset by read in FLEX_TWI_SR when TXCOMP is set.

Bit 5 – GACC General Call Access (cleared on read)
This bit is only used in Slave mode.
GACC behavior can be seen in figure Master Performs a General Call.

Value	Description
0	No general call has been detected.
1	A general call has been detected. After the detection of general call, if need be, the user may acknowledge this access and decode the following bytes and respond according to the value of the bytes.

Bit 4 – SVACC Slave Access
This bit is only used in Slave mode.
SVACC behavior can be seen in figures Read Access Ordered by a Master, Write Access Ordered by a Master, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	TWI is not addressed. SVACC is automatically cleared after a NACK or a STOP condition is detected.
1	Indicates that the address decoding sequence has matched (a master has sent SADR). SVACC remains high until a NACK or a STOP condition is detected.

Bit 3 – SVREAD Slave Read
This bit is only used in Slave mode. When SVACC is low (no slave access has been detected) SVREAD is irrelevant.
SVREAD behavior can be seen in figures Read Access Ordered by a Master, Write Access Ordered by a Master, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	Indicates that a write access is performed by a master.
1	Indicates that a read access is performed by a master.

Bit 2 – TXRDY Transmit Holding Register Ready (cleared by writing FLEX_TWI_THR)
TXRDY used in Master mode:
0: The transmit holding register has not been transferred into the internal shifter. Set to 0 when writing into FLEX_TWI_THR.
1: As soon as a data byte is transferred from FLEX_TWI_THR to internal shifter or if a NACK error is detected, TXRDY is set at the same time as TXCOMP and NACK. TXRDY is also set when MSEN is set (enables TWI).
TXRDY behavior in Master mode can be seen in figures Master Write with One Data Byte,Master Write with Multiple Data Bytes and Master Write with One Byte Internal Address and Multiple Data Bytes.
TXRDY used in Slave mode:
0: As soon as data is written in FLEX_TWI_THR, until this data has been transmitted and acknowledged (ACK or NACK).
1: Indicates that FLEX_TWI_THR is empty and that data has been transmitted and acknowledged.
If TXRDY is high and if a NACK has been detected, the transmission will be stopped. Thus when TRDY = NACK = 1, the user must not fill FLEX_TWI_THR to avoid losing it.
TXRDY behavior in Slave mode can be seen in figures Read Access Ordered by a Master, Clock Stretching in Read Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.
When FIFOs are enabled:
0: Transmit FIFO is full and cannot accept more data.
1: Transmit FIFO is not full; one or more data can be written according to TXRDYM field configuration.

Downloaded from Arrow.com.

TXRDY behavior with FIFOs enabled is illustrated in TXRDY and RXRDY Behavior.

Bit 1 – RXRDY Receive Holding Register Ready (cleared when reading FLEX_TWI_RHR)
When FIFOs are disabled:
0: No character has been received since the last FLEX_TWI_RHR read operation.
1: A byte has been received in FLEX_TWI_RHR since the last read.
RXRDY behavior in Master mode can be seen in figure Master Read with Multiple Data Bytes.
RXRDY behavior in Slave mode can be seen in figures Write Access Ordered by a Master, Clock Stretching in Write Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.
When FIFOs are enabled:
0: Receive FIFO is empty; no data to read.
1: At least one unread data is in the Receive FIFO.
RXRDY behavior with FIFO enabled is illustrated in TXRDY and RXRDY Behavior.

Bit 0 – TXCOMP Transmission Completed (cleared by writing FLEX_TWI_THR)
TXCOMP used in Master mode:
0: During the length of the current frame.
1: When both holding register and internal shifter are empty and STOP condition has been sent.
TXCOMP behavior in Master mode can be seen in figures Master Write with One Byte Internal Address and Multiple Data Bytes and Master Read with Multiple Data Bytes.
TXCOMP used in Slave mode:
0: As soon as a Start is detected.
1: After a Stop or a Repeated Start + an address different from SADR is detected.
TXCOMP behavior in Slave mode can be seen in figures Clock Stretching in Read Mode, Clock Stretching in Write Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Downloaded from Arrow.com.

45.10.79 TWI Interrupt Enable Register

Name: FLEX_TWI_IER
Offset: 0x624
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TWI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
Access			W	W	W	W		W
Reset			–	–	–	–		–

Bit	15	14	13	12	11	10	9	8
	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
Access	W	W	W	W		W	W	W
Reset	–	–	–	–		–	–	–

Bit 21 – SMBHHM SMBus Host Header Address Match Interrupt Enable

Bit 20 – SMBDAM SMBus Default Address Match Interrupt Enable

Bit 19 – PECERR PEC Error Interrupt Enable

Bit 18 – TOUT Timeout Error Interrupt Enable

Bit 16 – MCACK Master Code Acknowledge Interrupt Enable

Bit 15 – TXBUFE Transmit Buffer Empty Interrupt Enable

Bit 14 – RXBUFF Receive Buffer Full Interrupt Enable

Bit 13 – ENDTX End of Transmit Buffer Interrupt Enable

Bit 12 – ENDRX End of Receive Buffer Interrupt Enable

Bit 11 – EOSACC End Of Slave Access Interrupt Enable

Bit 10 – SCL_WS Clock Wait State Interrupt Enable

Bit 9 – ARBLST Arbitration Lost Interrupt Enable

Downloaded from Arrow.com.

Bit 8 – NACK Not Acknowledge Interrupt Enable

Bit 7 – UNRE Underrun Error Interrupt Enable

Bit 6 – OVRE Overrun Error Interrupt Enable

Bit 5 – GACC General Call Access Interrupt Enable

Bit 4 – SVACC Slave Access Interrupt Enable

Bit 2 – TXRDY Transmit Holding Register Ready Interrupt Enable

Bit 1 – RXRDY Receive Holding Register Ready Interrupt Enable

Bit 0 – TXCOMP Transmission Completed Interrupt Enable

Downloaded from Arrow.com.

45.10.80 TWI Interrupt Disable Register

Name: FLEX_TWI_IDR
Offset: 0x628
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TWI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
Access			W	W	W	W		W
Reset			–	–	–	–		–

Bit	15	14	13	12	11	10	9	8
	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
Access	W	W	W	W		W	W	W
Reset	–	–	–	–		–	–	–

Bit 21 – SMBHHM SMBus Host Header Address Match Interrupt Disable

Bit 20 – SMBDAM SMBus Default Address Match Interrupt Disable

Bit 19 – PECERR PEC Error Interrupt Disable

Bit 18 – TOUT Timeout Error Interrupt Disable

Bit 16 – MCACK Master Code Acknowledge Interrupt Disable

Bit 15 – TXBUFE Transmit Buffer Empty Interrupt Disable

Bit 14 – RXBUFF Receive Buffer Full Interrupt Disable

Bit 13 – ENDTX End of Transmit Buffer Interrupt Disable

Bit 12 – ENDRX End of Receive Buffer Interrupt Disable

Bit 11 – EOSACC End Of Slave Access Interrupt Disable

Bit 10 – SCL_WS Clock Wait State Interrupt Disable

Bit 9 – ARBLST Arbitration Lost Interrupt Disable

Downloaded from Arrow.com.

Bit 8 – NACK Not Acknowledge Interrupt Disable

Bit 7 – UNRE Underrun Error Interrupt Disable

Bit 6 – OVRE Overrun Error Interrupt Disable

Bit 5 – GACC General Call Access Interrupt Disable

Bit 4 – SVACC Slave Access Interrupt Disable

Bit 2 – TXRDY Transmit Holding Register Ready Interrupt Disable

Bit 1 – RXRDY Receive Holding Register Ready Interrupt Disable

Bit 0 – TXCOMP Transmission Completed Interrupt Disable

Downloaded from Arrow.com.

45.10.81 TWI Interrupt Mask Register

Name: FLEX_TWI_IMR
Offset: 0x62C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
Access			R	R	R	R		R
Reset			0	0	0	0		0

Bit	15	14	13	12	11	10	9	8
	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
Access	R	R	R	R		R	R	R
Reset	0	0	0	0		0	0	0

Bit 21 – SMBHHM SMBus Host Header Address Match Interrupt Mask

Bit 20 – SMBDAM SMBus Default Address Match Interrupt Mask

Bit 19 – PECERR PEC Error Interrupt Mask

Bit 18 – TOUT Timeout Error Interrupt Mask

Bit 16 – MCACK Master Code Acknowledge Interrupt Mask

Bit 15 – TXBUFE Transmit Buffer Empty Interrupt Mask

Bit 14 – RXBUFF Receive Buffer Full Interrupt Mask

Bit 13 – ENDTX End of Transmit Buffer Interrupt Mask

Bit 12 – ENDRX End of Receive Buffer Interrupt Mask

Bit 11 – EOSACC End Of Slave Access Interrupt Mask

Bit 10 – SCL_WS Clock Wait State Interrupt Mask

Bit 9 – ARBLST Arbitration Lost Interrupt Mask

Downloaded from Arrow.com.

Bit 8 – NACK Not Acknowledge Interrupt Mask

Bit 7 – UNRE Underrun Error Interrupt Mask

Bit 6 – OVRE Overrun Error Interrupt Mask

Bit 5 – GACC General Call Access Interrupt Mask

Bit 4 – SVACC Slave Access Interrupt Mask

Bit 2 – TXRDY Transmit Holding Register Ready Interrupt Mask

Bit 1 – RXRDY Receive Holding Register Ready Interrupt Mask

Bit 0 – TXCOMP Transmission Completed Interrupt Mask

Downloaded from Arrow.com.

45.10.82 TWI Receive Holding Register

Name: FLEX_TWI_RHR
Offset: 0x630
Reset: 0x00000000
Property: Read-only

If FIFO is enabled (FLEX_TWI_CR.FIFOEN bit) and FLEX_TWI_FMR.RXRDYM = 0, see TWI Single Data Mode for details.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				ASTATE[1:0]		PSTATE	SSTATE[1:0]	
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 12:11 – ASTATE[1:0] Acknowledge State (Slave Sniffer Mode only)

Value	Name	Description
0	NONE	No Acknowledge or Nacknowledge detected after previously logged data
1	ACK	Acknowledge (A) detected after previously logged data
2	NACK	Nacknowledge (NA) detected after previously logged data
3	UNDEF	Not defined

Bit 10 – PSTATE Stop State (Slave Sniffer Mode only)

Value	Description
0	No STOP (P) detected after previous logged data.
1	Stop detected (P) after previous logged data.

Bits 9:8 – SSTATE[1:0] Start State (Slave Sniffer Mode only)

Value	Name	Description
0	NOSTART	No START detected with the logged data
1	START	START (S) detected with the logged data
2	RSTART	Repeated START (Sr) detected with the logged data
3	UNDEF	Not defined

Bits 7:0 – RXDATA[7:0] Master or Slave Receive Holding Data

Downloaded from Arrow.com.

45.10.83 TWI Receive Holding Register (FIFO Enabled)

Name: FLEX_TWI_RHR (FIFO_ENABLED)
Offset: 0x630
Reset: 0x00000000
Property: Read-only

If FIFO is enabled (FLEX_TWI_CR.FIFOEN bit) and FLEX_TWI_FMR.RXRDYM > 0, see TWI Multiple Data Mode for details.

Bit	31	30	29	28	27	26	25	24
	RXDATA3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RXDATA2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RXDATA1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXDATA0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – RXDATA3[7:0] Master or Slave Receive Holding Data 3

Bits 23:16 – RXDATA2[7:0] Master or Slave Receive Holding Data 2

Bits 15:8 – RXDATA1[7:0] Master or Slave Receive Holding Data 1

Bits 7:0 – RXDATA0[7:0] Master or Slave Receive Holding Data 0

Downloaded from Arrow.com.

45.10.84 TWI Transmit Holding Register

Name: FLEX_TWI_THR
Offset: 0x634
Reset: –
Property: Write-only

If FIFO is enabled (FIFOEN bit in FLEX_TWI_CR) and FLEX_TWI_FMR.TXRDYM = 0, see TWI Single Data Mode for details.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	TXDATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 7:0 – TXDATA[7:0] Master or Slave Transmit Holding Data

Downloaded from Arrow.com.

45.10.85 TWI Transmit Holding Register (FIFO Enabled)

Name: FLEX_TWI_THR (FIFO_ENABLED)
Offset: 0x634
Reset: –
Property: Write-only

If FIFO is enabled (FLEX_US_CR.FIFOEN bit) and FLEX_TWI_FMR.TXRDYM > 0, see TWI Multiple Data Mode for details.

Bit	31	30	29	28	27	26	25	24
	TXDATA3[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	TXDATA2[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TXDATA1[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TXDATA0[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:24 – TXDATA3[7:0] Master or Slave Transmit Holding Data 3

Bits 23:16 – TXDATA2[7:0] Master or Slave Transmit Holding Data 2

Bits 15:8 – TXDATA1[7:0] Master or Slave Transmit Holding Data 1

Bits 7:0 – TXDATA0[7:0] Master or Slave Transmit Holding Data 0

Downloaded from Arrow.com.

45.10.86 TWI SMBus Timing Register

Name: FLEX_TWI_SMBTR
Offset: 0x638
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	THMAX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TLOWM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TLOWS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					PRESC[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 31:24 – THMAX[7:0] Clock High Maximum Cycles
Clock cycles in clock high maximum count. Prescaled by PRESC. Used for bus free detection. Used to time THIGH:MAX.

Bits 23:16 – TLOWM[7:0] Master Clock Stretch Maximum Cycles

Value	Description
0	TLOW:MEXT timeout check disabled.
1-255	Clock cycles in master maximum clock stretch count. Prescaled by PRESC. Used to time TLOW:MEXT.

Bits 15:8 – TLOWS[7:0] Slave Clock Stretch Maximum Cycles

Value	Description
0	TLOW:SEXT timeout check disabled.
1-255	Clock cycles in slave maximum clock stretch count. Prescaled by PRESC. Used to time TLOW:SEXT.

Bits 3:0 – PRESC[3:0] SMBus Clock Prescaler
Used to specify how to prescale the TLOWS, TLOWM and THMAX counters in SMBTR. Counters are prescaled according to the following formula: PRESC = Log(fMCK / $f_{Prescaled}$) / Log(2) - 1

Downloaded from Arrow.com.

45.10.87 TWI Alternative Command Register

Name: FLEX_TWI_ACR
Offset: 0x640
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							NPEC	NDIR
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	NDATAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
							PEC	DIR
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	DATAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 25 – NPEC Next PEC Request (SMBus Mode only)

Value	Description
0	The next transfer does not use a PEC byte.
1	The next transfer uses a PEC byte.

Bit 24 – NDIR Next Transfer Direction

Value	Description
0	Write direction.
1	Read direction.

Bits 23:16 – NDATAL[7:0] Next Data Length

Value	Description
0	No data to send (see 45.9.3.11 Alternative Command).
1-255	Number of bytes to send for the next transfer.

Bit 9 – PEC PEC Request (SMBus Mode only)

Value	Description
0	The transfer does not use a PEC byte.
1	The transfer uses a PEC byte.

Bit 8 – DIR Transfer Direction

Value	Description
0	Write direction.
1	Read direction.

Bits 7:0 – DATAL[7:0] Data Length

Value	Description
0	No data to send (see 45.9.3.11 Alternative Command).

Downloaded from Arrow.com.

Value	Description
1-255	Number of bytes to send during the transfer.

Downloaded from Arrow.com.

45.10.88 TWI Filter Register

Name: FLEX_TWI_FILTR
Offset: 0x644
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						THRES[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
							PADFEN	FILT
Access							R/W	R/W
Reset							0	0

Bits 10:8 – THRES[2:0] Digital Filter Threshold

Value	Description
0	No filtering applied on TWI inputs.
1-7	Maximum pulse width of spikes which will be suppressed by the input filter, defined in peripheral clock cycles.

Bit 1 – PADFEN PAD Filter Enable

Value	Description
0	PAD analog filter is disabled.
1	PAD analog filter is enabled. (The analog filter must be enabled if High-speed mode is enabled.)

Bit 0 – FILT RX Digital Filter
TWI digital input filtering follows a majority decision based on three samples from SDA/SCL lines at peripheral clock frequency.

Value	Description
0	No filtering applied on TWI inputs.
1	TWI input filtering is active. (Only in Standard and Fast modes)

Downloaded from Arrow.com.

45.10.89 TWI FIFO Mode Register

Name: FLEX_TWI_FMR
Offset: 0x650
Reset: 0x00000000
Property: Read/Write

This register reads '0' if the FIFO is disabled (see FLEX_TWI_CR to enable/disable the internal FIFO).
This register can only be written if the WPEN bit is cleared in the TWI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
			RXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			TXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RXRDYM[1:0]				TXRDYM[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 29:24 – RXFTHRES[5:0] Receive FIFO Threshold

Value	Description
0-16	Defines the Receive FIFO threshold value (number of data). The FLEX_TWI_FSR.RXFTH flag will be set when Receive FIFO goes from "below" threshold state to "equal to or above" threshold state.

Bits 21:16 – TXFTHRES[5:0] Transmit FIFO Threshold

Value	Description
0-16	Defines the Transmit FIFO threshold value (number of data). The FLEX_TWI_FSR.TXFTH flag will be set when Transmit FIFO goes from "above" threshold state to "equal to or below" threshold state.

Bits 5:4 – RXRDYM[1:0] Receiver Ready Mode
If FIFOs are enabled, the FLEX_TWI_SR.RXRDY flag behaves as follows.

Value	Name	Description
0	ONE_DATA	RXRDY will be at level '1' when at least one unread data is in the Receive FIFO
1	TWO_DATA	RXRDY will be at level '1' when at least two unread data are in the Receive FIFO
2	FOUR_DATA	RXRDY will be at level '1' when at least four unread data are in the Receive FIFO

Bits 1:0 – TXRDYM[1:0] Transmitter Ready Mode
If FIFOs are enabled, the FLEX_TWI_SR.TXRDY flag behaves as follows.

Value	Name	Description
0	ONE_DATA	TXRDY will be at level '1' when at least one data can be written in the Transmit FIFO
1	TWO_DATA	TXRDY will be at level '1' when at least two data can be written in the Transmit FIFO
2	FOUR_DATA	TXRDY will be at level '1' when at least four data can be written in the Transmit FIFO

Downloaded from Arrow.com.

45.10.90 TWI FIFO Level Register

Name: FLEX_TWI_FLR
Offset: 0x654
Reset: 0x00000000
Property: Read-only

This register reads '0' if the FIFO is disabled (see FLEX_TWI_CR to enable/disable the internal FIFO).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			RXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bits 21:16 – RXFL[5:0] Receive FIFO Level

Value	Description
0	There is no unread data in the Receive FIFO.
1-16	Indicates the number of unread data in the Receive FIFO.

Bits 5:0 – TXFL[5:0] Transmit FIFO Level

Value	Description
0	There is no data in the Transmit FIFO.
1-16	Indicates the number of data in the Transmit FIFO.

Downloaded from Arrow.com.

45.10.91 TWI FIFO Status Register

Name: FLEX_TWI_FSR
Offset: 0x660
Reset: 0x00000000
Property: Read-only

This register reads '0' if the FIFO is disabled (see FLEX_TWI_CR to enable/disable the internal FIFO)

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – RXFPTEF Receive FIFO Pointer Error Flag
See 45.9.6.10 FIFO Pointer Error for details.

Value	Description
0	No Receive FIFO pointer occurred.
1	Receive FIFO pointer error occurred. Receiver must be reset.

Bit 6 – TXFPTEF Transmit FIFO Pointer Error Flag
See 45.9.6.10 FIFO Pointer Error for details.

Value	Description
0	No Transmit FIFO pointer occurred.
1	Transmit FIFO pointer error occurred. Transceiver must be reset.

Bit 5 – RXFTHF Receive FIFO Threshold Flag

Value	Description
0	Number of unread data in Receive FIFO is below RXFTHRES threshold.
1	Number of unread data in Receive FIFO has reached RXFTHRES threshold since the last read of FLEX_TWI_FSR.

Bit 4 – RXFFF Receive FIFO Full Flag

Value	Description
0	Receive FIFO is not empty.
1	Receive FIFO has been filled since the last read of FLEX_TWI_FSR.

Bit 3 – RXFEF Receive FIFO Empty Flag

Value	Description
0	Receive FIFO is not empty.
1	Receive FIFO has been emptied since the last read of FLEX_TWI_FSR.

Downloaded from Arrow.com.

Bit 2 – TXFTHF Transmit FIFO Threshold Flag (cleared on read)

Value	Description
0	Number of data in Transmit FIFO is above TXFTHRES threshold.
1	Number of data in Transmit FIFO has reached TXFTHRES threshold since the last read of FLEX_TWI_FSR.

Bit 1 – TXFFF Transmit FIFO Full Flag (cleared on read)

Value	Description
0	Transmit FIFO is not full.
1	Transmit FIFO has been filled since the last read of FLEX_TWI_FSR.

Bit 0 – TXFEF Transmit FIFO Empty Flag (cleared on read)

Value	Description
0	Transmit FIFO is not empty.
1	Transmit FIFO has been emptied since the last read of FLEX_TWI_FSR.

Downloaded from Arrow.com.

45.10.92 TWI FIFO Interrupt Enable Register

Name: FLEX_TWI_FIER
Offset: 0x664
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TWI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 7 – RXFPTEF RXFPTEF Interrupt Enable

Bit 6 – TXFPTEF TXFPTEF Interrupt Enable

Bit 5 – RXFTHF RXFTHF Interrupt Enable

Bit 4 – RXFFF RXFFF Interrupt Enable

Bit 3 – RXFEF RXFEF Interrupt Enable

Bit 2 – TXFTHF TXFTHF Interrupt Enable

Bit 1 – TXFFF TXFFF Interrupt Enable

Bit 0 – TXFEF TXFEF Interrupt Enable

Downloaded from Arrow.com.

45.10.93 TWI FIFO Interrupt Disable Register

Name: FLEX_TWI_FIDR
Offset: 0x668
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TWI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 7 – RXFPTEF RXFPTEF Interrupt Disable

Bit 6 – TXFPTEF TXFPTEF Interrupt Disable

Bit 5 – RXFTHF RXFTHF Interrupt Disable

Bit 4 – RXFFF RXFFF Interrupt Disable

Bit 3 – RXFEF RXFEF Interrupt Disable

Bit 2 – TXFTHF TXFTHF Interrupt Disable

Bit 1 – TXFFF TXFFF Interrupt Disable

Bit 0 – TXFEF TXFEF Interrupt Disable

Downloaded from Arrow.com.

45.10.94 TWI FIFO Interrupt Mask Register

Name: FLEX_TWI_FIMR
Offset: 0x66C
Reset: 0x00000000
Property: Read-only

This register reads '0' if the FIFO is disabled (see FLEX_TWI_CR to enable/disable the internal FIFO).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – RXFPTEF RXFPTEF Interrupt Mask

Bit 6 – TXFPTEF TXFPTEF Interrupt Mask

Bit 5 – RXFTHF RXFTHF Interrupt Mask

Bit 4 – RXFFF RXFFF Interrupt Mask

Bit 3 – RXFEF RXFEF Interrupt Mask

Bit 2 – TXFTHF TXFTHF Interrupt Mask

Bit 1 – TXFFF TXFFF Interrupt Mask

Bit 0 – TXFEF TXFEF Interrupt Mask

Downloaded from Arrow.com.

45.10.95 TWI Write Protection Mode Register

Name: FLEX_TWI_WPMR
Offset: 0x6E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						WPCREN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x545749	PASSWD	Writing any other value in this field aborts the write operation of bits WPEN, WPITEN and WPCREN. Always reads as 0.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x545749 (“TWI” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x545749 (“TWI” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x545749 (“TWI” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x545749 (“TWI” in ASCII).

Bit 0 – WPEN Write Protection Enable
See TWI Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x545749 (“TWI” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x545749 (“TWI” in ASCII).

Downloaded from Arrow.com.

45.10.96 TWI Write Protection Status Register

Name: FLEX_TWI_WPSR
Offset: 0x6E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	WPVSRC[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 31:8 – WPVSRC[23:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protect Violation Status

Value	Description
0	No Write Protection Violation has occurred since the last read of FLEX_TWI_WPSR.
1	A Write Protection Violation has occurred since the last read of FLEX_TWI_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

46. Quad Serial Peripheral Interface (QSPI)

46.1 Description

The Quad Serial Peripheral Interface (QSPI) is a synchronous serial data link that provides communication with external devices in Master mode.

The QSPI can be used in SPI mode to interface to serial peripherals such as ADCs, DACs, LCD controllers, CAN controllers and sensors, or in Serial Memory mode to interface to serial Flash memories.

The QSPI allows the system to execute code directly from a serial Flash memory (XIP) without code shadowing to RAM. The serial Flash memory mapping is seen in the system as other memories such as ROM, SRAM, DRAM, embedded Flash memory, etc.

With the support of the Quad SPI protocol, the QSPI allows the system to use high-performance serial Flash memories which are small and inexpensive, in place of larger and more expensive parallel Flash memories.

Note: Stacked devices with a rollover in the memory address space at each die boundary are not supported.

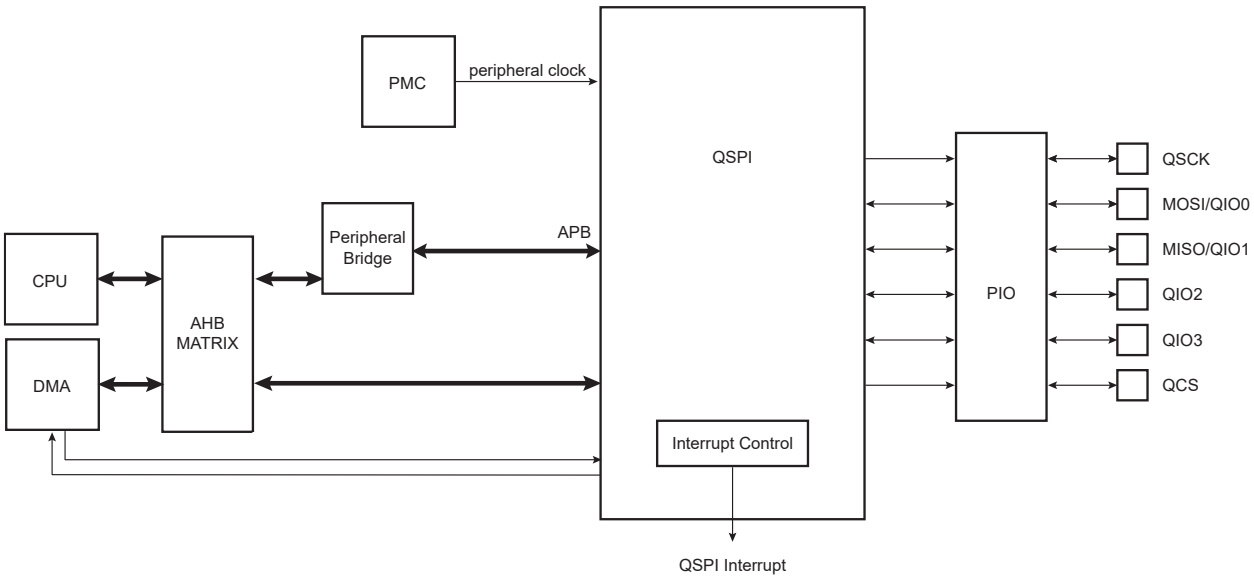
46.2 Embedded Characteristics

- Master SPI Interface
  - Programmable clock phase and clock polarity
  - Programmable transfer delays between consecutive transfers, between clock and data, between deactivation and activation of chip select
- SPI Mode
  - Interface to serial peripherals such as ADCs, DACs, LCD controllers, CAN controllers and sensors
  - 8-bit/16-bit/32-bit programmable data length
- Serial Memory Mode
  - Interface to serial Flash memories operating in Single-bit SPI, Dual SPI and Quad SPI
  - Interface to serial Flash Memories operating in Single Data Rate or Double Data Rate Modes
  - Supports “Execute In Place” (XIP)— code execution by the system directly from a serial Flash memory
  - Flexible instruction register for compatibility with all serial Flash memories
  - 32-bit address mode (default is 24-bit address) to support serial Flash memories larger than 128 Mbits
  - Continuous read mode
  - Scrambling/unscrambling “On-The-Fly”
- Connection to DMA Channel Capabilities Optimizes Data Transfers
  - One channel for the receiver, one channel for the transmitter
- Register Write Protection

Downloaded from Arrow.com.

46.3 Block Diagram

Figure 46-1. Block Diagram



46.4 Signal Description

Table 46-1. Signal Description

Pin Name	Pin Description	Type
QSCK	Serial Clock	Output
MOSI (QIO0) (1)(2)	Data Output (Data Input Output 0)	Output (Input/Output)
MISO (QIO1) (1)(2)	Data Input (Data Input Output 1)	Input (Input/Output)
QIO2 (3)	Data Input Output 2	Input/Output
QIO3 (3)	Data Input Output 3	Input/Output
QCS	Peripheral Chip Select	Output

Note:

1. MOSI and MISO are used for single-bit SPI operation.
2. QIO0–QIO1 are used for Dual SPI operation.
3. QIO0–QIO3 are used for Quad SPI operation.

46.5 Product Dependencies

46.5.1 I/O Lines

The pins used for interfacing the compliant external devices may be multiplexed with PIO lines. The programmer must first program the PIO controllers to assign the QSPI pins to their peripheral functions.

46.5.2 Power Management

The QSPI may be clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the QSPI clock.

Downloaded from Arrow.com.

46.5.3 Interrupt Sources

The QSPI has an interrupt line connected to the Interrupt Controller. Handling the QSPI interrupt requires programming the interrupt controller before configuring the QSPI.

46.5.4 Direct Memory Access Controller (DMA)

The QSPI can be used in conjunction with the Direct Memory Access Controller (DMA) in order to reduce processor overhead. For a full description of the DMA, refer to the section “DMA Controller (XDMAC)”.

46.6 Functional Description

46.6.1 Serial Clock Baud Rate

The QSPI baud rate clock is generated by dividing the peripheral clock by a value between 1 and 256.

46.6.2 Serial Clock Phase and Polarity

Four combinations of polarity and phase are available for data transfers. The clock polarity is programmed with the CPOL bit in the QSPI Serial Clock register (QSPI_SCR). The CPHA bit in the QSPI_SCR programs the clock phase. These two parameters determine the edges of the clock signal on which data is driven and sampled. Each of the two parameters has two possible states, resulting in four possible combinations that are incompatible with one another. Thus, the interfaced slave must use the same parameter values to communicate.

The table below shows the four modes and corresponding parameter settings.

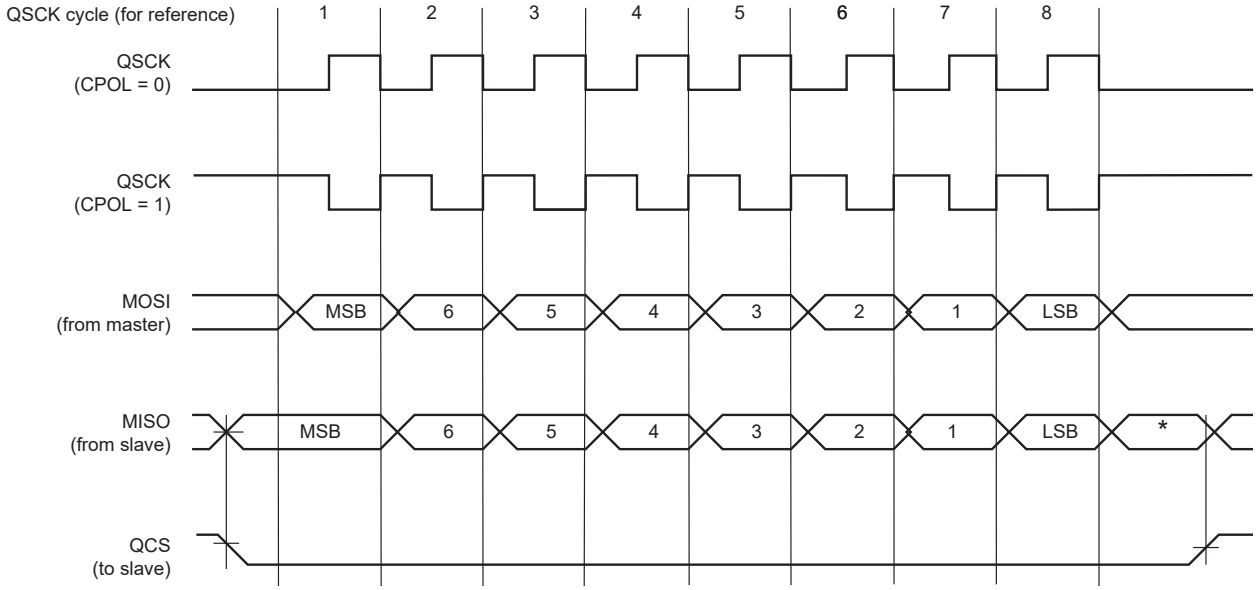
Table 46-2. QSPI Bus Clock Modes

QSPI Clock Mode	QSPI_SCR.CPOL	QSPI_SCR.CPHA	Shift QSCK Edge	Capture QSCK Edge	QSCK Inactive Level
0	0	0	Falling	Falling	Low
1	0	1	Rising	Rising	Low
2	1	0	Rising	Rising	High
3	1	1	Falling	Falling	High

The following figures show examples of data transfers.

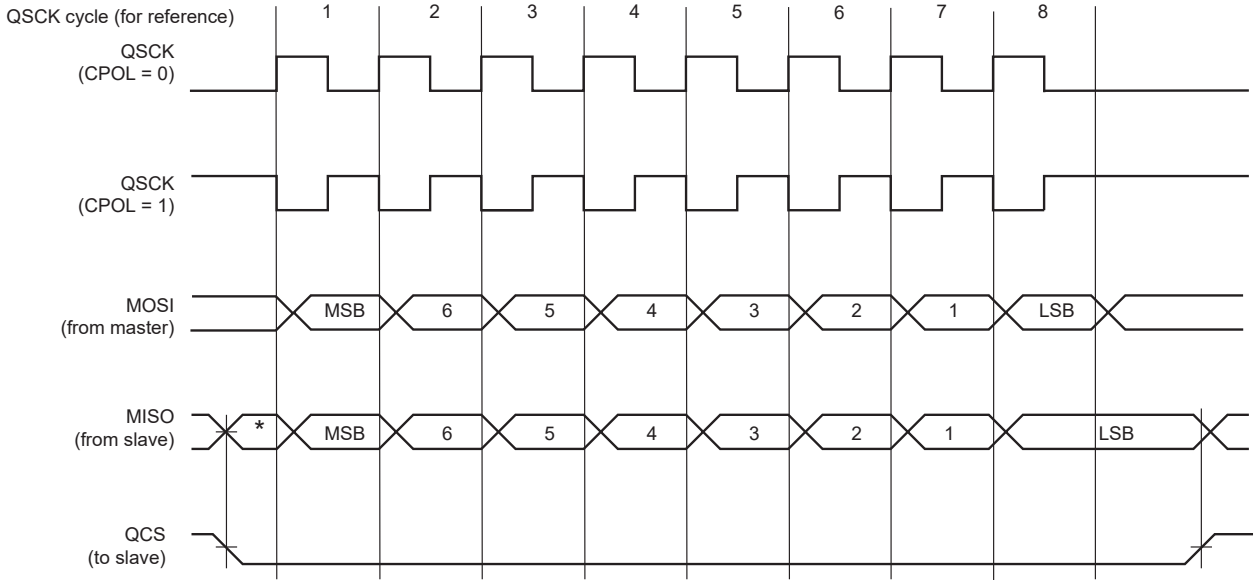
Downloaded from Arrow.com.

Figure 46-2. QSPI Transfer Format (QSPI_SCR.CPHA = 0, 8 bits per transfer)



* Not defined, but normally MSB of previous character received.

Figure 46-3. QSPI Transfer Format (QSPI_SCR.CPHA = 1, 8 bits per transfer)



* Not defined but normally LSB of previous character transmitted.

46.6.3 Transfer Delays

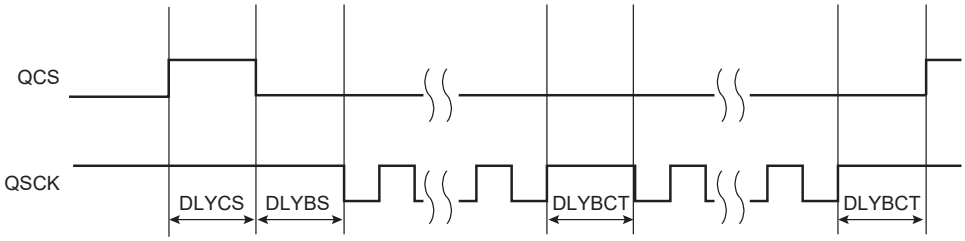
The figure below shows several consecutive transfers while the chip select is active. Three delays can be programmed to modify the transfer waveforms:

- The delay between the deactivation and the activation of QCS, programmed by writing QSPI_MR.DLYCS. Allows to adjust the minimum time of QCS at high level.
- The delay before QSCK, programmed by writing QSPI_SR.DLYBS. Allows the start of QSCK to be delayed after the chip select has been asserted.
- The delay between consecutive transfers, programmed by writing QSPI_MR.DLYBCT. Allows insertion of a delay between two consecutive transfers. In Serial Memory mode, this delay is not programmable and DLYBCT is ignored. In this mode, DLYBCT must be written to ‘0’.

Downloaded from Arrow.com.

These delays allow the QSPI to be adapted to the interfaced peripherals and their speed and bus release time.

Figure 46-4. Programmable Delays



46.6.4 QSPI SPI Mode

In SPI mode, the QSPI acts as a standard SPI Master.

To activate this mode, QSPI_MR.SMM must be written to ‘0’ in QSPI_MR.

46.6.4.1 SPI Mode Operations

The QSPI in standard SPI mode operates on the clock generated by the internal programmable baud rate generator. It fully controls the data transfers to and from the slave connected to the SPI bus. The QSPI drives the chip select line to the slave (QCS) and the serial clock signal (QSCK).

The QSPI features two holding registers, the Transmit Data register (QSPI_TDR) and the Receive Data register (QSPI_RDR), and a single internal shift register. The holding registers maintain the data flow at a constant rate.

After enabling the QSPI, a data transfer begins when the processor writes to the QSPI_TDR. The written data is immediately transferred to the internal shift register and transfer on the SPI bus starts. While the data in the internal shift register is shifted on the MOSI line, the MISO line is sampled and shifted to the internal shift register. Receiving data cannot occur without transmitting data. If receiving mode is not needed, for example when communicating with a slave receiver only (such as an LCD), the receive status flags in the Status register (QSPI_SR) can be discarded.

If new data is written in QSPI_TDR during the transfer, it is retained there until the current transfer is completed. Then, the received data is transferred from the internal shift register to the QSPI_RDR, the data in QSPI_TDR is loaded in the internal shift register and a new transfer starts.

The transfer of a data written in QSPI_TDR in the internal shift register is indicated by the Transmit Data Register Empty (TDRE) bit in the QSPI_SR. When new data is written in QSPI_TDR, this bit is cleared. QSPI_SR.TDRE is used to trigger the Transmit DMA channel.

The end of transfer is indicated by the TXEMPTY flag in the QSPI_SR. If a transfer delay (DLYBCT) is greater than 0 for the last transfer, QSPI_SR.TXEMPTY is set after the completion of this delay. The peripheral clock can be switched off at this time.

The transfer of received data from the internal shift register in QSPI_RDR is indicated by the Receive Data Register Full (RDRF) bit in the QSPI_SR. When the received data is read, QSPI_SR.RDRF bit is cleared.

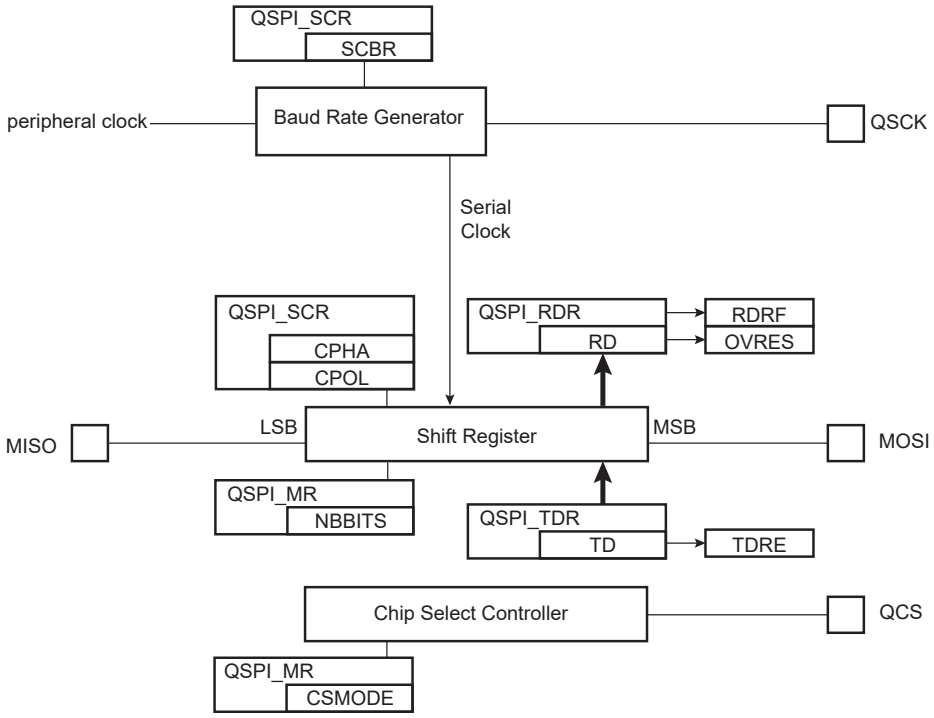
If the QSPI_RDR has not been read before new data is received, the Overrun Error Status (OVRES) bit in QSPI_SR is set. As long as this flag is set, data is loaded in QSPI_RDR. The user must read the QSPI_SR to clear the OVRES bit.

The following figures show, respectively, a block diagram of the SPI when operating in Master mode, and a flow chart describing how transfers are handled.

Downloaded from Arrow.com.

46.6.4.2 SPI Mode Block Diagram

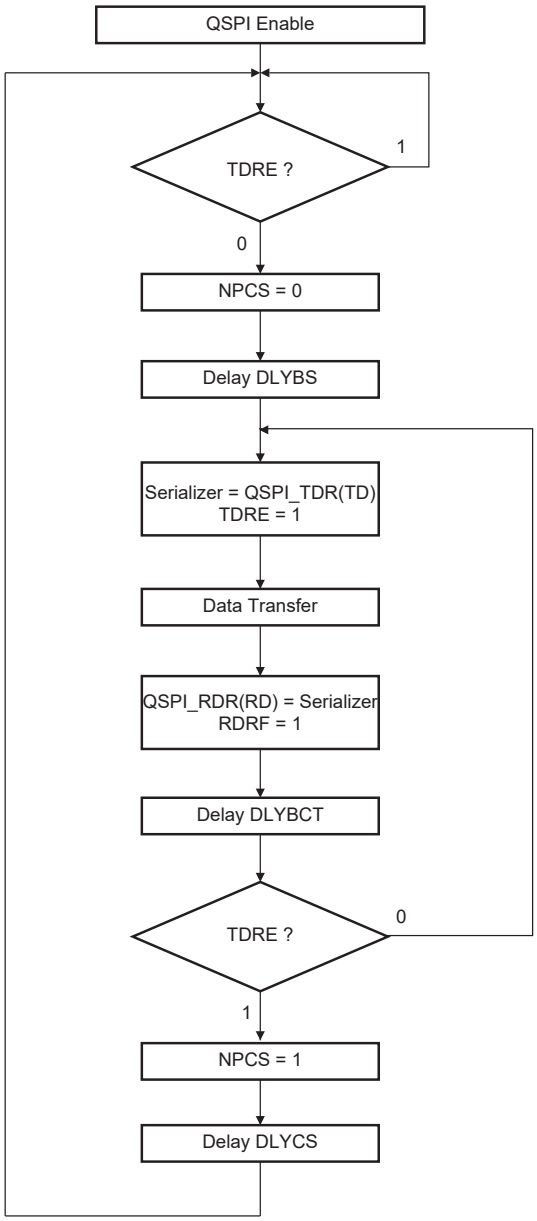
Figure 46-5. SPI Mode Block Diagram



Downloaded from Arrow.com.

46.6.4.3 SPI Mode Flow Diagram

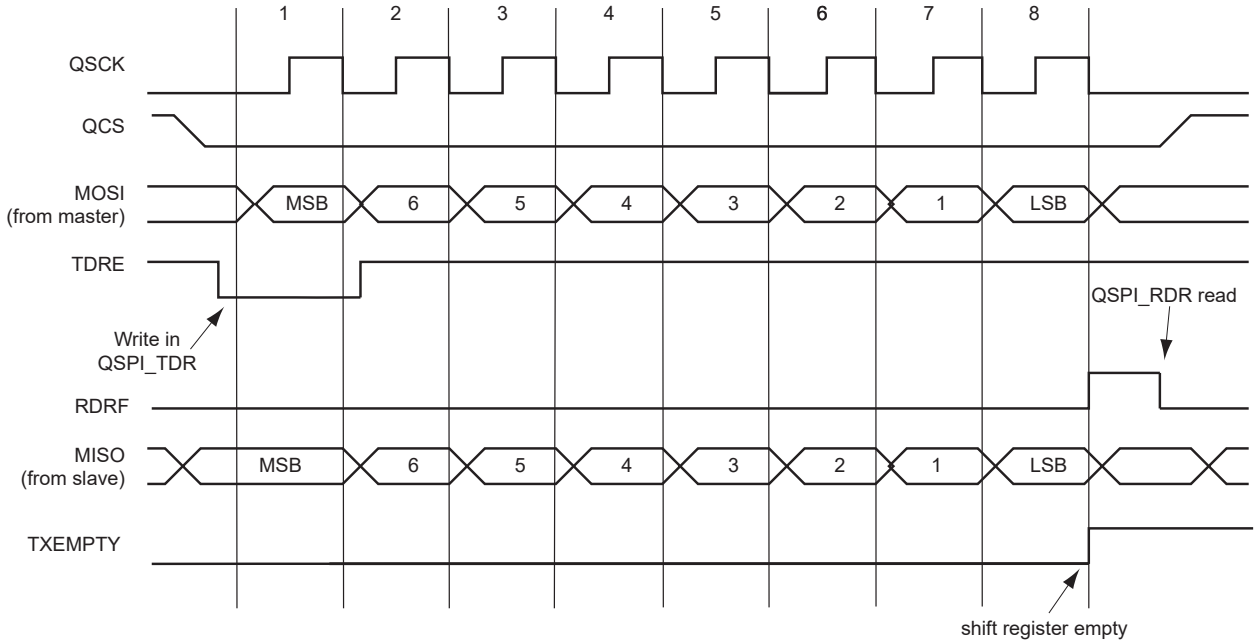
Figure 46-6. SPI Mode Flow Diagram



The figure below shows Transmit Data Register Empty (TDRE), Receive Data Register Full (RDRF) and Transmission Register Empty (TXEMPTY) status flags behavior within the QSPI_SR during an 8-bit data transfer in Fixed mode, without DMA.

Downloaded from Arrow.com.

Figure 46-7. Status Register Flags Behavior



46.6.4.4 Peripheral Deselection without DMA

During a transfer of more than one data on a Chip Select without the DMA, the QSPI_TDR is loaded by the processor and the flag TDRE rises as soon as the content of the QSPI_TDR is transferred into the internal shift register. When this flag is detected high, the QSPI_TDR can be reloaded. If this reload by the processor occurs before the end of the current transfer, the Chip Select is not deasserted between the two transfers. Depending on the application software handling the QSPI_SR flags (by interrupt or polling method) or servicing other interrupts or other tasks, the processor may not reload the QSPI_TDR in time to keep the chip select active (low). A null Delay Between Consecutive Transfer (DLYBCT) value in the QSPI_MR gives even less time for the processor to reload the QSPI_TDR. With some SPI slave peripherals, requiring the chip select line to remain active (low) during a full set of transfers may lead to communication errors.

To facilitate interfacing with such devices, QSPI_MR.CSMODE may be configured to ‘1’. This allows the chip select lines to remain in their current state (low = active) until the end of transfer is indicated by the Last Transfer (LASTXFER) bit in the Control register (QSPI_CR). Even if the QSPI_TDR is not reloaded, the chip select remains active. To have the chip select line rise at the end of the last data transfer, QSPI_CR.LASTXFER must be written to ‘1’ at the same time or after writing the last data to transmit into the QSPI_TDR.

46.6.4.5 Peripheral Deselection with DMA

When the DMA Controller is used, the Chip Select line remains low during the transfer since the TDRE flag is managed by the DMA itself. Reloading the QSPI_TDR by the DMA is done as soon as the TDRE flag is set. In this case, writing QSPI_MR.CSMODE to ‘1’ may not be needed. However, when other DMA channels connected to other peripherals are also in use, the QSPI DMA could be delayed by another DMA with a higher priority on the bus. Having DMA buffers in slower memories like Flash memory or SDRAM compared to fast internal SRAM, may lengthen the reload time of the QSPI_TDR by the DMA as well. This means that the QSPI_TDR might not be reloaded in time to keep the chip select line low. In this case, the chip select line may toggle between data transfer and according to some SPI Slave devices, the communication might get lost. It may be necessary to configure QSPI_MR.CSMODE to ‘1’.

When QSPI_MR.CSMODE is configured to ‘0’, the QCS does not rise in all cases between two transfers on the same peripheral. During a transfer on a Chip Select, the flag TDRE rises as soon as the content of the QSPI_TDR is transferred into the internal shifter. When this flag is detected, the QSPI_TDR can be reloaded. If this reload occurs before the end of the current transfer, the Chip Select is not deasserted between the two transfers. This might lead to difficulties for interfacing with some serial peripherals requiring the chip select to be deasserted after each transfer. To facilitate interfacing with such devices, the QSPI_MR may be configured with QSPI_MR.CSMODE at ‘2’.

Downloaded from Arrow.com.

46.6.5 QSPI Serial Memory Mode

In Serial Memory mode, the QSPI acts as a serial Flash memory controller. The QSPI can be used to read data from the serial Flash memory allowing the CPU to execute code from it (XIP execute in place). The QSPI can also be used to control the serial Flash memory (Program, Erase, Lock, etc.) by sending specific commands. In this mode, the QSPI is compatible with single-bit SPI, Dual SPI and Quad SPI protocols.

To activate this mode, QSPI_MR.SMM must be written to '1'.

In Serial Memory mode, data is transferred either by QSPI_TDR and QSPI_RDR or by writing or read in the QSPI memory space (0x70000000) depending on TFRTYP and SMRM configuration.

46.6.5.1 Instruction Frame

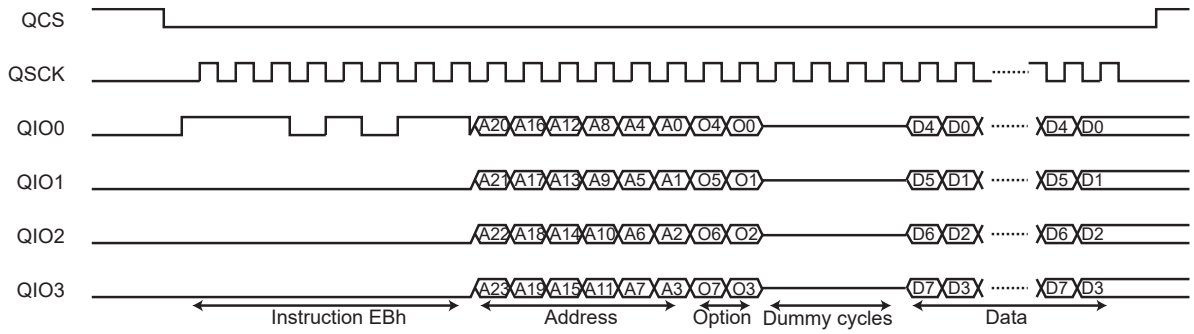
In order to control serial Flash memories, the QSPI is able to send instructions via the SPI bus (ex: READ, PROGRAM, ERASE, LOCK, etc.). Because the instruction set implemented in serial Flash memories is memory vendor-dependent, the QSPI includes a complete Instruction Frame register (QSPI_IFR), which makes it very flexible and compatible with all serial Flash memories.

An instruction frame includes:

- An instruction code (size: 8 bits). The instruction is optional in some cases (see section Continuous Read mode).
- An address (size: 24 bits or 32 bits). The address is optional but is required by instructions such as READ, PROGRAM, ERASE, LOCK. By default the address is 24 bits long, but it can be 32 bits long to support serial Flash memories larger than 128 Mbits (16 Mbytes).
- An option code (size: 1/2/4/8 bits). The option code is not required, but it is useful to activate the XIP mode or the Continuous Read mode (see section Continuous Read mode) for READ instructions, in some serial Flash memory devices. These modes improve the data read latency.
- Dummy cycles. Dummy cycles are optional but required by some READ instructions.
- Data bytes are optional. Data bytes are present for data transfer instructions such as READ or PROGRAM.

The instruction code, the address/option and the data can be sent with Single-bit SPI, Dual SPI or Quad SPI protocols.

Figure 46-8. Instruction Frame



46.6.5.2 Instruction Frame Transmission

To send an instruction frame, the user must first configure the address to send by writing the field ADDR in the Instruction Address register (QSPI_IAR). This step is required if the instruction frame includes an address and no data. When data is present, the address of the instruction is defined by the address of the data accesses in the QSPI memory space, not by QSPI_IAR.

If the instruction frame includes the instruction code and/or the option code, the user must configure the instruction code and/or the option code to send by writing the fields WRINST, WROPT, RDINST and RDOPT in the Write Instruction Code register (QSPI_WICR) and the Read Instruction Code register (QSPI_RICR). QSPI_WICR configures instruction code and option code for write accesses, and QSPI_RICR configures instruction code and option code for read accesses. If a frame is without data (QSPI_IFR.DATAEN = 0), then QSPI_WICR is used for instruction code and option code.

Then, the user must write QSPI_IFR to configure the instruction frame depending on which instruction must be sent. If the instruction frame does not include data, writing in this register triggers the send of the instruction frame in the

Downloaded from Arrow.com.

QSPI. If the instruction frame includes data, the send of the instruction frame is triggered by the first data access in the QSPI memory space.

The instruction frame is configured by the following bits and fields of QSPI_IFR:

- WIDTH field—used to configure which data lanes are used to send the instruction code, the address, the option code and to transfer the data. It is possible to use two unidirectional data lanes (MISO-MOSI Single-bit SPI), two bidirectional data lanes (QIO0-QIO1 Dual SPI) or four bidirectional data lanes (QIO0–QIO3 Quad SPI).
- INSTEN bit—used to enable the send of an instruction code.
- ADDREN bit—used to enable the send of an address after the instruction code.
- OPTEN bit—used to enable the send of an option code after the address.
- DATAEN bit—used to enable the transfer of data (READ or PROGRAM instruction).
- OPTL field—used to configure the option code length. The value written in OPTL must be consistent with the value written in the field WIDTH. For example: OPTL = 0 (1-bit option code) is not consistent with WIDTH = 6 (option code sent with QuadSPI protocol, thus the minimum length of the option code is 4 bits).
- ADDRL bit—used to configure the address length.
- TFRTYP field—used to define which type of data transfer must be performed.
- DDREN bit—used to configure the Double Data Rate mode; instruction code is still transmitted in Single Data Rate mode. Instruction code can be transmitted in DDR mode by writing a ‘1’ to QSPI_IFR.DDRCMDEN.
- NBDUM field—used to configure the number of dummy cycles when reading data from the serial Flash memory. Between the address/option and the data, with some instructions, dummy cycles are inserted by the serial Flash memory.
- APBTFRTYP bit—used to define the APB register transfer to memory type (read or write) when QSPI_IFR.TFRTYP is written to ‘0’.
- DDRCMDEN bit—used to define if the instruction code must be sent in DDR mode when QSPI_IFR.DDREN bit is written to ‘1’.

See 46.7.12 QSPI_IFR.

If data transfer is enabled, the user can access the serial memory by reading or writing the QSPI memory space:

- To read in the serial memory, but not a memory data, for example a JEDEC-ID or the QSPI_SR, QSPI_IFR.TFRTYP must be written to ‘0’.
- To read in the serial memory, and particularly a memory data, TFRTYP must be written to ‘1’.
- To write in the serial memory, but not a memory data, for example writing the configuration or the QSPI_SR, TFRTYP must be written to ‘0’.
- If the user wants to write in the serial memory in particular to program a memory data, TFRTYP must be written to ‘1’.

If QSPI_IFR.TFRTYP has a value other than ‘1’ and QSPI_MR.SMRM = 0, the address sent in the instruction frame is the address of the first system bus accesses. The addresses of the next accesses are not used by the QSPI. At each system bus access, an SPI transfer is performed with the same size. For example, a halfword system bus access leads to a 16-bit SPI transfer, and a byte system bus access leads to an 8-bit SPI transfer.

If SMRM = 1 and TFRTYP= 0, accesses are made via the QSPI registers and the address sent in the instruction frame is the address defined in QSPI_IAR.

QSPI_IFR.APBTFRTYP is used to define whether the access is a read access or a write access. Each time QSPI_IFR is written (in case of read access), or each time QSPI_TDR is written (in case of write transfer), an SPI transfer is performed with a byte size or halfword size if QSPI_IFR.WIDTH field is configured to QSPI_IFR.DDREN=1. Another byte or halfword is read each time QSPI_RDR is read (flag RDRF shows when a data can be read in QSPI_RDR) or written each time QSPI_TDR is written (flag TDRE shows when a new data can be written). The SPI transfer ends by writing QSPI_CR.LASTXFER.

If TFRTYP = 1, the address of the first instruction frame is the one of the first read access in the QSPI memory space. Each time the read accesses become nonsequential (addresses are not consecutive), a new instruction frame is sent with the last system bus access address. In this way, the system can read data at a random location in the serial memory. The size of the SPI transfers may differ from the size of the system bus read accesses.

Downloaded from Arrow.com.

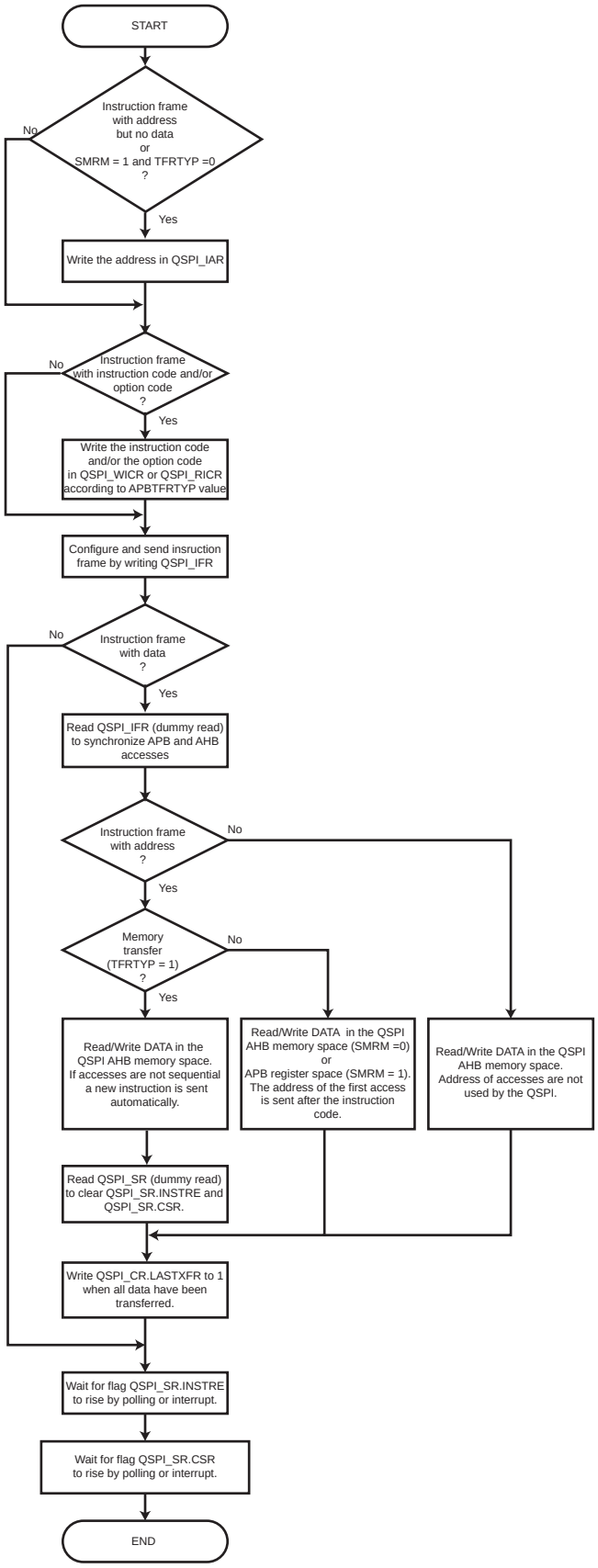
When data transfer is not enabled, the end of the instruction frame is indicated when QSPI_SR.INSTRE rises. (The QSPI_SR.CSR flag indicates when chip select rises. A delay between these flags may exist in case of high clock division or a high DLYBCT value).

When data transfer is enabled, the user must indicate when the data transfer is completed in the QSPI memory space by setting QSPI_CR.LASTXFR. The end of the instruction frame is indicated when QSPI_SR.INSTRE rises.

The following figure illustrates instruction transmission management.

Downloaded from Arrow.com.

Figure 46-9. Instruction Transmission Flow Diagram



Downloaded from Arrow.com.

46.6.5.3 Read Memory Transfer

The user can access the data of the serial memory by sending an instruction with QSPI_IFR.DATAEN = 1 and QSPI_IFR.TFRTYP = 1.

In this mode, the QSPI is able to read data at random address into the serial Flash memory, allowing the CPU to execute code directly from it (XIP execute-in-place).

In order to fetch data, the user must first configure the instruction frame by writing the QSPI_IFR. Then data can be read at any address in the QSPI address space mapping. The address of the system bus read accesses match the address of the data inside the serial Flash memory.

When Fetch mode is enabled, several instruction frames can be sent before writing QSPI_CR.LASTXFR. Each time the system bus read accesses become nonsequential (addresses are not consecutive), a new instruction frame is sent with the corresponding address.

46.6.5.4 Continuous Read Mode

The QSPI is compatible with the Continuous Read mode which is implemented in some serial Flash memories.

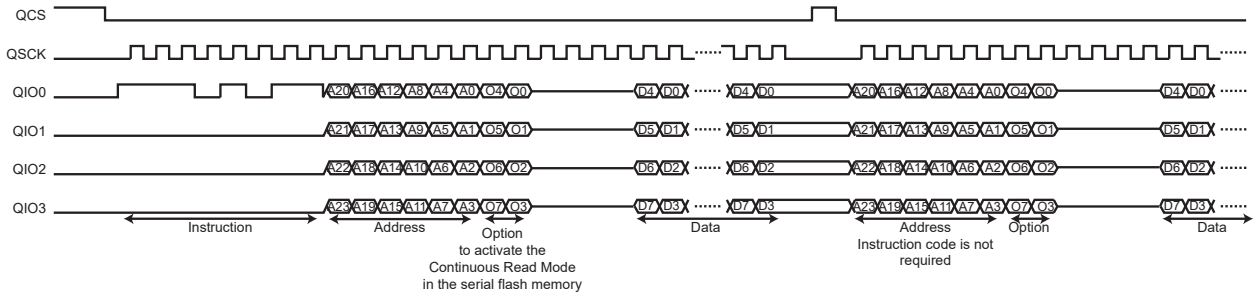
In Continuous Read mode, the instruction overhead is reduced by excluding the instruction code from the instruction frame. When the Continuous Read mode is activated in a serial Flash memory by a specific option code, the instruction code is stored in the memory. For the next instruction frames, the instruction code is not required as the memory uses the stored one.

In the QSPI, Continuous Read mode is used when reading data from the memory (QSPI_IFR.TFRTYP = 1). The addresses of the system bus read accesses are often nonsequential and this leads to many instruction frames that have the same instruction code. By disabling the send of the instruction code, the Continuous Read mode reduces the access time of the data.

To be functional, this mode must be enabled in both the QSPI and the serial Flash memory. The Continuous Read mode is enabled in the QSPI by writing CRM to ‘1’ in the QSPI_IFR (TFRTYP must equal 1). The Continuous Read mode is enabled in the serial Flash memory by sending a specific option code.

⚠ CAUTION If the Continuous Read mode is not supported by the serial Flash memory or disabled, CRM bit must not be written to ‘1’, otherwise data read out of the serial Flash memory is unpredictable.

Figure 46-10. Continuous Read Mode



46.6.5.5 Instruction Frame Transmission Examples

All waveforms in the following examples describe SPI transfers in SPI Clock mode 0 (QSPI_SCR.CPOL = 0 and QSPI_SCR.CPHA = 0; see section Serial Clock Phase and Polarity).

All system bus accesses described below refer to the system bus address phase. System bus wait cycles and system bus data phases are not shown.

Example 1:

Instruction in Single-bit SPI, without address, without option, without data.

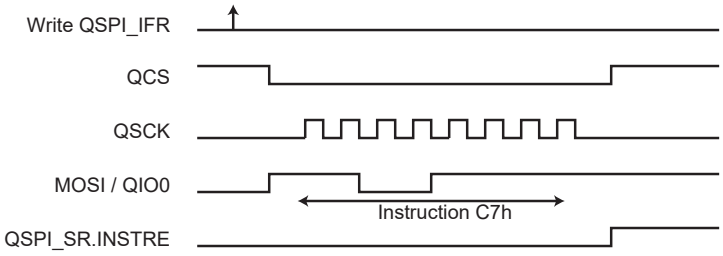
Command: CHIP ERASE (C7h).

- Write 0x0000_00C7 in QSPI_WICR.
- Write 0x0000_0010 in QSPI_IFR.

Downloaded from Arrow.com.

- Wait for QSPI_SR.INSTRE to rise.

Figure 46-11. Instruction Transmission Waveform 1



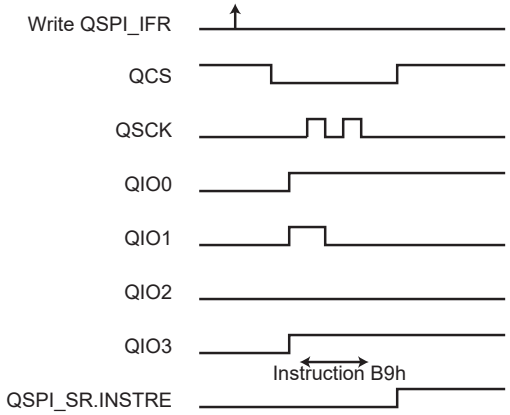
Example 2:

Instruction in Quad SPI, without address, without option, without data.

Command: POWER DOWN (B9h)

- Write 0x0000_00B9 in QSPI_WICR.
- Write 0x0000_0016 in QSPI_IFR.
- Wait for QSPI_SR.INSTRE to rise.

Figure 46-12. Instruction Transmission Waveform 2



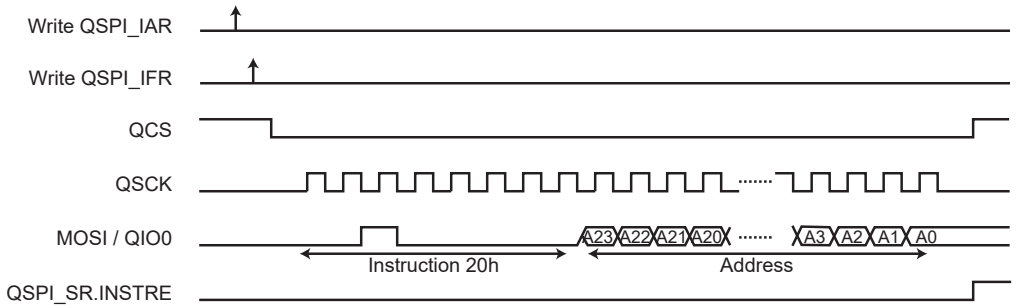
Example 3:

Instruction in Single-bit SPI, with address in Single-bit SPI, without option, without data.

Command: BLOCK ERASE (20h)

- Write the address (of the block to erase) in QSPI_AR.
- Write 0x0000_0020 in QSPI_WICR.
- Write 0x0000_0030 in QSPI_IFR.
- Wait for QSPI_SR.INSTRE to rise.

Figure 46-13. Instruction Transmission Waveform 3



Example 4:

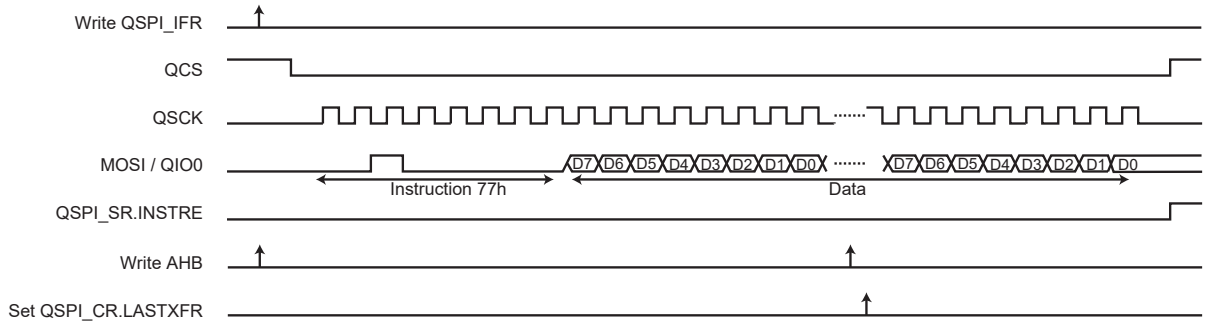
Downloaded from Arrow.com.

Instruction in Single-bit SPI, without address, without option, with data write in Single-bit SPI.

Command: SET BURST (77h)

- Write 0x0000_0077 in QSPI_WICR.
- Write 0x0000_0090 in QSPI_IFR.
- Read QSPI_IFR (dummy read) to synchronize system bus accesses.
- Write data in the system bus memory space (0x70000000).
  The address of system bus write accesses is not used.
- Write a '1' to QSPI_CR.LASTXFR.
- Wait for QSPI_SR.INSTRE to rise.

Figure 46-14. Instruction Transmission Waveform 4



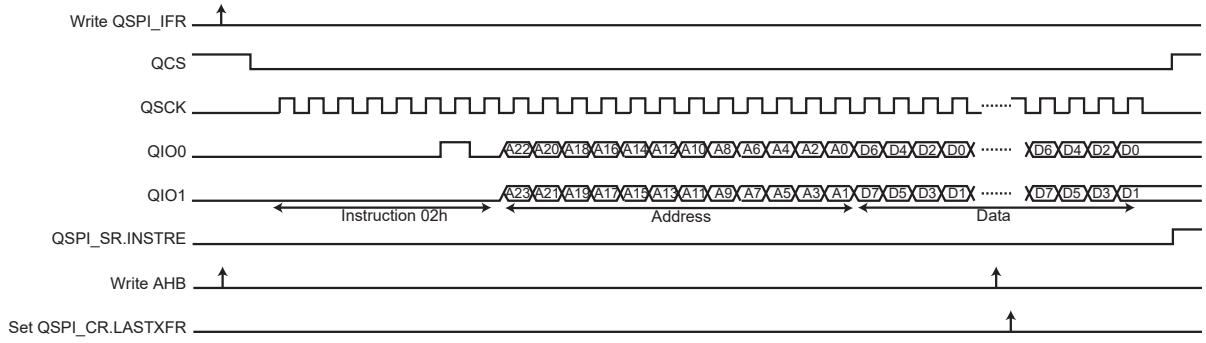
Example 5:

Instruction in Single-bit SPI, with address in Dual SPI, without option, with data write in Dual SPI.

Command: BYTE/PAGE PROGRAM (02h)

- Write 0x0000_0002 in QSPI_WICR.
- Write 0x0000_10B3 in QSPI_IFR.
- Read QSPI_IFR (dummy read) to synchronize system bus accesses.
- Write data in the QSPI system bus memory space (0x70000000).
  The address of the first system bus write access is sent in the instruction frame.
  The address of the next system bus write accesses is not used.
- Write a '1' to QSPI_CR.LASTXFR.
- Wait for QSPI_SR.INSTRE to rise.

Figure 46-15. Instruction Transmission Waveform 5



Example 6:

Instruction in Single-bit SPI, with address in Single-bit SPI, without option, with data read in Quad SPI, with eight dummy cycles.

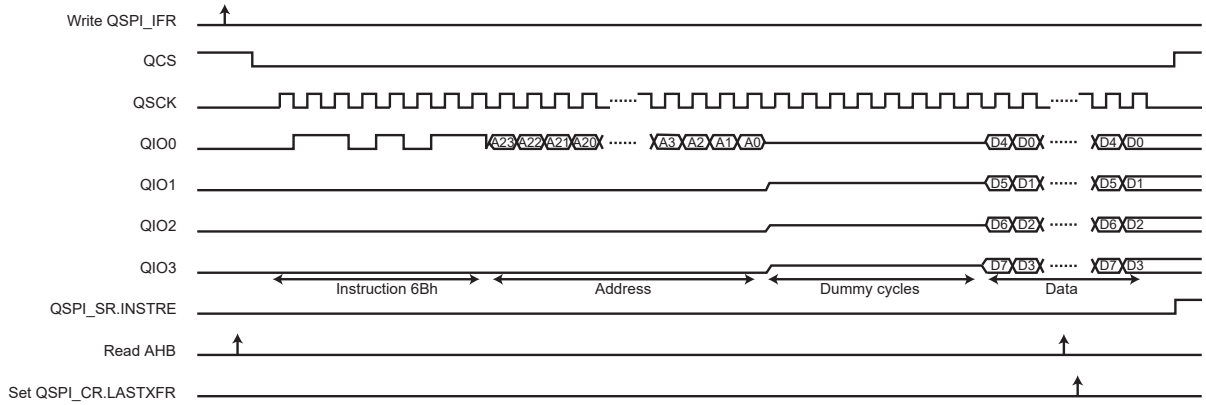
Command: QUAD_OUTPUT READ ARRAY (6Bh)

- Write 0x0000_006B in QSPI_RICR.
- Write 0x0008_10B2 in QSPI_IFR.

Downloaded from Arrow.com.

- Read QSPI_IR (dummy read) to synchronize system bus accesses.
- Read data in the QSPI system bus memory space (0x70000000).
  The address of the first system bus read access is sent in the instruction frame.
  The address of the next system bus read accesses is not used.
- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_SR.INSTRE to rise.

Figure 46-16. Instruction Transmission Waveform 6



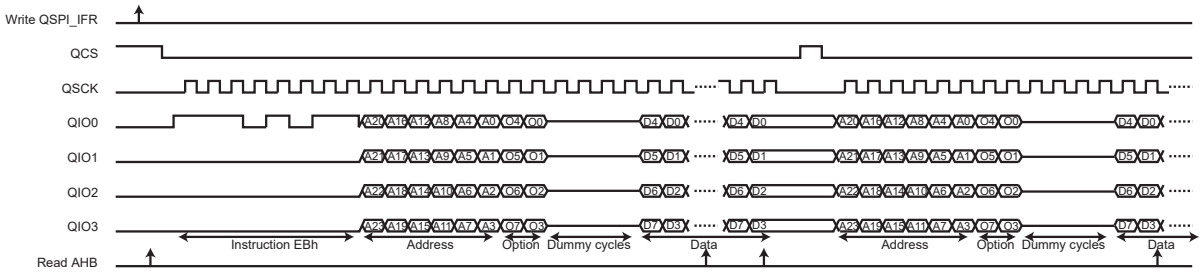
Example 7:

Instruction in Single-bit SPI, with address and option in Quad SPI, with data read in Quad SPI, with four dummy cycles, with fetch and continuous read.

Command: FAST READ QUAD I/O (EBh) - 8-BIT OPTION (0x30h)

- Write 0x0030_00EB in QSPI_RICR.
- Write 0x0004_33F4 in QSPI_IFR.
- Read QSPI_IFR (dummy read) to synchronize system bus accesses.
- Read data in the QSPI system bus memory space (0x70000000).
  Fetch is enabled, the address of the system bus read accesses is always used.
- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_SR.INSTRE to rise.

Figure 46-17. Instruction Transmission Waveform 7



Example 8:

Instruction in Quad SPI, with address in Quad SPI, without option, with data read in Quad SPI, with two dummy cycles, with fetch.

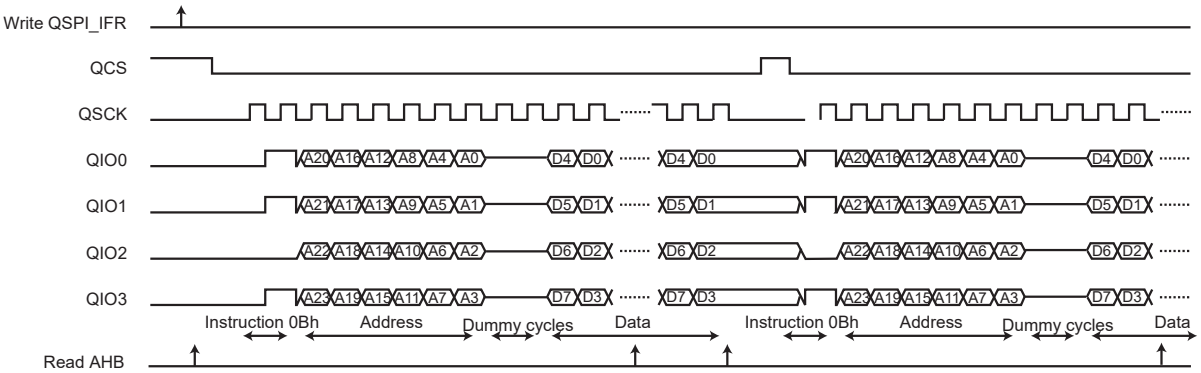
Command: HIGH-SPEED READ (0Bh)

- Write 0x0000_000B in QSPI_RICR.
- Write 0x0002_20B6 in QSPI_IFR.
- Read QSPI_IFR (dummy read) to synchronize system bus accesses.
- Read data in the QSPI system bus memory space (0x70000000).
  Fetch is enabled, the address of the system bus read accesses is always used.

Downloaded from Arrow.com.

- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_SR.INSTRE to rise.

Figure 46-18. Instruction Transmission Waveform 8



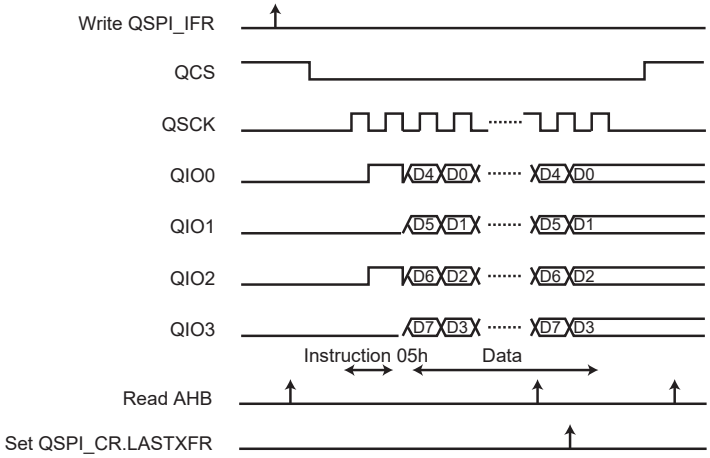
Example 9:

Instruction in Quad SPI, without address, without option, with data read in Quad SPI, without dummy cycles, without fetch.

Command: HIGH-SPEED READ (05h)

- Write 0x0000_0005 in QSPI_RICR.
- Write 0x0000_0096 in QSPI_IFR.
- Read QSPI_IFR (dummy read) to synchronize system bus accesses.
- Read data in the QSPI system bus memory space (0x70000000).
  Fetch is disabled.
- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_SR.INSTRE to rise.

Figure 46-19. Instruction Transmission Waveform 9



Example 10:

Instruction in Quad SPI, without address, without option, with data read in Quad SPI, without dummy cycles, without fetch, read launched through APB interface.

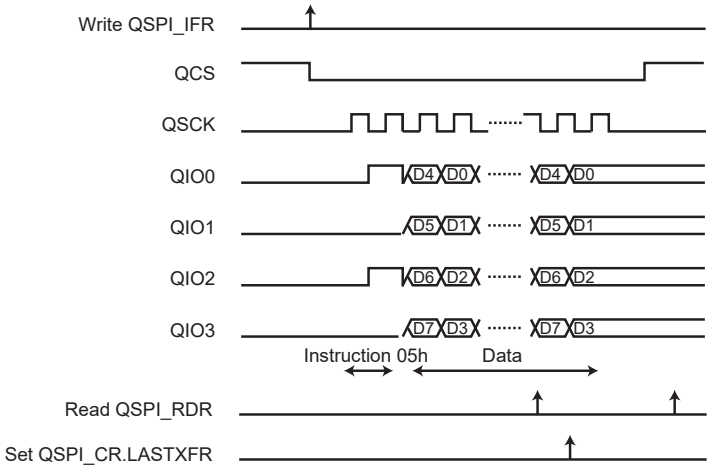
Command: HIGH-SPEED READ (05h)

- Set SMRM to ‘1’ in QSPI_MR
- Write 0x0000_0005 in QSPI_RICR.
- Write 0x0100_0096 in QSPI_IFR (will start the transfer).

Downloaded from Arrow.com.

- Wait flag RDRF and Read data in the QSPI_RDR register
  Fetch is disabled.
- Write a ‘1’ to QSPI_CR.LASTXFR.
- Wait for QSPI_SR.INSTRE to rise.

Figure 46-20. Instruction Transmission Waveform 10



46.6.6 Scrambling/Unscrambling Function

The scrambling/unscrambling function cannot be performed on devices other than memories. Data is scrambled when written to memory and unscrambled when data is read.

The external data lines can be scrambled in order to prevent intellectual property data located in off-chip memories from being easily recovered by analyzing data at the package pin level of either the microcontroller or the QSPI slave device (e.g., memory).

The scrambling/unscrambling function can be enabled by writing a ‘1’ to the SCREN bit in the QSPI Scrambling Mode Register (QSPI_SMR).

The scrambling and unscrambling are performed on-the-fly without impacting the throughput.

The scrambling method depends on the user-configurable user scrambling key (field USRK) in the QSPI Scrambling Key Register (QSPI_SKR). QSPI_SKR is only accessible in Write mode.

When QSPI_SMR.SCRKL has been written once to ‘1’, QSPI_SKR.USRK cannot be written again until the next reset.

If QSPI_SMR.RVDIS is written to ‘0’, the scrambling/unscrambling algorithm includes the user scrambling key plus a random value depending on device processing characteristics. Data scrambled by a given microcontroller cannot be unscrambled by another.

If QSPI_SMR.RVDIS is written to ‘1’, the scrambling/unscrambling algorithm includes only the user scrambling key. No random value is part of the key.

The user scrambling key or the seed for key generation must be securely stored in a reliable nonvolatile memory in order to recover data from the off-chip memory. Any data scrambled with a given key cannot be recovered if the key is lost.

46.6.6.1 Clearing Scrambling Keys on a Tamper Event

On a tamper detection event on WKUP pins, an immediate clear of the scrambling key (QSPI_SKR) is performed if QSPI_MR.TAMPCLR is set.

46.6.7 Register Write Protection

To prevent any single software error from corrupting QSPI behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the QSPI Write Protection Mode Register (QSPI_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the QSPI Write Protection Status Register (QSPI_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

Downloaded from Arrow.com.

The WPVS bit is automatically cleared after reading the QSPI_WPSR.

The following registers can be write-protected when WPEN is set in QSPI_WPMR:

- QSPI Mode Register
- QSPI Serial Clock Register
- QSPI Scrambling Mode Register
- QSPI Scrambling Key Register

The following registers can be write-protected when WPCREN is set in QSPI_WPMR:

- QSPI Control Register

The following registers can be write-protected when WPITEN is set in QSPI_WPMR:

- QSPI Interrupt Enable Register
- QSPI Interrupt Disable Register

Downloaded from Arrow.com.

46.7 Register Summary

Offset	Name	Bit Pos.								
0x00	QSPI_CR	31:24								LASTXFER
		23:16								
		15:8								
		7:0	SWRST						QSPIDIS	QSPIEN
0x04	QSPI_MR	31:24	DLYCS[7:0]							
		23:16	DLYBCT[7:0]							
		15:8			QICMEN		NBBITS[3:0]			
		7:0	TAMPCLR		CSMODE[1:0]		SMRM	WDRBT		SMM
0x08	QSPI_RDR	31:24								
		23:16								
		15:8	RD[15:8]							
		7:0	RD[7:0]							
0x0C	QSPI_TDR	31:24								
		23:16								
		15:8	TD[15:8]							
		7:0	TD[7:0]							
0x10	QSPI_SR	31:24								QSPIENS
		23:16								
		15:8						INSTRE	CSS	CSR
		7:0					OVRES	TXEMPTY	TDRE	RDRF
0x14	QSPI_IER	31:24								
		23:16								
		15:8						INSTRE	CSS	CSR
		7:0					OVRES	TXEMPTY	TDRE	RDRF
0x18	QSPI_IDR	31:24								
		23:16								
		15:8						INSTRE	CSS	CSR
		7:0					OVRES	TXEMPTY	TDRE	RDRF
0x1C	QSPI_IMR	31:24								
		23:16								
		15:8						INSTRE	CSS	CSR
		7:0					OVRES	TXEMPTY	TDRE	RDRF
0x20	QSPI_SCR	31:24								
		23:16	DLYBS[7:0]							
		15:8	SCBR[7:0]							
		7:0							CPHA	CPOL
0x24 ... 0x2F	Reserved									
0x30	QSPI_IAR	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x34	QSPI_WICR	31:24								
		23:16	WROPT[7:0]							
		15:8								
		7:0	WRINST[7:0]							
0x38	QSPI_IFR	31:24						DDRCMDEN		APBTFRTYP
		23:16				NBDUM[4:0]				
		15:8	DDREN	CRM		TFRTYP		ADDRL	OPTL[1:0]	
		7:0	DATAEN	OPTEN	ADDREN	INSTEN		WIDTH[2:0]		
0x3C	QSPI_RICR	31:24								
		23:16	RDOPT[7:0]							
		15:8								
		7:0	RDINST[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x40	QSPI_SMR	31:24								
		23:16								
		15:8								
		7:0						SCRKL	RVDIS	SCREN
0x44	QSPI_SKR	31:24	USRK[31:24]							
		23:16	USRK[23:16]							
		15:8	USRK[15:8]							
		7:0	USRK[7:0]							
0x48 ... 0xE3	Reserved									
0xE4	QSPI_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPEN
0xE8	QSPI_WPSR	31:24								
		23:16								
		15:8	WPVSRC[7:0]							
		7:0								WPVS

Downloaded from Arrow.com.

46.7.1 QSPI Control Register

Name: QSPI_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the QSPI Write Protection Mode Register .

Bit	31	30	29	28	27	26	25	24
								LASTXFER
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	SWRST						QSPIDIS	QSPIEN
Access	W						W	W
Reset	–						–	–

Bit 24 – LASTXFER Last Transfer

Value	Description
0	No effect.
1	The chip select is deasserted after the character written in QSPI_TDR.TD has been transferred.

Bit 7 – SWRST QSPI Software Reset
DMA channels are not affected by software reset.

Value	Description
0	No effect.
1	Reset the QSPI. A software-triggered hardware reset of the QSPI interface is performed.

Bit 1 – QSPIDIS QSPI Disable
As soon as QSPIDIS is set, the QSPI finishes its transfer.
All pins are set in Input mode and no data is received or transmitted.
If a transfer is in progress, the transfer is finished before the QSPI is disabled.
If both QSPIEN and QSPIDIS are equal to one when QSPI_CR is written, the QSPI is disabled.

Value	Description
0	No effect.
1	Disables the QSPI.

Bit 0 – QSPIEN QSPI Enable

Value	Description
0	No effect.
1	Enables the QSPI to transfer and receive data.

Downloaded from Arrow.com.

46.7.2 QSPI Mode Register

Name: QSPI_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	DLYCS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DLYBCT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			QICMEN		NBBITS[3:0]			
Access			R/W		R/W	R/W	R/W	R/W
Reset			0		0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TAMPCLR		CSMODE[1:0]		SMRM	WDRBT		SMM
Access	R/W		R/W	R/W	R/W	R/W		R/W
Reset	0		0	0	0	0		0

Bits 31:24 – DLYCS[7:0] Minimum Inactive QCS Delay
This field defines the minimum delay between the deactivation and the activation of QCS. The DLYCS time guarantees the slave minimum deselect time.
If DLYCS written to ‘0’, one peripheral clock period is inserted by default.
Otherwise, the following equation determines the delay:
DLYCS = Minimum inactive × $f_{peripheral\ clock}$

Bits 23:16 – DLYBCT[7:0] Delay Between Consecutive Transfers
This field defines the delay between two consecutive transfers with the same peripheral without removing the chip select. The delay is always inserted after each transfer and before removing the chip select if needed.
When DLYBCT is written to ‘0’, no delay between consecutive transfers is inserted and the clock keeps its duty cycle over the character transfers. In Serial Memory mode (SMM = 1), DLYBCT must be written to ‘0’ and no delay is inserted.
Otherwise, the following equation determines the delay:
DLYBCT = (Delay Between Consecutive Transfers × $f_{peripheral\ clock}$) / 32

Bit 13 – QICMEN QSPI Inter-chip Mode Enable
0 (DISABLED): QSPI_WICR.WROPT and QSPI_RICR.RDOPT define the transfer options and QSPI_IFR.NBDUM defines the number of dummy cycles.
1 (ENABLED): No dummy cycles are inserted for write accesses, whatever the value configured in QSPI_IFR.NBDUM. The option field of the frame is automatically generated by QSPI depending on the type system bus transfer (the fields QSPI_RICR.RDOPT and QSPI_WICR.WROPT have no effect). Refer to section “Slave Quad SPI Interface (SQSPI)” for more details.

Bits 11:8 – NBBITS[3:0] Number Of Bits Per Transfer
NBBITS is used only when SMM is set to '0'.

Value	Name	Description
0	8_BIT	8 bits for transfer

Downloaded from Arrow.com.

Value	Name	Description
8	16_BIT	16 bits for transfer

Bit 7 – TAMPCLR Tamper Clear Enable

Value	Description
0	A tamper detection event has no effect on QSPI scrambling keys.
1	A tamper detection event immediately clears QSPI scrambling keys.

Bits 5:4 – CSMODE[1:0] Chip Select Mode
The CSMODE field determines how the chip select is deasserted
Note: This field is forced to LASTXFER when SMM is written to '1'.

Value	Name	Description
0	NOT_RELOADED	The chip select is deasserted if QSPI_TDR.TD has not been reloaded before the end of the current transfer.
1	LASTXFER	The chip select is deasserted when the bit LASTXFER is written to '1' and the character written in QSPI_TDR.TD has been transferred.
2	SYSTEMATICALLY	The chip select is deasserted systematically after each transfer.

Bit 3 – SMRM Serial Memory Register Mode

Value	Description
0	Serial Memory registers are written via AHB access. See section Instruction Frame Transmission for details.
1	Serial Memory registers are written via APB access. See section Instruction Frame Transmission for details.

Bit 2 – WDRBT Wait Data Read Before Transfer
0 (DISABLED): No effect. In SPI mode, a transfer can be initiated whatever the state of the QSPI_RDR is.
1 (ENABLED): In SPI mode, a transfer can start only if the QSPI_RDR is empty, i.e., does not contain any unread data. This mode prevents overrun error in reception.

Bit 0 – SMM Serial Memory Mode
0 (SPI): The QSPI is in SPI mode.
1 (MEMORY): The QSPI is in Serial Memory mode.

Downloaded from Arrow.com.

46.7.3 QSPI Receive Data Register

Name: QSPI_RDR
Offset: 0x08
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RD[15:0] Receive Data
Data received by the QSPI is stored in this register right-justified. Unused bits read zero.

Downloaded from Arrow.com.

46.7.4 QSPI Transmit Data Register

Name: QSPI_TDR
Offset: 0x0C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	–

Bits 15:0 – TD[15:0] Transmit Data
Data to be transmitted by the QSPI is stored in this register. Information to be transmitted must be written to the Transmit Data register in a right-justified format.

Downloaded from Arrow.com.

46.7.5 QSPI Status Register

Name: QSPI_SR
Offset: 0x10
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
								QSPIENS
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						INSTRE	CSS	CSR
Access						R	R	R
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
					OVRES	TXEMPTY	TDRE	RDRF
Access					R	R	R	R
Reset					0	0	0	0

Bit 24 – QSPIENS QSPI Enable Status

Value	Description
0	QSPI is disabled.
1	QSPI is enabled.

Bit 10 – INSTRE Instruction End Status (cleared on read)

Value	Description
0	No instruction end has been detected since the last read of QSPI_SR.
1	At least one instruction end has been detected since the last read of QSPI_SR.

Bit 9 – CSS Chip Select Status

Value	Description
0	The chip select is asserted.
1	The chip select is not asserted.

Bit 8 – CSR Chip Select Rise (cleared on read)

Value	Description
0	No chip select rise has been detected since the last read of QSPI_SR.
1	At least one chip select rise has been detected since the last read of QSPI_SR.

Bit 3 – OVRES Overrun Error Status (cleared on read)
An overrun occurs when QSPI_RDR is loaded at least twice from the serializer since the last read of the QSPI_RDR.

Value	Description
0	No overrun has been detected since the last read of QSPI_SR.
1	At least one overrun error has occurred since the last read of QSPI_SR.

Bit 2 – TXEMPTY Transmission Registers Empty (cleared by writing QSPI_TDR)

Value	Description
0	As soon as data is written in QSPI_TDR.

Downloaded from Arrow.com.

Value	Description
1	QSPI_TDR and the internal shifter are empty. If a transfer delay has been defined, TXEMPTY is set after the completion of such delay.

Bit 1 – TDRE Transmit Data Register Empty (cleared by writing QSPI_TDR)
TDRE equals zero when the QSPI is disabled or at reset. The QSPI enable command sets this bit to one.

Value	Description
0	Data has been written to QSPI_TDR and not yet transferred to the serializer.
1	The last data written in the QSPI_TDR has been transferred to the serializer.

Bit 0 – RDRF Receive Data Register Full (cleared by reading QSPI_RDR)

Value	Description
0	No data has been received since the last read of QSPI_RDR.
1	Data has been received and the received data has been transferred from the serializer to QSPI_RDR since the last read of QSPI_RDR.

Downloaded from Arrow.com.

46.7.6 QSPI Interrupt Enable Register

Name: QSPI_IER
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the QSPI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						INSTRE	CSS	CSR
Access						W	W	W
Reset						–	–	–

Bit	7	6	5	4	3	2	1	0
					OVRES	TXEMPTY	TDRE	RDRF
Access					W	W	W	W
Reset					–	–	–	–

Bit 10 – INSTRE Instruction End Interrupt Enable

Bit 9 – CSS Chip Select Status Interrupt Enable

Bit 8 – CSR Chip Select Rise Interrupt Enable

Bit 3 – OVRES Overrun Error Interrupt Enable

Bit 2 – TXEMPTY Transmission Registers Empty Enable

Bit 1 – TDRE Transmit Data Register Empty Interrupt Enable

Bit 0 – RDRF Receive Data Register Full Interrupt Enable

Downloaded from Arrow.com.

46.7.7 QSPI Interrupt Disable Register

Name: QSPI_IDR
Offset: 0x18
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the QSPI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						INSTRE	CSS	CSR
Access						W	W	W
Reset						–	–	–

Bit	7	6	5	4	3	2	1	0
					OVRES	TXEMPTY	TDRE	RDRF
Access					W	W	W	W
Reset					–	–	–	–

Bit 10 – INSTRE Instruction End Interrupt Disable

Bit 9 – CSS Chip Select Status Interrupt Disable

Bit 8 – CSR Chip Select Rise Interrupt Disable

Bit 3 – OVRES Overrun Error Interrupt Disable

Bit 2 – TXEMPTY Transmission Registers Empty Disable

Bit 1 – TDRE Transmit Data Register Empty Interrupt Disable

Bit 0 – RDRF Receive Data Register Full Interrupt Disable

Downloaded from Arrow.com.

46.7.8 QSPI Interrupt Mask Register

Name: QSPI_IMR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						INSTRE	CSS	CSR
Access						R	R	R
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
					OVRES	TXEMPTY	TDRE	RDRF
Access					R	R	R	R
Reset					0	0	0	0

Bit 10 – INSTRE Instruction End Interrupt Mask

Bit 9 – CSS Chip Select Status Interrupt Mask

Bit 8 – CSR Chip Select Rise Interrupt Mask

Bit 3 – OVRES Overrun Error Interrupt Mask

Bit 2 – TXEMPTY Transmission Registers Empty Mask

Bit 1 – TDRE Transmit Data Register Empty Interrupt Mask

Bit 0 – RDRF Receive Data Register Full Interrupt Mask

Downloaded from Arrow.com.

46.7.9 QSPI Serial Clock Register

Name: QSPI_SCR
Offset: 0x20
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DLYBS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SCBR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
							CPHA	CPOL
Access							R/W	R/W
Reset							0	0

Bits 23:16 – DLYBS[7:0] Delay Before QSCK
This field defines the delay from QCS valid to the first valid QSCK transition.
When DLYBS equals zero, the QCS valid to QSCK transition is 1/2 the QSCK clock period.
Otherwise, the following equation determines the delay:
DLYBS = Delay Before QSCK × $f_{peripheral\ clock}$

Bits 15:8 – SCBR[7:0] Serial Clock Baud Rate
The QSPI uses a modulus counter to derive the QSCK baud rate from the peripheral clock. The baud rate is selected by writing a value from 0 to 255 in the SCBR field. The following equation determines the QSCK baud rate:
SCBR = ($f_{peripheral\ clock}$ / QSCK Baudrate) - 1

Bit 1 – CPHA Clock Phase
CPHA determines which edge of QSCK causes data to change and which edge causes data to be captured. CPHA is used with CPOL to produce the required clock/data relationship between master and slave devices.

Value	Description
0	Data is captured on the leading edge of QSCK and changed on the following edge of QSCK.
1	Data is changed on the leading edge of QSCK and captured on the following edge of QSCK.

Bit 0 – CPOL Clock Polarity
CPOL is used to determine the inactive state value of the serial clock (QSCK). It is used with CPHA to produce the required clock/data relationship between master and slave devices.

Value	Description
0	The inactive state value of QSCK is logic level zero.
1	The inactive state value of QSCK is logic level one.

Downloaded from Arrow.com.

46.7.10 QSPI Instruction Address Register

Name: QSPI_IAR
Offset: 0x30
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Address
Address to send to the serial Flash memory in the instruction frame.

Downloaded from Arrow.com.

46.7.11 QSPI Write Instruction Code Register

Name: QSPI_WICR
Offset: 0x34
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WROPT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	WRINST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – WROPT[7:0] Write Option Code
Option code to send to the serial Flash memory in case of write transfer.

Bits 7:0 – WRINST[7:0] Write Instruction Code
Instruction code to send to the serial Flash memory in case of write transfer.

Downloaded from Arrow.com.

46.7.12 QSPI Instruction Frame Register

Name: QSPI_IFR
Offset: 0x38
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						DDRCMDEN		APBTFRTYP
Access						R/W		R/W
Reset						0		0

Bit	23	22	21	20	19	18	17	16
				NBDUM[4:0]				
Access								
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DDREN	CRM		TFRTYP		ADDRL	OPTL[1:0]	
Access	R/W	R/W		R/W		R/W	R/W	R/W
Reset	0	0		0		0	0	0

Bit	7	6	5	4	3	2	1	0
	DATAEN	OPTEN	ADDREN	INSTEN		WIDTH[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit 26 – DDRCMDEN DDR Mode Command Enable
0 (DISABLED): Transfer of instruction field is performed in Single Data Rate mode even if DDREN bit is written to ‘1’.
1 (ENABLED): Transfer of instruction field is performed in Double Data Rate mode if DDREN bit is written to ‘1’. If DDREN bit is written to ‘0’, the instruction field is sent in Single Data Rate mode.

Bit 24 – APBTFRTYP APB Transfer Type

Value	Description
0	APB register transfer to the memory is a write transfer. Useful when TRFTYP is written to ‘0’ and SMRM to ‘1’.
1	APB register transfer to the memory is a read transfer. Useful when TRFTYP is written to ‘0’ and SMRM to ‘1’.

Bits 20:16 – NBDUM[4:0] Number Of Dummy Cycles
The NBDUM field defines the number of dummy cycles required by the serial Flash memory before data transfer.

Bit 15 – DDREN DDR Mode Enable
0 (DISABLED): Transfers are performed in Single Data Rate mode.
1 (ENABLED): Transfers are performed in Double Data Rate mode, whereas the instruction field is still transferred in Single Data Rate mode.
Note: The DDRCMDEN bit defines how the instruction field is sent when Double Data Rate mode is enabled. If DDRCMDEN bit is at ‘0’, the instruction field is sent in Single Data Rate mode.

Bit 14 – CRM Continuous Read Mode
0 (DISABLED): Continuous Read mode is disabled.
1 (ENABLED): Continuous Read mode is enabled.

Bit 12 – TFRTYP Data Transfer Type

Downloaded from Arrow.com.

Value	Name	Description
0	TRSFR_REGISTER	Read/Write transfer from the serial memory. Scrambling is not performed. Read at random location (fetch) in the serial Flash memory is not possible.
1	TRSFR_MEMORY	Read/Write data transfer from the serial memory. If enabled, scrambling is performed. Read at random location (fetch) in the serial Flash memory is possible.

Bit 10 – ADDRL Address Length
The ADDRL bit determines the length of the address.
0 (24_BIT): The address is 24 bits long.
1 (32_BIT): The address is 32 bits long.

Bits 9:8 – OPTL[1:0] Option Code Length
The OPTL field determines the length of the option code. The value written in OPTL must be consistent with the value written in the field WIDTH. For example, OPTL = 0 (1-bit option code) is not consistent with WIDTH = 6 (option code sent with QuadSPI protocol, thus the minimum length of the option code is 4 bits).

Value	Name	Description
0	OPTION_1BIT	The option code is 1 bit long.
1	OPTION_2BIT	The option code is 2 bits long.
2	OPTION_4BIT	The option code is 4 bits long.
3	OPTION_8BIT	The option code is 8 bits long.

Bit 7 – DATAEN Data Enable

Value	Description
0	No data is sent/received to/from the serial Flash memory.
1	Data is sent/received to/from the serial Flash memory.

Bit 6 – OPTEN Option Enable

Value	Description
0	The option is not sent to the serial Flash memory.
1	The option is sent to the serial Flash memory.

Bit 5 – ADDREN Address Enable

Value	Description
0	The transfer address is not sent to the serial Flash memory.
1	The transfer address is sent to the serial Flash memory.

Bit 4 – INSTEN Instruction Enable

Value	Description
0	The instruction is not sent to the serial Flash memory.
1	The instruction is sent to the serial Flash memory.

Bits 2:0 – WIDTH[2:0] Width of Instruction Code, Address, Option Code and Data

Value	Name	Description
0	SINGLE_BIT_SPI	Instruction: Single-bit SPI / Address-Option: Single-bit SPI / Data: Single-bit SPI
1	DUAL_OUTPUT	Instruction: Single-bit SPI / Address-Option: Single-bit SPI / Data: Dual SPI
2	QUAD_OUTPUT	Instruction: Single-bit SPI / Address-Option: Single-bit SPI / Data: Quad SPI
3	DUAL_IO	Instruction: Single-bit SPI / Address-Option: Dual SPI / Data: Dual SPI
4	QUAD_IO	Instruction: Single-bit SPI / Address-Option: Quad SPI / Data: Quad SPI
5	DUAL_CMD	Instruction: Dual SPI / Address-Option: Dual SPI / Data: Dual SPI
6	QUAD_CMD	Instruction: Quad SPI / Address-Option: Quad SPI / Data: Quad SPI

Downloaded from Arrow.com.

46.7.13 QSPI Read Instruction Code Register

Name: QSPI_RICR
Offset: 0x3C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RDOPT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RDINST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – RDOPT[7:0] Read Option Code
Option code to send to the serial Flash memory in case of read transfer.

Bits 7:0 – RDINST[7:0] Read Instruction Code
Instruction code to send to the serial Flash memory in case of read transfer.

Downloaded from Arrow.com.

46.7.14 QSPI Scrambling Mode Register

Name: QSPI_SMR
Offset: 0x40
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						SCRKL	RVDIS	SCREN
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SCRKL Scrambling Key Lock

Value	Description
0	No action.
1	QSPI_SKR.USRK cannot be written until the next VDDCORE reset.

Bit 1 – RVDIS Scrambling/Unscrambling Random Value Disable

Value	Description
0	The scrambling/unscrambling algorithm includes the user scrambling key plus a random value that may differ between devices.
1	The scrambling/unscrambling algorithm includes only the user scrambling key.

Bit 0 – SCREN Scrambling/Unscrambling Enable
0 (DISABLED): The scrambling/unscrambling is disabled.
1 (ENABLED): The scrambling/unscrambling is enabled.

Downloaded from Arrow.com.

46.7.15 QSPI Scrambling Key Register

Name: QSPI_SKR
Offset: 0x44
Reset: –
Property: Write-only

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	USRK[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	USRK[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	USRK[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	USRK[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – USRK[31:0] User Scrambling Key

Downloaded from Arrow.com.

46.7.16 QSPI Write Protection Mode Register

Name: QSPI_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						WPCREN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x515350	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 2 – WPCREN Write Protection Control Register Enable

Value	Description
0	Disables the write protection on the Control register if WPKEY corresponds to 0x515350.
1	Enables the write protection on the Control register if WPKEY corresponds to 0x515350.

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on Interrupt registers if WPKEY corresponds to 0x515350.
1	Enables the write protection on Interrupt registers if WPKEY corresponds to 0x515350.

Bit 0 – WPEN Write Protection Enable

See section Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x515350 (QSP in ASCII)
1	Enables the write protection if WPKEY corresponds to 0x515350 (QSP in ASCII)

Downloaded from Arrow.com.

46.7.17 QSPI Write Protection Status Register

Name: QSPI_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the QSPI_WPSR.
1	A write protection violation has occurred since the last read of the QSPI_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

47. Secure Digital MultiMedia Card Controller (SDMMC)

47.1 Description

The Secure Digital MultiMedia Card Controller (SDMMC) supports the embedded MultiMedia Card (e.MMC) Specification V4.51, the SD Memory Card Specification V3.0, and the SDIO V3.0 specification. It is compliant with the SD Host Controller Standard V3.0 specification.

The SDMMC includes the register set defined in the “SD Host Controller Simplified Specification V3.00” and additional registers to manage e.MMC devices and enhanced features.

The SDMMC is clocked by three asynchronous clocks and requires the PMC to be configured first.

47.2 Embedded Characteristics

- Compatible with SD Host Controller Standard Specification Version 3.00
- Compatible with MultiMedia Card Specification Version V4.51
- Compatible with SD Memory Card Specification Version 3.00
- Compatible with SDIO Specification Version 3.00
- Support for 1-bit/4-bit SD/SDIO Devices
- Support for 1-bit/4-bit e.MMC Devices
- Support for SD/SDIO Default Speed (Maximum SDCLK Frequency = 25 MHz)
- Support for SD/SDIO High Speed (Maximum SDCLK Frequency = 50 MHz)
- Support for SDSC, SDHC and SDXC
- Support for MMC/e.MMC Default Speed (Maximum SDCLK Frequency = 26 MHz)
- Support for MMC/e.MMC High Speed (Maximum SDCLK Frequency = 52 MHz)
- Support for e.MMC High Speed DDR (Maximum SDCLK Frequency = 52 MHz)
- e.MMC Boot Operation Mode Support
- Support for Block Size from 1 to 512 Bytes
- Support for Stream, Block and Multiblock Data Read and Write
  - Advanced DMA and SDMA capability
- Internal 1024-byte Dual Port RAM
- Support for both Synchronous and Asynchronous Abort
- Supports for SDIO Card Interrupt

47.3 Reference Documents

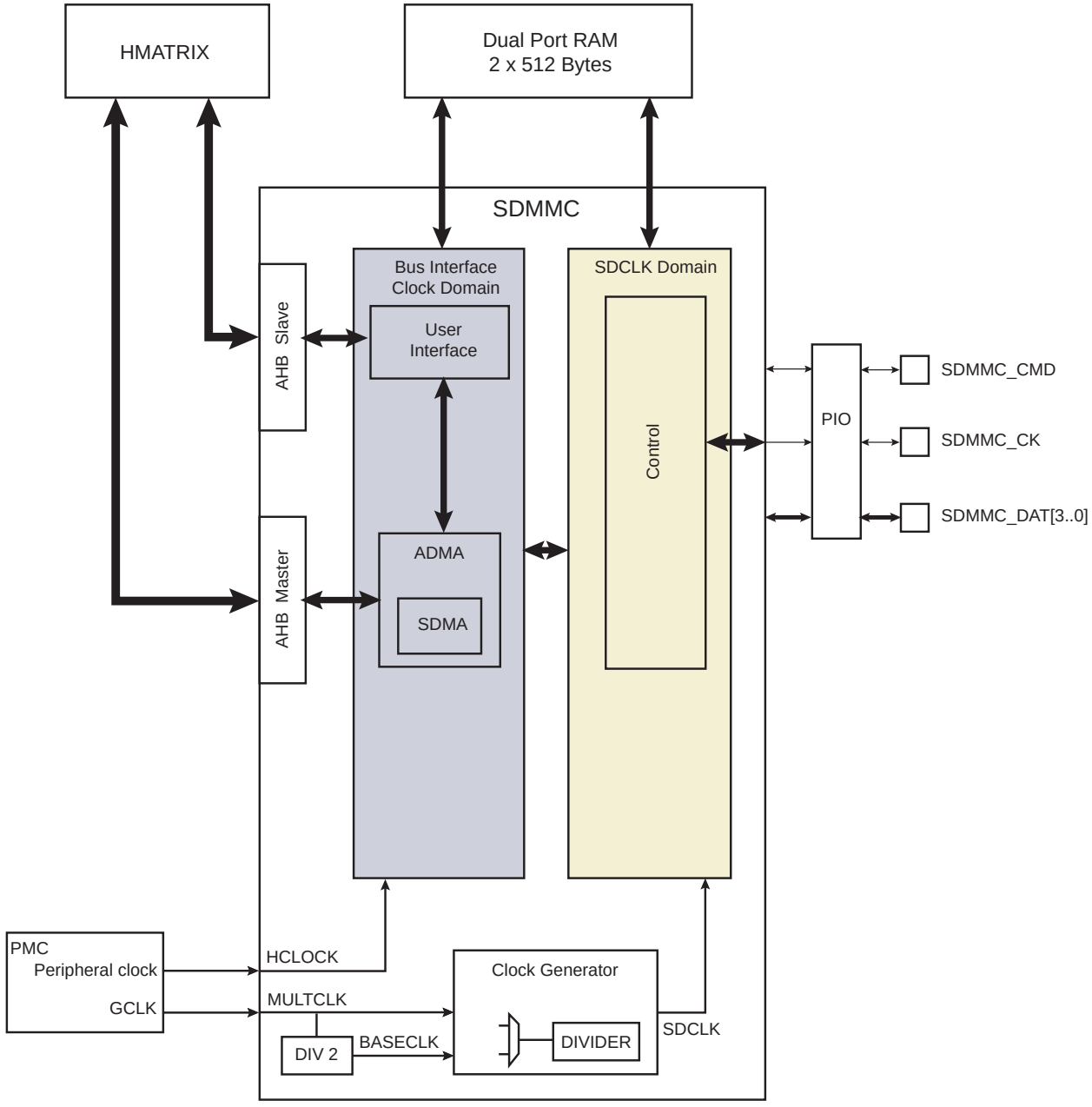
Table 47-1. Reference Documents

Name	Link
SD Host Controller Simplified Specification V3.00	https://www.sdcard.org
SDIO Simplified Specification V3.00	
Physical Layer Simplified Specification V3.01	
Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51	http://www.jedec.org

Downloaded from Arrow.com.

47.4 Block Diagram

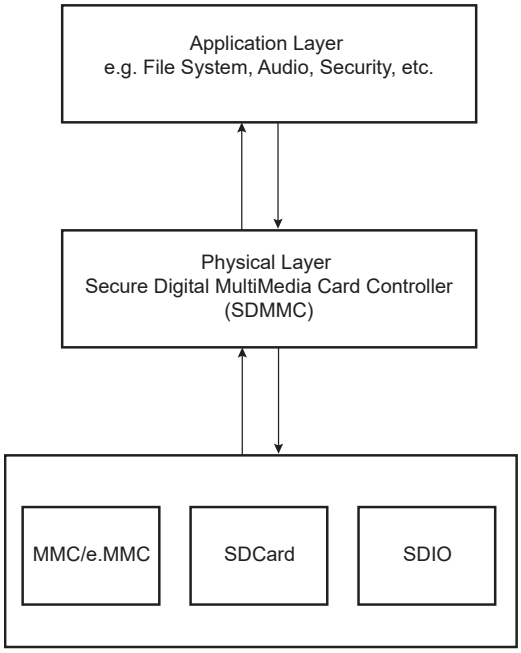
Figure 47-1. SDMMC Block Diagram



Downloaded from Arrow.com.

47.5 Application Block Diagram

Figure 47-2. Application Block Diagram



47.6 Pin Name List

Table 47-2. I/O Lines Description for 8-bit Configuration

Pin Name(1)	Pin Description	Type
SDMMC_CMD	SDCard / SDIO / e.MMC Command/Response Line	I/O
SDMMC_CK	SDCard / SDIO / e.MMC Clock Signal	Output
SDMMC_DAT[3..0]	SDCard / SDIO / e.MMC Data Lines	I/O

Notes: 1. When several SDMMCs are embedded in a product, SDMMC_CK refers to SDMMCx_CK, SDMMC_CMD to SDMMCx_CMD, SDMMC_DATy to SDMMCx_DATy.

47.7 Product Dependencies

47.7.1 I/O Lines

The pins used for interfacing the Secure Digital MultiMedia Card (SDMMC) Controller are multiplexed with PIO lines. The programmer must first program the PIO controller to assign the peripheral functions to SDMMC pins.

47.7.2 Power Management

The SDMMC is clocked through the Power Management Controller (PMC), so the programmer must first configure the PMC to enable the SDMMC clocks.

47.7.3 Interrupt Sources

The SDMMC has an interrupt line connected to the interrupt controller.

Handling the SDMMC interrupt requires programming the interrupt controller before configuring the SDMMC.

Downloaded from Arrow.com.

47.8 SD/SDIO Operating Mode

The SDMMC is fully compliant with the “SD Host Controller Simplified Specification V3.00” for SD/SDIO devices. See this specification for the SDMMC configuration.

See “Physical Layer Simplified Specification V3.01” and “SDIO Simplified Specification V3.00” for SD/SDIO management.

47.9 e.MMC Operating Mode

The SDMMC supports management of e.MMC devices. As the “SD Host Controller Simplified Specification V3.00” does not apply to e.MMC devices, some registers have been added to those described in this specification in order to manage e.MMC devices. Most of the registers described in the “SD Host Controller Simplified Specification V3.00” must be used for e.MMC management, but e.MMC-specific features are managed using SDMMC_MC1R and SDMMC_MC2R.

47.9.1 Boot Operation Mode

In Boot Operation mode, the processor can read boot data from the e.MMC device by keeping the CMD line low after poweron before issuing the CMD1. The data can be read from either one of the boot partitions or the user area according to BOOT_PARTITION_ENABLE in the Extended CSD register (see “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51”).

47.9.1.1 Boot Procedure, Processor Mode

1. Configure the SDMMC:
   a. Set the data bus width using SDMMC_HC1R.DW and SDMMC_HC1R.EXTDW according to the BOOT_BUS_WIDTH in the Extended CSD Register (see “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51”).

   b. Select the speed mode (using SDMMC_HC1R.HSEN or SDMMC_MC1R.DDR) according to BOOT_MODE in the Extended CSD Register.

   c. Set the SDCLK frequency according to the selected speed mode.

   d. If the Boot Acknowledge is sent by the e.MMC device (BOOT_ACK = 1 in the Extended CSD Register), set the Boot Acknowledge Enable to ‘1’ (SDMMC_MC1R.BOOTA = 1).

   e. Enable the interrupt on Boot Acknowledge Received (SDMMC_NISTER.BOOTAR = 1 and SDMMC_NISIER.BOOTAR = 1).

   f. Set the e.MMC Command Type to BOOT (SDMMC_MC1R.CMDTYP = 3)

   g. Set SDMMC_TMR to read multiple blocks for the e.MMC device (SDMMC_TMR.MSBSEL = 1 and SDMMC_TMR.DTDSEL = 1).

   h. Select the NonDMA transfer (SDMMC_TMR.DMAEN = 0).

   i. Optional: select the Auto CMD method (using SDMMC_TMR.ACMDEN).

   j. Set the block size to 512 bytes (SDMMC_BSR.BLKSIZE = 512).

   k. Set the required number of read blocks (using SDMMC_BCR.BLKCNT). SDMMC_TMR.BCEN must be set to ‘1’.
2. Write SDMMC_CR = 20(hexa) to set the e.MMC in Boot Operation mode.
3. Wait for interrupt on Boot Acknowledge Received (BOOTAR).
4. The user can copy the boot data sequentially as soon as the BRDRDY flag is asserted.
5. When the data transfer is completed, the boot operation must be terminated by setting SDMMC_MC2R.ABOOT to ‘1’.

47.9.1.2 Boot Procedure, SDMA Mode

1. Configure SDMMC:

Downloaded from Arrow.com.

a. Set the data bus width using SDMMC_HC1R.DW and SDMMC_HC1R.EXTDW according to BOOT_BUS_WIDTH in the Extended CSD Register (see “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51”).

b. Select the speed mode (SDMMC_HC1R.HSEN or SDMMC_MC1R.DDR) according to BOOT_MODE in the Extended CSD Register.

c. Set the SDCLK frequency according to the selected speed mode.

d. If the Boot Acknowledge is sent by the e.MMC device (BOOT_ACK = 1 in the Extended CSD Register), set the Boot Acknowledge Enable to 1 (SDMMC_MC1R.BOOTA = 1).

e. Enable interrupt on Boot Acknowledge Received (SDMMC_NISTER.BOOTAR = 1 and SDMMC_NISIER.BOOTAR = 1).

f. Set the e.MMC Command Type to BOOT (SDMMC_MC1R.CMDTYP = 3).

g. Set SDMMC_TMR to read multiple blocks for the e.MMC device (SDMMC_TMR.MSBSEL = 1 and SDMMC_TMR.TDSEL = 1).

h. Select the SDMA transfer (SDMMC_TMR.DMAEN = 1 and SDMMC_HC1R.DMASEL = 0).

i. Write the SDMA system address where the boot data will be copied (SDMMC_SSAR.ADDR).

j. Optional: select the Auto CMD method (SDMMC_TMR.ACMDEN).
Note: Auto CMD23 cannot be used with SDMA.

k. Set the block size to 512 bytes (SDMMC_BSR.BLKSIZE = 512).

l. Set the required number of read blocks (SDMMC_BCR.BLKCNT). SDMMC_TMR.BCEN must be set to 1.

2. Write SDMMC_CR = 20(hexa) to set the e.MMC in Boot Operation mode.
3. Wait for interrupt on Boot Acknowledge Received (BOOTAR).
4. The user can copy the boot data sequentially as soon as the BRDRDY flag is asserted.
5. When the data transfer is completed, the boot operation must be terminated by setting SDMMC_MC2R.ABOOT to ‘1’.

47.9.1.3 Boot Procedure, ADMA Mode

1. Configure the SDMMC:
   a. Set the data bus width using SDMMC_HC1R.DW and SDMMC_HC1R.EXTDW according to BOOT_BUS_WIDTH in the Extended CSD Register (see “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51”).

   b. Select the speed mode (SDMMC_HC1R.HSEN or SDMMC_MC1R.DDR) according to BOOT_MODE in the Extended CSD register.

   c. Set the SDCLK frequency according to the selected speed mode.

   d. If the Boot Acknowledge is sent by the e.MMC device (BOOT_ACK = 1 in the Extended CSD Register), set the Boot Acknowledge Enable to ‘1’ (SDMMC_MC1R.BOOTA = 1).

   e. Enable interrupt on Boot Acknowledge Received (SDMMC_NISTER.BOOTAR = 1 and SDMMC_NISIER.BOOTAR = 1).

   f. Set the e.MMC Command Type to BOOT (SDMMC_MC1R.CMDTYP = 3).

   g. Set SDMMC_TMR to read multiple blocks for the e.MMC device (SDMMC_TMR.MSBSEL = 1 and SDMMC_TMR.DTDSEL = 1).

   h. Select the ADMA transfer (SDMMC_TMR.DMAEN = 1 and SDMMC_HC1R.DMASEL = 2 or 3).

   i. Write the address of the descriptor table in the ADMA system address (SDMMC_ASARx [0..1].ADMASA).

   j. Optional: select the Auto CMD method (SDMMC_TMR.ACMDEN).

   k. Set the block size to 512 bytes (SDMMC_BSR.BLKSIZE = 512).

   l. Set the required number of read blocks (SDMMC_BCR.BLKCNT). SDMMC_TMR.BCEN must be set to ‘1’.

2. Write SDMMC_CR = 20(hexa) to set the e.MMC in Boot Operation Mode.
3. Wait for interrupt on Boot Acknowledge Received (BOOTAR).

Downloaded from Arrow.com.

4. The user can copy the boot data sequentially as soon as the BRDRDY flag is asserted.
5. When the data transfer is completed, the boot operation must be terminated by setting SDMMC_MC2R.ABOOT to ‘1’.

Downloaded from Arrow.com.

47.10 Register Summary

Offset	Name	Bit Pos.								
0x00	SDMMC_SSAR	31:24	ADDR/ARG2[31:24]							
		23:16	ADDR/ARG2[23:16]							
		15:8	ADDR/ARG2[15:8]							
		7:0	ADDR/ARG2[7:0]							
0x04	SDMMC_BSR	15:8		BOUNDARY[2:0]					BLKSIZE[9:8]	
		7:0	BLKSIZE[7:0]							
0x06	SDMMC_BCR	15:8	BLKCNT[15:8]							
		7:0	BLKCNT[7:0]							
0x08	SDMMC_ARG1R	31:24	ARG1[31:24]							
		23:16	ARG1[23:16]							
		15:8	ARG1[15:8]							
		7:0	ARG1[7:0]							
0x0C	SDMMC_TMR	15:8								
		7:0			MSBSEL	DTDSEL	ACMDEN[1:0]		BCEN	DMAEN
0x0E	SDMMC_CR	15:8			CMDIDX[5:0]					
		7:0	CMDTYP[1:0]		DPSEL	CMDICEN	CMDCCEN		RESPTYP[1:0]	
0x10	SDMMC_RR0	31:24	CMDRESP[31:24]							
		23:16	CMDRESP[23:16]							
		15:8	CMDRESP[15:8]							
		7:0	CMDRESP[7:0]							
0x14	SDMMC_RR1	31:24	CMDRESP[31:24]							
		23:16	CMDRESP[23:16]							
		15:8	CMDRESP[15:8]							
		7:0	CMDRESP[7:0]							
0x18	SDMMC_RR2	31:24	CMDRESP[31:24]							
		23:16	CMDRESP[23:16]							
		15:8	CMDRESP[15:8]							
		7:0	CMDRESP[7:0]							
0x1C	SDMMC_RR3	31:24	CMDRESP[31:24]							
		23:16	CMDRESP[23:16]							
		15:8	CMDRESP[15:8]							
		7:0	CMDRESP[7:0]							
0x20	SDMMC_BDPR	31:24	BUFDATA[31:24]							
		23:16	BUFDATA[23:16]							
		15:8	BUFDATA[15:8]							
		7:0	BUFDATA[7:0]							
0x24	SDMMC_PSR	31:24								CMDLL
		23:16	DATLL[3:0]							
		15:8					BUFRDEN	BUFWREN	RTACT	WTACT
		7:0						DLACT	CMDINHD	CMDINHC
0x28	SDMMC_HC1R (SD_SDIO)	7:0		CARDDTL		DMASEL[1:0]		HSEN	DW	LEDCTRL
0x28	SDMMC_HC1R (e.MMC)	7:0			EXTDW	DMASEL[1:0]		HSEN	DW	
0x29	SDMMC_PCR	7:0								SDBPWR
0x2A	SDMMC_BGCR (SD_SDIO)	7:0					INTBG	RWCTRL	CONTR	STPBGR
0x2A	SDMMC_BGCR (e.MMC)	7:0							CONTR	STPBGR
0x2B	SDMMC_WCR (SD_SDIO)	7:0								WKENCINT
0x2C	SDMMC_CCR	15:8	SDCLKFSEL[7:0]							
		7:0	USDCLKFSEL[1:0]		CLKGSEL			SDCLKEN	INTCLKS	INTCLKEN
0x2E	SDMMC_TCR	7:0					DTCVAL[3:0]			
0x2F	SDMMC_SRR	7:0						SWRSTDAT	SWRSTCMD	SWRSTALL

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x30	SDMMC_NISTR (SD_SDIO)	15:8	ERRINT							CINT
		7:0			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x30	SDMMC_NISTR (e.MMC)	15:8	ERRINT	BOOTAR						
		7:0			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x32	SDMMC_EISTR (SD_SDIO)	15:8							ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x32	SDMMC_EISTR (e.MMC)	15:8				BOOTAE			ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x34	SDMMC_NISTER (SD_SDIO)	15:8								CINT
		7:0			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x34	SDMMC_NISTER (e.MMC)	15:8		BOOTAR						
		7:0			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x36	SDMMC_EISTER (SD_SDIO)	15:8							ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x36	SDMMC_EISTER (e.MMC)	15:8				BOOTAE			ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x38	SDMMC_NISIER (SD_SDIO)	15:8								CINT
		7:0			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x38	SDMMC_NISIER (e.MMC)	15:8		BOOTAR						
		7:0			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x3A	SDMMC_EISIER (SD_SDIO)	15:8							ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x3A	SDMMC_EISIER (e.MMC)	15:8				BOOTAE			ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x3C	SDMMC_ACESR	15:8								
		7:0	CMDNI			ACMDIDX	ACMDEND	ACMDCRC	ACMDTEO	ACMD12NE
0x3E	SDMMC_HC2R (SD_SDIO)	15:8	PVALEN	ASINTEN						
		7:0								
0x3E	SDMMC_HC2R (e.MMC)	15:8	PVALEN							
		7:0								
0x40	SDMMC_CA0R	31:24	SLTYPE[1:0]		ASINTSUP	SB64SUP		V18VSUP	V30VSUP	V33VSUP
		23:16	SRSUP	SDMASUP	HSSUP		ADMA2SUP	ED8SUP	MAXBLKL[1:0]	
		15:8	BASECLKF[7:0]							
		7:0	TEOCLKU		TEOCLKF[5:0]					
0x44	SDMMC_CA1R	31:24								
		23:16	CLKMULT[7:0]							
		15:8								
		7:0		DRVDSUP	DRVCSUP	DRVASUP		DDR50SUP	SDR104SUP	SDR50SUP
0x48	SDMMC_MCCAR	31:24								
		23:16	MAXCUR18V[7:0]							
		15:8	MAXCUR30V[7:0]							
		7:0	MAXCUR33V[7:0]							
0x4C ... 0x4F	Reserved									
0x50	SDMMC_FERACES	15:8								
		7:0	CMDNI			ACMDIDX	ACMDEND	ACMDCRC	ACMDTEO	ACMD12NE
0x52	SDMMC_FEREIS	15:8				BOOTAE			ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x54	SDMMC_AESR	7:0						LMIS	ERRST[1:0]	
0x55 ... 0x57	Reserved									
0x58	SDMMC_ASAR0	31:24	ADMASA[31:24]							
		23:16	ADMASA[23:16]							
		15:8	ADMASA[15:8]							
		7:0	ADMASA[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.									
0x5C ... 0x5F	Reserved										
0x60	SDMMC_PVR0	15:8						CLKGSEL	SDCLKFSEL[9:8]		
		7:0	SDCLKFSEL[7:0]								
0x62	SDMMC_PVR1	15:8						CLKGSEL	SDCLKFSEL[9:8]		
		7:0	SDCLKFSEL[7:0]								
0x64	SDMMC_PVR2	15:8						CLKGSEL	SDCLKFSEL[9:8]		
		7:0	SDCLKFSEL[7:0]								
0x66	SDMMC_PVR3	15:8						CLKGSEL	SDCLKFSEL[9:8]		
		7:0	SDCLKFSEL[7:0]								
0x68 ... 0xFB	Reserved										
0xFC	SDMMC_SISR	15:8									
		7:0							INTSSL[1:0]		
0xFE	SDMMC_HCVR	15:8	VVER[7:0]								
		7:0	SVER[7:0]								
0x0100 ... 0x01FF	Reserved										
0x0200	SDMMC_APSR	31:24									
		23:16									
		15:8									
		7:0					HDATLL[3:0]				
0x0204	SDMMC_MC1R	7:0			BOOTA	OPD	DDR		CMDTYP[1:0]		
0x0205	SDMMC_MC2R	7:0							ABOOT	SRESP	
0x0206 ... 0x0207	Reserved										
0x0208	SDMMC_ACR	31:24									
		23:16									
		15:8									
		7:0							BMAX[1:0]		
0x020C	SDMMC_CC2R	31:24									
		23:16									
		15:8									
		7:0								FSDCLKD	
0x0210 ... 0x022F	Reserved										
0x0230	SDMMC_CACR	31:24									
		23:16									
		15:8	KEY[7:0]								
		7:0								CAPWREN	
0x0234	SDMMC_DBGR	31:24									
		23:16									
		15:8									
		7:0								NIDBG	

Downloaded from Arrow.com.

47.10.1 SDMMC SDMA System Address / Argument 2 Register

Name: SDMMC_SSAR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

This register contains the physical system memory address used for SDMA transfers or the second argument for Auto CMD23.

Bit	31	30	29	28	27	26	25	24
	ADDR/ARG2[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR/ARG2[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR/ARG2[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR/ARG2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR/ARG2[31:0] SDMA System Address/Argument 2
ADDR: This field is the system memory address for a SDMA transfer. When the SDMMC stops an SDMA transfer, this field points to the system address of the next contiguous data position. This field can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value. An interrupt can be generated to instruct the software to update this field. Writing the next system address of the next data position restarts the SDMA transfer.
ARG2: This field is used with Auto CMD23 to set a 32-bit block count value to the CMD23 argument while executing Auto CMD23. If Auto CMD23 is used with ADMA, the full 32-bit block count value can be used. If Auto CMD23 is used without ADMA, the available block count value is limited by SDMMC_BCR. In this case, 65535 blocks is the maximum value.

Downloaded from Arrow.com.

47.10.2 SDMMC Block Size Register

Name: SDMMC_BSR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

Bit	15	14	13	12	11	10	9	8
		BOUNDARY[2:0]					BLKSIZE[9:8]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		0	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	BLKSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 14:12 – BOUNDARY[2:0] SDMA Buffer Boundary
This field specifies the size of the contiguous buffer in the system memory. The SDMA transfer waits at every boundary specified by this field and the SDMMC generates the DMA Interrupt to instruct the software to update SDMMC_SSAR. If this field is set to 0 (buffer size = 4 Kbytes), the lowest 12 bits of SDMMC_SSAR.ADDRESS point to data in the contiguous buffer, and the upper 20 bits point to the location of the buffer in the system memory. This function is active when SDMMC_TMR.DMAEN is set.

Value	Name	Description
0	4K	4-Kbyte boundary
1	8K	8-Kbyte boundary
2	16K	16-Kbyte boundary
3	32K	32-Kbyte boundary
4	64K	64-Kbyte boundary
5	128K	128-Kbyte boundary
6	256k	256-Kbyte boundary
7	512K	512-Kbyte boundary

Bits 9:0 – BLKSIZE[9:0] Transfer Block Size
This field specifies the block size of data transfers for CMD14, CMD17, CMD18, CMD19, CMD24, CMD25, CMD53 and other data transfer commands such as CMD6, CMD8, ACMD13 and ACMD51. Values ranging from 1 to 512 can be set. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations are ignored.

Downloaded from Arrow.com.

47.10.3 SDMMC Block Count Register

Name: SDMMC_BCR
Offset: 0x06
Reset: 0x0000
Property: Read/Write

Bit	15	14	13	12	11	10	9	8
	BLKCNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BLKCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – BLKCNT[15:0] Block Count for Current Transfer
This field is used only if SDMMC_TMR.BCEN (Block Count Enable) is set to 1 and is valid only for multiple block transfers. BLKCNT is the number of blocks to be transferred and it must be set to a value between 1 and the maximum block count. The SDMMC decrements the block count after each block transfer and stops when the count reaches 0. When this field is set to 0, no data block is transferred.
This register should be accessed only when no transaction is executing (i.e., after transactions are stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored.
When a suspend command is completed, the number of blocks yet to be transferred can be determined by reading this register. Before issuing a resume command, the previously saved block count is restored.

Downloaded from Arrow.com.

47.10.4 SDMMC Argument 1 Register

Name: SDMMC_ARG1R
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ARG1[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ARG1[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ARG1[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ARG1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ARG1[31:0] Argument 1
This register contains the SD command argument which is specified as the bit 39-8 of Command-Format in the “Physical Layer Simplified Specification V3.01” or “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51” .

Downloaded from Arrow.com.

47.10.5 SDMMC Transfer Mode Register

Name: SDMMC_TMR
Offset: 0x0C
Reset: 0x0000
Property: Read/Write

This register is used to control data transfers. The user shall set this register before issuing a command which transfers data (see SDMMC_CR.DPSEL), or before issuing a Resume command. The user must save the value of this register when the data transfer is suspended (as a result of a Suspend command) and restore it before issuing a Resume command. To prevent data loss, this register cannot be written while data transactions are in progress. Writes to this register are ignored when SDMMC_PSR.CMDINHD is 1.

Table 47-3. Determining the Transfer Type

MSBSEL	BCEN	BLKCNT (SDMMC_BCR)	Function
0	Don't care	Don't care	Single Transfer
1	0	Don't care	Infinite Transfer
1	1	Not Zero	Multiple Transfer
1	1	Zero	Stop Multiple Transfer

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			MSBSEL	DTDSEL	ACMDEN[1:0]		BCEN	DMAEN
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – MSBSEL Multi/Single Block Selection
This bit is set to 1 when issuing multiple-block transfer commands using DAT line(s). For any other commands, set this bit to 0. If this bit is 0, it is not necessary to set SDMMC_BCR (see the table above, "Determining the Transfer Type").

Bit 4 – DTDSEL Data Transfer Direction Selection
This bit defines the direction of the DAT lines data transfers. Set this bit to 1 to transfer data from the device (SD Card/SDIO/e.MMC) to the SDMMC, and to 0 for all other commands.
0 (WRITE): Writes data from the SDMMC to the device.
1 (READ): Reads data from the device to the SDMMC.

Bits 3:2 – ACMDEN[1:0] Auto Command Enable
Two methods can be used to stop Multiple-block read and write operation:

- Auto CMD12: when the ACMDEN field is set to 1, the SDMMC issues CMD12 automatically when the last block transfer is completed. An Auto CMD12 error is indicated to SDMMC_ACESR. Auto CMD12 is not enabled if the command does not require CMD12.
- Auto CMD23: when the ACMDEN field is set to 2, the SDMMC issues a CMD23 automatically before issuing a command specified in SDMMC_CR.

The following conditions are required to use Auto CMD23:

- A memory card that supports CMD23 (SCR[33] = 1)
- If DMA is used, it must be ADMA (SDMA not supported).
- Only CMD18 or CMD25 is issued.

Note: The SDMMC does not check the command index.

Downloaded from Arrow.com.

Auto CMD23 can be used with or without ADMA. By writing SDMMC_CR, the SDMMC issues a CMD23 first and then issues a command specified by the SDMMC_CR.CMDIDX field. If CMD23 response errors are detected, the second command is not issued. A CMD23 error is indicated in SDMMC_ACESR. The CMD23 argument (32-bit block count value) is set in SDMMC_SSAR.
This field determines the use of auto command functions.

Value	Name	Description
0	DISABLED	Auto Command Disabled
1	CMD12	Auto CMD12 Enabled
2	CMD23	Auto CMD23 Enabled
3	–	Reserved

Bit 1 – BCEN Block Count Enable
This bit is used to enable SDMMC_BCR, which is only relevant for multiple block transfers. When this bit is 0, SDMMC_BCR is disabled, which is useful when executing an infinite transfer (see the table above). If an ADMA2 transfer is more than 65535 blocks, this bit is set to 0 and the data transfer length is designated by the Descriptor Table.
0 (DISABLED): Block count is disabled.
1 (ENABLED): Block count is enabled.

Bit 0 – DMAEN DMA Enable
This bit enables the DMA functionality described in section “Supporting DMA” in “SD Host Controller Simplified Specification V3.00” . DMA can be enabled only if it is supported as indicated by the bit SDMMC_CA0R.ADMA2SUP. One of the DMA modes can be selected using the field SDMMC_HC1R.DMASEL. If DMA is not supported, this bit is meaningless and then always reads 0. When this bit is set to 1, a DMA operation begins when the user writes to the upper byte of SDMMC_CR.
0 (DISABLED): DMA functionality is disabled.
1 (ENABLED): DMA functionality is enabled.

Downloaded from Arrow.com.

47.10.6 SDMMC Command Register

Name: SDMMC_CR
Offset: 0x0E
Reset: 0x0000
Property: Read/Write

Bit	15	14	13	12	11	10	9	8
			CMDIDX[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CMDTYP[1:0]		DPSEL	CMDICEN	CMDCCEN		RESPTYP[1:0]	
Access	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Reset	0	0	0	0	0		0	0

Bits 13:8 – CMDIDX[5:0] Command Index
This bit shall be set to the command number (CMD0–63, ACMD0–63) that is specified in bits 45–40 of the Command-Format in the “Physical Layer Simplified Specification V3.01” , “SDIO Simplified Specification V3.00” , and “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51” .

Bits 7:6 – CMDTYP[1:0] Command Type

Value	Name	Description
0	NORMAL	Other commands
1	SUSPEND	CMD52 to write “Bus Suspend” in the Card Common Control Registers (CCCR) (for SDIO only)
2	RESUME	CMD52 to write “Function Select” in the Card Common Control Registers (CCCR) (for SDIO only)
3	ABORT	CMD12, CMD52 to write “I/O Abort” in the Card Common Control Registers (CCCR) (for SDIO only)

Bit 5 – DPSEL Data Present Select
This bit is set to 1 to indicate that data is present and shall be transferred using the DAT lines. It is set to 0 for the following:

- Commands using only CMD line (Ex. CMD52)
- Commands with no data transfer but using Busy signal on DAT[0] line (Ex. CMD38)
- Resume command

Value	Description
0	No data present
1	Data present

Bit 4 – CMDICEN Command Index Check Enable
If this bit is set to 1, the SDMMC checks the Index field in the response to see if it has the same value as the command index. If it has not, it is reported as a Command Index Error (CMDIDX) in SDMMC_EISTR. If this bit is set to 0, the Index field of the response is not checked.
0 (DISABLED): The Command Index Check is disabled.
1 (ENABLED): The Command Index Check is enabled.

Bit 3 – CMDCCEN Command CRC Check Enable
If this bit is set to 1, the SDMMC checks the CRC field in the response. If an error is detected, it is reported as a Command CRC Error (CMDCRC) in SDMMC_EISTR. If this bit is set to 0, the CRC field is not checked. The position of the CRC field is determined according to the length of the response.
0 (DISABLED): The Command CRC Check is disabled.
1 (ENABLED): The Command CRC Check is enabled.

Downloaded from Arrow.com.

Bits 1:0 – RESPTYP[1:0] Response Type
This field is set according to the response type expected for the command index (CMDIDX).

Value	Name	Description
0	NORESP	No Response
1	RL136	Response Length 136
2	RL48	Response Length 48
3	RL48BUSY	Response Length 48 with Busy

Downloaded from Arrow.com.

47.10.7 SDMMC Response Register x

Name: SDMMC_RRx
Offset: 0x10 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	CMDRESP[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CMDRESP[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CMDRESP[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CMDRESP[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CMDRESP[31:0] Command Response
The table below describes the mapping of command responses from the SD_SDIO/e.MMC bus to these registers for each responses type. In this table, R[] refers to a bit range of the response data as transmitted on the SD_SDIO/e.MMC bus.

Type of response	Meaning of response	Response field	Response register
R1, R1b (normal response)	Card Status	R[39:8]	SDMMC_RR0[31:0]
R1b (Auto CMD12 response)	Card Status for Auto CMD12	R[39:8]	SDMMC_RR3[31:0]
R1 (Auto CMD23 response)	Card Status for Auto CMD23	R[39:8]	SDMMC_RR3[31:0]
R2 (CID, CSD register)	CID or CSD register	R[127:8]	SDMMC_RR0[31:0] SDMMC_RR1[31:0] SDMMC_RR2[31:0] SDMMC_RR3[23:0]
R3 (OCR register)	OCR register for memory	R[39:8]	SDMMC_RR0[31:0]
R4 (OCR register)	OCR register for I/O	R[39:8]	SDMMC_RR0[31:0]
R5, R5b	SDIO response	R[39:8]	SDMMC_RR0[31:0]
R6 (Published RCA response)	New published RCA[31:16] and Card status bits	R[39:8]	SDMMC_RR0[31:0]

Downloaded from Arrow.com.

47.10.8 SDMMC Buffer Data Port Register

Name: SDMMC_BDPR
Offset: 0x20
Reset: –
Property: Read/Write

Note: The reset value is an unpredictable value read from the dual port RAM.

Bit	31	30	29	28	27	26	25	24
	BUFDATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	BUFDATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	BUFDATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	BUFDATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – BUFDATA[31:0] Buffer Data
The SDMMC data buffer can be accessed through this 32-bit Data Port register.

Downloaded from Arrow.com.

47.10.9 SDMMC Present State Register

Name: SDMMC_PSR
Offset: 0x24
Reset: 0x00F80000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
								CMDLL
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
	DATLL[3:0]							
Access	R	R	R	R				
Reset	1	1	1	1				

Bit	15	14	13	12	11	10	9	8
					BUFRDEN	BUFWREN	RTACT	WTACT
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
						DLACT	CMDINHD	CMDINHC
Access						R	R	R
Reset						0	0	0

Bit 24 – CMDLL CMD Line Level
This status is used to check the CMD line level to recover from errors, and for debugging.

Bits 23:20 – DATLL[3:0] DAT[3:0] Line Level
This status is used to check the DAT line level to recover from errors, and for debugging. This is especially useful in detecting the Busy signal level from DAT[0].

Bit 11 – BUFRDEN Buffer Read Enable
This bit is used for nonDMA read transfers. This flag indicates that valid data exists in the SDMMC data buffer. If this bit is 1, readable data exists in the buffer.
A change from 1 to 0 occurs when all the block data is read from the buffer.
A change from 0 to 1 occurs when block data is ready in the buffer. This raises the Buffer Read Ready (BRDRDY) status flag in SDMMC_NISTR if SDMMC_NISTER.BRDRDY is set to 1. An interrupt is generated if SDMMC_NISIER.BRDRDY is set to 1.

Bit 10 – BUFWREN Buffer Write Enable
This bit is used for nonDMA write transfers. This flag indicates if space is available for write data. If this bit is 1, data can be written to the buffer.
A change from 1 to 0 occurs when all the block data are written to the buffer.
A change from 0 to 1 occurs when top of block data can be written to the buffer. This raises the Buffer Write Ready (BWRRDY) status flag in SDMMC_NISTR if SDMMC_NISTER.BWRRDY is set to 1. An interrupt is generated if SDMMC_NISIER.BWRRDY is set to 1.

Bit 9 – RTACT Read Transfer Active
This bit is used to detect completion of a read transfer. See section “Read Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.
This bit is set to 1 in either of the following conditions:

- After the end bit of the read command.
- When a read operation is restarted by writing a 1 to SDMMC_BGCR.CONTR (Continue Request).

Downloaded from Arrow.com.

This bit is cleared to 0 in either of the following conditions:

- When the last data block as specified by Transfer Block Size (BLKSIZE) is transferred to the system.
- In case of ADMA2, end of read is designated by the descriptor table.
- When all valid data blocks in the SDMMC have been transferred to the system and no current block transfers are being sent as a result of the Stop At Block Gap Request (STPBGR) of SDMMC_BGCR being set to 1.

A change from 1 to 0 raises the Transfer Complete (TRFC) status flag in SDMMC_NISTR if SDMMC_NISTER.TRFC is set to 1. An interrupt is generated if SDMMC_NISIER.TRFC is set to 1.

Bit 8 – WTACT Write Transfer Active
This bit indicates a write transfer is active. If this bit is 0, it means no valid write data exists in the SDMMC. See section “Write Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.
This bit is set to 1 in either of the following conditions:

- After the end bit of the write command.
- When a write operation is restarted by writing a 1 to SDMMC_BGCR.CONTR (Continue Request).

This bit is cleared to 0 in either of the following conditions:

- After getting the CRC status of the last data block as specified by the transfer count (single and multiple). In case of ADMA2, transfer count is designated by the descriptor table.
- After getting the CRC status of any block where a data transmission is about to be stopped by a Stop At Block Gap Request (STPBGR) of SDMMC_BGCR.

During a write transaction and as the result of the Stop At Block Gap Request (STPBGR) being set, a change from 1 to 0 raises the Block Gap Event (BLKGE) status flag in SDMMC_NISTR if SDMMC_NISTER.BLKGE is set to 1. An interrupt is generated if BLKGE is set to 1 in SDMMC_NISIER. This status is useful to determine whether nonDAT line commands can be issued during Write Busy.

Bit 2 – DLACT DAT Line Active
This bit indicates whether one of the DAT lines on the bus is in use.
In the case of read transactions:

- This status indicates whether a read transfer is executing on the bus. A change from 1 to 0 resulting from setting the Stop At Block Gap Request (STPBGR) raises the Block Gap Event (BLKGE) status flag in SDMMC_NISTR if SDMMC_NISTER.BLKGE is set to 1. An interrupt is generated if SDMMC_NISIER.BLKGE is set to 1. See the section “Read Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for details on timing.
- This bit is set in either of the following cases:
  - After the end bit of the read command.
  - When writing 1 to SDMMC_BGCR.CONTR (Continue Request) to restart a read transfer.
- This bit is SDMMC cleared in either of the following cases:
  - When the end bit of the last data block is sent from the bus to the SDMMC. In case of ADMA2, the last block is designated by the last transfer of the Descriptor Table.
  - When a read transfer is stopped at the block gap initiated by a Stop At Block Gap Request (STPBGR).
- The SDMMC stops a read operation at the start of the interrupt cycle by driving the Read Wait (DAT[2] line) or by stopping the SD Clock. If the Read Wait signal is already driven (due to the fact that the data buffer cannot receive data), the SDMMC can continue to stop the read operation by driving the Read Wait signal. It is necessary to support the Read Wait in order to use the Suspend/Resume operation.

In the case of write transactions:

- This status indicates that a write transfer is executing on the bus. A change from 1 to 0 raises the Transfer Complete (TRFC) status flag in SDMMC_NISTR if SDMMC_NISTER.TRFC is set to 1. An interrupt is generated if SDMMC_NISIER.TRFC is set to 1. See the section “Write Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for details on timing.
- This bit is set in either of the following cases:
  - After the end bit of the write command.
  - When writing 1 to SDMMC_BGCR.CONTR (Continue Request) to continue a write transfer.
- This bit is cleared in either of the following cases:

Downloaded from Arrow.com.

– When the card releases Write Busy of the last data block. If the card does not drive a Busy signal for 8 SDCLK, the SDMMC considers the card drive “Not Busy”. In the case of ADMA2, the last block is designated by the last transfer of the Descriptor Table.
– When the card releases Write Busy prior to wait for write transfer as a result of a Stop At Block Gap Request (STPBGR).

Command with Busy:
This status indicates whether a command that indicates Busy (ex. erase command for memory) is executing on the bus. This bit is set to 1 after the end bit of the command with Busy and cleared when Busy is deasserted. A change from 1 to 0 raises the Transfer Complete (TRFC) status flag in SDMMC_NISTR if SDMMC_NISTER.TRFC is set to 1. An interrupt is generated if SDMMC_NISIER.TRFC is set to 1. See Figures 2.11 to 2.13 in the “SD Host Controller Simplified Specification V3.00” .

Value	Description
0	DAT line inactive.
1	DAT line active.

Bit 1 – CMDINHD Command Inhibit (DAT)
This status bit is 1 if either the DAT Line Active (DLACT) or the Read Transfer Active (RTACT) is set to 1. If this bit is 0, it indicates that the SDMMC can issue the next command. Commands with a Busy signal belong to Command Inhibit (DAT) (ex. R1b, R5b type). A change from 1 to 0 raises the Transfer Complete (TRFC) status flag in SDMMC_NISTR if SDMMC_NISTER.TRFC is set to 1. An interrupt is generated if SDMMC_NISIER.TRFC is set to 1.
Note: The software can save registers in the 000–00Dh range for a suspend transaction after this bit has changed from 1 to 0.

Value	Description
0	Can issue a command which uses the DAT line(s).
1	Cannot issue a command which uses the DAT line(s).

Bit 0 – CMDINHC Command Inhibit (CMD)
If this bit is 0, it indicates the CMD line is not in use and the SDMMC can issue a command using the CMD line. This bit is set to 1 immediately after SDMMC_CR is written. This bit is cleared when the command response is received. Auto CMD12 and Auto CMD23 consist of two responses. In this case, this bit is not cleared by the CMD12 or CMD23 response, but by the Read/Write command response.
Status issuing Auto CMD12 is not read from this bit. So, if a command is issued during Auto CMD12 operation, the SDMMC manages to issue both commands: CMD12 and a command set by SDMMC_CR.
Even if the Command Inhibit (DAT) is set to 1, commands using only the CMD line can be issued if this bit is 0.
A change from 1 to 0 raises the Command Complete (CMDC) status flag in SDMMC_NISTR if SDMMC_NISTER.CMDC is set to 1. An interrupt is generated if SDMMC_NISIER.CMDC is set to 1.
If the SDMMC cannot issue the command because of a command conflict error (see SDMMC_EISTR.CMDCRC) or because of a ‘Command Not Issued By Auto CMD12’ error (see section “SDMMC Auto CMD Error Status Register”), this bit remains 1 and Command Complete is not set.

Value	Description
0	Can issue a command using only CMD line.
1	Cannot issue a command.

Downloaded from Arrow.com.

47.10.10 SDMMC Host Control 1 Register (SD_SDIO)

Name: SDMMC_HC1R (SD_SDIO)
Offset: 0x28
Reset: 0x00
Property: Read/Write

Note: This register configuration is specific to the SD/SDIO operation mode.

Bit	7	6	5	4	3	2	1	0
		CARDDTL		DMASEL[1:0]		HSEN	DW	LEDCTRL
Access		R/W		R/W	R/W	R/W	R/W	R/W
Reset		0		0	0	0	0	0

Bit 6 – CARDDTL Card Detect Test Level
This bit is enabled while the Card Detect Signal Selection (CARDDSEL) is set to 1 and it indicates whether the card is inserted or not.

Value	Description
0	No card.
1	Card inserted.

Bits 4:3 – DMASEL[1:0] DMA Select
One of the supported DMA modes can be selected. The DMA modes supported are given in SDMMC_CA0R. Use of a selected DMA is determined by DMA Enable (DMAEN) in SDMMC_TMR.

Value	Name	Description
0	SDMA	SDMA is selected
1	–	Reserved
2	ADMA32	32-bit Address ADMA2 is selected
3	–	Reserved

Bit 2 – HSEN High Speed Enable
Before setting this bit, the user must check High Speed Support (HSSUP) in SDMMC_CA0R.
If this bit is set to 0 (default), the SDMMC outputs CMD line and DAT lines at the falling edge of the SD clock (up to 25 MHz). If this bit is set to 1, the SDMMC outputs the CMD line and the DAT lines at the rising edge of the SD clock (up to 50 MHz).
If Preset Value Enable (PVALEN) in SDMMC_HC2R is set to 1, the user needs to reset SD Clock Enable (SDCLKEN) before changing this bit to avoid generating clock glitches. After setting this bit to 1, the user sets SDCLEN to 1 again.
Note: This bit is effective only if SDMMC_MC1R.DDR is set to 0.

Note: The clock divider (DIV) in SDMMC_CCR must be set to a value different from 0 when HSEN is 1.

Value	Description
0	Normal Speed mode.
1	High Speed mode.

Bit 1 – DW Data Width
This bit selects the data width of the SDMMC. It must be set to match the data width of the card.
0 (1_BIT): 1-bit mode.
1 (4_BIT): 4-bit mode.
Note: If the Extended Data Transfer Width is 1, this bit has no effect and the data width is 8-bit mode.

Bit 0 – LEDCTRL LED Control
This bit is used to caution the user not to remove the card while it is being accessed. If the software is going to issue multiple commands, this bit is set to 1 during all transactions.
0 (OFF): LED off.
1 (ON): LED on.

Downloaded from Arrow.com.

47.10.11 SDMMC Host Control 1 Register (e.MMC)

Name: SDMMC_HC1R (e.MMC)
Offset: 0x28
Reset: 0x00
Property: Read/Write

Note: This register configuration is specific to the e.MMC operation mode.

Bit	7	6	5	4	3	2	1	0
			EXTDW	DMASEL[1:0]		HSEN	DW	
Access			R/W	R/W	R/W	R/W	R/W	
Reset			0	0	0	0	0	

Bit 5 – EXTDW Extended Data Width
This bit controls the 8-bit Bus Width mode for embedded devices. Support of this function is indicated in 8-bit Support for Embedded Device in SDMMC_CA0R. If a device supports the 8-bit mode, this may be set to 1. If this bit is 0, the bus width is controlled by Data Width (DW).

Bits 4:3 – DMASEL[1:0] DMA Select
One of the supported DAM modes can be selected. The DMA modes supported are given in SDMMC_CA0R. Use of selected DMA is determined by DMA Enable (DMAEN) in SDMMC_TMR.

Value	Name	Description
0	SDMA	SDMA is selected
1	–	Reserved
2	ADMA32	32-bit Address ADMA2 is selected
3	–	Reserved

Bit 2 – HSEN High Speed Enable
Before setting this bit, the user must check High Speed Support (HSSUP) in SDMMC_CA0R.
If this bit is set to 0 (default), the SDMMC outputs CMD line and DAT lines at the falling edge of the SD clock (up to 25 MHz). If this bit is set to 1, the SDMMC outputs the CMD line and the DAT lines at the rising edge of the SD clock (up to 50 MHz).
If Preset Value Enable (PVALEN) in SDMMC_HC2R is set to 1, the user needs to reset the SD Clock Enable (SDCLKEN) before changing this bit to avoid generating clock glitches. After setting this bit to 1, the user sets SDCLEN to 1 again.
Note: This bit is effective only if SDMMC_MC1R.DDR is set to 0.

Note: The clock divider (DIV) in SDMMC_CCR must be set to a value different from 0 when HSEN is 1.

Value	Description
0	Normal Speed mode.
1	High Speed mode.

Bit 1 – DW Data Width
This bit selects the data width of the SDMMC. It must be set to match the data width of the card.
0 (1_BIT): 1-bit mode.
1 (4_BIT): 4-bit mode.
Note: If the Extended Data Transfer Width is 1, this bit has no effect and the data width is 8-bit mode.

Downloaded from Arrow.com.

47.10.12 SDMMC Power Control Register

Name: SDMMC_PCR
Offset: 0x29
Reset: 0x0E
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
								SDBPWR
Access								R/W
Reset								0

Bit 0 – SDBPWR SD Bus Power
This bit is automatically cleared by the SDMMC if the card is removed. If this bit is cleared, the SDMMC stops driving SDMMC_CMD and SDMMC_DAT[3:0] (tri-state) and drives SDMMC_CK to low level.

Downloaded from Arrow.com.

47.10.13 SDMMC Block Gap Control Register (SD_SDIO)

Name: SDMMC_BGCR (SD_SDIO)
Offset: 0x2A
Reset: 0x00
Property: Read/Write

Note: This register configuration is specific to the SD/SDIO operation mode.

Bit	7	6	5	4	3	2	1	0
					INTBG	RWCTRL	CONTR	STPBGR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – INTBG Interrupt at Block Gap
This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. If the SDIO card cannot signal an interrupt during a multiple block transfer, this bit should be set to 0. When the software detects an SDIO card insertion, it sets this bit according to the CCCR of the SDIO card.

Value	Name	Description
0	DISABLED	Interrupt detection disabled.
1	ENABLED	Interrupt detection enabled.

Bit 2 – RWCTRL Read Wait Control
The Read Wait control is optional for SDIO cards. If the card supports Read Wait, set this bit to enable use of the Read Wait protocol to stop read data using the SDMMC_DAT[2] line. Otherwise, the SDMMC stops the SDCLK to hold read data, which restricts command generation. When the software detects an SD card insertion, this bit must be set according to the CCCR of the SDIO card. If the card does not support Read Wait, this bit shall never be set to 1, otherwise an SDMMC_DAT line conflict may occur. If this bit is set to 0, Suspend/Resume cannot be supported.

Value	Description
0	Disables Read Wait control.
1	Enables Read Wait control.

Bit 1 – CONTR Continue Request
This bit is used to restart a transaction which was stopped using a Stop At Block Gap Request (STPBGR). To cancel stop at the block gap, set STPBGR to 0 and set this bit to 1 to restart the transfer.
The SDMMC automatically clears this bit in either of the following cases:
– In the case of a read transaction, the DAT Line Active (DLACT) changes from 0 to 1 as a read transaction restarts.
– In the case of a write transaction, the Write Transfer Active (WTACT) changes from 0 to 1 as the write transaction restarts.
Therefore, it is not necessary to set this bit to 0. If STPBGR is set to 1, any write to this bit is ignored.
See the “Abort Transaction” and “Suspend/Resume” sections in the “SD Host Controller Simplified Specification V3.00” for more details.

Value	Description
0	No effect.
1	Restart.

Bit 0 – STPBGR Stop At Block Gap Request
This bit is used to stop executing read and write transactions at the next block gap for nonDMA, SDMA, and ADMA transfers. The user must leave this bit set to 1 until Transfer Complete (TRFC) in SDMMC_NISTR. Clearing both Stop At Block Gap Request and Continue Request does not cause the transaction to restart. This bit can be set whether the card supports the Read Wait signal or not.
During read transfers, the SDMMC stops the transaction by using the Read Wait signal (SDMMC_DAT[2]) if supported, or by stopping the SD clock otherwise.
In case of write transfers in which the user writes data to SDMMC_BDPR, this bit must be set to 1 after all the block of data is written. If this bit is set to 1, the user does not write data to SDMMC_BDPR.

Downloaded from Arrow.com.

This bit affects Read Transfer Active (RTACT), Write Transfer Active (WTACT), DAT Line Active (DLACT) and Command Inhibit (DAT) (CMDINHD) in SDMMC_PSR.
See the “Abort Transaction” and “Suspend/Resume” sections in the “SD Host Controller Simplified Specification V3.00” for more details.

Value	Description
0	Transfer
1	Stop

Downloaded from Arrow.com.

47.10.14 SDMMC Block Gap Control Register (e.MMC)

Name: SDMMC_BGCR (e.MMC)
Offset: 0x2A
Reset: 0x00
Property: Read/Write

Note: This register configuration is specific to the e.MMC operation mode.

Bit	7	6	5	4	3	2	1	0
							CONTR	STPBGR
Access							R/W	R/W
Reset							0	0

Bit 1 – CONTR Continue Request
This bit is used to restart a transaction which was stopped using a Stop At Block Gap Request (STPBGR). To cancel stop at the block gap, set STPBGR to 0 and set this bit to 1 to restart the transfer.
The SDMMC automatically clears this bit in either of the following cases:
– In the case of a read transaction, the DAT Line Active (DLACT) changes from 0 to 1 as a read transaction restarts.
– In the case of a write transaction, the Write Transfer Active (WTACT) changes from 0 to 1 as the write transaction restarts.
Therefore, it is not necessary to set this bit to 0. If STPBGR is set to 1, any write to this bit is ignored.
See the “Abort Transaction” and “Suspend/Resume” sections in the “SD Host Controller Simplified Specification V3.00” for more details.

Value	Description
0	No effect.
1	Restart.

Bit 0 – STPBGR Stop At Block Gap Request
This bit is used to stop executing read and write transactions at the next block gap for nonDMA, SDMA, and ADMA transfers. The user must leave this bit set to 1 until Transfer Complete (TRFC) in SDMMC_NISTR. Clearing both Stop At Block Gap Request and Continue Request does not cause the transaction to restart. This bit can be set whether the card supports the Read Wait signal or not.
During read transfers, the SDMMC stops the transaction by using the Read Wait signal (SDMMC_DAT[2]) if supported, or by stopping the SD clock otherwise.
In case of write transfers in which the user writes data to SDMMC_BDPR, this bit must be set to 1 after all the block of data is written. If this bit is set to 1, the user does not write data to SDMMC_BDPR.
This bit affects Read Transfer Active (RTACT), Write Transfer Active (WTACT), DAT Line Active (DLACT) and Command Inhibit (DAT) (CMDINHD) in SDMMC_PSR.
See the “Abort Transaction” and “Suspend/Resume” sections in the “SD Host Controller Simplified Specification V3.00” for more details.

Value	Description
0	Transfer
1	Stop

Downloaded from Arrow.com.

47.10.15 SDMMC Wakeup Control Register (SD_SDIO)

Name: SDMMC_WCR (SD_SDIO)
Offset: 0x2B
Reset: 0x00
Property: Read/Write

Note: This register configuration is specific to the SD/SDIO operation mode.

Bit	7	6	5	4	3	2	1	0
								WKENCINT
Access								R/W
Reset								0

Bit 0 – WKENCINT Wakeup Event Enable on Card Interrupt
This bit enables a wakeup event via Card Interrupt (CINT) in SDMMC_NISTR. This bit can be set to 1 if FN_WUS (Wakeup Support) in the CIS (Card Information Structure) is set to 1 in the SDIO card.
0 (DISABLED): Wakeup Event disabled.
1 (ENABLED): Wakeup Event enabled.

Downloaded from Arrow.com.

47.10.16 SDMMC Clock Control Register

Name: SDMMC_CCR
Offset: 0x2C
Reset: 0x0000
Property: Read/Write

Bit	15	14	13	12	11	10	9	8
	SDCLKFSEL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	USDCLKFSEL[1:0]		CLKGSEL			SDCLKEN	INTCLKS	INTCLKEN
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 15:8 – SDCLKFSEL[7:0] SDCLK Frequency Select
This register is used to select the frequency of the SDCLK pin. There are two SDCLK Frequency modes according to Clock Generator Select (CLKGSEL).
The length of the clock divider (DIV) is extended to 10 bits (DIV[9:8] = USDCLKFSEL, DIV[7:0] = SDCLKFSEL)
– 10-bit Divided Clock Mode (CLKGSEL = 0): $f_{SDCLK} = f_{BASECLK}/(2 \times DIV)$. If DIV = 0 then $f_{SDCLK} = f_{BASECLK}$
– Programmable Clock Mode (CLKGSEL = 1): $f_{SDCLK} = f_{MULTCLK}/(DIV + 1)$
When HSEN is set in SDMMC_HC1R, or DDR is set in SDMMC_MC1R, the clock divider (DIV) must be non-zero.This field depends on the setting of Preset Value Enable (PVALEN) in SDMMC_HC2R.
If PVALEN = 0, this field is set by the user.
If PVALEN = 1, this field is automatically set to a value specified in one of the SDMMC_PVR.

Bits 7:6 – USDCLKFSEL[1:0] Upper Bits of SDCLK Frequency Select
These bits expand the SDCLK Frequency Select (SDCLKFSEL) to 10 bits. These two bits are assigned to bit 09-08 of the clock divider as described in SDCLKFSEL.

Bit 5 – CLKGSEL Clock Generator Select
This bit is used to select the clock generator mode in the SDCLK Frequency Select field. If the Programmable mode is not supported (SDMMC_CA1R.CLKMULT (Clock Multiplier) set to 0), then this bit cannot be written and is always read at 0.
This bit depends on the setting of Preset Value Enable (PVALEN) in SDMMC_HC2R.
If PVALEN = 0, this bit is set by the user.
If PVALEN = 1, this bit is automatically set to a value specified in one of the SDMMC_PVRx.

Value	Description
0	Divided Clock mode (BASECLK is used to generate SDCLK).
1	Programmable Clock mode (MULTCLK is used to generate SDCLK).

Bit 2 – SDCLKEN SD Clock Enable
The SDMMC stops the SD Clock when writing this bit to 0. SDCLK Frequency Select (SDCLKFSEL) can be changed when this bit is 0. Then, the SDMMC maintains the same clock frequency until SDCLK is stopped (Stop at SDCLK = 0). If Card Inserted (CARDINS) in SDMMC_PSR is cleared, this bit is also cleared.

Value	Description
0	SD Clock disabled
1	SD Clock enabled

Bit 1 – INTCLKS Internal Clock Stable
This bit is set to 1 when the SD clock is stable after setting SDMMC_CCR.INTCLKEN (Internal Clock Enable) to 1. The user must wait to set SD Clock Enable (SDCLKEN) until this bit is set to 1.

Value	Description
0	Internal clock not ready.

Downloaded from Arrow.com.

Value	Description
1	Internal clock ready.

Bit 0 – INTCLKEN Internal Clock Enable
This bit is set to 0 when the SDMMC is not used or is awaiting a wakeup interrupt. In this case, its internal clock is stopped to reach a very low power state. Registers are still able to be read and written. The clock starts to oscillate when this bit is set to 1. Once the clock oscillation is stable, the SDMMC sets Internal Clock Stable (INTCLKS) in this register to 1.

Value	Description
0	The internal clock stops.
1	The internal clock oscillates.

Downloaded from Arrow.com.

47.10.17 SDMMC Timeout Control Register

Name: SDMMC_TCR
Offset: 0x2E
Reset: 0x00
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
					DTCVAL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – DTCVAL[3:0] Data Timeout Counter Value
This value determines the interval at which DAT line timeouts are detected. For more information about timeout generation, see Data Timeout Error (DATTEO) in SDMMC_EISTR. When setting this register, the user can prevent inadvertent timeout events by clearing the Data Timeout Error Status Enable (in SDMMC_EISTER).

$$\mathrm{TIMEOUT}_{(\mu s)} = \frac{2^{13 + \mathrm{DTCVAL}}}{f_{\mathrm{FTEOCLK(MHz)}}}$$

Note: DTCVAL = $f_{(Hexa)}$ is reserved.

Downloaded from Arrow.com.

47.10.18 SDMMC Software Reset Register

Name: SDMMC_SRR
Offset: 0x2F
Reset: 0x00
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
						SWRSTDAT	SWRSTCMD	SWRSTALL
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SWRSTDAT Software Reset for DAT Line
Only part of a data circuit is reset. The DMA circuit is also reset.
The following registers and bits are cleared by this bit:
"SDMMC Buffer Data Port Register"
– Buffer is cleared and initialized.
"SDMMC Present State Register"
– Buffer Read Enable (BUFRDEN)
– Buffer Write Enable (BUFWREN)
– Read Transfer Active (RTACT)
– Write Transfer Active (WTACT)
– DAT Line Active (DATLL)
– Command Inhibit (DAT) (CMDINHD)
"SDMMC Block Gap Control Register (SD_SDIO)"
– Continue Request (CONTR)
– Stop At Block Gap Request (STPBGR)
"SDMMC Normal Interrupt Status Register (SD_SDIO)"
– Buffer Read Ready (BRDRDY)
– Buffer Write Ready (BWRRDY)
– DMA Interrupt (DMAINT)
– Block Gap Event (BLKGE)
– Transfer Complete (TRFC)

Value	Description
0	Work
1	Reset

Bit 1 – SWRSTCMD Software Reset for CMD Line
Only part of a command circuit is reset.
The following registers and bits are cleared by this bit:
"SDMMC Present State Register"
– Command Inhibit (CMD) (CMDINHC)
"SDMMC Normal Interrupt Status Register (SD_SDIO)" and "SDMMC Normal Interrupt Status Register (e.MMC)"
– Command Complete (CMDC)

Value	Description
0	Work
1	Reset

Bit 0 – SWRSTALL Software Reset for All
This reset affects the entire SDMMC. During initialization, the SDMMC must be reset by setting this bit to '1'. This bit is automatically cleared to '0' when SDMMC_CA0R and SDMMC_CA1R are valid and the user can read them. If this bit is set to '1', the user should issue a reset command and reinitialize the card.
List of registers cleared to '0':
– "SDMMC SDMA System Address / Argument 2 Register"
– "SDMMC Block Size Register"
– "SDMMC Block Count Register"
– "SDMMC Argument 1 Register"

Downloaded from Arrow.com.

– “SDMMC Command Register”
– “SDMMC Transfer Mode Register”
– “SDMMC Response Register”
– “SDMMC Buffer Data Port Register”
– “SDMMC Present State Register” (except CMDLL, DATLL, WRPPL, CARDDDPL, CARDSS, CARDINS)
– “SDMMC Host Control 1 Register (SD_SDIO)”
– “SDMMC Host Control 1 Register (e.MMC)”
– “SDMMC Power Control Register”
– “SDMMC Block Gap Control Register (SD_SDIO)”
– “SDMMC Block Gap Control Register (e.MMC)”
– “SDMMC Wakeup Control Register (SD_SDIO)”
– “SDMMC Clock Control Register”
– “SDMMC Timeout Control Register”
– “SDMMC Normal Interrupt Status Register (SD_SDIO)”
– “SDMMC Error Interrupt Status Register (SD_SDIO)”
– “SDMMC Normal Interrupt Status Enable Register (SD_SDIO)”
– “SDMMC Error Interrupt Status Enable Register (SD_SDIO)”
– “SDMMC Normal Interrupt Signal Enable Register (SD_SDIO)”
– “SDMMC Error Interrupt Signal Enable Register (SD_SDIO)”
– “SDMMC Auto CMD Error Status Register”
– “SDMMC Host Control 2 Register (SD_SDIO)”
– “SDMMC ADMA Error Status Register”
– “SDMMC ADMA System Address Register”
– “SDMMC Slot Interrupt Status Register”
– “SDMMC e.MMC Control 1 Register”
– “SDMMC e.MMC Control 2 Register”
– “SDMMC AHB Control Register”
– “SDMMC Clock Control 2 Register”
– “SDMMC Capabilities Control Register” (except KEY)

Value	Description
0	Work
1	Reset

Downloaded from Arrow.com.

47.10.19 SDMMC Normal Interrupt Status Register (SD_SDIO)

Name: SDMMC_NISTR (SD_SDIO)
Offset: 0x30
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the SD/SDIO operation mode.

Bit	15	14	13	12	11	10	9	8
	ERRINT							CINT
Access	R							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 15 – ERRINT Error Interrupt
If any of the bits in SDMMC_EISTR are set, then this bit is set. Therefore, the user can efficiently test for an error by checking this bit first. This bit is read-only.

Value	Description
0	No error.
1	Error.

Bit 8 – CINT Card Interrupt
Writing this bit to '1' does not clear this bit. It is cleared by resetting the SD card interrupt factor. In 1-bit mode, the SDMMC detects the Card Interrupt without SDCLK to support wakeup. In 4-bit mode, the Card Interrupt signal is sampled during the interrupt cycle, so there are some sample delays between the interrupt signal from the SD card and the interrupt to the system.
When this bit is set to '1' and the user needs to start this interrupt service, Card Interrupt Status Enable (CINT) in SDMMC_NISTER may be set to '0' in order to clear the card interrupt statuses latched in the SDMMC and to stop driving the interrupt signal to the system. After completion of the card interrupt service (it should reset interrupt factors in the SD card and the interrupt signal may not be asserted), set SDMMC_NISTER.CINT to '1' and start sampling the interrupt signal again.
Interrupt detected by DAT[1] is supported when there is one card per slot.
This bit can only be set to 1 if SDMMC_NISTER.CINT is set to 1. An interrupt can only be generated if SDMMC_NISIER.CINT is set to 1.

Value	Description
0	No card interrupt.
1	Card interrupt.

Bit 5 – BRDRDY Buffer Read Ready
This status is set to '1' if the Buffer Read Enable (BUFRDEN) changes from '0' to '1'. See BUFRDEN in SDMMC_PSR.
This bit can only be set to '1' if SDMMC_NISTER.BRDRDY is set to '1'. An interrupt can only be generated if SDMMC_NISIER.BRDRDY is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	Not ready to read buffer.
1	Ready to read buffer.

Bit 4 – BWRRDY Buffer Write Ready
This status is set to '1' if the Buffer Write Enable (BUFWREN) changes from '0' to '1'. See BUFWREN in SDMMC_PSR.

Downloaded from Arrow.com.

This bit can only be set to '1' if SDMMC_NISTER.BWRRDY is set to '1'. An interrupt can only be generated if SDMMC_NISIER.BWRRDY is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	Not ready to write buffer.
1	Ready to write buffer.

Bit 3 – DMAINT DMA Interrupt
This status is set if the SDMMC detects the Host SDMA Buffer boundary during transfer. See SDMA Buffer Boundary (BOUNDARY) in SDMMC_BSR.
In case of ADMA, by setting the “int” field in the descriptor table, the SDMMC raises this status flag when the descriptor line is completed. This status flag does not rise after Transfer Complete (TRFC).
This bit can only be set to '1' if SDMMC_NISTER.DMAINT is set to '1'. An interrupt can only be generated if SDMMC_NISIER.DMAINT is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No DMA Interrupt.
1	DMA Interrupt.

Bit 2 – BLKGE Block Gap Event
If the Stop At Block Gap Request (STPBGR) in SDMMC_BGCR is set to 1, this bit is set when either a read or a write transaction is stopped at a block gap. If STPBGR is not set to 1, this bit is not set to 1.
In the case of a Read transaction:
This bit is set at the falling edge of the DAT Line Active (DLACT) status (when the transaction is stopped at SD bus timing). The Read Wait must be supported in order to use this function. See section “Read Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” about the detailed timing.
In the case of a Write transaction:
This bit is set at the falling edge of the Write Transfer Active (WTACT) status (after getting the CRC status at SD bus timing). See section “Write Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.
This bit can only be set to '1' if SDMMC_NISTER.BLKGE is set to '1'. An interrupt can only be generated if SDMMC_NISIER.BLKGE is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No block gap event.
1	Transaction stopped at block gap.

Bit 1 – TRFC Transfer Complete
This bit is set when a read/write transfer and a command with Busy is completed.
In the case of a Read Transaction:
This bit is set at the falling edge of the Read Transfer Active Status. The interrupt is generated in two cases. The first is when a data transfer is completed as specified by the data length (after the last data was read to the system). The second is when data has stopped at the block gap and completed the data transfer by setting the Stop At Block Gap Request (STPBGR) in SDMMC_BGCR (after valid data was read to the system). See section “Read Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.
In the case of a Write Transaction:
This bit is set at the falling edge of the DAT Line Active (DLACT) status. This interrupt is generated in two cases. The first is when the last data is written to the card as specified by the data length and the Busy signal is released. The second is when data transfers are stopped at the block gap by setting Stop At Block Gap Request (STPBGR) in SDMMC_BGCR and data transfers are completed. (After valid data is written to the card and the Busy signal is released). See section “Write Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.
In the case of command with Busy:
This bit is set when Busy is deasserted. See DAT Line Active (DLACT) and Command Inhibit (DAT) (CMDINHD) in SDMMC_PSR.
This bit can only be set to '1' if SDMMC_NISTER.TRFC is set to '1'. An interrupt can only be generated if SDMMC_NISIER.TRFC is set to '1'.

Downloaded from Arrow.com.

Writing this bit to '1' clears this bit.
The table below shows that Transfer Complete (TRFC) has a higher priority than Data Timeout Error (DATTEO). If both bits are set to '1', execution of a command can be considered to be completed.

TRFC	DATTEO	Meaning of the status
0	0	Interrupted by another factor
0	1	Timeout occurred during transfer
1	Don’t Care	Command execution complete

Value	Description
0	Command execution is not complete.
1	Command execution is complete.

Bit 0 – CMDC Command Complete
This bit is set when getting the end bit of the command response. Auto CMD12 and Auto CMD23 consist of two responses. Command Complete is not generated by the response of CMD12 or CMD23, but it is generated by the response of a read/write command. See Command Inhibit (CMD) in SDMMC_PSR for details on how to control this bit.
This bit can only be set to 1 if SDMMC_NISTER.CMDC is set to 1. An interrupt can only be generated if SDMMC_NISIER.CMDC is set to 1.
Writing this bit to 1 clears this bit.
The table below shows that Command Timeout Error (CMDTEO) has a higher priority than Command Complete (CMDC). If both bits are set to 1, it can be considered that the response was not received correctly.

CMDC	CMDTEO	Meaning of the status
0	0	Interrupted by another factor
Don’t care	1	Response not received within 64 SDCLK cycles
1	0	Response received

Value	Description
0	No command complete.
1	Command complete.

Downloaded from Arrow.com.

47.10.20 SDMMC Normal Interrupt Status Register (e.MMC)

Name: SDMMC_NISTR (e.MMC)
Offset: 0x30
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the e.MMC operation mode.

Bit	15	14	13	12	11	10	9	8
	ERRINT	BOOTAR						
Access	R	R/W						
Reset	0	0						

Bit	7	6	5	4	3	2	1	0
			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 15 – ERRINT Error Interrupt
If any of the bits in SDMMC_EISTR are set, then this bit is set. Therefore, the user can efficiently test for an error by checking this bit first. This bit is read-only.

Value	Description
0	No error.
1	Error.

Bit 14 – BOOTAR Boot Acknowledge Received
This bit is set to '1' when the SDMMC received a Boot Acknowledge pattern from the e.MMC.
This bit can only be set to '1' if SDMMC_NISTER.BOOTAR is set to '1'. An interrupt can only be generated if SDMMC_NISIER.BOOTAR is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	Boot Acknowledge pattern not received.
1	Boot Acknowledge pattern received.

Bit 5 – BRDRDY Buffer Read Ready
This status is set to '1' if Buffer Read Enable (BUFRDEN) changes from '0' to '1'. See Buffer Read Enable (BUFRDEN) in SDMMC_PSR.
This bit can only be set to '1' if SDMMC_NISTER.BRDRDY is set to '1'. An interrupt can only be generated if SDMMC_NISIER.BRDRDY is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	Not ready to read buffer.
1	Ready to read buffer.

Bit 4 – BWRRDY Buffer Write Ready
This status is set to 1 if Buffer Write Enable (BUFWREN) changes from '0' to '1'. See Buffer Write Enable (BUFWREN) in SDMMC_PSR.
This bit can only be set to '1' if SDMMC_NISTER.BWRRDY is set to '1'. An interrupt can only be generated if SDMMC_NISIER.BWRRDY is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	Not ready to write buffer.
1	Ready to write buffer.

Downloaded from Arrow.com.

Bit 3 – DMAINT DMA Interrupt
This status is set if the SDMMC detects the Host SDMA Buffer boundary during transfer. See SDMA Buffer Boundary (BOUNDARY) in SDMMC_BSR.
In case of ADMA, by setting “int” field in the descriptor table, the SDMMC raises this status flag when the descriptor line is completed. This status flag does not rise after the Transfer Complete (TRFC).
This bit can only be set to '1' if SDMMC_NISTER.DMAINT is set to '1'. An interrupt can only be generated if SDMMC_NISIER.DMAINT is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No DMA interrupt.
1	DMA interrupt.

Bit 2 – BLKGE Block Gap Event
If the Stop At Block Gap Request (STPBGR) in SDMMC_BGCR is set to '1', this bit is set when either a read or a write transaction is stopped at a block gap. If STPBGR is not set to '1', this bit is not set to '1'.
In the case of a Read transaction:
This bit is set at the falling edge of the DAT Line Active (DLACT) status (when the transaction is stopped at SD bus timing). The Read Wait must be supported in order to use this function. See section “Read Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” about the detailed timing.
In the case of a Write transaction:
This bit is set at the falling edge of the Write Transfer Active (WTACT) status (after getting the CRC status at SD bus timing). See section “Write Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.
This bit can only be set to '1' if SDMMC_NISTER.BLKGE is set to '1'. An interrupt can only be generated if SDMMC_NISIER.BLKGE is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No block gap event.
1	Transaction stopped at block gap.

Bit 1 – TRFC Transfer Complete
This bit is set when a read/write transfer and a command with Busy is completed.
In the case of a Read Transaction:
This bit is set at the falling edge of the Read Transfer Active Status. The interrupt is generated in two cases. The first is when a data transfer is completed as specified by the data length (after the last data was read to the system). The second is when data has stopped at the block gap and completed the data transfer by setting the Stop At Block Gap Request (STPBGR) in SDMMC_BGCR (after valid data was read to the system). See section “Read Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.
In the case of a Write Transaction:
This bit is set at the falling edge of the DAT Line Active (DLACT) status. This interrupt is generated in two cases. The first is when the last data is written to the card as specified by the data length and the Busy signal is released. The second is when data transfers are stopped at the block gap by setting Stop At Block Gap Request (STPBGR) in SDMMC_BGCR and data transfers are completed. (After valid data is written to the card and the Busy signal is released). See section “Write Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.
In the case of command with Busy:
This bit is set when Busy is deasserted. See DAT Line Active (DLACT) and Command Inhibit (DAT) (CMDINHD) in SDMMC_PSR.
This bit can only be set to '1' if SDMMC_NISTER.TRFC is set to '1'. An interrupt can only be generated if SDMMC_NISIER.TRFC is set to '1'.
Writing this bit to '1' clears this bit.
The table below shows that Transfer Complete (TRFC) has a higher priority than Data Timeout Error (DATTEO). If both bits are set to '1', execution of a command can be considered to be completed.

TRFC	DATTEO	Meaning of the status
0	0	Interrupted by another factor

Downloaded from Arrow.com.

.....continued

TRFC	DATTEO	Meaning of the status
0	1	Timeout occurred during transfer
1	Don’t Care	Command execution complete

Value	Description
0	Command execution is not complete.
1	Command execution is complete.

Bit 0 – CMDC Command Complete
This bit is set when getting the end bit of the command response. Auto CMD12 and Auto CMD23 consist of two responses. Command Complete is not generated by the response of CMD12 or CMD23, but it is generated by the response of a read/write command. See CMRINHC in SDMMC_PSR for details on how to control this bit.
This bit can only be set to '1' if SDMMC_NISTER.CMDC is set to '1'. An interrupt can only be generated if SDMMC_NISIER.CMDC is set to '1'.
Writing this bit to '1' clears this bit.
The table below shows that Command Timeout Error (CMDTEO) has a higher priority than Command Complete (CMDC). If both bits are set to '1', it can be considered that the response was not received correctly.

CMDC	CMDTEO	Meaning of the status
0	0	Interrupted by another factor
Don’t care	1	Response not received within 64 SDCLK cycles
1	0	Response received

Value	Description
0	No command complete.
1	Command complete.

Downloaded from Arrow.com.

47.10.21 SDMMC Error Interrupt Status Register (SD_SDIO)

Name: SDMMC_EISTR (SD_SDIO)
Offset: 0x32
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the SD/SDIO operation mode.

Bit	15	14	13	12	11	10	9	8
							ADMA	ACMD
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
Access	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 9 – ADMA ADMA Error
This bit is set to '1' when the SDMMC detects errors during an ADMA-based data transfer. The state of the ADMA at an error occurrence is saved in SDMMC_AESR.
In addition, the SDMMC raises this status flag when it detects some invalid description data (Valid = 0) at the ST_FDS state (see section “Advanced DMA” in the “SD Host Controller Simplified Specification V3.00”). ADMA Error Status (ERRST) in SDMMC_AESR indicates that an error occurred in ST_FDS state. The user may find that the Valid bit is not set at the error descriptor.
This bit can only be set to '1' if SDMMC_EISTER.ADMA is set to '1'. An interrupt can only be generated if SDMMC_EISIER.ADMA is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No error.
1	Error.

Bit 8 – ACMD Auto CMD Error
Auto CMD12 and Auto CMD23 use this error status. This bit is set to '1' when detecting that one of the 0 to 4 bits in SDMMC_ACESR[4:0] has changed from '0' to '1'. In the case of Auto CMD12, this bit is set to '1', not only when errors occur in Auto CMD12 but also when auto CMD12 is not executed due to the previous command error.
This bit can only be set to '1' if SDMMC_EISTER.ACMD is set to '1'. An interrupt can only be generated if SDMMC_EISIER.ACMD is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No error.
1	Error.

Bit 7 – CURLIM Current Limit Error
By setting SD Bus Power (SDBPWR) in SDMMC_PCR, the SDMMC is requested to supply power for the SD Bus. The SDMMC is protected from an illegal card by stopping power supply to the card, in which case this bit indicates a failure status. Reading 1 means the SDMMC is not supplying power to the card due to some failure. Reading 0 means that the SDMMC is supplying power and no error has occurred. The SDMMC may require some sampling time to detect the current limit.
This bit can only be set to '1' if SDMMC_EISTER.CURLIM is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CURLIM is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No error.
1	Error.

Downloaded from Arrow.com.

Bit 6 – DATEND Data End Bit Error
This bit is set to '1' either when detecting 0 at the end bit position of read data which uses the DAT line or at the end bit position of the CRC Status.
This bit can only be set to '1' if SDMMC_EISTER.DATEND is set to '1'. An interrupt can only be generated if SDMMC_EISIER.DATEND is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No error.
1	Error.

Bit 5 – DATCRC Data CRC error
This bit is set to '1' when detecting a CRC error when transferring read data which uses the DAT line or when detecting that the Write CRC Status has a value other than '010'.
This bit can only be set to '1' if SDMMC_EISTER.DATCRC is set to '1'. An interrupt can only be generated if SDMMC_EISIER.DATCRC is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No error.
1	Error.

Bit 4 – DATTEO Data Timeout Error
This bit is set to '1' when detecting one of following timeout conditions.
– Busy timeout for R1b, R5b response type (see “Physical Layer Simplified Specification V3.01” and “SD Host Controller Simplified Specification V3.00”).
– Busy timeout after Write CRC status.
– Write CRC Status timeout.
– Read data timeout
This bit can only be set to '1' if SDMMC_EISTER.DATTEO is set to '1'. An interrupt can only be generated if SDMMC_EISIER.DATTEO is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No error.
1	Error.

Bit 3 – CMDIDX Command Index Error
This bit is set to '1' if a Command Index error occurs in the command response.
This bit can only be set to '1' if SDMMC_EISTER.CMDIDX is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CMDIDX is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No error.
1	Error.

Bit 2 – CMDEND Command End Bit Error
This bit is set to '1' when detecting that the end bit of a command response is 0.
This bit can only be set to '1' if SDMMC_EISTER.CMDEND is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CMDEND is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No error.
1	Error.

Bit 1 – CMDCRC Command CRC Error
The Command CRC Error is generated in two cases.
If a response is returned and the Command Timeout Error (CMDTEO) is set to 0 (indicating no command timeout), this bit is set to '1' when detecting a CRC error in the command response.
The SDMMC detects a CMD line conflict by monitoring the CMD line when a command is issued. If the SDMMC drives the CMD line to 1 level, but detects 0 level on the CMD line at the next SDCLK edge, then the SDMMC aborts

Downloaded from Arrow.com.

the command (stops driving the CMD line) and sets this bit to '1'. CMDTEO is also set to '1' to indicate a CMD line conflict (see the table above).
This bit can only be set to '1' if SDMMC_EISTER.CMDCRC is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CMDCRC is set to '1'.
Writing this bit to '1' clears this bit.

Bit 0 – CMDTEO Command Timeout Error
This bit is set to '1' only if no response is returned within 64 SDCLK cycles from the end bit of the command. If the SDMMC detects a CMD line conflict, in which case Command CRC Error (CMDCRC) is also set to '1' as shown in the table below, this bit is set without waiting for 64 SDCLK cycles because the command is aborted by the SDMMC.
This bit can only be set to '1' if SDMMC_EISTER.CMDTEO is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CMDTEO is set to '1'.
Writing this bit to '1' clears this bit.

CMDCRC	CMDTEO	Types of error
0	0	No error
0	1	Response timeout error
1	0	Response CRC error
1	1	CMD line conflict

Downloaded from Arrow.com.

47.10.22 SDMMC Error Interrupt Status Register (e.MMC)

Name: SDMMC_EISTR (e.MMC)
Offset: 0x32
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the e.MMC operation mode.

Bit	15	14	13	12	11	10	9	8
				BOOTAE			ADMA	ACMD
Access				R/W			R/W	R/W
Reset				0			0	0

Bit	7	6	5	4	3	2	1	0
	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 12 – BOOTAE Boot Acknowledge Error
This bit is set to '1' when detecting that the e.MMC Boot Acknowledge Status has a value other than '010'.
This bit can only be set to '1' if SDMMC_EISTER.BOOTAE is set to '1'. An interrupt can only be generated if SDMMC_EISIER.BOOTAE is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No error.
1	Error.

Bit 9 – ADMA ADMA Error
This bit is set to 1 when the SDMMC detects errors during an ADMA-based data transfer. The state of the ADMA at an error occurrence is saved in SDMMC_AESR.
In addition, the SDMMC raises this status flag when it detects some invalid description data (Valid = 0) at the ST_FDS state (see section “Advanced DMA” in the “SD Host Controller Simplified Specification V3.00”). ADMA Error Status (ERRST) in SDMMC_AESR indicates that an error occurred in ST_FDS state. The user may find that the Valid bit is not set at the error descriptor.
This bit can only be set to '1' if SDMMC_EISTER.ADMA is set to '1'. An interrupt can only be generated if SDMMC_EISIER.ADMA is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No error.
1	Error.

Bit 8 – ACMD Auto CMD Error
Auto CMD12 and Auto CMD23 use this error status. This bit is set to '1' when detecting that one of the 0 to 4 bits in SDMMC_ACESR[4:0] has changed from 0 to 1. In the case of Auto CMD12, this bit is set to '1', not only when errors occur in Auto CMD12, but also when Auto CMD12 is not executed due to the previous command error.
This bit can only be set to '1' if SDMMC_EISTER.ACMD is set to '1'. An interrupt can only be generated if SDMMC_EISIER.ACMD is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No error.
1	Error.

Bit 7 – CURLIM Current Limit Error
By setting SD Bus Power (SDBPWR) in SDMMC_PSR, the SDMMC is requested to supply power for the SD Bus. The SDMMC is protected from an illegal card by stopping power supply to the card, in which case this bit indicates a failure status. Reading 1 means the SDMMC is not supplying power to the card due to some failure. Reading 0

Downloaded from Arrow.com.

means that the SDMMC is supplying power and no error has occurred. The SDMMC may require some sampling time to detect the current limit.
This bit can only be set to '1' if SDMMC_EISTER.CURLIM is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CURLIM is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No error.
1	Error.

Bit 6 – DATEND Data End Bit Error
This bit is set to '1' either when detecting 0 at the end bit position of read data which uses the DAT line or at the end bit position of the CRC Status.
This bit can only be set to '1' if SDMMC_EISTER.DATEND is set to '1'. An interrupt can only be generated if SDMMC_EISIER.DATEND is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No error.
1	Error.

Bit 5 – DATCRC Data CRC Error
This bit is set to '1' when detecting a CRC error during a transfer of read data which uses the DAT line or when detecting that the Write CRC Status has a value other than '010'.
This bit can only be set to '1' if SDMMC_EISTER.DATCRC is set to '1'. An interrupt can only be generated if SDMMC_EISIER. DATCRC is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No error.
1	Error.

Bit 4 – DATTEO Data Timeout error
This bit is set to '1' when detecting one of following timeout conditions.
– Busy timeout for R1b, R5b response type (see “Physical Layer Simplified Specification V3.01” and “SDIO Simplified Specification V3.00”).
– Busy timeout after Write CRC Status.
– Write CRC Status timeout.
– Read data timeout
This bit can only be set to '1' if SDMMC_EISTER.DATTEO is set to '1'. An interrupt can only be generated if SDMMC_EISIER.DATTEO is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No error.
1	Error.

Bit 3 – CMDIDX Command Index Error
This bit is set to '1' if a Command Index error occurs in the command response.
This bit can only be set to '1' if SDMMC_EISTER.CMDIDX is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CMDIDX is set to '1'.
Writing this bit to '1' clears this bit.

Value	Description
0	No error.
1	Error.

Bit 2 – CMDEND Command End Bit Error
This bit is set to '1' when detecting that the end bit of a command response is 0.
This bit can only be set to '1' if SDMMC_EISTER.CMDEND is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CMDEND is set to '1'.
Writing this bit to '1' clears this bit.

Downloaded from Arrow.com.

Value	Description
0	No error.
1	Error.

Bit 1 – CMDCRC Command CRC Error
The Command CRC Error is generated in two cases.
If a response is returned and Command Timeout Error (CMDTEO) is set to 0 (indicating no command timeout), this bit is set to '1' when detecting a CRC error in the command response.
The SDMMC detects a CMD line conflict by monitoring the CMD line when a command is issued. If the SDMMC drives the CMD line to 1 level, but detects 0 level on the CMD line at the next SDCLK edge, then the SDMMC aborts the command (stops driving the CMD line) and sets this bit to '1'. CMDTEO is also set to '1' to indicate a CMD line conflict (see he table “Relations between CMDCRC and CMDTEO”).
This bit can only be set to '1' if SDMMC_EISTER.CMDCRC is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CMDCRC is set to '1'.
Writing this bit to 1 clears this bit.

Bit 0 – CMDTEO Command Timeout Error
This bit is set to '1' only if no response is returned within 64 SDCLK cycles from the end bit of the command. If the SDMMC detects a CMD line conflict, in which case Command CRC Error (CMDCRC) is also set to '1' as shown in the table “Relations between CMDCRC and CMDTEO”, this bit is set without waiting for 64 SDCLK cycles because the command is aborted by the SDMMC.
This bit can only be set to '1' if SDMMC_EISTER.CMDTEO is set to '1'. An interrupt can only be generated if SDMMC_EISIER.CMDTEO is set to '1'.
Writing this bit to '1' clears this bit.

CMDCRC	CMDTEO	Types of error
0	0	No error
0	1	Response timeout error
1	0	Response CRC error
1	1	CMD line conflict

Downloaded from Arrow.com.

47.10.23 SDMMC Normal Interrupt Status Enable Register (SD_SDIO)

Name: SDMMC_NISTER (SD_SDIO)
Offset: 0x34
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the SD/SDIO operation mode.

Bit	15	14	13	12	11	10	9	8
								CINT
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 8 – CINT Card Interrupt Status Enable
If this bit is set to 0, the SDMMC clears interrupt requests to the system. The Card Interrupt detection is stopped when this bit is cleared and restarted when this bit is set to 1. The user may clear this bit before servicing the Card Interrupt and may set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts.
0 (MASKED): The CINT status flag in SDMMC_NISTR is masked.
1 (ENABLED): The CINT status flag in SDMMC_NISTR is enabled.

Bit 5 – BRDRDY Buffer Read Ready Status Enable
0 (MASKED): The BRDRDY status flag in SDMMC_NISTR is masked.
1 (ENABLED): The BRDRDY status flag in SDMMC_NISTR is enabled.

Bit 4 – BWRRDY Buffer Write Ready Status Enable
0 (MASKED): The BWRRDY status flag in SDMMC_NISTR is masked.
1 (ENABLED): The BWRRDY status flag in SDMMC_NISTR is enabled.

Bit 3 – DMAINT DMA Interrupt Status Enable
0 (MASKED): The DMAINT status flag in SDMMC_NISTR is masked.
1 (ENABLED): The DMAINT status flag in SDMMC_NISTR is enabled.

Bit 2 – BLKGE Block Gap Event Status Enable
0 (MASKED): The BLKGE status flag in SDMMC_NISTR is masked.
1 (ENABLED): The BLKGE status flag in SDMMC_NISTR is enabled.

Bit 1 – TRFC Transfer Complete Status Enable
0 (MASKED): The TRFC status flag in SDMMC_NISTR is masked.
1 (ENABLED): The TRFC status flag in SDMMC_NISTR is enabled.

Bit 0 – CMDC Command Complete Status Enable
0 (MASKED): The CMDC status flag in SDMMC_NISTR is masked.
1 (ENABLED): The CMDC status flag in SDMMC_NISTR is enabled.

Downloaded from Arrow.com.

47.10.24 SDMMC Normal Interrupt Status Enable Register (e.MMC)

Name: SDMMC_NISTER (e.MMC)
Offset: 0x34
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the e.MMC operation mode.

Bit	15	14	13	12	11	10	9	8
		BOOTAR						
Access		R/W						
Reset		0						

Bit	7	6	5	4	3	2	1	0
			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 14 – BOOTAR Boot Acknowledge Received Status Enable
0 (MASKED): The BOOTAR status flag in SDMMC_NISTR is masked.
1 (ENABLED): The BOOTAR status flag in SDMMC_NISTR is enabled.

Bit 5 – BRDRDY Buffer Read Ready Status Enable
0 (MASKED): The BRDRDY status flag in SDMMC_NISTR is masked.
1 (ENABLED): The BRDRDY status flag in SDMMC_NISTR is enabled.

Bit 4 – BWRRDY Buffer Write Ready Status Enable
0 (MASKED): The BWRRDY status flag in SDMMC_NISTR is masked.
1 (ENABLED): The BWRRDY status flag in SDMMC_NISTR is enabled.

Bit 3 – DMAINT DMA Interrupt Status Enable
0 (MASKED): The DMAINT status flag in SDMMC_NISTR is masked.
1 (ENABLED): The DMAINT status flag in SDMMC_NISTR is enabled.

Bit 2 – BLKGE Block Gap Event Status Enable
0 (MASKED): The BLKGE status flag in SDMMC_NISTR is masked.
1 (ENABLED): The BLKGE status flag in SDMMC_NISTR is enabled.

Bit 1 – TRFC Transfer Complete Status Enable
0 (MASKED): The TRFC status flag in SDMMC_NISTR is masked.
1 (ENABLED): The TRFC status flag in SDMMC_NISTR is enabled.

Bit 0 – CMDC Command Complete Status Enable
0 (MASKED): The CMDC status flag in SDMMC_NISTR is masked.
1 (ENABLED): The CMDC status flag in SDMMC_NISTR is enabled.

Downloaded from Arrow.com.

47.10.25 SDMMC Error Interrupt Status Enable Register (SD_SDIO)

Name: SDMMC_EISTER (SD_SDIO)
Offset: 0x36
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the SD/SDIO operation mode.

Bit	15	14	13	12	11	10	9	8
							ADMA	ACMD
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 9 – ADMA ADMA Error Status Enable
0 (MASKED): The ADMA status flag in SDMMC_EISTR is masked.
1 (ENABLED): The ADMA status flag in SDMMC_EISTR is enabled.

Bit 8 – ACMD Auto CMD Error Status Enable
0 (MASKED): The ACMD status flag in SDMMC_EISTR is masked.
1 (ENABLED): The ACMD status flag in SDMMC_EISTR is enabled.

Bit 7 – CURLIM Current Limit Error Status Enable
0 (MASKED): The CURLIM status flag in SDMMC_EISTR is masked.
1 (ENABLED): The CURLIM status flag in SDMMC_EISTR is enabled.

Bit 6 – DATEND Data End Bit Error Status Enable
0 (MASKED): The DATEND status flag in SDMMC_EISTR is masked.
1 (ENABLED): The DATEND status flag in SDMMC_EISTR is enabled.

Bit 5 – DATCRC Data CRC Error Status Enable
0 (MASKED): The DATCRC status flag in SDMMC_EISTR is masked.
1 (ENABLED): The DATCRC status flag in SDMMC_EISTR is enabled.

Bit 4 – DATTEO Data Timeout Error Status Enable
0 (MASKED): The DATTEO status flag in SDMMC_EISTR is masked.
1 (ENABLED): The DATTEO status flag in SDMMC_EISTR is enabled.

Bit 3 – CMDIDX Command Index Error Status Enable
0 (MASKED): The CMDIDX status flag in SDMMC_EISTR is masked.
1 (ENABLED): The CMDIDX status flag in SDMMC_EISTR is enabled.

Bit 2 – CMDEND Command End Bit Error Status Enable
0 (MASKED): The CMDEND status flag in SDMMC_EISTR is masked.
1 (ENABLED): The CMDEND status flag in SDMMC_EISTR is enabled.

Bit 1 – CMDCRC Command CRC Error Status Enable
0 (MASKED): The CMDCRC status flag in SDMMC_EISTR is masked.
1 (ENABLED): The CMDCRC status flag in SDMMC_EISTR is enabled.

Bit 0 – CMDTEO Command Timeout Error Status Enable
0 (MASKED): The CMDTEO status flag in SDMMC_EISTR is masked.

Downloaded from Arrow.com.

1 (ENABLED): The CMDTEO status flag in SDMMC_EISTR is enabled.

Downloaded from Arrow.com.

47.10.26 SDMMC Error Interrupt Status Enable Register (e.MMC)

Name: SDMMC_EISTER (e.MMC)
Offset: 0x36
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the e.MMC operation mode.

Bit	15	14	13	12	11	10	9	8
				BOOTAE			ADMA	ACMD
Access				R/W			R/W	R/W
Reset				0			0	0

Bit	7	6	5	4	3	2	1	0
	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 12 – BOOTAE Boot Acknowledge Error Status Enable
0 (MASKED): The BOOTAE status flag in SDMMC_EISTR is masked.
1 (ENABLED): The BOOTAE status flag in SDMMC_EISTR is enabled.

Bit 9 – ADMA ADMA Error Status Enable
0 (MASKED): The ADMA status flag in SDMMC_EISTR is masked.
1 (ENABLED): The ADMA status flag in SDMMC_EISTR is enabled.

Bit 8 – ACMD Auto CMD Error Status Enable
0 (MASKED): The ACMD status flag in SDMMC_EISTR is masked.
1 (ENABLED): The ACMD status flag in SDMMC_EISTR is enabled.

Bit 7 – CURLIM Current Limit Error Status Enable
0 (MASKED): The CURLIM status flag in SDMMC_EISTR is masked.
1 (ENABLED): The CURLIM status flag in SDMMC_EISTR is enabled.

Bit 6 – DATEND Data End Bit Error Status Enable
0 (MASKED): The DATEND status flag in SDMMC_EISTR is masked.
1 (ENABLED): The DATEND status flag in SDMMC_EISTR is enabled.

Bit 5 – DATCRC Data CRC Error Status Enable
0 (MASKED): The DATCRC status flag in SDMMC_EISTR is masked.
1 (ENABLED): The DATCRC status flag in SDMMC_EISTR is enabled.

Bit 4 – DATTEO Data Timeout Error Status Enable
0 (MASKED): The DATTEO status flag in SDMMC_EISTR is masked.
1 (ENABLED): The DATTEO status flag in SDMMC_EISTR is enabled.

Bit 3 – CMDIDX Command Index Error Status Enable
0 (MASKED): The CMDIDX status flag in SDMMC_EISTR is masked.
1 (ENABLED): The CMDIDX status flag in SDMMC_EISTR is enabled.

Bit 2 – CMDEND Command End Bit Error Status Enable
0 (MASKED): The CMDEND status flag in SDMMC_EISTR is masked.
1 (ENABLED): The CMDEND status flag in SDMMC_EISTR is enabled.

Bit 1 – CMDCRC Command CRC Error Status Enable
0 (MASKED): The CMDCRC status flag in SDMMC_EISTR is masked.

Downloaded from Arrow.com.

1 (ENABLED): The CMDCRC status flag in SDMMC_EISTR is enabled.

Bit 0 – CMDTEO Command Timeout Error Status Enable
0 (MASKED): The CMDTEO status flag in SDMMC_EISTR is masked.
1 (ENABLED): The CMDTEO status flag in SDMMC_EISTR is enabled.

Downloaded from Arrow.com.

47.10.27 SDMMC Normal Interrupt Signal Enable Register (SD_SDIO)

Name: SDMMC_NISIER (SD_SDIO)
Offset: 0x38
Reset: 0x0000
Property: Read/Write

Bit	15	14	13	12	11	10	9	8
								CINT
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 8 – CINT Card Interrupt Signal Enable
0 (MASKED): No interrupt is generated when the CINT status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the CINT status rises in SDMMC_NISTR.

Bit 5 – BRDRDY Buffer Read Ready Signal Enable
0 (MASKED): No interrupt is generated when the BRDRDY status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the BRDRDY status rises in SDMMC_NISTR.

Bit 4 – BWRRDY Buffer Write Ready Signal Enable
0 (MASKED): No interrupt is generated when the BWRRDY status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the BWRRDY status rises in SDMMC_NISTR.

Bit 3 – DMAINT DMA Interrupt Signal Enable
0 (MASKED): No interrupt is generated when the DMAINT status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the DMAINT status rises in SDMMC_NISTR.

Bit 2 – BLKGE Block Gap Event Signal Enable
0 (MASKED): No interrupt is generated when the BLKGE status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the BLKGE status rises in SDMMC_NISTR.

Bit 1 – TRFC Transfer Complete Signal Enable
0 (MASKED): No interrupt is generated when the TRFC status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the TRFC status rises in SDMMC_NISTR.

Bit 0 – CMDC Command Complete Signal Enable
0 (MASKED): No interrupt is generated when the CMDC status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the CMDC status rises in SDMMC_NISTR.

Downloaded from Arrow.com.

47.10.28 SDMMC Normal Interrupt Signal Enable Register (e.MMC)

Name: SDMMC_NISIER (e.MMC)
Offset: 0x38
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the e.MMC operation mode.

Bit	15	14	13	12	11	10	9	8
		BOOTAR						
Access		R/W						
Reset		0						

Bit	7	6	5	4	3	2	1	0
			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 14 – BOOTAR Boot Acknowledge Received Signal Enable
0 (MASKED): No interrupt is generated when the BOOTAR status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the BOOTAR status rises in SDMMC_NISTR.

Bit 5 – BRDRDY Buffer Read Ready Signal Enable
0 (MASKED): No interrupt is generated when the BRDRDY status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the BRDRDY status rises in SDMMC_NISTR.

Bit 4 – BWRRDY Buffer Write Ready Signal Enable
0 (MASKED): No interrupt is generated when the BWRRDY status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the BWRRDY status rises in SDMMC_NISTR.

Bit 3 – DMAINT DMA Interrupt Signal Enable
0 (MASKED): No interrupt is generated when the DMAINT status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the DMAINT status rises in SDMMC_NISTR.

Bit 2 – BLKGE Block Gap Event Signal Enable
0 (MASKED): No interrupt is generated when the BLKGE status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the BLKGE status rises in SDMMC_NISTR.

Bit 1 – TRFC Transfer Complete Signal Enable
0 (MASKED): No interrupt is generated when the TRFC status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the TRFC status rises in SDMMC_NISTR.

Bit 0 – CMDC Command Complete Signal Enable
0 (MASKED): No interrupt is generated when the CMDC status rises in SDMMC_NISTR.
1 (ENABLED): An interrupt is generated when the CMDC status rises in SDMMC_NISTR.

Downloaded from Arrow.com.

47.10.29 SDMMC Error Interrupt Signal Enable Register (SD_SDIO)

Name: SDMMC_EISIER (SD_SDIO)
Offset: 0x3A
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the SD/SDIO operation mode.

Bit	15	14	13	12	11	10	9	8
							ADMA	ACMD
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 9 – ADMA ADMA Error Signal Enable
0 (MASKED): No interrupt is generated when the ADMA status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the ADMA status rises in SDMMC_EISTR.

Bit 8 – ACMD Auto CMD Error Signal Enable
0 (MASKED): No interrupt is generated when the ACMD status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the ACMD status rises in SDMMC_EISTR.

Bit 7 – CURLIM Current Limit Error Signal Enable
0 (MASKED): No interrupt is generated when the CURLIM status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the CURLIM status rises in SDMMC_EISTR.

Bit 6 – DATEND Data End Bit Error Signal Enable
0 (MASKED): No interrupt is generated when the DATEND status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the DATEND status rises in SDMMC_EISTR.

Bit 5 – DATCRC Data CRC Error Signal Enable
0 (MASKED): No interrupt is generated when the DATCRC status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the DATCRC status rises in SDMMC_EISTR.

Bit 4 – DATTEO Data Timeout Error Signal Enable
0 (MASKED): No interrupt is generated when the DATTEO status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the DATTEO status rises in SDMMC_EISTR.

Bit 3 – CMDIDX Command Index Error Signal Enable
0 (MASKED): No interrupt is generated when the CMDIDX status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the CMDIDX status rises in SDMMC_EISTR.

Bit 2 – CMDEND Command End Bit Error Signal Enable
0 (MASKED): No interrupt is generated when the CMDEND status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the CMDEND status rises in SDMMC_EISTR.

Bit 1 – CMDCRC Command CRC Error Signal Enable
0 (MASKED): No interrupt is generated when the CDMCRC status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the CMDCRC status rises in SDMMC_EISTR.

Bit 0 – CMDTEO Command Timeout Error Signal Enable
0 (MASKED): No interrupt is generated when the CMDTEO status rises in SDMMC_EISTR.

Downloaded from Arrow.com.

1 (ENABLED): An interrupt is generated when the CMDTEO status rises in SDMMC_EISTR.

Downloaded from Arrow.com.

47.10.30 SDMMC Error Interrupt Signal Enable Register (e.MMC)

Name: SDMMC_EISIER (e.MMC)
Offset: 0x3A
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the e.MMC operation mode.

Bit	15	14	13	12	11	10	9	8
				BOOTAE			ADMA	ACMD
Access								
Reset				0			0	0

Bit	7	6	5	4	3	2	1	0
	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
Access								
Reset	0	0	0	0	0	0	0	0

Bit 12 – BOOTAE Boot Acknowledge Error Signal Enable
0 (MASKED): No interrupt is generated when the BOOTAE status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the BOOTAE status rises in SDMMC_EISTR.

Bit 9 – ADMA ADMA Error Signal Enable
0 (MASKED): No interrupt is generated when the ADMA status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the ADMA status rises in SDMMC_EISTR.

Bit 8 – ACMD Auto CMD Error Signal Enable
0 (MASKED): No interrupt is generated when the ACMD status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the ACMD status rises in SDMMC_EISTR.

Bit 7 – CURLIM Current Limit Error Signal Enable
0 (MASKED): No interrupt is generated when the CURLIM status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the CURLIM status rises in SDMMC_EISTR.

Bit 6 – DATEND Data End Bit Error Signal Enable
0 (MASKED): No interrupt is generated when the DATEND status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the DATEND status rises in SDMMC_EISTR.

Bit 5 – DATCRC Data CRC Error Signal Enable
0 (MASKED): No interrupt is generated when the DATCRC status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the DATCRC status rises in SDMMC_EISTR.

Bit 4 – DATTEO Data Timeout Error Signal Enable
0 (MASKED): No interrupt is generated when the DATTEO status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the DATTEO status rises in SDMMC_EISTR.

Bit 3 – CMDIDX Command Index Error Signal Enable
0 (MASKED): No interrupt is generated when the CMDIDX status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the CMDIDX status rises in SDMMC_EISTR.

Bit 2 – CMDEND Command End Bit Error Signal Enable
0 (MASKED): No interrupt is generated when the CMDEND status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the CMDEND status rises in SDMMC_EISTR.

Bit 1 – CMDCRC Command CRC Error Signal Enable
0 (MASKED): No interrupt is generated when the CDMCRC status rises in SDMMC_EISTR.

Downloaded from Arrow.com.

1 (ENABLED): An interrupt is generated when the CMDCRC status rises in SDMMC_EISTR.

Bit 0 – CMDTEO Command Timeout Error Signal Enable
0 (MASKED): No interrupt is generated when the CMDTEO status rises in SDMMC_EISTR.
1 (ENABLED): An interrupt is generated when the CMDTEO status rises in SDMMC_EISTR.

Downloaded from Arrow.com.

47.10.31 SDMMC Auto CMD Error Status Register

Name: SDMMC_ACESR
Offset: 0x3C
Reset: 0x0000
Property: Read-only

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CMDNI			ACMDIDX	ACMDEND	ACMDCRC	ACMDTEO	ACMD12NE
Access	R			R	R	R	R	R
Reset	0			0	0	0	0	0

Bit 7 – CMDNI Command Not Issued by Auto CMD12 Error
This bit is set to 1 when CMD_wo_DAT is not executed due to an Auto CMD12 error (SDMMC_ACESR[4:1]). This bit is set to 0 when Auto CMD Error is generated by Auto CMD23.

Value	Description
0	No error.
1	Error.

Bit 4 – ACMDIDX Auto CMD Index Error
This bit is set to 1 when the Command Index error occurs in response to a command.

Value	Description
0	No error.
1	Error.

Bit 3 – ACMDEND Auto CMD End Bit Error
This bit is set to 1 when detecting that the end bit of the command response is 0.

Value	Description
0	No error.
1	Error.

Bit 2 – ACMDCRC Auto CMD CRC Error
This bit is set to 1 when detecting a CRC error in the command response (see the table above for more details).

Bit 1 – ACMDTEO Auto CMD Timeout Error
This bit is set to 1 if no response is returned within 64 SDCLK cycles from the end bit of the command. If this bit is set to 1, the other error status bits (SDMMC_ACESR[4:2]) are meaningless.

Table 47-4. Relation between ACMDCRC and ACMDTEO

ACMDCRC	ACMDTEO	Types of error
0	0	No error
0	1	Response Timeout error
1	0	Response CRC error
1	1	CMD line conflict

Bit 0 – ACMD12NE Auto CMD12 Not Executed
If a memory multiple block data transfer is not started due to a command error, this bit is not set to 1 because it is not necessary to issue Auto CMD12. Setting this bit to 1 means the SDMMC cannot issue Auto CMD12 to stop a

Downloaded from Arrow.com.

memory multiple block data transfer due to some error. If this bit is set to 1, other error status bits (SDMMC_ACESR[4:1]) are meaningless.
This bit is set to 0 when an Auto CMD error is generated by Auto CMD23.

Value	Description
0	No error.
1	Error.

Downloaded from Arrow.com.

47.10.32 SDMMC Host Control 2 Register (SD_SDIO)

Name: SDMMC_HC2R (SD_SDIO)
Offset: 0x3E
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the SD/SDIO operation mode.

Bit	15	14	13	12	11	10	9	8
	PVALEN	ASINTEN						
Access	R/W	R/W						
Reset	0	0						

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 15 – PVALEN Preset Value Enable
As the operating SDCLK frequency and I/O driver strength depend on the system implementation, it is difficult to determine these parameters in the standard host driver. When PVALEN is set to 1, automatic SDCLK frequency generation and driver strength selection are performed without considering system-specific conditions. This bit enables the functions defined in SDMMC_PVR.
If this bit is set to 0, SDMMC_CCR.SDCLKFSEL and SDMMC_CCR.CLKGSEL are set by the user.
If this bit is set to 1, SDMMC_CCR.SDCLKFSEL and SDMMC_CCR.CLKGSEL are set by the SDMMC as specified in SDMMC_PVR.

Value	Description
0	SDCLK and Driver strength are controlled by the user.
1	Automatic selection by Preset Value is enabled.

Bit 14 – ASINTEN Asynchronous Interrupt Enable
This bit can be set to 1 if a card supports asynchronous interrupts and Asynchronous Interrupt Support (ASINTSUP) is set to 1 in SDMMC_CA0R. Asynchronous interrupt is effective when DAT[1] interrupt is used in 4-bit SD mode. If this bit is set to 1, the user can stop the SDCLK during the asynchronous interrupt period to save power. During this period, the SDMMC continues to deliver the Card Interrupt to the host when it is asserted by the card.

Value	Description
0	Disabled
1	Enabled

Downloaded from Arrow.com.

47.10.33 SDMMC Host Control 2 Register (e.MMC)

Name: SDMMC_HC2R (e.MMC)
Offset: 0x3E
Reset: 0x0000
Property: Read/Write

Note: This register configuration is specific to the e.MMC operation mode.

Bit	15	14	13	12	11	10	9	8
	PVALEN							
Access	R/W							
Reset	0							

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 15 – PVALEN Preset Value Enable
As the operating SDCLK frequency and I/O driver strength depend on the system implementation, it is difficult to determine these parameters in the standard host driver. When Preset Value Enable (PVALEN) is set to 1, automatic SDCLK frequency generation and driver strength selection are performed without considering system-specific conditions. This bit enables the functions defined in SDMMC_PVR.
If this bit is set to 0, SDMMC_CCR.SDCLKFSEL and SDMMC_CCR.CLKGSEL are set by the user.
If this bit is set to 1, SDMMC_CCR.SDCLKFSEL and SDMMC_CCR.CLKGSEL are set by the SDMMC as specified in SDMMC_PVR.

Value	Description
0	SDCLK and Driver strength are controlled by the user.
1	Automatic selection by Preset Value is enabled.

Downloaded from Arrow.com.

47.10.34 SDMMC Capabilities 0 Register

Name: SDMMC_CA0R
Offset: 0x40
Reset: 0x27E832B2
Property: Read/Write

Note: The reset values match the capabilities of the MPU alone. The user should adjust the capability registers so that they also match board design. Modify preset values only if the Capabilities Write Enable (CAPWREN) bit is set to 1 in SDMMC_CACR.

Bit	31	30	29	28	27	26	25	24
	SLTYPE[1:0]		ASINTSUP	SB64SUP		V18VSUP	V30VSUP	V33VSUP
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	1	0		1	1	1

Bit	23	22	21	20	19	18	17	16
	SRSUP	SDMASUP	HSSUP		ADMA2SUP	ED8SUP	MAXBLKL[1:0]	
Access	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset	1	1	1		1	0	0	0

Bit	15	14	13	12	11	10	9	8
	BASECLKF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	0	0	1	0

Bit	7	6	5	4	3	2	1	0
	TEOCLKU		TEOCLKF[5:0]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	1		1	1	0	0	1	0

Bits 31:30 – SLTYPE[1:0] Slot Type
This field indicates usage of a slot by a specific system. An SDMMC control register set is defined per slot. Embedded Slot for One Device means that only one nonremovable device is connected to a bus slot.
The Standard Host Driver controls a removable card (SLTYPE = 0) or one embedded device (SLTYPE = 1) connected to an SD bus slot.

Value	Name	Description
0	REMOVABLECARD	Removable Card Slot
1	EMBEDDED	Embedded Slot for One Device
2	–	Reserved
3	–	Reserved

Bit 29 – ASINTSUP Asynchronous Interrupt Support
See section “Asynchronous Interrupt” in the “SDIO Simplified Specification V3.00” .

Value	Description
0	Asynchronous interrupt not supported.
1	Asynchronous interrupt supported.

Bit 28 – SB64SUP 64-Bit System Bus Support
Reading this bit to 1 means that the SDMMC supports the 64-bit Address Descriptor mode and is connected to the 64-bit address system bus.

Value	Description
0	64-bit address bus not supported.
1	64-bit address bus supported.

Bit 26 – V18VSUP Voltage Support 1.8V

Downloaded from Arrow.com.

Value	Description
0	1.8V voltage supply not supported.
1	1.8V voltage supply supported.

Bit 25 – V30VSUP Voltage Support 3.0V

Value	Description
0	3.0V voltage supply not supported.
1	3.0V voltage supply supported.

Bit 24 – V33VSUP Voltage Support 3.3V

Value	Description
0	3.3V voltage supply not supported.
1	3.3V voltage supply supported.

Bit 23 – SRSUP Suspend/Resume Support
This bit indicates whether the SDMMC supports the Suspend/Resume functionality. If this bit is set to 0, the user does not issue either Suspend or Resume commands because the Suspend and Resume mechanism (see “Suspend and Resume Mechanism” in the “SD Host Controller Simplified Specification V3.00”) is not supported.

Value	Description
0	Suspend/Resume not supported.
1	Suspend/Resume supported.

Bit 22 – SDMASUP SDMA Support
This bit indicates whether the SDMMC is capable of using SDMA to transfer data between system memory and the SDMMC directly.

Value	Description
0	SDMA not supported.
1	SDMA supported.

Bit 21 – HSSUP High Speed Support
This bit indicates whether the SDMMC and the system support High Speed mode and they can supply SDCLK frequency from 25 MHz to 50 MHz.

Value	Description
0	High Speed not supported.
1	High Speed supported.

Bit 19 – ADMA2SUP ADMA2 Support
This bit indicates whether the SDMMC is capable of using ADMA2.

Value	Description
0	ADMA2 not supported.
1	ADMA2 supported.

Bit 18 – ED8SUP 8-Bit Support for Embedded Device
This bit indicates whether the SDMMC is capable of using the 8-bit Bus Width mode.

Value	Description
0	8-bit bus width not supported.
1	8-bit bus width supported.

Bits 17:16 – MAXBLKL[1:0] Max Block Length
This field indicates the maximum block size that the user can read and write to the buffer in the SDMMC. Three sizes can be defined, as shown below. It is noted that the transfer block length is always 512 bytes for SD Memory Cards regardless of this field.

Value	Name	Description
0	512	512 bytes
1	1024	1024 bytes
2	2048	2048 bytes
3	NONE	Reserved

Downloaded from Arrow.com.

Bits 15:8 – BASECLKF[7:0] Base Clock Frequency
This value indicates the frequency of the base clock (BASECLK). The user uses this value to calculate the clock divider value (see SDCLK Frequency Select (SDCLKFSEL) in SDMMC_CCR).
If this field is set to 0, the user must get the information via another method.
$f_{\mathrm{BASECLK}} = \mathrm{BASECLKF}_{\mathrm{MHz}}$

Bit 7 – TEOCLKU Timeout Clock Unit
This bit shows the unit of the base clock frequency used to detect Data Timeout Error.

Value	Description
0	KHz
1	MHz

Bits 5:0 – TEOCLKF[5:0] Timeout Clock Frequency
This bit shows the timeout clock frequency (TEOCLK) used to detect Data Timeout Error.
If this field is set to 0, the user must get the information via another method.
The Timeout Clock Unit (TEOCLKU) defines the unit of this field's value.
– TEOCLKU = 0: $f_{\mathrm{TEOCLK}} = \mathrm{TEOCLKF}_{\mathrm{KHz}}$
– TEOCLKU = 1: $f_{\mathrm{TEOCLK}} = \mathrm{TEOCLKF}_{\mathrm{MHz}}$

Downloaded from Arrow.com.

47.10.35 SDMMC Capabilities 1 Register

Name: SDMMC_CA1R
Offset: 0x44
Reset: 0x27E832B2
Property: Read/Write

Note: The reset values match the capabilities of the MPU alone. The user should adjust the capability registers so that they also match board design. Modify preset values only if the Capabilities Write Enable (CAPWREN) bit is set to 1 in SDMMC_CACR.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CLKMULT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		DRVDSUP	DRVCSUP	DRVASUP		DDR50SUP	SDR104SUP	SDR50SUP
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	1	1		0	1	0

Bits 23:16 – CLKMULT[7:0] Clock Multiplier
This field indicates the multiplier factor between the Base Clock (BASECLK) used for the Divided Clock Mode and the Multiplied Clock (MULTCLK) used for the Programmable Clock mode (see SDMMC_CCR).
Reading this field to 0 means that the Programmable Clock mode is not supported.
$f_{\mathrm{MULTCLK}} = f_{\mathrm{BASECLK}} \times (\mathrm{CLKMULT} + 1)$

Bit 6 – DRVDSUP Driver Type D Support

Value	Description
0	Driver type D is not supported.
1	Driver type D is supported.

Bit 5 – DRVCSUP Driver Type C Support

Value	Description
0	Driver type C is not supported.
1	Driver type C is supported.

Bit 4 – DRVASUP Driver Type A Support

Value	Description
0	Driver type A is not supported.
1	Driver type A is supported.

Bit 2 – DDR50SUP DDR50 Support

Value	Description
0	DDR50 mode is not supported.
1	DDR50 mode is supported.

Downloaded from Arrow.com.

Bit 1 – SDR104SUP SDR104 Support

Value	Description
0	SDR104 mode is not supported.
1	SDR104 mode is supported.

Bit 0 – SDR50SUP SDR50 Support

Value	Description
0	SDR50 mode is not supported.
1	SDR50 mode is supported.

Downloaded from Arrow.com.

47.10.36 SDMMC Maximum Current Capabilities Register

Name: SDMMC_MCCAR
Offset: 0x48
Reset: 0x00000000
Property: Read/Write

Note: The user should adjust the Maximum Current Capabilities register so that it matches board design. The preset values can be modified only if the Capabilities Write Enable (CAPWREN) bit is set to 1 in SDMMC_CACR.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	MAXCUR18V[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MAXCUR30V[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MAXCUR33V[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – MAXCUR18V[7:0] Maximum Current for 1.8V
This field indicates the maximum current capability for 1.8V voltage. This value is meaningful only if V18VSUP is set to 1 in SDMMC_CA0R. Reading MAXCUR18V at 0 means that the user must get information via another method.
$\mathrm{Imax}_{\mathrm{mA}} = 4 \times \mathrm{MAXCURR18}V$

Bits 15:8 – MAXCUR30V[7:0] Maximum Current for 3.0V
This field indicates the maximum current capability for 3.0V voltage. This value is meaningful only if V30VSUP is set to 1 in SDMMC_CA0R. Reading MAXCUR30V at 0 means that the user must get information via another method.
$\mathrm{Imax}_{\mathrm{mA}} = 4 \times \mathrm{MAXCURR30}V$

Bits 7:0 – MAXCUR33V[7:0] Maximum Current for 3.3V
This field indicates the maximum current capability for 3.3V voltage. This value is meaningful only if V33VSUP is set to 1 in SDMMC_CA0R. Reading MAXCUR33V at 0 means that the user must get information via another method.
$\mathrm{Imax}_{\mathrm{mA}} = 4 \times \mathrm{MAXCURR33}V$

Downloaded from Arrow.com.

47.10.37 SDMMC Force Event Register for Auto CMD Error Status

Name: SDMMC_FERACES
Offset: 0x50
Reset: –
Property: Write-only

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CMDNI			ACMDIDX	ACMDEND	ACMDCRC	ACMDTEO	ACMD12NE
Access	W			W	W	W	W	W
Reset	–			–	–	–	–	–

Bit 7 – CMDNI Force Event for Command Not Issued by Auto CMD12 Error
For test purposes, the user can write this bit to 1 to raise the CMDNI status flag in SDMMC_ACESR.
Writing this bit to 0 has no effect.

Bit 4 – ACMDIDX Force Event for Auto CMD Index Error
For test purposes, the user can write this bit to 1 to raise the ACMDIDX status flag in SDMMC_ACESR.
Writing this bit to 0 has no effect.

Bit 3 – ACMDEND Force Event for Auto CMD End Bit Error
For test purposes, the user can write this bit to 1 to raise the ACMDEND status flag in SDMMC_ACESR.
Writing this bit to 0 has no effect.

Bit 2 – ACMDCRC Force Event for Auto CMD CRC Error
For test purposes, the user can write this bit to 1 to raise the ACMDCRC status flag in SDMMC_ACESR.
Writing this bit to 0 has no effect.

Bit 1 – ACMDTEO Force Event for Auto CMD Timeout Error
For test purposes, the user can write this bit to 1 to raise the ACMDTEO status flag in SDMMC_ACESR.
Writing this bit to 0 has no effect.

Bit 0 – ACMD12NE Force Event for Auto CMD12 Not Executed
For test purposes, the user can write this bit to 1 to raise the ACMD12NE status flag in SDMMC_ACESR.
Writing this bit to 0 has no effect.

Downloaded from Arrow.com.

47.10.38 SDMMC Force Event Register for Error Interrupt Status

Name: SDMMC_FEREIS
Offset: 0x52
Reset: –
Property: Write-only

Bit	15	14	13	12	11	10	9	8
				BOOTAE			ADMA	ACMD
Access				W			W	W
Reset				–			–	–

Bit	7	6	5	4	3	2	1	0
	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 12 – BOOTAE Force Event for Boot Acknowledge Error
For test purposes, the user can write this bit to 1 to raise the BOOTAE status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 9 – ADMA Force Event for ADMA Error
For test purposes, the user can write this bit to 1 to raise the ADMA status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 8 – ACMD Force Event for Auto CMD Error
For test purposes, the user can write this bit to 1 to raise the ACMD status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 7 – CURLIM Force Event for Current Limit Error
For test purposes, the user can write this bit to 1 to raise the CURLIM status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 6 – DATEND Force Event for Data End Bit Error
For test purposes, the user can write this bit to 1 to raise the DATEND status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 5 – DATCRC Force Event for Data CRC error
For test purposes, the user can write this bit to 1 to raise the DATCRC status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 4 – DATTEO Force Event for Data Timeout error
For test purposes, the user can write this bit to 1 to raise the DATTEO status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 3 – CMDIDX Force Event for Command Index Error
For test purposes, the user can write this bit to 1 to raise the CMDIDX status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 2 – CMDEND Force Event for Command End Bit Error
For test purposes, the user can write this bit to 1 to raise the CDMEND status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Bit 1 – CMDCRC Force Event for Command CRC Error
For test purposes, the user can write this bit to 1 to raise the CMDCRC status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Downloaded from Arrow.com.

Bit 0 – CMDTEO Force Event for Command Timeout Error
For test purposes, the user can write this bit to 1 to raise the CMDTEO status flag in SDMMC_EISTR.
Writing this bit to 0 has no effect.

Downloaded from Arrow.com.

47.10.39 SDMMC ADMA Error Status Register

Name: SDMMC_AESR
Offset: 0x54
Reset: 0x00
Property: Read-only

Bit	7	6	5	4	3	2	1	0
						LMIS	ERRST[1:0]	
Access						R	R	R
Reset						0	0	0

Bit 2 – LMIS ADMA Length Mismatch Error
This error occurs in the following two cases:
– While Block Count Enable (BCEN) is being set, the total data length specified by the Descriptor table is different from that specified by the Block Count (BLKCNT) and Transfer Block Size (BLKSIZE).
– The total data length cannot be divided by the Transfer Block Size (BLKSIZE).

Value	Description
0	No error
1	Error

Bits 1:0 – ERRST[1:0] ADMA Error State
This field indicates the state of ADMA when an error has occurred during an ADMA data transfer. This field never reads 2 because ADMA never stops in this state.

Value	Name	Description
0	STOP	(Stop DMA) SDMMC_ASAR points to the descriptor following the error descriptor
1	FDS	(Fetch Descriptor) SDMMC_ASAR points to the error descriptor
2	–	(Not used)
3	TFR	(Transfer Data) SDMMC_ASAR points to the descriptor following the error descriptor

Downloaded from Arrow.com.

47.10.40 SDMMC ADMA System Address Register 0

Name: SDMMC_ASAR0
Offset: 0x58
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ADMASA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADMASA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADMASA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADMASA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADMASA[31:0] ADMA System Address
This field holds the byte address of the executing command of the descriptor table. The 32-bit address descriptor uses SDMMC_ASAR. At the start of ADMA, the user must set the start address of the descriptor table. The ADMA increments this register address, which points to the next Descriptor line to be fetched.
When the ADMA Error (ADMA) status flag rises, this field holds a valid descriptor address depending on the ADMA Error State (ERRST). The user must program Descriptor Table on 32-bit boundary and set 32-bit boundary address to this register. ADMA2 ignores the lower 2 bits of this register and assumes it to be 0.

Downloaded from Arrow.com.

47.10.41 SDMMC Preset Value Register

Name: SDMMC_PVRx
Offset: 0x60 + x*0x02 [x=0..3]
Reset: 0x00000000
Property: Read/Write

One of the Preset Value Registers is effective based on the selected bus speed mode. The table below defines the conditions to select one of the SDMMC_PVRs.

Table 47-5. Preset Value Register Select Condition

Selected Bus Speed Mode	HSEN (SDMMC_HC1R)
Default Speed	0
High Speed	1

The table below shows the effective Preset Value Register according to the Selected Bus Speed mode.

Table 47-6. Preset Value Registers

SDMMC_PVRx	Selected Bus Speed Mode	Signal Voltage
SDMMC_PVR0	Initialization	3.3V
SDMMC_PVR1	Default Speed	3.3V
SDMMC_PVR2	High Speed	3.3V

When Preset Value Enable (PVALEN) in SDMMC_HC2R is set to 1, SDCLK Frequency Select (SDLCKFSEL) and Clock Generator Select (CLKGSEL) in SDMMC_CCR are automatically set based on the Selected Bus Speed mode. This means that the user does not need to set these fields when preset is enabled. A Preset Value Register for Initialization (SDMMC_PVR0) is not selected by Bus Speed mode. Before starting the initialization sequence, the user needs to set a clock preset value to SDCLKFSEL in SDMMC_CCR.PVALEN can be set to 1 after the initialization is completed.

Bit	15	14	13	12	11	10	9	8
						CLKGSEL	SDCLKFSEL[9:8]	
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	SDCLKFSEL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 10 – CLKGSEL Clock Generator Select
See CLKGSEL in SDMMC_CCR.

Bits 9:0 – SDCLKFSEL[9:0] SDCLK Frequency Select
See SDCLKFSEL in SDMMC_CCR.

Downloaded from Arrow.com.

47.10.42 SDMMC Slot Interrupt Status Register

Name: SDMMC_SISR
Offset: 0xFC
Reset: 0x0000
Property: Read-only

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							INTSSL[1:0]	
Access							R	R
Reset							0	0

Bits 1:0 – INTSSL[1:0] Interrupt Signal for Each Slot
These status bits indicate the logical OR of Interrupt Signals and Wakeup Signal for each SDMMC instance in the product (INTSSL[x] corresponds to SDMMCx instance in the product).

Downloaded from Arrow.com.

47.10.43 SDMMC Host Controller Version Register

Name: SDMMC_HCVR
Offset: 0xFE
Reset: 0x1802
Property: Read-only

Bit	15	14	13	12	11	10	9	8
	VVER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset								

Bit	7	6	5	4	3	2	1	0
	SVER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset								

Bits 15:8 – VVER[7:0] Vendor Version Number
Reserved. Value subject to change. No functionality associated. This is the internal version of the module.

Bits 7:0 – SVER[7:0] Specification Version Number
This status indicates the SD Host Controller Specification Version.

Value	Name
0	SD Host Specification Version 1.00
1	SD Host Specification Version 2.00, including the feature of the ADMA and Test Register
2	SD Host Specification Version 3.00

Downloaded from Arrow.com.

47.10.44 SDMMC Additional Present State Register

Name: SDMMC_APSR
Offset: 0x200
Reset: –
Property: Read-only

The register reset values are:

Instance	Reset Value
SDMMC0, SDMMC1	0x0000000F

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					HDATLL[3:0]			
Access					R	R	R	R
Reset					–	–	–	–

Bits 3:0 – HDATLL[3:0] DAT[7:4] High Line Level
This status is used to check the DAT[7:4] line level to recover from errors, and for debugging.

Downloaded from Arrow.com.

47.10.45 SDMMC e.MMC Control 1 Register

Name: SDMMC_MC1R
Offset: 0x204
Reset: 0x00
Property: Read/Write

Bit	7	6	5	4	3	2	1	0
			BOOTA	OPD	DDR		CMDTYP[1:0]	
Access			R/W	R/W	R/W		R/W	R/W
Reset			0	0	0		0	0

Bit 5 – BOOTA e.MMC Boot Acknowledge Enable
This bit must be set according to the value of BOOT_ACK in the Extended CSD Register (see “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51”).
When this bit is set to 1, the SDMMC waits for boot acknowledge pattern from the e.MMC before receiving boot data.
If the boot acknowledge pattern is wrong, the BOOTAE status flag rises in SDMMC_EISTR if BOOTAE is set in SDMMC_EISTER. An interrupt is generated if BOOTAE is set in SDMMC_EISIER.
If the no boot acknowledge pattern is received, the DATTEO status flag rises in SDMMC_EISTR if DATTEO is set in SDMMC_EISTER. An interrupt is generated if DATTEO is set in SDMMC_EISIER.

Bit 4 – OPD e.MMC Open Drain Mode
This bit sets the command line in open drain.

Value	Description
0	The command line is in push-pull.
1	The command line is in open drain.

Bit 3 – DDR e.MMC HSDDR Mode
This bit selects the High Speed DDR mode.
The clock divider (DIV) in SDMMC_CCR must be set to a value different from 0 when DDR is 1.

Value	Description
0	High Speed DDR is not selected.
1	High Speed DDR is selected.

Bits 1:0 – CMDTYP[1:0] e.MMC Command Type

Value	Name	Description
0	NORMAL	The command is not an e.MMC specific command.
1	WAITIRQ	This bit must be set to 1 when the e.MMC is in Interrupt mode (CMD40). See “Interrupt Mode” in the “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51” .
2	STREAM	This bit must be set to 1 in the case of Stream Read(CMD11) or Stream Write (CMD20). Only effective for e.MMC up to revision 4.41.
3	BOOT	Starts a Boot Operation mode at the next write to SDMMC_CR. Boot data are read directly from e.MMC device.

Downloaded from Arrow.com.

47.10.46 SDMMC e.MMC Control 2 Register

Name: SDMMC_MC2R
Offset: 0x205
Reset: –
Property: Write-only

Bit	7	6	5	4	3	2	1	0
							ABOOT	SRESP
Access							W	W
Reset							–	–

Bit 1 – ABOOT e.MMC Abort Boot
This bit is used to exit from Boot mode. Writing this bit to 1 exits the Boot Operation mode. Writing 0 is ignored.

Bit 0 – SRESP e.MMC Abort Wait IRQ
This bit is used to exit from the Interrupt mode. When this bit is written to 1, the SDMMC sends the CMD40 response automatically. This brings the e.MMC from Interrupt mode to the standard Data Transfer mode. Writing this bit to 0 is ignored.
This bit is only effective when CMD_TYP in SDMMC_MC1R is set to WAITIRQ.

Downloaded from Arrow.com.

47.10.47 SDMMC AHB Control Register

Name: SDMMC_ACR
Offset: 0x208
Reset: 0x00
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							BMAX[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – BMAX[1:0] AHB Maximum Burst
This field selects the maximum burst size in case of DMA transfer.

Value	Name	Description
0	INCR16	The maximum burst size is INCR16.
1	INCR8	The maximum burst size is INCR8.
2	INCR4	The maximum burst size is INCR4.
3	SINGLE	Only SINGLE transfers are performed.

Downloaded from Arrow.com.

47.10.48 SDMMC Clock Control 2 Register

Name: SDMMC_CC2R
Offset: 0x20C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								R/W
Reset								0

Bit 0 – FSDCLKD Force SDCLK Disabled
The user can choose to maintain the SDCLK during 8 SDCLK cycles after the end bit of the last data block in case of a read transaction, or after the end bit of the CRC status in case of a write transaction.

Value	Description
0	The SDCLK is forced and it cannot be stopped immediately after the transaction.
1	The SDCLK is not forced and it can be stopped immediately after the transaction.

Downloaded from Arrow.com.

47.10.49 SDMMC Capabilities Control Register

Name: SDMMC_CACR
Offset: 0x230
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	KEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								CAPWREN
Access								R/W
Reset								0

Bits 15:8 – KEY[7:0] Key

Value	Name	Description
0x46	KEY	Writing any other value in this field aborts the write operation of the CAPWREN bit. Always reads as 0.

Bit 0 – CAPWREN Capabilities Write Enable
This bit can only be written if the value of KEY corresponds to 0x46.

Value	Description
0	Capabilities registers (SDMMC_CA0R, SDMMC_CA1R and SDMMC_CA1R) cannot be written.
1	Capabilities registers (SDMMC_CA0R, SDMMC_CA1R and SDMMC_CA1R) can be written.

Downloaded from Arrow.com.

47.10.50 SDMMC Debug Register

Name: SDMMC_DBGR
Offset: 0x234
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								NIDBG
Access								R/W
Reset								0

Bit 0 – NIDBG Nonintrusive Debug
0 (DISABLED): Reading the SDMMC_BDPR via debugger increments the dual port RAM read pointer.
1 (ENABLED): Reading the SDMMC_BDPR via debugger does not increment the dual port RAM read pointer.

Downloaded from Arrow.com.

48. Image Sensor Interface (ISI)

48.1 Description

The Image Sensor Interface (ISI) connects a CMOS-type image sensor to the processor and provides image capture in various formats. The ISI performs data conversion, if necessary, before the storage in memory through DMA.

The ISI supports color CMOS image sensor and grayscale image sensors with a reduced set of functionalities.

In Grayscale mode, the data stream is stored in memory without any processing and so is not compatible with the LCD controller.

Internal FIFOs on the preview and codec paths are used to store the incoming data. The RGB output on the preview path is compatible with the LCD controller. This module outputs the data in RGB format (LCD compatible) and has scaling capabilities to make it compliant to the LCD display resolution (see the table RGB Format in Default Mode, RGB_CFG = 00, No Swap).

Several input formats such as preprocessed RGB or YCbCr are supported through the data bus interface.

The ISI supports two synchronization modes:

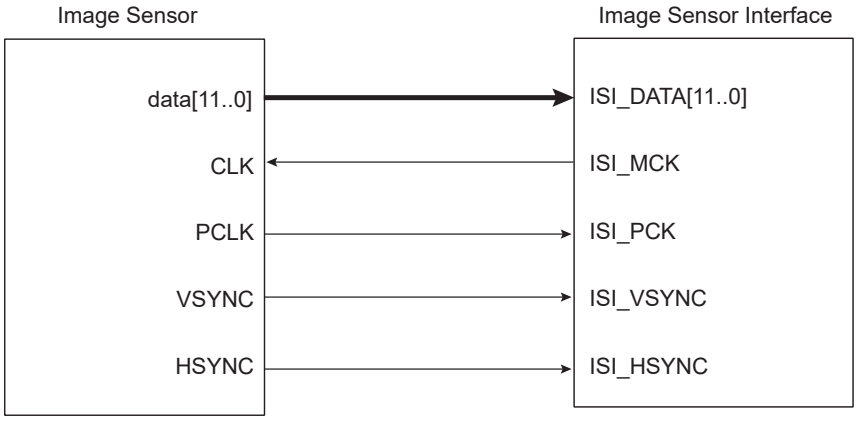
- Hardware with ISI_VSYNC and ISI_HSYNC signals
- International Telecommunication Union Recommendation ITU-R BT.656-4 Start-of-Active-Video (SAV) and End-of-Active-Video (EAV) synchronization sequence

Using EAV/SAV for synchronization reduces the pin count (ISI_VSYNC, ISI_HSYNC not used). The polarity of the synchronization pulse is programmable to comply with the sensor signals.

Table 48-1. I/O Description

Signal	Direction	Description
ISI_VSYNC	In	Vertical Synchronization
ISI_HSYNC	In	Horizontal Synchronization
ISI_DATA[11..0]	In	Sensor Pixel Data
ISI_MCK	Out	Master Clock provided to the Image Sensor. See Clocks.
ISI_PCK	In	Pixel Clock provided by the Image Sensor

Figure 48-1. ISI Connection Example



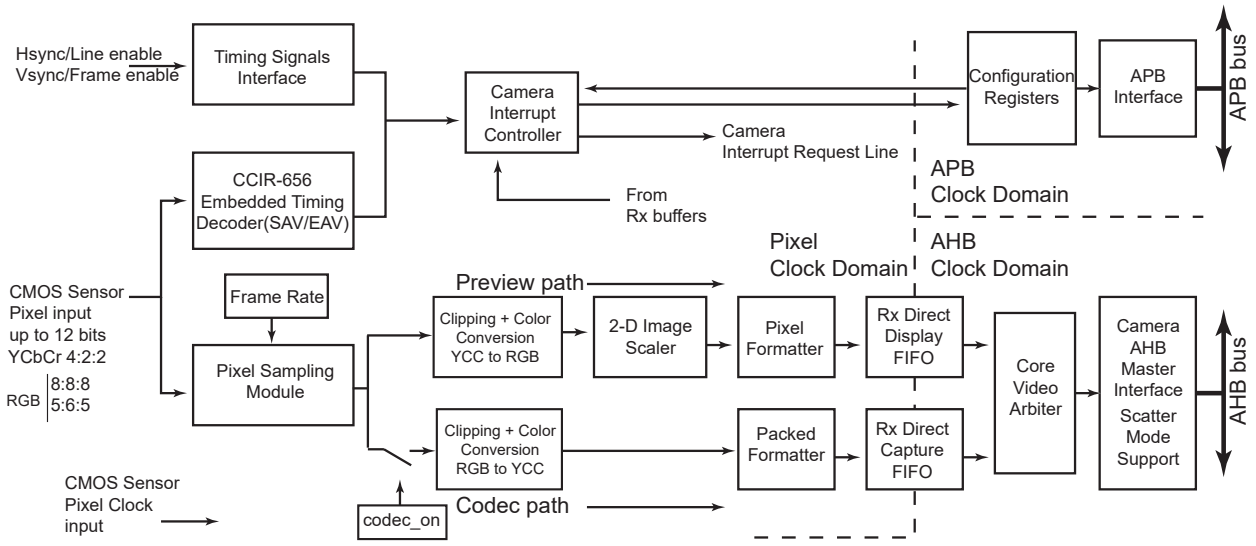
Downloaded from Arrow.com.

48.2 Embedded Characteristics

- ITU-R BT. 601/656 8-bit Mode External Interface Support
- Supports up to 12-bit Grayscale CMOS Sensors
- Support for ITU-R BT.656-4 SAV and EAV Synchronization
- Vertical and Horizontal Resolutions up to 2048 × 2048
- Preview Path up to 640 × 480 in RGB Mode
- Codec Path up to 2048 × 2048
- 256-byte FIFO on Codec Path
- 256-byte FIFO on Preview Path
- Support for Packed Data Formatting for YCbCr 4:2:2 Formats
- Preview Scaler to Generate Smaller Size image
- Programmable Frame Capture Rate
- VGA, QVGA, CIF, QCIF Formats Supported for LCD Preview
- Custom Formats with Horizontal and Vertical Preview Size as Multiples of 16 Also Supported for LCD Preview

48.3 Block Diagram

Figure 48-2. ISI Block Diagram



48.4 Product Dependencies

48.4.1 I/O Lines

The pins used for interfacing the compliant external devices can be multiplexed with PIO lines. The programmer must first program the PIO controllers to assign the ISI pins to their peripheral functions.

48.4.2 Power Management

The ISI can be clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the ISI clock.

Note: The pixel clock frequency must be lower than the bus clock frequency (peripheral clock).

48.4.3 Interrupt Sources

The ISI interface has an interrupt line connected to the interrupt controller. Handling the ISI interrupt requires programming the interrupt controller before configuring the ISI.

Downloaded from Arrow.com.

48.5 Functional Description

The Image Sensor Interface (ISI) supports direct connection to the ITU-R BT. 601/656 8-bit mode compliant sensors and up to 12-bit grayscale sensors. It receives the image data stream from the image sensor on the 12-bit data bus.

This module receives up to 12 bits for data, the horizontal and vertical synchronizations and the pixel clock. The reduced pin count alternative for synchronization is supported for sensors that embed SAV (start of active video) and EAV (end of active video) delimiters in the data stream.

The Image Sensor Interface interrupt line is connected to the Advanced Interrupt Controller and can trigger an interrupt at the beginning of each frame and at the end of a DMA frame transfer. If the SAV/EAV synchronization is used, an interrupt can be triggered on each delimiter event.

For 8-bit color sensors, the data stream received can be in several possible formats: YCbCr 4:2:2, RGB 8:8:8, RGB 5:6:5 and may be processed before the storage in memory. When the preview DMA channel is configured and enabled, the preview path is activated and an ‘RGB frame’ is moved to memory. The preview path frame rate is configured with the FRATE field of the ISI_CFG1 register. When the codec DMA channel is configured and enabled, the codec path is activated and a ‘YCbCr 4:2:2 frame’ is captured as soon as the ISI_CDC bit of the ISI Control Register (ISI_CR) is set.

When the FULL bit of the ISI_CFG1 register is set, both preview DMA channel and codec DMA channel can operate simultaneously. When a zero is written to the FULL bit of the ISI_CFG1 register, a hardware scheduler checks the FRATE field. If its value is zero, a preview frame is skipped and a codec frame is moved to memory instead. If its value is other than zero, at least one free frame slot is available. The scheduler postpones the codec frame to that free available frame slot.

The data stream may be sent on both preview path and codec path if the value of bit ISI_CDC in the ISI_CR is one. To optimize the bandwidth, the codec path should be enabled only when a capture is required.

In Grayscale mode, the input data stream is stored in memory without any processing. The 12-bit data, which represent the grayscale level for the pixel, is stored in memory one or two pixels per word, depending on the GS_MODE bit in the ISI_CFG2 register. The codec datapath is not available when grayscale image is selected.

A frame rate counter allows users to capture all frames or 1 out of every 2 to 8 frames.

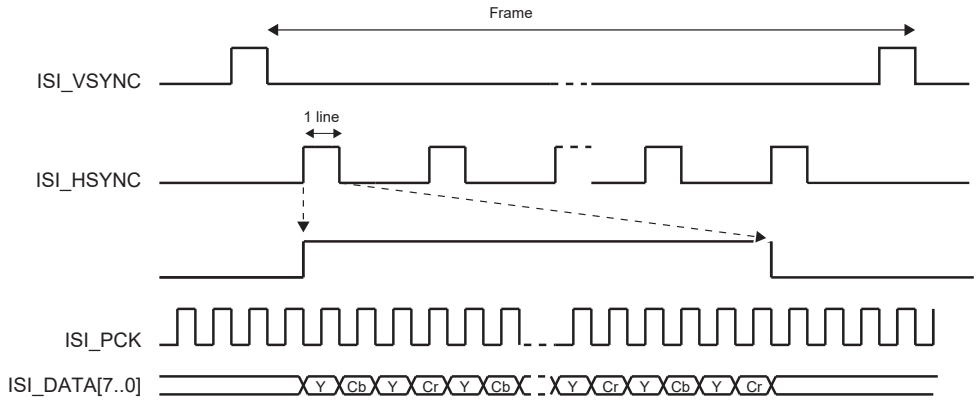
48.5.1 Data Timing

48.5.1.1 VSYNC/HSYNC Data Timing

In the VSYNC/HSYNC synchronization, the valid data is captured with the active edge of the pixel clock (ISI_PCK), after SFD lines of vertical blanking and SLD pixel clock periods delay programmed in the ISI_CR.

The data timing using horizontal and vertical synchronization are shown in the following figure.

Figure 48-3. HSYNC and VSYNC Synchronization



48.5.1.2 SAV/EAV Data Timing

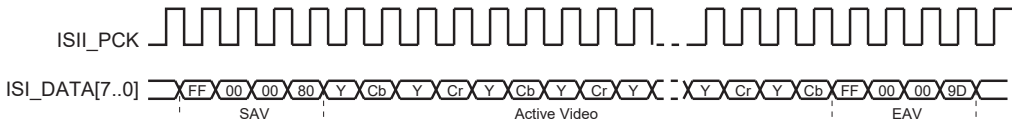
The ITU-RBT.656-4 standard defines the functional timing for an 8-bit wide interface.

Downloaded from Arrow.com.

There are two timing reference signals, one at the beginning of each video data block SAV (0xFF000080) and one at the end of each video data block EAV (0xFF00009D). Only data sent between EAV and SAV is captured. Horizontal blanking and vertical blanking are ignored. Use of the SAV and EAV synchronization eliminates the ISI_VSYNC and ISI_HSYNC signals from the interface, thereby reducing the pin count. In order to retrieve both frame and line synchronization properly, at least one line of vertical blanking is mandatory.

The data timing using EAV/SAV sequence synchronization are shown in the following figure.

Figure 48-4. SAV and EAV Sequence Synchronization



48.5.2 Data Ordering

The RGB color space format is required for viewing images on a display screen preview, and the YCbCr color space format is required for encoding.

All the sensors do not output the YCbCr or RGB components in the same order. The ISI allows the user to program the same component order as the sensor, reducing software treatments to restore the right format.

Table 48-2. Data Ordering in YCbCr Mode

Mode	Byte 0	Byte 1	Byte 2	Byte 3
Default	Cb(i)	Y(i)	Cr(i)	Y(i+1)
Mode 1	Cr(i)	Y(i)	Cb(i)	Y(i+1)
Mode 2	Y(i)	Cb(i)	Y(i+1)	Cr(i)
Mode 3	Y(i)	Cr(i)	Y(i+1)	Cb(i)

Table 48-3. RGB Format in Default Mode, RGB_CFG = 00, No Swap

Mode	Byte	D7	D6	D5	D4	D3	D2	D1	D0
RGB 8:8:8	Byte 0	R7(i)	R6(i)	R5(i)	R4(i)	R3(i)	R2(i)	R1(i)	R0(i)
	Byte 1	G7(i)	G6(i)	G5(i)	G4(i)	G3(i)	G2(i)	G1(i)	G0(i)
	Byte 2	B7(i)	B6(i)	B5(i)	B4(i)	B3(i)	B2(i)	B1(i)	B0(i)
	Byte 3	R7(i+1)	R6(i+1)	R5(i+1)	R4(i+1)	R3(i+1)	R2(i+1)	R1(i+1)	R0(i+1)
RGB 5:6:5	Byte 0	R4(i)	R3(i)	R2(i)	R1(i)	R0(i)	G5(i)	G4(i)	G3(i)
	Byte 1	G2(i)	G1(i)	G0(i)	B4(i)	B3(i)	B2(i)	B1(i)	B0(i)
	Byte 2	R4(i+1)	R3(i+1)	R2(i+1)	R1(i+1)	R0(i+1)	G5(i+1)	G4(i+1)	G3(i+1)
	Byte 3	G2(i+1)	G1(i+1)	G0(i+1)	B4(i+1)	B3(i+1)	B2(i+1)	B1(i+1)	B0(i+1)

Table 48-4. RGB Format, RGB_CFG = 10 (Mode 2), No Swap

Mode	Byte	D7	D6	D5	D4	D3	D2	D1	D0
RGB 5:6:5	Byte 0	G2(i)	G1(i)	G0(i)	R4(i)	R3(i)	R2(i)	R1(i)	R0(i)
	Byte 1	B4(i)	B3(i)	B2(i)	B1(i)	B0(i)	G5(i)	G4(i)	G3(i)
	Byte 2	G2(i+1)	G1(i+1)	G0(i+1)	R4(i+1)	R3(i+1)	R2(i+1)	R1(i+1)	R0(i+1)
	Byte 3	B4(i+1)	B3(i+1)	B2(i+1)	B1(i+1)	B0(i+1)	G5(i+1)	G4(i+1)	G3(i+1)

Downloaded from Arrow.com.

Table 48-5. RGB Format in Default Mode, RGB_CFG = 00, Swap Activated

Mode	Byte	D7	D6	D5	D4	D3	D2	D1	D0
RGB 8:8:8	Byte 0	R0(i)	R1(i)	R2(i)	R3(i)	R4(i)	R5(i)	R6(i)	R7(i)
	Byte 1	G0(i)	G1(i)	G2(i)	G3(i)	G4(i)	G5(i)	G6(i)	G7(i)
	Byte 2	B0(i)	B1(i)	B2(i)	B3(i)	B4(i)	B5(i)	B6(i)	B7(i)
	Byte 3	R0(i+1)	R1(i+1)	R2(i+1)	R3(i+1)	R4(i+1)	R5(i+1)	R6(i+1)	R7(i+1)
RGB 5:6:5	Byte 0	G3(i)	G4(i)	G5(i)	R0(i)	R1(i)	R2(i)	R3(i)	R4(i)
	Byte 1	B0(i)	B1(i)	B2(i)	B3(i)	B4(i)	G0(i)	G1(i)	G2(i)
	Byte 2	G3(i+1)	G4(i+1)	G5(i+1)	R0(i+1)	R1(i+1)	R2(i+1)	R3(i+1)	R4(i+1)
	Byte 3	B0(i+1)	B1(i+1)	B2(i+1)	B3(i+1)	B4(i+1)	G0(i+1)	G1(i+1)	G2(i+1)

The RGB 5:6:5 input format is processed to be displayed as RGB 5:6:5 format, compliant with the 16-bit mode of the LCD controller.

48.5.3 Clocks

The sensor master clock (ISI_MCK) can be generated either by the Advanced Power Management Controller (APMC) through a Programmable Clock output or by an external oscillator connected to the sensor.

None of the sensors embed a power management controller, so providing the clock by the APMC is a simple and efficient way to control power consumption of the system.

Care must be taken when programming the system clock. The ISI has two clock domains, the sensor master clock and the pixel clock provided by sensor. The two clock domains are not synchronized, but the sensor master clock must be faster than the pixel clock.

48.5.4 Preview Path

48.5.4.1 Scaling, Decimation (Subsampling)

This module resizes captured 8-bit color sensor images to fit the LCD display format. The resize module performs only downscaling. The same ratio is applied for both horizontal and vertical resize, then a fractional decimation algorithm is applied.

The decimation factor is a multiple of 1/16; values 0 to 15 are forbidden.

Table 48-6. Decimation Factor

Decimation Value	0–15	16	17	18	19	...	124	125	126	127
Decimation Factor	—	1	1.063	1.125	1.188	...	7.750	7.813	7.875	7.938

Table 48-7. Decimation and Scaler Offset Values

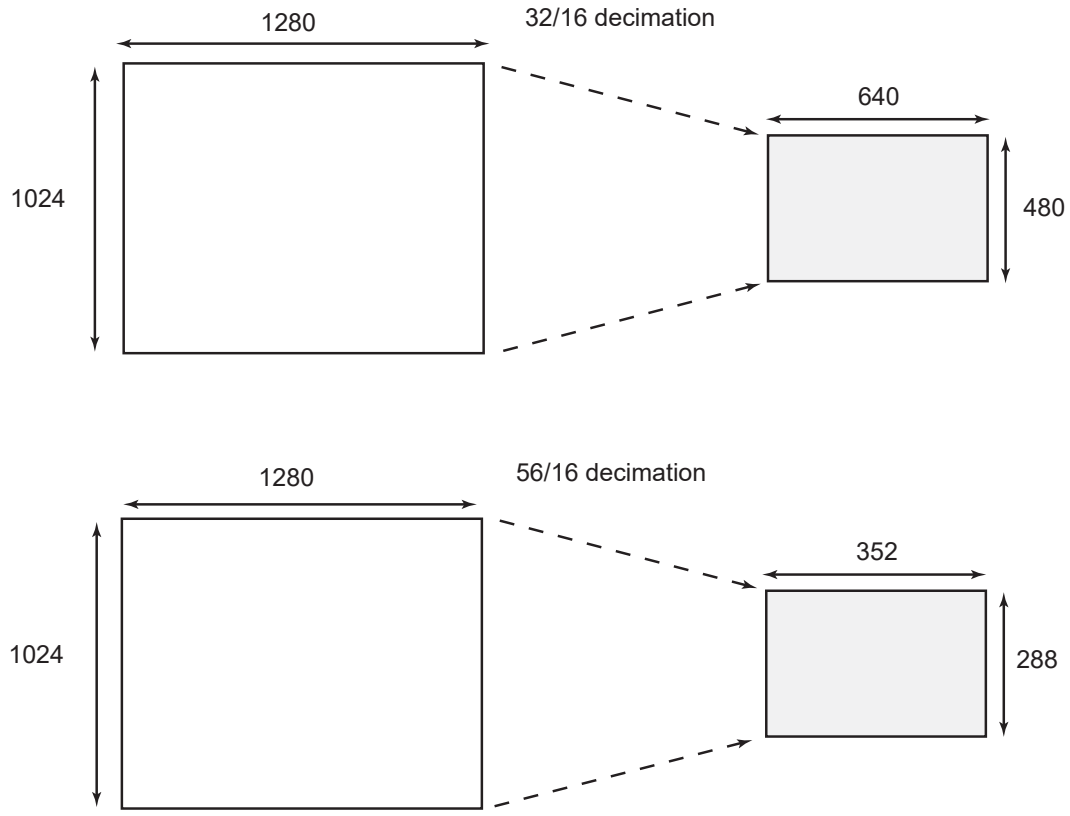
OUTPUT	INPUT	352 × 288	640 × 480	800 × 600	1280 × 1024	1600 × 1200	2048 × 1536
VGA 640 × 480	F	—	16	20	32	40	51
QVGA 320 × 240	F	16	32	40	64	80	102
CIF 352 × 288	F	16	26	33	56	66	85
QCIF 176 × 144	F	32	53	66	113	133	170

Downloaded from Arrow.com.

Example:

- Input 1280 × 1024 Output = 640 × 480
- Hratio = 1280/640 = 2
- Vratio = 1024/480 = 2.1333
- The decimation factor is 2 so 32/16.

Figure 48-5. Resize Examples



48.5.4.2 Color Space Conversion

This module converts YCrCb or YUV pixels to RGB color space. Clipping is performed to ensure that the samples value do not exceed the allowable range. The conversion matrix is defined below and is fully programmable:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} C_0 & 0 & C_1 \\ C_0 & -C_2 & -C_3 \\ C_0 & C_4 & 0 \end{bmatrix} \times \begin{bmatrix} Y - Y_{\text{off}} \\ C_b - C_{\text{boff}} \\ C_r - C_{\text{roff}} \end{bmatrix}$$

Example of programmable value to convert YCrCb to RGB:

$$\begin{cases} R = 1.164 \cdot (Y - 16) + 1.596 \cdot (C_r - 128) \\ G = 1.164 \cdot (Y - 16) - 0.813 \cdot (C_r - 128) - 0.392 \cdot (C_b - 128) \\ B = 1.164 \cdot (Y - 16) + 2.107 \cdot (C_b - 128) \end{cases}$$

An example of programmable value to convert from YUV to RGB:

$$\begin{cases} R = Y + 1.596 \cdot V \\ G = Y - 0.394 \cdot U - 0.436 \cdot V \\ B = Y + 2.032 \cdot U \end{cases}$$

Downloaded from Arrow.com.

48.5.4.3 Memory Interface

48.5.4.3.1 RGB Mode

The preview datapath contains a data formatter that converts 8:8:8 pixel to RGB 5:6:5 format compliant with the 16-bit format of the LCD controller. In general, when converting from a color channel with more bits to one with fewer bits, the formatter module discards the lower-order bits.

For example, converting from RGB 8:8:8 to RGB 5:6:5, the formatter module discards the three LSBs from the red and blue channels, and two LSBs from the green channel.

48.5.4.3.2 12-bit Grayscale Mode

ISI_DATA[11:0] is the physical interface to the ISI. These bits are sampled and written to memory.

When 12-bit Grayscale mode is enabled, two memory formats are supported:

ISI_CFG2.GS_MODE = 0: two pixels per word

ISI_CFG2.GS_MODE = 1: one pixel per word

The following tables illustrate the Grayscale memory mapping configuration for 12-bit data for both formats.

Table 48-8. ISI_CFG2.GS_MODE = 0 (two pixels per word), with ISI_CFG1.GRAYLE = 0

Bit	31	30	29	28	27	26	25	24
	Pixel 0 [11:4]							
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	Pixel 0 [3:0]				–	–	–	–
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	Pixel 1 [11:4]							
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Pixel 1 [3:0]				–	–	–	–

Table 48-9. ISI_CFG2.GS_MODE = 0 (two pixels per word), with ISI_CFG1.GRAYLE = 1

Bit	31	30	29	28	27	26	25	24
	Pixel 1 [11:4]							
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	Pixel 1 [3:0]				–	–	–	–
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	Pixel 0 [11:4]							
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Pixel 0 [3:0]				–	–	–	–

Table 48-10. ISI_CFG2.GS_MODE = 1 (one pixel per word)

Bit	31	30	29	28	27	26	25	24
	Pixel 0 [11:4]							
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	Pixel 0 [3:0]				–	–	–	–
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	–	–	–	–	–	–	–	–

Downloaded from Arrow.com.

Bit	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–

48.5.4.3.3 8-bit Grayscale Mode

For 8-bit Grayscale mode, ISI_DATA[7:0] on the 12-bit data bus is the physical interface to the ISI. These bits are sampled and written to memory.

To enable 8-bit Grayscale mode, configure ISI_CFG2 as follows:

- Clear ISI_CFG2.GRAYSCALE.
- Clear ISI_CFG2.RGB_SWAP.
- Clear ISI_CFG2.COL_SPACE.
- Configure the field ISI_CFG2.YCC_SWAP to value 0.
- Configure the field ISI_CFG2.IM_VSIZE with the vertical resolution of the image minus 1.
- Configure the field ISI_CFG2.IM_HSIZE with the horizontal resolution of the image divided by 2. The horizontal resolution must be a multiple of 2.

The codec datapath is used to capture the 8-bit grayscale image. Use the following configuration:

- Set ISI_DMA_C_CTRL.C_FETCH.
- Configure ISI_DMA_C_DSCR.C_DSCR with the descriptor address.
- Write a one to the bit ISI_DMA_CHER.C_CH_EN.

Table 48-11. Memory Mapping for 8-bit Grayscale Mode

Bit	31	30	29	28	27	26	25	24
	Pixel 3							
Bit	23	22	21	20	19	18	17	16
	Pixel 2							
Bit	15	14	13	12	11	10	9	8
	Pixel 1							
Bit	7	6	5	4	3	2	1	0
	Pixel 0							

48.5.4.4 FIFO and DMA Features

Both preview and codec datapaths contain FIFOs. These asynchronous buffers are used to safely transfer formatted pixels from the pixel clock domain to the AHB clock domain. A video arbiter is used to manage FIFO thresholds and triggers a relevant DMA request through the AHB master interface. Thus, depending on the FIFO state, a specified length burst is asserted. Regarding AHB master interface, it supports Scatter DMA mode through linked list operation. This mode of operation improves flexibility of image buffer location and allows the user to allocate two or more frame buffers. The destination frame buffers are defined by a series of Frame Buffer Descriptors (FBD). Each FBD controls the transfer of one entire frame and then optionally loads a further FBD to switch the DMA operation at another frame buffer address. The FBD is defined by a series of three words. The first word defines the current frame buffer address (named DMA_X_ADDR register), the second defines control information (named DMA_X_CTRL register) and the third defines the next descriptor address (named DMA_X_DSCR). DMA Transfer mode with linked list support is available for both codec and preview datapaths. The data to be transferred described by an FBD requires several burst accesses. In the following example, the use of two ping-pong frame buffers is described.

Example:

The first FBD, stored at address 0x00030000, defines the location of the first frame buffer. This address is programmed in the ISI user interface DMA_P_DSCR. To enable the descriptor fetch operation, the value 0x00000001 must be written to the DMA_P_CTRL register. LLI_0 and LLI_1 are the two descriptors of the linked list.

Destination address: frame buffer ID0 0x02A000 (LLI_0.DMA_P_ADDR)

Downloaded from Arrow.com.

Transfer 0 Control Information, fetch and writeback: 0x00000003 (LLI_0.DMA_P_CTRL)

Next FBD address: 0x00030010 (LLI_0.DMA_P_DSCR)

The second FBD, stored at address 0x00030010, defines the location of the second frame buffer.

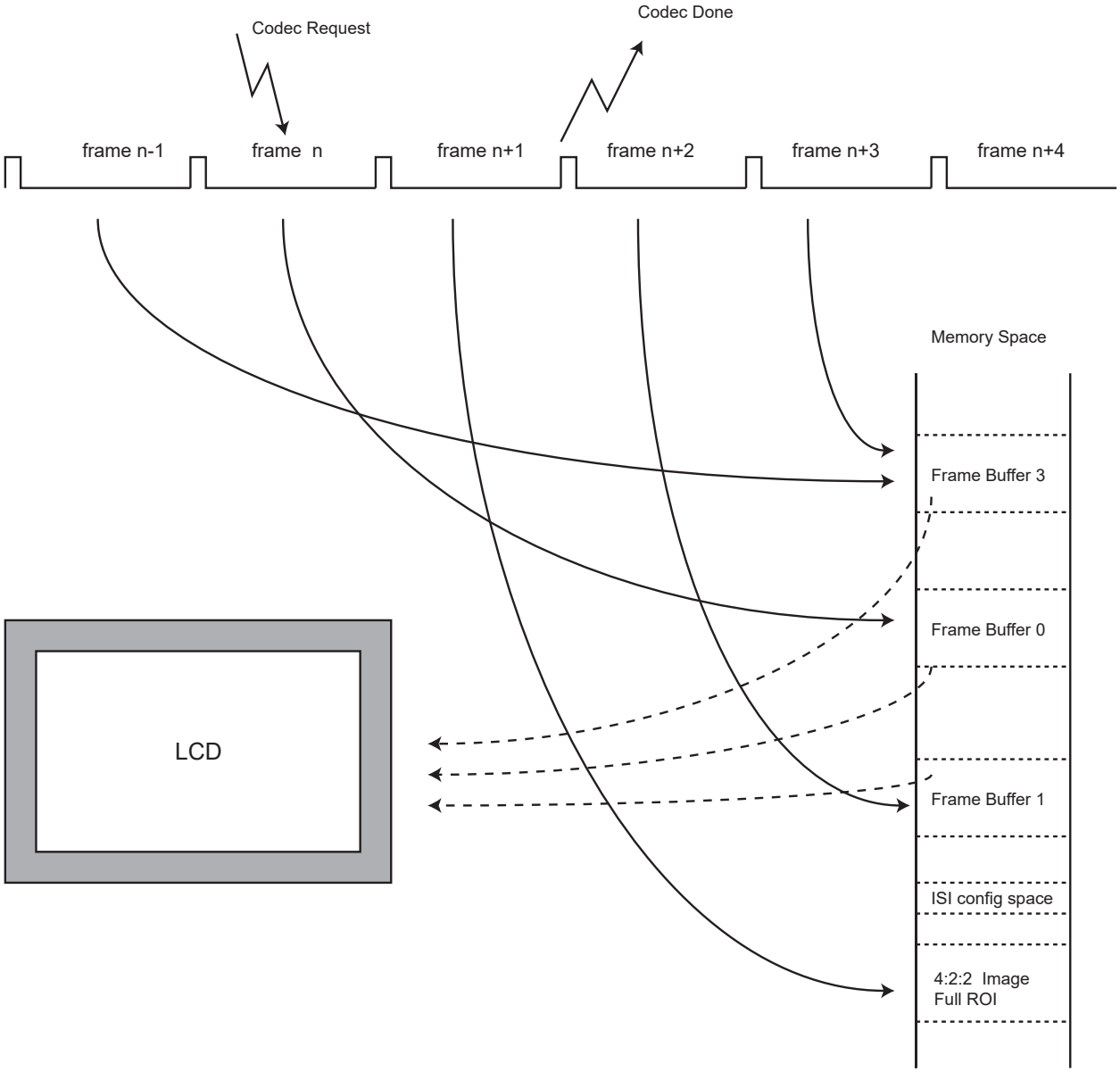
Destination address: frame buffer ID1 0x0003A000 (LLI_1.DMA_P_ADDR)

Transfer 1 Control information fetch and writeback: 0x00000003 (LLI_1.DMA_P_CTRL)

The third FBD address: 0x00030000, wrapping to first FBD (LLI_1.DMA_P_DSCR)

Using this technique, several frame buffers can be configured through the linked list. The following figure illustrates a typical three-frame buffer application. Frame n is mapped to frame buffer 0, frame n+1 is mapped to frame buffer 1, frame n+2 is mapped to frame buffer 2 and further frames wrap. A codec request occurs, and the full-size 4:2:2 encoded frame is stored in a dedicated memory space.

Figure 48-6. Three Frame Buffers Application and Memory Mapping



Downloaded from Arrow.com.

48.5.5 Codec Path

48.5.5.1 Color Space Conversion

Depending on user selection, this module can be bypassed so that input YCrCb stream is directly connected to the format converter module. If the RGB input stream is selected, this module converts RGB to YCrCb color space with the formulas given below:

$$\begin{bmatrix} Y \\ C_r \\ C_b \end{bmatrix} = \begin{bmatrix} C_0 & C_1 & C_2 \\ C_3 & -C_4 & -C_5 \\ -C_6 & -C_7 & C_8 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} Y_{\text{off}} \\ \text{Cr}_{\text{off}} \\ \text{Cb}_{\text{off}} \end{bmatrix}$$

An example of coefficients is given below:

$$\begin{cases} Y = 0.257 \cdot R + 0.504 \cdot G + 0.098 \cdot B + 16 \\ C_r = 0.439 \cdot R - 0.368 \cdot G - 0.071 \cdot B + 128 \\ C_b = -0.148 \cdot R - 0.291 \cdot G + 0.439 \cdot B + 128 \end{cases}$$

48.5.5.2 Memory Interface

Dedicated FIFOs are used to support packed memory mapping. YCrCb pixel components are sent in a single 32-bit word in a contiguous space (packed). Data is stored in the order of natural scan lines. Planar mode is not supported.

48.5.5.3 DMA Features

Like preview datapath, codec datapath DMA mode uses linked list operation.

Downloaded from Arrow.com.

48.6 Register Summary

Note: Several parts of the ISI controller use the pixel clock provided by the image sensor (ISI_PCK). Thus the user must first program the image sensor to provide this clock (ISI_PCK) before programming the Image Sensor Controller.

Offset	Name	Bit Pos.								
0x00	ISI_CFG1	31:24	SFD[7:0]							
		23:16	SLD[7:0]							
		15:8		THMASK[1:0]		FULL	DISCR	FRATE[2:0]		
		7:0	CRC_SYNC	EMB_SYNC	GRAYLE	PIXCLK_POL	VSYNC_POL	HSYNC_POL		
0x04	ISI_CFG2	31:24	RGB_CFG[1:0]		YCC_SWAP[1:0]			IM_HSIZE[10:8]		
		23:16	IM_HSIZE[7:0]							
		15:8	COL_SPACE	RGB_SWAP	GRAYSCALE	RGB_MODE	GS_MODE	IM_VSIZE[10:8]		
		7:0	IM_VSIZE[7:0]							
0x08	ISI_PSIZE	31:24							PREV_HSIZE[9:8]	
		23:16	PREV_HSIZE[7:0]							
		15:8							PREV_VSIZE[9:8]	
		7:0	PREV_VSIZE[7:0]							
0x0C	ISI_PDECF	31:24								
		23:16								
		15:8								
		7:0	DEC_FACTOR[7:0]							
0x10	ISI_Y2R_SET0	31:24	C3[7:0]							
		23:16	C2[7:0]							
		15:8	C1[7:0]							
		7:0	C0[7:0]							
0x14	ISI_Y2R_SET1	31:24								
		23:16								
		15:8		Cboff	Croff	Yoff				C4[8]
		7:0	C4[7:0]							
0x18	ISI_R2Y_SET0	31:24								Roff
		23:16		C2[6:0]						
		15:8		C1[6:0]						
		7:0		C0[6:0]						
0x1C	ISI_R2Y_SET1	31:24								Goff
		23:16		C5[6:0]						
		15:8		C4[6:0]						
		7:0		C3[6:0]						
0x20	ISI_R2Y_SET2	31:24								Boff
		23:16		C8[6:0]						
		15:8		C7[6:0]						
		7:0		C6[6:0]						
0x24	ISI_CR	31:24								
		23:16								
		15:8								ISI_CDC
		7:0						ISI_SRST	ISI_DIS	ISI_EN
0x28	ISI_SR	31:24					FR_OVR	CRC_ERR	C_OVR	P_OVR
		23:16					SIP		CXFR_DONE	PXFR_DONE
		15:8						VSYNC		CDC_PND
		7:0						SRST	DIS_DONE	ENABLE
0x2C	ISI_IER	31:24					FR_OVR	CRC_ERR	C_OVR	P_OVR
		23:16							CXFR_DONE	PXFR_DONE
		15:8						VSYNC		
		7:0						SRST	DIS_DONE	
0x30	ISI_IDR	31:24					FR_OVR	CRC_ERR	C_OVR	P_OVR
		23:16							CXFR_DONE	PXFR_DONE
		15:8						VSYNC		
		7:0						SRST	DIS_DONE	

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x34	ISI_IMR	31:24					FR_OVR	CRC_ERR	C_OVR	P_OVR
		23:16							CXFR_DONE	PXFR_DONE
		15:8						VSYNC		
		7:0						SRST	DIS_DONE	
0x38	ISI_DMA_CHER	31:24								
		23:16								
		15:8								
		7:0							C_CH_EN	P_CH_EN
0x3C	ISI_DMA_CHDR	31:24								
		23:16								
		15:8								
		7:0							C_CH_DIS	P_CH_DIS
0x40	ISI_DMA_CHSR	31:24								
		23:16								
		15:8								
		7:0							C_CH_S	P_CH_S
0x44	ISI_DMA_P_ADDR	31:24	P_ADDR[29:22]							
		23:16	P_ADDR[21:14]							
		15:8	P_ADDR[13:6]							
		7:0	P_ADDR[5:0]							
0x48	ISI_DMA_P_CTRL	31:24								
		23:16								
		15:8								
		7:0					P_DONE	P_IEN	P_WB	P_FETCH
0x4C	ISI_DMA_P_DSCR	31:24	P_DSCR[29:22]							
		23:16	P_DSCR[21:14]							
		15:8	P_DSCR[13:6]							
		7:0	P_DSCR[5:0]							
0x50	ISI_DMA_C_ADDR	31:24	C_ADDR[29:22]							
		23:16	C_ADDR[21:14]							
		15:8	C_ADDR[13:6]							
		7:0	C_ADDR[5:0]							
0x54	ISI_DMA_C_CTRL	31:24								
		23:16								
		15:8								
		7:0					C_DONE	C_IEN	C_WB	C_FETCH
0x58	ISI_DMA_C_DSCR	31:24	C_DSCR[29:22]							
		23:16	C_DSCR[21:14]							
		15:8	C_DSCR[13:6]							
		7:0	C_DSCR[5:0]							
0x5C ... 0xE3	Reserved									
0xE4	ISI_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0xE8	ISI_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

Downloaded from Arrow.com.

48.6.1 ISI Configuration 1 Register

Name: ISI_CFG1
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	SFD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SLD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		THMASK[1:0]		FULL	DISCR	FRATE[2:0]		
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CRC_SYNC	EMB_SYNC	GRAYLE	PIXCLK_POL	VSYNC_POL	HSYNC_POL		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:24 – SFD[7:0] Start of Frame Delay
SFD lines are skipped at the beginning of the frame.

Bits 23:16 – SLD[7:0] Start of Line Delay
SLD pixel clock periods to wait before the beginning of a line.

Bits 14:13 – THMASK[1:0] Threshold Mask

Value	Name	Description
0	BEATS_4	Only 4 beats AHB burst allowed
1	BEATS_8	Only 4 and 8 beats AHB burst allowed
2	BEATS_16	4, 8 and 16 beats AHB burst allowed

Bit 12 – FULL Full Mode is Allowed

Value	Description
0	The codec frame is transferred to memory when an available frame slot is detected.
1	Both preview and codec DMA channels are operating simultaneously.

Bit 11 – DISCR Disable Codec Request

Value	Description
0	Codec datapath DMA interface requires a request to restart.
1	Codec datapath DMA automatically restarts.

Bits 10:8 – FRATE[2:0] Frame Rate [0..7]

Value	Description
0	All the frames are captured, else one frame every FRATE + 1 is captured.

Bit 7 – CRC_SYNC Embedded Synchronization Correction

Downloaded from Arrow.com.

Value	Description
0	No CRC correction is performed on embedded synchronization.
1	CRC correction is performed. If the correction is not possible, the current frame is discarded and the CRC_ERR bit is set in the ISI_SR.

Bit 6 – EMB_SYNC Embedded Synchronization

Value	Description
0	Synchronization by HSYNC, VSYNC.
1	Synchronization by embedded synchronization sequence SAV/EAV.

Bit 5 – GRAYLE Grayscale Little Endian

See the tables ISI_CFG2.GS_MODE = 0 (two pixels per word), with ISI_CFG1.GRAYLE = 0 and ISI_CFG2.GS_MODE = 0 (two pixels per word), with ISI_CFG1.GRAYLE = 1 for details.

Value	Description
0	The two pixels are represented in big-endian format within a 32-bit register.
1	The two pixels are represented in little-endian format within a 32-bit register.

Bit 4 – PIXCLK_POL Pixel Clock Polarity

Value	Description
0	Data is sampled on rising edge of pixel clock.
1	Data is sampled on falling edge of pixel clock.

Bit 3 – VSYNC_POL Vertical Synchronization Polarity

Value	Description
0	VSYNC active high.
1	VSYNC active low.

Bit 2 – HSYNC_POL Horizontal Synchronization Polarity

Value	Description
0	HSYNC active high.
1	HSYNC active low.

Downloaded from Arrow.com.

48.6.2 ISI Configuration 2 Register

Name: ISI_CFG2
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	RGB_CFG[1:0]		YCC_SWAP[1:0]			IM_HSIZE[10:8]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
	IM_HSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COL_SPACE	RGB_SWAP	GRAYSCALE	RGB_MODE	GS_MODE	IM_VSIZE[10:8]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IM_VSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:30 – RGB_CFG[1:0] RGB Pixel Mapping Configuration
Defines RGB pattern when RGB_MODE is set to 1.
If RGB_MODE is set to RGB 8:8:8, then RGB_CFG = 0 implies RGB color sequence, else it implies BGR color sequence.

Value	Name	Description
0	DEFAULT	Byte 0 R/G(MSB) Byte 1 G(LSB)/B Byte 2 R/G(MSB) Byte 3 G(LSB)/B
1	MODE1	Byte 0 B/G(MSB) Byte 1 G(LSB)/R Byte 2 B/G(MSB) Byte 3 G(LSB)/R
2	MODE2	Byte 0 G(LSB)/R Byte 1 B/G(MSB) Byte 2 G(LSB)/R Byte 3 B/G(MSB)
3	MODE3	Byte 0 G(LSB)/B Byte 1 R/G(MSB) Byte 2 G(LSB)/B Byte 3 R/G(MSB)

Downloaded from Arrow.com.

Bits 29:28 – YCC_SWAP[1:0] YCrCb Format Swap Mode
Defines the YCC image data.

Value	Name	Description
0	DEFAULT	Byte 0 Cb(i) Byte 1 Y(i) Byte 2 Cr(i) Byte 3 Y(i+1)
1	MODE1	Byte 0 Cr(i) Byte 1 Y(i) Byte 2 Cb(i) Byte 3 Y(i+1)
2	MODE2	Byte 0 Y(i) Byte 1 Cb(i) Byte 2 Y(i+1) Byte 3 Cr(i)
3	MODE3	Byte 0 Y(i) Byte 1 Cr(i) Byte 2 Y(i+1) Byte 3 Cb(i)

Bits 26:16 – IM_HSIZE[10:0] Horizontal Size of the Image Sensor [0..2047]
If 8-bit Grayscale mode is enabled, IM_HSIZE = (Horizontal size/2) - 1.
Else IM_HSIZE = Horizontal size - 1.

Bit 15 – COL_SPACE Color Space for the Image Data

Value	Description
0	YCbCr.
1	RGB.

Bit 14 – RGB_SWAP RGB Format Swap Mode
The RGB_SWAP has no effect when Grayscale mode is enabled.

Value	Description
0	D7 → R7.
1	D0 → R7.

Bit 13 – GRAYSCALE Grayscale Mode Format Enable

Value	Description
0	Grayscale mode is disabled.
1	Input image is assumed to be grayscale-coded.

Bit 12 – RGB_MODE RGB Input Mode

Value	Description
0	RGB 8:8:8 24 bits.
1	RGB 5:6:5 16 bits.

Bit 11 – GS_MODE Grayscale Pixel Format Mode

Value	Description
0	2 pixels per word.
1	1 pixel per word.

Downloaded from Arrow.com.

Bits 10:0 – IM_VSIZE[10:0] Vertical Size of the Image Sensor [0..2047]
IM_VSIZE = Vertical size - 1

Downloaded from Arrow.com.

48.6.3 ISI Preview Size Register

Name: ISI_PSIZE
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							PREV_HSIZE[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	PREV_HSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
							PREV_VSIZE[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	PREV_VSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 25:16 – PREV_HSIZE[9:0] Horizontal Size for the Preview Path
PREV_HSIZE = Horizontal Preview size - 1 (640 max only in RGB mode).

Bits 9:0 – PREV_VSIZE[9:0] Vertical Size for the Preview Path
PREV_VSIZE = Vertical Preview size - 1 (480 max only in RGB mode).

Downloaded from Arrow.com.

48.6.4 ISI Preview Decimation Factor Register

Name: ISI_PDECF
Offset: 0x0C
Reset: 0x00000010
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DEC_FACTOR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	0	0	0	0

Bits 7:0 – DEC_FACTOR[7:0] Decimation Factor
DEC_FACTOR is 8-bit width, range is from 16 to 255. Values from 0 to 16 do not perform any decimation.

Downloaded from Arrow.com.

48.6.5 ISI Color Space Conversion YCrCb to RGB Set 0 Register

Name: ISI_Y2R_SET0
Offset: 0x10
Reset: 0x6832CC95
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	C3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	0	1	0	0	0

Bit	23	22	21	20	19	18	17	16
	C2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	0	0	1	0

Bit	15	14	13	12	11	10	9	8
	C1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	0	0	1	1	0	0

Bit	7	6	5	4	3	2	1	0
	C0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	1	0	1	0	1

Bits 31:24 – C3[7:0] Color Space Conversion Matrix Coefficient C3
C3 element default step is 1/128, ranges from 0 to 1.9921875.

Bits 23:16 – C2[7:0] Color Space Conversion Matrix Coefficient C2
C2 element default step is 1/128, ranges from 0 to 1.9921875.

Bits 15:8 – C1[7:0] Color Space Conversion Matrix Coefficient C1
C1 element default step is 1/128, ranges from 0 to 1.9921875.

Bits 7:0 – C0[7:0] Color Space Conversion Matrix Coefficient C0
C0 element default step is 1/128, ranges from 0 to 1.9921875.

Downloaded from Arrow.com.

48.6.6 ISI Color Space Conversion YCrCb to RGB Set 1 Register

Name: ISI_Y2R_SET1
Offset: 0x14
Reset: 0x00007102
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		Cboff	Croff	Yoff				C4[8]
Access		R/W	R/W	R/W				R/W
Reset		1	1	1				1

Bit	7	6	5	4	3	2	1	0
	C4[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	0

Bit 14 – Cboff Color Space Conversion Blue Chrominance Default Offset

Value	Description
0	No offset.
1	Offset = 16.

Bit 13 – Croff Color Space Conversion Red Chrominance Default Offset

Value	Description
0	No offset.
1	Offset = 16.

Bit 12 – Yoff Color Space Conversion Luminance Default Offset

Value	Description
0	No offset.
1	Offset = 128.

Bits 8:0 – C4[8:0] Color Space Conversion Matrix Coefficient C4
C4 element default step is 1/128, ranges from 0 to 3.9921875.

Downloaded from Arrow.com.

48.6.7 ISI Color Space Conversion RGB to YCrCb Set 0 Register

Name: ISI_R2Y_SET0
Offset: 0x18
Reset: 0x01324145
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
								Roff
Access								R/W
Reset								1

Bit	23	22	21	20	19	18	17	16
		C2[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	1	1	0	0	1	0

Bit	15	14	13	12	11	10	9	8
		C1[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		1	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
		C0[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		1	0	0	0	1	0	1

Bit 24 – Roff Color Space Conversion Red Component Offset

Value	Description
0	No offset
1	Offset = 16

Bits 22:16 – C2[6:0] Color Space Conversion Matrix Coefficient C2
C2 element default step is 1/512, from 0 to 0.2480468875.

Bits 14:8 – C1[6:0] Color Space Conversion Matrix Coefficient C1
C1 element default step is 1/128, from 0 to 0.9921875.

Bits 6:0 – C0[6:0] Color Space Conversion Matrix Coefficient C0
C0 element default step is 1/256, from 0 to 0.49609375.

Downloaded from Arrow.com.

48.6.8 ISI Color Space Conversion RGB to YCrCb Set 1 Register

Name: ISI_R2Y_SET1
Offset: 0x1C
Reset: 0x01245E38
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
								Goff
Access								R/W
Reset								1

Bit	23	22	21	20	19	18	17	16
		C5[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	1	0	0	1	0	0

Bit	15	14	13	12	11	10	9	8
		C4[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		1	0	1	1	1	1	0

Bit	7	6	5	4	3	2	1	0
		C3[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	1	1	1	0	0	0

Bit 24 – Goff Color Space Conversion Green Component Offset

Value	Description
0	No offset.
1	Offset = 128.

Bits 22:16 – C5[6:0] Color Space Conversion Matrix Coefficient C5
C1 element default step is 1/512, ranges from 0 to 0.2480468875.

Bits 14:8 – C4[6:0] Color Space Conversion Matrix Coefficient C4
C1 element default step is 1/256, ranges from 0 to 0.49609375.

Bits 6:0 – C3[6:0] Color Space Conversion Matrix Coefficient C3
C0 element default step is 1/128, ranges from 0 to 0.9921875.

Downloaded from Arrow.com.

48.6.9 ISI Color Space Conversion RGB to YCrCb Set 2 Register

Name: ISI_R2Y_SET2
Offset: 0x20
Reset: 0x01384A4B
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
								Boff
Access								R/W
Reset								1

Bit	23	22	21	20	19	18	17	16
		C8[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	1	1	1	0	0	0

Bit	15	14	13	12	11	10	9	8
		C7[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		1	0	0	1	0	1	0

Bit	7	6	5	4	3	2	1	0
		C6[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		1	0	0	1	0	1	1

Bit 24 – Boff Color Space Conversion Blue Component Offset

Value	Description
0	No offset.
1	Offset = 128.

Bits 22:16 – C8[6:0] Color Space Conversion Matrix Coefficient C8
C8 element default step is 1/128, ranges from 0 to 0.9921875.

Bits 14:8 – C7[6:0] Color Space Conversion Matrix Coefficient C7
C7 element default step is 1/256, ranges from 0 to 0.49609375.

Bits 6:0 – C6[6:0] Color Space Conversion Matrix Coefficient C6
C6 element default step is 1/512, ranges from 0 to 0.2480468875.

Downloaded from Arrow.com.

48.6.10 ISI Control Register

Name: ISI_CR
Offset: 0x24
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								ISI_CDC
Reset								W
								–

Bit	7	6	5	4	3	2	1	0
Access						ISI_SRST	ISI_DIS	ISI_EN
Reset						W	W	W
						–	–	–

Bit 8 – ISI_CDC ISI Codec Request
Write a one to this bit to enable the codec datapath and capture a full resolution frame. A new request cannot be taken into account while CDC_PND bit is active in the ISI_SR.

Bit 2 – ISI_SRST ISI Software Reset Request
Write a one to this bit to request a software reset of the module. Software must poll the SRST bit in the ISI_SR to verify that the software request command has terminated.

Bit 1 – ISI_DIS ISI Module Disable Request
Write a one to this bit to disable the module. If both ISI_EN and ISI_DIS are asserted at the same time, the disable request is not taken into account. Software must poll the DIS_DONE bit in the ISI_SR to verify that the command has successfully completed.

Bit 0 – ISI_EN ISI Module Enable Request
Write a one to this bit to enable the module. Software must poll the ENABLE bit in the ISI_SR to verify that the command has successfully completed.

Downloaded from Arrow.com.

48.6.11 ISI Status Register

Name: ISI_SR
Offset: 0x28
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
					FR_OVR	CRC_ERR	C_OVR	P_OVR
Access					R	R	R	R
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					SIP		CXFR_DONE	PXFR_DONE
Access					R		R	R
Reset					0		0	0

Bit	15	14	13	12	11	10	9	8
						VSYNC		CDC_PND
Access						R		R
Reset						0		0

Bit	7	6	5	4	3	2	1	0
						SRST	DIS_DONE	ENABLE
Access						R	R	R
Reset						0	0	0

Bit 27 – FR_OVR Frame Rate Overrun (cleared on read)

Value	Description
0	No frame overrun
1	Frame overrun. The current frame is being skipped because a vsync signal has been detected while flushing FIFOs since the last read of ISI_SR.

Bit 26 – CRC_ERR CRC Synchronization Error (cleared on read)

Value	Description
0	No CRC error in the embedded synchronization frame (SAV/EAV)
1	Embedded Synchronization Correction is enabled (CRC_SYNC bit is set) in the ISI_CR and an error has been detected and not corrected since the last read of ISI_SR. The frame is discarded and the ISI waits for a new one.

Bit 25 – C_OVR Codec Datapath Overflow (cleared on read)

Value	Description
0	No overflow
1	An overrun condition has occurred in input FIFO on the codec path. The overrun happens when the FIFO is full and an attempt is made to write a new sample to the FIFO since the last read of ISI_SR.

Bit 24 – P_OVR Preview Datapath Overflow (cleared on read)

Value	Description
0	No overflow
1	An overrun condition has occurred in input FIFO on the preview path. The overrun happens when the FIFO is full and an attempt is made to write a new sample to the FIFO since the last read of ISI_SR.

Bit 19 – SIP Synchronization in Progress
When the status of the preview or codec DMA channel is modified, a minimum amount of time is required to perform the clock domain synchronization.

Downloaded from Arrow.com.

Value	Description
0	The clock domain synchronization process is terminated.
1	This bit is set when the clock domain synchronization operation occurs. No modification of the channel status is allowed when this bit is set, to guarantee data integrity.

Bit 17 – CXFR_DONE Codec DMA Transfer has Terminated (cleared on read)

Value	Description
0	Codec transfer done not detected.
1	Codec transfer done detected. When set, this bit indicates that the data transfer on the codec channel has completed since the last read of ISI_SR.

Bit 16 – PXFR_DONE Preview DMA Transfer has Terminated (cleared on read)

Value	Description
0	Preview transfer done not detected.
1	Preview transfer done detected. When set, this bit indicates that the data transfer on the preview channel has completed since the last read of ISI_SR.

Bit 10 – VSYNC Vertical Synchronization (cleared on read)

Value	Description
0	Indicates that the vertical synchronization has not been detected since the last read of the ISI_SR.
1	Indicates that a vertical synchronization has been detected since the last read of the ISI_SR.

Bit 8 – CDC_PND Pending Codec Request

Value	Description
0	Indicates that no codec request is pending
1	Indicates that the request has been taken into account but cannot be serviced within the current frame. The operation is postponed to the next frame.

Bit 2 – SRST Module Software Reset Request has Terminated (cleared on read)

Value	Description
0	Indicates that the request is not completed (if a request was issued).
1	Software reset request has completed. This flag is reset after a read operation.

Bit 1 – DIS_DONE Module Disable Request has Terminated (cleared on read)

Value	Description
0	Indicates that the request is not completed (if a request was issued).
1	Disable request has completed. This flag is reset after a read operation.

Bit 0 – ENABLE Module Enable

Value	Description
0	Module is disabled.
1	Module is enabled.

Downloaded from Arrow.com.

48.6.12 ISI Interrupt Enable Register

Name: ISI_IER
Offset: 0x2C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
					FR_OVR	CRC_ERR	C_OVR	P_OVR
Access					W	W	W	W
Reset					–	–	–	–

Bit	23	22	21	20	19	18	17	16
							CXFR_DONE	PXFR_DONE
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
						VSYNC		
Access						W		
Reset						–		

Bit	7	6	5	4	3	2	1	0
						SRST	DIS_DONE	
Access						W	W	
Reset						–	–	

Bit 27 – FR_OVR Frame Rate Overflow Interrupt Enable

Value	Description
0	No effect.
1	Enables the corresponding interrupt.

Bit 26 – CRC_ERR Embedded Synchronization CRC Error Interrupt Enable

Value	Description
0	No effect.
1	Enables the corresponding interrupt.

Bit 25 – C_OVR Codec Datapath Overflow Interrupt Enable

Value	Description
0	No effect.
1	Enables the corresponding interrupt.

Bit 24 – P_OVR Preview Datapath Overflow Interrupt Enable

Value	Description
0	No effect.
1	Enables the corresponding interrupt.

Bit 17 – CXFR_DONE Codec DMA Transfer Done Interrupt Enable

Value	Description
0	No effect.
1	Enables the corresponding interrupt.

Bit 16 – PXFR_DONE Preview DMA Transfer Done Interrupt Enable

Value	Description
0	No effect.

Downloaded from Arrow.com.

Value	Description
1	Enables the corresponding interrupt.

Bit 10 – VSYNC Vertical Synchronization Interrupt Enable

Value	Description
0	No effect.
1	Enables the corresponding interrupt.

Bit 2 – SRST Software Reset Interrupt Enable

Value	Description
0	No effect.
1	Enables the corresponding interrupt.

Bit 1 – DIS_DONE Disable Done Interrupt Enable

Value	Description
0	No effect.
1	Enables the corresponding interrupt.

Downloaded from Arrow.com.

48.6.13 ISI Interrupt Disable Register

Name: ISI_IDR
Offset: 0x30
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
					FR_OVR	CRC_ERR	C_OVR	P_OVR
Access					W	W	W	W
Reset					–	–	–	–

Bit	23	22	21	20	19	18	17	16
							CXFR_DONE	PXFR_DONE
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
						VSYNC		
Access						W		
Reset						–		

Bit	7	6	5	4	3	2	1	0
						SRST	DIS_DONE	
Access						W	W	
Reset						–	–	

Bit 27 – FR_OVR Frame Rate Overflow Interrupt Disable

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

Bit 26 – CRC_ERR Embedded Synchronization CRC Error Interrupt Disable

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

Bit 25 – C_OVR Codec Datapath Overflow Interrupt Disable

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

Bit 24 – P_OVR Preview Datapath Overflow Interrupt Disable

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

Bit 17 – CXFR_DONE Codec DMA Transfer Done Interrupt Disable

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

Bit 16 – PXFR_DONE Preview DMA Transfer Done Interrupt Disable

Value	Description
0	No effect.

Downloaded from Arrow.com.

Value	Description
1	Disables the corresponding interrupt.

Bit 10 – VSYNC Vertical Synchronization Interrupt Disable

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

Bit 2 – SRST Software Reset Interrupt Disable

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

Bit 1 – DIS_DONE Disable Done Interrupt Disable

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

Downloaded from Arrow.com.

48.6.14 ISI Interrupt Mask Register

Name: ISI_IMR
Offset: 0x34
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
					FR_OVR	CRC_ERR	C_OVR	P_OVR
Access					R	R	R	R
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
							CXFR_DONE	PXFR_DONE
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
						VSYNC		
Access						R		
Reset						0		

Bit	7	6	5	4	3	2	1	0
						SRST	DIS_DONE	
Access						R	R	
Reset						0	0	

Bit 27 – FR_OVR Frame Rate Overrun

Value	Description
0	The Frame Rate Overrun interrupt is disabled.
1	The Frame Rate Overrun is enabled.

Bit 26 – CRC_ERR CRC Synchronization Error

Value	Description
0	The CRC Synchronization Error interrupt is disabled.
1	The CRC Synchronization Error interrupt is enabled.

Bit 25 – C_OVR Codec FIFO Overflow

Value	Description
0	The Codec FIFO Overflow interrupt is disabled.
1	The Codec FIFO Overflow interrupt is enabled.

Bit 24 – P_OVR Preview FIFO Overflow

Value	Description
0	The Preview FIFO Overflow interrupt is disabled.
1	The Preview FIFO Overflow interrupt is enabled.

Bit 17 – CXFR_DONE Codec DMA Transfer Completed

Value	Description
0	The Codec DMA Transfer Completed interrupt is disabled.
1	The Codec DMA Transfer Completed interrupt is enabled.

Bit 16 – PXFR_DONE Preview DMA Transfer Completed

Value	Description
0	The Preview DMA Transfer Completed interrupt is disabled.

Downloaded from Arrow.com.

Value	Description
1	The Preview DMA Transfer Completed interrupt is enabled.

Bit 10 – VSYNC Vertical Synchronization

Value	Description
0	The Vertical Synchronization interrupt is disabled.
1	The Vertical Synchronization interrupt is enabled.

Bit 2 – SRST Software Reset Completed

Value	Description
0	The Software Reset Completed interrupt is disabled.
1	The Software Reset Completed interrupt is enabled.

Bit 1 – DIS_DONE Module Disable Operation Completed

Value	Description
0	The Module Disable Operation Completed interrupt is disabled.
1	The Module Disable Operation Completed interrupt is enabled.

Downloaded from Arrow.com.

48.6.15 DMA Channel Enable Register

Name: ISI_DMA_CHER
Offset: 0x38
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							C_CH_EN	P_CH_EN
Access							W	W
Reset							–	–

Bit 1 – C_CH_EN Codec Channel Enable
Write a one to this bit to enable the codec DMA channel.

Bit 0 – P_CH_EN Preview Channel Enable
Write a one to this bit to enable the preview DMA channel.

Downloaded from Arrow.com.

48.6.16 DMA Channel Disable Register

Name: ISI_DMA_CHDR
Offset: 0x3C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							C_CH_DIS	P_CH_DIS
Access							W	W
Reset							–	–

Bit 1 – C_CH_DIS Codec Channel Disable Request

Value	Description
0	No effect.
1	Disables the channel. Poll C_CH_S in DMA_CHSR to verify that the codec channel status has been successfully modified.

Bit 0 – P_CH_DIS Preview Channel Disable Request

Value	Description
0	No effect.
1	Disables the channel. Poll P_CH_S in DMA_CHSR to verify that the preview channel status has been successfully modified.

Downloaded from Arrow.com.

48.6.17 DMA Channel Status Register

Name: ISI_DMA_CHSR
Offset: 0x40
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							C_CH_S	P_CH_S
Access							R	R
Reset							0	0

Bit 1 – C_CH_S Code DMA Channel Status

Value	Description
0	Indicates that the Codec DMA channel is disabled.
1	Indicates that the Codec DMA channel is enabled.

Bit 0 – P_CH_S Preview DMA Channel Status

Value	Description
0	Indicates that the Preview DMA channel is disabled.
1	Indicates that the Preview DMA channel is enabled.

Downloaded from Arrow.com.

48.6.18 DMA Preview Base Address Register

Name: ISI_DMA_P_ADDR
Offset: 0x44
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	P_ADDR[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P_ADDR[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P_ADDR[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P_ADDR[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – P_ADDR[29:0] Preview Image Base Address
This address is word-aligned.

Downloaded from Arrow.com.

48.6.19 DMA Preview Control Register

Name: ISI_DMA_P_CTRL
Offset: 0x48
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					P_DONE	P_IEN	P_WB	P_FETCH
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – P_DONE Preview Transfer Done
This bit is only updated in the memory.

Value	Description
0	The transfer related to this descriptor has not been performed.
1	The transfer related to this descriptor has completed. This bit is updated in memory at the end of the transfer, when writeback operation is enabled.

Bit 2 – P_IEN Transfer Done Flag Control

Value	Description
0	Preview transfer done flag generation is enabled.
1	Preview transfer done flag generation is disabled.

Bit 1 – P_WB Descriptor Writeback Control Bit

Value	Description
0	Preview channel writeback operation is disabled.
1	Preview channel writeback operation is enabled.

Bit 0 – P_FETCH Descriptor Fetch Control Bit

Value	Description
0	Preview channel fetch operation is disabled.
1	Preview channel fetch operation is enabled.

Downloaded from Arrow.com.

48.6.20 DMA Preview Descriptor Address Register

Name: ISI_DMA_P_DSCR
Offset: 0x4C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	P_DSCR[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P_DSCR[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P_DSCR[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P_DSCR[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – P_DSCR[29:0] Preview Descriptor Base Address
This address is word-aligned.

Downloaded from Arrow.com.

48.6.21 DMA Codec Base Address Register

Name: ISI_DMA_C_ADDR
Offset: 0x50
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	C_ADDR[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	C_ADDR[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	C_ADDR[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	C_ADDR[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – C_ADDR[29:0] Codec Image Base Address
This address is word-aligned.

Downloaded from Arrow.com.

48.6.22 DMA Codec Control Register

Name: ISI_DMA_C_CTRL
Offset: 0x54
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					C_DONE	C_IEN	C_WB	C_FETCH
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – C_DONE Codec Transfer Done
This bit is only updated in the memory.

Value	Description
0	The transfer related to this descriptor has not been performed.
1	The transfer related to this descriptor has completed. This bit is updated in memory at the end of the transfer when writeback operation is enabled.

Bit 2 – C_IEN Transfer Done Flag Control

Value	Description
0	Codec transfer done flag generation is enabled.
1	Codec transfer done flag generation is disabled.

Bit 1 – C_WB Descriptor Writeback Control Bit

Value	Description
0	Codec channel writeback operation is disabled.
1	Codec channel writeback operation is enabled.

Bit 0 – C_FETCH Descriptor Fetch Control Bit

Value	Description
0	Codec channel fetch operation is disabled.
1	Codec channel fetch operation is enabled.

Downloaded from Arrow.com.

48.6.23 DMA Codec Descriptor Address Register

Name: ISI_DMA_C_DSCR
Offset: 0x58
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	C_DSCR[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	C_DSCR[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	C_DSCR[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	C_DSCR[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – C_DSCR[29:0] Codec Descriptor Base Address
This address is word-aligned.

Downloaded from Arrow.com.

48.6.24 ISI Write Protection Mode Register

Name: ISI_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key Password

Value	Name	Description
0x495349	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x495349 (“ISI” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x495349 (“ISI” in ASCII).

Downloaded from Arrow.com.

48.6.25 ISI Write Protection Status Register

Name: ISI_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source

Value	Name
0	No Write Protection Violation occurred since the last read of this register (ISI_WPSR).
1	Write access in ISI_CFG1 while Write Protection was enabled (since the last read).
2	Write access in ISI_CFG2 while Write Protection was enabled (since the last read).
3	Write access in ISI_PSIZE while Write Protection was enabled (since the last read).
4	Write access in ISI_PDECF while Write Protection was enabled (since the last read).
5	Write access in ISI_Y2R_SET0 while Write Protection was enabled (since the last read).
6	Write access in ISI_Y2R_SET1 while Write Protection was enabled (since the last read).
7	Write access in ISI_R2Y_SET0 while Write Protection was enabled (since the last read).
8	Write access in ISI_R2Y_SET1 while Write Protection was enabled (since the last read).
9	Write access in ISI_R2Y_SET2 while Write Protection was enabled (since the last read).

Bit 0 – WPVS Write Protection Violation Status

Value	Name
0	No write protection violation occurred since the last read of ISI_WPSR.
1	A write protection violation has occurred since the last read of the ISI_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

49. Controller Area Network (CAN)

49.1 Description

The CAN controller provides all the features required to implement the serial communication protocol CAN defined by Robert Bosch GmbH, the CAN specification as referred to by ISO/11898A (2.0 Part A and 2.0 Part B) for high speeds and ISO/11519-2 for low speeds. The CAN Controller is able to handle all types of frames (Data, Remote, Error and Overload) and achieves a bitrate of 1 Mbit/s.

CAN controller accesses are made through configuration registers. 8 independent message objects (mailboxes) are implemented.

Any mailbox can be programmed as a reception buffer block (even non-consecutive buffers). For the reception of defined messages, one or several message objects can be masked without participating in the buffer feature. An interrupt is generated when the buffer is full. According to the mailbox configuration, the first message received can be locked in the CAN controller registers until the application acknowledges it, or this message can be discarded by new received messages.

Any mailbox can be programmed for transmission. Several transmission mailboxes can be enabled in the same time. A priority can be defined for each mailbox independently.

An internal 16-bit timer is used to stamp each received and sent message. This timer starts counting as soon as the CAN controller is enabled. This counter can be reset by the application or automatically after a reception in the last mailbox in Time-triggered mode.

The CAN controller offers optimized features to support the Time-Triggered Communication (TTC) protocol.

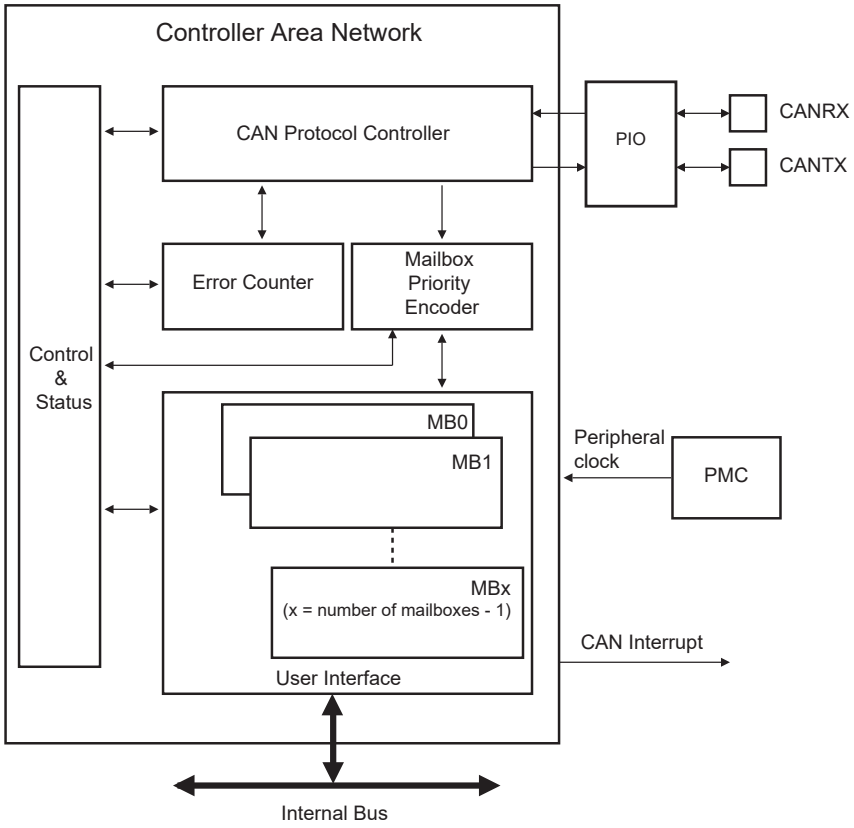
49.2 Embedded Characteristics

- Fully Compliant with CAN 2.0 Part A and 2.0 Part B
- Bit Rates up to 1 Mbit/s
- 8 Object Oriented Mailboxes with the Following Properties:
  - CAN Specification 2.0 Part A or 2.0 Part B Programmable for Each Message
  - Object Configurable in Receive (with Overwrite or Not) or Transmit Modes
  - Independent 29-bit Identifier and Mask Defined for Each Mailbox
  - 32-bit Access to Data Registers for Each Mailbox Data Object
  - Uses a 16-bit Timestamp on Receive and Transmit Messages
  - Hardware Concatenation of ID Masked Bitfields To Speed Up Family ID Processing
- 16-bit Internal Timer for Timestamping and Network Synchronization
- Programmable Reception Buffer Length up to 8 Mailbox Objects
- Priority Management between Transmission Mailboxes
- Autobaud and Listening Mode
- Low-power Mode and Programmable Wake-up on Bus Activity or by the Application
- Data, Remote, Error and Overload Frame Handling
- Register Write Protection

Downloaded from Arrow.com.

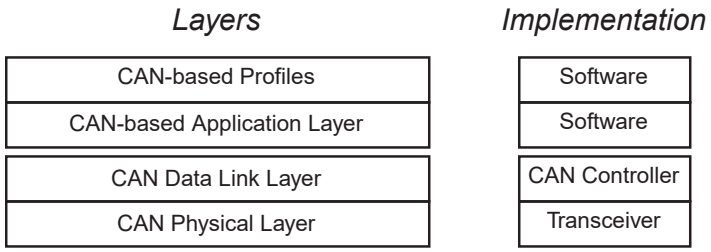
49.3 Block Diagram

Figure 49-1. CAN Block Diagram



49.4 Application Block Diagram

Figure 49-2. Application Block Diagram



49.5 I/O Lines Description

Table 49-1. I/O Lines Description

Name	Description	Type
CANRX	CAN Receive Serial Data	Input
CANTX	CAN Transmit Serial Data	Output

Downloaded from Arrow.com.

49.6 Product Dependencies

49.6.1 I/O Lines

The pins used for interfacing the CAN may be multiplexed with the PIO lines. The programmer must first program the PIO controller to assign the desired CAN pins to their peripheral function. If I/O lines of the CAN are not used by the application, they can be used for other purposes by the PIO Controller.

49.6.2 Power Management

The programmer must first enable the CAN clock in the Power Management Controller (PMC) before using the CAN.

A Low-power mode is defined for the CAN controller. If the application does not require CAN operations, the CAN clock can be stopped when not needed and be restarted later. Before stopping the clock, the CAN Controller must be in Low-power mode to complete the current transfer. After restarting the clock, the application must disable the Low-power mode of the CAN controller.

49.6.3 Interrupt Sources

The CAN interrupt line is connected on one of the internal sources of the interrupt controller. Using the CAN interrupt requires the interrupt controller to be programmed first. Note that it is not recommended to use the CAN interrupt line in edge-sensitive mode.

49.7 CAN Controller Features

49.7.1 CAN Protocol Overview

The Controller Area Network (CAN) is a multi-master serial communication protocol that efficiently supports real-time control with a very high level of security with bit rates up to 1 Mbit/s.

The CAN protocol supports four different frame types:

- Data frames: They carry data from a transmitter node to the receiver nodes. The overall maximum data frame length is 108 bits for a standard frame and 128 bits for an extended frame.
- Remote frames: A destination node can request data from the source by sending a remote frame with an identifier that matches the identifier of the required data frame. The appropriate data source node then sends a data frame as a response to this node request.
- Error frames: An error frame is generated by any node that detects a bus error.
- Overload frames: They provide an extra delay between the preceding and the successive data frames or remote frames.

The CAN controller provides the CPU with full functionality of the CAN protocol V2.0 Part A and V2.0 Part B. It minimizes the CPU load in communication overhead. The Data Link Layer and part of the physical layer are automatically handled by the CAN controller itself.

The CPU reads or writes data or messages via the CAN controller mailboxes. An identifier is assigned to each mailbox. The CAN controller encapsulates or decodes data messages to build or to decode bus data frames. Remote frames, error frames and overload frames are automatically handled by the CAN controller under supervision of the software application.

49.7.2 Mailbox Organization

The CAN module has 8 buffers, also called channels or mailboxes. An identifier that corresponds to the CAN identifier is defined for each active mailbox. Message identifiers can match the standard frame identifier or the extended frame identifier. This identifier is defined for the first time during the CAN initialization, but can be dynamically reconfigured later so that the mailbox can handle a new message family. Several mailboxes can be configured with the same ID.

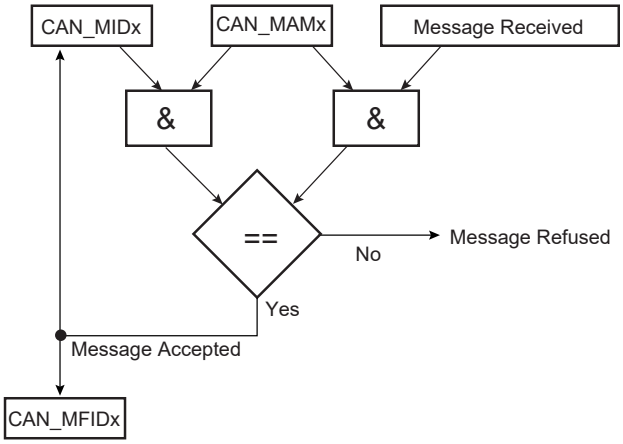
Each mailbox can be configured in receive or in transmit mode independently. The mailbox object type is defined in the MOT field of the CAN_MMRx.

Downloaded from Arrow.com.

49.7.2.1 Message Acceptance Procedure

If the MIDE field in the CAN_MIDx register is set, the mailbox can handle the extended format identifier; otherwise, the mailbox handles the standard format identifier. Once a new message is received, its ID is masked with the CAN_MAMx value and compared with the CAN_MIDx value. If accepted, the message ID is copied to the CAN_MIDx register.

Figure 49-3. Message Acceptance Procedure



If a mailbox is dedicated to receiving several messages (a family of messages) with different IDs, the acceptance mask defined in the CAN_MAMx register must mask the variable part of the ID family. Once a message is received, the application must decode the masked bits in the CAN_MIDx. To speed up the decoding, masked bits are grouped in the family ID register (CAN_MFIDx).

For example, if the following message IDs are handled by the same mailbox:

```
ID0 10100010010000100100000100 0 11 00b
ID1 10100010010000100100000100 0 11 01b
ID2 10100010010000100100000100 0 11 10b
ID3 10100010010000100100000100 0 11 11b
ID4 10100010010000100100000100 1 11 00b
ID5 10100010010000100100000100 1 11 01b
ID6 10100010010000100100000100 1 11 10b
ID7 10100010010000100100000100 1 11 11b
```

The CAN_MIDx and CAN_MAMx of Mailbox x must be initialized to the corresponding values:

```
CAN_MIDx = 001 10100010010000100100000100 x 11 xxb
CAN_MAMx = 001 11111111111111111111111111 0 11 00b
```

If Mailbox x receives a message with ID6, then CAN_MIDx and CAN_MFIDx are set:

```
CAN_MIDx = 001 10100010010000100100000100 1 11 10b
CAN_MFIDx = 00000000000000000000000000000110b
```

If the application associates a handler for each message ID, it may define an array of pointers to functions:

```
void (*pHandler[8])(void);
```

When a message is received, the corresponding handler can be invoked using CAN_MFIDx register and there is no need to check masked bits:

```
unsigned int MFID0_register;
MFID0_register = Get_CAN_MFID0_Register();
// Get_CAN_MFID0_Register() returns the value of the CAN_MFID0 register
pHandler[MFID0_register]();
```


Downloaded from Arrow.com.

49.7.2.2 Receive Mailbox

When the CAN module receives a message, it looks for the first available mailbox with the lowest number and compares the received message ID with the mailbox ID. If such a mailbox is found, then the message is stored in its data registers. Depending on the configuration, the mailbox is disabled as long as the message has not been acknowledged by the application (Receive only), or, if new messages with the same ID are received, then they overwrite the previous ones (Receive with overwrite).

It is also possible to configure a mailbox in Consumer Mode. In this mode, after each transfer request, a remote frame is automatically sent. The first answer received is stored in the corresponding mailbox data registers.

Several mailboxes can be chained to receive a buffer. They must be configured with the same ID in Receive Mode, except for the last one, which can be configured in Receive with Overwrite Mode. The last mailbox can be used to detect a buffer overflow.

Table 49-2. Receive Mailbox Objects

Object Type	Description
Receive	The first message received is stored in mailbox data registers. Data remain available until the next transfer request.
Receive with overwrite	The last message received is stored in mailbox data register. The next message always overwrites the previous one. The application has to check whether a new message has not overwritten the current one while reading the data registers.
Consumer	A remote frame is sent by the mailbox. The answer received is stored in mailbox data register. This extends Receive mailbox features. Data remain available until the next transfer request.

49.7.2.3 Transmit Mailbox

When transmitting a message, the message length and data are written to the transmit mailbox with the correct identifier. For each transmit mailbox, a priority is assigned. The controller automatically sends the message with the highest priority first (set with the field PRIOR in CAN_MMRx).

It is also possible to configure a mailbox in Producer Mode. In this mode, when a remote frame is received, the mailbox data are sent automatically. By enabling this mode, a producer can be done using only one mailbox instead of two: one to detect the remote frame and one to send the answer.

Table 49-3. Transmit Mailbox Objects

Object Type	Description
Transmit	The message stored in the mailbox data registers will try to win the bus arbitration immediately or later according to or not the Time Management Unit configuration (see Time Management Unit). The application is notified that the message has been sent or aborted.
Producer	The message prepared in the mailbox data registers will be sent after receiving the next remote frame. This extends transmit mailbox features.

49.7.3 Time Management Unit

The CAN Controller integrates a free-running 16-bit internal timer. The counter is driven by the bit clock of the CAN bus line. It is enabled when the CAN controller is enabled (CANEN set in the CAN_MR). It is automatically cleared in the following cases:

- after a reset
- when the CAN controller is in Low-power mode is enabled (LPM bit set in the CAN_MR and SLEEP bit set in the CAN_SR)
- after a reset of the CAN controller (CANEN bit in the CAN_MR)
- in Time-triggered mode, when a message is accepted by the last mailbox (rising edge of the MRDY signal in the $CAN_MSR_{last_mailbox_number}$ register).

The application can also reset the internal timer by setting TIMRST in the CAN_TCR. The current value of the internal timer is always accessible by reading the CAN_TIM register.

Downloaded from Arrow.com.

When the timer rolls-over from FFFFh to 0000h, TOVF (Timer Overflow) signal in the CAN_SR is set. TOVF bit in the CAN_SR is cleared by reading the CAN_SR. Depending on the corresponding interrupt mask in the CAN_IMR, an interrupt is generated while TOVF is set.

In a CAN network, some CAN devices may have a larger counter. In this case, the application can also decide to freeze the internal counter when the timer reaches FFFFh and to wait for a restart condition from another device. This feature is enabled by setting TIMFRZ in the CAN_MR. The CAN_TIM register is frozen to the FFFFh value. A clear condition described above restarts the timer. A timer overflow (TOVF) interrupt is triggered.

To monitor the CAN bus activity, the CAN_TIM register is copied to the CAN _TIMESTP register after each start of frame or end of frame and a TSTP interrupt is triggered. If TEOF bit in the CAN_MR is set, the value is captured at each End Of Frame, else it is captured at each Start Of Frame. Depending on the corresponding mask in the CAN_IMR, an interrupt is generated while TSTP is set in the CAN_SR. TSTP bit is cleared by reading the CAN_SR.

The time management unit can operate in one of the following two modes:

- Timestamping mode: The value of the internal timer is captured at each Start Of Frame or each End Of Frame
- Time-triggered mode: A mailbox transfer operation is triggered when the internal timer reaches the mailbox trigger.

Timestamping Mode is enabled by clearing TTM field in the CAN_MR. Time-triggered mode is enabled by setting the CAN_MR.TTM field.

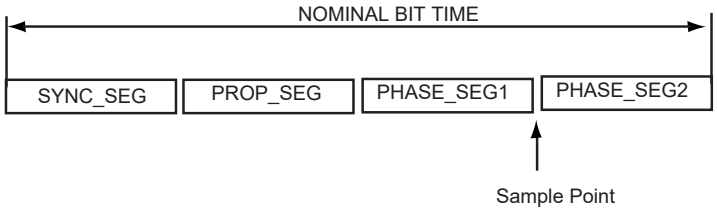
49.7.4 CAN 2.0 Standard Features

49.7.4.1 CAN Bit Timing Configuration

All controllers on a CAN bus must have the same bit rate and bit length. At different clock frequencies of the individual controllers, the bit rate has to be adjusted by the time segments.

The CAN protocol specification partitions the nominal bit time into four different segments.

Figure 49-4. Partition of the CAN Bit Time



- SYNC SEG: SYNChronization Segment
  This part of the bit time is used to synchronize the various nodes on the bus. An edge is expected to lie within this segment. It is one TQ long.
- PROP SEG: PROPagation Segment
  This part of the bit time is used to compensate for the physical delay times within the network. It is twice the sum of the signal's propagation time on the bus line, the input comparator delay, and the output driver delay. It is programmable to be 1,2,..., 8 TQ long.

  This parameter is defined in the PROPAG field of the CAN Baudrate Register.
- PHASE SEG1, PHASE SEG2: PHASE Segment 1 and 2
  The Phase-Buffer-Segments are used to compensate for edge phase errors. These segments can be lengthened (PHASE SEG1) or shortened (PHASE SEG2) by resynchronization.

  Phase Segment 1 is programmable to be 1, 2, ..., 8 TQ long.

  Phase Segment 2 length has to be at least as long as the Information Processing Time (IPT) and may not be more than the length of Phase Segment 1.

  These parameters are defined in the PHASE1 and PHASE2 fields of the CAN Baudrate Register.
- TIME QUANTUM
  The TIME QUANTUM (TQ) is a fixed unit of time derived from the peripheral clock period. The total number of TIME QUANTA in a bit time is programmable from 8 to 25.
- INFORMATION PROCESSING TIME

Downloaded from Arrow.com.

The Information Processing Time (IPT) is the time required for the logic to determine the bit level of a sampled bit. The IPT begins at the sample point, is measured in TQ and is fixed at two TQ for the CAN. Since Phase Segment 2 also begins at the sample point and is the last segment in the bit time, PHASE SEG2 shall not be less than the IPT.

- SAMPLE POINT
  The SAMPLE POINT is the point in time at which the bus level is read and interpreted as the value of that respective bit. Its location is at the end of PHASE_SEG1.
- SJW: ReSynchronization Jump Width
  The ReSynchronization Jump Width defines the limit to the amount of lengthening or shortening of the phase segments.

  SJW is programmable to be the minimum of PHASE SEG1 and four TQ.

If the SMP field in the CAN_BR is set, then the incoming bit stream is sampled three times with a period of half a CAN clock period, centered on sample point.

In the CAN controller, the length of a bit on the CAN bus is determined by the parameters (BRP, PROPAG, PHASE1 and PHASE2).

$t_{BIT} = t_{CSC} + t_{PRS} + t_{PHS1} + t_{PHS2}$

The time quantum is calculated as follows:

$t_{CSC} = (BRP+1)/t_{peripheral\ clock}$

Note: The BRP field must be within the range [1, 0x7F], i.e., BRP = 0 is not authorized.

$t_{PRS} = t_{CSC} \times (PROPAG+1)$

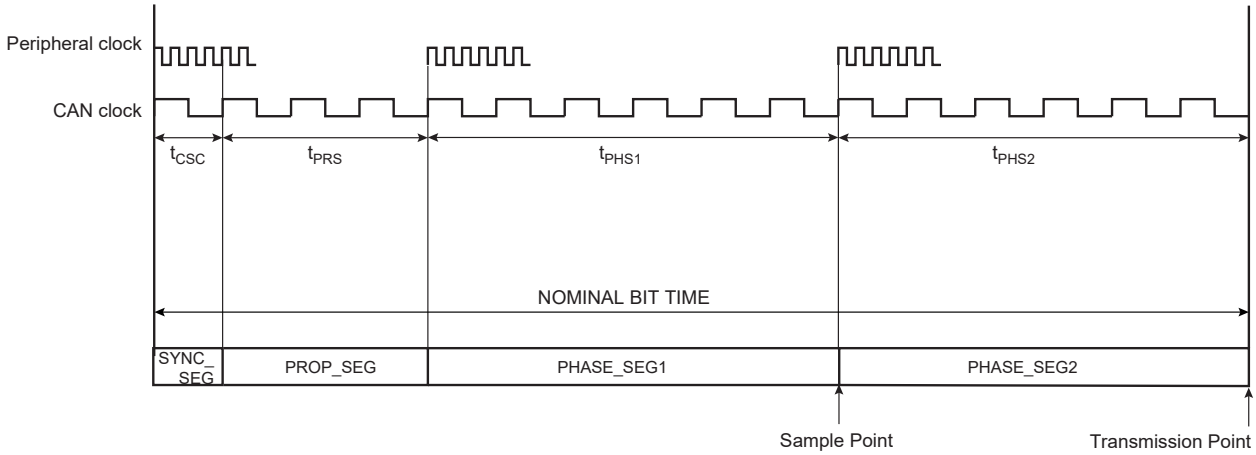
$t_{PHS1} = t_{CSC} \times (PHASE1+1)$

$t_{PHS2} = t_{CSC} \times (PHASE2+1)$

To compensate for phase shifts between clock oscillators of different controllers on the bus, the CAN controller must resynchronize on any relevant signal edge of the current transmission. The resynchronization shortens or lengthens the bit time so that the position of the sample point is shifted with regard to the detected edge. The resynchronization jump width (SJW) defines the maximum of time by which a bit period may be shortened or lengthened by resynchronization.

$t_{SJW} = t_{CSC} \times (SJW+1)$

Figure 49-5. CAN Bit Timing



Example of bit timing determination for CAN baudrate of 500 kbit/s:

$f_{Peripheral\ clock}$ = 48 MHz

CAN baudrate = 500 kbit/s => bit time = 2 µs

Delay of the bus driver: 50 ns

Downloaded from Arrow.com.

Delay of the receiver: 30 ns

Delay of the bus line (20 m): 110 ns

The total number of time quanta in a bit time must be comprised between 8 and 25. If we fix the bit time to 16 time quanta:

t_{CSC} = 1 time quanta = bit time / 16 = 125 ns

=> BRP = (t_{CSC} x $f_{peripheral\ clock}$) - 1 = 5

The propagation segment time is equal to twice the sum of the signal's propagation time on the bus line, the receiver delay and the output driver delay:

t_{PRS} = 2 * (50+30+110) ns = 380 ns = 3 t_{CSC}

=> PROPAG = t_{PRS}/t_{CSC} - 1 = 2

The remaining time for the two phase segments is:

$t_{PHS}1 + t_{PHS}2$ = bit time - t_{CSC} - t_{PRS} = (16 - 1 - 3)t_{CSC}

$t_{PHS}1 + t_{PHS}2$ = 12 t_{CSC}

Because this number is even, we choose $t_{PHS}2 = t_{PHS}1$ (else we would choose $t_{PHS}2 = t_{PHS}1 + t_{CSC}$).

$t_{PHS}1 = t_{PHS}2$ = (12/2) t_{CSC} = 6 t_{CSC}

=> PHASE1 = PHASE2 = $t_{PHS}1/t_{CSC}$ - 1 = 5

The resynchronization jump width must comprise between one t_{CSC} and the minimum of four t_{CSC} and $t_{PHS}1$. We choose its maximum value:

t_{SJW} = Min(4 t_{CSC},$t_{PHS}1$) = 4 t_{CSC}

=> SJW = t_{SJW}/t_{CSC} - 1 = 3

Finally: CAN_BR = 0x00053255

49.7.4.1.1 CAN Bus Synchronization

Two types of synchronization are distinguished: "hard synchronization" at the start of a frame and "resynchronization" inside a frame. After a hard synchronization, the bit time is restarted with the end of the SYNC_SEG segment, regardless of the phase error. Resynchronization causes a reduction or increase in the bit time so that the position of the sample point is shifted with respect to the detected edge.

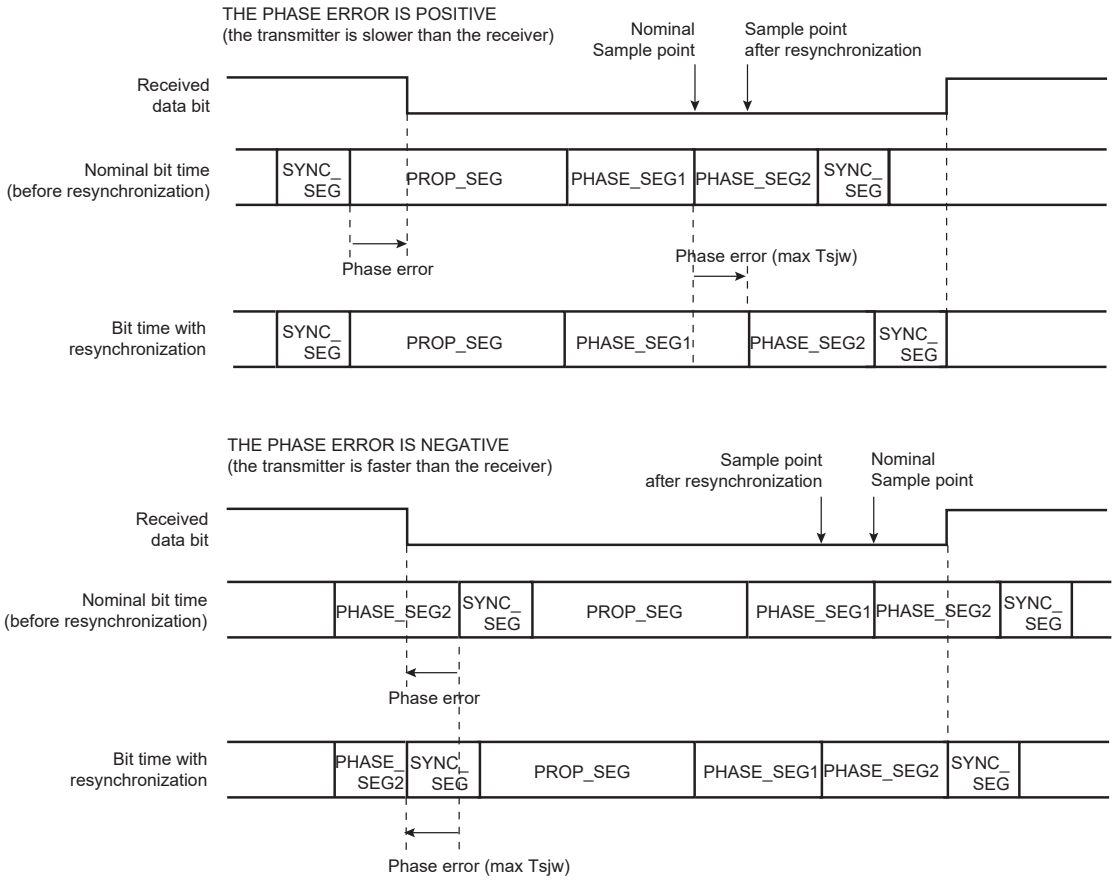
The effect of resynchronization is the same as that of hard synchronization when the magnitude of the phase error of the edge causing the resynchronization is less than or equal to the programmed value of the resynchronization jump width (t_{SJW}).

When the magnitude of the phase error is larger than the resynchronization jump width and

- the phase error is positive, then PHASE_SEG1 is lengthened by an amount equal to the resynchronization jump width.
- the phase error is negative, then PHASE_SEG2 is shortened by an amount equal to the resynchronization jump width.

Downloaded from Arrow.com.

Figure 49-6. CAN Resynchronization



49.7.4.1.2 Autobaud Mode

The autobaud feature is enabled by setting the ABM field in the CAN_MR. In this mode, the CAN controller is only listening to the line without acknowledging the received messages. It can not send any message. The errors flags are updated. The bit timing can be adjusted until no error occurs (good configuration found). In this mode, the error counters are frozen. To go back to the standard mode, the ABM bit must be cleared in the CAN_MR.

49.7.4.2 Error Detection

There are five different error types that are not mutually exclusive. Each error concerns only specific fields of the CAN data frame (refer to the Bosch CAN specification for their correspondence):

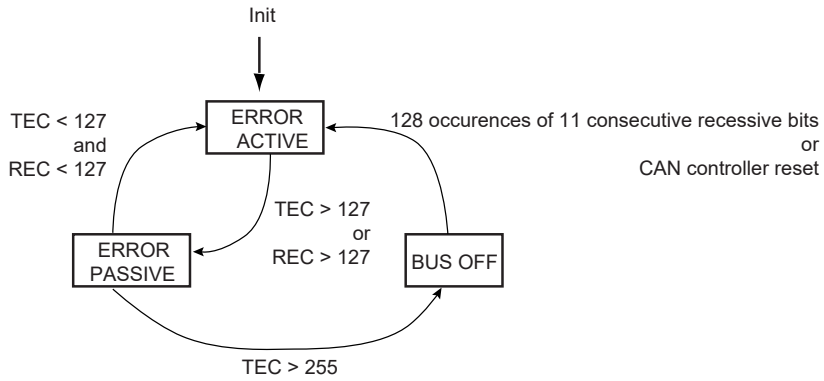
- CRC error (CERR bit in the CAN_SR): With the CRC, the transmitter calculates a checksum for the CRC bit sequence from the Start of Frame bit until the end of the Data Field. This CRC sequence is transmitted in the CRC field of the Data or Remote Frame.
- Bit-stuffing error (SERR bit in the CAN_SR): If a node detects a sixth consecutive equal bit level during the bit-stuffing area of a frame, it generates an Error Frame starting with the next bit-time.
- Bit error (BERR bit in CAN_SR): A bit error occurs if a transmitter sends a dominant bit but detects a recessive bit on the bus line, or if it sends a recessive bit but detects a dominant bit on the bus line. An error frame is generated and starts with the next bit time.
- Form Error (FERR bit in the CAN_SR): If a transmitter detects a dominant bit in one of the fix-formatted segments CRC Delimiter, ACK Delimiter or End of Frame, a form error has occurred and an error frame is generated.
- Acknowledgment error (AERR bit in the CAN_SR): The transmitter checks the Acknowledge Slot, which is transmitted by the transmitting node as a recessive bit, contains a dominant bit. If this is the case, at least one other node has received the frame correctly. If not, an Acknowledge Error has occurred and the transmitter will start in the next bit-time an Error Frame transmission.

Downloaded from Arrow.com.

49.7.4.2.1 Fault Confinement

To distinguish between temporary and permanent failures, every CAN controller has two error counters: REC (Receive Error Counter) and TEC (Transmit Error Counter). The two counters are incremented upon detected errors and are decremented upon correct transmissions or receptions, respectively. Depending on the counter values, the state of the node changes: the initial state of the CAN controller is Error Active, meaning that the controller can send Error Active flags. The controller changes to the Error Passive state if there is an accumulation of errors. If the CAN controller fails or if there is an extreme accumulation of errors, there is a state transition to Bus Off.

Figure 49-7. Line Error Mode



An error active unit takes part in bus communication and sends an active error frame when the CAN controller detects an error.

An error passive unit cannot send an active error frame. It takes part in bus communication, but when an error is detected, a passive error frame is sent. Also, after a transmission, an error passive unit waits before initiating further transmission.

A bus off unit is not allowed to have any influence on the bus.

For fault confinement, two errors counters (TEC and REC) are implemented. These counters are accessible via the CAN_ECR. The state of the CAN controller is automatically updated according to these counter values. If the CAN controller enters Error Active state, then the ERRA bit is set in the CAN_SR. The corresponding interrupt is pending while the interrupt is not masked in the CAN_IMR. If the CAN controller enters Error Passive Mode, then the ERRP bit is set in the CAN_SR and an interrupt remains pending while the ERRP bit is set in the CAN_IMR. If the CAN enters Bus Off Mode, then the BOFF bit is set in the CAN_SR. As for ERRP and ERRA, an interrupt is pending while the BOFF bit is set in the CAN_IMR.

When one of the error counters values exceeds 96, an increased error rate is indicated to the controller through the WARN bit in CAN_SR, but the node remains error active. The corresponding interrupt is pending while the interrupt is set in the CAN_IMR.

Refer to the Bosch CAN specification v2.0 for details on fault confinement.

49.7.4.2.2 Error Interrupt Handler

ERRA, WARN, ERRP and BOFF (CAN_SR) store the key transitions of the CAN bus status as defined in figure Line Error Mode. The transitions depend on the TEC and REC (CAN_ECR) values as described in Fault Confinement.

These flags are latched to keep from triggering a spurious interrupt in case these bits are used as the source of an interrupt. Thus, these flags may not reflect the current status of the CAN bus.

The current CAN bus state can be determined by reading the TEC and REC fields of CAN_ECR.

49.7.4.3 Overload

The overload frame is provided to request a delay of the next data or remote frame by the receiver node (“Request overload frame”) or to signal certain error conditions (“Reactive overload frame”) related to the intermission field respectively.

Reactive overload frames are transmitted after detection of the following error conditions:

- Detection of a dominant bit during the first two bits of the intermission field
- Detection of a dominant bit in the last bit of EOF by a receiver, or detection of a dominant bit by a receiver or a transmitter at the last bit of an error or overload frame delimiter

Downloaded from Arrow.com.

The CAN controller can generate a request overload frame automatically after each message sent to one of the CAN controller mailboxes. This feature is enabled by setting the OVL bit in the CAN_MR.

Reactive overload frames are automatically handled by the CAN controller even if the OVL bit in the CAN_MR is not set. An overload flag is generated in the same way as an error flag, but error counters do not increment.

49.7.5 Low-power Mode

In Low-power mode, the CAN controller cannot send or receive messages. All mailboxes are inactive.

In Low-power mode, the SLEEP signal in the CAN_SR is set; otherwise, the WAKEUP signal in the CAN_SR is set. These two bits are exclusive except after a CAN controller reset (WAKEUP and SLEEP are stuck at 0 after a reset). After power-up reset, the Low-power mode is disabled and the WAKEUP bit is set in the CAN_SR only after detection of 11 consecutive recessive bits on the bus.

49.7.5.1 Enabling Low-power Mode

A software application can enable Low-power mode by setting the LPM bit in the CAN_MR global register. The CAN controller enters Low-power mode once all pending transmit messages are sent.

When the CAN controller enters Low-power mode, the SLEEP signal in the CAN_SR is set. Depending on the corresponding mask in the CAN_IMR, an interrupt is generated while SLEEP is set.

The SLEEP signal in the CAN_SR is automatically cleared once WAKEUP is set. The WAKEUP signal is automatically cleared once SLEEP is set.

Reception is disabled while the SLEEP signal is set to one in the CAN_SR. It is important to note that those messages with higher priority than the last message transmitted can be received between the LPM command and entry in Low-power mode.

Once in Low-power mode, the CAN controller clock can be switched off by programming the chip's Power Management Controller (PMC). The CAN controller drains only the static current.

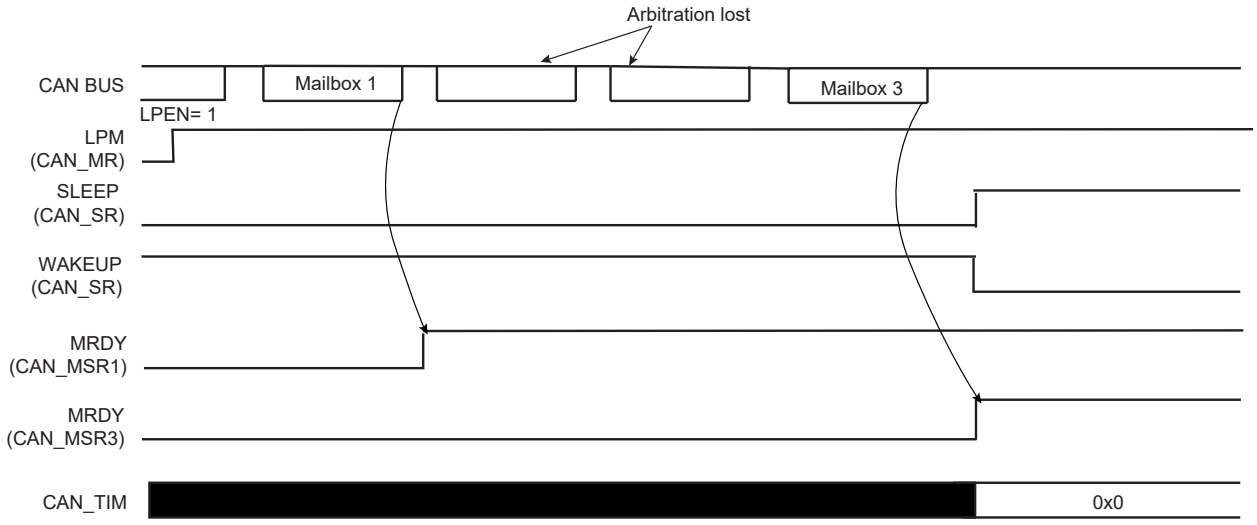
Error counters are disabled while the SLEEP signal is set to one.

Thus, to enter Low-power mode, the software application must:

- Set LPM field in the CAN_MR
- Wait for SLEEP signal rising

Now the CAN Controller clock can be disabled. This is done by programming the Power Management Controller (PMC).

Figure 49-8. Enabling Low-power Mode



Downloaded from Arrow.com.

49.7.5.2 Disabling Low-power Mode

The CAN controller can be awake after detecting a CAN bus activity. Bus activity detection is done by an external module that may be embedded in the chip. When it is notified of a CAN bus activity, the software application disables Low-power mode by programming the CAN controller.

To disable Low-power mode, the software application must:

- Enable the CAN Controller clock. This is done by programming the Power Management Controller (PMC).
- Clear the LPM field in the CAN_MR

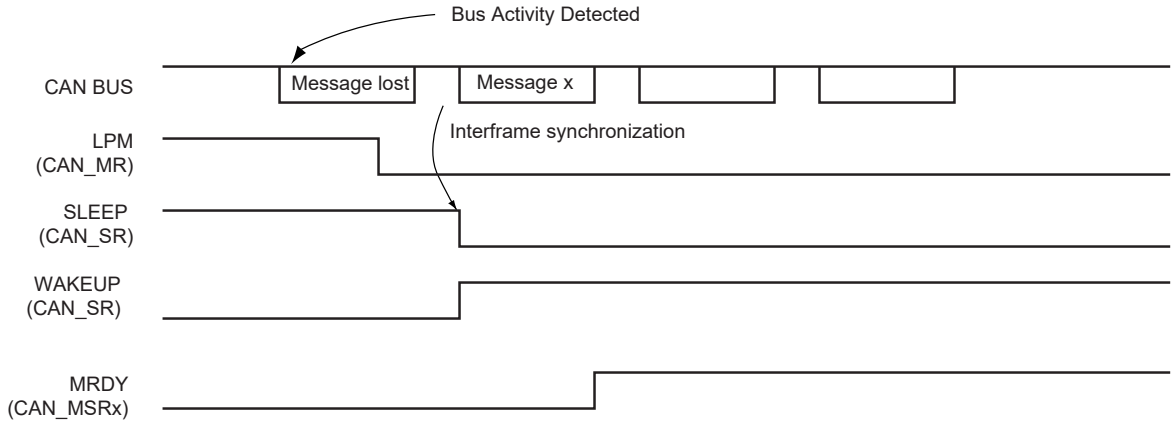
The CAN controller synchronizes itself with the bus activity by checking for eleven consecutive “recessive” bits. Once synchronized, the WAKEUP signal in the CAN_SR is set.

Depending on the corresponding mask in the CAN_IMR, an interrupt is generated while WAKEUP is set. The SLEEP signal in the CAN_SR is automatically cleared once WAKEUP is set. WAKEUP signal is automatically cleared once SLEEP is set.

If no message is being sent on the bus, then the CAN controller is able to send a message eleven bit times after disabling Low-power mode.

If there is bus activity when Low-power mode is disabled, the CAN controller is synchronized with the bus activity in the next interframe. The previous message is lost (see the figure below).

Figure 49-9. Disabling Low-power Mode



49.8 Functional Description

49.8.1 CAN Controller Initialization

After power-up reset, the CAN controller is disabled. The CAN controller clock must be activated by the Power Management Controller (PMC) and the CAN controller interrupt line must be enabled by the interrupt controller.

The CAN controller must be initialized with the CAN network parameters. The CAN_BR defines the sampling point in the bit time period. CAN_BR must be set before the CAN controller is enabled.

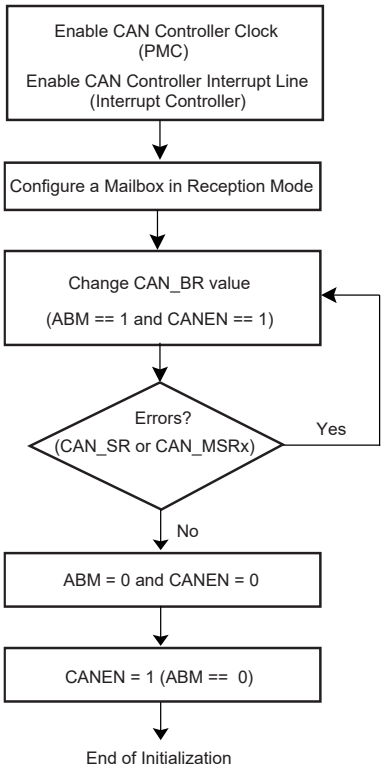
The CAN controller is enabled by setting the CANEN bit in the CAN_MR. At this stage, the internal CAN controller state machine is reset, error counters are reset to 0, and error flags are reset to 0.

Once the CAN controller is enabled, bus synchronization is done automatically by scanning eleven recessive bits. The WAKEUP bit in the CAN_SR is automatically set to 1 when the CAN controller is synchronized (WAKEUP and SLEEP are stuck at 0 after a reset).

The CAN controller can start listening to the network in Autobaud Mode. In this case, the error counters are locked and a mailbox may be configured in Receive Mode. By scanning error flags, the CAN_BR values synchronized with the network. Once no error has been detected, the application disables the Autobaud Mode, clearing the ABM bit in the CAN_MR.

Downloaded from Arrow.com.

Figure 49-10. Possible Initialization Procedure



49.8.2 CAN Controller Interrupt Handling

There are two different types of interrupts. One type of interrupt is a message-object related interrupt, the other is a system interrupt that handles errors or system-related interrupt sources.

All interrupt sources can be masked by writing the corresponding field in the CAN_IDR. They can be unmasked by writing to the CAN_IER. After a power-up reset, all interrupt sources are disabled (masked). The current mask status can be checked by reading the CAN_IMR.

The CAN_SR gives all interrupt source states.

The following events may initiate one of the two interrupts:

- Message object interrupt
  - Data registers in the mailbox object are available to the application. In Receive Mode, a new message was received. In Transmit Mode, a message was transmitted successfully.
  - A sent transmission was aborted.
- System interrupts
  - Bus off interrupt: The CAN module enters the bus off state.
  - Error passive interrupt: The CAN module enters Error Passive Mode.
  - Error Active Mode: The CAN module is neither in Error Passive Mode nor in Bus Off mode.
  - Warn Limit interrupt: The CAN module is in Error-active Mode, but at least one of its error counter value exceeds 96.
  - Wake-up interrupt: This interrupt is generated after a wake-up and a bus synchronization.
  - Sleep interrupt: This interrupt is generated after a Low-power mode enable once all pending messages in transmission have been sent.
  - Internal timer counter overflow interrupt: This interrupt is generated when the internal timer rolls over.
  - Timestamp interrupt: This interrupt is generated after the reception or the transmission of a start of frame or an end of frame. The value of the internal counter is copied in the CAN_TIMESTP register.

Downloaded from Arrow.com.

All interrupts are cleared by clearing the interrupt source except for the internal timer counter overflow interrupt and the timestamp interrupt. These interrupts are cleared by reading the CAN_SR.

49.8.3 CAN Controller Message Handling

49.8.3.1 Receive Handling

Two modes are available to configure a mailbox to receive messages. In Receive Mode, the first message received is stored in the mailbox data register. In Receive with Overwrite Mode, the last message received is stored in the mailbox.

49.8.3.1.1 Simple Receive Mailbox

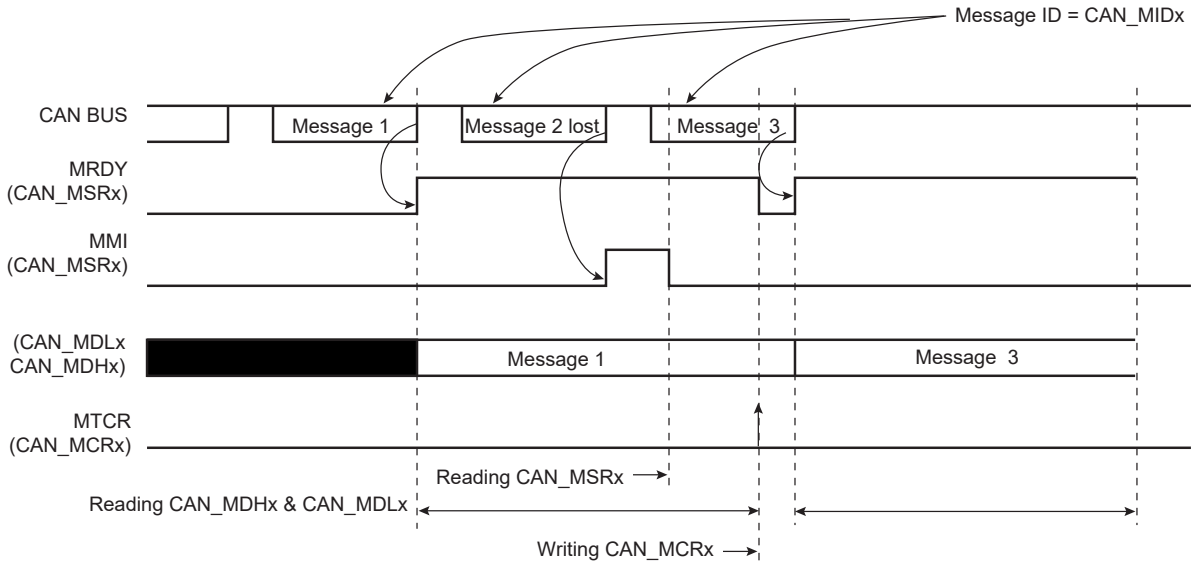
A mailbox is in Receive Mode once the MOT field in the CAN_MMRx has been configured. Message ID and Message Acceptance Mask must be set before the Receive Mode is enabled.

After Receive Mode is enabled, the MRDY flag in the CAN_MSR is automatically cleared until the first message is received. When the first message has been accepted by the mailbox, the MRDY flag is set. An interrupt is pending for the mailbox while the MRDY flag is set. This interrupt can be masked depending on the mailbox flag in the CAN_IMR global register.

Message data are stored in the mailbox data register until the software application notifies that data processing has ended. This is done by asking for a new transfer command, setting the MTCR flag in the CAN_MCRx. This automatically clears the MRDY signal.

The MMI flag in the CAN_MSRx notifies the software that a message has been lost by the mailbox. This flag is set when messages are received while MRDY is set in the CAN_MSRx. This flag is cleared by reading the CAN_MSRs register. A receive mailbox prevents from overwriting the first message by new ones while MRDY flag is set in the CAN_MSRx. See the figure below.

Figure 49-11. Receive Mailbox



Note: In the case of ARM architecture, CAN_MSRx, CAN_MDLx, CAN_MDHx can be read using an optimized ldm assembler instruction.

49.8.3.1.2 Receive with Overwrite Mailbox

A mailbox is in Receive with Overwrite Mode once the MOT field in the CAN_MMRx has been configured. Message ID and Message Acceptance masks must be set before Receive Mode is enabled.

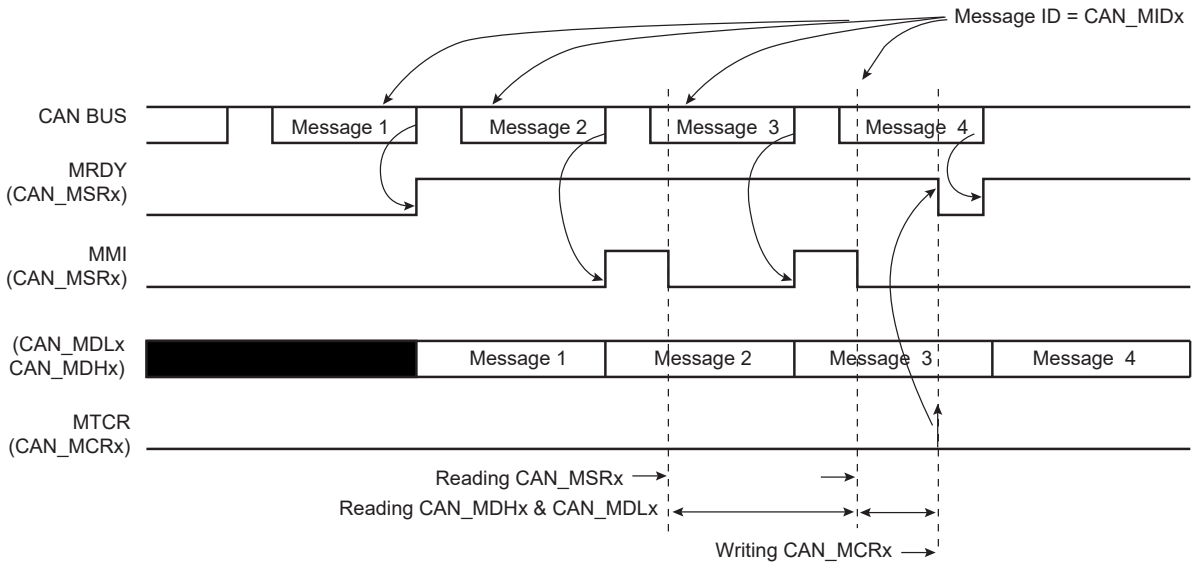
After Receive Mode is enabled, the MRDY flag in the CAN_MSR is automatically cleared until the first message is received. When the first message has been accepted by the mailbox, the MRDY flag is set. An interrupt is pending for the mailbox while the MRDY flag is set. This interrupt is masked depending on the mailbox flag in the CAN_IMR global register.

Downloaded from Arrow.com.

If a new message is received while the MRDY flag is set, this new message is stored in the mailbox data register, overwriting the previous message. The MMI flag in the CAN_MSRx notifies the software that a message has been dropped by the mailbox. This flag is cleared when reading the CAN_MSRx.

The CAN controller may store a new message in the CAN data registers while the application reads them. To check that CAN_MDHx and CAN_MDLx do not belong to different messages, the application must check the MMI bit in the CAN_MSRx before and after reading CAN_MDHx and CAN_MDLx. If the MMI flag is set again after the data registers have been read, the software application has to re-read CAN_MDHx and CAN_MDLx (see the figure below).

Figure 49-12. Receive with Overwrite Mailbox



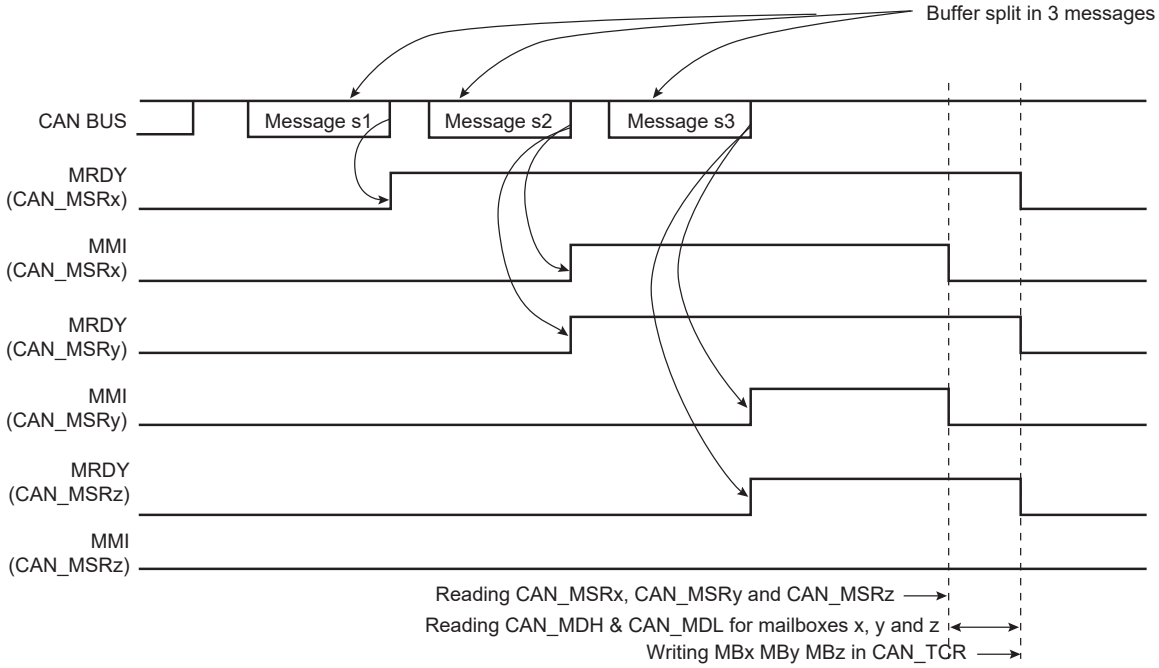
49.8.3.1.3 Chaining Mailboxes

Several mailboxes may be used to receive a buffer split into several messages with the same ID. In this case, the mailbox with the lowest number is serviced first. In the receive and receive with overwrite modes, the field PRIOR in the CAN_MMRx has no effect. If Mailbox 0 and Mailbox 5 accept messages with the same ID, the first message is received by Mailbox 0 and the second message is received by Mailbox 5. Mailbox 0 must be configured in Receive Mode (i.e., the first message received is considered) and Mailbox 5 must be configured in Receive with Overwrite Mode. Mailbox 0 cannot be configured in Receive with Overwrite Mode; otherwise, all messages are accepted by this mailbox and Mailbox 5 is never serviced.

If several mailboxes are chained to receive a buffer split into several messages, all mailboxes except the last one (with the highest number) must be configured in Receive Mode. The first message received is handled by the first mailbox, the second one is refused by the first mailbox and accepted by the second mailbox, the last message is accepted by the last mailbox and refused by previous ones (see the figure below).

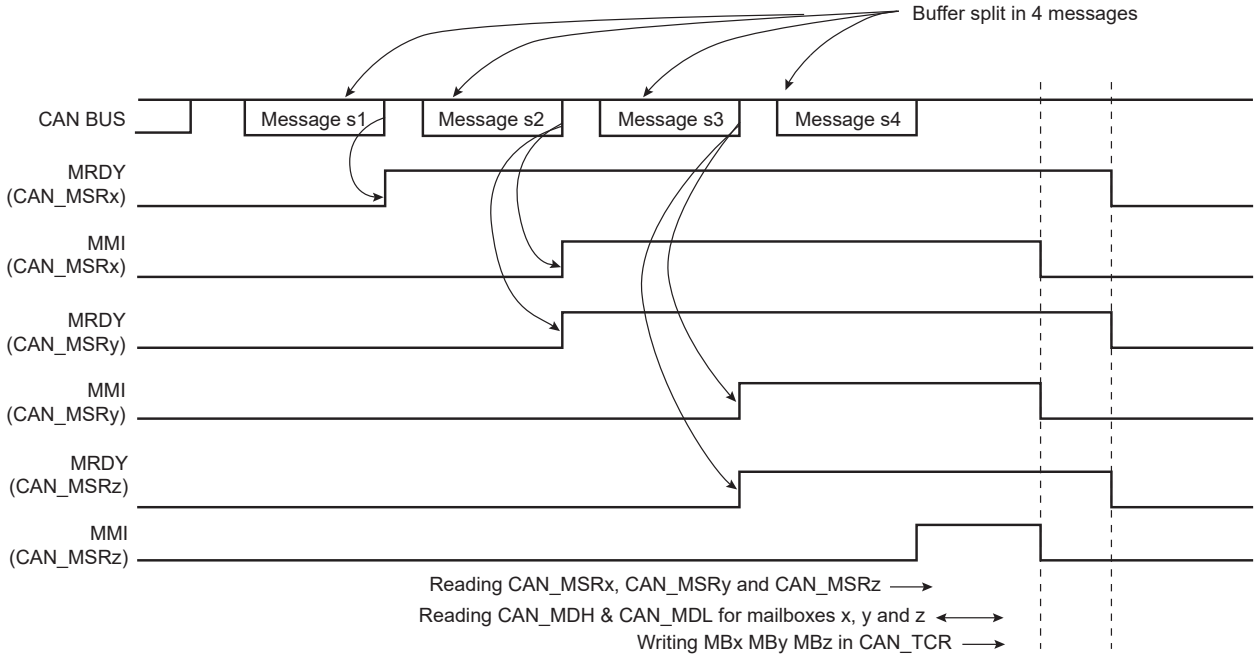
Downloaded from Arrow.com.

Figure 49-13. Chaining Three Mailboxes to Receive a Buffer Split into Three Messages



If the number of mailboxes is not sufficient (the MMI flag of the last mailbox raises), the user must read each data received on the last mailbox in order to retrieve all the messages of the buffer split (see the figure below).

Figure 49-14. Chaining Three Mailboxes to Receive a Buffer Split into Four Messages



49.8.3.2 Transmission Handling

A mailbox is in Transmit Mode once the MOT field in the CAN_MMRx has been configured. Message ID and Message Acceptance mask must be set before Receive Mode is enabled.

After Transmit Mode is enabled, the MRDY flag in the CAN_MSR is automatically set until the first command is sent. When the MRDY flag is set, the software application can prepare a message to be sent by writing to the CAN_MDx registers. The message is sent once the software asks for a transfer command setting the MTCR bit and the message data length in the CAN_MCRx.

Downloaded from Arrow.com.

The MRDY flag remains at zero as long as the message has not been sent or aborted. It is important to note that no access to the mailbox data register is allowed while the MRDY flag is cleared. An interrupt is pending for the mailbox while the MRDY flag is set. This interrupt can be masked depending on the mailbox flag in the CAN_IMR global register.

It is also possible to send a remote frame setting the MRTR bit instead of setting the MDLC field. The answer to the remote frame is handled by another reception mailbox. In this case, the device acts as a consumer but with the help of two mailboxes. It is possible to handle the remote frame emission and the answer reception using only one mailbox configured in Consumer Mode. Refer to the section Remote Frame Handling.

Several messages can try to win the bus arbitration in the same time. The message with the highest priority is sent first. Several transfer request commands can be generated at the same time by setting MBx bits in the CAN_TCR. The priority is set in the PRIOR field of the CAN_MMRx. Priority 0 is the highest priority, priority 15 is the lowest priority. Thus it is possible to use a part of the message ID to set the PRIOR field. If two mailboxes have the same priority, the message of the mailbox with the lowest number is sent first. Thus if mailbox 0 and mailbox 5 have the same priority and have a message to send at the same time, then the message of the mailbox 0 is sent first.

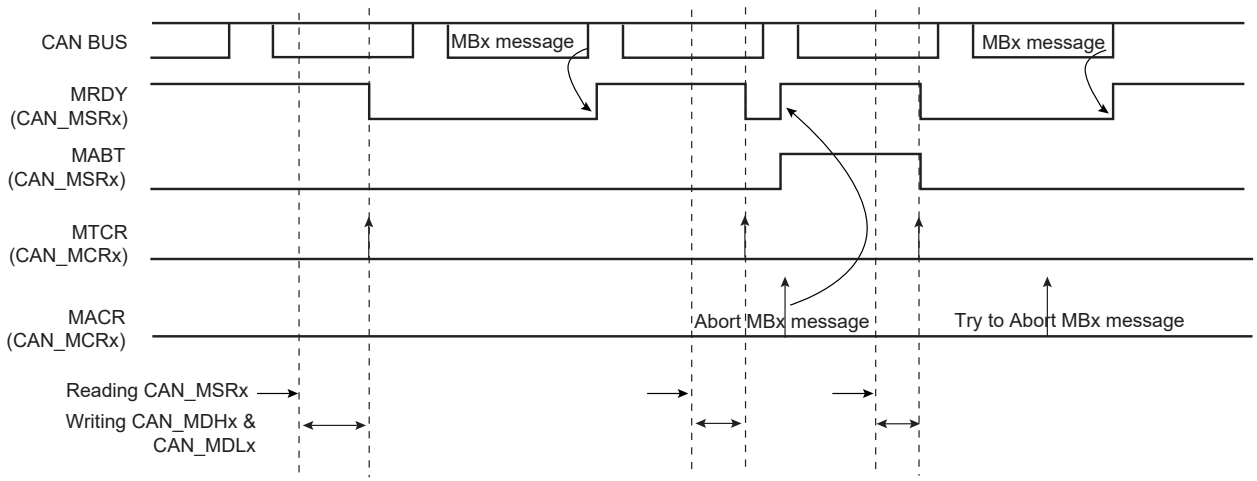
Setting the MACR bit in the CAN_MCRx aborts the transmission. Transmission for several mailboxes can be aborted by writing MBx fields in the CAN_ACR. If the message is being sent when the abort command is set, then the application is notified by the MRDY bit set and not the MABT in the CAN_MSRx. Otherwise, if the message has not been sent, then the MRDY and the MABT are set in the CAN_MSR.

When the bus arbitration is lost by a mailbox message, the CAN controller tries to win the next bus arbitration with the same message if this one still has the highest priority. Messages to be sent are re-tried automatically until they win the bus arbitration. This feature can be disabled by setting the bit DRPT in the CAN_MR. In this case if the message was not sent the first time it was transmitted to the CAN transceiver, it is automatically aborted. The MABT flag is set in the CAN_MSRx until the next transfer command.

The following figure shows three MBx message attempts being made (MRDY of MBx set to 0).

The first MBx message is sent, the second is aborted and the last one is trying to be aborted but too late because it has already been transmitted to the CAN transceiver.

Figure 49-15. Transmitting Messages

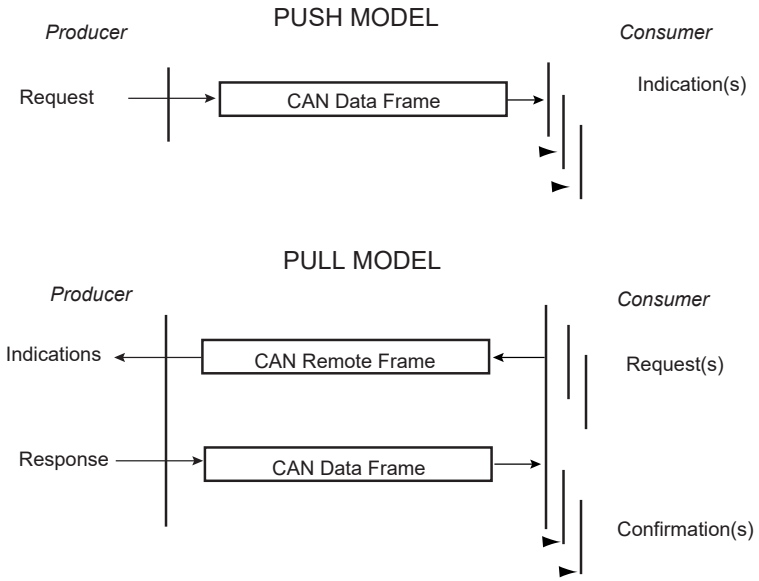


49.8.3.3 Remote Frame Handling

Producer/consumer model is an efficient means of handling broadcasted messages. The push model allows a producer to broadcast messages; the pull model allows a customer to ask for messages.

Downloaded from Arrow.com.

Figure 49-16. Producer / Consumer Model



In Pull Mode, a consumer transmits a remote frame to the producer. When the producer receives a remote frame, it sends the answer accepted by one or many consumers. Using transmit and receive mailboxes, a consumer must dedicate two mailboxes, one in Transmit Mode to send remote frames, and at least one in Receive Mode to capture the producer's answer. The same structure is applicable to a producer: one reception mailbox is required to get the remote frame and one transmit mailbox to answer.

Mailboxes can be configured in Producer or Consumer Mode. A lonely mailbox can handle the remote frame and the answer. With 8 mailboxes, the CAN controller can handle 8 independent producers/consumers.

49.8.3.3.1 Producer Configuration

A mailbox is in Producer Mode once the MOT field in the CAN_MMRx has been configured. Message ID and Message Acceptance masks must be set before Receive Mode is enabled.

After Producer Mode is enabled, the MRDY flag in the CAN_MSR is automatically set until the first transfer command. The software application prepares data to be sent by writing to the CAN_MDHx and the CAN_MDLx registers, then by setting the MTCR bit in the CAN_MCRx. Data is sent after the reception of a remote frame as soon as it wins the bus arbitration.

The MRDY flag remains at zero as long as the message has not been sent or aborted. No access to the mailbox data register can be done while MRDY flag is cleared. An interrupt is pending for the mailbox while the MRDY flag is set. This interrupt can be masked according to the mailbox flag in the CAN_IMR global register.

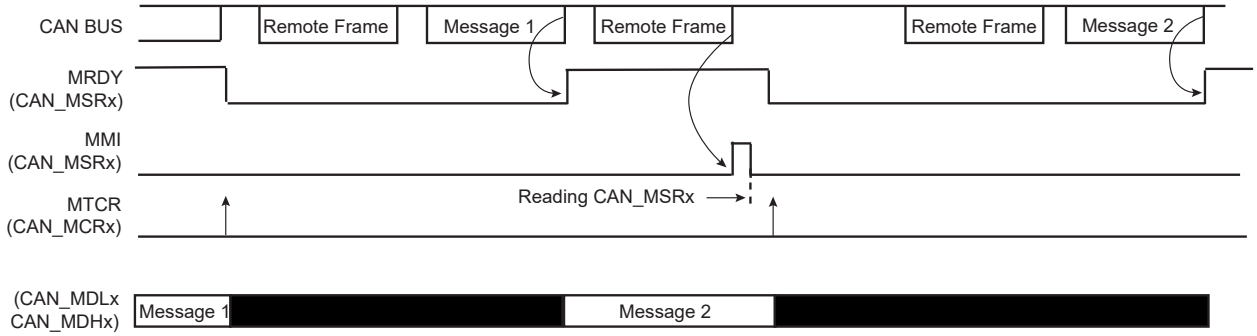
If a remote frame is received while no data are ready to be sent (signal MRDY set in the CAN_MSRx), then the MMI signal is set in the CAN_MSRx. This bit is cleared by reading the CAN_MSRx.

The MRTR field in the CAN_MSRx has no meaning. This field is used only when using Receive and Receive with Overwrite modes.

After a remote frame has been received, the mailbox functions like a transmit mailbox. The message with the highest priority is sent first. The transmitted message may be aborted by setting the MACR bit in the CAN_MCR. Please refer to the section Transmission Handling.

Downloaded from Arrow.com.

Figure 49-17. Producer Handling



49.8.3.3.2 Consumer Configuration

A mailbox is in Consumer Mode once the MOT field in the CAN_MMRx has been configured. Message ID and Message Acceptance masks must be set before Receive Mode is enabled.

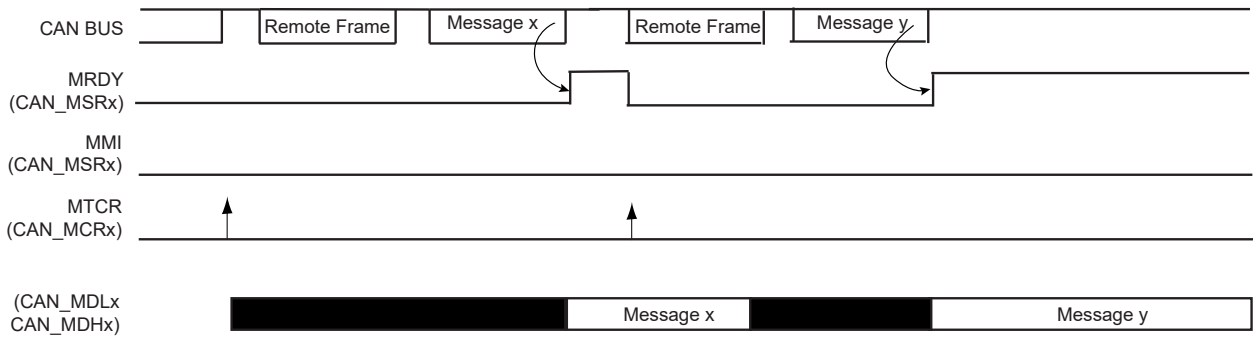
After Consumer Mode is enabled, the MRDY flag in the CAN_MSR is automatically cleared until the first transfer request command. The software application sends a remote frame by setting the MTCR bit in the CAN_MCRx or the MBx bit in the global CAN_TCR. The application is notified of the answer by the MRDY flag set in the CAN_MSRx. The application can read the data contents in the CAN_MDHx and CAN_MDLx registers. An interrupt is pending for the mailbox while the MRDY flag is set. This interrupt can be masked according to the mailbox flag in the CAN_IMR global register.

The MRTR bit in the CAN_MCRx has no effect. This field is used only when using Transmit Mode.

After a remote frame has been sent, the consumer mailbox functions as a reception mailbox. The first message received is stored in the mailbox data registers. If other messages intended for this mailbox have been sent while the MRDY flag is set in the CAN_MSRx, they will be lost. The application is notified by reading the MMI bit in the CAN_MSRx. The read operation automatically clears the MMI flag.

If several messages are answered by the Producer, the CAN controller may have one mailbox in consumer configuration, zero or several mailboxes in Receive Mode and one mailbox in Receive with Overwrite Mode. In this case, the consumer mailbox must have a lower number than the Receive with Overwrite mailbox. The transfer command can be triggered for all mailboxes at the same time by setting several MBx fields in the CAN_TCR.

Figure 49-18. Consumer Handling



49.8.4 CAN Controller Timing Modes

Using the free running 16-bit internal timer, the CAN controller can be set in one of the following timing modes:

- Timestamping mode: the value of the internal timer is captured at each Start Of Frame or each End Of Frame.
- Time-triggered mode: the mailbox transfer operation is triggered when the internal timer reaches the mailbox trigger.

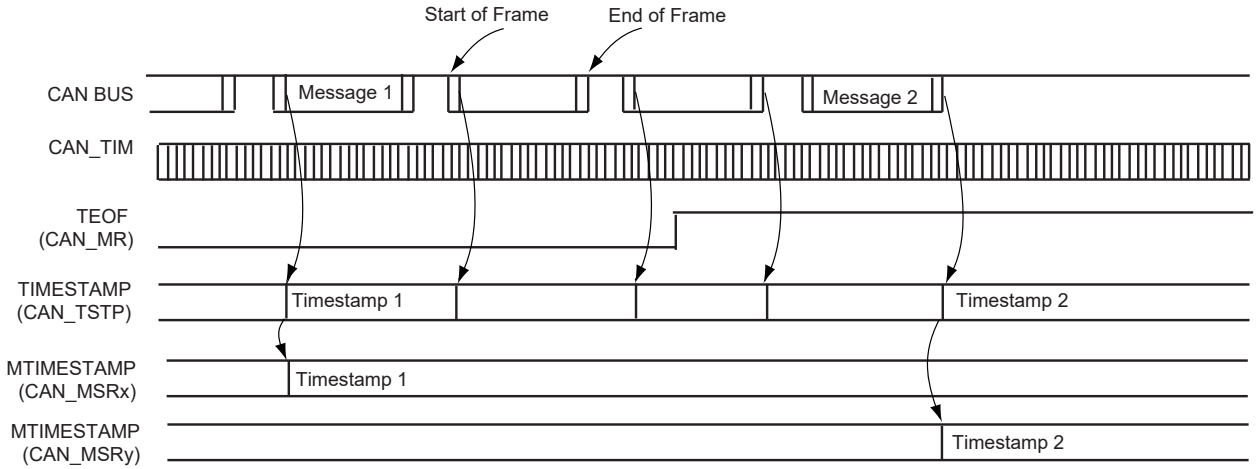
Timestamping mode is enabled by clearing the CAN_MR.TTM bit. Time-triggered mode is enabled by setting the CAN_MR.TTM bit.

Downloaded from Arrow.com.

49.8.4.1 Timestamping Mode

Each mailbox has its own timestamp value. Each time a message is sent or received by a mailbox, the 16-bit value MTIMESTAMP of the CAN_TIMESTP register is transferred to the LSB bits of the CAN_MSRx. The value read in the CAN_MSRx corresponds to the internal timer value at the Start Of Frame or the End Of Frame of the message handled by the mailbox.

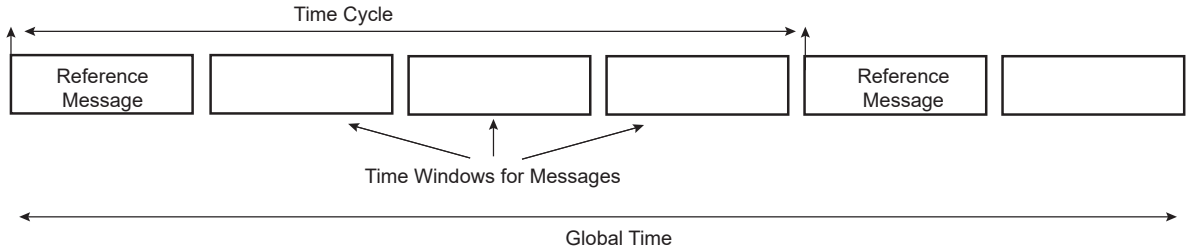
Figure 49-19. Mailbox Timestamp



49.8.4.2 Time-triggered Mode

In Time-triggered mode, basic cycles can be split into several time windows. A basic cycle starts with a reference message. Each time a window is defined from the reference message, a transmit operation should occur within a predefined time window. A mailbox must not win the arbitration in a previous time window, and it must not be retried if the arbitration is lost in the time window.

Figure 49-20. Time-triggered Principle



Time-triggered mode is enabled by setting the CAN_MR.TTM bit. In Time-triggered mode, as in Timestamp mode, the CAN_TIMESTP field captures the values of the internal counter, but the MTIMESTAMP fields in the CAN_MSRx registers are not active and are read at 0.

49.8.4.2.1 Synchronization by a Reference Message

In Time-triggered mode, the internal timer counter is automatically reset when a new message is received in the last mailbox. This reset occurs after the reception of the End Of Frame on the rising edge of the MRDY signal in the CAN_MSRx. This allows synchronization of the internal timer counter with the reception of a reference message and the start a new time window.

49.8.4.2.2 Transmitting within a Time Window

A time mark is defined for each mailbox. It is defined in the 16-bit MTIMEMARK field of the CAN_MMRx. At each internal timer clock cycle, the value of the CAN_TIM is compared with each mailbox time mark. When the internal timer counter reaches the MTIMEMARK value, an internal timer event for the mailbox is generated for the mailbox.

In Time-triggered mode, transmit operations are delayed until the internal timer event for the mailbox. The application prepares a message to be sent by setting the MTCR in the CAN_MCRx. The message is not sent until the CAN_TIM value is less than the MTIMEMARK value defined in the CAN_MMRx.

If the transmit operation is failed, i.e., the message loses the bus arbitration and the next transmit attempt is delayed until the next internal time trigger event. This prevents overlapping the next time window, but the message is still

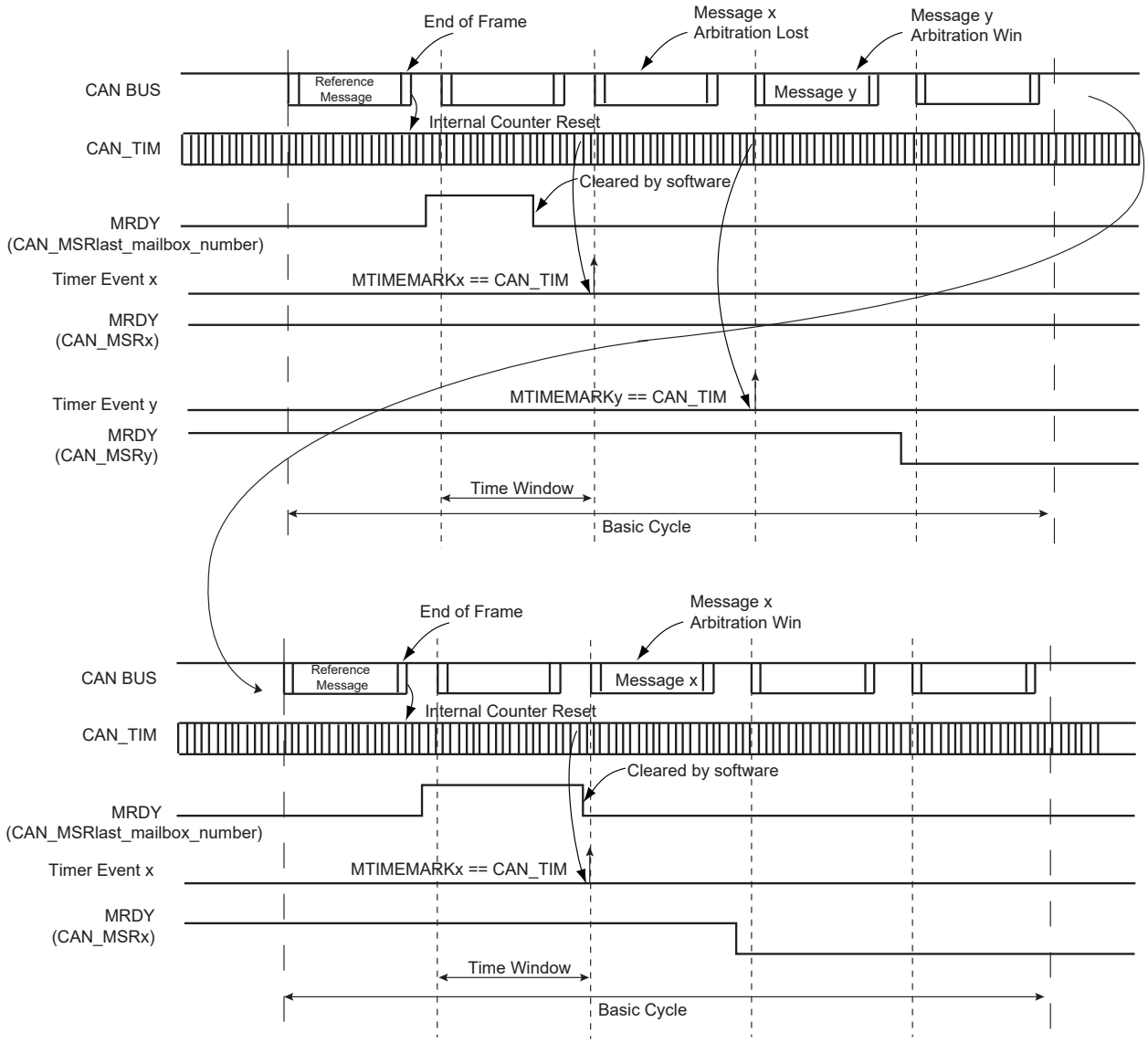
Downloaded from Arrow.com.

pending and is retried in the next time window when CAN_TIM value equals the MTIMEMARK value. It is also possible to prevent a retry by setting the DRPT field in the CAN_MR.

49.8.4.2.3 Freezing the Internal Timer Counter

The internal counter can be frozen by setting TIMFRZ in the CAN_MR. This prevents an unexpected roll-over when the counter reaches FFFFh. When this occurs, it automatically freezes until a new reset is issued, either due to a message received in the last mailbox or any other reset counter operations. The TOVF bit in the CAN_SR is set when the counter is frozen. The TOVF bit in the CAN_SR is cleared by reading the CAN_SR. Depending on the corresponding interrupt mask in the CAN_IMR, an interrupt is generated when TOVF is set.

Figure 49-21. Time-Triggered Operations



49.8.5 Register Write Protection

To prevent any single software error that may corrupt CAN behavior, the registers listed below can be write-protected by setting the WPEN bit in the CAN Write Protection Mode Register (CAN_WPMR).

If a write access in a write-protected register is detected, then the WPVS flag in the CAN Write Protection Status Register (CAN_WPSR) is set and the field WPVSRC indicates in which register the write access has been attempted.

The WPVS flag is automatically reset after reading the CAN_WPSR.

Downloaded from Arrow.com.

The following registers can be write-protected:

- CAN Mode Register
- CAN Baudrate Register
- CAN Message Mode Register
- CAN Message Acceptance Mask Register
- CAN Message ID Register

Downloaded from Arrow.com.

49.9 Register Summary

Offset	Name	Bit Pos.								
0x00	CAN_MR	31:24								
		23:16								
		15:8								
		7:0	DRPT	TIMFRZ	TTM	TEOF	OVL	ABM	LPM	CANEN
0x04	CAN_IER	31:24				BERR	FERR	AERR	SERR	CERR
		23:16	TSTP	TOVF	WAKEUP	SLEEP	BOFF	ERRP	WARN	ERRA
		15:8	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8
		7:0	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
0x08	CAN_IDR	31:24				BERR	FERR	AERR	SERR	CERR
		23:16	TSTP	TOVF	WAKEUP	SLEEP	BOFF	ERRP	WARN	ERRA
		15:8	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8
		7:0	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
0x0C	CAN_IMR	31:24				BERR	FERR	AERR	SERR	CERR
		23:16	TSTP	TOVF	WAKEUP	SLEEP	BOFF	ERRP	WARN	ERRA
		15:8	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8
		7:0	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
0x10	CAN_SR	31:24	OVLSY	TBSY	RBSY	BERR	FERR	AERR	SERR	CERR
		23:16	TSTP	TOVF	WAKEUP	SLEEP	BOFF	ERRP	WARN	ERRA
		15:8	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8
		7:0	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
0x14	CAN_BR	31:24								SMP
		23:16		BRP[6:0]						
		15:8			SJW[1:0]			PROPAG[2:0]		
		7:0		PHASE1[2:0]				PHASE2[2:0]		
0x18	CAN_TIM	31:24								
		23:16								
		15:8	TIMER[15:8]							
		7:0	TIMER[7:0]							
0x1C	CAN_TIMESTP	31:24								
		23:16								
		15:8	MTIMESTAMP[15:8]							
		7:0	MTIMESTAMP[7:0]							
0x20	CAN_ECR	31:24								TEC[8]
		23:16	TEC[7:0]							
		15:8								
		7:0	REC[7:0]							
0x24	CAN_TCR	31:24	TIMRST							
		23:16								
		15:8	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8
		7:0	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
0x28	CAN_ACR	31:24								
		23:16								
		15:8	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8
		7:0	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
0x2C ... 0xE3	Reserved									
0xE4	CAN_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0xE8	CAN_WPSR	31:24								
		23:16								
		15:8	WPVSRC[7:0]							
		7:0								WPVS

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.									
0xEC ... 0x01FF	Reserved										
0x0200	CAN_MMR0	31:24						MOT[2:0]			
		23:16					PRIOR[3:0]				
		15:8	MTIMEMARK[15:8]								
		7:0	MTIMEMARK[7:0]								
0x0204	CAN_MAM0	31:24			MIDE	MIDvA[10:6]					
		23:16	MIDvA[5:0]						MIDvB[17:16]		
		15:8	MIDvB[15:8]								
		7:0	MIDvB[7:0]								
0x0208	CAN_MID0	31:24			MIDE	MIDvA[10:6]					
		23:16	MIDvA[5:0]						MIDvB[17:16]		
		15:8	MIDvB[15:8]								
		7:0	MIDvB[7:0]								
0x020C	CAN_MFID0	31:24				MFID[28:24]					
		23:16	MFID[23:16]								
		15:8	MFID[15:8]								
		7:0	MFID[7:0]								
0x0210	CAN_MSR0	31:24								MMI	
		23:16	MRDY	MABT		MRTR	MDLC[3:0]				
		15:8	MTIMESTAMP[15:8]								
		7:0	MTIMESTAMP[7:0]								
0x0214	CAN_MDL0	31:24	MDL[31:24]								
		23:16	MDL[23:16]								
		15:8	MDL[15:8]								
		7:0	MDL[7:0]								
0x0218	CAN_MDH0	31:24	MDH[31:24]								
		23:16	MDH[23:16]								
		15:8	MDH[15:8]								
		7:0	MDH[7:0]								
0x021C	CAN_MCR0	31:24									
		23:16	MTCR	MACR		MRTR	MDLC[3:0]				
		15:8									
		7:0									
0x0220	CAN_MMR1	31:24						MOT[2:0]			
		23:16					PRIOR[3:0]				
		15:8	MTIMEMARK[15:8]								
		7:0	MTIMEMARK[7:0]								
0x0224	CAN_MAM1	31:24			MIDE	MIDvA[10:6]					
		23:16	MIDvA[5:0]						MIDvB[17:16]		
		15:8	MIDvB[15:8]								
		7:0	MIDvB[7:0]								
0x0228	CAN_MID1	31:24			MIDE	MIDvA[10:6]					
		23:16	MIDvA[5:0]						MIDvB[17:16]		
		15:8	MIDvB[15:8]								
		7:0	MIDvB[7:0]								
0x022C	CAN_MFID1	31:24				MFID[28:24]					
		23:16	MFID[23:16]								
		15:8	MFID[15:8]								
		7:0	MFID[7:0]								
0x0230	CAN_MSR1	31:24								MMI	
		23:16	MRDY	MABT		MRTR	MDLC[3:0]				
		15:8	MTIMESTAMP[15:8]								
		7:0	MTIMESTAMP[7:0]								
0x0234	CAN_MDL1	31:24	MDL[31:24]								
		23:16	MDL[23:16]								
		15:8	MDL[15:8]								
		7:0	MDL[7:0]								

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x0238	CAN_MDH1	31:24	MDH[31:24]							
		23:16	MDH[23:16]							
		15:8	MDH[15:8]							
		7:0	MDH[7:0]							
0x023C	CAN_MCR1	31:24								
		23:16	MTCR	MACR		MRTR	MDLC[3:0]			
		15:8								
		7:0								
0x0240	CAN_MMR2	31:24						MOT[2:0]		
		23:16					PRIOR[3:0]			
		15:8	MTIMEMARK[15:8]							
		7:0	MTIMEMARK[7:0]							
0x0244	CAN_MAM2	31:24			MIDE	MIDvA[10:6]				
		23:16	MIDvA[5:0]						MIDvB[17:16]	
		15:8	MIDvB[15:8]							
		7:0	MIDvB[7:0]							
0x0248	CAN_MID2	31:24			MIDE	MIDvA[10:6]				
		23:16	MIDvA[5:0]						MIDvB[17:16]	
		15:8	MIDvB[15:8]							
		7:0	MIDvB[7:0]							
0x024C	CAN_MFID2	31:24				MFID[28:24]				
		23:16	MFID[23:16]							
		15:8	MFID[15:8]							
		7:0	MFID[7:0]							
0x0250	CAN_MSR2	31:24								MMI
		23:16	MRDY	MABT		MRTR	MDLC[3:0]			
		15:8	MTIMESTAMP[15:8]							
		7:0	MTIMESTAMP[7:0]							
0x0254	CAN_MDL2	31:24	MDL[31:24]							
		23:16	MDL[23:16]							
		15:8	MDL[15:8]							
		7:0	MDL[7:0]							
0x0258	CAN_MDH2	31:24	MDH[31:24]							
		23:16	MDH[23:16]							
		15:8	MDH[15:8]							
		7:0	MDH[7:0]							
0x025C	CAN_MCR2	31:24								
		23:16	MTCR	MACR		MRTR	MDLC[3:0]			
		15:8								
		7:0								
0x0260	CAN_MMR3	31:24						MOT[2:0]		
		23:16					PRIOR[3:0]			
		15:8	MTIMEMARK[15:8]							
		7:0	MTIMEMARK[7:0]							
0x0264	CAN_MAM3	31:24			MIDE	MIDvA[10:6]				
		23:16	MIDvA[5:0]						MIDvB[17:16]	
		15:8	MIDvB[15:8]							
		7:0	MIDvB[7:0]							
0x0268	CAN_MID3	31:24			MIDE	MIDvA[10:6]				
		23:16	MIDvA[5:0]						MIDvB[17:16]	
		15:8	MIDvB[15:8]							
		7:0	MIDvB[7:0]							
0x026C	CAN_MFID3	31:24				MFID[28:24]				
		23:16	MFID[23:16]							
		15:8	MFID[15:8]							
		7:0	MFID[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x0270	CAN_MSR3	31:24								MMI
		23:16	MRDY	MABT		MRTR	MDLC[3:0]			
		15:8	MTIMESTAMP[15:8]							
		7:0	MTIMESTAMP[7:0]							
0x0274	CAN_MDL3	31:24	MDL[31:24]							
		23:16	MDL[23:16]							
		15:8	MDL[15:8]							
		7:0	MDL[7:0]							
0x0278	CAN_MDH3	31:24	MDH[31:24]							
		23:16	MDH[23:16]							
		15:8	MDH[15:8]							
		7:0	MDH[7:0]							
0x027C	CAN_MCR3	31:24								
		23:16	MTCR	MACR		MRTR	MDLC[3:0]			
		15:8								
		7:0								
0x0280	CAN_MMR4	31:24						MOT[2:0]		
		23:16					PRIOR[3:0]			
		15:8	MTIMEMARK[15:8]							
		7:0	MTIMEMARK[7:0]							
0x0284	CAN_MAM4	31:24			MIDE	MIDvA[10:6]				
		23:16	MIDvA[5:0]						MIDvB[17:16]	
		15:8	MIDvB[15:8]							
		7:0	MIDvB[7:0]							
0x0288	CAN_MID4	31:24			MIDE	MIDvA[10:6]				
		23:16	MIDvA[5:0]						MIDvB[17:16]	
		15:8	MIDvB[15:8]							
		7:0	MIDvB[7:0]							
0x028C	CAN_MFID4	31:24				MFID[28:24]				
		23:16	MFID[23:16]							
		15:8	MFID[15:8]							
		7:0	MFID[7:0]							
0x0290	CAN_MSR4	31:24								MMI
		23:16	MRDY	MABT		MRTR	MDLC[3:0]			
		15:8	MTIMESTAMP[15:8]							
		7:0	MTIMESTAMP[7:0]							
0x0294	CAN_MDL4	31:24	MDL[31:24]							
		23:16	MDL[23:16]							
		15:8	MDL[15:8]							
		7:0	MDL[7:0]							
0x0298	CAN_MDH4	31:24	MDH[31:24]							
		23:16	MDH[23:16]							
		15:8	MDH[15:8]							
		7:0	MDH[7:0]							
0x029C	CAN_MCR4	31:24								
		23:16	MTCR	MACR		MRTR	MDLC[3:0]			
		15:8								
		7:0								
0x02A0	CAN_MMR5	31:24						MOT[2:0]		
		23:16					PRIOR[3:0]			
		15:8	MTIMEMARK[15:8]							
		7:0	MTIMEMARK[7:0]							
0x02A4	CAN_MAM5	31:24			MIDE	MIDvA[10:6]				
		23:16	MIDvA[5:0]						MIDvB[17:16]	
		15:8	MIDvB[15:8]							
		7:0	MIDvB[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x02A8	CAN_MID5	31:24			MIDE	MIDvA[10:6]				
		23:16	MIDvA[5:0]						MIDvB[17:16]	
		15:8	MIDvB[15:8]							
		7:0	MIDvB[7:0]							
0x02AC	CAN_MFID5	31:24				MFID[28:24]				
		23:16	MFID[23:16]							
		15:8	MFID[15:8]							
		7:0	MFID[7:0]							
0x02B0	CAN_MSR5	31:24								MMI
		23:16	MRDY	MABT		MRTR	MDLC[3:0]			
		15:8	MTIMESTAMP[15:8]							
		7:0	MTIMESTAMP[7:0]							
0x02B4	CAN_MDL5	31:24	MDL[31:24]							
		23:16	MDL[23:16]							
		15:8	MDL[15:8]							
		7:0	MDL[7:0]							
0x02B8	CAN_MDH5	31:24	MDH[31:24]							
		23:16	MDH[23:16]							
		15:8	MDH[15:8]							
		7:0	MDH[7:0]							
0x02BC	CAN_MCR5	31:24								
		23:16	MTCR	MACR		MRTR	MDLC[3:0]			
		15:8								
		7:0								
0x02C0	CAN_MMR6	31:24						MOT[2:0]		
		23:16					PRIOR[3:0]			
		15:8	MTIMEMARK[15:8]							
		7:0	MTIMEMARK[7:0]							
0x02C4	CAN_MAM6	31:24			MIDE	MIDvA[10:6]				
		23:16	MIDvA[5:0]						MIDvB[17:16]	
		15:8	MIDvB[15:8]							
		7:0	MIDvB[7:0]							
0x02C8	CAN_MID6	31:24			MIDE	MIDvA[10:6]				
		23:16	MIDvA[5:0]						MIDvB[17:16]	
		15:8	MIDvB[15:8]							
		7:0	MIDvB[7:0]							
0x02CC	CAN_MFID6	31:24				MFID[28:24]				
		23:16	MFID[23:16]							
		15:8	MFID[15:8]							
		7:0	MFID[7:0]							
0x02D0	CAN_MSR6	31:24								MMI
		23:16	MRDY	MABT		MRTR	MDLC[3:0]			
		15:8	MTIMESTAMP[15:8]							
		7:0	MTIMESTAMP[7:0]							
0x02D4	CAN_MDL6	31:24	MDL[31:24]							
		23:16	MDL[23:16]							
		15:8	MDL[15:8]							
		7:0	MDL[7:0]							
0x02D8	CAN_MDH6	31:24	MDH[31:24]							
		23:16	MDH[23:16]							
		15:8	MDH[15:8]							
		7:0	MDH[7:0]							
0x02DC	CAN_MCR6	31:24								
		23:16	MTCR	MACR		MRTR	MDLC[3:0]			
		15:8								
		7:0								

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x02E0	CAN_MMR7	31:24						MOT[2:0]		
		23:16					PRIOR[3:0]			
		15:8	MTIMEMARK[15:8]							
		7:0	MTIMEMARK[7:0]							
0x02E4	CAN_MAM7	31:24			MIDE	MIDvA[10:6]				
		23:16	MIDvA[5:0]						MIDvB[17:16]	
		15:8	MIDvB[15:8]							
		7:0	MIDvB[7:0]							
0x02E8	CAN_MID7	31:24			MIDE	MIDvA[10:6]				
		23:16	MIDvA[5:0]						MIDvB[17:16]	
		15:8	MIDvB[15:8]							
		7:0	MIDvB[7:0]							
0x02EC	CAN_MFID7	31:24				MFID[28:24]				
		23:16	MFID[23:16]							
		15:8	MFID[15:8]							
		7:0	MFID[7:0]							
0x02F0	CAN_MSR7	31:24								MMI
		23:16	MRDY	MABT		MRTR	MDLC[3:0]			
		15:8	MTIMESTAMP[15:8]							
		7:0	MTIMESTAMP[7:0]							
0x02F4	CAN_MDL7	31:24	MDL[31:24]							
		23:16	MDL[23:16]							
		15:8	MDL[15:8]							
		7:0	MDL[7:0]							
0x02F8	CAN_MDH7	31:24	MDH[31:24]							
		23:16	MDH[23:16]							
		15:8	MDH[15:8]							
		7:0	MDH[7:0]							
0x02FC	CAN_MCR7	31:24								
		23:16	MTCR	MACR		MRTR	MDLC[3:0]			
		15:8								
		7:0								

Downloaded from Arrow.com.

49.9.1 CAN Mode Register

Name: CAN_MR
Offset: 0x0000
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the CAN Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DRPT	TIMFRZ	TTM	TEOF	OVL	ABM	LPM	CANEN
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – DRPT Disable Repeat

Value	Description
0	When a transmit mailbox loses the bus arbitration, the transfer request remains pending.
1	When a transmit mailbox loses the bus arbitration, the transfer request is automatically aborted. It automatically raises the MABT and MRDT flags in the corresponding CAN_MSRx.

Bit 6 – TIMFRZ Enable Timer Freeze

Value	Description
0	The internal timer continues to be incremented after it reached 0xFFFF.
1	The internal timer stops incrementing after reaching 0xFFFF. It is restarted after a timer reset. See Freezing the Internal Timer Counter.

Bit 5 – TTM Disable/Enable Time-triggered Mode

Value	Description
0	Time-triggered mode is disabled.
1	Time-triggered mode is enabled.

Bit 4 – TEOF Timestamp messages at each end of Frame

Value	Description
0	The value of CAN_TIM is captured in the CAN_TIMESTP register at each Start Of Frame.
1	The value of CAN_TIM is captured in the CAN_TIMESTP register at each End Of Frame.

Bit 3 – OVL Disable/Enable Overload Frame

Value	Description
0	No overload frame is generated.
1	An overload frame is generated after each successful reception for mailboxes configured in Receive with/without overwrite Mode, Producer and Consumer.

Downloaded from Arrow.com.

Bit 2 – ABM Disable/Enable Autobaud/Listen mode

Value	Description
0	Disable Autobaud/listen mode.
1	Enable Autobaud/listen mode.

Bit 1 – LPM Disable/Enable Low-power Mode

CAN controller enters Low-power mode once all pending messages have been transmitted.

Value	Description
0	Disable Low-power mode.
1	Enable Low-power mode.

Bit 0 – CANEN CAN Controller Enable

Value	Description
0	The CAN Controller is disabled.
1	The CAN Controller is enabled.

Downloaded from Arrow.com.

49.9.2 CAN Interrupt Enable Register

Name: CAN_IER
Offset: 0x0004
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
				BERR	FERR	AERR	SERR	CERR
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	TSTP	TOVF	WAKEUP	SLEEP	BOFF	ERRP	WARN	ERRA
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 28 – BERR Bit Error Interrupt Enable

Value	Description
0	No effect.
1	Enable Bit Error interrupt.

Bit 27 – FERR Form Error Interrupt Enable

Value	Description
0	No effect.
1	Enable Form Error interrupt.

Bit 26 – AERR Acknowledgment Error Interrupt Enable

Value	Description
0	No effect.
1	Enable Acknowledgment Error interrupt.

Bit 25 – SERR Stuffing Error Interrupt Enable

Value	Description
0	No effect.
1	Enable Stuffing Error interrupt.

Bit 24 – CERR CRC Error Interrupt Enable

Value	Description
0	No effect.
1	Enable CRC Error interrupt.

Bit 23 – TSTP TimeStamp Interrupt Enable

Value	Description
0	No effect.

Downloaded from Arrow.com.

Value	Description
1	Enable TSTP interrupt.

Bit 22 – TOVF Timer Overflow Interrupt Enable

Value	Description
0	No effect.
1	Enable TOVF interrupt.

Bit 21 – WAKEUP Wakeup Interrupt Enable

Value	Description
0	No effect.
1	Enable SLEEP interrupt.

Bit 20 – SLEEP Sleep Interrupt Enable

Value	Description
0	No effect.
1	Enable SLEEP interrupt.

Bit 19 – BOFF Bus Off Mode Interrupt Enable

Value	Description
0	No effect.
1	Enable BOFF interrupt.

Bit 18 – ERRP Error Passive Mode Interrupt Enable

Value	Description
0	No effect.
1	Enable ERRP interrupt.

Bit 17 – WARN Warning Limit Interrupt Enable

Value	Description
0	No effect.
1	Enable WARN interrupt.

Bit 16 – ERRA Error Active Mode Interrupt Enable

Value	Description
0	No effect.
1	Enable ERRA interrupt.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – MBx Mailbox x Interrupt Enable

Value	Description
0	No effect.
1	Enable Mailbox x interrupt.

Downloaded from Arrow.com.

49.9.3 CAN Interrupt Disable Register

Name: CAN_IDR
Offset: 0x0008
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
				BERR	FERR	AERR	SERR	CERR
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	TSTP	TOVF	WAKEUP	SLEEP	BOFF	ERRP	WARN	ERRA
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 28 – BERR Bit Error Interrupt Disable

Value	Description
0	No effect.
1	Disable Bit Error interrupt.

Bit 27 – FERR Form Error Interrupt Disable

Value	Description
0	No effect.
1	Disable Form Error interrupt.

Bit 26 – AERR Acknowledgment Error Interrupt Disable

Value	Description
0	No effect.
1	Disable Acknowledgment Error interrupt.

Bit 25 – SERR Stuffing Error Interrupt Disable

Value	Description
0	No effect.
1	Disable Stuffing Error interrupt.

Bit 24 – CERR CRC Error Interrupt Disable

Value	Description
0	No effect.
1	Disable CRC Error interrupt.

Bit 23 – TSTP TimeStamp Interrupt Disable

Value	Description
0	No effect.

Downloaded from Arrow.com.

Value	Description
1	Disable TSTP interrupt.

Bit 22 – TOVF Timer Overflow Interrupt

Value	Description
0	No effect.
1	Disable TOVF interrupt.

Bit 21 – WAKEUP Wakeup Interrupt Disable

Value	Description
0	No effect.
1	Disable WAKEUP interrupt.

Bit 20 – SLEEP Sleep Interrupt Disable

Value	Description
0	No effect.
1	Disable SLEEP interrupt.

Bit 19 – BOFF Bus Off Mode Interrupt Disable

Value	Description
0	No effect.
1	Disable BOFF interrupt.

Bit 18 – ERRP Error Passive Mode Interrupt Disable

Value	Description
0	No effect.
1	Disable ERRP interrupt.

Bit 17 – WARN Warning Limit Interrupt Disable

Value	Description
0	No effect.
1	Disable WARN interrupt.

Bit 16 – ERRA Error Active Mode Interrupt Disable

Value	Description
0	No effect.
1	Disable ERRA interrupt.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – MBx Mailbox x Interrupt Disable

Value	Description
0	No effect.
1	Disable Mailbox x interrupt.

Downloaded from Arrow.com.

49.9.4 CAN Interrupt Mask Register

Name: CAN_IMR
Offset: 0x000C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
				BERR	FERR	AERR	SERR	CERR
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TSTP	TOVF	WAKEUP	SLEEP	BOFF	ERRP	WARN	ERRA
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 28 – BERR Bit Error Interrupt Mask

Value	Description
0	Bit Error interrupt is disabled.
1	Bit Error interrupt is enabled.

Bit 27 – FERR Form Error Interrupt Mask

Value	Description
0	Form Error interrupt is disabled.
1	Form Error interrupt is enabled.

Bit 26 – AERR Acknowledgment Error Interrupt Mask

Value	Description
0	Acknowledgment Error interrupt is disabled.
1	Acknowledgment Error interrupt is enabled.

Bit 25 – SERR Stuffing Error Interrupt Mask

Value	Description
0	Bit Stuffing Error interrupt is disabled.
1	Bit Stuffing Error interrupt is enabled.

Bit 24 – CERR CRC Error Interrupt Mask

Value	Description
0	CRC Error interrupt is disabled.
1	CRC Error interrupt is enabled.

Bit 23 – TSTP Timestamp Interrupt Mask

Value	Description
0	TSTP interrupt is disabled.

Downloaded from Arrow.com.

Value	Description
1	TSTP interrupt is enabled.

Bit 22 – TOVF Timer Overflow Interrupt Mask

Value	Description
0	TOVF interrupt is disabled.
1	TOVF interrupt is enabled.

Bit 21 – WAKEUP Wakeup Interrupt Mask

Value	Description
0	WAKEUP interrupt is disabled.
1	WAKEUP interrupt is enabled.

Bit 20 – SLEEP Sleep Interrupt Mask

Value	Description
0	SLEEP interrupt is disabled.
1	SLEEP interrupt is enabled.

Bit 19 – BOFF Bus Off Mode Interrupt Mask

Value	Description
0	BOFF interrupt is disabled.
1	BOFF interrupt is enabled.

Bit 18 – ERRP Error Passive Mode Interrupt Mask

Value	Description
0	ERRP interrupt is disabled.
1	ERRP interrupt is enabled.

Bit 17 – WARN Warning Limit Interrupt Mask

Value	Description
0	Warning Limit interrupt is disabled.
1	Warning Limit interrupt is enabled.

Bit 16 – ERRA Error Active Mode Interrupt Mask

Value	Description
0	ERRA interrupt is disabled.
1	ERRA interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – MBx Mailbox x Interrupt Mask

Value	Description
0	Mailbox x interrupt is disabled.
1	Mailbox x interrupt is enabled.

Downloaded from Arrow.com.

49.9.5 CAN Status Register

Name: CAN_SR
Offset: 0x0010
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	OVLSY	TBSY	RBSY	BERR	FERR	AERR	SERR	CERR
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TSTP	TOVF	WAKEUP	SLEEP	BOFF	ERRP	WARN	ERRA
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 31 – OVLSY Overload busy
It is automatically reset when the bus is not transmitting an overload frame.

Value	Description
0	CAN transmitter is not transmitting an overload frame.
1	CAN transmitter is transmitting a overload frame.

Bit 30 – TBSY Transmitter Busy
Transmitter busy. This status bit is set by hardware while CAN transmitter is generating a frame (remote, data, overload or error frame). It is automatically reset when CAN is not transmitting.

Value	Description
0	CAN transmitter is not transmitting a frame.
1	CAN transmitter is transmitting a frame.

Bit 29 – RBSY Receiver Busy
Receiver busy. This status bit is set by hardware while CAN receiver is acquiring or monitoring a frame (remote, data, overload or error frame). It is automatically reset when CAN is not receiving.

Value	Description
0	CAN receiver is not receiving a frame.
1	CAN receiver is receiving a frame.

Bit 28 – BERR Bit Error (automatically cleared by reading CAN_SR)
A bit error is set when the bit value monitored on the line is different from the bit value sent.

Value	Description
0	No bit error occurred during a previous transfer.
1	A bit error occurred during a previous transfer.

Bit 27 – FERR Form Error (automatically cleared by reading CAN_SR)
A form error results from violations on one or more of the fixed form of the following bit fields:

Downloaded from Arrow.com.

Value	Description
0	No form error occurred during a previous transfer
1	A form error occurred during a previous transfer – CRC delimiter – ACK delimiter – End of frame – Error delimiter – Overload delimiter

Bit 26 – AERR Acknowledgment Error (automatically cleared by reading CAN_SR)
An acknowledgment error is detected when no detection of the dominant bit in the acknowledge slot occurs.

Value	Description
0	No acknowledgment error occurred during a previous transfer.
1	An acknowledgment error occurred during a previous transfer.

Bit 25 – SERR Mailbox Stuffing Error (automatically cleared by reading CAN_SR)
A form error results from the detection of more than five consecutive bit with the same polarity.

Value	Description
0	No stuffing error occurred during a previous transfer.
1	A stuffing error occurred during a previous transfer.

Bit 24 – CERR Mailbox CRC Error (automatically cleared by reading CAN_SR)
A CRC error has been detected during last reception.

Value	Description
0	No CRC error occurred during a previous transfer.
1	A CRC error occurred during a previous transfer.

Bit 23 – TSTP Timestamp (automatically cleared by reading CAN_SR)

Value	Description
0	No bus activity has been detected.
1	A start of frame or an end of frame has been detected (according to the TEOF field in the CAN_MR).

Bit 22 – TOVF Timer Overflow (automatically cleared by reading CAN_SR)

Value	Description
0	The timer has not rolled-over FFFFh to 0000h.
1	The timer rolls-over FFFFh to 0000h.

Bit 21 – WAKEUP CAN Controller is not in Low-power Mode
When a WAKEUP event occurs, the CAN controller is synchronized with the bus activity. Messages can be transmitted or received. The CAN controller clock must be available when a WAKEUP event occurs. This flag is automatically reset when the CAN Controller enters Low-power mode.

Value	Description
0	CAN controller is in Low-power mode.
1	CAN controller is not in Low-power mode.

Bit 20 – SLEEP CAN Controller in Low-power Mode
This flag is automatically reset when Low-power mode is disabled

Value	Description
0	CAN controller is not in Low-power mode.
1	CAN controller is in Low-power mode.

Bit 19 – BOFF Bus Off Mode (automatically cleared by reading CAN_SR)
This flag is set depending on TEC counter value. A node is in bus off state when TEC counter is greater or equal to 256 (decimal).

Downloaded from Arrow.com.

Value	Description
0	CAN controller has not reached Bus Off Mode.
1	CAN controller has reached Bus Off Mode since the last read of CAN_SR.
0	CAN controller is not in Bus Off Mode.
1	CAN controller is in Bus Off Mode.

Bit 18 – ERRP Error Passive Mode (automatically cleared by reading CAN_SR)
This flag is set depending on TEC and REC counters values.
A node is in error passive state when TEC counter is greater or equal to 128 (decimal) or when the REC counter is greater or equal to 128 (decimal).

Value	Description
0	CAN controller has not reached Error Passive Mode since the last read of CAN_SR.
1	CAN controller has reached Error Passive Mode since the last read of CAN_SR.
0	CAN controller is not in Error Passive Mode.
1	CAN controller is in Error Passive Mode.

Bit 17 – WARN Warning Limit (automatically cleared by reading CAN_SR)
This flag is set depending on TEC and REC counter values. It is set when at least one of the counter values has reached a value greater or equal to 96.

Value	Description
0	CAN controller Warning Limit has not been reached since the last read of CAN_SR.
1	CAN controller Warning Limit has been reached since the last read of CAN_SR.
0	CAN controller Warning Limit is not reached.
1	CAN controller Warning Limit is reached.

Bit 16 – ERRA Error Active Mode (automatically cleared by reading CAN_SR)
This flag is set depending on TEC and REC counter values. It is set when a node is neither in Error Passive Mode nor in Bus Off Mode.

Value	Description
0	CAN controller has not reached Error Active Mode since the last read of CAN_SR.
1	CAN controller has reached Error Active Mode since the last read of CAN_SR.
0	CAN controller is not in Error Active Mode.
1	CAN controller is in Error Active Mode.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – MBx Mailbox x Event
An event corresponds to MRDY, MABT bits in the CAN_MSRx.

Value	Description
0	No event occurred on Mailbox x.
1	An event occurred on Mailbox x.

Downloaded from Arrow.com.

49.9.6 CAN Baudrate Register

Name: CAN_BR
Offset: 0x0014
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the CAN Write Protection Mode Register.

Any modification on one of the fields of the CAN_BR must be done while CAN module is disabled.

To compute the different bit timings, see CAN Bit Timing Configuration.

Bit	31	30	29	28	27	26	25	24
								SMP
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
		BRP[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			SJW[1:0]			PROPAG[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
		PHASE1[2:0]				PHASE2[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit 24 – SMP Sampling Mode
0 (ONCE): The incoming bit stream is sampled once at sample point.
1 (THREE): The incoming bit stream is sampled three times with a period of a peripheral clock, centered on sample point.
SMP Sampling Mode is automatically disabled if BRP = 0.

Bits 22:16 – BRP[6:0] Baudrate Prescaler
This field allows user to program the period of the CAN system clock to determine the individual bit timing.
$t_{CSC} = (BRP+1)/t_{peripheral\ clock}$

The BRP field must be within the range [1, 0x7F], i.e., BRP = 0 is not authorized.

Bits 13:12 – SJW[1:0] Re-synchronization Jump Width
To compensate for phase shifts between clock oscillators of different controllers on bus. The controller must re-synchronize on any relevant signal edge of the current transmission. The synchronization jump width defines the maximum of clock cycles a bit period may be shortened or lengthened by re-synchronization.
$t_{SJW} = t_{CSC} \times (SJW+1)$

Bits 10:8 – PROPAG[2:0] Programming Time Segment
This part of the bit time is used to compensate for the physical delay times within the network.
$t_{PRS} = t_{CSC} \times (PROPAG+1)$

Bits 6:4 – PHASE1[2:0] Phase 1 Segment
This phase is used to compensate for edge phase error.
$t_{PHS1} = t_{CSC} \times (PHASE1+1)$

Downloaded from Arrow.com.

Bits 2:0 – PHASE2[2:0] Phase 2 Segment
This phase is used to compensate the edge phase error.
$t_{PHS2} = t_{CSC} \times (PHASE2+1)$

Note: PHASE2 value must be different from 0.

Downloaded from Arrow.com.

49.9.7 CAN Timer Register

Name: CAN_TIM
Offset: 0x0018
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TIMER[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TIMER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TIMER[15:0] Timer
This field represents the internal CAN controller 16-bit timer value.

Downloaded from Arrow.com.

49.9.8 CAN Timestamp Register

Name: CAN_TIMESTP
Offset: 0x001C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	MTIMESTAMP[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MTIMESTAMP[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – MTIMESTAMP[15:0] Timestamp
This field carries the value of the internal CAN controller 16-bit timer value at the start or end of frame.
If the TEOF bit is cleared in the CAN_MR, the internal Timer Counter value is captured in the MTIMESTAMP field at each start of frame else the value is captured at each end of frame. When the value is captured, the TSTP flag is set in the CAN_SR. If the TSTP mask in the CAN_IMR is set, an interrupt is generated while TSTP flag is set in the CAN_SR. The TSTP flag is cleared by reading the CAN_SR.
Note: The CAN_TIMESTP register is reset when the CAN is disabled then enabled via the CANEN bit in the CAN_MR.

Downloaded from Arrow.com.

49.9.9 CAN Error Counter Register

Name: CAN_ECR
Offset: 0x0020
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
								TEC[8]
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
	TEC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	REC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 24:16 – TEC[8:0] Transmit Error Counter
When a transmitter sends an ERROR FLAG, TEC is increased by 8 except when:
– the transmitter is error passive and detects an ACKNOWLEDGMENT ERROR because of not detecting a dominant ACK and does not detect a dominant bit while sending its PASSIVE ERROR FLAG.
– the transmitter sends an ERROR FLAG because a STUFF ERROR occurred during arbitration and should have been recessive and has been sent as recessive but monitored as dominant.
When a transmitter detects a BIT ERROR while sending an ACTIVE ERROR FLAG or an OVERLOAD FLAG, the TEC will be increased by 8.
Any node tolerates up to 7 consecutive dominant bits after sending an ACTIVE ERROR FLAG, PASSIVE ERROR FLAG or OVERLOAD FLAG. After detecting the 14th consecutive dominant bit (in case of an ACTIVE ERROR FLAG or an OVERLOAD FLAG) or after detecting the 8th consecutive dominant bit following a PASSIVE ERROR FLAG, and after each sequence of additional eight consecutive dominant bits every transmitter increases its TEC by 8.
After a successful transmission the TEC is decreased by 1 unless it was already 0.

Bits 7:0 – REC[7:0] Receive Error Counter
When a receiver detects an error, REC will be increased by one, except when the detected error is a BIT ERROR while sending an ACTIVE ERROR FLAG or an OVERLOAD FLAG.
When a receiver detects a dominant bit as the first bit after sending an ERROR FLAG, REC is increased by 8.
When a receiver detects a BIT ERROR while sending an ACTIVE ERROR FLAG, REC is increased by 8.
Any node tolerates up to 7 consecutive dominant bits after sending an ACTIVE ERROR FLAG, PASSIVE ERROR FLAG or OVERLOAD FLAG. After detecting the 14th consecutive dominant bit (in case of an ACTIVE ERROR FLAG or an OVERLOAD FLAG) or after detecting the 8th consecutive dominant bit following a PASSIVE ERROR FLAG, and after each sequence of additional eight consecutive dominant bits, each receiver increases its REC by 8.
After successful reception of a message, REC is decreased by 1 if it was between 1 and 127. If REC was 0, it stays 0, and if it was greater than 127, then it is set to a value between 119 and 127.

Downloaded from Arrow.com.

49.9.10 CAN Transfer Command Register

Name: CAN_TCR
Offset: 0x0024
Reset: –
Property: Write-only

This register initializes several transfer requests at the same time.

Bit	31	30	29	28	27	26	25	24
	TIMRST							
Access	W							
Reset	–							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 31 – TIMRST Timer Reset
Resets the internal timer counter. If the internal timer counter is frozen, this command automatically re-enables it.
This command is useful in Time Triggered mode.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – MBx Transfer Request for Mailbox x
This flag clears the MRDY and MABT flags in the corresponding CAN_MSRx.
When several mailboxes are requested to be transmitted simultaneously, they are transmitted in turn, starting with the mailbox with the highest priority. If several mailboxes have the same priority, then the mailbox with the lowest number is sent first (i.e., MB0 will be transferred before MB1).

Mailbox Object Type	Description
Receive	It receives the next message.
Receive with overwrite	This triggers a new reception.
Transmit	Sends data prepared in the mailbox as soon as possible.
Consumer	Sends a remote frame.
Producer	Sends data prepared in the mailbox after receiving a remote frame from a consumer.

Downloaded from Arrow.com.

49.9.11 CAN Abort Command Register

Name: CAN_ACR
Offset: 0x0028
Reset: –
Property: Write-only

This register initializes several abort requests at the same time.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – MBx Abort Request for Mailbox x
It is possible to set the MACR field (in the CAN_MCRx) for each mailbox.

Mailbox Object Type	Description
Receive	No action
Receive with overwrite	No action
Transmit	Cancels transfer request if the message has not been transmitted to the CAN transceiver.
Consumer	Cancels the current transfer before the remote frame has been sent.
Producer	Cancels the current transfer. The next remote frame is not serviced.

Downloaded from Arrow.com.

49.9.12 CAN Write Protection Mode Register

Name: CAN_WPMR
Offset: 0x00E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key Password

Value	Name	Description
0x43414E	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0

Bit 0 – WPEN Write Protection Enable
See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x43414E (“CAN” written in ASCII)
1	Enables the write protection if WPKEY corresponds to 0x43414E (“CAN” written in ASCII)

Downloaded from Arrow.com.

49.9.13 CAN Write Protection Status Register

Name: CAN_WPSR
Offset: 0x00E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the CAN_WPSR.
1	A write protection violation has occurred since the last read of the CAN_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

49.9.14 CAN Message Mode Register

Name: CAN_MMRx
Offset: 0x0200 + x*0x20 [x=0..7]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the CAN Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
						MOT[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
					PRIOR[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MTIMEMARK[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MTIMEMARK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:24 – MOT[2:0] Mailbox Object Type
This field allows the user to define the type of the mailbox. All mailboxes are independently configurable. Five different types are possible for each mailbox.

Value	Name	Description
0	MB_DISABLED	Mailbox is disabled. This prevents receiving or transmitting any messages with this mailbox.
1	MB_RX	Reception Mailbox. Mailbox is configured for reception. If a message is received while the mailbox data register is full, it is discarded.
2	MB_RX_OVERWRITE	Reception mailbox with overwrite. Mailbox is configured for reception. If a message is received while the mailbox is full, it overwrites the previous message.
3	MB_TX	Transmit mailbox. Mailbox is configured for transmission.
4	MB_CONSUMER	Consumer Mailbox. Mailbox is configured in reception but behaves as a Transmit Mailbox, i.e., it sends a remote frame and waits for an answer.
5	MB_PRODUCER	Producer Mailbox. Mailbox is configured in transmission but also behaves like a reception mailbox, i.e., it waits to receive a Remote Frame before sending its contents.
6	Reserved	

Bits 19:16 – PRIOR[3:0] Mailbox Priority
This field has no effect in receive and receive with overwrite modes. In these modes, the mailbox with the lowest number is serviced first.
When several mailboxes try to transmit a message at the same time, the mailbox with the highest priority is serviced first. If several mailboxes have the same priority, the mailbox with the lowest number is serviced first (i.e., MBx0 is serviced before MBx 15 if they have the same priority).

Downloaded from Arrow.com.

Bits 15:0 – MTIMEMARK[15:0] Mailbox Timemark
This field is active in Time Triggered Mode. Transmit operations are allowed when the internal timer counter reaches the Mailbox Timemark. See Transmitting within a Time Window.
In Timestamp Mode, MTIMEMARK is set to 0.

Downloaded from Arrow.com.

49.9.15 CAN Message Acceptance Mask Register

Name: CAN_MAMx
Offset: 0x0204 + x*0x20 [x=0..7]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the CAN Write Protection Mode Register.

To prevent concurrent access with the internal CAN core, the application must disable the mailbox before writing to CAN_MAMx registers.

Bit	31	30	29	28	27	26	25	24
			MIDE	MIDvA[10:6]				
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MIDvA[5:0]						MIDvB[17:16]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MIDvB[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MIDvB[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 29 – MIDE Identifier Version

Value	Description
0	Compares IDvA from the received frame with the CAN_MIDx register masked with CAN_MAMx register.
1	Compares IDvA and IDvB from the received frame with the CAN_MIDx register masked with CAN_MAMx register.

Bits 28:18 – MIDvA[10:0] Identifier for standard frame mode
Acceptance mask for corresponding field of the message IDvA register of the mailbox.

Value	Description
0	The corresponding message ID bit is not masked
1	The corresponding message ID bit is masked

Bits 17:0 – MIDvB[17:0] Complementary bits for identifier in extended frame mode
Acceptance mask for corresponding field of the message IDvB register of the mailbox.

Value	Description
0	The corresponding message ID bit is not masked
1	The corresponding message ID bit is masked

Downloaded from Arrow.com.

49.9.16 CAN Message ID Register

Name: CAN_MIDx
Offset: 0x0208 + x*0x20 [x=0..7]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the CAN Write Protection Mode Register.

To prevent concurrent access with the internal CAN core, the application must disable the mailbox before writing to CAN_MIDx registers.

Bit	31	30	29	28	27	26	25	24
			MIDE	MIDvA[10:6]				
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MIDvA[5:0]						MIDvB[17:16]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MIDvB[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MIDvB[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 29 – MIDE Identifier Version
This bit allows the user to define the version of messages processed by the mailbox. If set, mailbox is dealing with version 2.0 Part B messages; otherwise, mailbox is dealing with version 2.0 Part A messages.

Bits 28:18 – MIDvA[10:0] Identifier for standard frame mode

Bits 17:0 – MIDvB[17:0] Complementary bits for identifier in extended frame mode
If MIDE is cleared, MIDvB value is 0.

Downloaded from Arrow.com.

49.9.17 CAN Message Family ID Register

Name: CAN_MFIDx
Offset: 0x020C + x*0x20 [x=0..7]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				MFID[28:24]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MFID[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MFID[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MFID[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 28:0 – MFID[28:0] Family ID
This field contains the concatenation of CAN_MIDx register bits masked by the CAN_MAMx register. This field is useful to speed up message ID decoding. The message acceptance procedure is described below.
As an example:

```
CAN_MIDx = 0x305A4321
CAN_MAMx = 0x3FF0F0FF
CAN_MFIDx = 0x000000A3
```


Downloaded from Arrow.com.

49.9.18 CAN Message Status Register

Name: CAN_MSRx
Offset: 0x0210 + x*0x20 [x=0..7]
Reset: 0
Property: Read/Write

These register fields are updated each time a message transfer is received or aborted.

WARNING MRTR and MDLC state depends partly on the mailbox object type.

Bit	31	30	29	28	27	26	25	24
								MMI
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
	MRDY	MABT		MRTR	MDLC[3:0]			
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MTIMESTAMP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MTIMESTAMP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 24 – MMI Mailbox Message Ignored (cleared by reading CAN_MSRx)

Mailbox Object Type	Description
Receive	Set when at least two messages intended for the mailbox have been sent. The first one is available in the mailbox data register. Others have been ignored. A mailbox with a lower priority may have accepted the message.
Receive with overwrite	Set when at least two messages intended for the mailbox have been sent. The last one is available in the mailbox data register. Previous ones have been lost.
Transmit	Reserved
Consumer	A remote frame has been sent by the mailbox but several messages have been received. The first one is available in the mailbox data register. Others have been ignored. Another mailbox with a lower priority may have accepted the message.
Producer	A remote frame has been received, but no data are available to be sent.

Value	Description
0	No message has been ignored during the previous transfer
1	At least one message has been ignored during the previous transfer

Downloaded from Arrow.com.

Bit 23 – MRDY Mailbox Ready (cleared by writing MTCR or MACR in the CAN_MCRx)
An interrupt is triggered when MRDY is set.

Mailbox Object Type	Description
Receive	At least one message has been received since the last mailbox transfer order. Data from the first frame received can be read in the CAN_MDxx registers. After setting the MOT field in the CAN_MMR, MRDY is reset to 0.
Receive with overwrite	At least one frame has been received since the last mailbox transfer order. Data from the last frame received can be read in the CAN_MDxx registers. After setting the MOT field in the CAN_MMR, MRDY is reset to 0.
Transmit	Mailbox data have been transmitted. After setting the MOT field in the CAN_MMR, MRDY is reset to 1.
Consumer	At least one message has been received since the last mailbox transfer order. Data from the first message received can be read in the CAN_MDxx registers. After setting the MOT field in the CAN_MMR, MRDY is reset to 0.
Producer	A remote frame has been received, mailbox data have been transmitted. After setting the MOT field in the CAN_MMR, MRDY is reset to 1.

Value	Description
0	Mailbox data registers can not be read/written by the software application. CAN_MDx are locked by the CAN_MDx.
1	Mailbox data registers can be read/written by the software application.

Bit 22 – MABT Mailbox Message Abort (cleared by writing MTCR or MACR in the CAN_MCRx)
An interrupt is triggered when MABT is set.

Mailbox Object Type	Description
Receive	Reserved
Receive with overwrite	Reserved
Transmit	Previous transfer has been aborted
Consumer	The remote frame transfer request has been aborted.
Producer	The response to the remote frame transfer has been aborted.

Value	Description
0	Previous transfer is not aborted.
1	Previous transfer has been aborted.

Bit 20 – MRTR Mailbox Remote Transmission Request

Mailbox Object Type	Description
Receive	The first frame received has the RTR bit set.
Receive with overwrite	The last frame received has the RTR bit set.
Transmit	Reserved
Consumer	Reserved. After setting the MOT field in the CAN_MMR, MRTR is reset to 1.
Producer	Reserved. After setting the MOT field in the CAN_MMR, MRTR is reset to 0.

Downloaded from Arrow.com.

Bits 19:16 – MDLC[3:0] Mailbox Data Length Code

Mailbox Object Type	Description
Receive	Length of the first mailbox message received
Receive with overwrite	Length of the last mailbox message received
Transmit	No action
Consumer	Length of the mailbox message received
Producer	Length of the mailbox message to be sent after the remote frame reception

Bits 15:0 – MTIMESTAMP[15:0] Timer Value
This field is updated only when time-triggered operations are disabled (TTM cleared in CAN_MR). If the field CAN_MR.TEOF is cleared, TIMESTAMP is the internal timer value at the start of frame of the last message received or sent by the mailbox. If the field CAN_MR.TEOF is set, TIMESTAMP is the internal timer value at the end of frame of the last message received or sent by the mailbox.
In Time Triggered Mode, MTIMESTAMP is set to 0.

Downloaded from Arrow.com.

49.9.19 CAN Message Data Low Register

Name: CAN_MDLx
Offset: 0x0214 + x*0x20 [x=0..7]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	MDL[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MDL[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MDL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MDL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – MDL[31:0] Message Data Low Value
When MRDY bit is set in the CAN_MSRx, the lower 32 bits of a received message can be read or written by the software application. Otherwise, the MDL value is locked by the CAN controller to send/receive a new message.
In Receive with overwrite, the CAN controller may modify MDL value while the software application reads MDH and MDL registers. To check that MDH and MDL do not belong to different messages, the application has to check the MMI bit in the CAN_MSRx. In this mode, the software application must re-read CAN_MDH and CAN_MDL, while the MMI bit in the CAN_MSRx is set.
Bytes are received/sent on the bus in the following order:
1. CAN_MDL[7:0]
2. CAN_MDL[15:8]
3. CAN_MDL[23:16]
4. CAN_MDL[31:24]
5. CAN_MDH[7:0]
6. CAN_MDH[15:8]
7. CAN_MDH[23:16]
8. CAN_MDH[31:24]

Downloaded from Arrow.com.

49.9.20 CAN Message Data High Register

Name: CAN_MDHx
Offset: 0x0218 + x*0x20 [x=0..7]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	MDH[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MDH[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MDH[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MDH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – MDH[31:0] Message Data High Value
When MRDY bit is set in the CAN_MSRx, the upper 32 bits of a received message are read or written by the software application. Otherwise, the MDH value is locked by the CAN controller to send/receive a new message.
In Receive with overwrite, the CAN controller may modify MDH value while the software application reads MDH and MDL registers. To check that MDH and MDL do not belong to different messages, the application has to check the MMI bit in the CAN_MSRx. In this mode, the software application must re-read CAN_MDH and CAN_MDL, while the MMI bit in the CAN_MSRx is set.
Bytes are received/sent on the bus in the following order:
1. CAN_MDL[7:0]
2. CAN_MDL[15:8]
3. CAN_MDL[23:16]
4. CAN_MDL[31:24]
5. CAN_MDH[7:0]
6. CAN_MDH[15:8]
7. CAN_MDH[23:16]
8. CAN_MDH[31:24]

Downloaded from Arrow.com.

49.9.21 CAN Message Control Register

Name: CAN_MCRx
Offset: 0x021C + x*0x20 [x=0..7]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	MTCR	MACR		MRTR	MDLC[3:0]			
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 23 – MTCR Mailbox Transfer Command
This flag clears the MRDY and MABT flags in the CAN_MSRx.
When several mailboxes are requested to be transmitted simultaneously, they are transmitted in turn. The mailbox with the highest priority is serviced first. If several mailboxes have the same priority, the mailbox with the lowest number is serviced first (i.e., MBx0 will be serviced before MBx 15 if they have the same priority).
It is possible to set MTCR for several mailboxes at the same time by writing to the CAN_TCR.

Mailbox Object Type	Description
Receive	Allows the reception of the next message.
Receive with overwrite	Triggers a new reception.
Transmit	Sends data prepared in the mailbox as soon as possible.
Consumer	Sends a remote transmission frame.
Producer	Sends data prepared in the mailbox after receiving a remote frame from a Consumer.

Bit 22 – MACR Abort Request for Mailbox x
This flag clears the MRDY and MABT flags in the CAN_MSRx.
It is possible to set the MACR field for several mailboxes in the same time, setting several bits to the CAN_ACR.

Mailbox Object Type	Description
Receive	No action
Receive with overwrite	No action
Transmit	Cancels transfer request if the message has not been transmitted to the CAN transceiver.
Consumer	Cancels the current transfer before the remote frame has been sent.

Downloaded from Arrow.com.

Producer	Cancels the current transfer. The next remote frame will not be serviced.

Bit 20 – MRTR Mailbox Remote Transmission Request
Consumer situations can be handled automatically by setting the mailbox object type in Consumer. This requires only one mailbox.
It can also be handled using two mailboxes, one in reception, the other in transmission. The MRTR and the MTCR bits must be set in the same time.

Mailbox Object Type	Description
Receive	No action
Receive with overwrite	No action
Transmit	Sets the RTR bit in the sent frame
Consumer	No action, the RTR bit in the sent frame is set automatically
Producer	No action

Bits 19:16 – MDLC[3:0] Mailbox Data Length Code

Mailbox Object Type	Description
Receive	No action.
Receive with overwrite	No action.
Transmit	Length of the mailbox message.
Consumer	No action.
Producer	Length of the mailbox message to be sent after the remote frame reception.

Downloaded from Arrow.com.

50. Timer Counter (TC)

50.1 Description

A Timer Counter (TC) module includes three identical TC channels. The number of implemented TC modules is device-specific.

Each TC channel can be independently programmed to perform a wide range of functions including frequency measurement, event counting, interval measurement, pulse generation, delay timing and pulse width modulation.

Each channel has three external clock inputs, five internal clock inputs and two multipurpose input/output signals which can be configured by the user. Each channel drives an internal interrupt signal which can be programmed to generate processor interrupts.

The TC embeds a quadrature decoder (QDEC) connected in front of the timers and driven by TIOA0, TIOB0 and TIOB1 inputs. When enabled, the QDEC performs the input lines filtering, decoding of quadrature signals and connects to the timers/counters in order to read the position and speed of the motor through the user interface.

The TC block has two global registers which act upon all TC channels:

- Block Control register (TC_BCR)—allows channels to be started simultaneously with the same instruction
- Block Mode register (TC_BMR)—defines the external clock inputs for each channel, allowing them to be chained

50.2 Embedded Characteristics

- Total of Six Channels
- 32-bit Channel Size
- Wide Range of Functions Including:
  - Frequency measurement
  - Event counting
  - Interval measurement
  - Pulse generation
  - Delay timing
  - Pulse Width Modulation
  - Up/down capabilities
  - Quadrature decoder with real time filtering reports
  - 2-bit Gray up/down count for stepper motor
- Each Channel is User-Configurable and Contains:
  - Three external clock inputs
  - Five internal clock inputs
  - Two multipurpose input/output signals acting as trigger event
  - Trigger/capture events can be directly synchronized by PWM signals
- Internal Interrupt Signal
- Read of the Capture Registers by the DMAC
- Compare Event Fault Generation for PWM
- Register Write Protection
- Safety/Security Reports

Downloaded from Arrow.com.

50.3 Block Diagram

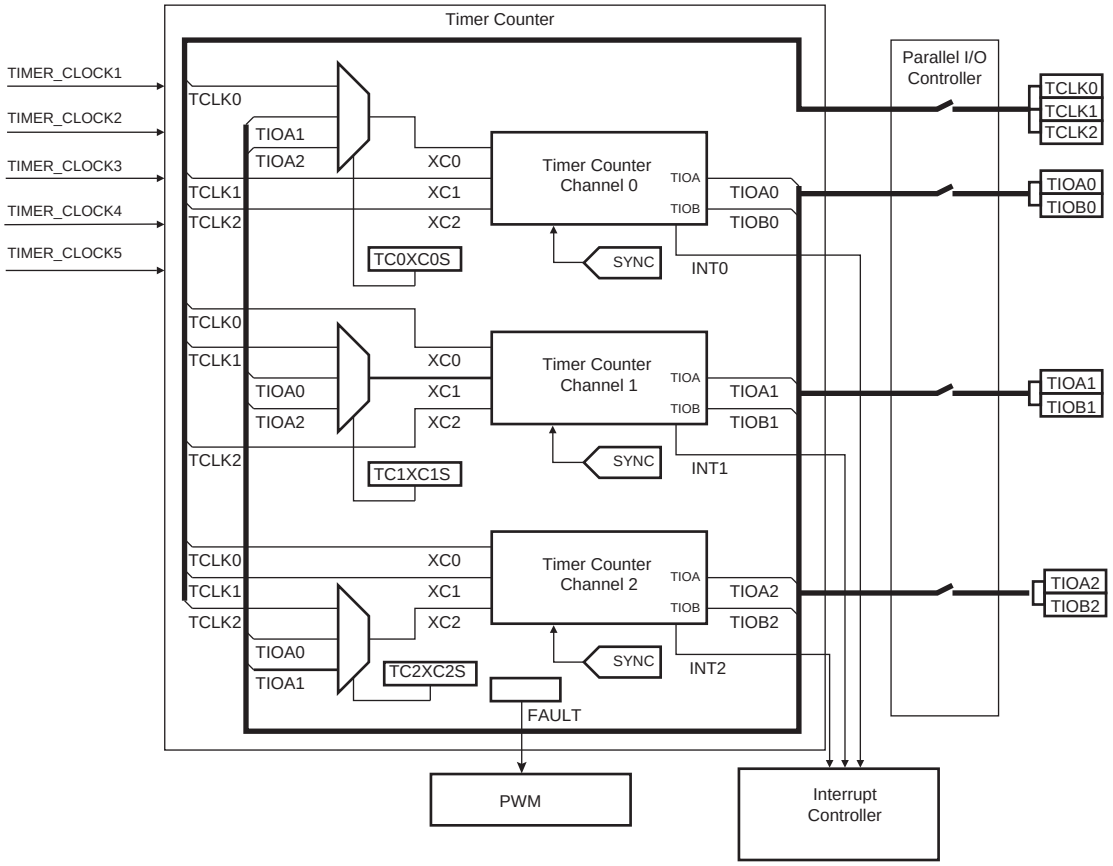
Table 50-1. Timer Counter Clock Assignment

Name	Definition
TIMER_CLOCK1	GCLK [17], GLCK[45]
TIMER_CLOCK2	MCK/8
TIMER_CLOCK3	MCK/32
TIMER_CLOCK4	MCK/128
TIMER_CLOCK5 (See Note)	MD_SLCK

Note:

1. When MD_SLCK is selected for Peripheral Clock (CSS = 0 in PMC Master Clock register), MD_SLCK input is equivalent to Peripheral Clock.
2. The GCLK [TC_ID] frequency must be at least three times lower than peripheral clock frequency.

Figure 50-1. TC Block Diagram



Note:
The QDEC connections are detailed in Figure 50-17.

Table 50-2. Channel Signal Description

Signal Name	Description
XC0, XC1, XC2	External Clock Inputs

Downloaded from Arrow.com.

.....continued	
Signal Name	**Description**
TIOAx	Capture Mode: Timer Counter Input Waveform Mode: Timer Counter Output
TIOBx	Capture Mode: Timer Counter Input Waveform Mode: Timer Counter Input/Output
INT	Interrupt Signal Output (internal signal)
SYNC	Synchronization Input Signal (from configuration register)

50.4 Pin List

Table 50-3. Pin List

Pin Name	Description	Type
TCLK0–TCLK2	External Clock Input	Input
TIOA0–TIOA2	I/O Line A	I/O
TIOB0–TIOB2	I/O Line B	I/O

50.5 Product Dependencies

50.5.1 I/O Lines

The pins used for interfacing the compliant external devices may be multiplexed with PIO lines. The programmer must first program the PIO controllers to assign the TC pins to their peripheral functions.

50.5.2 Power Management

The TC is clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the Timer Counter clock.

50.5.3 Interrupt Sources

The TC has an interrupt line connected to the interrupt controller. Handling the TC interrupt requires programming the interrupt controller before configuring the TC.

50.5.4 Synchronization Inputs from PWM

The TC has trigger/capture inputs internally connected to the PWM. See Synchronization with PWM and refer to the implementation of the Pulse Width Modulation (PWM) in this product.

50.5.5 Fault Output

The TC has the FAULT output internally connected to the fault input of PWM. See Fault Mode and refer to the implementation of the Pulse Width Modulation (PWM) in this product.

50.6 Functional Description

50.6.1 Description

All channels of the Timer Counter are independent and identical in operation except when the QDEC is enabled. The registers for channel programming are listed in Register Summary.

Downloaded from Arrow.com.

50.6.2 32-bit Counter

Each 32-bit channel is organized around a 32-bit counter. The value of the counter is incremented at each positive edge of the selected clock. When the counter has reached the value 2^{32}-1 and passes to zero, an overflow occurs and the COVFS bit in the Interrupt Status register (TC_SR) is set.

The current value of the counter is accessible in real time by reading the Counter Value register (TC_CV). The counter can be reset by a trigger. In this case, the counter value passes to zero on the next valid edge of the selected clock.

50.6.3 Clock Selection

At block level, input clock signals of each channel can be connected either to the external inputs TCLKx, or to the internal I/O signals TIOAx for chaining(1) by programming the Block Mode register (TC_BMR). See Clock Chaining Selection.

Each channel can independently select an internal or external clock source for its counter(2):

- External clock signals: XC0, XC1 or XC2
- Internal clock signals: GCLK [17], GLCK[45], MCK/8, MCK/32, MCK/128, MD_SLCK

This selection is made by the TCCLKS bits in the Channel Mode register (TC_CMRx).

The selected clock can be inverted with TC_CMRx.CLKI. This allows counting on the opposite edges of the clock.

The burst function allows the clock to be validated when an external signal is high. The BURST parameter in the TC_CMRx defines this signal (none, XC0, XC1, XC2). See Clock Selection.

Note:

1. In Waveform mode, to chain two timers, it is mandatory to initialize some parameters:
   – Configure TIOx outputs to 1 or 0 by writing the required value to TC_CMRx.ASWTRG.
   – Bit TC_BCR.SYNC must be written to 1 to start the channels at the same time.
2. In all cases, if an external clock or asynchronous internal clock GCLK [TC_ID] is used, the duration of each of its levels must be longer than the peripheral clock period, so the clock frequency will be at least 2.5 times lower than the peripheral clock.

Downloaded from Arrow.com.

Figure 50-2. Clock Chaining Selection

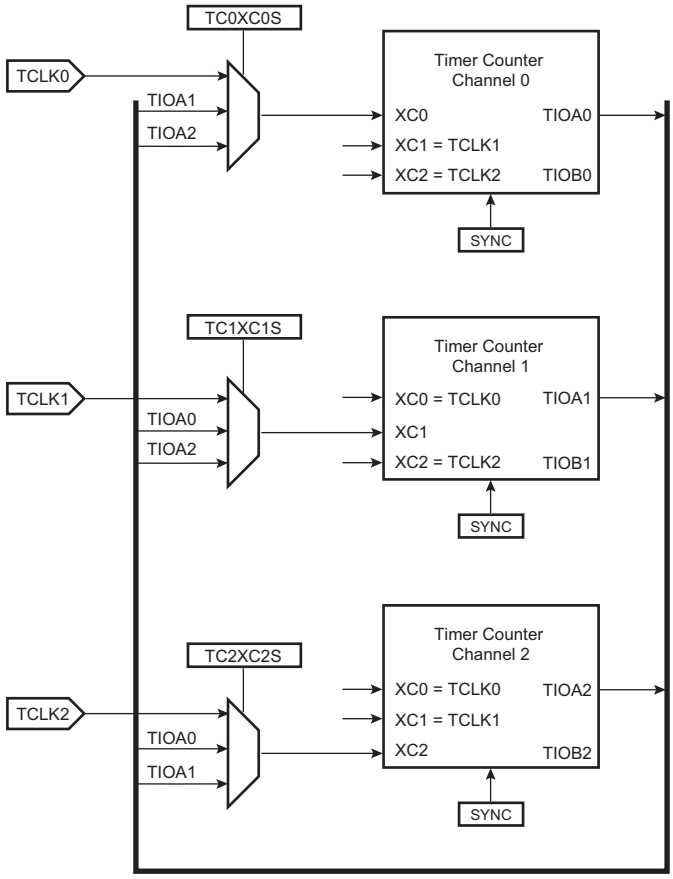
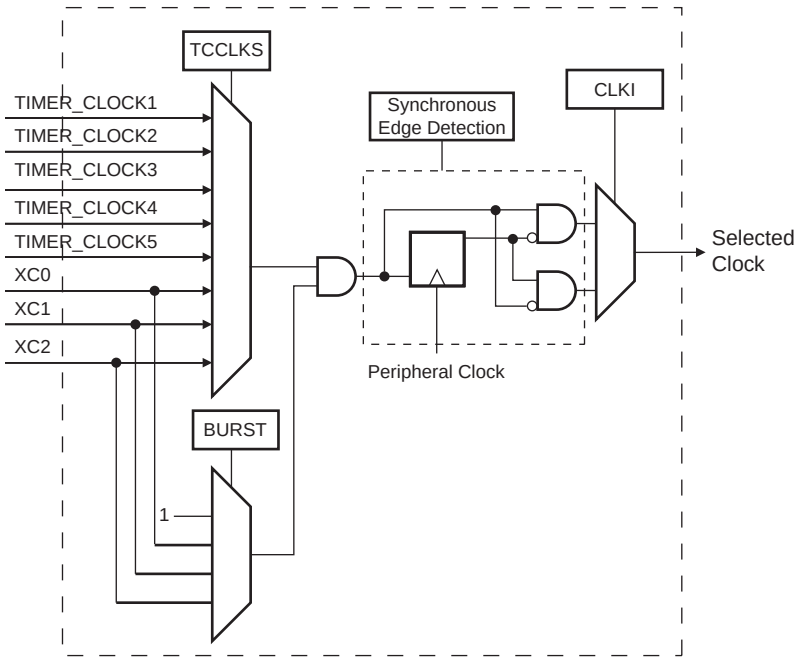


Figure 50-3. Clock Selection



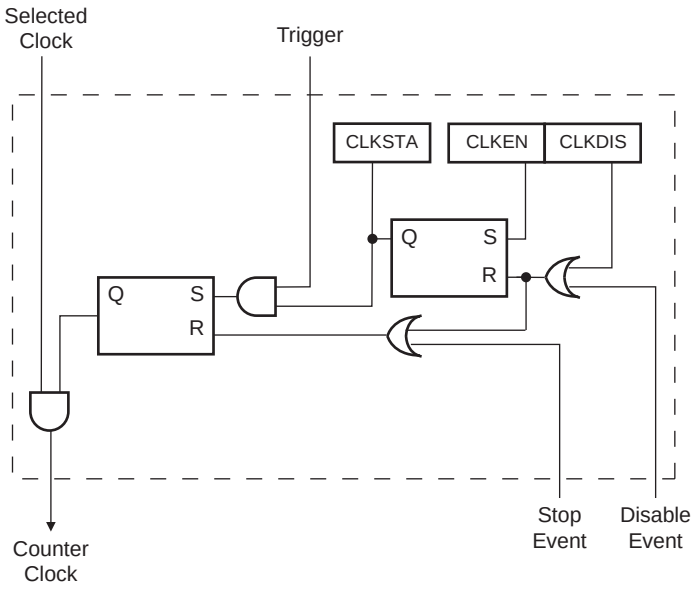
Downloaded from Arrow.com.

50.6.4 Clock Control

The clock of each counter can be controlled in two different ways: it can be enabled/disabled and started/stopped, as shown in the following figure.

- The clock can be enabled or disabled by the user with the CLKEN and the CLKDIS commands in the Channel Control register (TC_CCR). In Capture mode it can be disabled by an RB load event if TC_CMRx.LDBDIS is set to ‘1’. In Waveform mode, it can be disabled by an RC Compare event if TC_CMRx.CPCDIS is set to ‘1’. When disabled, the start or the stop actions have no effect: only a CLKEN command in the TC_CCR can reenable the clock. When the clock is enabled, TC_SR.CLKSTA is set.
- The clock can also be started or stopped: a trigger (software, synchro, external or compare) always starts the clock. The clock can be stopped by an RB load event in Capture mode (TC_CMRx.LDBSTOP = 1) or an RC compare event in Waveform mode (TC_CMRx.CPCSTOP = 1). The start and the stop commands are effective only if the clock is enabled.

Figure 50-4. Clock Control



50.6.5 Operating Modes

Each channel can operate independently in two different modes:

- Capture mode provides measurement on signals.
- Waveform mode provides wave generation.

The TC operating mode is programmed with TC_CMRx.WAVE.

In Capture mode, TIOAx and TIOBx are configured as inputs.

In Waveform mode, TIOAx is always configured to be an output and TIOBx is an output if it is not selected to be the external trigger.

50.6.6 Trigger

A trigger resets the counter and starts the counter clock. Three types of triggers are common to both modes, and a fourth external trigger is available to each mode.

Regardless of the trigger used, it will be taken into account at the following active edge of the selected clock. This means that the counter value can be read differently from zero just after a trigger, especially when a low frequency signal is selected as the clock.

The following triggers are common to both modes:

- Software Trigger: Each channel has a software trigger, available by setting TC_CCR.SWTRG.

Downloaded from Arrow.com.

- SYNC: Each channel has a synchronization signal SYNC. When asserted, this signal has the same effect as a software trigger. The SYNC signals of all channels are asserted simultaneously by writing TC_BCR with SYNC set.
- Compare RC Trigger: RC is implemented in each channel and can provide a trigger when the counter value matches the RC value if TC_CMRx.CPCTRG is set .

The channel can also be configured to have an external trigger. In Capture mode, the external trigger signal can be selected between TIOAx and TIOBx. In Waveform mode, an external event can be programmed on one of the following signals: TIOBx, XC0, XC1 or XC2. This external event can then be programmed to perform a trigger by setting TC_CMRx.ENETRG.

If an external trigger is used, the duration of the pulses must be longer than the peripheral clock period in order to be detected.

50.6.7 Capture Mode

Capture mode is entered by clearing TC_CMRx.WAVE.

Capture mode allows the TC channel to perform measurements such as pulse timing, frequency, period, duty cycle and phase on TIOAx and TIOBx signals which are considered as inputs.

Figure 50-6 shows the configuration of the TC channel when programmed in Capture mode.

50.6.8 Capture Registers A and B

Registers A and B (TC_RA and TC_RB) are used as capture registers. They can be loaded with the counter value when a programmable event occurs on the signal TIOAx.

TC_CMRx.LDRA defines the TIOAx selected edge for the loading of TC_RA, and TC_CMRx.LDRB defines the TIOAx selected edge for the loading of TC_RB.

The subsampling ratio defined by TC_CMRx.SBSMPLR is applied to these selected edges, so that the loading of Register A and Register B occurs once every 1, 2, 4, 8 or 16 selected edges.

TC_RA is loaded only if it has not been loaded since the last trigger or if TC_RB has been loaded since the last loading of TC_RA.

TC_RB is loaded only if TC_RA has been loaded since the last trigger or the last loading of TC_RB.

Loading TC_RA or TC_RB before the read of the last value loaded sets TC_SR.LOVRS. In this case, the old value is overwritten.

When DMA is used, the Register AB (TC_RAB) address must be configured as source address of the transfer. TC_RAB provides the next unread value from TC_RA and TC_RB. It may be read by the DMA after a request has been triggered upon loading TC_RA or TC_RB.

50.6.9 Transfer with DMAC in Capture Mode

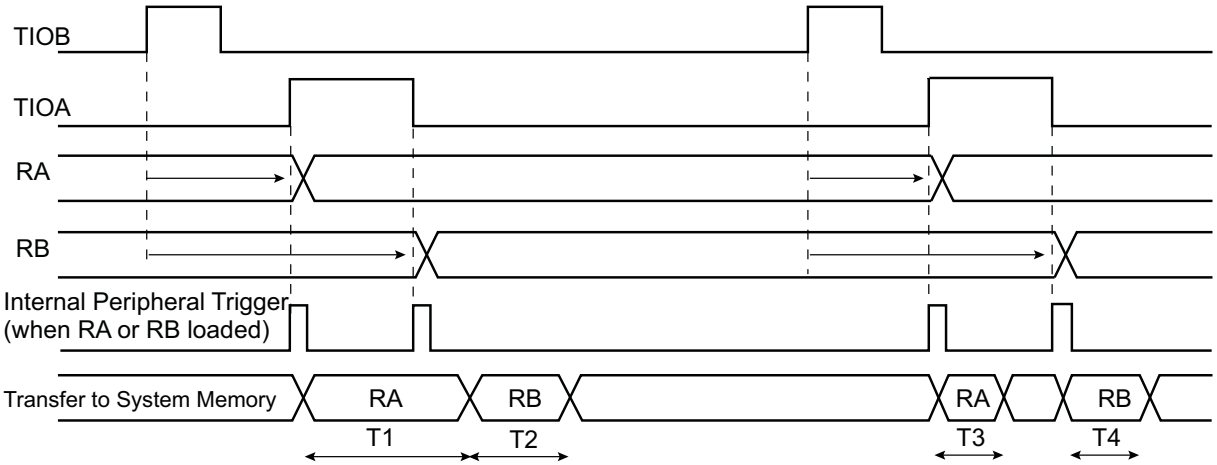
The DMAC can perform access from the TC to system memory in Capture mode only.

The following figure illustrates how TC_RA and TC_RB can be loaded in the system memory without processor intervention.

Downloaded from Arrow.com.

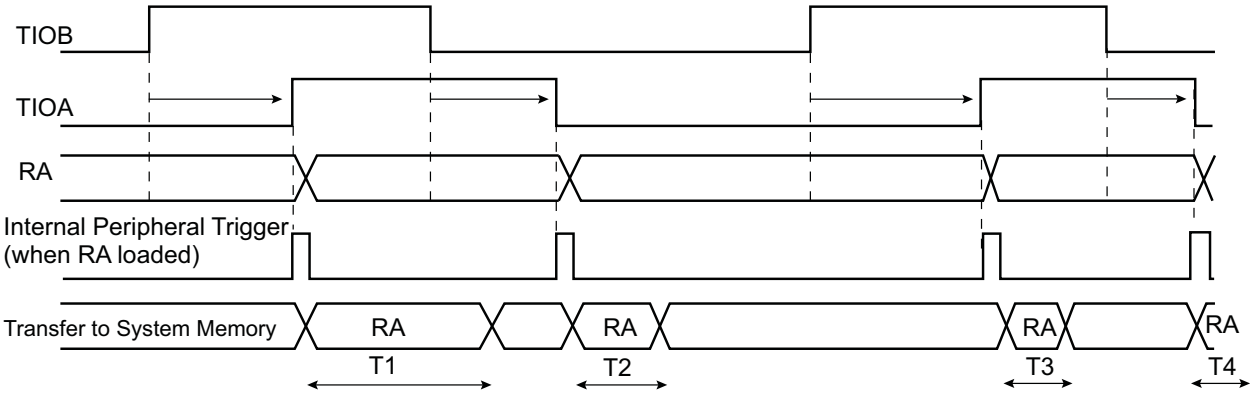
Figure 50-5. Example of Transfer with DMAC in Capture Mode

ETRGEDG = 1, LDRA = 1, LDRB = 2, ABETRG = 0



T1,T2,T3,T4 = System Bus load dependent (t_{min} = 8 Peripheral Clocks)

ETRGEDG = 3, LDRA = 3, LDRB = 0, ABETRG = 0



T1,T2,T3,T4 = System Bus load dependent (t_{min} = 8 Peripheral Clocks)

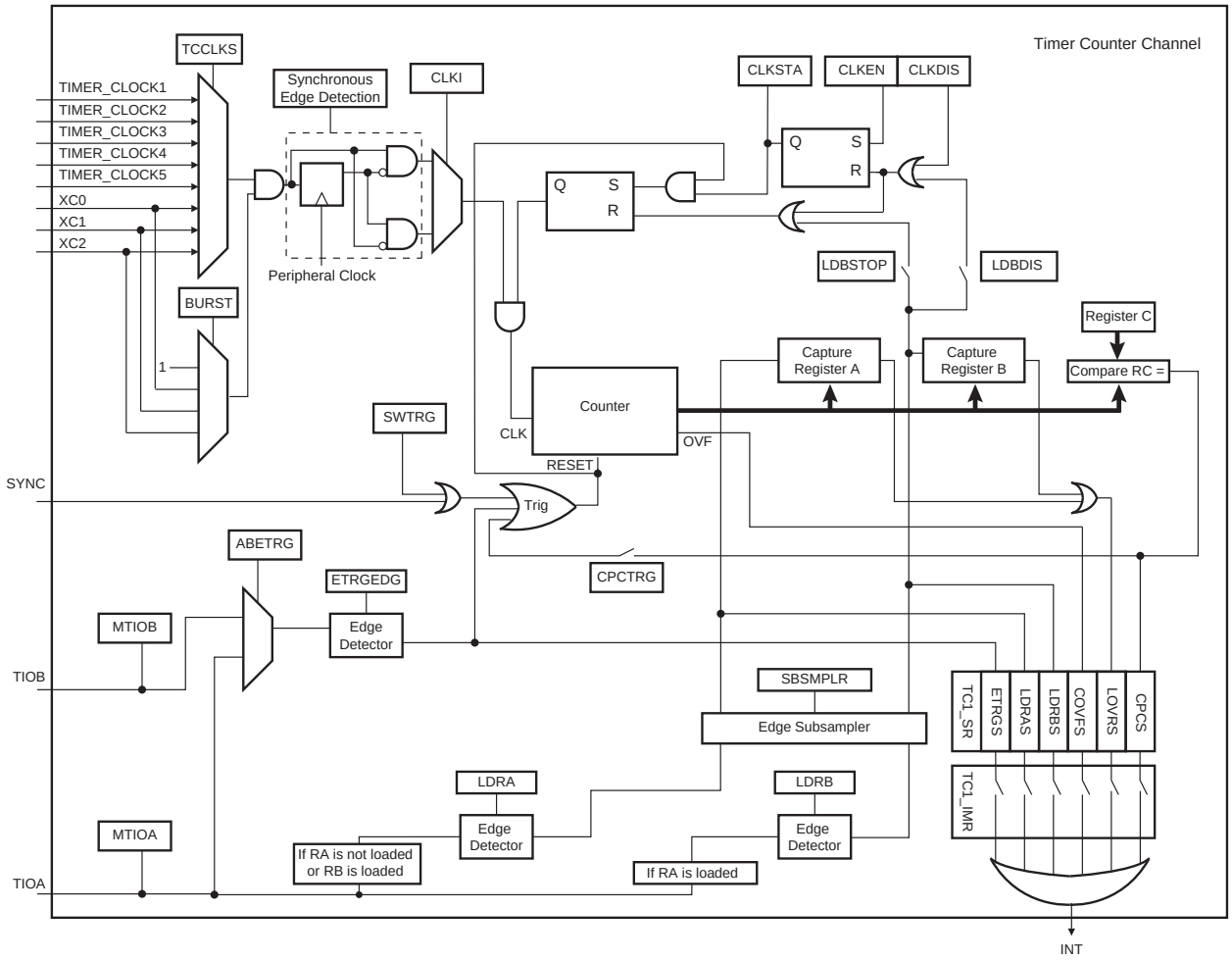
50.6.10 Trigger Conditions

In addition to the SYNC signal, the software trigger and the RC compare trigger, an external trigger can be defined.

The ABETRG bit in the TC_CMR selects TIOAx or TIOBx input signal as an external trigger or the trigger signal from the output comparator of the PWM module. The External Trigger Edge Selection parameter (ETRGEDG field in TC_CMR) defines the edge (rising, falling, or both) detected to generate an external trigger. If ETRGEDG = 0 (none), the external trigger is disabled.

Downloaded from Arrow.com.

Figure 50-6. Capture Mode



50.6.11 Waveform Mode

Waveform mode is entered by setting the TC_CMRx.WAVE bit.

In Waveform mode, the TC channel generates one or two PWM signals with the same frequency and independently programmable duty cycles, or generates different types of one-shot or repetitive pulses.

In this mode, TIOAx is configured as an output and TIOBx is defined as an output if it is not used as an external event (EEVT parameter in TC_CMR).

Waveform Mode shows the configuration of the TC channel when programmed in Waveform operating mode.

50.6.12 Waveform Selection

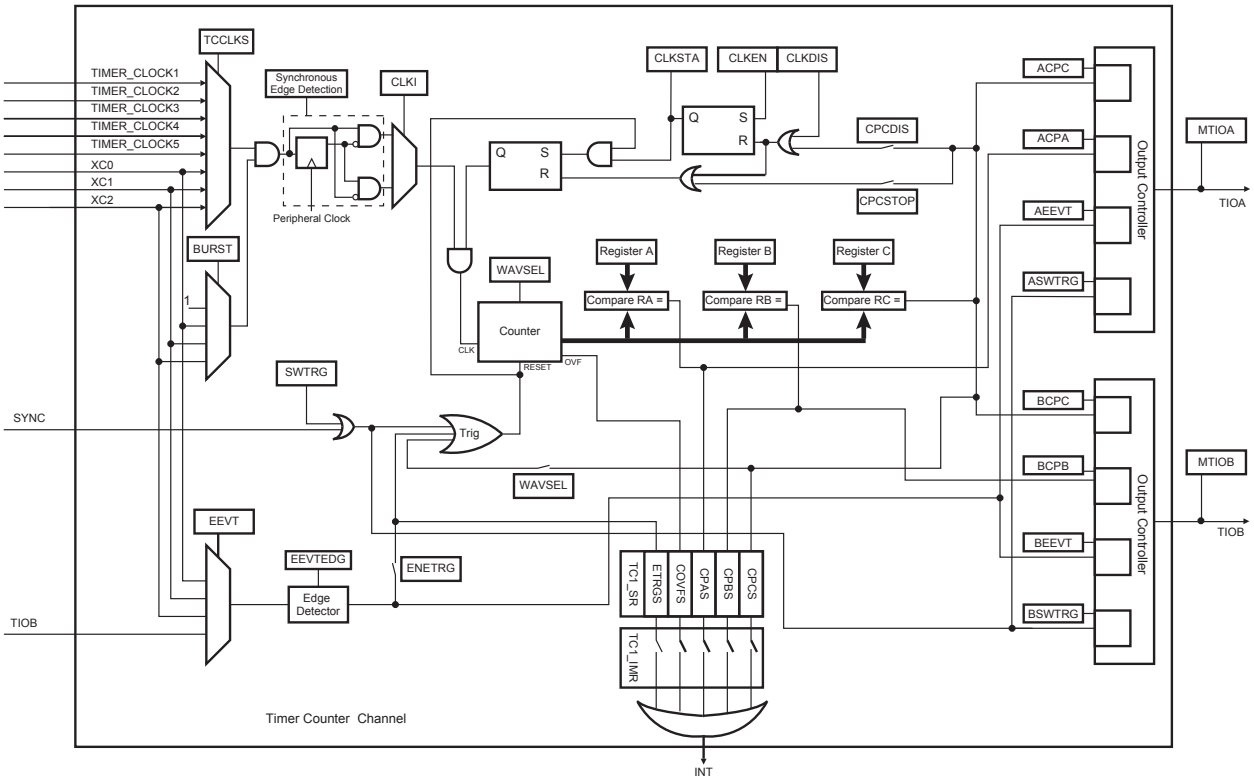
Depending on the WAVSEL parameter in TC_CMR, the behavior of TC_CV varies.

With any selection, TC_RA, TC_RB and TC_RC can all be used as compare registers.

RA Compare is used to control the TIOAx output, RB Compare is used to control the TIOBx output (if correctly configured) and RC Compare is used to control TIOAx and/or TIOBx outputs.

Downloaded from Arrow.com.

Figure 50-7. Waveform Mode



50.6.12.1 WAVSEL = 00

When WAVSEL = 00, the value of TC_CV is incremented from 0 to 2^{32}-1. Once 2^{32}-1 has been reached, the value of TC_CV is reset. Incrementation of TC_CV starts again and the cycle continues.

An external event trigger or a software trigger can reset the value of TC_CV. It is important to note that the trigger may occur at any time.

See the following figures.

RC Compare cannot be programmed to generate a trigger in this configuration. At the same time, RC Compare can stop the counter clock (CPCSTOP = 1 in TC_CMR) and/or disable the counter clock (CPCDIS = 1 in TC_CMR).

Downloaded from Arrow.com.

Figure 50-8. WAVSEL = 00 without Trigger

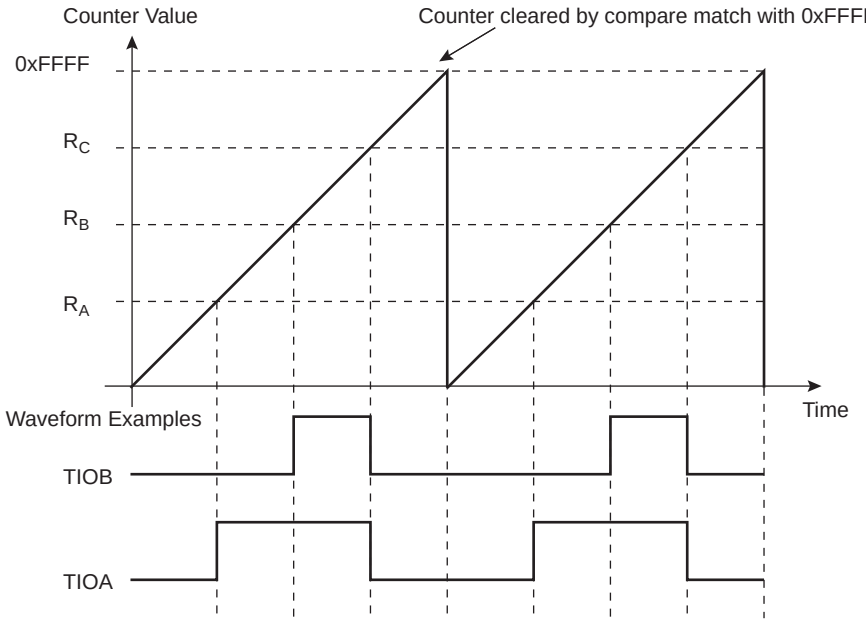
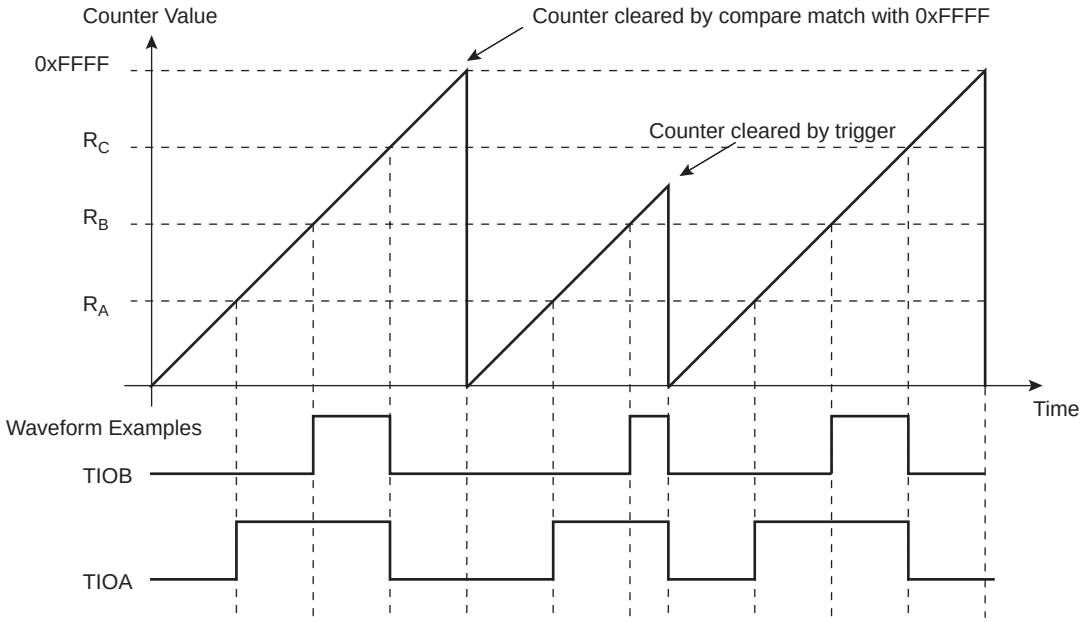


Figure 50-9. WAVSEL = 00 with Trigger



50.6.12.2 WAVSEL = 10

When WAVSEL = 10, the value of TC_CV is incremented from 0 to the value of RC, then automatically reset on a RC Compare. Once the value of TC_CV has been reset, it is then incremented and so on.

It is important to note that TC_CV can be reset at any time by an external event or a software trigger if both are programmed correctly.

See the following figures.

In addition, RC Compare can stop the counter clock (CPCSTOP = 1 in TC_CMR) and/or disable the counter clock (CPCDIS = 1 in TC_CMR).

Downloaded from Arrow.com.

Figure 50-10. WAVSEL = 10 without Trigger

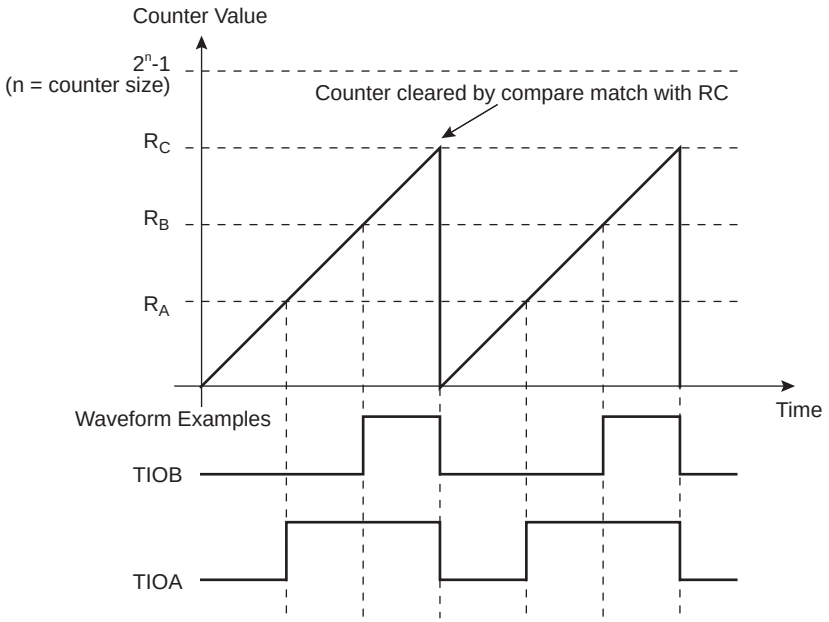
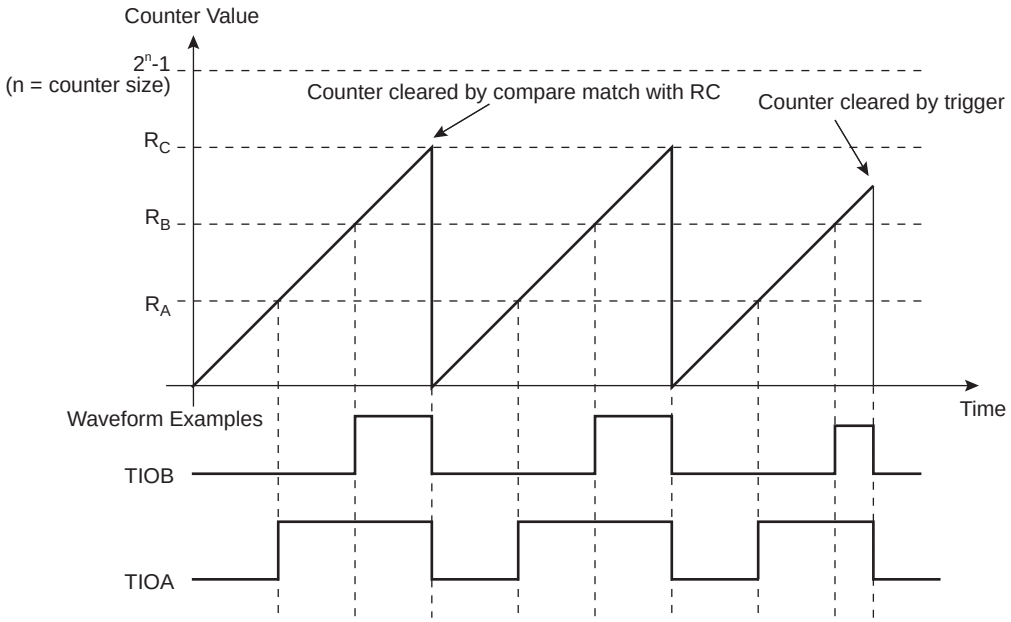


Figure 50-11. WAVSEL = 10 with Trigger



50.6.12.3 WAVSEL = 01

When WAVSEL = 01, the value of TC_CV is incremented from 0 to 2^{32}-1 . Once 2^{32}-1 is reached, the value of TC_CV is decremented to 0, then reincremented to 2^{32}-1 and so on.

A trigger such as an external event or a software trigger can modify TC_CV at any time. If a trigger occurs while TC_CV is incrementing, TC_CV then decrements. If a trigger is received while TC_CV is decrementing, TC_CV then increments.

See the following figures.

RC Compare cannot be programmed to generate a trigger in this configuration.

At the same time, RC Compare can stop the counter clock (CPCSTOP = 1) and/or disable the counter clock (CPCDIS = 1).

Downloaded from Arrow.com.

Figure 50-12. WAVSEL = 01 without Trigger

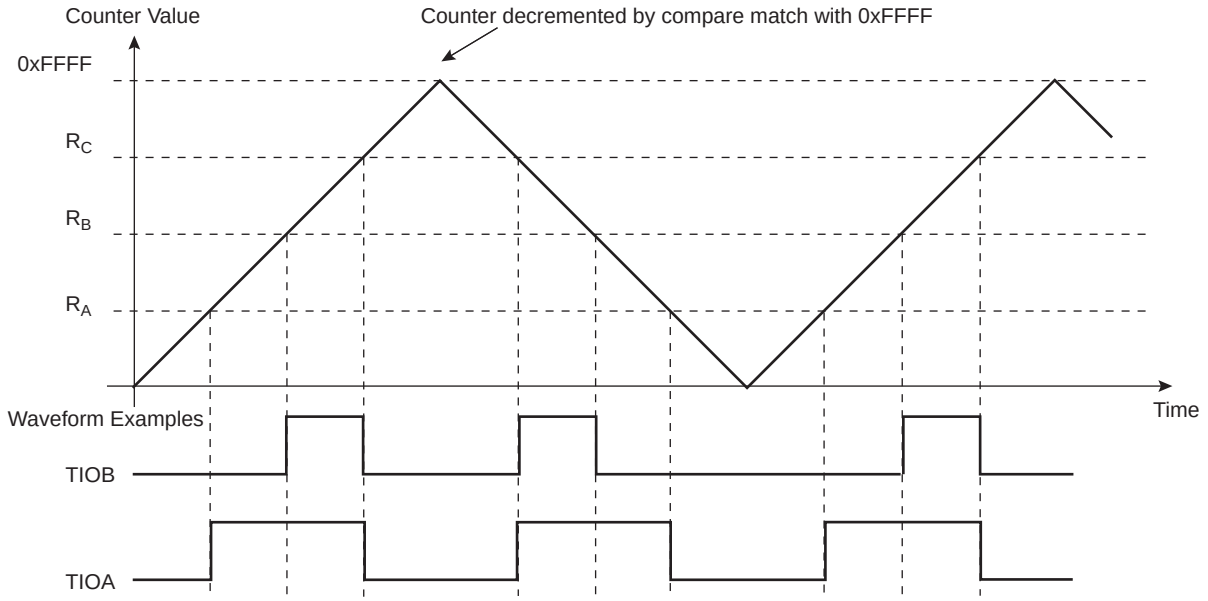
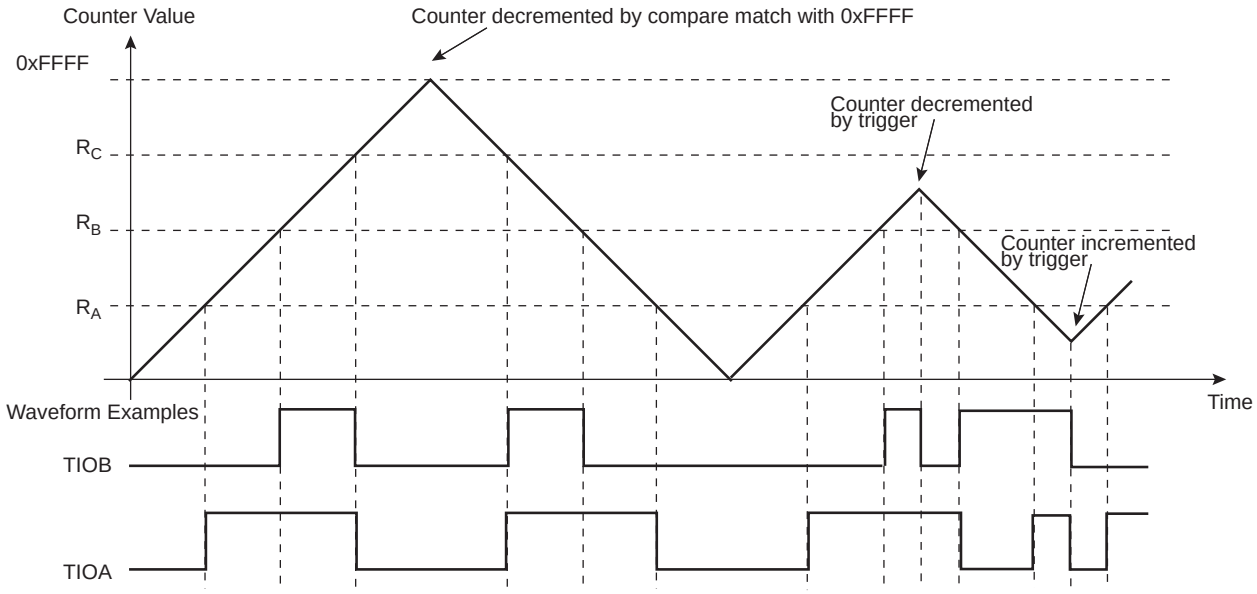


Figure 50-13. WAVSEL = 01 with Trigger



50.6.12.4 WAVSEL = 11

When WAVSEL = 11, the value of TC_CV is incremented from 0 to RC. Once RC is reached, the value of TC_CV is decremented to 0, then reincremented to RC and so on.

A trigger such as an external event or a software trigger can modify TC_CV at any time. If a trigger occurs while TC_CV is incrementing, TC_CV then decrements. If a trigger is received while TC_CV is decrementing, TC_CV then increments.

See the following figures.

RC Compare can stop the counter clock (CPCSTOP = 1) and/or disable the counter clock (CPCDIS = 1).

Downloaded from Arrow.com.

Figure 50-14. WAVSEL = 11 without Trigger

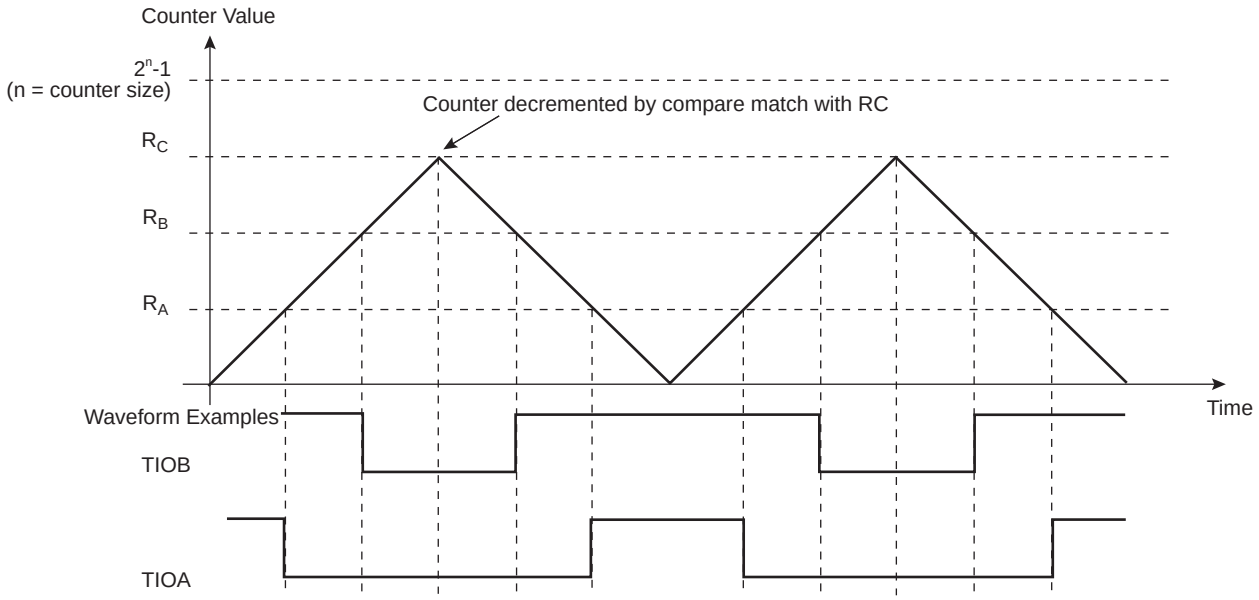
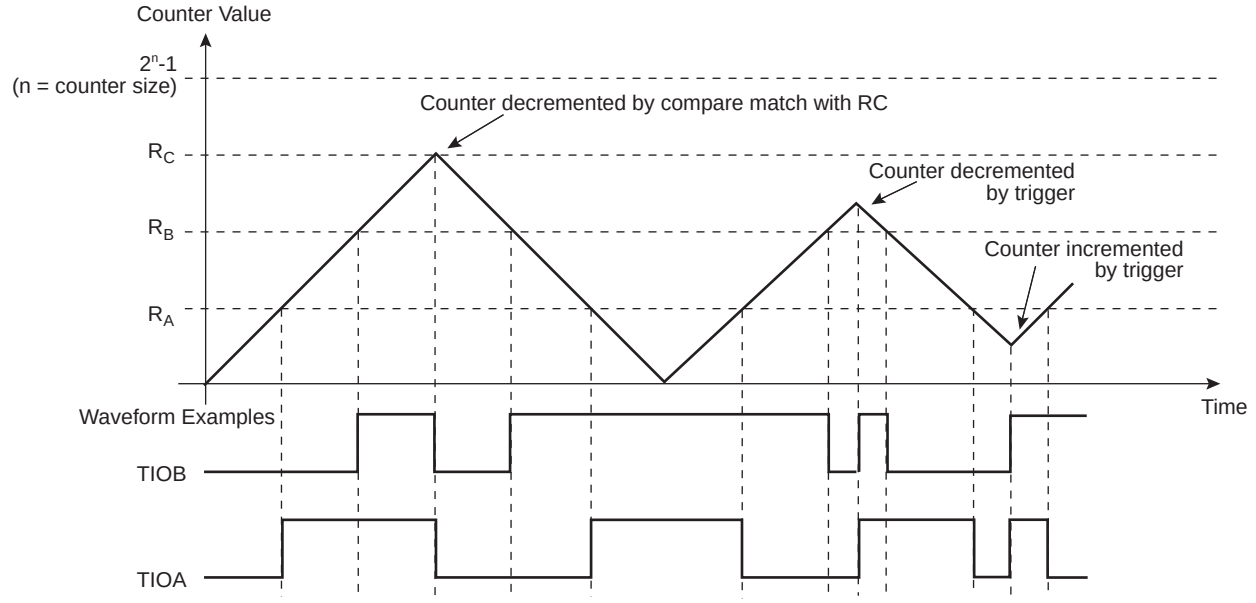


Figure 50-15. WAVSEL = 11 with Trigger



50.6.13 External Event/Trigger Conditions

An external event can be programmed to be detected on one of the clock sources (XC0, XC1, XC2) or TIOBx. The external event selected can then be used as a trigger.

The event trigger is selected using TC_CMR.EEVT. The trigger edge (rising, falling or both) for each of the possible external triggers is defined in TC_CMR.EEVTEDG. If EEVTEDG is cleared (none), no external event is defined.

If TIOBx is defined as an external event signal (EEVT = 0), TIOBx is no longer used as an output and the compare register B is not used to generate waveforms and subsequently no IRQs. In this case, the TC channel can only generate a waveform on TIOAx.

When an external event is defined, it can be used as a trigger by setting TC_CMR.ENETRG.

As in Capture mode, the SYNC signal and the software trigger are also available as triggers. RC Compare can also be used as a trigger depending on the parameter WAVSEL.

Downloaded from Arrow.com.

50.6.14 Synchronization with PWM

The inputs TIOAx/TIOBx can be bypassed, and thus channel trigger/capture events can be directly driven by the independent PWM module.

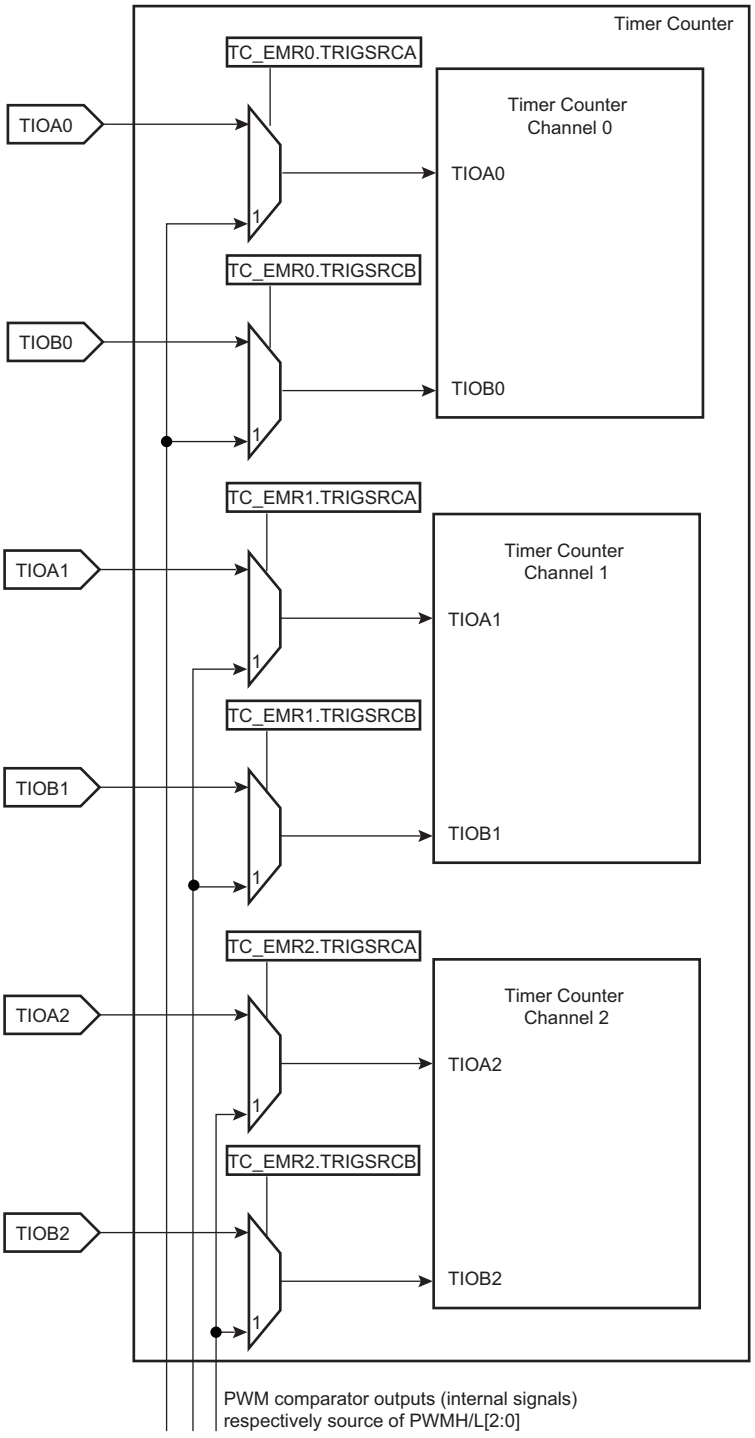
PWM comparator outputs (internal signals without dead-time insertion - OCx), respectively source of the PWMH/L[2:0] outputs, are routed to the internal TC inputs. These specific TC inputs are multiplexed with TIOA/B input signal to drive the internal trigger/capture events.

The selection is made in the Extended Mode register (TC_EMR) fields TRIGSRCA and TRIGSRCB (see TC_EMRx).

Each channel of the TC module can be synchronized by a different PWM channel as described in the following figure.

Downloaded from Arrow.com.

Figure 50-16. Synchronization with PWM



50.6.15 Output Controller

The output controller defines the output level changes on TIOAx and TIOBx following an event. TIOBx control is used only if TIOBx is defined as output (not as an external event).

The following events control TIOAx and TIOBx:

- Software trigger
- External event

Downloaded from Arrow.com.

- RC compare

RA Compare controls TIOAx, and RB Compare controls TIOBx. Each of these events can be programmed to set, clear or toggle the output as defined in the corresponding parameter in TC_CMR.

50.6.16 Quadrature Decoder

50.6.16.1 Description

The quadrature decoder (QDEC) is driven by TIOA0, TIOB0 and TIOB1 input pins and drives the timer counter of channel 0 and 1. Channel 2 can be used as a time base in case of speed measurement requirements (see the following figure).

When writing a ‘0’ to TC_BMR.QDEN, the QDEC is bypassed and the IO pins are directly routed to the timer counter function.

TIOA0 and TIOB0 are to be driven by the two dedicated quadrature signals from a rotary sensor mounted on the shaft of the off-chip motor.

A third signal from the rotary sensor can be processed through pin TIOB1 and is typically dedicated to be driven by an index signal if it is provided by the sensor. This signal is not required to decode the quadrature signals PHA, PHB.

TC_CMRx.TCCLKS must be configured to select XC0 input (i.e., 0x101). Field TC0XC0S has no effect as soon as the QDEC is enabled.

Either speed or position/revolution can be measured. Position channel 0 accumulates the edges of PHA, PHB input signals giving a high accuracy on motor position whereas channel 1 accumulates the index pulses of the sensor, therefore the number of rotations. Concatenation of both values provides a high level of precision on motion system position.

In Speed mode, position cannot be measured but revolution can be measured.

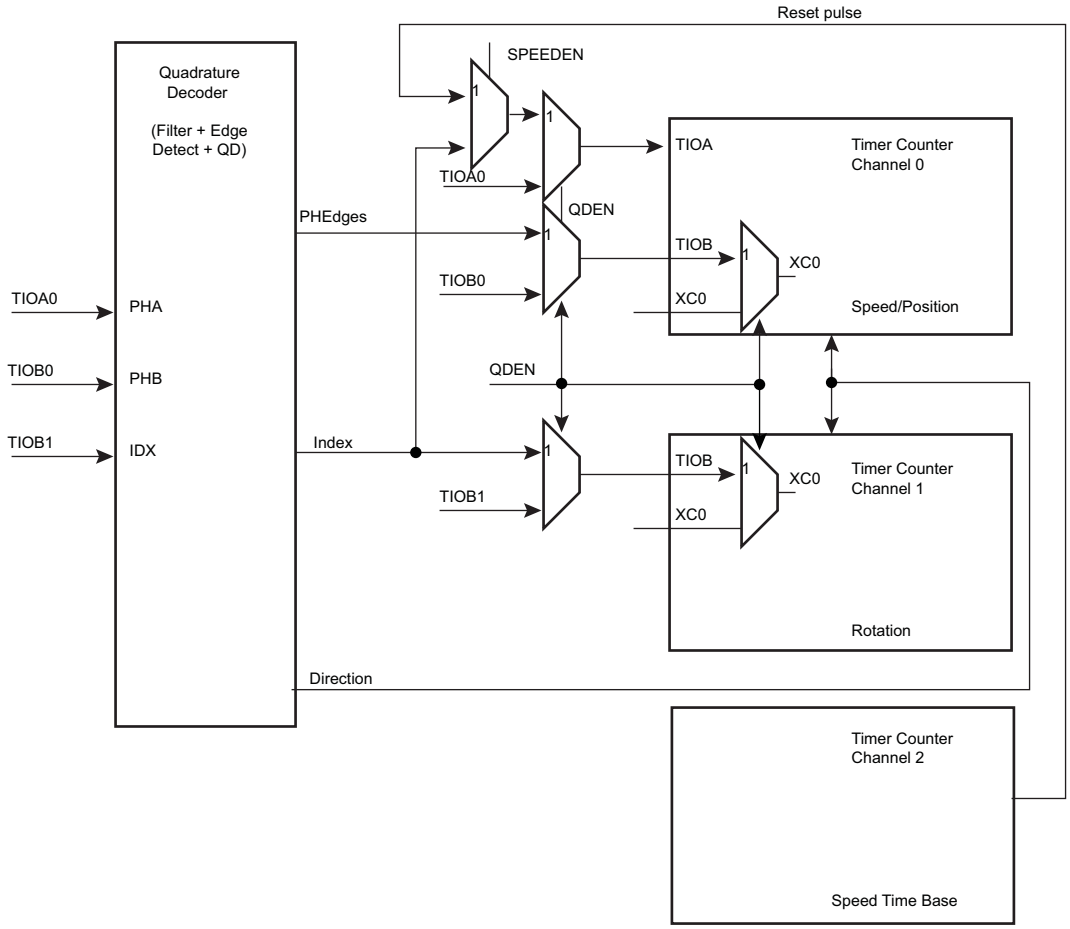
Inputs from the rotary sensor can be filtered prior to downstream processing. Accommodation of input polarity, phase definition and other factors are configurable.

Interruptions can be generated on different events.

A compare function (using TC_RC) is available on channel 0 (speed/position) or channel 1 (rotation) and can generate an interrupt by means of TC_SRx.CPCS.

Downloaded from Arrow.com.

Figure 50-17. Predefined Connection of the Quadrature Decoder with Timer Counters



50.6.16.2 Input Preprocessing

Input preprocessing consists of capabilities to take into account rotary sensor factors such as polarities and phase definition followed by configurable digital filtering.

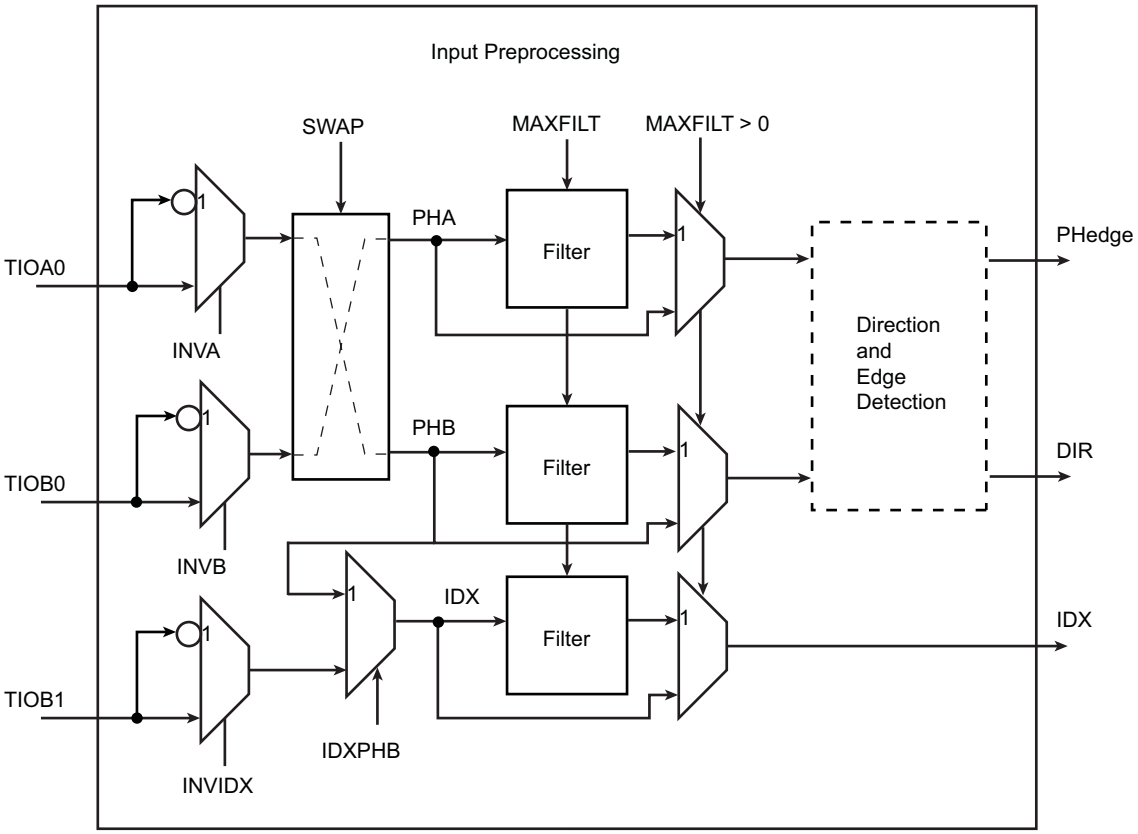
Each input can be negated and swapping PHA, PHB is also configurable.

TC_BMR. MAXFILT is used to configure a minimum duration for which the pulse is stated as valid. When the filter is active, pulses with a duration lower than (MAXFILT +1) × $t_{peripheral\ clock}$ are not passed to downstream logic.

The value of (MAXFILT +1) × $t_{peripheral\ clock}$ must not be greater than 10% of the minimum pulse on PHA, PHB or index when the rotary encoder speed is at its maximum. This speed depends on the application.

Downloaded from Arrow.com.

Figure 50-18. Input Stage

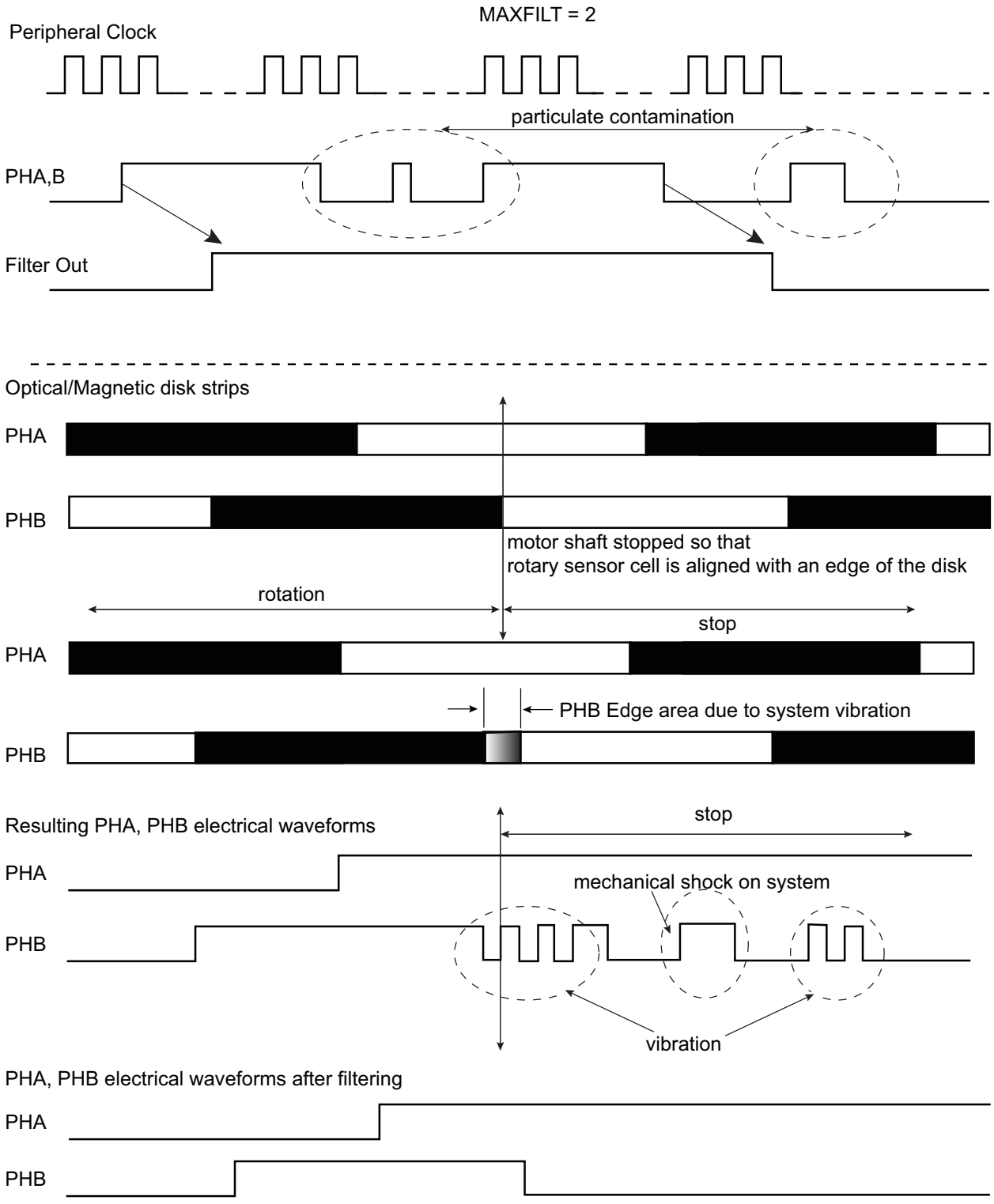


Input filtering can efficiently remove spurious pulses that might be generated by the presence of particulate contamination on the optical or magnetic disk of the rotary sensor.

Spurious pulses can also occur in environments with high levels of electromagnetic interference. Or, simply if vibration occurs even when rotation is fully stopped and the shaft of the motor is in such a position that the beginning of one of the reflective or magnetic bars on the rotary sensor disk is aligned with the light or magnetic (Hall) receiver cell of the rotary sensor. Any vibration can make the PHA, PHB signals toggle for a short duration.

Downloaded from Arrow.com.

Figure 50-19. Filtering Examples



50.6.16.3 Direction Status and Change Detection

After filtering, the quadrature signals are analyzed to extract the rotation direction and edges of the two quadrature signals detected in order to be counted by TC logic downstream.

Downloaded from Arrow.com.

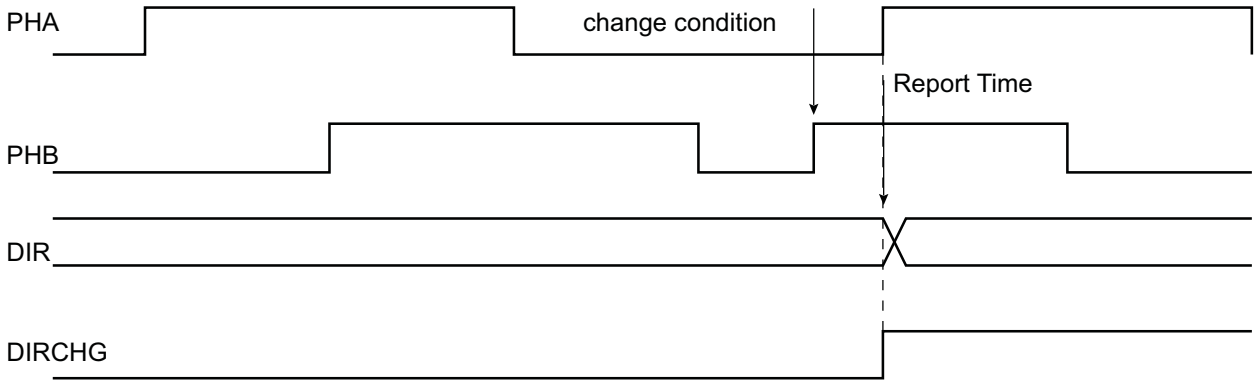
The direction status can be directly read at anytime in the TC_QISR. The polarity of the direction flag status depends on the configuration written in TC_BMR. INVA, INVB, INVIDX, SWAP modify the polarity of DIR flag.

Any change in rotation direction is reported in the TC_QISR and can generate an interrupt.

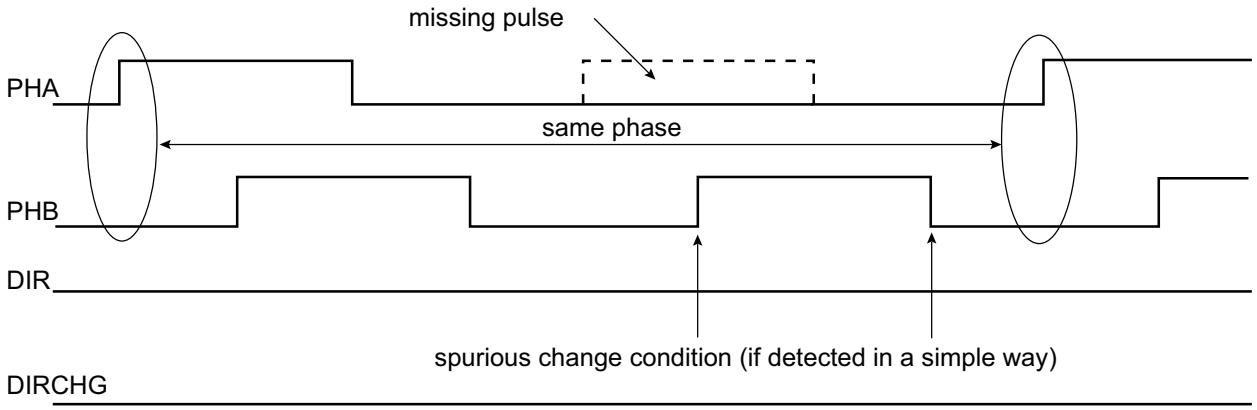
The direction change condition is reported as soon as two consecutive edges on a phase signal have sampled the same value on the other phase signal and there is an edge on the other signal. The two consecutive edges of one phase signal sampling the same value on other phase signal is not sufficient to declare a direction change, as particulate contamination may mask one or more reflective bars on the optical or magnetic disk of the sensor. See the following figure for waveforms.

Figure 50-20. Rotation Change Detection

Direction Change under normal conditions



No direction change due to particulate contamination masking a reflective bar

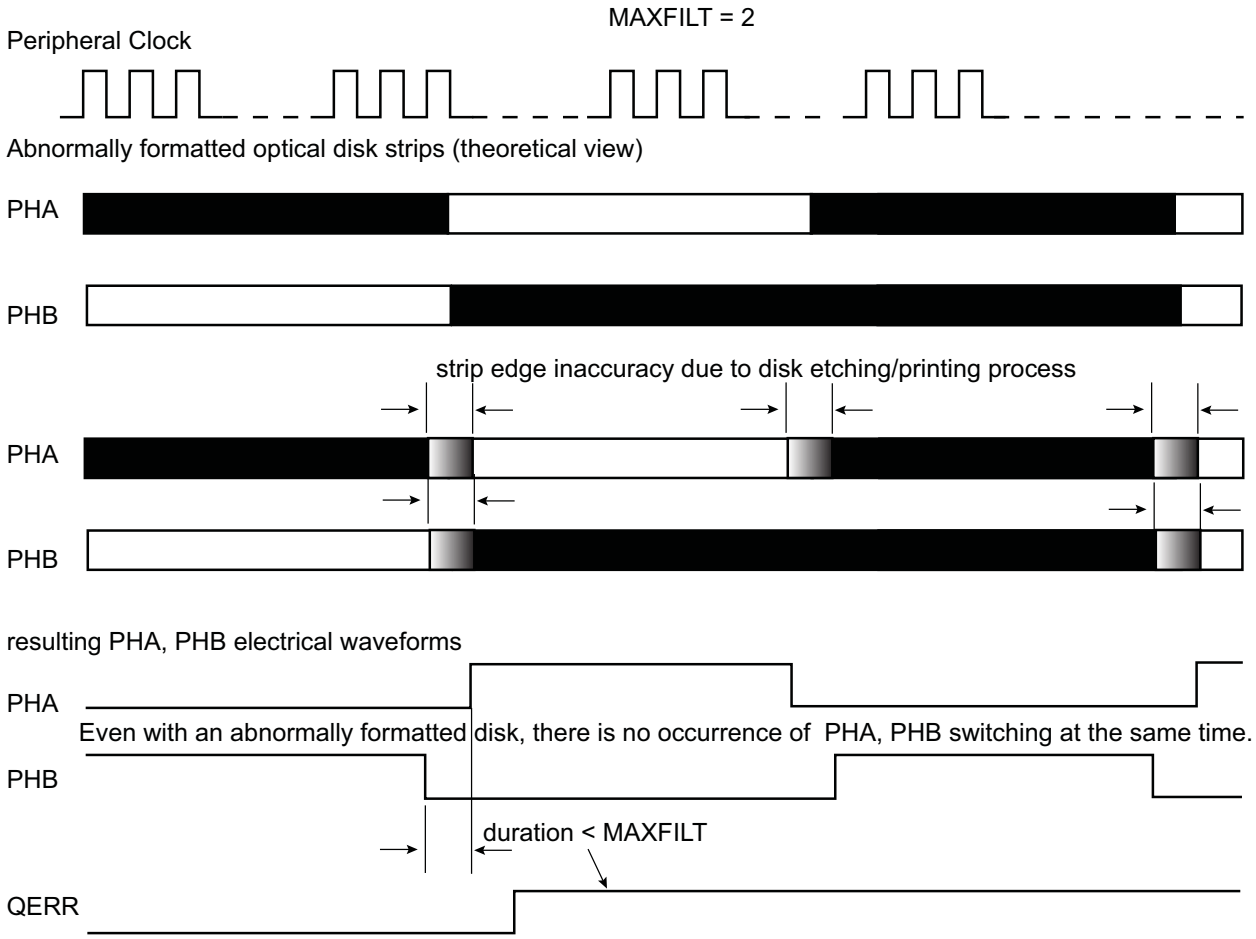


The direction change detection is disabled when TC_BMR.QDTRANS is set. In this case, the DIR flag report must not be used.

A quadrature error is also reported by the QDEC via TC_QISR.QERR. This error is reported if the time difference between two edges on PHA, PHB is lower than a predefined value. This predefined value is configurable and corresponds to (TC_BMR.MAXFILT + 1) × $t_{peripheral\ clock}$ ns. After being filtered, there is no reason to have two edges closer than (TC_BMR.MAXFILT + 1) × $t_{peripheral\ clock}$ ns under normal mode of operation.

Downloaded from Arrow.com.

Figure 50-21. Quadrature Error Detection



MAXFILT must be tuned according to several factors such as the peripheral clock frequency, type of rotary sensor and rotation speed to be achieved.

50.6.16.4 Position and Rotation Measurement

When TC_BMR.POSEN is set, the motor axis position is processed on channel 0 (by means of the PHA, PHB edge detections) and the number of motor revolutions are recorded on channel 1 if the IDX signal is provided on the TIOB1 input. If no IDX signal is available, the internal counter can be cleared for each revolution if the number of counts per revolution is configured in TC_RC0.RC and the TC_CMR.CPCTRG bit is written to '1'. The position measurement can be read in the TC_CV0 register and the rotation measurement can be read in the TC_CV1 register.

Channel 0 and 1 must be configured in Capture mode (TC_CMR0.WAVE = 0). 'Rising edge' must be selected as the External Trigger Edge (TC_CMR.ETRGEDG = 0x01) and 'TIOAx' must be selected as the External Trigger (TC_CMR.ABETRG = 0x1). The process must be started by configuring TC_CCR.CLKEN and TC_CCR.SWTRG.

In parallel, the number of edges are accumulated on TC channel 0 and can be read on the TC_CV0 register.

Therefore, the accurate position can be read on both TC_CV registers and concatenated to form a 32-bit word.

The TC channel 0 is cleared for each increment of IDX count value.

Depending on the quadrature signals, the direction is decoded and allows to count up or down in TC channels 0 and 1. The direction status is reported on TC_QISR.

50.6.16.5 Speed Measurement

When TC_BMR.SPEEDEN is set, the speed measure is enabled on channel 0.

A time base must be defined on channel 2 by writing the TC_RC2 period register. Channel 2 must be configured in Waveform mode (WAVE bit set) in TC_CMR2. The WAVSEL field must be defined with 0x10 to clear the counter by comparison and matching with TC_RC value. Field ACPC must be defined at 0x11 to toggle TIOAx output.

Downloaded from Arrow.com.

This time base is automatically fed back to TIOAx of channel 0 when QDEN and SPEEDEN are set.

Channel 0 must be configured in Capture mode (WAVE = 0 in TC_CMR0). TC_CMR0.ABETRG must be configured at 1 to select TIOAx as a trigger for this channel.

EDGTRG must be set to 0x01, to clear the counter on a rising edge of the TIOAx signal and field LDRA must be set accordingly to 0x01, to load TC_RA0 at the same time as the counter is cleared (LDRB must be set to 0x01). As a consequence, at the end of each time base period the differentiation required for the speed calculation is performed.

The process must be started by configuring bits CLKEN and SWTRG in the TC_CCR.

The speed can be read on field RA in TC_RA0.

Channel 1 can still be used to count the number of revolutions of the motor.

50.6.16.6 Detecting a Missing Index Pulse

To detect a missing index pulse due contamination, dust, etc., the TC_SR0.CPCS flag can be used. It is also possible to assert the interrupt line if the TC_SR0.CPCS flag is enabled as a source of the interrupt by writing a ‘1’ to TC_IER0.CPCS.

The TC_RC0.RC field must be written with the nominal number of counts per revolution provided by the rotary encoder, plus a margin to eliminate potential noise (e.g., if nominal count per revolution is 1024, then TC_RC0.RC=1026).

If the index pulse is missing, the timer value is not cleared and the nominal value is exceeded, then the comparator on the RC triggers an event, TC_SR0.CPCS=1, and the interrupt line is asserted if TC_IER0.CPCS=1.

The missing index pulse detection is only valid if the bit TC_QISR.DIRCHG=0.

50.6.16.7 Detecting a Badly Located Index

The digital filter reduces effects of dust, scratches or contamination on the index line. If the contamination creates a pulse surpassing the filter capacity (in particular at low speed), this can be interpreted as an index pulse, even if it is badly placed.

An on-the-fly detection of a badly-placed index is enabled when TC_BMR.BIDXCE=1. TC_RC0.RC must be written with the nominal number of counts per revolution provided by the rotary encoder + 2, e.g., if the nominal count per revolution is 1024, then TC_RC0.RC=1026. This margin of 2 eliminates potential noise.

If the pulse is processed while the current value of the counter (TC_CV0) is outside the value programmed in TC_RC0.RC ±2, the TC_SR0.LDRAS flag is written to ‘1’ and the location of the pulse is written in TC_RA0.

This detection circuitry does not prevent a rotary encoder integrity check before use.

50.6.16.8 Detecting Contamination/Dust at Rotary Encoder Low Speed

The contamination/dust that can be filtered when the rotary encoder speed is high may not be filtered at low speed, thus creating unsollicited direction change, etc.

At low speed, even a minor contamination may appear as a long pulse, and thus not filtered and processed as a standard quadrature encoder pulse.

This contamination can be detected by using the similar method as the missing index detection.

A contamination exists on a phase line if TC_SR.CPCS = 1 and TC_QISR.DIRCHG = 1 when there is no sollicited change of direction.

50.6.16.9 Report of Filtered Pulses due to Contamination/Dust

When the digital filter removes pulses created by contamination/dust, a report is provided in the TC QDEC Interrupt Status Register. A separate flag is provided for each line (PHA, PHB, Index) and one flag is provided when a missing pulse is corrected (a ‘1’ must be written to TC_BMR.AUTOC). If TC_QISR.FPHA=1, a pulse has been filtered on PHA line. If TC_QISR.FPHB=1, a pulse has been filtered on PHB line. If TC_QISR.FIDX=1, a pulse has been filtered on Index line. If TC_QISR.FMP=1, a missing pulse has been corrected by the missing pulse detection logic.

The on-the-fly detection and associated flags can be used to anticipate further effects of contamination/dust, in particular if the flag is written to ‘1’ when the rotary encoder speed is high. This may cause abnormal behavior when the rotary encoder slows down if the abnormal pulse is no longer filtered (surpassing digital filter capabilities).

This detection circuitry does not prevent rotary encoder integrity check before use.

Downloaded from Arrow.com.

50.6.17 2-bit Gray Up/Down Counter for Stepper Motor

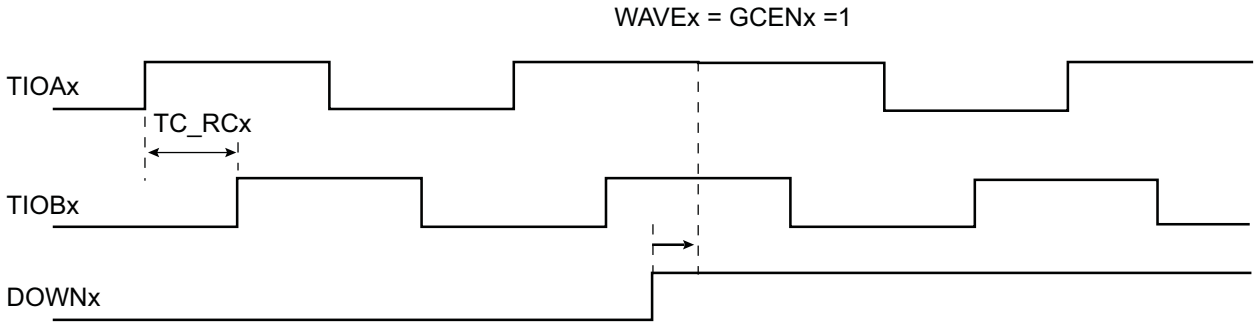
Each channel can be independently configured to generate a 2-bit Gray count waveform on corresponding TIOAx, TIOBx outputs by means of TC_SMMRx.GCEN.

Up or Down count can be defined by writing TC_SMMRx.DOWN.

It is mandatory to configure the channel in Waveform mode in the TC_CMR.

The period of the counters can be programmed in TC_RCx.

Figure 50-22. 2-bit Gray Up/Down Counter



50.6.18 Fault Mode

At any time, the TC_RCx registers can be used to perform a comparison on the respective current channel counter value (TC_CVx) with the value of TC_RCx register.

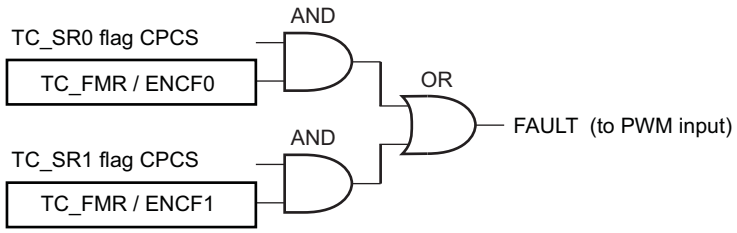
The CPCSx flags can be set accordingly and an interrupt can be generated.

This interrupt is processed but requires an unpredictable amount of time to be achieve the required action.

It is possible to trigger the FAULT output of the TIMER1 with CPCS from TC_SR0 and/or CPCS from TC_SR1. Each source can be independently enabled/disabled in the TC_FMR.

This can be useful to detect an overflow on speed and/or position when QDEC is processed and to act immediately by using the FAULT output.

Figure 50-23. Fault Output Generation



50.6.19 Register Write Protection

To prevent any single software error from corrupting TC behavior, certain registers in the address space can be write-protected by setting the WPEN bit , WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) in the TC Write Protection Mode Register (TC_WPMR).

If a write access to the protected registers is detected, the WPVS flag in the TC Safety Status Register (TRNG_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The Timer Counter clock of the first channel must be enabled to access TC_WPMR.

The following registers can be write-protected when WPEN is set:

- TC Block Mode Register
- TC Channel Mode Register Capture Mode
- TC Channel Mode Register Waveform Mode
- TC Fault Mode Register

Downloaded from Arrow.com.

- TC Stepper Motor Mode Register
- TC Register A
- TC Register B
- TC Register C
- TC Extended Mode Register

The following registers can be write-protected when WPITEN is set:

- TC Interrupt Enable Register
- TC Interrupt Disable Register
- TC QDEC Interrupt Enable Register
- TC QDEC Interrupt Disable Register

The following register can be write-protected when WPCREN is set:

- TC Channel Control Register
- TC Block Control Register

50.6.20 Security and Safety Analysis and Reports

Several type of checks are performed when a TC channel is enabled.

The peripheral clock of the TC is monitored by a specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the TDES. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the flag TC_SSRx.CGD is set, an abnormal condition occurred on the internal clock net. This flag is not set under normal operating conditions.

The internal counter of a TC channel is also monitored and if an abnormal state is detected, the flag TC_SSRx.SEQE is set. This flag cannot be set under normal operating conditions.

The software accesses to the TC are monitored and if an incorrect access is performed, the flag TC_SSRx.SWE is set. The type of incorrect/abnormal software access is reported in TC_SSRx.SWETYP (see TC_CSRx for details). As an example, when the TC channel is configured in Capture mode, reading TC_RAx when TC_SRx.LDRAS=0 asserts the SWE flags. TC_SSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The flags CGD, SEQE, SWE, WPVS are automatically cleared when TC_SSRx is read.

If one of these flags is set, the flag TC_SRx.SECE is set and triggers an interrupt if the TC_IMRx.SECE bit is ‘1’. SECE is cleared by reading TC_SRx.

Downloaded from Arrow.com.

50.7 Register Summary

Note: The register TC_CMR has two modes, Capture Mode and Waveform Mode. In this register summary, both modes are displayed

Offset	Name	Bit Pos.								
0x00	TC_CCR0	31:24								
		23:16								
		15:8								
		7:0						SWTRG	CLKDIS	CLKEN
0x04	TC_CMR0	31:24								
		23:16		SBSMPLR[2:0]			LDRB[1:0]		LDRA[1:0]	
		15:8	WAVE	CPCTRG				ABETRG	ETRGEDG[1:0]	
		7:0	LDBDIS	LDBSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
0x04	TC_CMR0	31:24	BSWTRG[1:0]		BEEVT[1:0]		BCPC[1:0]		BCPB[1:0]	
		23:16	ASWTRG[1:0]		AEEVT[1:0]		ACPC[1:0]		ACPA[1:0]	
		15:8	WAVE	WAVSEL[1:0]		ENETRG	EEVT[1:0]		EEVTEDG[1:0]	
		7:0	CPCDIS	CPCSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
0x08	TC_SMMR0	31:24								
		23:16								
		15:8								
		7:0							DOWN	GCEN
0x0C	TC_RAB0	31:24	RAB[31:24]							
		23:16	RAB[23:16]							
		15:8	RAB[15:8]							
		7:0	RAB[7:0]							
0x10	TC_CV0	31:24	CV[31:24]							
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x14	TC_RA0	31:24	RA[31:24]							
		23:16	RA[23:16]							
		15:8	RA[15:8]							
		7:0	RA[7:0]							
0x18	TC_RB0	31:24	RB[31:24]							
		23:16	RB[23:16]							
		15:8	RB[15:8]							
		7:0	RB[7:0]							
0x1C	TC_RC0	31:24	RC[31:24]							
		23:16	RC[23:16]							
		15:8	RC[15:8]							
		7:0	RC[7:0]							
0x20	TC_SR0	31:24								
		23:16						MTIOB	MTIOA	CLKSTA
		15:8								SECE
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x24	TC_IER0	31:24								
		23:16								
		15:8						SECE		
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x28	TC_IDR0	31:24								
		23:16								
		15:8						SECE		
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x2C	TC_IMR0	31:24								
		23:16								
		15:8						SECE		
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x30	TC_EMR0	31:24								
		23:16								
		15:8								NODIVCLK
		7:0			TRIGSRCB[1:0]				TRIGSRCA[1:0]	
0x34	TC_CSR0	31:24								
		23:16						MTIOB	MTIOA	CLKSTA
		15:8								
		7:0								
0x38	TC_SSR0	31:24	ECLASS				SWETYP[3:0]			
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS
0x3C ... 0x3F	Reserved									
0x40	TC_CCR1	31:24								
		23:16								
		15:8								
		7:0						SWTRG	CLKDIS	CLKEN
0x44	TC_CMR1	31:24								
		23:16		SBSMPLR[2:0]			LDRB[1:0]		LDRA[1:0]	
		15:8	WAVE	CPCTRG				ABETRG	ETRGEDG[1:0]	
		7:0	LDBDIS	LDBSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
0x44	TC_CMR1	31:24	BSWTRG[1:0]		BEEVT[1:0]		BCPC[1:0]		BCPB[1:0]	
		23:16	ASWTRG[1:0]		AEEVT[1:0]		ACPC[1:0]		ACPA[1:0]	
		15:8	WAVE	WAVSEL[1:0]		ENETRG	EEVT[1:0]		EEVTEDG[1:0]	
		7:0	CPCDIS	CPCSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
0x48	TC_SMMR1	31:24								
		23:16								
		15:8								
		7:0							DOWN	GCEN
0x4C	TC_RAB1	31:24	RAB[31:24]							
		23:16	RAB[23:16]							
		15:8	RAB[15:8]							
		7:0	RAB[7:0]							
0x50	TC_CV1	31:24	CV[31:24]							
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x54	TC_RA1	31:24	RA[31:24]							
		23:16	RA[23:16]							
		15:8	RA[15:8]							
		7:0	RA[7:0]							
0x58	TC_RB1	31:24	RB[31:24]							
		23:16	RB[23:16]							
		15:8	RB[15:8]							
		7:0	RB[7:0]							
0x5C	TC_RC1	31:24	RC[31:24]							
		23:16	RC[23:16]							
		15:8	RC[15:8]							
		7:0	RC[7:0]							
0x60	TC_SR1	31:24								
		23:16						MTIOB	MTIOA	CLKSTA
		15:8								SECE
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x64	TC_IER1	31:24								
		23:16								
		15:8						SECE		
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x68	TC_IDR1	31:24								
		23:16								
		15:8						SECE		
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x6C	TC_IMR1	31:24								
		23:16								
		15:8						SECE		
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x70	TC_EMR1	31:24								
		23:16								
		15:8								NODIVCLK
		7:0			TRIGSRCB[1:0]				TRIGSRCA[1:0]	
0x74	TC_CSR1	31:24								
		23:16						MTIOB	MTIOA	CLKSTA
		15:8								
		7:0								
0x78	TC_SSR1	31:24	ECLASS				SWETYP[3:0]			
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS
0x7C ... 0x7F	Reserved									
0x80	TC_CCR2	31:24								
		23:16								
		15:8								
		7:0						SWTRG	CLKDIS	CLKEN
0x84	TC_CMR2	31:24								
		23:16		SBSMPLR[2:0]			LDRB[1:0]		LDRA[1:0]	
		15:8	WAVE	CPCTRG				ABETRG	ETRGEDG[1:0]	
		7:0	LDBDIS	LDBSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
0x84	TC_CMR2	31:24	BSWTRG[1:0]		BEEVT[1:0]		BCPC[1:0]		BCPB[1:0]	
		23:16	ASWTRG[1:0]		AEEVT[1:0]		ACPC[1:0]		ACPA[1:0]	
		15:8	WAVE	WAVSEL[1:0]		ENETRG	EEVT[1:0]		EEVTEDG[1:0]	
		7:0	CPCDIS	CPCSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
0x88	TC_SMMR2	31:24								
		23:16								
		15:8								
		7:0							DOWN	GCEN
0x8C	TC_RAB2	31:24	RAB[31:24]							
		23:16	RAB[23:16]							
		15:8	RAB[15:8]							
		7:0	RAB[7:0]							
0x90	TC_CV2	31:24	CV[31:24]							
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x94	TC_RA2	31:24	RA[31:24]							
		23:16	RA[23:16]							
		15:8	RA[15:8]							
		7:0	RA[7:0]							
0x98	TC_RB2	31:24	RB[31:24]							
		23:16	RB[23:16]							
		15:8	RB[15:8]							
		7:0	RB[7:0]							
0x9C	TC_RC2	31:24	RC[31:24]							
		23:16	RC[23:16]							
		15:8	RC[15:8]							
		7:0	RC[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0xA0	TC_SR2	31:24								
		23:16						MTIOB	MTIOA	CLKSTA
		15:8								SECE
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0xA4	TC_IER2	31:24								
		23:16								
		15:8						SECE		
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0xA8	TC_IDR2	31:24								
		23:16								
		15:8						SECE		
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0xAC	TC_IMR2	31:24								
		23:16								
		15:8						SECE		
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0xB0	TC_EMR2	31:24								
		23:16								
		15:8								NODIVCLK
		7:0			TRIGSRCB[1:0]				TRIGSRCA[1:0]	
0xB4	TC_CSR2	31:24								
		23:16						MTIOB	MTIOA	CLKSTA
		15:8								
		7:0								
0xB8	TC_SSR2	31:24	ECLASS				SWETYP[3:0]			
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS
0xBC ... 0xBF	Reserved									
0xC0	TC_BCR	31:24								
		23:16								
		15:8								
		7:0								SYNC
0xC4	TC_BMR	31:24							MAXFILT[5:4]	
		23:16	MAXFILT[3:0]						IDXPHB	SWAP
		15:8	INVIDX	INVB	INVA	EDGPHA	QDTRANS	SPEEDEN	POSEN	QDEN
		7:0			TC2XC2S[1:0]		TC1XC1S[1:0]		TC0XC0S[1:0]	
0xC8	TC_QIER	31:24								
		23:16								
		15:8								
		7:0	FMP	FIDX	FPHB	FPHA		QERR	DIRCHG	IDX
0xCC	TC_QIDR	31:24								
		23:16								
		15:8								
		7:0	FMP	FIDX	FPHB	FPHA		QERR	DIRCHG	IDX
0xD0	TC_QIMR	31:24								
		23:16								
		15:8								
		7:0	FMP	FIDX	FPHB	FPHA		QERR	DIRCHG	IDX
0xD4	TC_QISR	31:24								
		23:16								
		15:8								DIR
		7:0	FMP	FIDX	FPHB	FPHA		QERR	DIRCHG	IDX
0xD8	TC_FMR	31:24								
		23:16								
		15:8								
		7:0							ENCF1	ENCF0

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0xDC	TC_QSR	31:24								
		23:16								
		15:8								DIR
		7:0								
0xE0 ... 0xE3	Reserved									
0xE4	TC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0				FIRSTE		WPCREN	WPITEN	WPEN

Downloaded from Arrow.com.

50.7.1 TC Channel Control Register

Name: TC_CCRx
Offset: 0x00 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						SWTRG	CLKDIS	CLKEN
Access						W	W	W
Reset						–	–	–

Bit 2 – SWTRG Software Trigger Command

Value	Description
0	No effect.
1	A software trigger is performed: the counter is reset and the clock is started.

Bit 1 – CLKDIS Counter Clock Disable Command

Value	Description
0	No effect.
1	Disables the clock.

Bit 0 – CLKEN Counter Clock Enable Command

Value	Description
0	No effect.
1	Enables the clock if CLKDIS is not 1.

Downloaded from Arrow.com.

50.7.2 TC Channel Mode Register: Capture Mode

Name: TC_CMRx
Offset: 0x04 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can be written only if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		SBSMPLR[2:0]			LDRB[1:0]		LDRA[1:0]	
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WAVE	CPCTRG				ABETRG	ETRGEDG[1:0]	
Access	R/W	R/W				R/W	R/W	R/W
Reset	0	0				0	0	0

Bit	7	6	5	4	3	2	1	0
	LDBDIS	LDBSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 22:20 – SBSMPLR[2:0] Loading Edge Subsampling Ratio

Value	Name	Description
0	ONE	Load a Capture register each selected edge.
1	HALF	Load a Capture register every 2 selected edges.
2	FOURTH	Load a Capture register every 4 selected edges.
3	EIGHTH	Load a Capture register every 8 selected edges.
4	SIXTEENTH	Load a Capture register every 16 selected edges.

Bits 19:18 – LDRB[1:0] RB Loading Edge Selection

Value	Name	Description
0	NONE	None
1	RISING	Rising edge of TIOAx
2	FALLING	Falling edge of TIOAx
3	EDGE	Each edge of TIOAx

Bits 17:16 – LDRA[1:0] RA Loading Edge Selection

Value	Name	Description
0	NONE	None
1	RISING	Rising edge of TIOAx
2	FALLING	Falling edge of TIOAx
3	EDGE	Each edge of TIOAx

Bit 15 – WAVE Waveform Mode

Value	Description
0	Capture mode is enabled.
1	Capture mode is disabled (Waveform mode is enabled).

Downloaded from Arrow.com.

Bit 14 – CPCTRG RC Compare Trigger Enable

Value	Description
0	RC Compare has no effect on the counter and its clock.
1	RC Compare resets the counter and starts the counter clock.

Bit 10 – ABETRG TIOAx or TIOBx External Trigger Selection

Value	Description
0	TIOBx is used as an external trigger.
1	TIOAx is used as an external trigger.

Bits 9:8 – ETRGEDG[1:0] External Trigger Edge Selection

Value	Name	Description
0	NONE	The clock is not gated by an external signal.
1	RISING	Rising edge
2	FALLING	Falling edge
3	EDGE	Each edge

Bit 7 – LDBDIS Counter Clock Disable with RB Loading

Value	Description
0	Counter clock is not disabled when RB loading occurs.
1	Counter clock is disabled when RB loading occurs.

Bit 6 – LDBSTOP Counter Clock Stopped with RB Loading

Value	Description
0	Counter clock is not stopped when RB loading occurs.
1	Counter clock is stopped when RB loading occurs.

Bits 5:4 – BURST[1:0] Burst Signal Selection

Value	Name	Description
0	NONE	The clock is not gated by an external signal.
1	XC0	XC0 is ANDed with the selected clock.
2	XC1	XC1 is ANDed with the selected clock.
3	XC2	XC2 is ANDed with the selected clock.

Bit 3 – CLKI Clock Invert

Value	Description
0	Counter is incremented on rising edge of the clock.
1	Counter is incremented on falling edge of the clock.

Bits 2:0 – TCCLKS[2:0] Clock Selection

To operate at maximum peripheral clock frequency, see TC_EMRx.

Value	Name	Description
0	TIMER_CLOCK1	Clock selected: internal GCLK [17], GLCK[45] clock signal (from PMC)
1	TIMER_CLOCK2	Clock selected: internal MCK/8 clock signal (from PMC)
2	TIMER_CLOCK3	Clock selected: internal MCK/32 clock signal (from PMC)
3	TIMER_CLOCK4	Clock selected: internal MCK/128 clock signal (from PMC)
4	TIMER_CLOCK5	Clock selected: internal MD_SLCK clock signal (from PMC)
5	XC0	Clock selected: XC0
6	XC1	Clock selected: XC1
7	XC2	Clock selected: XC2

Downloaded from Arrow.com.

50.7.3 TC Channel Mode Register: Waveform Mode

Name: TC_CMRx
Offset: 0x04 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	BSWTRG[1:0]		BEEVT[1:0]		BCPC[1:0]		BCPB[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ASWTRG[1:0]		AEEVT[1:0]		ACPC[1:0]		ACPA[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WAVE	WAVSEL[1:0]		ENETRG	EEVT[1:0]		EEVTEDG[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CPCDIS	CPCSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:30 – BSWTRG[1:0] Software Trigger Effect on TIOBx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 29:28 – BEEVT[1:0] External Event Effect on TIOBx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 27:26 – BCPC[1:0] RC Compare Effect on TIOBx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 25:24 – BCPB[1:0] RB Compare Effect on TIOBx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Downloaded from Arrow.com.

Bits 23:22 – ASWTRG[1:0] Software Trigger Effect on TIOAx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 21:20 – AEEVT[1:0] External Event Effect on TIOAx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 19:18 – ACPC[1:0] RC Compare Effect on TIOAx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 17:16 – ACPA[1:0] RA Compare Effect on TIOAx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bit 15 – WAVE Waveform Mode

Value	Description
0	Waveform mode is disabled (Capture mode is enabled).
1	Waveform mode is enabled.

Bits 14:13 – WAVSEL[1:0] Waveform Selection

Value	Name	Description
0	UP	UP mode without automatic trigger on RC Compare
1	UPDOWN	UPDOWN mode without automatic trigger on RC Compare
2	UP_RC	UP mode with automatic trigger on RC Compare
3	UPDOWN_RC	UPDOWN mode with automatic trigger on RC Compare

Bit 12 – ENETRG External Event Trigger Enable
Whatever the value programmed in ENETRG, the selected external event only controls the TIOAx output and TIOBx if not used as input (trigger event input or other input used).

Value	Description
0	The external event has no effect on the counter and its clock.
1	The external event resets the counter and starts the counter clock.

Bits 11:10 – EEVT[1:0] External Event Selection
Signal selected as external event.

Value	Name	Description	TIOB Direction
0	TIOB	TIOB	Input
1	XC0	XC0	Output
2	XC1	XC1	Output

Downloaded from Arrow.com.

.....continued			
Value	**Name**	**Description**	**TIOB Direction**
3	XC2	XC2	Output

Note: If TIOB is chosen as the external event signal, it is configured as an input and no longer generates waveforms and subsequently no IRQs.

Bits 9:8 – EEVTEDG[1:0] External Event Edge Selection

Value	Name	Description
0	NONE	None
1	RISING	Rising edge
2	FALLING	Falling edge
3	EDGE	Each edge

Bit 7 – CPCDIS Counter Clock Disable with RC Compare

Value	Description
0	Counter clock is not disabled when counter reaches RC.
1	Counter clock is disabled when counter reaches RC.

Bit 6 – CPCSTOP Counter Clock Stopped with RC Compare

Value	Description
0	Counter clock is not stopped when counter reaches RC.
1	Counter clock is stopped when counter reaches RC.

Bits 5:4 – BURST[1:0] Burst Signal Selection

Value	Name	Description
0	NONE	The clock is not gated by an external signal.
1	XC0	XC0 is ANDed with the selected clock.
2	XC1	XC1 is ANDed with the selected clock.
3	XC2	XC2 is ANDed with the selected clock.

Bit 3 – CLKI Clock Invert

Value	Description
0	Counter is incremented on rising edge of the clock.
1	Counter is incremented on falling edge of the clock.

Bits 2:0 – TCCLKS[2:0] Clock Selection

To operate at maximum peripheral clock frequency, see TC_EMRx.

Value	Name	Description
0	TIMER_CLOCK1	Clock selected: internal GCLK [17], GLCK[45] clock signal (from PMC)
1	TIMER_CLOCK2	Clock selected: internal MCK/8 clock signal (from PMC)
2	TIMER_CLOCK3	Clock selected: internal MCK/32 clock signal (from PMC)
3	TIMER_CLOCK4	Clock selected: internal MCK/128 clock signal (from PMC)
4	TIMER_CLOCK5	Clock selected: internal MD_SLCK clock signal (from PMC)
5	XC0	Clock selected: XC0
6	XC1	Clock selected: XC1
7	XC2	Clock selected: XC2

Downloaded from Arrow.com.

50.7.4 TC Stepper Motor Mode Register

Name: TC_SMMRx
Offset: 0x08 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: R/W

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							DOWN	GCEN
Access							R/W	R/W
Reset							0	0

Bit 1 – DOWN Down Count

Value	Description
0	Up counter.
1	Down counter.

Bit 0 – GCEN Gray Count Enable

Value	Description
0	TIOAx [x=0..2] and TIOBx [x=0..2] are driven by internal counter of channel x.
1	TIOAx [x=0..2] and TIOBx [x=0..2] are driven by a 2-bit Gray counter.

Downloaded from Arrow.com.

50.7.5 TC Register AB

Name: TC_RABx
Offset: 0x0C + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RAB[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RAB[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RAB[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RAB[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RAB[31:0] Register A or Register B
RAB contains the next unread capture Register A or Register B value in real time. It is usually read by the DMA after a request due to a valid load edge on TIOAx.
When DMA is used, the RAB register address must be configured as source address of the transfer.

Downloaded from Arrow.com.

50.7.6 TC Counter Value Register

Name: TC_CVx
Offset: 0x10 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CV[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CV[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CV[31:0] Counter Value
CV contains the counter value in real time.

Downloaded from Arrow.com.

50.7.7 TC Register A

Name: TC_RAx
Offset: 0x14 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register has access Read-only if TC_CMRx.WAVE = 0, Read/Write if TC_CMRx.WAVE = 1.
This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	RA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RA[31:0] Register A
RA contains the Register A value in real time.

Downloaded from Arrow.com.

50.7.8 TC Register B

Name: TC_RBx
Offset: 0x18 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register has access Read-only if TC_CMRx.WAVE = 0, Read/Write if TC_CMRx.WAVE = 1.
This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	RB[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RB[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RB[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RB[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RB[31:0] Register B
RB contains the Register B value in real time.

Downloaded from Arrow.com.

50.7.9 TC Register C

Name: TC_RCx
Offset: 0x1C + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	RC[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RC[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RC[31:0] Register C
RC contains the Register C value in real time.

Downloaded from Arrow.com.

50.7.10 TC Interrupt Status Register

Name: TC_SRx
Offset: 0x20 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						MTIOB	MTIOA	CLKSTA
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
								SECE
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 18 – MTIOB TIOBx Mirror

Value	Description
0	TIOBx is low. If TC_CMRx.WAVE = 0, TIOBx pin is low. If TC_CMRx.WAVE = 1, TIOBx is driven low.
1	TIOBx is high. If TC_CMRx.WAVE = 0, TIOBx pin is high. If TC_CMRx.WAVE = 1, TIOBx is driven high.

Bit 17 – MTIOA TIOAx Mirror

Value	Description
0	TIOAx is low. If TC_CMRx.WAVE = 0, TIOAx pin is low. If TC_CMRx.WAVE = 1, TIOAx is driven low.
1	TIOAx is high. If TC_CMRx.WAVE = 0, TIOAx pin is high. If TC_CMRx.WAVE = 1, TIOAx is driven high.

Bit 16 – CLKSTA Clock Enabling Status

Value	Description
0	Clock is disabled.
1	Clock is enabled.

Bit 8 – SECE Security and/or Safety Event (cleared on read)

Value	Description
0	No security or safety event occurred.
1	One or more safety or security event occurred since the last read of TC_SRx. For details on the event, see TC_SSRx.

Bit 7 – ETRGS External Trigger Status (cleared on read)

Value	Description
0	External trigger has not occurred since the last read of the Status Register.
1	External trigger has occurred since the last read of the Status Register.

Bit 6 – LDRBS RB Loading Status (cleared on read)

Downloaded from Arrow.com.

Value	Description
0	RB Load has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 1.
1	RB Load has occurred since the last read of the Status Register, if TC_CMRx.WAVE = 0.

Bit 5 – LDRAS RA Loading Status (cleared on read)

Value	Description
0	RA Load has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 1.
1	RA Load has occurred since the last read of the Status Register, if TC_CMRx.WAVE = 0.

Bit 4 – CPCS RC Compare Status (cleared on read)

Value	Description
0	RC Compare has not occurred since the last read of the Status Register.
1	RC Compare has occurred since the last read of the Status Register.

Bit 3 – CPBS RB Compare Status (cleared on read)

Value	Description
0	RB Compare has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 0.
1	RB Compare has occurred since the last read of the Status Register, if TC_CMRx.WAVE = 1.

Bit 2 – CPAS RA Compare Status (cleared on read)

Value	Description
0	RA Compare has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 0.
1	RA Compare has occurred since the last read of the Status Register, if TC_CMRx.WAVE = 1.

Bit 1 – LOVRS Load Overrun Status (cleared on read)

Value	Description
0	Load overrun has not occurred since the last read of the Status Register or TC_CMRx.WAVE = 1.
1	RA or RB have been loaded at least twice without any read of the corresponding register since the last read of the Status Register, if TC_CMRx.WAVE = 0.

Bit 0 – COVFS Counter Overflow Status (cleared on read)

Value	Description
0	No counter overflow has occurred since the last read of the Status Register.
1	A counter overflow has occurred since the last read of the Status Register.

Downloaded from Arrow.com.

50.7.11 TC Interrupt Enable Register

Name: TC_IERx
Offset: 0x24 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						SECE		
Access						W		
Reset						–		

Bit	7	6	5	4	3	2	1	0
	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 10 – SECE Security and/or Safety Event Interrupt Enable

Bit 7 – ETRGS External Trigger

Bit 6 – LDRBS RB Loading

Bit 5 – LDRAS RA Loading

Bit 4 – CPCS RC Compare

Bit 3 – CPBS RB Compare

Bit 2 – CPAS RA Compare

Bit 1 – LOVRS Load Overrun

Bit 0 – COVFS Counter Overflow

Downloaded from Arrow.com.

50.7.12 TC Interrupt Disable Register

Name: TC_IDRx
Offset: 0x28 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						SECE		
Access						W		
Reset						–		

Bit	7	6	5	4	3	2	1	0
	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 10 – SECE Security and/or Safety Event Interrupt Disable

Bit 7 – ETRGS External Trigger

Bit 6 – LDRBS RB Loading

Bit 5 – LDRAS RA Loading

Bit 4 – CPCS RC Compare

Bit 3 – CPBS RB Compare

Bit 2 – CPAS RA Compare

Bit 1 – LOVRS Load Overrun

Bit 0 – COVFS Counter Overflow

Downloaded from Arrow.com.

50.7.13 TC Interrupt Mask Register

Name: TC_IMRx
Offset: 0x2C + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						SECE		
Access						R		
Reset						0		

Bit	7	6	5	4	3	2	1	0
	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 10 – SECE Security and/or Safety Event Interrupt Mask

Bit 7 – ETRGS External Trigger

Bit 6 – LDRBS RB Loading

Bit 5 – LDRAS RA Loading

Bit 4 – CPCS RC Compare

Bit 3 – CPBS RB Compare

Bit 2 – CPAS RA Compare

Bit 1 – LOVRS Load Overrun

Bit 0 – COVFS Counter Overflow

Downloaded from Arrow.com.

50.7.14 TC Extended Mode Register

Name: TC_EMRx
Offset: 0x30 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								NODIVCLK
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
			TRIGSRCB[1:0]				TRIGSRCA[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit 8 – NODIVCLK No Divided Clock

Value	Description
0	The selected clock is defined by field TCCLKS in TC_CMRx.
1	The selected clock is peripheral clock and TCCLKS field (TC_CMRx) has no effect.

Bits 5:4 – TRIGSRCB[1:0] Trigger Source for Input B

Value	Name	Description
0	EXTERNAL_TIOBx	The trigger/capture input B is driven by external pin TIOBx
1	PWMx	The trigger/capture input B is driven internally by the comparator output (see Synchronization with PWM) of the PWMx.

Bits 1:0 – TRIGSRCA[1:0] Trigger Source for Input A

Value	Name	Description
0	EXTERNAL_TIOAx	The trigger/capture input A is driven by external pin TIOAx
1	PWMx	The trigger/capture input A is driven internally by PWMx

Downloaded from Arrow.com.

50.7.15 TC Channel Status Register

Name: TC_CSRx
Offset: 0x34 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Note: The flags in this register are a copy of the similar flags in the TC_SRx register. Reading the TC_CSRx does not perform a clear-on-read of TC_SRx flags.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						MTIOB	MTIOA	CLKSTA
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 18 – MTIOB TIOBx Mirror

Value	Description
0	TIOBx is low. If TC_CMRx.WAVE = 0, TIOBx is low. If TC_CMRx.WAVE = 1, TIOBx is driven low.
1	TIOBx is high. If TC_CMRx.WAVE = 0, TIOBx is high. If TC_CMRx.WAVE = 1, TIOBx is driven high.

Bit 17 – MTIOA TIOAx Mirror

Value	Description
0	TIOAx is low. If TC_CMRx.WAVE = 0, TIOAx is low. If TC_CMRx.WAVE = 1, TIOAx is driven low.
1	TIOAx is high. If TC_CMRx.WAVE = 0, TIOAx is high. If TC_CMRx.WAVE = 1, TIOAx is driven high.

Bit 16 – CLKSTA Clock Enabling Status

Value	Description
0	Clock is disabled.
1	Clock is enabled.

Downloaded from Arrow.com.

50.7.16 TC Safety Status Register

Name: TC_SSRx
Offset: 0x38 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS				SWETYP[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SWE	SEQE	CGD	WPVS
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – ECLASS Software Error Class

Value	Name	Description
0	WARNING	An abnormal access that does not have any impact.
1	ERROR	An abnormal access that may have an impact.

Bits 27:24 – SWETYP[3:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	TC Channel x is enabled and a write-only register has been read (Warning).
1	WRITE_RO	TC Channel x is enabled and a write access has been performed on a read-only register (Warning).
2	UNDEF_RW	Access to an undefined address of the TC (Warning).
3	W_RARB_CAPT	TC_RAx or TC_RBx are written while channel is enabled and configured in capture mode (Error).

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source (cleared on read)
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of TC_SSRx.
1	A software error has occurred since the last read of TC_SSRx. The field SWETYP details the type of software error encountered.

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No internal counter error has occurred since the last read of TC_SSRx. In normal operating conditions, SEQE is cleared.

Downloaded from Arrow.com.

Value	Description
1	An internal counter error has occurred since the last read of TC_SSRx. This flag can be set only under abnormal operating conditions resulting in clock glitch, etc. The detection is enabled if TC_CSRx.CLKSTA=1, TC_CMRx.WAVE=1, TC_CMRx.CPCTRG=1 and flag is set if TC_CVx > TC_RCx.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	The clock monitoring has not been corrupted since the last read of TC_SSRx.
1	The clock monitoring has been corrupted since the last read of TC_SSRx. This flag can be set under abnormal operating conditions.

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protection violation has occurred since the last read of TC_SSRx.
1	A write protection violation has occurred since the last read of TC_SSRx. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

50.7.17 TC Block Control Register

Name: TC_BCR
Offset: 0xC0
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SYNC
Access								W
Reset								–

Bit 0 – SYNC Synchro Command

Value	Description
0	No effect.
1	Asserts the SYNC signal which generates a software trigger simultaneously for each of the channels.

Downloaded from Arrow.com.

50.7.18 TC Block Mode Register

Name: TC_BMR
Offset: 0xC4
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
							MAXFILT[5:4]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	MAXFILT[3:0]						IDXPHB	SWAP
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	15	14	13	12	11	10	9	8
	INVIDX	INVB	INVA	EDGPHA	QDTRANS	SPEEDEN	POSEN	QDEN
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			TC2XC2S[1:0]		TC1XC1S[1:0]		TC0XC0S[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 25:20 – MAXFILT[5:0] Maximum Filter
Pulses with a period shorter than MAXFILT+1 peripheral clock cycles are discarded. For more details on MAXFILT constraints, see Input Preprocessing.

Value	Description
1-63	Defines the filtering capabilities.

Bit 17 – IDXPHB Index Pin is PHB Pin

Value	Description
0	IDX pin of the rotary sensor must drive TIOA1.
1	IDX pin of the rotary sensor must drive TIOB0.

Bit 16 – SWAP Swap PHA and PHB

Value	Description
0	No swap between PHA and PHB.
1	Swap PHA and PHB internally, prior to driving the QDEC.

Bit 15 – INVIDX Inverted Index

Value	Description
0	IDX (TIOA1) is directly driving the QDEC.
1	IDX is inverted before driving the QDEC.

Bit 14 – INVB Inverted PHB

Value	Description
0	PHB (TIOB0) is directly driving the QDEC.
1	PHB is inverted before driving the QDEC.

Bit 13 – INVA Inverted PHA

Downloaded from Arrow.com.

Value	Description
0	PHA (TIOA0) is directly driving the QDEC.
1	PHA is inverted before driving the QDEC.

Bit 12 – EDGPHA Edge on PHA Count Mode

Value	Description
0	Edges are detected on PHA only.
1	Edges are detected on both PHA and PHB.

Bit 11 – QDTRANS Quadrature Decoding Transparent

Value	Description
0	Full quadrature decoding logic is active (direction change detected).
1	Quadrature decoding logic is inactive (direction change inactive) but input filtering and edge detection are performed.

Bit 10 – SPEEDEN Speed Enabled

Value	Description
0	Disabled.
1	Enables the speed measure on channel 0, the time base being provided by channel 2.

Bit 9 – POSEN Position Enabled

Value	Description
0	Disable position.
1	Enables the position measure on channel 0 and 1.

Bit 8 – QDEN Quadrature Decoder Enabled
Quadrature decoding (direction change) can be disabled using QDTRANS bit.
One of the POSEN or SPEEDEN bits must be also enabled.

Value	Description
0	Disabled.
1	Enables the QDEC (filter, edge detection and quadrature decoding).

Bits 5:4 – TC2XC2S[1:0] External Clock Signal 2 Selection

Value	Name	Description
0	TCLK2	Signal connected to XC2: TCLK2
1	–	Reserved
2	TIOA0	Signal connected to XC2: TIOA0
3	TIOA1	Signal connected to XC2: TIOA1

Bits 3:2 – TC1XC1S[1:0] External Clock Signal 1 Selection

Value	Name	Description
0	TCLK1	Signal connected to XC1: TCLK1
1	–	Reserved
2	TIOA0	Signal connected to XC1: TIOA0
3	TIOA2	Signal connected to XC1: TIOA2

Bits 1:0 – TC0XC0S[1:0] External Clock Signal 0 Selection

Value	Name	Description
0	TCLK0	Signal connected to XC0: TCLK0
1	–	Reserved
2	TIOA1	Signal connected to XC0: TIOA1
3	TIOA2	Signal connected to XC0: TIOA2

Downloaded from Arrow.com.

50.7.19 TC QDEC Interrupt Enable Register

Name: TC_QIER
Offset: 0xC8
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FMP	FIDX	FPHB	FPHA		QERR	DIRCHG	IDX
Access	W	W	W	W		W	W	W
Reset	–	–	–	–		–	–	–

Bit 7 – FMP Filtered Missing Pulse

Value	Description
0	No effect.
1	Enables the interrupt when phase A or phase B has a corrected missing pulse.

Bit 6 – FIDX Filtered Index Line

Value	Description
0	No effect.
1	Enables the interrupt when index line has a filtered contamination.

Bit 5 – FPHB Filtered Phase B Line

Value	Description
0	No effect.
1	Enables the interrupt when phase B line has a filtered contamination.

Bit 4 – FPHA Filtered Phase A Line

Value	Description
0	No effect.
1	Enables the interrupt when phase A line has a filtered contamination.

Bit 2 – QERR Quadrature Error

Value	Description
0	No effect.
1	Enables the interrupt when a quadrature error occurs on PHA, PHB.

Bit 1 – DIRCHG Direction Change

Value	Description
0	No effect.

Downloaded from Arrow.com.

Value	Description
1	Enables the interrupt when a change on rotation direction is detected.

Bit 0 – IDX Index

Value	Description
0	No effect.
1	Enables the interrupt when a rising edge occurs on IDX input.

Downloaded from Arrow.com.

50.7.20 TC QDEC Interrupt Disable Register

Name: TC_QIDR
Offset: 0xCC
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FMP	FIDX	FPHB	FPHA		QERR	DIRCHG	IDX
Access	W	W	W	W		W	W	W
Reset	–	–	–	–		–	–	–

Bit 7 – FMP Filtered Missing Pulse

Value	Description
0	No effect.
1	Disables the interrupt when phase A or phase B has a corrected missing pulse.

Bit 6 – FIDX Filtered Index Line

Value	Description
0	No effect.
1	Disables the interrupt when index line has a filtered contamination.

Bit 5 – FPHB Filtered Phase B Line

Value	Description
0	No effect.
1	Disables the interrupt when phase B line has a filtered contamination.

Bit 4 – FPHA Filtered Phase A Line

Value	Description
0	No effect.
1	Disables the interrupt when phase A line has a filtered contamination.

Bit 2 – QERR Quadrature Error

Value	Description
0	No effect.
1	Disables the interrupt when a quadrature error occurs on PHA, PHB.

Bit 1 – DIRCHG Direction Change

Value	Description
0	No effect.

Downloaded from Arrow.com.

Value	Description
1	Disables the interrupt when a change on rotation direction is detected.

Bit 0 – IDX Index

Value	Description
0	No effect.
1	Disables the interrupt when a rising edge occurs on IDX input.

Downloaded from Arrow.com.

50.7.21 TC QDEC Interrupt Mask Register

Name: TC_QIMR
Offset: 0xD0
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FMP	FIDX	FPHB	FPHA		QERR	DIRCHG	IDX
Access	R	R	R	R		R	R	R
Reset	0	0	0	0		0	0	0

Bit 7 – FMP Filtered Missing Pulse

Value	Description
0	The interrupt on auto-corrected missing pulse is disabled.
1	The interrupt on auto-corrected missing pulse is enabled.

Bit 6 – FIDX Filtered Index Line

Value	Description
0	The interrupt on index line filtered contamination is disabled.
1	The interrupt on index line filtered contamination is enabled.

Bit 5 – FPHB Filtered Phase B Line

Value	Description
0	The interrupt on phase B line filtered contamination is disabled.
1	The interrupt on phase B line filtered contamination is enabled.

Bit 4 – FPHA Filtered Phase A Line

Value	Description
0	The interrupt on phase A line filtered contamination is disabled.
1	The interrupt on phase A line filtered contamination is enabled.

Bit 2 – QERR Quadrature Error

Value	Description
0	The interrupt on quadrature error is disabled.
1	The interrupt on quadrature error is enabled.

Bit 1 – DIRCHG Direction Change

Value	Description
0	The interrupt on rotation direction change is disabled.
1	The interrupt on rotation direction change is enabled.

Downloaded from Arrow.com.

Bit 0 – IDX Index

Value	Description
0	The interrupt on IDX input is disabled.
1	The interrupt on IDX input is enabled.

Downloaded from Arrow.com.

50.7.22 TC QDEC Interrupt Status Register

Name: TC_QISR
Offset: 0xD4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								DIR
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
	FMP	FIDX	FPHB	FPHA		QERR	DIRCHG	IDX
Access	R	R	R	R		R	R	R
Reset	0	0	0	0		0	0	0

Bit 8 – DIR Direction
Returns an image of the current rotation direction.

Bit 7 – FMP Filtered Missing Pulse

Value	Description
0	No correction of missing pulse on phase A or B lines occured since the last read of TC_QISR.
1	A correction of missing pulse on phase A or B lines occured since the last read of TC_QISR.

Bit 6 – FIDX Filtered Index Line

Value	Description
0	No filtered contamination on index line since the last read of TC_QISR.
1	A contamination has been successfully on index line since the last read of TC_QISR.

Bit 5 – FPHB Filtered Phase B Line

Value	Description
0	No filtered contamination on phase B line since the last read of TC_QISR.
1	A contamination has been successfully on phase B line since the last read of TC_QISR.

Bit 4 – FPHA Filtered Phase A Line

Value	Description
0	No filtered contamination on phase A line since the last read of TC_QISR.
1	A contamination has been successfully on phase A line since the last read of TC_QISR.

Bit 2 – QERR Quadrature Error

Value	Description
0	No quadrature error since the last read of TC_QISR.
1	A quadrature error occurred since the last read of TC_QISR.

Downloaded from Arrow.com.

Bit 1 – DIRCHG Direction Change

Value	Description
0	No change on rotation direction since the last read of TC_QISR.
1	The rotation direction changed since the last read of TC_QISR.

Bit 0 – IDX Index

Value	Description
0	No Index input change since the last read of TC_QISR.
1	The IDX input has changed since the last read of TC_QISR.

Downloaded from Arrow.com.

50.7.23 TC Fault Mode Register

Name: TC_FMR
Offset: 0xD8
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ENCF1	ENCF0
Access							R/W	R/W
Reset							0	0

Bit 1 – ENCF1 Enable Compare Fault Channel 1

Value	Description
0	Disables the FAULT output source (CPCS flag) from channel 1.
1	Enables the FAULT output source (CPCS flag) from channel 1.

Bit 0 – ENCF0 Enable Compare Fault Channel 0

Value	Description
0	Disables the FAULT output source (CPCS flag) from channel 0.
1	Enables the FAULT output source (CPCS flag) from channel 0.

Downloaded from Arrow.com.

50.7.24 TC QDEC Status Register

Name: TC_QSR
Offset: 0xDC
Reset: 0x00000000
Property: Read-only

Note: The flag in this register is a copy of the similar flag in the TC_QISR register. Reading the TC_QSR does not perform a clear-on-read of TC_QISR flags.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 8 – DIR Direction
Returns an image of the current rotation direction.

Downloaded from Arrow.com.

50.7.25 TC Write Protection Mode Register

Name: TC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				FIRSTE		WPCREN	WPITEN	WPEN
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in TC_SSRx.WPVSRC and the last software control error type is reported in TC_SSRx.SWETYP. The TC_SRx.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in TC_SSRx.WPVSRC and only the first software control error type is reported in TC_SSRx.SWETYP. The TC_SRx.SECE flag is set at the first error occurrence within a series.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x54494D (“TIM” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x54494D (“TIM” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x54494D (“TIM” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x54494D (“TIM” in ASCII).

Bit 0 – WPEN Write Protection Enable
The Timer Counter clock of the first channel must be enabled to access this register.
See Register Write Protection for a list of registers that can be write-protected and Timer Counter clock conditions.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x54494D (“TIM” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x54494D (“TIM” in ASCII).

Downloaded from Arrow.com.

51. Pulse Width Modulation Controller (PWM)

51.1 Description

The Pulse Width Modulation Controller (PWM) controls several channels independently. Each channel controls one square output waveform. Characteristics of the output waveform such as period, duty-cycle and polarity are configurable through the user interface. Each channel selects and uses one of the clocks provided by the clock generator. The clock generator provides several clocks resulting from the division of the PWM macrocell master clock.

Channels can be synchronized, to generate non overlapped waveforms. All channels integrate a double buffering system in order to prevent an unexpected output waveform while modifying the period or the duty-cycle.

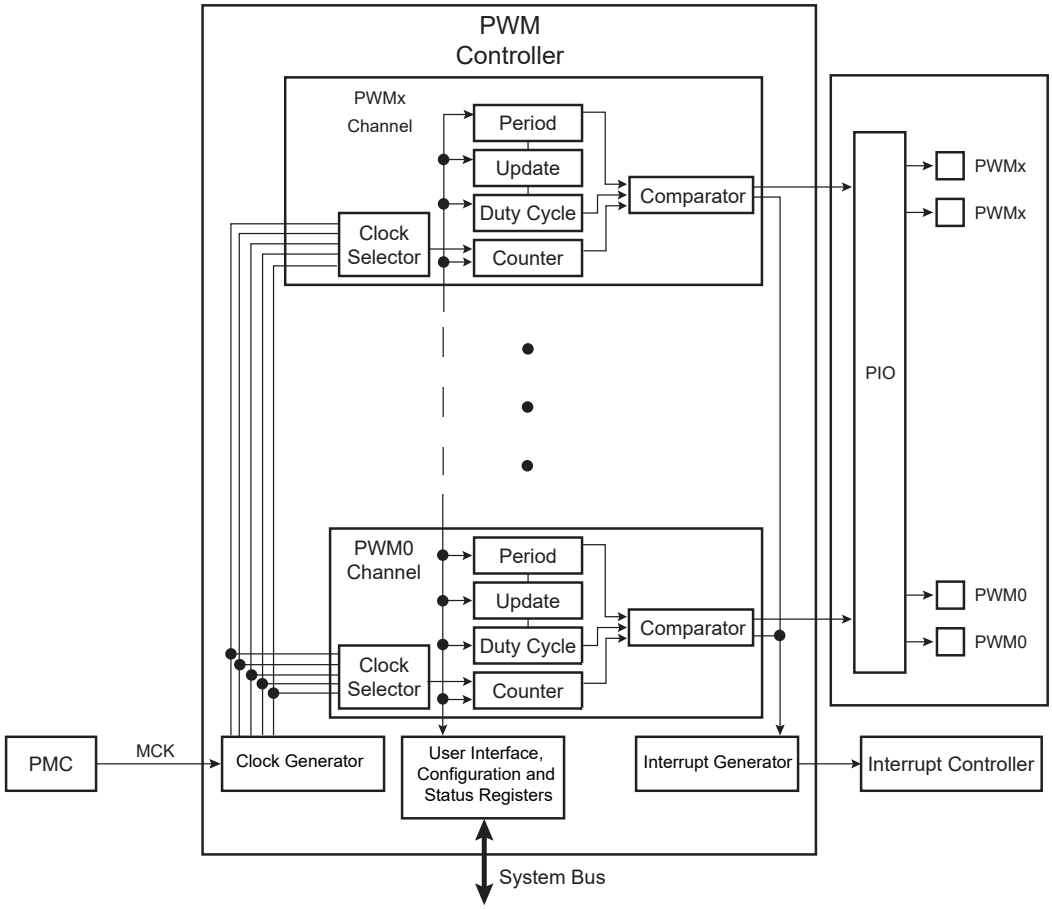
51.2 Embedded Characteristics

- 4 Channels
- One 32-bit Counter Per Channel
- Common Clock Generator Providing Thirteen Different Clocks
  - A Modulo n Counter Providing Eleven Clocks
  - Two Independent Linear Dividers Working on Modulo n Counter Outputs
- Independent Channels
  - Independent Enable Disable Command for Each Channel
  - Independent Clock Selection for Each Channel
  - Independent Period and Duty Cycle for Each Channel
  - Double Buffering of Period or Duty Cycle for Each Channel
  - Programmable Selection of The Output Waveform Polarity for Each Channel
  - Programmable Center or Left Aligned Output Waveform for Each Channel Block Diagram

Downloaded from Arrow.com.

51.3 Block Diagram

Figure 51-1. Pulse Width Modulation Controller Block Diagram



51.4 I/O Lines Description

Each channel outputs one waveform on one external I/O line.

Table 51-1. I/O Line Description

Name	Description	Type
PWMx	PWM Waveform Output for channel x	Output

51.5 Product Dependencies

51.5.1 I/O Lines

The pins used for interfacing the PWM may be multiplexed with PIO lines. The programmer must first program the PIO controller to assign the desired PWM pins to their peripheral function. If I/O lines of the PWM are not used by the application, they can be used for other purposes by the PIO controller.

All of the PWM outputs may or may not be enabled. If an application requires only four channels, then only four PIO lines will be assigned to PWM outputs.

Downloaded from Arrow.com.

51.5.2 Power Management

The PWM is not continuously clocked. The programmer must first enable the PWM clock in the Power Management Controller (PMC) before using the PWM. However, if the application does not require PWM operations, the PWM clock can be stopped when not needed and be restarted later. In this case, the PWM will resume its operations where it left off.

All the PWM registers except PWM Channel Duty Cycle Register (PWM_CDTYx) and PWM Channel Period Register (PWM_CPRDx) can be read without the PWM peripheral clock enabled. All the registers can be written without the peripheral clock enabled.

51.5.3 Interrupt Sources

The PWM interrupt line is connected on one of the internal sources of the Interrupt Controller. Using the PWM interrupt requires the Interrupt Controller to be programmed first. Note that it is not recommended to use the PWM interrupt line in edge sensitive mode.

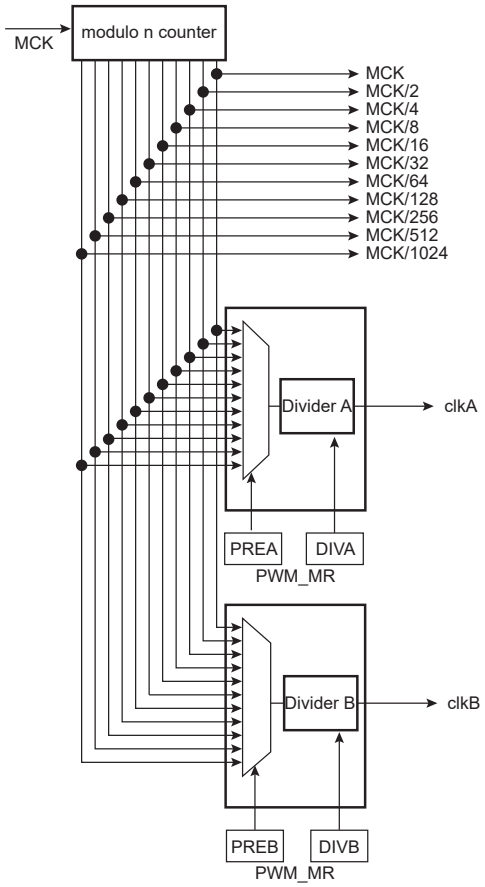
51.6 Functional Description

The PWM macrocell is primarily composed of a clock generator module and 4 channels.

- Clocked by the system clock, MCK, the clock generator module provides 13 clocks.
- Each channel can independently choose one of the clock generator outputs.
- Each channel generates an output waveform with attributes that can be defined independently for each channel through the user interface registers.

51.6.1 PWM Clock Generator

Figure 51-2. Functional View of the Clock Generator Block Diagram



Downloaded from Arrow.com.

⚠ CAUTION Before using the PWM macrocell, the programmer must first enable the PWM clock in the Power Management Controller (PMC).

The PWM macrocell master clock, MCK, is divided in the clock generator module to provide different clocks available for all channels. Each channel can independently select one of the divided clocks.

The clock generator is divided in three blocks:

- a modulo n counter which provides 11 clocks: f_{MCK}, $f_{MCK}/2$, $f_{MCK}/4$, $f_{MCK}/8$, $f_{MCK}/16$, $f_{MCK}/32$, $f_{MCK}/64$, $f_{MCK}/128$, $f_{MCK}/256$, $f_{MCK}/512$, $f_{MCK}/1024$
- two linear dividers (1, 1/2, 1/3,... 1/255) that provide two separate clocks: clkA and clkB

Each linear divider can independently divide one of the clocks of the modulo n counter. The selection of the clock to be divided is made according to the PREA (PREB) field of the PWM Mode Register (PWM_MR). The resulting clock clkA (clkB) is the clock selected divided by DIVA (DIVB) field value in PWM_MR.

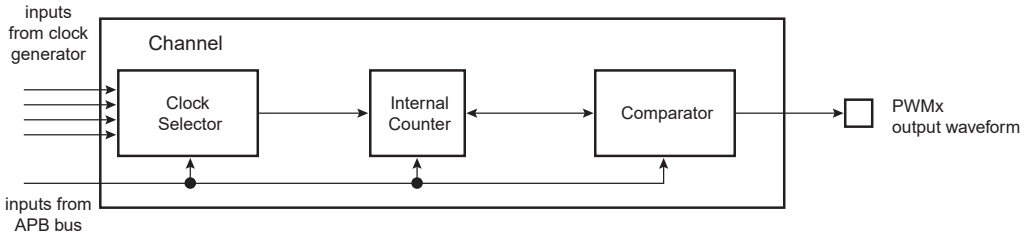
After a reset of the PWM controller, DIVA (DIVB) and PREA (PREB) in PWM_MR are set to 0. This implies that after reset clkA (clkB) are turned off.

At reset, all clocks provided by the modulo n counter are turned off except clock “clk”. This situation is also true when the PWM master clock is turned off through the Power Management Controller.

51.6.2 PWM Channel

51.6.2.1 Block Diagram

Figure 51-3. Functional View of the Channel Block Diagram



Each of the 4 channels is composed of three blocks:

- A clock selector which selects one of the clocks provided by the clock generator described in the section "PWM Clock Generator".
- An internal counter clocked by the output of the clock selector. This internal counter is incremented or decremented according to the channel configuration and comparators events. The size of the internal counter is 32 bits.
- A comparator used to generate events according to the internal counter value. It also computes the PWMx output waveform according to the configuration.

51.6.2.2 Waveform Properties

The different properties of output waveforms are:

- the ***internal clock selection***. The internal channel counter is clocked by one of the clocks provided by the clock generator described in the previous section. This channel parameter is defined in the CPRE field of the PWM Channel Mode Register (PWM_CMRx) This field is reset at 0.
- ***the waveform period***. This channel parameter is defined in the CPRD field of PWM_CPRDx.
  - If the waveform is left-aligned, then the output waveform period depends on the counter source clock and can be calculated:
    - By using the Master Clock (MCK) divided by an X given prescaler value (with X being 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024), the resulting period formula will be:

$$\frac{(X \times \mathrm{CPRD})}{\mathrm{MCK}}$$

Downloaded from Arrow.com.

- By using a Master Clock divided by one of both DIVA or DIVB divider, the formula becomes, respectively:

$$\frac{(X*\text{CPRD}*\text{DIVA})}{\text{MCK}} \text{ or } \frac{(X*\text{CPRD}*\text{DIVB})}{\text{MCK}}$$

– If the waveform is center-aligned then the output waveform period depends on the counter source clock and can be calculated:
  - By using the Master Clock (MCK) divided by an X given prescaler value (with X being 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula will be:

$$\frac{(2 \times X \times \text{CPRD})}{\text{MCK}}$$

  - By using a Master Clock divided by one of both DIVA or DIVB divider, the formula becomes, respectively:

$$\frac{(2*X*\text{CPRD}*\text{DIVA})}{\text{MCK}} \text{ or } \frac{(2*X*\text{CPRD}*\text{DIVB})}{\text{MCK}}$$

- ***the waveform duty cycle***. This channel parameter is defined in the CDTY field of PWM_CDTYx. If the waveform is left-aligned, then:

$$\text{duty cycle} = \frac{(\text{PWM_period} - (\text{CDTY} \times \text{SRC_period}))}{\text{PWM_period}}$$

If the waveform is center-aligned, then:

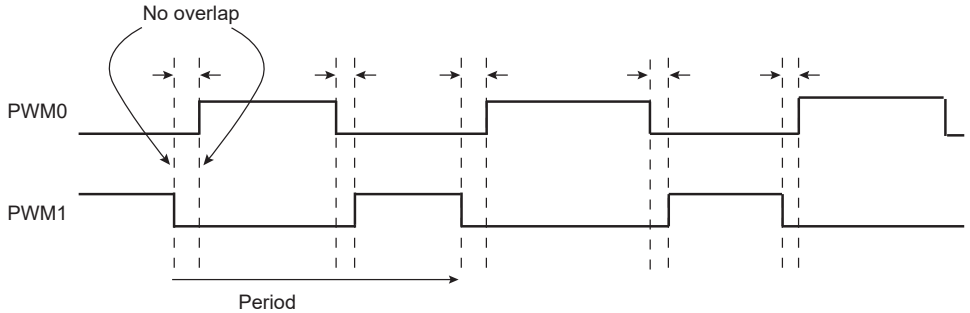
$$\text{duty cycle} = \frac{\left(\left(\frac{PWM_period}{2}\right) - \left(\text{CDTY} \times \text{SRC_period}\right)\right)}{\left(\frac{PWM_period}{2}\right)}$$

where:

"PWM_period" is the PWM output period and "SRC_period" is the PWM source clock period. "SRC_period" depends on the peripheral clock period and on PWM_CMRx.CPRE, PWM_CLK.(PREA, PREB, DIVA, DIVB), PWM_CPRDx.CPRD and PWM_CMRx.CPRE as per previous formulas.

- ***the waveform polarity***. At the beginning of the period, the signal can be at high or low level. This property is defined in the CPOL field of the PWM_CMRx. By default the signal starts by a low level.
- ***the waveform alignment***. The output waveform can be left or center-aligned. Center-aligned waveforms can be used to generate non overlapped waveforms. This property is defined in the CALG field of the PWM_CMRx. The default mode is left-aligned.

Figure 51-4. Non Overlapped Center-aligned Waveforms



Note: See the figure below for a detailed description of center-aligned waveforms.

When center-aligned, the internal channel counter increases up to CPRD and decreases down to 0. This ends the period.

When left-aligned, the internal channel counter increases up to CPRD and is reset. This ends the period.

Downloaded from Arrow.com.

Thus, for the same CPRD value, the period for a center-aligned channel is twice the period for a left-aligned channel.

Waveforms are fixed at 0 when:

- CDTY = CPRD and CPOL = 0
- CDTY = 0 and CPOL = 1

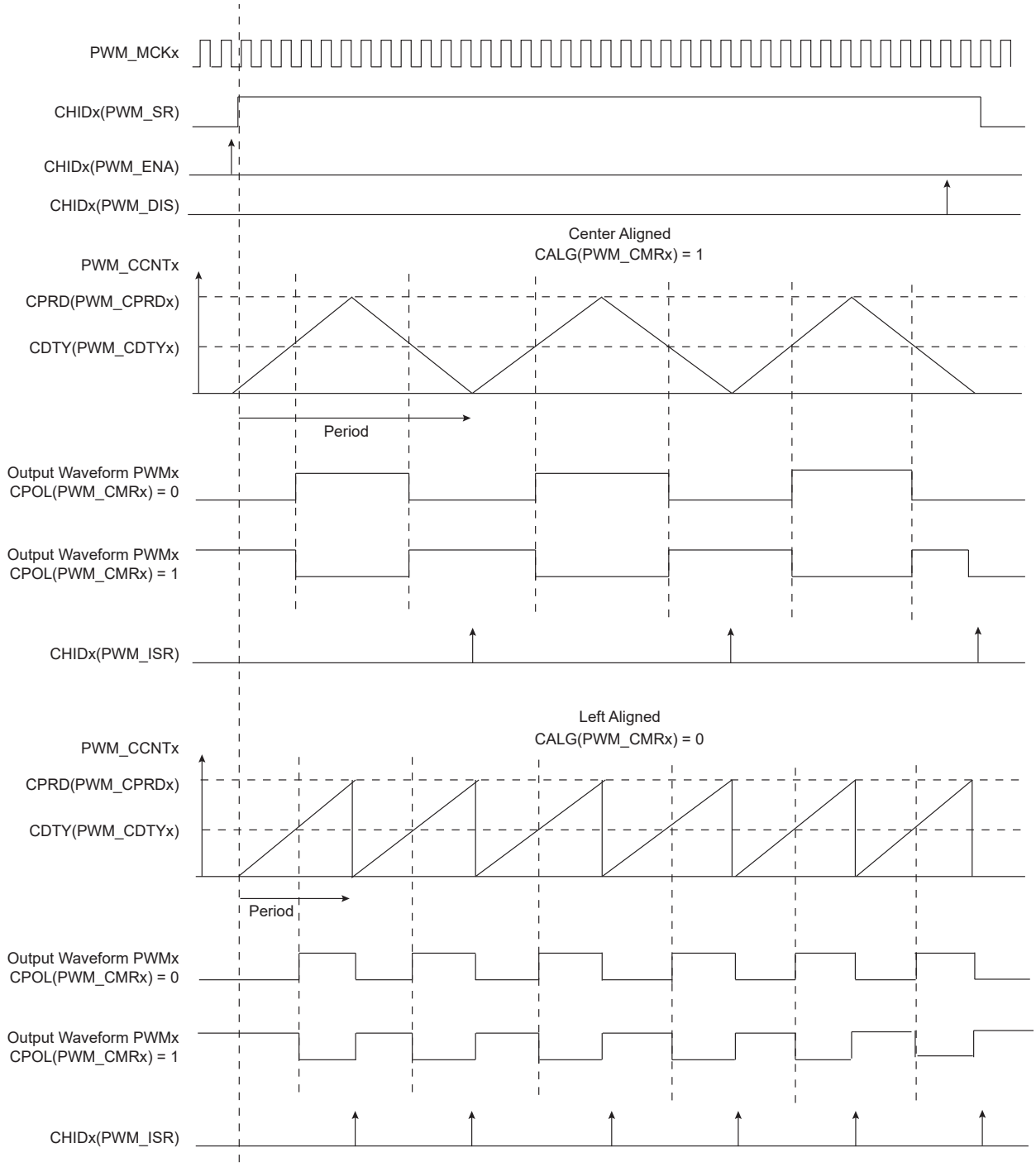
Waveforms are fixed at 1 (once the channel is enabled) when:

- CDTY = 0 and CPOL = 0
- CDTY = CPRD and CPOL = 1

The waveform polarity must be set before enabling the channel. This immediately affects the channel output level. Changes on channel polarity are not taken into account while the channel is enabled.

Downloaded from Arrow.com.

Figure 51-5. Waveform Properties



51.6.3 PWM Controller Operations

51.6.3.1 Initialization

Before enabling the output channel, this channel must have been configured by the software application:

- Configuration of the clock generator if DIVA and DIVB are required
- Selection of the clock for each channel (CPRE field in PWM_CMRx)
- Configuration of the waveform alignment for each channel (CALG field in PWM_CMRx)

Downloaded from Arrow.com.

- Configuration of the period for each channel (CPRD in PWM_CPRDx). Writing in PWM_CPRDx is possible while the channel is disabled. After validation of the channel, the user must use the PWM Channel Update Register (PWM_CUPDx) to update PWM_CPRDx as explained below.
- Configuration of the duty cycle for each channel (CDTY in PWM_CDTYx). Writing in PWM_CDTYx is possible while the channel is disabled. After validation of the channel, the user must use PWM_CUPDx to update PWM_CDTYx as explained below.
- Configuration of the output waveform polarity for each channel (CPOL in PWM_CMRx)
- Enable Interrupts (set CHIDx in the PWM Interrupt Enable Register (PWM_IER))
- Enable the PWM channel (set CHIDx in the PWM Enable Register (PWM_ENA))

It is possible to synchronize different channels by enabling them at the same time by means of simultaneously setting several CHIDx bits in PWM_ENA.

- In such a situation, all channels may have the same clock selector configuration and the same period specified.

51.6.3.2 Source Clock Selection Criteria

The large number of source clocks can make selection difficult. The relationship between the value in the Channel Period Register (PWM_CPRDx) and the Channel Duty Cycle Register (PWM_CDTYx) can help the user in choosing. The event number written in PWM_CPRDx gives the PWM accuracy. The Duty Cycle quantum cannot be lower than 1/PWM_CPRDx value. The higher the value of PWM_CPRDx, the greater the PWM accuracy.

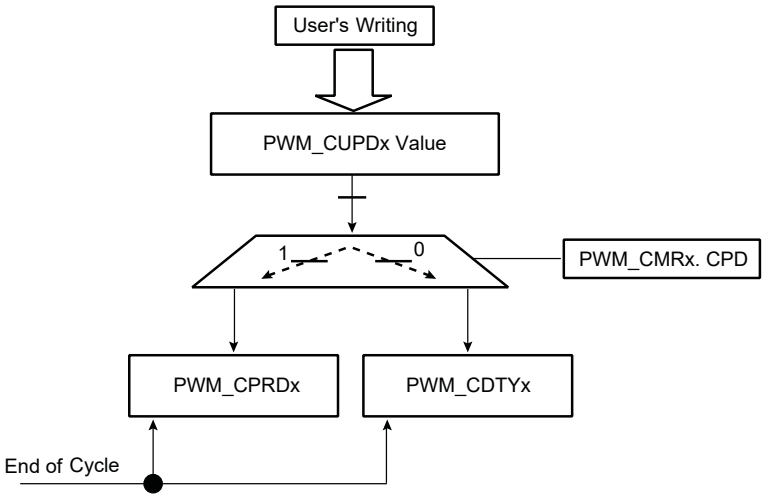
For example, if the user sets 15 (in decimal) in PWM_CPRDx, the user is able to set a value between 1 up to 14 in PWM_CDTYx. The resulting duty cycle quantum cannot be lower than 1/15 of the PWM period.

51.6.3.3 Changing the Duty Cycle or the Period

It is possible to modulate the output waveform duty cycle or period.

To prevent unexpected output waveform, the user must use PWM_CUPDx to change waveform parameters while the channel is still enabled. The user can write a new period value or duty cycle value in PWM_CUPDx. This register holds the new value until the end of the current cycle and updates the value for the next cycle. Depending on the CPD field in PWM_CMRx, PWM_CUPDx either updates PWM_CPRDx or PWM_CDTYx. Note that even if the update register is used, the period must not be smaller than the duty cycle.

Figure 51-6. Synchronized Period or Duty Cycle Update



To prevent overwriting the PWM_CUPDx by software, the user can use status events in order to synchronize his software. Two methods are possible. In both, the user must enable the dedicated interrupt in PWM_IER at PWM Controller level.

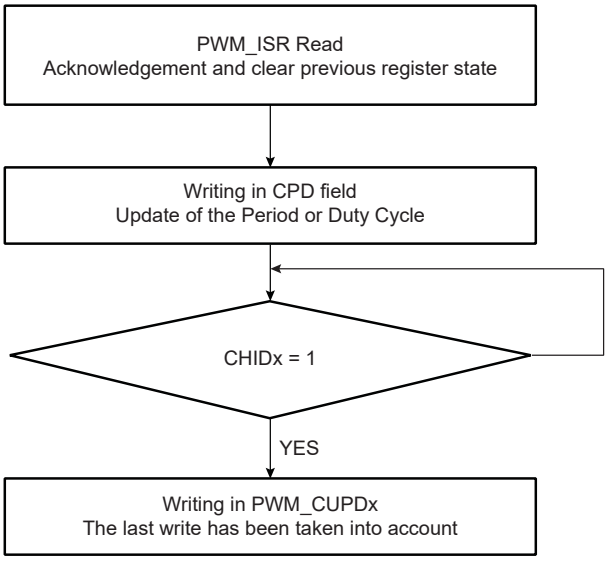
The first method (polling method) consists of reading the relevant status bit in the PWM Interrupt Status Register (PWM_ISR) according to the enabled channel(s). See the figure "Polling Method".

The second method uses an Interrupt Service Routine associated with the PWM channel.

Downloaded from Arrow.com.

Note: Reading the PWM_ISR automatically clears CHIDx flags.

Figure 51-7. Polling Method



Note: Polarity and alignment can be modified only when the channel is disabled.

Downloaded from Arrow.com.

51.7 Register Summary

Offset	Name	Bit Pos.								
0x00	PWM_MR	31:24					PREB[3:0]			
		23:16	DIVB[7:0]							
		15:8					PREA[3:0]			
		7:0	DIVA[7:0]							
0x04	PWM_ENA	31:24								
		23:16								
		15:8								
		7:0					CHID3	CHID2	CHID1	CHID0
0x08	PWM_DIS	31:24								
		23:16								
		15:8								
		7:0					CHID3	CHID2	CHID1	CHID0
0x0C	PWM_SR	31:24								
		23:16								
		15:8								
		7:0					CHID3	CHID2	CHID1	CHID0
0x10	PWM_IER	31:24								
		23:16								
		15:8								
		7:0					CHID3	CHID2	CHID1	CHID0
0x14	PWM_IDR	31:24								
		23:16								
		15:8								
		7:0					CHID3	CHID2	CHID1	CHID0
0x18	PWM_IMR	31:24								
		23:16								
		15:8								
		7:0					CHID3	CHID2	CHID1	CHID0
0x1C	PWM_ISR	31:24								
		23:16								
		15:8								
		7:0					CHID3	CHID2	CHID1	CHID0
0x20 ... 0x01FF	Reserved									
0x0200	PWM_CMR0	31:24								
		23:16								
		15:8						CPD	CPOL	CALG
		7:0					CPRE[3:0]			
0x0204	PWM_CDTY0	31:24	CDTY[31:24]							
		23:16	CDTY[23:16]							
		15:8	CDTY[15:8]							
		7:0	CDTY[7:0]							
0x0208	PWM_CPRD0	31:24	CPRD[31:24]							
		23:16	CPRD[23:16]							
		15:8	CPRD[15:8]							
		7:0	CPRD[7:0]							
0x020C	PWM_CCNT0	31:24	CNT[31:24]							
		23:16	CNT[23:16]							
		15:8	CNT[15:8]							
		7:0	CNT[7:0]							
0x0210	PWM_CUPD0	31:24	CUPD[31:24]							
		23:16	CUPD[23:16]							
		15:8	CUPD[15:8]							
		7:0	CUPD[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x0214 ... 0x021F	Reserved									
0x0220	PWM_CMR1	31:24								
		23:16								
		15:8						CPD	CPOL	CALG
		7:0					CPRE[3:0]			
0x0224	PWM_CDTY1	31:24	CDTY[31:24]							
		23:16	CDTY[23:16]							
		15:8	CDTY[15:8]							
		7:0	CDTY[7:0]							
0x0228	PWM_CPRD1	31:24	CPRD[31:24]							
		23:16	CPRD[23:16]							
		15:8	CPRD[15:8]							
		7:0	CPRD[7:0]							
0x022C	PWM_CCNT1	31:24	CNT[31:24]							
		23:16	CNT[23:16]							
		15:8	CNT[15:8]							
		7:0	CNT[7:0]							
0x0230	PWM_CUPD1	31:24	CUPD[31:24]							
		23:16	CUPD[23:16]							
		15:8	CUPD[15:8]							
		7:0	CUPD[7:0]							
0x0234 ... 0x023F	Reserved									
0x0240	PWM_CMR2	31:24								
		23:16								
		15:8						CPD	CPOL	CALG
		7:0					CPRE[3:0]			
0x0244	PWM_CDTY2	31:24	CDTY[31:24]							
		23:16	CDTY[23:16]							
		15:8	CDTY[15:8]							
		7:0	CDTY[7:0]							
0x0248	PWM_CPRD2	31:24	CPRD[31:24]							
		23:16	CPRD[23:16]							
		15:8	CPRD[15:8]							
		7:0	CPRD[7:0]							
0x024C	PWM_CCNT2	31:24	CNT[31:24]							
		23:16	CNT[23:16]							
		15:8	CNT[15:8]							
		7:0	CNT[7:0]							
0x0250	PWM_CUPD2	31:24	CUPD[31:24]							
		23:16	CUPD[23:16]							
		15:8	CUPD[15:8]							
		7:0	CUPD[7:0]							
0x0254 ... 0x025F	Reserved									
0x0260	PWM_CMR3	31:24								
		23:16								
		15:8						CPD	CPOL	CALG
		7:0					CPRE[3:0]			
0x0264	PWM_CDTY3	31:24	CDTY[31:24]							
		23:16	CDTY[23:16]							
		15:8	CDTY[15:8]							
		7:0	CDTY[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.									
0x0268	PWM_CPRD3	31:24	CPRD[31:24]								
		23:16	CPRD[23:16]								
		15:8	CPRD[15:8]								
		7:0	CPRD[7:0]								
0x026C	PWM_CCNT3	31:24	CNT[31:24]								
		23:16	CNT[23:16]								
		15:8	CNT[15:8]								
		7:0	CNT[7:0]								
0x0270	PWM_CUPD3	31:24	CUPD[31:24]								
		23:16	CUPD[23:16]								
		15:8	CUPD[15:8]								
		7:0	CUPD[7:0]								

Downloaded from Arrow.com.

51.7.1 PWM Mode Register

Name: PWM_MR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					PREB[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIVB[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					PREA[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIVA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:24 – PREB[3:0] CLKB Source Clock Selection

Value	Name	Description
0000	MCK	Master Clock
0001	MCKDIV2	Master Clock divided by 2
0010	MCKDIV4	Master Clock divided by 4
0011	MCKDIV8	Master Clock divided by 8
0100	MCKDIV16	Master Clock divided by 16
0101	MCKDIV32	Master Clock divided by 32
0110	MCKDIV64	Master Clock divided by 64
0111	MCKDIV128	Master Clock divided by 128
1000	MCKDIV256	Master Clock divided by 256
1001	MCKDIV512	Master Clock divided by 512
1010	MCKDIV1024	Master Clock divided by 1024
Other	–	Reserved

Bits 23:16 – DIVB[7:0] CLKB Divide Factor

Value	Name	Description
0	CLK_OFF	CLKB clock is turned off
1	CLK_DIV1	CLKB clock is clock selected by PREB
2-255	–	CLKB clock is clock selected by PREB divided by DIVB factor

Bits 11:8 – PREA[3:0] CLKA Source Clock Selection

Value	Name	Description
0000	MCK	Master Clock
0001	MCKDIV2	Master Clock divided by 2
0010	MCKDIV4	Master Clock divided by 4
0011	MCKDIV8	Master Clock divided by 8
0100	MCKDIV16	Master Clock divided by 16

Downloaded from Arrow.com.

Value	Name	Description
0101	MCKDIV32	Master Clock divided by 32
0110	MCKDIV64	Master Clock divided by 64
0111	MCKDIV128	Master Clock divided by 128
1000	MCKDIV256	Master Clock divided by 256
1001	MCKDIV512	Master Clock divided by 512
1010	MCKDIV1024	Master Clock divided by 1024
Other	–	Reserved

Bits 7:0 – DIVA[7:0] CLKA Divide Factor

Value	Name	Description
0	CLK_OFF	CLKA clock is turned off
1	CLK_DIV1	CLKA clock is clock selected by PREA
2-255	–	CLKA clock is clock selected by PREA divided by DIVA factor

Downloaded from Arrow.com.

51.7.2 PWM Enable Register

Name: PWM_ENA
Offset: 0x04
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CHID3	CHID2	CHID1	CHID0
Access					W	W	W	W
Reset					–	–	–	–

Bits 0, 1, 2, 3 – CHIDx Channel ID

Value	Description
0	No effect.
1	Enable PWM output for channel x.

Downloaded from Arrow.com.

51.7.3 PWM Disable Register

Name: PWM_DIS
Offset: 0x08
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CHID3	CHID2	CHID1	CHID0
Access					W	W	W	W
Reset					–	–	–	–

Bits 0, 1, 2, 3 – CHIDx Channel ID

Value	Description
0	No effect.
1	Disable PWM output for channel x.

Downloaded from Arrow.com.

51.7.4 PWM Status Register

Name: PWM_SR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CHID3	CHID2	CHID1	CHID0
Access					R	R	R	R
Reset					0	0	0	0

Bits 0, 1, 2, 3 – CHIDx Channel ID

Value	Description
0	PWM output for channel x is disabled.
1	PWM output for channel x is enabled.

Downloaded from Arrow.com.

51.7.5 PWM Interrupt Enable Register

Name: PWM_IER
Offset: 0x10
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CHID3	CHID2	CHID1	CHID0
Access					W	W	W	W
Reset					–	–	–	–

Bits 0, 1, 2, 3 – CHIDx Channel ID

Value	Description
0	No effect.
1	Enable interrupt for PWM channel x.

Downloaded from Arrow.com.

51.7.6 PWM Interrupt Disable Register

Name: PWM_IDR
Offset: 0x14
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CHID3	CHID2	CHID1	CHID0
Access					W	W	W	W
Reset					–	–	–	–

Bits 0, 1, 2, 3 – CHIDx Channel ID

Value	Description
0	No effect.
1	Disable interrupt for PWM channel x.

Downloaded from Arrow.com.

51.7.7 PWM Interrupt Mask Register

Name: PWM_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CHID3	CHID2	CHID1	CHID0
Access					R	R	R	R
Reset					0	0	0	0

Bits 0, 1, 2, 3 – CHIDx Channel ID

Value	Description
0	Interrupt for PWM channel x is disabled.
1	Interrupt for PWM channel x is enabled.

Downloaded from Arrow.com.

51.7.8 PWM Interrupt Status Register

Name: PWM_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Note: Reading PWM_ISR automatically clears CHIDx flags.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CHID3	CHID2	CHID1	CHID0
Access					R	R	R	R
Reset					0	0	0	0

Bits 0, 1, 2, 3 – CHIDx Channel ID

Value	Description
0	No new channel period has been achieved since the last read of the PWM_ISR.
1	At least one new channel period has been achieved since the last read of the PWM_ISR.

Downloaded from Arrow.com.

51.7.9 PWM Channel Mode Register x

Name: PWM_CMRx
Offset: 0x0200 + x*0x20 [x=0..3]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						CPD	CPOL	CALG
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
					CPRE[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 10 – CPD Channel Update Period

Value	Description
0	Writing to the PWM_CUPDx will modify the duty cycle at the next period start event.
1	Writing to the PWM_CUPDx will modify the period at the next period start event.

Bit 9 – CPOL Channel Polarity

Value	Description
0	The output waveform starts at a low level.
1	The output waveform starts at a high level.

Bit 8 – CALG Channel Alignment

Value	Description
0	The period is left aligned.
1	The period is center aligned.

Bits 3:0 – CPRE[3:0] Channel Prescaler

Value	Name	Description
0000	MCK	Master Clock
0001	MCKDIV2	Master Clock divided by 2
0010	MCKDIV4	Master Clock divided by 4
0011	MCKDIV8	Master Clock divided by 8
0100	MCKDIV16	Master Clock divided by 16
0101	MCKDIV32	Master Clock divided by 32
0110	MCKDIV64	Master Clock divided by 64
0111	MCKDIV128	Master Clock divided by 128
1000	MCKDIV256	Master Clock divided by 256
1001	MCKDIV512	Master Clock divided by 512
1010	MCKDIV1024	Master Clock divided by 1024

Downloaded from Arrow.com.

Value	Name	Description
1011	CLKA	Clock A
1100	CLKB	Clock B
Other	–	Reserved

Downloaded from Arrow.com.

51.7.10 PWM Channel Duty Cycle Register x

Name: PWM_CDTYx
Offset: 0x0204 + x*0x20 [x=0..3]
Reset: 0x00000000
Property: Read/Write

Only the first 32 bits (internal channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
	CDTY[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CDTY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CDTY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CDTY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CDTY[31:0] Channel Duty Cycle
Defines the waveform duty cycle. This value must be defined between 0 and CPRD (PWM_CPRx).

Downloaded from Arrow.com.

51.7.11 PWM Channel Period Register

Name: PWM_CPRDx
Offset: 0x0208 + x*0x20 [x=0..3]
Reset: 0x00000000
Property: Read/Write

Only the first 32 bits (internal channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
	CPRD[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CPRD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CPRD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CPRD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CPRD[31:0] Channel Period
If the waveform is left-aligned, then the output waveform period depends on the counter source clock and can be calculated:
– By using the Master Clock (MCK) divided by an X given prescaler value (with X being 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula will be:

$$\frac{(X \times \mathrm{CPRD})}{\mathrm{MCK}}$$

– By using a Master Clock divided by one of both DIVA or DIVB divider, the formula becomes, respectively:

$$\frac{(\mathrm{CPRD} \times \mathrm{DIVA})}{\mathrm{MCK}} \text{ or } \frac{(\mathrm{CPRD} \times \mathrm{DIVB})}{\mathrm{MCK}}$$

If the waveform is center-aligned, then the output waveform period depends on the counter source clock and can be calculated:
– By using the Master Clock (MCK) divided by an X given prescaler value (with X being 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula will be:

$$\frac{(2 \times X \times \mathrm{CPRD})}{\mathrm{MCK}}$$

– By using a Master Clock divided by one of both DIVA or DIVB divider, the formula becomes, respectively:

$$\frac{(2 \times \mathrm{CPRD} \times \mathrm{DIVA})}{\mathrm{MCK}} \text{ or } \frac{(2 \times \mathrm{CPRD} \times \mathrm{DIVB})}{\mathrm{MCK}}$$

Downloaded from Arrow.com.

51.7.12 PWM Channel Counter Register x

Name: PWM_CCNTx
Offset: 0x020C + x*0x20 [x=0..3]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CNT[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CNT[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CNT[31:0] Channel Counter Register
Internal counter value. This register is reset when:

- the channel is enabled (PWM_ENA.CHIDx = 1)
- the counter reaches CPRD value defined in PWM_CPRDx if the waveform is left aligned.

Downloaded from Arrow.com.

51.7.13 PWM Channel Update Register x

Name: PWM_CUPDx
Offset: 0x0210 + x*0x20 [x=0..3]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	CUPD[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	CUPD[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	CUPD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	CUPD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – CUPD[31:0] Channel Update Register
This register acts as a double buffer for the period or the duty cycle. This prevents an unexpected waveform when modifying the waveform period or duty-cycle.
Only the first 32 bits (internal channel counter size) are significant.
When PWM_CMRx.CPD = 0, the duty-cycle (CDTY of PWM_CDTYx) is updated with the CUPD value at the beginning of the next period.
When PWM_CMRx.CPD = 1, the period (CPRD of PWM_CPRDx) is updated with the CUPD value at the beginning of the next period.

Downloaded from Arrow.com.

52. Advanced Encryption Standard (AES)

52.1 Description

The Advanced Encryption Standard (AES) is compliant with the American FIPS (Federal Information Processing Standard) Publication 197 specification.

The AES supports the following confidentiality modes of operation for symmetrical key block cipher algorithms: ECB, CBC, OFB, CFB, CTR, and XTS), as specified in the NIST Special Publication 800-38A Recommendation, as well as Galois/Counter Mode (GCM) as specified in the NIST Special Publication 800-38D Recommendation. It is compatible with all these modes via DMA Controller channels, minimizing processor intervention for large buffer transfers.

The AES key can be either loaded by the software or loaded in an invisible manner from the software.

The 128-bit/192-bit/256-bit AES key is stored in the AES Key register made of four/six/eight 32-bit write-only AES Key Word registers (AES_KEYWR0–7). For a software-invisible key transfer, the Private Key Bus accesses the Private Key Internal Register from the TRNG. The bit PKRS in the Extended Mode register (AES_EMR) selects either AES_KEYWRx or the Private Key Internal Register.

The 128-bit input data and initialization vector (for some modes) are each stored in four 32-bit write-only AES Input Data registers (AES_IDATAR0–3) and AES Initialization Vector registers (AES_IVR0–3).

As soon as the initialization vector, the input data and the key are configured, the encryption/decryption process may be started. Then the encrypted/decrypted data are ready to be read out on the four 32-bit AES Output Data registers (AES_ODATAR0–3) or through the DMA channels.

52.2 Embedded Characteristics

- Compliant with FIPS Publication 197, Advanced Encryption Standard (AES)
- 128-bit/192-bit/256-bit Cryptographic Key
- 10/12/14 Clock Cycles Encryption/Decryption Inherent Processing Time with a 128-bit/192-bit/256-bit Cryptographic Key
- Double Input Buffer Optimizes Runtime
- Automatic Padding supported for IPSEC and SSL standards
- IPSEC and SSL Protocol Layers Improved Performances (Tightly coupled with SHA)
- Support of the Modes of Operation Specified in the NIST Special Publication 800-38A and NIST Special Publication 800-38D:
  - Electronic Codebook (ECB)
  - Cipher Block Chaining (CBC) including CBC-MAC
  - Cipher Feedback (CFB)
  - Output Feedback (OFB)
  - Counter (CTR)
  - Galois/Counter Mode (GCM)
  - XEX-Based Tweaked-Codebook Mode (XTS)
- 8, 16, 32, 64 and 128-bit Data Sizes Possible in CFB Mode
- Last Output Data Mode Allows Optimized Message Authentication Code (MAC) Generation
- Abnormal Software Access and Internal Sequencer Integrity Check Reports
- Register Write Protection
- Temporary Secure Storage for Keys
- Private Key Bus Access to the Private Key Internal Register Not Readable from any Peripheral or Software
- Connection to DMA Optimizes Data Transfers for all Operating Modes

Downloaded from Arrow.com.

52.3 Product Dependencies

52.3.1 Power Management

The AES is clocked through the Power Management Controller (PMC), so the programmer must first configure the PMC to enable the AES clock.

52.3.2 Interrupt Sources

The AES interface has an interrupt line connected to the Interrupt Controller.

Handling the AES interrupt requires programming the Interrupt Controller before configuring the AES.

52.4 Functional Description

The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information.

Encryption converts data to an unintelligible form called ciphertext. Decrypting the ciphertext converts the data back into its original form, called plaintext. The CIPHER bit in the AES Mode register (AES_MR) allows selection between the encryption and the decryption processes.

The AES is capable of using cryptographic keys of 128/192/256 bits to encrypt and decrypt data in blocks of 128 bits. This 128-bit/192-bit/256-bit key is defined in the user interface AES_KEYWRx register or in the Private Key Internal Register that is only writable from the Private Key Bus.

The input to the encryption processes of the CBC, CFB, and OFB modes includes, in addition to the plaintext, a 128-bit data block called the initialization vector (IV), which must be set in AES_IVRx. The initialization vector is used in an initial step in the encryption of a message and in the corresponding decryption of the message. AES_IVRx are also used by the CTR mode to set the counter value.

52.4.1 AES Register Endianness

In Arm processor-based products, the system bus and processors manipulate data in little-endian form. The AES interface requires little-endian format words. However, in accordance with the protocol of the FIPS 197 specification, data is collected, processed and stored by the AES algorithm in big-endian form.

The following example illustrates how to configure the AES:

If the first 64 bits of a message (according to FIPS 197, i.e., big-endian format) to be processed is 0xcafedeca_01234567, then AES_IDATAR0 and AES_IDATAR1 registers must be written with the following pattern:

- AES_IDATAR0 = 0xcadefeca
- AES_IDATAR1 = 0x67452301

52.4.2 Operating Modes

The AES supports the following modes of operation:

- ECB: Electronic Codebook
- CBC: Cipher Block Chaining
  - CBC-MAC: Useful for CMAC hardware acceleration
- OFB: Output Feedback
- CFB: Cipher Feedback
  - CFB8 (CFB where the length of the data segment is 8 bits)
  - CFB16 (CFB where the length of the data segment is 16 bits)
  - CFB32 (CFB where the length of the data segment is 32 bits)
  - CFB64 (CFB where the length of the data segment is 64 bits)
  - CFB128 (CFB where the length of the data segment is 128 bits)
- CTR: Counter

Downloaded from Arrow.com.

- GCM: Galois/Counter Mode
- XTS: XEX-based Tweaked-codebook Mode

Data pre-processing, data post-processing and data chaining for the concerned modes are performed automatically. Refer to the *NIST Special Publication 800-38A* and *NIST Special Publication 800-38D* for more complete information.

Mode selection is done by configuring AES_MR.OPMOD.

When switching from an operating mode requiring the initialization vectors (e.g. CBC, GCM) to another operating mode that does not require initialization vectors (e.g. ECB) and a message of one block has been processed, initialization vector registers (AES_IVRx) must be cleared before switching to the new mode.

In CFB mode, five data sizes are possible (8, 16, 32, 64 or 128 bits), configurable by means of AES_MR.CFBS.

In CTR mode, the size of the block counter embedded in the module is 16 bits. Therefore, there is a rollover after processing 1 Mbyte of data. If the file to be processed is greater than 1 Mbyte, this file must be split into fragments of 1 Mbyte or less for the first fragment if the initial value of the counter is greater than 0. Prior to loading the first fragment into AES_IDATARx, AES_IVRx must be fully programmed with the initial counter value. For any fragment, after the transfer is completed and prior to transferring the next fragment, AES_IVRx must be programmed with the appropriate counter value.

If the initial value of the counter is greater than 0 and the data buffer size to be processed is greater than 1 Mbyte, the size of the first fragment to be processed must be 1 Mbyte minus 16 × (initial value) to prevent a rollover of the internal 16-bit counter.

To have a sequential increment, the counter value must be programmed with the value programmed for the previous fragment + 2^{16} (or less for the first fragment).

All AES_IVRx fields must be programmed to take into account the possible carry propagation.

52.4.3 Last Output Data Mode (CBC-MAC)

This mode is used to generate cryptographic checksums on data (MAC) by means of cipher block chaining encryption algorithm (CBC-MAC algorithm for example).

The CMAC algorithm is a variant of CBC-MAC with post-processing requiring one-block encryption in ECB mode. Thus CBC-MAC is useful to accelerate CMAC.

After each end of encryption/decryption, the output data are available either on AES_ODATARx for Manual and Auto mode, or at the address specified in the receive buffer pointer for DMA mode (see the table Last Output Data Mode Behavior versus Start Modes).

AES_MR.LOD allows retrieval of only the last data of several encryption/decryption processes.

Therefore, there is no need to define a read buffer in DMA mode.

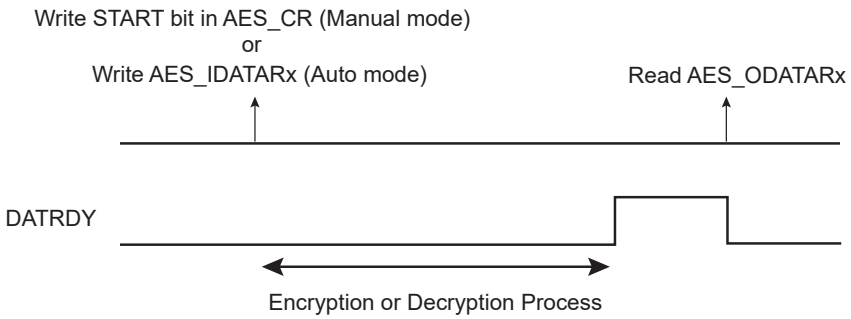
This data are only available in AES_ODATARx.

52.4.3.1 Manual and Auto Modes

52.4.3.1.1 If AES_MR.LOD = 0

The DATRDY flag is cleared when at least one AES_ODATARx is read (see the following figure).

Figure 52-1. Manual and Auto Modes with AES_MR.LOD = 0



If the user does not want to read AES_ODATARx between each encryption/decryption, the DATRDY flag will not be cleared. If the DATRDY flag is not cleared, the user cannot know the end of the following encryptions/decryptions.

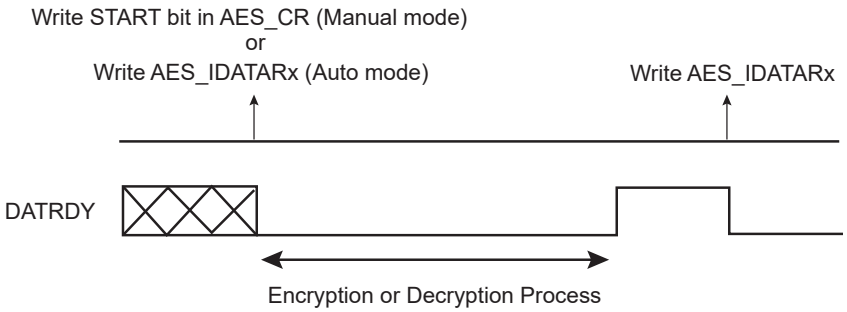
Downloaded from Arrow.com.

52.4.3.1.2 If AES_MR.LOD = 1

This mode is optimized to process AES CBC-MAC operating mode.

The DATRDY flag is cleared when at least one AES_IDATAR is written (see the following figure). No additional AES_ODATAR reads are necessary between consecutive encryptions/decryptions.

Figure 52-2. Manual and Auto Modes with AES_MR.LOD = 1



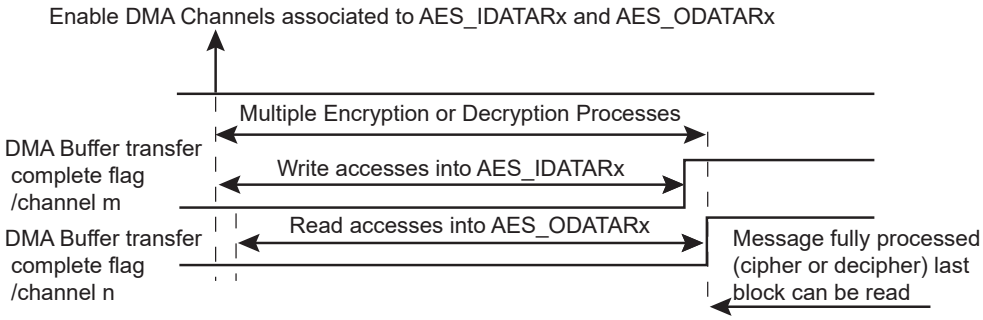
52.4.3.2 DMA Mode

52.4.3.2.1 If AES_MR.LOD = 0

This mode may be used for all AES operating modes except CBC-MAC where AES_MR.LOD = 1 mode is recommended.

The end of the encryption/decryption is indicated by the end of DMA transfer associated to AES_ODATARx (see the following figure). Two DMA channels are required: one for writing message blocks to AES_IDATARx and one to obtain the result from AES_ODATARx.

Figure 52-3. DMA Transfer with AES_MR.LOD = 0



52.4.3.2.2 If AES_MR.LOD = 1

This mode is optimized to process AES CBC-MAC operating mode.

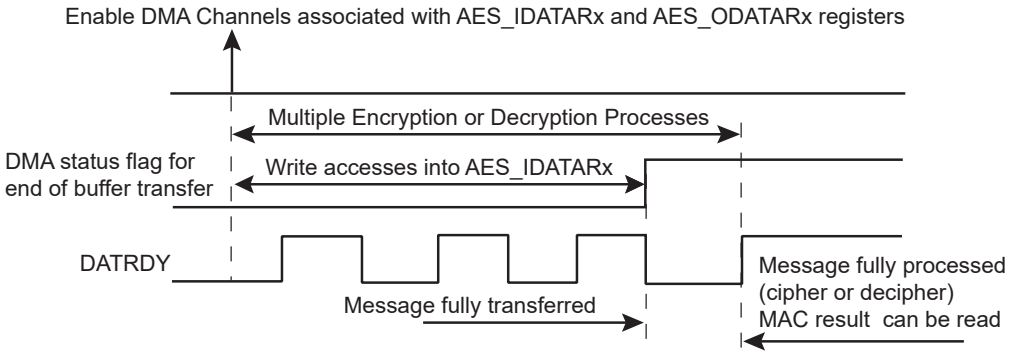
The user must first wait for the DMA buffer transfer complete flag, then for the flag DATRDY to rise to ensure that the encryption/decryption is completed (see the following figure).

The DMA receive channel must not be used. Prior to reading the CBC-MAC result, AES_MR.SMOD must be written to ‘0’. To restart a CBC-MAC on a new buffer, AES_MR.SMOD must be written to ‘2’.

The output data are only available on AES_ODATARx.

Downloaded from Arrow.com.

Figure 52-4. DMA Transfer with AES_MR.LOD = 1



The following table summarizes the different cases.

Table 52-1. Last Output Data Mode Behavior versus Start Modes

Sequence	Manual and Auto Modes		DMA Transfer	
	AES_MR.LOD = 0	AES_MR.LOD = 1	AES_MR.LOD = 0	AES_MR.LOD = 1
DATRDY Flag Clearing Condition(1)	At least one AES_ODATAR must be read	At least one AES_IDATAR must be written	Not used	Managed by the DMA
End of Encryption/ Decryption Notification	DATRDY	DATRDY	2 DMA Buffer transfer complete flags (channel m and channel n)	DMA buffer transfer complete flag, then AES DATRDY flag
Encrypted/ Decrypted Data Result Location	In AES_ODATARx	In AES_ODATARx	At the address specified in the Channel Buffer Transfer Descriptor	In AES_ODATARx

Note:

1. Depending on the mode, there are other ways of clearing the DATRDY flag. See AES_ISR.

WARNING In DMA mode, reading AES_ODATARx before the last data transfer may lead to unpredictable results.

52.4.4 Galois/Counter Mode (GCM)

52.4.4.1 Description

GCM comprises the AES engine in CTR mode along with a universal hash function (GHASH engine) that is defined over a binary Galois field to produce a message authentication tag (the AES CTR engine and the GHASH engine are depicted in the following figure.

The GHASH engine processes data packets after the AES operation. GCM assures the confidentiality of data through the AES Counter mode of operation for encryption. Authenticity of the confidential data is assured through the GHASH engine. GCM can also provide assurance of data that is not encrypted. Refer to *NIST Special Publication 800-38D* for more complete information.

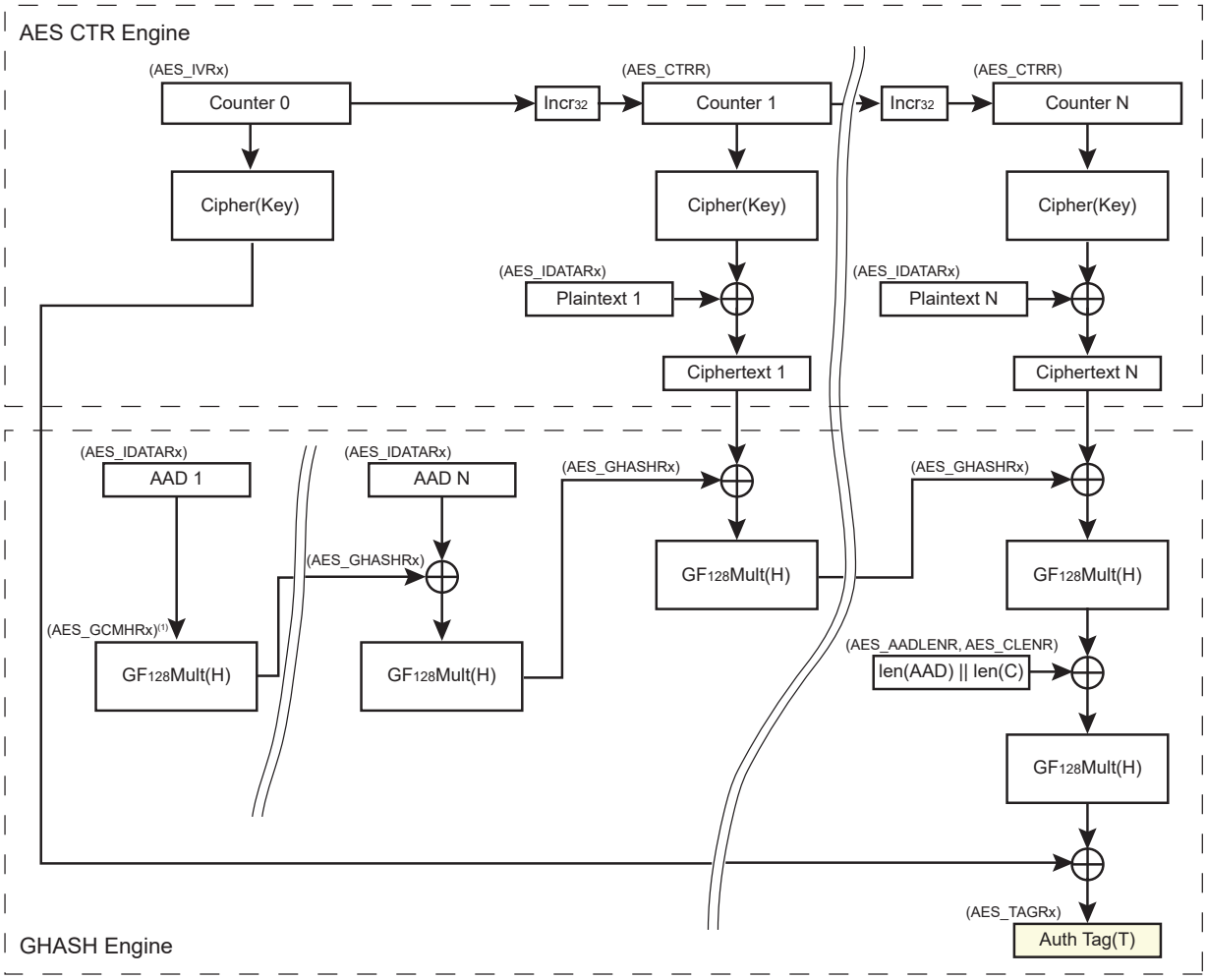
GCM can be used with or without the DMA master. Messages may be processed as a single complete packet of data or they may be broken into multiple packets of data over time.

GCM processing is computed on 128-bit input data fields. There is no support for unaligned data. The AES key length can be whatever length is supported by the AES module.

The recommended programming procedure when using DMAPDC is described in the section GCM Processing.

Downloaded from Arrow.com.

Figure 52-5. GCM Block Diagram



Note: 1. Optional

52.4.4.2 Key Writing and Automatic Hash Subkey Calculation

Whenever a new key is written to the hardware, two automatic actions are processed:

- GCM Hash Subkey *H* generation—The GCM hash subkey (*H*) is automatically generated. The GCM hash subkey generation must be complete before doing any other action. AES_ISR.DATRDY indicates when the subkey generation is complete (with interrupt if configured). The GCM hash subkey calculation is processed with the formula *H* = CIPHER(Key, <128 bits to zero>). The generated GCM *H* value is then available in AES_GCMHRx. If the application software requires a specific hash subkey, the automatically generated *H* value can be overwritten in AES_GCMHRx.
  AES_GCMHRx can be written after the end of the hash subkey generation (see AES_ISR.DATRDY) and prior to starting the input data feed.
- AES_GHASHRx Clear—AES_GHASHRx are automatically cleared. If a hash initial value is needed for the GHASH, it must be written to AES_GHASHRx
  - after writing AES_KEYWRx, if any
  - before starting the input data feed

52.4.4.3 GCM Processing

GCM processing is made up of three phases:

1. Processing the Additional Authenticated Data (*AAD*), hash computation only.
2. Processing the Ciphertext (C), hash computation + ciphering/deciphering.

Downloaded from Arrow.com.

3. Generating the Tag using length of *AAD*, length of C and J_0 (refer to NIST documentation for details).

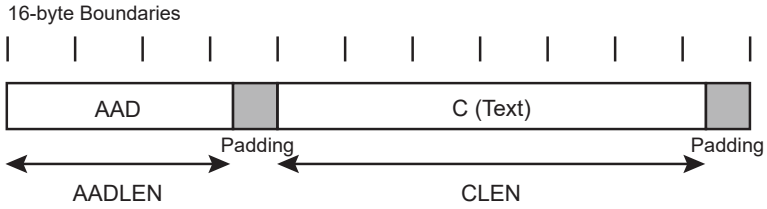
The Tag generation can be done either automatically, after the end of *AAD*/C processing if AES_MR.GTAGEN is set, or manually using AES_GHASHRx.GHASH (see subsections Processing a Complete Message with Tag Generation and Manual GCM Tag Generation for details).

52.4.4.3.1 Processing a Complete Message with Tag Generation

Use this procedure only if J0 four LSB bytes ≠ 0xFFFFFFFF.

Note: If J_0 four LSB bytes = 0xFFFFFFFF or if the value is unknown, use the procedure described in Processing a Complete Message without Tag Generation followed by the procedure in Manual GCM Tag Generation.

Figure 52-6. Full Message Alignment



To process a complete message with Tag generation, the sequence is as follows:

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '1'.
2. Write the key and wait until AES_ISR.DATRDY is set (GCM hash subkey generation complete); use interrupt if needed. See Key Writing and Automatic Hash Subkey Calculation.
3. Calculate the J_0 value as described in NIST documentation $J_0 = IV \| 0^{31} \| 1$ when len(IV) = 96 and $J_0 = GHASH_H(IV \| 0^{s+64} \| [len(IV)]64)$ if len(IV) ≠ 96. See Processing a Message with only AAD (GHASHH) for J_0 generation.
4. Set AES_IVRx.IV with inc32(J_0) (J_0 + 1 on 32 bits).
5. Configure AES_AADLENR.AADLEN and AES_CLENR.CLEN.
6. Fill AES_IDATARx.IDATA with the message to process according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).
7. Wait for TAGRDY to be set (use interrupt if needed), then read AES_TAGRx.TAG to obtain the authentication tag of the message.

52.4.4.3.2 Processing a Complete Message without Tag Generation

Processing a message without generating the Tag can be used to customize the Tag generation, or to process a fragmented message. To manually generate the GCM Tag, see Manual GCM Tag Generation.

To process a complete message without Tag generation, the sequence is as follows:

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '0'.
2. Write the key and wait until AES_ISR.DATRDY is set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in AES_GCMHRx. See Key Writing and Automatic Hash Subkey Calculation.
3. Calculate the J0 value as described in NIST documentation $J_0 = IV \| 0^{31} \| 1$ when len(IV) = 96 and $J_0 = GHASH_H(IV \| 0^{s+64} \| [len(IV)]64)$ if len(IV) ≠ 96. See Processing a Message with only AAD (GHASHH) for J_0 generation example when len(IV) ≠ 96.
4. Set AES_IVRx.IV with inc32(J_0) (J_0 + 1 on 32 bits).
5. Configure AES_AADLENR.AADLEN and AES_CLENR.CLEN.
6. Fill AES_IDATARx.IDATA with the message to process according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).
7. Make sure the last output data have been read if AES_CLENR.CLEN ≠ 0 (or wait for DATRDY), then read AES_GHASHRx.GHASH to obtain the hash value after the last processed data.

52.4.4.3.3 Processing a Fragmented Message without Tag Generation

If needed, a message can be processed by fragments, in such case automatic GCM Tag generation is not supported.

Downloaded from Arrow.com.

To process a message by fragments, the sequence is as follows:

- First fragment:

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '0'.
2. Write the key and wait for AES_ISR.DATRDY to be set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in AES_GCMHRx. See Key Writing and Automatic Hash Subkey Calculation.
3. Calculate the J_0 value as described in NIST documentation $J_0 = IV \,||\, 0^{31} \,||\, 1$ when len(IV) = 96 and $J_0 = GHASH_H(IV \,||\, 0^{s+64} \,||\, [len(IV)]64)$ if len(IV) ≠ 96. See Processing a Message with only AAD (GHASHH) for J_0 generation example when len(IV) ≠ 96.
4. Set AES_IVRx.IV with inc32(J_0) (J_0 + 1 on 32 bits).
5. Configure AES_AADLENR.AADLEN and AES_CLENR.CLEN according to the length of the first fragment, or set the fields with the full message length (both configurations work).
6. Fill AES_IDATARx.IDATA with the first fragment of the message to process (aligned on 16-byte boundary) according to the SMOD configuration used. If Manual Mode or Auto Mode is used the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).
7. Make sure the last output data have been read if the fragment ends in C phase (or wait for DATRDY if the fragment ends in *AAD* phase), then read AES_GHASHRx.GHASH to obtain the value of the hash after the last processed data and finally read AES_CTR.CTR to obtain the value of the CTR encryption counter (not needed when the fragment ends in AAD phase).

- Next fragment (or last fragment):

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '0'.
2. Write the key and wait until AES_ISR.DATRDY is set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in AES_GCMHRx. See Key Writing and Automatic Hash Subkey Calculation.
3. Set AES_IVRx.IV as follows:
   – If the first block of the fragment is a block of Additional Authenticated data, set AES_IVRx.IV with the J_0 initial value
   – If the first block of the fragment is a block of Plaintext data, set AES_IVRx.IV with a value constructed as follows: 'LSB96(J_0) || CTR' value, (96 bit LSB of J_0 concatenated with saved CTR value from previous fragment).
4. Configure AES_AADLENR.AADLEN and AES_CLENR.CLEN according to the length of the current fragment, or set the fields with the remaining message length, both configurations work.
5. Fill AES_GHASHRx.GHASH with the value stored after the previous fragment.
6. Fill AES_IDATARx.IDATA with the current fragment of the message to process (aligned on 16 byte boundary) according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).
7. Make sure the last output data have been read if the fragment ends in C phase (or wait for DATRDY if the fragment ends in AAD phase), then read AES_GHASHRx.GHASH to obtain the value of the hash after the last processed data and finally read AES_CTR.CTR to obtain the value of the CTR encryption counter (not needed when the fragment ends in *AAD* phase).

Note: Step 1 and 2 are required only if the value of the concerned registers has been modified.

Once the last fragment has been processed, the GHASH value will allow manual generation of the GCM tag. See Manual GCM Tag Generation.

52.4.4.3.4 Manual GCM Tag Generation

This section describes the last steps of the GCM Tag generation.

The Manual GCM Tag Generation is used to complete the GCM Tag Generation when the message has been processed without Tag Generation.

Note: The Message Processing without Tag Generation must be finished before processing the Manual GCM Tag Generation.

To generate a GCM Tag manually, the sequence is as follows:

Downloaded from Arrow.com.

Processing S = GHASH$_H$ (*AAD* || 0v || C || 0u || [len(*AAD*)]64 || [len(C)]64):

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '0'.
2. Write the key and wait for AES_ISR.DATRDY to be set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in AES_GCMHRx. See Key Writing and Automatic Hash Subkey Calculation.
3. Configure AES_AADLENR.AADLEN to 0x10 (16 bytes) and AES_CLENR.CLEN to '0'. This will allow running a single GHASH$_H$ on a 16-byte input data (see the following figure).
4. Fill AES_GHASHRx.GHASH with the state of the GHASH field stored at the end of the message processing.
5. Fill AES_IDATARx.IDATA according to the SMOD configuration used with 'len(*AAD*)64 || len(*C*)64' value as described in the NIST documentation and wait for DATRDY to be set; use interrupt if needed.
6. Read AES_GHASHRx.GHASH to obtain the current value of the hash.

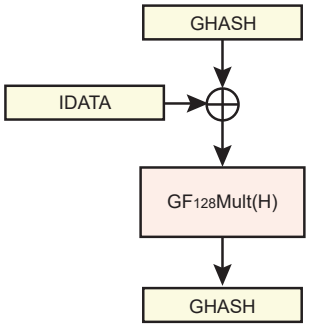
Processing *T* = GCTRK(J_0, S):

1. Set AES_MR.OPMOD to CTR.
2. Set AES_IVRx.IV with 'J_0' value.
3. Fill AES_IDATARx.IDATA with the GHASH value read at step 6 and wait for DATRDY to be set (use interrupt if needed).
4. Read AES_ODATARx.ODATA to obtain the GCM Tag value.

Note: Step 4 is optional if the GHASH field is to be filled with value '0' (0 length packet for instance).

52.4.4.3.5 Processing a Message with only AAD (GHASHH)

Figure 52-7. Single GHASH$_H$ Block Diagram (AADLEN ≤ 0x10 and CLEN = 0)



It is possible to process a message with only *AAD* setting the CLEN field to '0' in AES_CLENR, this can be used for J0 generation when len(*IV*) ≠ 96 for instance.

Example: Processing J_0 when len(*IV*) ≠ 96

To process J_0 = GHASH$_H$(*IV* || 0^{s+64} || [len(*IV*)]64), the sequence is as follows:

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '0'.
2. Write AES_KEYWRx and wait until AES_ISR.DATRDY is set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in AES_GCMHRx. See Key Writing and Automatic Hash Subkey Calculation.
3. Configure AES_AADLENR.AADLEN with 'len(*IV* || 0^{s+64} || [len(*IV*)]64)' in and AES_CLENR.CLEN to '0'. This will allow running a GHASH$_H$ only.
4. Fill AES_IDATARx.IDATA with the message to process (*IV* || 0^{s+64} || [len(*IV*)]64) according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when a GHASH$_H$ step is over (use interrupt if needed).
5. Read AES_GHASHRx.GHASH to obtain the J_0 value.
   Note: The GHASH value can be overwritten at any time by writing the value of AES_GHASHRx.GHASH, used to perform a GHASH$_H$ with an initial value for GHASH (write GHASH field between step 3 and step 4 in this case).

52.4.4.3.6 Processing a Single GF128 Multiplication

The AES can also be used to process a single multiplication in the Galois field on 128 bits (GF$_{128}$) using a single GHASH$_H$ with custom *H* value (see the figure above).

Downloaded from Arrow.com.

To run a GF_{128} multiplication (A x B), the sequence is as follows:

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '0'.

1. Configure AES_AADLENR.AADLEN with 0x10 (16 bytes) and AES_CLENR.CLEN to '0'. This will allow running a single $GHASH_H$.
2. Fill AES_GCMHRx.H with B value.
3. Fill AES_IDATARx.IDATA with the A value according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when a GHASHH computation is over (use interrupt if needed).
4. Read AES_GHASHRx.GHASH to obtain the result.

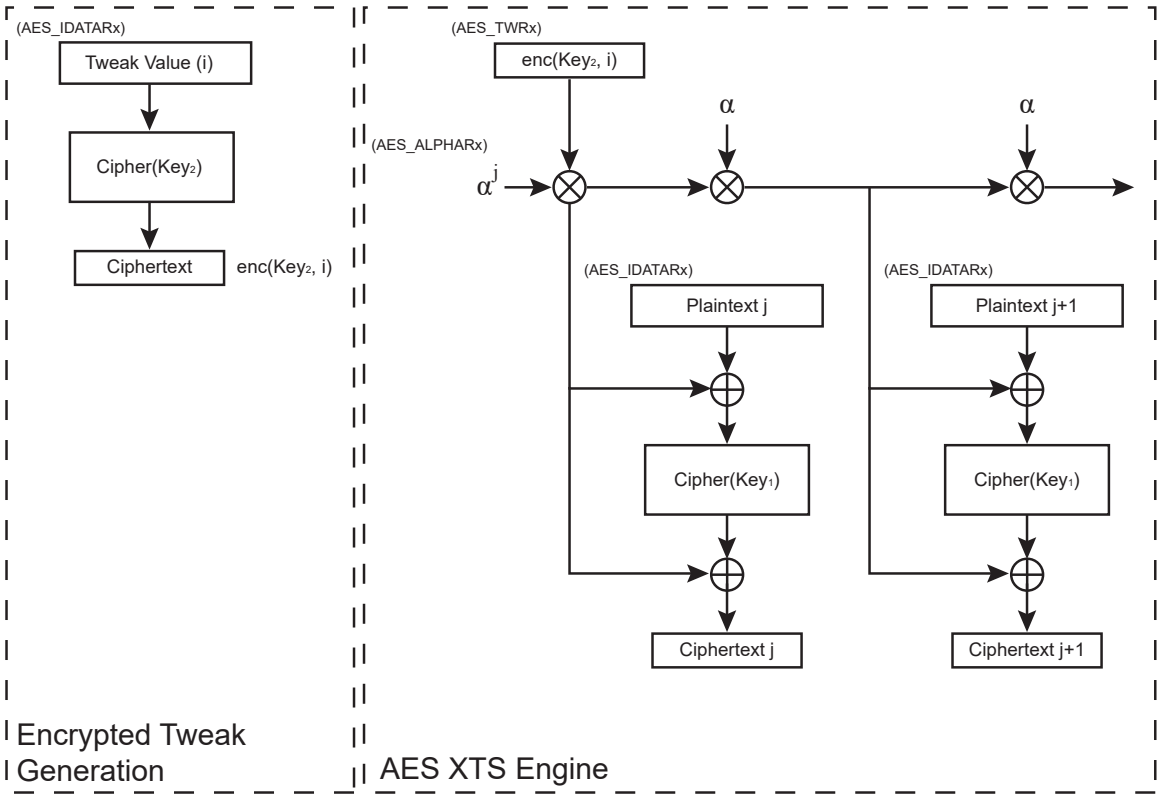
Note: AES_GHASHRx.GHASH can be initialized with a value C between step 3 and step 4 to run a ((A XOR C) x B) GF_{128} multiplication.

52.4.5 XEX-based Tweaked-codebook Mode (XTS)

XTS mode comprises the AES engine with XOR on inputs and outputs. After each encryption/decryption, the value used for the XOR is multiplied by the first GF(2^{128}) alpha primitive (0x2) and then used for the next encryption/decryption. The XTS mode uses two different keys and defines a Tweak Value (i) as additional input.

XTS processing is computed on 128-bit input data fields. There is no support for unaligned data (padding must be done manually if needed). The AES key length can be any length supported by the AES module.

Figure 52-8. XTS Block Diagram



52.4.5.1 XTS Processing Procedure

XTS processing comprises two phases:

1. Generate encrypted tweak with Key2 (this step is only required for the first processing, further consecutive processing does not require this step).
2. Process the data giving encrypted tweak and first alpha primitive for the first encryption/decryption.

52.4.5.1.1 Encrypted Tweak Generation

In the case of a new encryption/decryption, it is necessary to first encrypt the Tweak Value (i) with Key2. Here are the steps to follow to perform this step:

Downloaded from Arrow.com.

1. Set AES_MR.OPMODE to ECB and AES_MR.CIPHER to ‘1’.
2. Write the Key2.
3. Fill AES_IDATARx.IDATA with the Tweak value (i) according to the SMOD configuration used. If Manual mode or Auto mode is used, the DATRDY bit indicates when the data have been processed and can be read in AES_ODATARx.

52.4.5.1.2 Data Processing

To process data using XTS mode, follow the steps below:

1. Set AES_MR.OPMODE to XTS.
2. Write the Key1.
3. Only if the data to process is the first to be processed in the data unit, or if the data block to process is not consecutive to the previous processed data block in the same data unit, then two additional mandatory steps are required:
   a. AES_TWRx must be written with the encrypted Tweak Value (see Encrypted Tweak Generation for details) with bytes swapped as described in AES Register Endianness.
   b. Write AES_ALPHARx with the alpha primitive corresponding to the block number in the data unit.
4. Fill AES_IDATARx.IDATA with the data to process according to the SMOD configuration used. If Manual mode or Auto mode is used, the DATRDY bit indicates when the data have been processed and can be read in AES_ODATARx. Repeat Step 4 as long as consecutive data blocks are processed in the same data unit.

52.4.6 Double Input Buffer

AES_IDATARx can be double-buffered to reduce the runtime of large files.

This mode allows a new message block to be written when the previous message block is being processed. This is only possible when DMA accesses are performed (AES_MR.SMOD = 2).

AES_MR.DUALBUFF must be set to ‘1’ to access the double buffer.

52.4.7 Temporary Secured Storage for Keys

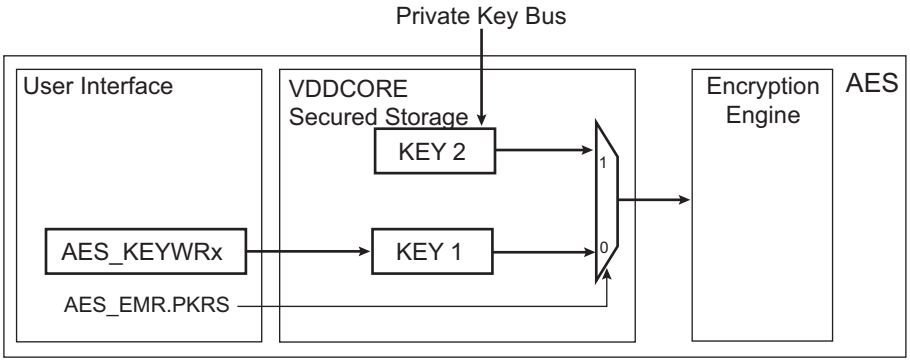
The AES provides secure storage for up to 256-bit keys. The storage is available while VDDCORE voltage is supplied.

The keys can be only written in AES internal registers and are not readable. Moreover, the internal registers holding the keys are buried in the overall product logic area during the physical implementation.

One key can be loaded by software by writing the Key Word registers (AES_KEYWRx).

One key can be loaded by private key bus only.

Figure 52-9. Temporary Secured Storage for Keys



52.4.8 Start Modes

AES_MR.SMOD allows selection of the encryption (or decryption) Start mode.

52.4.8.1 Manual Mode

The sequence of actions is as follows:

Downloaded from Arrow.com.

1. Write AES_MR with all required fields, including but not limited to SMOD and OPMOD. (Write AES_EMR.PKRS according to the type of key to be loaded).
2. Write the 128-bit/192-bit/256-bit AES key in AES_KEYWRx or in the private key internal register.
3. Write the initialization vector (or counter) in AES_IVRx.
   Note: AES_IVRx concerns all modes except ECB.
4. Set the bit DATRDY (Data Ready) in the AES Interrupt Enable register (AES_IER), depending on whether an interrupt is required or not at the end of processing.
5. Write the data to be encrypted/decrypted in the authorized AES_IDATARx (see the following table).
6. Set the START bit in the AES Control register (AES_CR) to begin the encryption or the decryption process.
7. When processing completes, the DATRDY flag in the AES Interrupt Status register (AES_ISR) is raised. If an interrupt has been enabled by setting AES_IER.DATRDY, the interrupt line of the AES is activated.
8. When software reads one of AES_ODATARx, AES_IER.DATRDY is automatically cleared.

Table 52-2. Authorized Input Data Registers

Operating Mode	Input Data Registers to Write
ECB	All
CBC	All
OFB	All
128-bit CFB	All
64-bit CFB	AES_IDATAR0 and AES_IDATAR1
32-bit CFB	AES_IDATAR0
16-bit CFB	AES_IDATAR0
8-bit CFB	AES_IDATAR0
CTR	All
GCM	All
XTS	All

Note:

1. In 64-bit CFB mode, writing to AES_IDATAR2 and AES_IDATAR3 is not allowed and may lead to errors in processing.
2. In 32, 16, and 8-bit CFB modes, writing to AES_IDATAR1, AES_IDATAR2 and AES_IDATAR3 is not allowed and may lead to errors in processing.

52.4.8.2 Auto Mode

The Auto Mode is similar to the manual one, except that in this mode, as soon as the correct number of AES_IDATARx is written, processing is automatically started without any action in AES_CR.

52.4.8.3 DMA Mode

The DMA Controller can be used in association with the AES to perform an encryption/decryption of a buffer without any action by software during processing.

AES_MR.SMOD must be configured to 2 and the DMA must be configured with non-incremental addresses.

For all operating modes except CBC-MAC (AES_MR.LOD=1), 2 DMA channels must be programmed (transmit and receive). In CBC-MAC, only 1 transmit channel must be programmed.

The start address of any transfer descriptor must be configured with the address of AES_IDATAR0.

The DMA chunk size configuration depends on the AES mode of operation and is summarized in the following table.

When writing data to AES with a first DMA channel, data are first fetched from a memory buffer (source data). It is recommended to configure the size of source data to “words” even for CFB modes. On the contrary, the destination data size depends on the mode of operation. When reading data from the AES with the second DMA channel, the

Downloaded from Arrow.com.

source data is the data read from AES and data destination is the memory buffer. In this case, the source data size depends on the AES mode of operation, as shown in the following table.

Table 52-3. DMA Data Transfer Type for the Different Operating Modes

Operating Mode	Chunk Size	Destination/Source Data Transfer Type
ECB	4	Word
CBC	4	Word
OFB	4	Word
CFB 128-bit	4	Word
CFB 64-bit	1	Word
CFB 32-bit	1	Word
CFB 16-bit	1	Half-word
CFB 8-bit	1	Byte
CTR	4	Word
GCM	4	Word
XTS	4	Word

52.4.9 Automatic Padding Mode

When Automatic Padding mode is configured, the message is automatically padded after the last block is written. Depending on the size of the message, either a padding is performed after the last part of the message and padding blocks are added, or only padding blocks are added.

IPSEC and SSL padding standards are both supported.

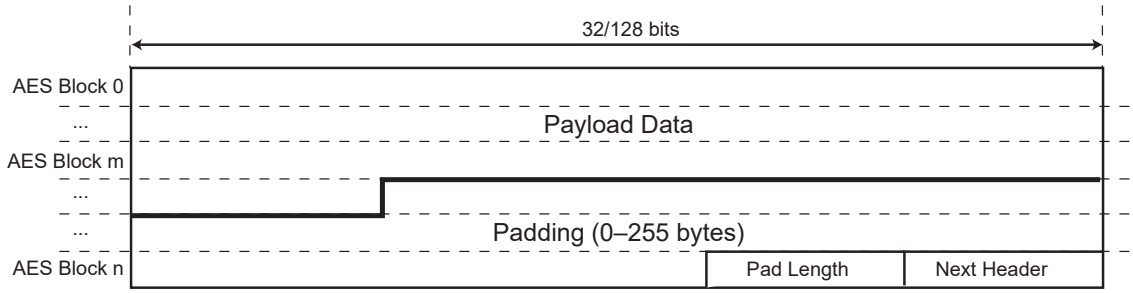
The auto padding feature only supports CBC and CTR modes.

Note: When automatic padding is enabled and AES_MR.SMOD=2, AES_MR.DUALBUFF must be cleared.

52.4.9.1 IPSEC Padding

Automatic Padding is enabled by writing a ‘1’ to AES_EMR.APEN. IPSEC padding mode is selected by writing a ‘0’ to AES_EMR.APM.

Figure 52-10. IPSEC Padding



Each byte of the padding area contains incremental integer values.

The “Pad Length” in bytes is configured in AES_EMR.PADLEN and the “Next Header” value is configured in AES_EMR.NHEAD. AES_EMR.PADLEN must be configured with the length of the padding section, not including the length of the “Pad Length” and “Next Header” sections.

The BCNT field in the AES Byte Counter register (AES_BCNT) defines the length, in bytes, of the message to process. It must be configured before writing the first data in AES_IDATARx and the remaining bytes to process can be read at anytime (BCNT value is decremented after each AES_IDATARx access).

Downloaded from Arrow.com.

AES_BCNT.BCNT and AES_EMR.PADLEN must be configured so that the sum of the length of the message (Payload Data) and of the length of the Padding, Pad Length (1 byte) and Next Header (1 byte) sections is a multiple of the AES block size (128 bits).

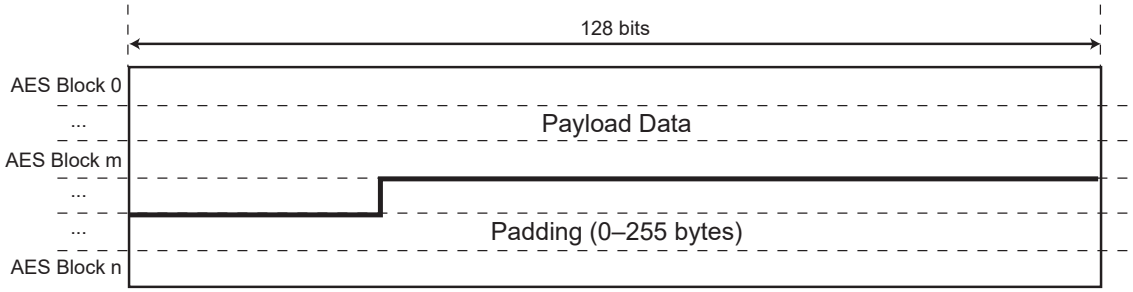
To process an IPSEC message using auto-padding, the sequence is as follows:

1. Set AES_MR.OPMOD to either CBC or CTR mode.
2. Set AES_EMR.APEN to ‘1’, AES_EMR.APM to ‘0’, AES_EMR.PADLEN to the desired padding length in byte and AES_EMR.NHEAD to the desired Next Header field value.
3. Configure AES_BCNT.BCNT with the whole message length, without padding, in byte.
4. Write the key.
5. Set AES_IVRx.IV if needed.
6. Fill AES_IDATARx.IDATA with the message to process according to the SMOD configuration used. On the last data block, write only what is necessary (e.g., write only AES_IDATAR0 if last block size is ≤ 32 bits).
7. Wait for the DATRDY flag to be raised, meaning auto-padding completion and last block processing.

52.4.9.2 SSL Padding

Auto Padding is enabled by writing a ‘1’ to AES_EMR.APEN and SSL padding mode is selected by writing a ‘1’ to AES_EMR.APM.

Figure 52-11. SSL Padding



Each byte of the padding area contains the padding length.

The padding length is configured in AES_EMR.PADLEN.

AES_BCNT.BCNT defines the length, in bytes, of the message to process. It must be configured before writing the first data in AES_IDATARx and the remaining bytes to process can be read at anytime (BCNT value is decremented after each AES_IDATARx access).

AES_BCNT.BCNT and AES_EMR.PADLEN must be configured so that the length of the message plus the length of the padding section is a multiple of the AES block size (128 bits).

To process a complete SSL message, the sequence is as follows:

1. Set AES_MR.OPMOD to either CBC or CTR mode.
2. Set AES_EMR.APEN to ‘1’, AES_EMR.APM to ‘1’, AES_EMR.PADLEN to the desired padding length in bytes.
3. Set AES_BCNT.BCNT with the whole message length, without padding, in bytes.
4. Write the key.
5. Set AES_IVRx.IV if needed.
6. Fill AES_IDATARx.IDATA with the message to process according to the SMOD configuration used. On the last data block write only what is necessary (e.g., write only AES_IDATAR0 if last block size is ≤ 32 bits).
7. Wait for the DATRDY flag to be raised, meaning auto-padding completion and last block processing.

52.4.9.3 Flags

AES_ISR.EOPAD rises as soon as the automatic padding phase is over, meaning that all the extra padding blocks have been processed. Reading AES_ISR clears this flag.

AES_ISR.PLENERR indicates an error in the frame configuration, meaning that the whole message length including padding does not respect the standard selected. AES_ISR.PLENERR rises at the end of the frame in case of wrong message length and is cleared reading AES_ ISR.

Downloaded from Arrow.com.

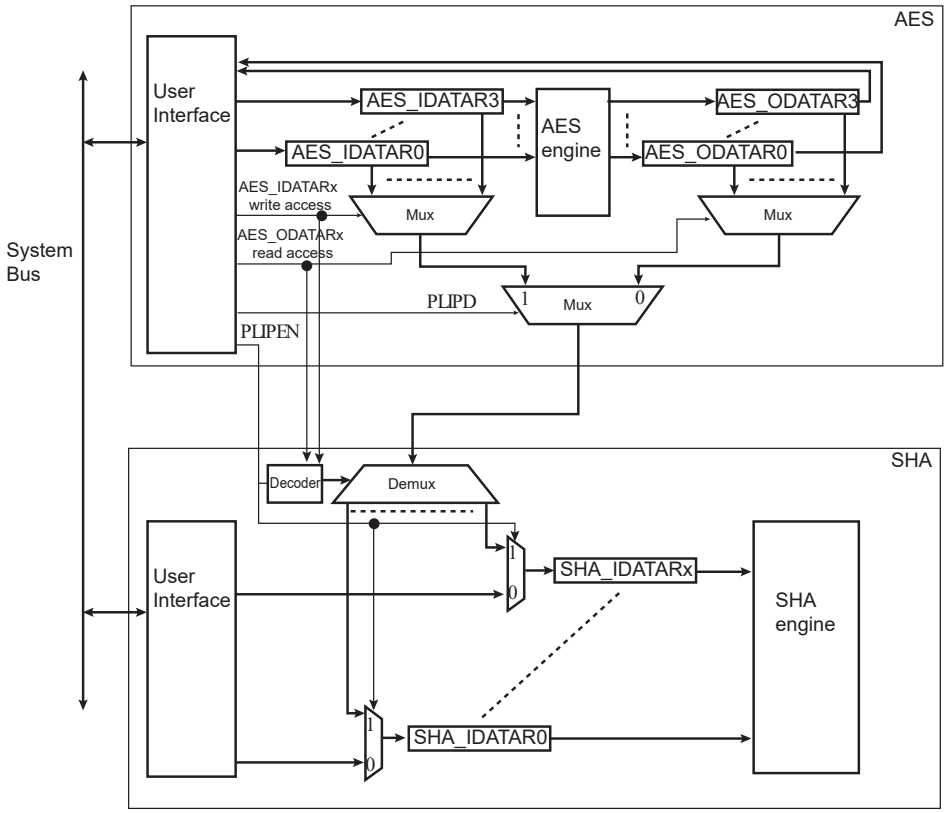
In IPSEC/SSL standard message length including padding must be a multiple of the AES block size when CBC mode is used and multiple of 32-bit if CTR mode is used.

52.4.10 Secure Protocol Layers Improved Performances

Secure protocol layers such as IPSec require encryption and authentication. For IPSec, the authentication is based on HMAC, thus SHA is required. To optimize performance, the AES embeds a mode of operation that enables the SHA module to process the input or output data of the AES module. If this mode is enabled, write access is required only into AES_IDATARx registers, since SHA input data registers are automatically written by AES without software intervention. When the DMA is configured to transfer a buffer of data (input frame), only one transfer descriptor is required for both authentication and encryption/decryption processes and only one buffer is transferred through the system bus (reducing the load of the system bus).

Improved performance for secure protocol layers requires AES_EMR.PLIPEN to be set.

Figure 52-12. Secure Protocol Layers Improved Performances Block Diagram



52.4.10.1 Cipher Mode

When AES_EMR.PLIPD is cleared and AES_EMR.PLIPEN=1, the message written into AES_IDATARx is first encrypted with the AES module and the encrypted message is authenticated with the SHA module. Therefore, when AES_EMR.PLIPD is cleared, AES_ODATARx are selected and sent to SHA_IDATARx as soon as AES_ODATARx are read. A read access in AES corresponds to a write access to the corresponding SHA_IDATARx. The number of SHA_IDATARx is greater than the number of AES_ODATARx, but the SHA module embeds the decoding logic to automatically dispatch AES_ODATARx values into the corresponding SHA_IDATARx without software intervention.

52.4.10.2 Decipher Mode

When AES_EMR.PLIPD is written to ‘1’ and AES_EMR.PLIPEN=1, the message written into AES_IDATARx is decrypted with the AES module and also sent to SHA for authentication. Therefore, when AES_EMR.PLIPD=1, AES_IDATARx are selected and sent to SHA_IDATARx as soon as AES_IDATARx are written. A write access in AES corresponds to a write access to the corresponding SHA_IDATARx. The number of SHA_IDATARx is greater than the number of AES_ODATARx, but the SHA module embeds the decoding logic to automatically dispatch AES_IDATARx values into the corresponding SHA_IDATARx without software intervention.

Downloaded from Arrow.com.

52.4.10.3 Encapsulating Security Payload (ESP) IPSec Examples

The following examples describe how to configure AES and SHA to optimize processing an ESP IPSec frame for maximum performance.

The cipher (or decipher) of an ESP IPSec frame requires both encryption (or decryption) and authentication.

For cipher, the input frame located in the system memory must first be padded and the resulting buffer encrypted. The encrypted frame must be written back to the system memory and sent to the authentication module.

When the AES module is configured to improve the performance of the secure protocol layers (AES_EMR.PLIPEN = 1), the data transfers are simplified, limiting the bandwidth requirements on the system bus.

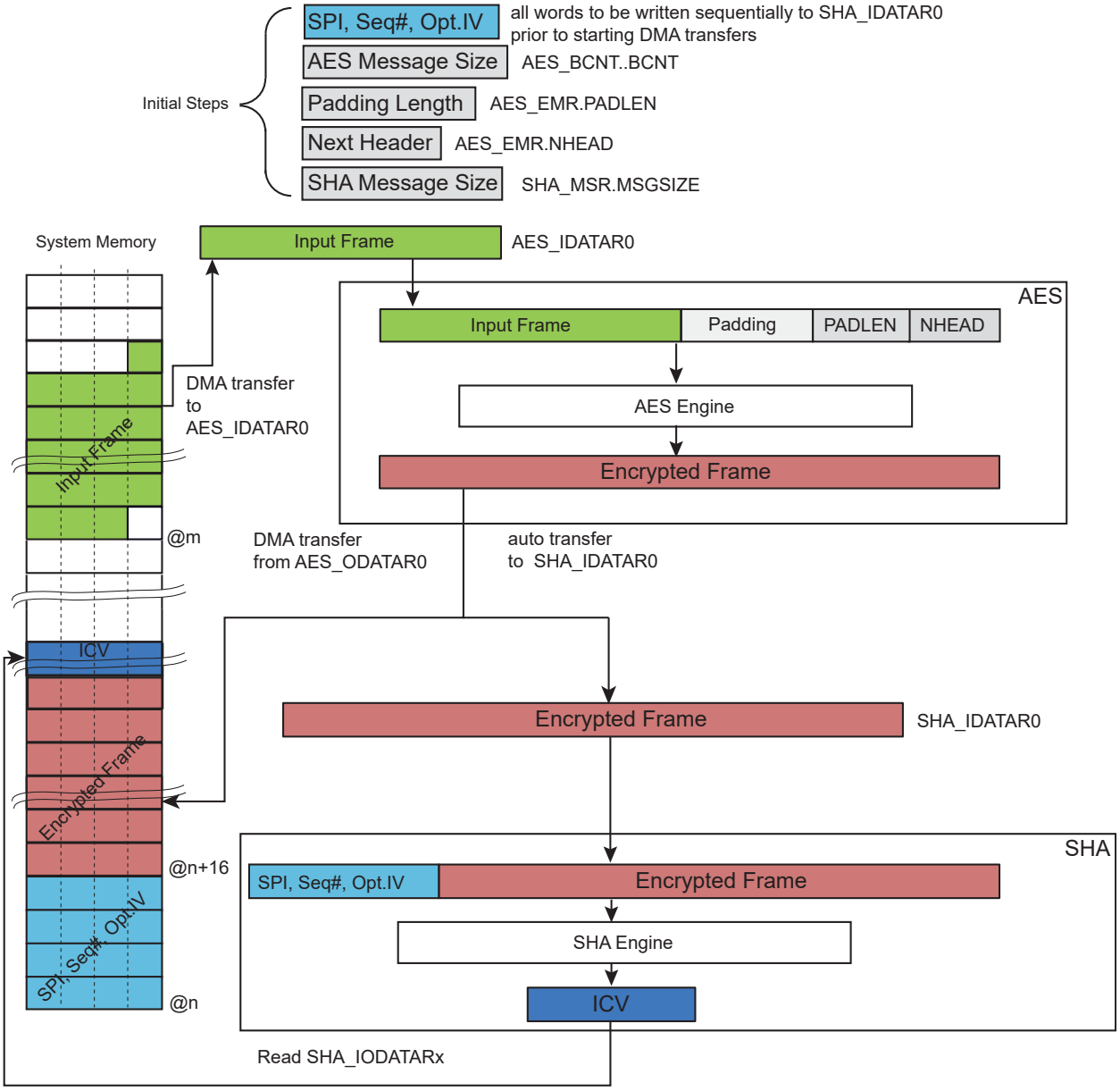
Before configuring the DMA to start the transfer of the data buffer (input frame) to the AES, the following actions must be taken in registers:

- AES_BCNT.BCNT must be configured with the length of the message (Input Frame).
- The padding length of the AES must be configured in AES_EMR.PADLEN. See Automatic Padding Mode to configure Automatic Padding mode.
- The next header value must be configured in AES_EMR.NHEAD.
- AES_MR.SMOD and SHA_MR.SMOD must be configured to 2.
  Note: When automatic padding is enabled and AES_MR.SMOD = 2 , AES_MR.DUALBUFF must be cleared.
- The SHA_MSR.MSGSIZE must be configured with the length of the authentication message including the optional extended sequence number (ESN) and header and trailer information required by the authentication algorithm used (HMAC, etc.). Refer to the section “Secure Hash Algorithm (SHA)” for more details on configuration for optimized processing of header information.
- The Security Parameter Index (SPI, sequence number (SEQ#)) and the optional Initialization Vector (IV) must be configured sequentially in SHA_IDATAR0.
- A first DMA transfer descriptor must be configured to transfer the input frame from the system memory to the AES input data registers (AES_IDATARx), and a second DMA descriptor must be configured to transfer the encrypted frame from AES to the system memory.
  Note: If AES_EMR.PLIPEN = 1 , there is no need to define a transfer descriptor to load the encrypted frame into the SHA input data registers because the transfer is automatically performed while the second descriptor transfer is in progress.

See the following figures.

Downloaded from Arrow.com.

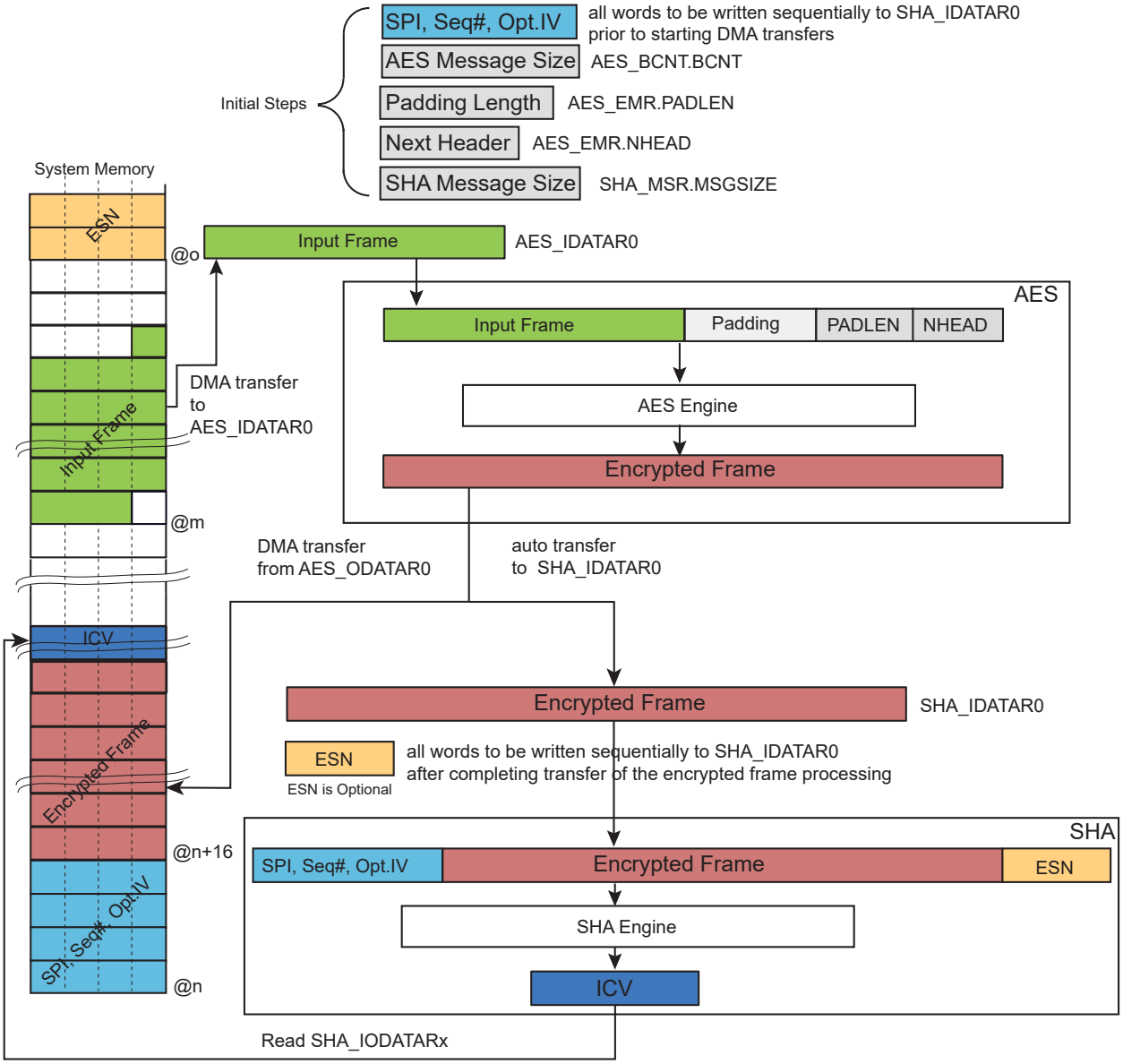
Figure 52-13. Generation of an ESP IPSec Frame without ESN



If the optional extended sequence number is required for authentication, wait for the AES-to-system memory DMA buffer transfer to complete before configuring the ESN value. The ESN value must be configured in the SHA by writing sequentially each 32-bit word of the ESN into the SHA_IDATAR0 register. Wait for SHA_ISR.WRDY=1 before each write in the SHA_IDATAR0 register. See the following figure.

Downloaded from Arrow.com.

Figure 52-14. Generation of an ESP IPSec Frame with ESN



To decipher an ESP IPSec frame without the optional ESN trailer information, two DMA channels are required and the SHA must be configured in Automatic padding mode.

Note: AES automatic padding must be disabled when deciphering a frame.

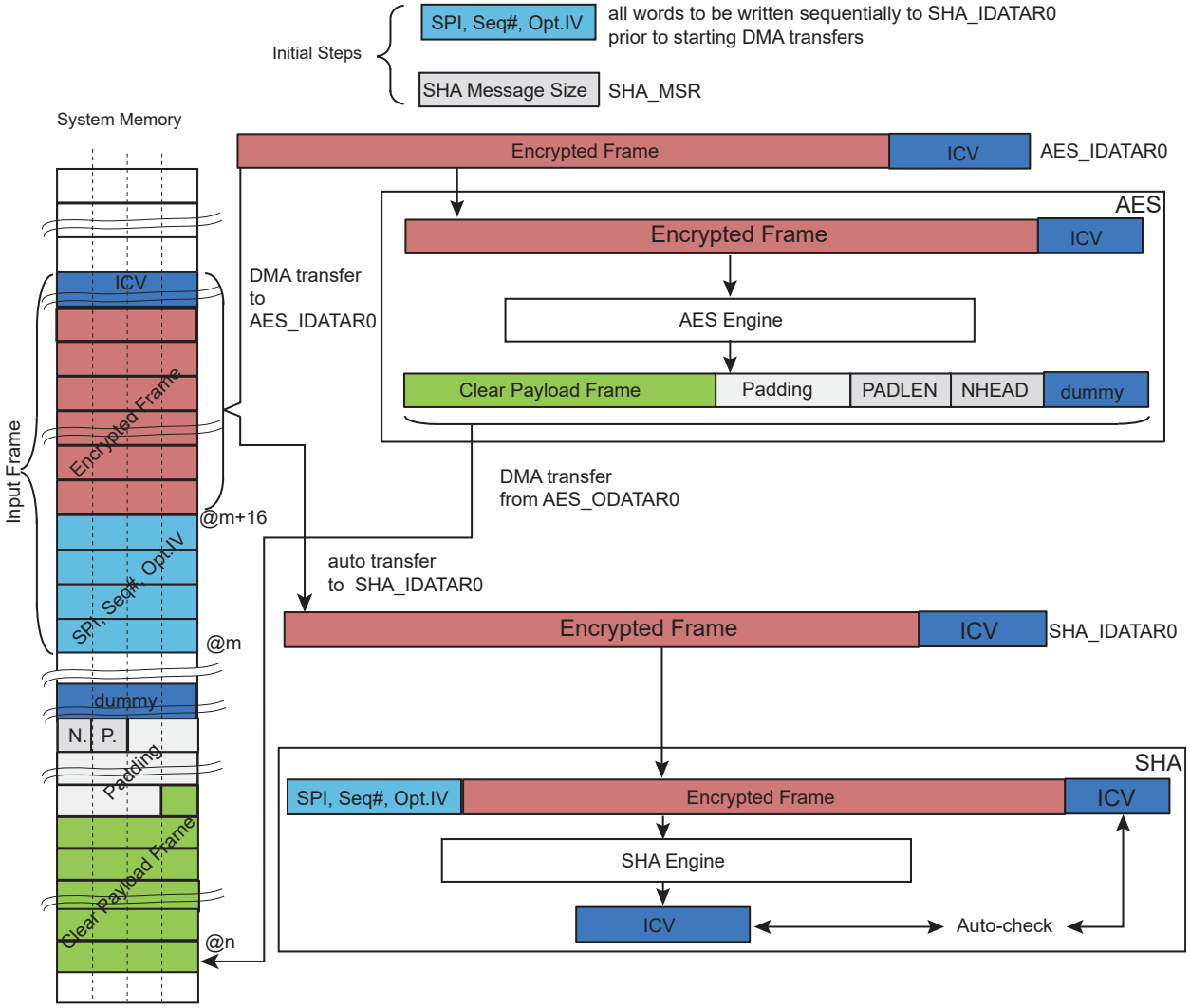
- A first DMA transfer descriptor must be configured to load the received encrypted frame from the system memory to AES_IDATARx for decryption. The start address of the first transfer descriptor must be defined after the SPI, SEQ#, and optional IV (see the following figure).
- A second DMA descriptor must be configured to transfer the decrypted frame from AES_ODATARx to the system memory.
- AES_EMR.PLIPEN and AES_EMR.PLIPD must be written to '1' so that the data buffer is written in AES_IDATARx and in SHA_IDATARx.

The SHA has the capability to perform an automatic check with an expected integrity check value if this value is appended at the end of the frame buffer (SHA_MR.CHECK=2). Thus, if the first transfer descriptor includes the ICV for SHA, the first DMA transfer allows the decryption and authentication processes including the automatic check. The decrypted part resulting from ICV is not required for downstream processing and must be considered as dummy data.

Downloaded from Arrow.com.

The end of the decryption and authentication processes occur when flag SHA_ISR.CHECKF=1. The authentication status is provided by SHA_ISR.CHKST.

Figure 52-15. Decryption of an ESP IP Sec Frame without ESN

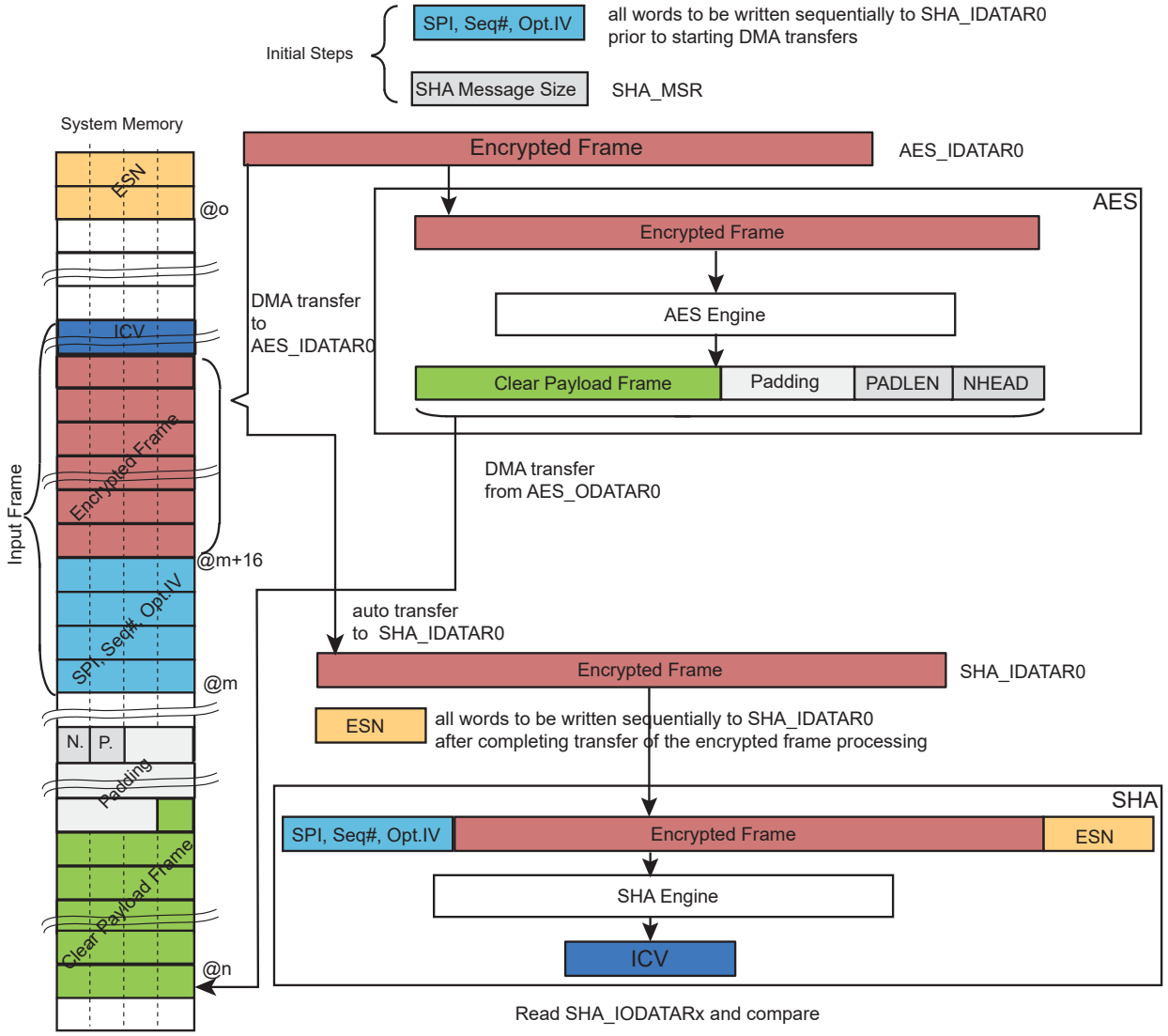


If the optional ESN trailer information is part of the ICV (see the following figure), the ESN must be manually written into SHA_IDATAR0. The ESN value must be written after completion of the system memory-to-AES DMA buffer transfer. The ESN value must be configured in the SHA by writing sequentially each 32-bit word of the ESN into the SHA_IDATAR0 register. Wait for SHA_ISR.WRDY=1 before each write in the SHA_IDATAR0 register.

When the optional ESN trailer information is part of the ICV, it is not possible to include the ICV received in the input frame to the first transfer descriptor. Moreover, if the HMAC algorithm is used for authentication, no automatic check can be performed when optimizing the processing performances of the SHA module. For more details, refer to the section “Secure Hash Algorithm (SHA)”. The result of the HMAC read in the SHA_IODATARx must be manually compared with the ICV value of the input frame. The comparison must be performed after the end of the authentication process. The authentication process is completed when the SHA_ISR.DATRDY flag is set.

Downloaded from Arrow.com.

Figure 52-16. Decryption of an ESP IPSec Frame with ESN



52.4.11 Security Features

52.4.11.1 Private Key Bus

The AES provides secure key transfer that requires a transfer command only, thus avoiding any manipulation of the key by software.

The AES features a set of private key internal registers that can be accessed only through the dedicated private key bus from the TRNG.

The private key internal registers cannot be read from any peripheral or from software.

The AES key used by the encryption/decryption engine is either the private key internal registers content or the AES_KEYWRx registers loaded via the AES_KEYWRx.

To select the private key internal registers as the source of the AES key, AES_EMR.PKRS must be written to ‘1’.

When AES_EMR.PKRS is modified, it is mandatory to load the corresponding key value even if the key has been previously written with same value.

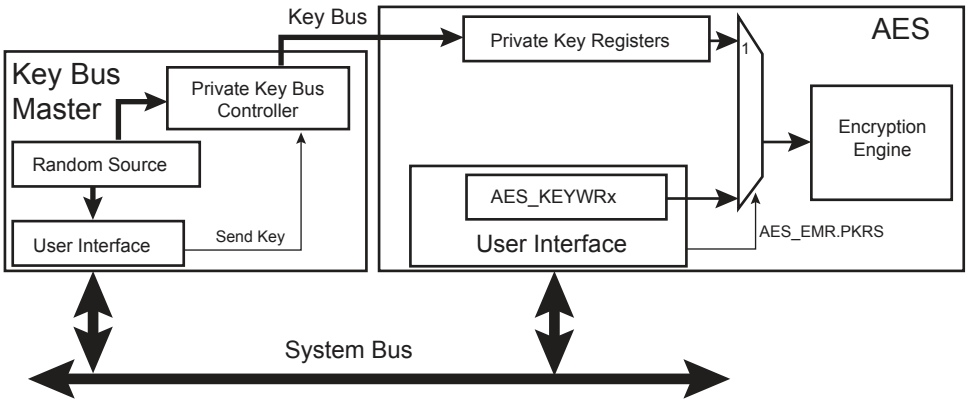
To write the private key internal registers, the software must:

1. Write a ‘1’ in AES_EMR.PKRS.

Downloaded from Arrow.com.

2. Trigger the key transfer over the private key bus from the TRNG key bus master.
3. Wait for completion of the transfer signaled in the TRNG status register.
4. Check for any access violation in AES_WPSR.PKRPVS.

Figure 52-17. Key Selection



52.4.11.2 Unspecified Register Access Detection

When an unspecified register access occurs, AES_ISR.URAD is raised. Its source is then reported in AES_ISR.URAT. Only the last unspecified register access is available through the AES_ISR.URAT.

Several kinds of unspecified register accesses can occur:

- Input Data register written during the data processing when SMOD = IDATAR0_START
- Output Data register read during data processing
- Mode register written during data processing
- Output Data register read during sub-keys generation
- Mode register written during sub-keys generation
- Write-only register read access

AES_ISR.URAD and AES_ISR.URAT can only be reset by AES_CR.SWRST.

52.4.11.3 Clearing Key on Tamper Event

On a tamper detection event on WKUP1..8 pins, an immediate clear of the key (internal registers) can be performed if AES_MR.TAMPCLR=1. For configuration details, refer to the section “Real-Time Clock (RTC)”.

52.4.11.4 Register Write Protection

To prevent any single software error from corrupting AES behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the AES Write Protection Mode Register (AES_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the AES Write Protection Status Register (AES_WPSR) is set and the Write Protection Violation Source (WPVSRC) field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading AES_WPSR.

The following register(s) can be write-protected when WPEN is set in AES_WPMR:

- AES Mode Register
- AES Key Word Register x
- AES Initialization Vector Register x
- AES Additional Authenticated Data Length Register
- AES Plaintext/Ciphertext Length Register
- AES GCM Intermediate Hash Word Register x
- AES GCM H Word Register x
- AES Extended Mode Register

Downloaded from Arrow.com.

- AES Byte Counter Register
- AES Tweak Word Register x
- AES Alpha Word Register x

The following register(s) can be write-protected when WPITEN is set:

- AES Interrupt Enable Register
- AES Interrupt Disable Register

The following register(s) can be write-protected when WPCREN is set:

- AES Control Register

52.4.11.5 Security and Safety Analysis and Reports

Several types of checks are performed when the AES is enabled.

The peripheral clock of the AES is monitored by specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the AES. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the flag AES_WPSR.CGD is set, an abnormal condition occurred on the peripheral clock. This flag is not set under normal operating conditions.

The internal sequencer of the AES is also monitored and if an abnormal state is detected, the flag AES_WPSR.SEQE is set. This flag is not set under normal operating conditions.

The software accesses to the AES are monitored and if an incorrect access is performed, the flag AES_WPSR.SWE is set. The type of incorrect/abnormal software access is reported in AES_WPSR.SWETYP (see AES_WPSR for details). e.g., writing the AES_ODATARx is an error, as well as reading the AES_IDATARx, when the AES_ISR.DATRDY flag is cleared. AES_WPSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The flags CGD, SEQE, SWE and WPVS are automatically cleared when AES_WPSR is read.

If one of these flags is set, the flag AES_ISR.SECE is set and can trigger an interrupt if the AES_IMR.SECE bit is ‘1’. SECE is cleared by reading AES_ISR.

It is possible to configure an action to be performed by AES as soon as an abnormal event detection occurs. If AES_WPMR.ACTION > 0, either a lock is performed or a lock and immediate clear of the AES_KEYWRx key. If a lock is performed, the current processing is ended normally but any new processing is not performed whatever the start mode of operation (see AES_MR.SMOD).

A locked state of the AES is unlocked as follows:

1. Read AES_WPSR.
2. Disable the source of tamper if the tamper is enabled to perform a clear of the key.
3. Write a ‘1’ to AES_CR.UNLOCK.

It is possible to select the type of event that will lock the AES in case of abnormal event detection. See AES_WPMR.ACTION for details.

If the AES_MR.TMPCLR=1 and the tamper pin is active, the AES is locked.

Downloaded from Arrow.com.

52.5 Register Summary

Offset	Name	Bit Pos.								
0x00	AES_CR	31:24								UNLOCK
		23:16								
		15:8								SWRST
		7:0								
0x04	AES_MR	31:24	TAMPCLR							
		23:16	CKEY[3:0]					CFBS[2:0]		
		15:8	LOD	OPMOD[2:0]			KEYSIZE[1:0]		SMOD[1:0]	
		7:0	PROCDLY[3:0]				DUALBUFF		GTAGEN	CIPHER
0x08 ... 0x0F	Reserved									
0x10	AES_IER	31:24								
		23:16					SECE	PLENERR	EOPAD	TAGRDY
		15:8								URAD
		7:0								DATRDY
0x14	AES_IDR	31:24								
		23:16					SECE	PLENERR	EOPAD	TAGRDY
		15:8								URAD
		7:0								DATRDY
0x18	AES_IMR	31:24								
		23:16					SECE	PLENERR	EOPAD	TAGRDY
		15:8								URAD
		7:0								DATRDY
0x1C	AES_ISR	31:24								
		23:16					SECE	PLENERR	EOPAD	TAGRDY
		15:8	URAT[3:0]							URAD
		7:0								DATRDY
0x20	AES_KEYWR0	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x24	AES_KEYWR1	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x28	AES_KEYWR2	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x2C	AES_KEYWR3	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x30	AES_KEYWR4	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x34	AES_KEYWR5	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x38	AES_KEYWR6	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.	
0x3C	AES_KEYWR7	31:24	KEYW[31:24]
		23:16	KEYW[23:16]
		15:8	KEYW[15:8]
		7:0	KEYW[7:0]
0x40	AES_IDATAR0	31:24	IDATA[31:24]
		23:16	IDATA[23:16]
		15:8	IDATA[15:8]
		7:0	IDATA[7:0]
0x44	AES_IDATAR1	31:24	IDATA[31:24]
		23:16	IDATA[23:16]
		15:8	IDATA[15:8]
		7:0	IDATA[7:0]
0x48	AES_IDATAR2	31:24	IDATA[31:24]
		23:16	IDATA[23:16]
		15:8	IDATA[15:8]
		7:0	IDATA[7:0]
0x4C	AES_IDATAR3	31:24	IDATA[31:24]
		23:16	IDATA[23:16]
		15:8	IDATA[15:8]
		7:0	IDATA[7:0]
0x50	AES_ODATAR0	31:24	ODATA[31:24]
		23:16	ODATA[23:16]
		15:8	ODATA[15:8]
		7:0	ODATA[7:0]
0x54	AES_ODATAR1	31:24	ODATA[31:24]
		23:16	ODATA[23:16]
		15:8	ODATA[15:8]
		7:0	ODATA[7:0]
0x58	AES_ODATAR2	31:24	ODATA[31:24]
		23:16	ODATA[23:16]
		15:8	ODATA[15:8]
		7:0	ODATA[7:0]
0x5C	AES_ODATAR3	31:24	ODATA[31:24]
		23:16	ODATA[23:16]
		15:8	ODATA[15:8]
		7:0	ODATA[7:0]
0x60	AES_IVR0	31:24	IV[31:24]
		23:16	IV[23:16]
		15:8	IV[15:8]
		7:0	IV[7:0]
0x64	AES_IVR1	31:24	IV[31:24]
		23:16	IV[23:16]
		15:8	IV[15:8]
		7:0	IV[7:0]
0x68	AES_IVR2	31:24	IV[31:24]
		23:16	IV[23:16]
		15:8	IV[15:8]
		7:0	IV[7:0]
0x6C	AES_IVR3	31:24	IV[31:24]
		23:16	IV[23:16]
		15:8	IV[15:8]
		7:0	IV[7:0]
0x70	AES_AADLENR	31:24	AADLEN[31:24]
		23:16	AADLEN[23:16]
		15:8	AADLEN[15:8]
		7:0	AADLEN[7:0]

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x74	AES_CLENR	31:24	CLEN[31:24]							
		23:16	CLEN[23:16]							
		15:8	CLEN[15:8]							
		7:0	CLEN[7:0]							
0x78	AES_GHASHR0	31:24	GHASH[31:24]							
		23:16	GHASH[23:16]							
		15:8	GHASH[15:8]							
		7:0	GHASH[7:0]							
0x7C	AES_GHASHR1	31:24	GHASH[31:24]							
		23:16	GHASH[23:16]							
		15:8	GHASH[15:8]							
		7:0	GHASH[7:0]							
0x80	AES_GHASHR2	31:24	GHASH[31:24]							
		23:16	GHASH[23:16]							
		15:8	GHASH[15:8]							
		7:0	GHASH[7:0]							
0x84	AES_GHASHR3	31:24	GHASH[31:24]							
		23:16	GHASH[23:16]							
		15:8	GHASH[15:8]							
		7:0	GHASH[7:0]							
0x88	AES_TAGR0	31:24	TAG[31:24]							
		23:16	TAG[23:16]							
		15:8	TAG[15:8]							
		7:0	TAG[7:0]							
0x8C	AES_TAGR1	31:24	TAG[31:24]							
		23:16	TAG[23:16]							
		15:8	TAG[15:8]							
		7:0	TAG[7:0]							
0x90	AES_TAGR2	31:24	TAG[31:24]							
		23:16	TAG[23:16]							
		15:8	TAG[15:8]							
		7:0	TAG[7:0]							
0x94	AES_TAGR3	31:24	TAG[31:24]							
		23:16	TAG[23:16]							
		15:8	TAG[15:8]							
		7:0	TAG[7:0]							
0x98	AES_CTRR	31:24	CTR[31:24]							
		23:16	CTR[23:16]							
		15:8	CTR[15:8]							
		7:0	CTR[7:0]							
0x9C	AES_GCMHR0	31:24	H[31:24]							
		23:16	H[23:16]							
		15:8	H[15:8]							
		7:0	H[7:0]							
0xA0	AES_GCMHR1	31:24	H[31:24]							
		23:16	H[23:16]							
		15:8	H[15:8]							
		7:0	H[7:0]							
0xA4	AES_GCMHR2	31:24	H[31:24]							
		23:16	H[23:16]							
		15:8	H[15:8]							
		7:0	H[7:0]							
0xA8	AES_GCMHR3	31:24	H[31:24]							
		23:16	H[23:16]							
		15:8	H[15:8]							
		7:0	H[7:0]							
0xAC ... 0xAF	Reserved									

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0xB0	AES_EMR	31:24	BPE							
		23:16	NHEAD[7:0]							
		15:8	PADLEN[7:0]							
		7:0	PKRS		PLIPD	PLIPEN			APM	APEN
0xB4	AES_BCNT	31:24	BCNT[31:24]							
		23:16	BCNT[23:16]							
		15:8	BCNT[15:8]							
		7:0	BCNT[7:0]							
0xB8 ... 0xBF	Reserved									
0xC0	AES_TWR0	31:24	TWEAK[31:24]							
		23:16	TWEAK[23:16]							
		15:8	TWEAK[15:8]							
		7:0	TWEAK[7:0]							
0xC4	AES_TWR1	31:24	TWEAK[31:24]							
		23:16	TWEAK[23:16]							
		15:8	TWEAK[15:8]							
		7:0	TWEAK[7:0]							
0xC8	AES_TWR2	31:24	TWEAK[31:24]							
		23:16	TWEAK[23:16]							
		15:8	TWEAK[15:8]							
		7:0	TWEAK[7:0]							
0xCC	AES_TWR3	31:24	TWEAK[31:24]							
		23:16	TWEAK[23:16]							
		15:8	TWEAK[15:8]							
		7:0	TWEAK[7:0]							
0xD0	AES_ALPHAR0	31:24	ALPHA[31:24]							
		23:16	ALPHA[23:16]							
		15:8	ALPHA[15:8]							
		7:0	ALPHA[7:0]							
0xD4	AES_ALPHAR1	31:24	ALPHA[31:24]							
		23:16	ALPHA[23:16]							
		15:8	ALPHA[15:8]							
		7:0	ALPHA[7:0]							
0xD8	AES_ALPHAR2	31:24	ALPHA[31:24]							
		23:16	ALPHA[23:16]							
		15:8	ALPHA[15:8]							
		7:0	ALPHA[7:0]							
0xDC	AES_ALPHAR3	31:24	ALPHA[31:24]							
		23:16	ALPHA[23:16]							
		15:8	ALPHA[15:8]							
		7:0	ALPHA[7:0]							
0xE0 ... 0xE3	Reserved									
0xE4	AES_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0	ACTION[2:0]			FIRSTE		WPCREN	WPITEN	WPEN
0xE8	AES_WPSR	31:24	ECLASS				SWETYP[3:0]			
		23:16								
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS

Downloaded from Arrow.com.

52.5.1 AES Control Register

Name: AES_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								UNLOCK
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								SWRST
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 24 – UNLOCK Unlock Processing
AES_WPSR must be cleared before performing the unlock command.

Value	Description
0	No effect.
1	Unlocks the processing in case of abnormal event detection if AES_WPMR.ACTION > 0.

Bit 8 – SWRST Software Reset

Value	Description
0	No effect.
1	Resets the AES. A software-triggered reset of the AES interface is performed.

Downloaded from Arrow.com.

52.5.2 AES Mode Register

Name: AES_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	TAMPCLR							
Access	R/W							
Reset	–							

Bit	23	22	21	20	19	18	17	16
	CKEY[3:0]					CFBS[2:0]		
Access	W	W	W	W		R/W	R/W	R/W
Reset	0	0	0	–		0	0	0

Bit	15	14	13	12	11	10	9	8
	LOD	OPMOD[2:0]			KEYSIZE[1:0]		SMOD[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PROCDLY[3:0]				DUALBUFF		GTAGEN	CIPHER
Access	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Reset	0	0	0	0	0		0	0

Bit 31 – TAMPCLR Tamper Clear Enable

Value	Description
0	A tamper detection event has no effect on the AES_KEYWRx key.
1	A tamper detection event immediately clears the AES_KEYWRx key.

Bits 23:20 – CKEY[3:0] Key

Value	Name	Description
0xE	PASSWD	This field must be written with 0xE the first time AES_MR is programmed. For subsequent programming of AES_MR, any value can be written, including that of 0xE. Always reads as 0.

Bits 18:16 – CFBS[2:0] Cipher Feedback Data Size

Value	Name	Description
0	SIZE_128BIT	128-bit
1	SIZE_64BIT	64-bit
2	SIZE_32BIT	32-bit
3	SIZE_16BIT	16-bit
4	SIZE_8BIT	8-bit

Bit 15 – LOD Last Output Data Mode

WARNING In DMA mode, reading to the Output Data registers before the last data encryption/decryption process may lead to unpredictable results.

Downloaded from Arrow.com.

Value	Description
0	No effect. After each end of encryption/decryption, the output data are available either on the output data registers (Manual and Auto modes) or at the address specified in the Channel Buffer Transfer Descriptor for DMA mode. In Manual and Auto modes, the DATRDY flag is cleared when at least one of the Output Data registers is read.
1	The DATRDY flag is cleared when at least one of the Input Data Registers is written. No more Output Data Register reads are necessary between consecutive encryptions/decryptions (see Last Output Data Mode).

Bits 14:12 – OPMOD[2:0] Operating Mode
For CBC-MAC operating mode, set OPMOD to CBC and LOD to 1.
When switching from an operating mode requiring the initialization vectors (e.g. CBC, GCM) to another operating mode that does not require initialization vectors (e.g. ECB) and a message of one block has been processed, initialization vector registers (AES_IVRx) must be cleared before switching to the new mode.

Value	Name	Description
0	ECB	ECB: Electronic Codebook mode
1	CBC	CBC: Cipher Block Chaining mode
2	OFB	OFB: Output Feedback mode
3	CFB	CFB: Cipher Feedback mode
4	CTR	CTR: Counter mode (16-bit internal counter)
5	GCM	GCM: Galois/Counter mode
6	XTS	XTS: XEX-based tweaked-codebook mode

Bits 11:10 – KEYSIZE[1:0] Key Size

Value	Name	Description
0	AES128	AES Key Size is 128 bits
1	AES192	AES Key Size is 192 bits
2	AES256	AES Key Size is 256 bits

Bits 9:8 – SMOD[1:0] Start Mode
If a DMA transfer is used, configure SMOD to 2. See DMA Mode for more details.

Value	Name	Description
0	MANUAL_START	Manual Mode
1	AUTO_START	Auto Mode
2	IDATAR0_START	AES_IDATAR0 access only Auto Mode (DMA)

Bits 7:4 – PROCDLY[3:0] Processing Delay
Processing Time = N × (PROCDLY + 1)
where

- N = 10 when KEYSIZE = 0
- N = 12 when KEYSIZE = 1
- N = 14 when KEYSIZE = 2

The processing time represents the number of clock cycles that the AES needs in order to perform one encryption/decryption.
Note: The best performance is achieved with PROCDLY equal to 0.

Bit 3 – DUALBUFF Dual Input Buffer

Value	Name	Description
0	INACTIVE	AES_IDATARx cannot be written during processing of previous block.
1	ACTIVE	AES_IDATARx can be written during processing of previous block when SMOD = 2. It speeds up the overall runtime of large files.

Bit 1 – GTAGEN GCM Automatic Tag Generation Enable

Downloaded from Arrow.com.

Value	Description
0	Automatic GCM Tag generation disabled.
1	Automatic GCM Tag generation enabled.

Bit 0 – CIPHER Processing Mode

Value	Description
0	Decrypts data.
1	Encrypts data.

Downloaded from Arrow.com.

52.5.3 AES Interrupt Enable Register

Name: AES_IER
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the AES Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE	PLENERR	EOPAD	TAGRDY
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								W
Reset								–

Bit 19 – SECE Security and/or Safety Event Interrupt Enable

Bit 18 – PLENERR Padding Length Error Interrupt Enable

Bit 17 – EOPAD End of Padding Interrupt Enable

Bit 16 – TAGRDY GCM Tag Ready Interrupt Enable

Bit 8 – URAD Unspecified Register Access Detection Interrupt Enable

Bit 0 – DATRDY Data Ready Interrupt Enable

Downloaded from Arrow.com.

52.5.4 AES Interrupt Disable Register

Name: AES_IDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the AES Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE	PLENERR	EOPAD	TAGRDY
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								W
Reset								–

Bit 19 – SECE Security and/or Safety Event Interrupt Disable

Bit 18 – PLENERR Padding Length Error Interrupt Disable

Bit 17 – EOPAD End of Padding Interrupt Disable

Bit 16 – TAGRDY GCM Tag Ready Interrupt Disable

Bit 8 – URAD Unspecified Register Access Detection Interrupt Disable

Bit 0 – DATRDY Data Ready Interrupt Disable

Downloaded from Arrow.com.

52.5.5 AES Interrupt Mask Register

Name: AES_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE	PLENERR	EOPAD	TAGRDY
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
								URAD
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								R
Reset								0

Bit 19 – SECE Security and/or Safety Event Interrupt Mask

Bit 18 – PLENERR Padding Length Error Interrupt Mask

Bit 17 – EOPAD End of Padding Interrupt Mask

Bit 16 – TAGRDY GCM Tag Ready Interrupt Mask

Bit 8 – URAD Unspecified Register Access Detection Interrupt Mask

Bit 0 – DATRDY Data Ready Interrupt Mask

Downloaded from Arrow.com.

52.5.6 AES Interrupt Status Register

Name: AES_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE	PLENERR	EOPAD	TAGRDY
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	URAT[3:0]							URAD
Access	R	R	R	R				R
Reset	0	0	0	0				0

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								R
Reset								0

Bit 19 – SECE Security and/or Safety Event (cleared on read)

Value	Description
0	There is no security report in AES_WPSR.
1	One security flag is set in AES_WPSR.

Bit 18 – PLENERR Padding Length Error

Value	Description
0	No Padding Length Error occurred.
1	Padding Length Error detected.

Bit 17 – EOPAD End of Padding

Value	Description
0	Padding is not over.
1	Padding phase is over.

Bit 16 – TAGRDY GCM Tag Ready

Value	Description
0	GCM Tag is not valid.
1	GCM Tag generation is complete (cleared by reading GCM Tag, starting another processing or when writing a new key).

Bits 15:12 – URAT[3:0] Unspecified Register Access (cleared by writing SWRST in AES_CR)
Only the last Unspecified Register Access Type is available through the URAT field.

Value	Name	Description
0	IDR_WR_PROCESSING	Input Data register written during the data processing when SMOD = 2 mode.
1	ODR_RD_PROCESSING	Output Data register read during the data processing.
2	MR_WR_PROCESSING	Mode register written during the data processing.
3	ODR_RD_SUBKGEN	Output Data register read during the sub-keys generation.

Downloaded from Arrow.com.

Value	Name	Description
4	MR_WR_SUBKGEN	Mode register written during the sub-keys generation.
5	WOR_RD_ACCESS	Write-only register read access.

Bit 8 – URAD Unspecified Register Access Detection Status (cleared by writing SWRST in AES_CR)

Value	Description
0	No unspecified register access has been detected since the last SWRST.
1	At least one unspecified register access has been detected since the last SWRST.

Bit 0 – DATRDY Data Ready (cleared by setting bit START or bit SWRST in AES_CR or by reading AES_ODATARx)

Value	Description
0	Output data not valid.
1	Encryption or decryption process is completed.

Note: If AES_MR.LOD = 1: In Manual and Auto mode, the DATRDY flag can also be cleared by writing at least one AES_IDATARx.

Downloaded from Arrow.com.

52.5.7 AES Key Word Register x

Name: AES_KEYWRx
Offset: 0x20 + x*0x04 [x=0..7]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

These registers are write-only to prevent the key from being read by another application.

Note: AES_KEYWRx registers are not used if the private key internal registers are selected (AES_EMR.PKRS=1).

Bit	31	30	29	28	27	26	25	24
	KEYW[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	KEYW[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	KEYW[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	KEYW[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – KEYW[31:0] Key Word
The four/six/eight 32-bit Key Word registers set the 128-bit/192-bit/256-bit cryptographic key used for AES encryption/decryption. Depending on AES_EMR.KSEL, the first key or the second key is written. See Temporary Secured Storage for Keys.
AES_KEYWR0 corresponds to the first word of the key and respectively AES_KEYWR3/AES_KEYWR5/ AES_KEYWR7 to the last one.
Whenever a new key (AES_KEYWRx) is written to the hardware, two automatic actions are processed:

- GCM hash subkey generation
- AES_GHASHRx Clear

See Key Writing and Automatic Hash Subkey Calculation for details.
These registers are write-only to prevent the key from being read by another application.
Note: To write AES_KEYWRx and start using the key immediately, AES_EMR.PKRS must be written to 0 prior to writing AES_KEYWRx.

Downloaded from Arrow.com.

52.5.8 AES Input Data Register x

Name: AES_IDATARx
Offset: 0x40 + x*0x04 [x=0..3]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	IDATA[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	IDATA[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	IDATA[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	IDATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – IDATA[31:0] Input Data Word
The four 32-bit Input Data registers set the 128-bit data block used for encryption/decryption.
AES_IDATAR0 corresponds to the first word of the data to be encrypted/decrypted, and AES_IDATAR3 to the last one.
These registers are write-only to prevent the input data from being read by another application.

Downloaded from Arrow.com.

52.5.9 AES Output Data Register x

Name: AES_ODATARx
Offset: 0x50 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ODATA[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ODATA[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ODATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ODATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ODATA[31:0] Output Data
The four 32-bit Output Data registers contain the 128-bit data block that has been encrypted/decrypted. AES_ODATAR0 corresponds to the first word, AES_ODATAR3 to the last one.

Downloaded from Arrow.com.

52.5.10 AES Initialization Vector Register x

Name: AES_IVRx
Offset: 0x60 + x*0x04 [x=0..3]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

The four 32-bit Initialization Vector registers set the 128-bit Initialization Vector data block that is used by some modes of operation as an additional initial input.

AES_IVR0 corresponds to the first word of the Initialization Vector, AES_IVR3 to the last one.

These registers are write-only to prevent the Initialization Vector from being read by another application.

For CBC, OFB and CFB modes, the IV input value corresponds to the initialization vector.

For CTR mode, the IV input value corresponds to the initial counter value.

These registers are not used in ECB mode and must not be written.

When switching from an operating mode requiring the initialization vectors (e.g. CBC, GCM) to another operating mode that does not require initialization vectors (e.g. ECB) and a message of one block has been processed, AES_IVRx must be cleared before switching to the new mode

Bit	31	30	29	28	27	26	25	24
	IV[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	IV[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	IV[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	IV[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – IV[31:0] Initialization Vector

Downloaded from Arrow.com.

52.5.11 AES Additional Authenticated Data Length Register

Name: AES_AADLENR
Offset: 0x70
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	AADLEN[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	AADLEN[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	AADLEN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	AADLEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – AADLEN[31:0] Additional Authenticated Data Length
Length in bytes of the Additional Authenticated Data (AAD) that is to be processed.
Note: The maximum byte length of the AAD portion of a message is limited to the 32-bit counter length.

Downloaded from Arrow.com.

52.5.12 AES Plaintext/Ciphertext Length Register

Name: AES_CLENR
Offset: 0x74
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	CLEN[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CLEN[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CLEN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CLEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CLEN[31:0] Plaintext/Ciphertext Length
Length in bytes of the plaintext/ciphertext (C) data that is to be processed.
Note: The maximum byte length of the C portion of a message is limited to the 32-bit counter length.

Downloaded from Arrow.com.

52.5.13 AES GCM Intermediate Hash Word Register x

Name: AES_GHASHRx
Offset: 0x78 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: R/W

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	GHASH[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GHASH[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GHASH[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GHASH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GHASH[31:0] Intermediate GCM Hash Word x
The four 32-bit Intermediate Hash Word registers expose the intermediate GHASH value. May be read to save the current GHASH value so processing can later be resumed, presumably on a later message fragment. Whenever a new key is written in AES_KEYWRx, two automatic actions are processed:

- GCM hash subkey generation
- AES_GHASHRx Clear

See Key Writing and Automatic Hash Subkey Calculation for details.
If an application software-specific hash initial value is needed for the GHASH, it must be written to AES_GHASHRx:

- after writing AES_KEYWRx, if any
- before starting the input data feed.

Downloaded from Arrow.com.

52.5.14 AES GCM Authentication Tag Word Register x

Name: AES_TAGRx
Offset: 0x88 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	TAG[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TAG[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TAG[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TAG[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TAG[31:0] GCM Authentication Tag x
The four 32-bit Tag registers contain the final 128-bit GCM Authentication tag (*T*) when GCM processing is complete. TAG0 corresponds to the first word, TAG3 to the last word.

Downloaded from Arrow.com.

52.5.15 AES GCM Encryption Counter Value Register

Name: AES_CTRR
Offset: 0x98
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CTR[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CTR[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CTR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CTR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CTR[31:0] GCM Encryption Counter
Reports the current value of the 32-bit GCM counter.

Downloaded from Arrow.com.

52.5.16 AES GCM H Word Register x

Name: AES_GCMHRx
Offset: 0x9C + x*0x04 [x=0..3]
Reset: 0x00000000
Property: R/W

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	H[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	H[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	H[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	H[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – H[31:0] GCM H Word x
The four 32-bit H Word registers contain the 128-bit GCM hash subkey *H*value.
Whenever a new key is written in AES_KEYWRx, two automatic actions are processed:

- GCM hash subkey *H* generation
- AES_GHASHRx Clear

If the application software requires a specific hash subkey, the automatically-generated H value can be overwritten in AES_GCMHRx. See Key Writing and Automatic Hash Subkey Calculation for details.
Generating a GCM hash subkey *H* by a write in AES_GCMHRx enables to:

- select the GCM hash subkey *H* for GHASH operations,
- select one operand to process a single GF128 multiply.

Downloaded from Arrow.com.

52.5.17 AES Extended Mode Register

Name: AES_EMR
Offset: 0xB0
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	BPE							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
	NHEAD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PADLEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PKRS		PLIPD	PLIPEN			APM	APEN
Access	R/W		R/W	R/W			R/W	R/W
Reset	0		0	0			0	0

Bit 31 – BPE Block Processing End

Value	Description
0	AES_ISR.DATRDY flag reports only the end message encryption processing. No intermediate block processing is reported when SMOD=0x2 or 0x3. When a DMA is used to transfer data, BPE must be cleared.
1	AES_ISR.DATRDY flag reports each end of block processing when SMOD=0x2 or 0x3. When AES_IDATARx are not loaded by a DMA and SMOD=0x2/0x3, this bit can be written to 1 to rise the AES_ISR.DATRDY flag when a new data block can be written.

Bits 23:16 – NHEAD[7:0] IPSEC Next Header

Value	Description
0-255	IPSEC Next Header field

Bits 15:8 – PADLEN[7:0] Auto Padding Length

Value	Description
0-255	Padding length in bytes

Bit 7 – PKRS Private Key Internal Register Select

Value	Description
0	The key used by the AES is in the AES_KEYWRx registers.
1	The key used by the AES is in the private key internal register written through the private key bus.

Bit 5 – PLIPD Protocol Layer Improved Performance Decipher

Value	Description
0	Protocol layer improved performance is in ciphering mode.
1	Protocol layer improved performance is in deciphering mode.

Bit 4 – PLIPEN Protocol Layer Improved Performance Enable

Downloaded from Arrow.com.

Value	Description
0	Protocol layer improved performance is disabled.
1	Protocol layer improved performance is enabled.

Bit 1 – APM Auto Padding Mode

Value	Description
0	Auto Padding performed according to IPSEC standard.
1	Auto Padding performed according to SSL standard.

Bit 0 – APEN Auto Padding Enable

Value	Description
0	Auto Padding feature is disabled.
1	Auto Padding feature is enabled.

Downloaded from Arrow.com.

52.5.18 AES Byte Counter Register

Name: AES_BCNT
Offset: 0xB4
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	BCNT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BCNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BCNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BCNT[31:0] Auto Padding Byte Counter
Auto padding byte counter value. BCNT must be greater than 0.

Downloaded from Arrow.com.

52.5.19 AES Tweak Word Register x

Name: AES_TWRx
Offset: 0xC0 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: R/W

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	TWEAK[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TWEAK[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TWEAK[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TWEAK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TWEAK[31:0] Tweak Word x
The four 32-bit Tweak Word registers contain the 128-bit Tweak value.

Downloaded from Arrow.com.

52.5.20 AES Alpha Word Register x

Name: AES_ALPHARx
Offset: 0xD0 + x*0x04 [x=0..3]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	ALPHA[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	ALPHA[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	ALPHA[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	ALPHA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – ALPHA[31:0] Alpha Word x
The four 32-bit Alpha Word registers contain the 128-bit primitive of GF(2^{128}) to use for the first processing.

Downloaded from Arrow.com.

52.5.21 AES Write Protection Mode Register

Name: AES_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ACTION[2:0]			FIRSTE		WPCREN	WPITEN	WPEN
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x414553	PASSWD	Writing any other value in this field aborts the write operation of the WPEN,WPITEN,WPCREN bits. Always reads as 0.

Bits 7:5 – ACTION[2:0] Action on Abnormal Event Detection
When the field AES_WPMR.ACTION differs from 0 and an abnormal event or internal state is detected, the AES is locked until the unlock command is issued (AES_CR.UNLOCK=1). The lock source must be cleared before performing the unlock command. If AES_WPSR.SEQE=1, the following two actions must be performed:
1/ Read AES_WPSR.
2/ Issue software reset by writing a 1 in AES_CR.SWRST.
A specific configuration applies where the sequence does not clear the lock source (AES_WPSR=0).
If AES_WPSR.SEQE remains high after the clearing sequence, then only a hardware reset will unlock the AES. Hardware reset can be performed by issuing a reset controller software reset (refer to the section “Reset Controller (RSTC)”). This condition can be met when AES_EMR.PKWL=1 and a key has been loaded through the private key bus. The key loaded through the key bus is corrupted, but it is impossible to reload a new key unless a hardware reset is issued.

Value	Name	Description
0	REPORT_ONLY	No action (stop or clear key) is performed when one of WPVS, CGD, SEQE, or SWE flags is set.
1	LOCK_WPVS_SWE	If a processing is in progress when the AES_WPSR.WPVS/SWE event detection occurs, the current processing is ended normally but no other processing is started while a AES_CR.UNLOCK command is issued.
2	LOCK_CGD_SEQE	If a processing is in progress when the AES_WPSR.CGD/SEQE event detection occurs, the current processing is ended normally but no other processing is started while a AES_CR.UNLOCK command is issued.

Downloaded from Arrow.com.

Value	Name	Description
3	LOCK_ANY_EV	If a processing is in progress when the AES_WPSR.WPVS/CGD/SEQE/SWE events detection occurs, the current processing is ended normally but no other processing is started while a AES_CR.UNLOCK command is issued.
4	CLEAR_WPVS_SWE	If a processing is in progress when the AES_WPSR.WPVS/SWE events detection occurs, the current processing is ended normally but no other processing is started while a AES_CR.UNLOCK command is issued. Moreover, the AES_KEYWRx key is immediately cleared.
5	CLEAR_CGD_SEQE	If a processing is in progress when the AES_WPSR.CGD/SEQE events detection occurs, the current processing is ended normally but no other processing is started while a AES_CR.UNLOCK command is issued. Moreover, the AES_KEYWRx key is immediately cleared.
6	CLEAR_ANY_EV	If a processing is in progress when the AES_WPSR.WPVS/CGD/SEQE/SWE events detection occurs, the current processing is ended normally but no other processing is started while a AES_CR.UNLOCK command is issued. Moreover, the AES_KEYWRx key is immediately cleared.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in AES_WPSR.WPVSRC and the last software control error type is reported in AES_WPSR.SWETYP. The AES_ISR.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in AES_WPSR.WPVSRC and only the first software control error type is reported in AES_WPSR.SWETYP. The AES_ISR.SECE flag is set at the first error occurrence within a series.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x414553 (“AES” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x414553 (“AES” in ASCII).

Bit 1 – WPITEN Write Protection Interruption Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).

Bit 0 – WPEN Write Protection Configuration Enable
See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on configuration registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).
1	Enables the write protection on configuration registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).

Downloaded from Arrow.com.

52.5.22 AES Write Protection Status Register

Name: AES_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS				SWETYP[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SWE	SEQE	CGD	WPVS
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – ECLASS Software Error Class (cleared on read)
0 (WARNING): An abnormal access that does not affect system functionality
1 (ERROR): An access is performed into key, input data, control registers while the AES is performing an encryption/decryption or a start is request by software or DMA while the key is not fully configured.

Bits 27:24 – SWETYP[3:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	A write-only register has been read (Warning).
1	WRITE_RO	AES is enabled and a write access has been performed on a read-only register (Warning).
2	UNDEF_RW	Access to an undefined address (Warning).
3	CTRL_START	Abnormal use of AES_CR.START command when DMA access is configured.
4	WEIRD_ACTION	A key write, init value write, output data read, AES_MR and AES_EMR write, GCM configuration registers write, AES_TWRx and AES_ALPHARx registers write, AES_BCNT write has been performed while a current processing is in progress (abnormal).
5	INCOMPLETE_KEY	A tentative of start is required while the key is not fully loaded into the AES_KEYWRx registers.

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS=1, WPVSRC indicates the register address offset at which a write access has been attempted.
When WPVS=0 and SWE=1, WPVSRC reports the address of the incorrect software access. As soon as WPVS=1, WPVSRC returns the address of the write-protected violation.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of AES_WPSR.
1	A software error has occurred since the last read of AES_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Downloaded from Arrow.com.

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of AES_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of AES_WPSR. This flag can only be set under abnormal operating conditions.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	The clock monitoring circuitry has not been corrupted since the last read of AES_WPSR. Under normal operating conditions, this bit is always cleared.
1	The clock monitoring circuitry has been corrupted since the last read of AES_WPSR. This flag can only be set in case of abnormal clock signal waveform (glitch).

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protect violation has occurred since the last read of AES_WPSR.
1	A write protect violation has occurred since the last read of AES_WPSR. The address offset of the violated register is reported into field WPVSRC.

Downloaded from Arrow.com.

53. Secure Hash Algorithm (SHA)

53.1 Description

The Secure Hash Algorithm (SHA) is compliant with the American *FIPS (Federal Information Processing Standard) Publication 180-2* specification.

The 512/1024-bit block of message is respectively stored in 16/32 x 32-bit registers, (SHA_IDATARx/ SHA_IODATARx) which are write-only.

As soon as the input data is written, hash processing can be started. The registers comprising the block of a message must be entered consecutively. Then, after the processing period, the message digest is ready to be read out on the 5 up to 8/16 x 32-bit output data registers (SHA_IODATARx) or through the DMA channels.

53.2 Embedded Characteristics

- Supports Hash-based Message Authentication Code (HMAC) Algorithm (HMAC-SHA1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512,)
- Compliant with FIPS Publication 180-2
- Supports Automatic Padding of Messages
- Supports Up to 2 Sets of Initial Hash Values Registers (HMAC Acceleration or other)
- Supports Automatic Check of the Hash (HMAC Acceleration or other)
- Tightly Coupled to AES for Protocol Layers Improved Performances
- Configurable Processing Period:
  - 85 clock cycles to obtain a fast SHA1 runtime, 88 clock cycles for SHA384, SHA512 or 209 Clock Cycles for Maximizing Bandwidth of Other Applications
  - 72 clock cycles to obtain a fast SHA224, SHA256 runtime or 194 clock cycles for maximizing bandwidth of other applications
- Connection to DMA Channel Capabilities Optimizes Data Transfers
- Double Input Buffer Optimizes Runtime
- Register Write Protection

53.3 Product Dependencies

53.3.1 Power Management

The SHA may be clocked through the Power Management Controller (PMC), so the programmer must first configure the PMC to enable the SHA clock.

53.3.2 Interrupt Sources

The SHA interface has an interrupt line connected to the Interrupt Controller.

Handling the SHA interrupt requires programming the Interrupt Controller before configuring the SHA.

53.4 Functional Description

The Secure Hash Algorithm (SHA) module requires a padded message according to FIPS180-2 specification. This message can be provided with the padding to the SHA module, or the padding can be automatically computed by the SHA module if the size of the message is provided. The first block of the message must be indicated to the module by a specific command. The SHA module produces an N-bit message digest each time a block is written and processing period ends, where N is 160 for SHA1, 224 for SHA224, 256 for SHA256, 384 for SHA384, 512 for SHA512. The SHA module is also capable of computing Hash-based Message Authentication Code (HMAC) algorithm.

Downloaded from Arrow.com.

53.4.1 SHA Algorithm

The SHA can process SHA1, SHA224, SHA256, SHA384, SHA512 by configuring the ALGO field in the SHA Mode register (SHA_MR).

53.4.2 HMAC Algorithm

The HMAC algorithm is as follows:

$HMAC_K(m) = h((K_0 \oplus opad) || h((K_0 \oplus ipad) || m))$

where:

- h = SHA function
- K_0 = the key K after any necessary pre-processing to form a block size key
- m = message to authenticate
- || = concatenation operator
- $\oplus$ = XOR operator
- ipad = predefined constant (0x3636...3636)
- opad = predefined constant (0x5C5C...5C5C)

The SHA provides a fully optimized processing of the HMAC algorithm by executing the following operations:

- starting the SHA algorithm from any user predefined hash value, thus ‘$h(K_0 \oplus ipad)$’ for first HMAC hash and ‘$h(K_0 \oplus opad)$’ for second HMAC hash
- performing automatic padding
- routing automatically the first hash result ‘$h((K_0 \oplus ipad) || m)$’ to the source of the second hash processing ‘$h((K_0 \oplus opad) || (\text{first hash result}))$’ including the concatenation of the first hash result to ‘$K_0 \oplus opad$’.

To perform the HMAC operation, the ALGO field value must be greater than 7, the automatic padding feature must be enabled (MSGSIZE and BYTCNT fields differ from 0) and the SHA internal initial hash value registers 0 and 1 must be configured, respectively, with the hash results of input blocks “$K_0 \oplus ipad$” and “$K_0 \oplus opad$” (see Internal Registers for Initial Hash Value or Expected Hash Result).

The size of the message (‘m’) must be written in the MSGSIZE and BYTCNT fields.

The FIRST bit in the SHA Control register (SHA_CR) should be set before writing the first block of the message.

The SHA can process HMAC-SHA1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512 by configuring the ALGO field in the SHA_MR.

53.4.3 Processing Period

When SHA is enabled and DMA is used to write the messages, the inherent processing period may result, depending on the application, in a significant bandwidth usage at system bus level. In some applications, it may be important to keep as much bandwidth as possible for the other peripherals (e.g. CPU, other DMA channels). The SHA engine inherent processing period can be configured to reduce the bandwidth required by writing SHA_MR.PROCDLY=1.

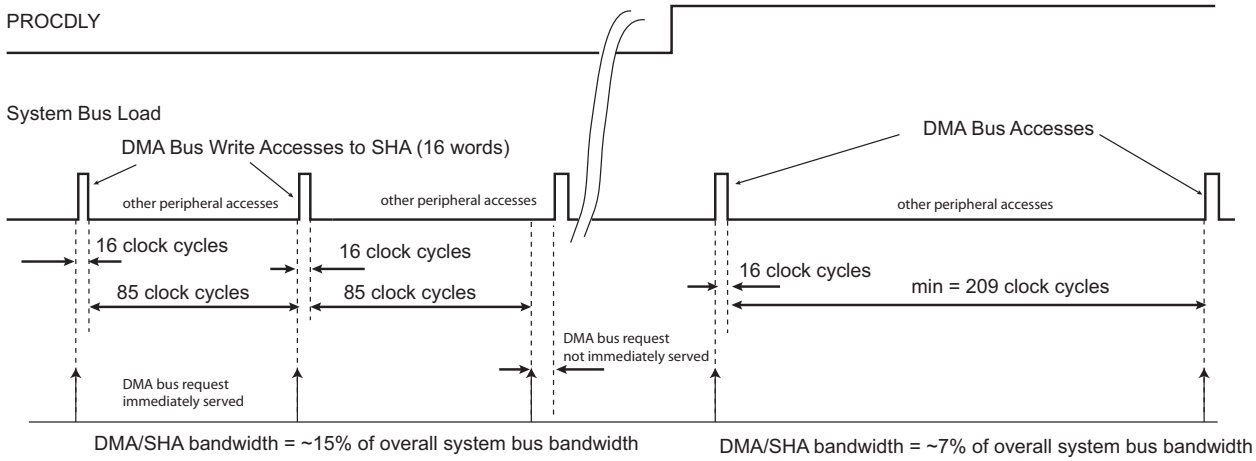
In SHA1 mode, the shortest processing period is 85 clock cycles + 2 clock cycles for start command synchronization (SHA_MR.PROCDLY=0). The longest period is 209 clock cycles + 2 clock cycles when SHA_MR.PROCDLY=1 (see the figure below).

In SHA256 mode, the shortest processing period is 72 clock cycles + 2 clock cycles for start command synchronization (SHA_MR.PROCDLY=0). The longest period is 194 clock cycles + 2 clock cycles when SHA_MR.PROCDLY=1.

In SHA384 or SHA512 mode, the shortest processing period is 88 clock cycles + 2 clock cycles for start command synchronization. The longest period is 209 clock cycles + 2 clock cycles.

Downloaded from Arrow.com.

Figure 53-1. Bandwidth Usage in SHA-1 Mode



53.4.4 Double Input Buffer

The SHA Input Data registers (SHA_IDATARx) can be double-buffered to reduce the runtime of large messages.

Double-buffering allows a new message block to be written while the previous message block is being processed. This is only possible when DMA accesses are performed (SMOD = 2).

The DUALBUFF bit in the SHA_MR must be set to have double input buffer access.

53.4.5 Internal Registers for Initial Hash Value or Expected Hash Result

The SHA module embeds two sets of internal registers (IR0, IR1) to store different data used by the SHA or HMAC algorithms (See the figure below). These internal registers are accessed through SHA Input Data registers (SHA_IDATARx).

When the ALGO field selects SHA algorithms, IR0 can be configured with a user initial hash value. This initial hash value can be used to compute a custom hash algorithm with two sets of different initial constants, or to continue a hash computation by providing the intermediate hash value previously returned by the SHA module.

When the ALGO field selects SHA algorithms, IR1 can be configured with either a user initial hash value or an expected hash result. The expected hash result must be configured in the IR1 if the field CHECK = 1 (refer to Automatic Check). If the field CHECK = 0 or 2, IR1 can be configured with a user initial hash value that differs from IR0 value.

When the ALGO field selects HMAC algorithms, IR0 must be configured with the hash result of $K_0 \oplus$ ipad and IR1 must be configured with the hash result of $K_0 \oplus$ opad. These pre-computed first blocks speed up the HMAC computation by saving the time to compute the intermediate hash values of the first block which is constant while the secret key is constant (See HMAC Algorithm).

Table 53-1. Configuration Values of Internal Registers

Register	SHA Modes (ALGO < 8)			HMAC Modes (ALGO > 7)
	CHECK = 0	CHECK = 1	CHECK = 2	
IR0	User Initial Hash	User Initial Hash	User Initial Hash	hash($K_0 \oplus$ ipad)
IR1	User Initial Hash	Expected Hash Result	User Initial Hash	hash($K_0 \oplus$ opad)

To calculate the initial HMAC values, follow this sequence:

1. Calculate K_0.
2. Calculate $K_0 \oplus$ ipad and $K_0 \oplus$ opad.
3. Perform a hash of the result of $K_0 \oplus$ ipad and $K_0 \oplus$ opad (auto-padding must be disabled for that type of hash).
4. Write h($K_0 \oplus$ ipad) and h($K_0 \oplus$ opad) in IR0 and IR1 respectively.

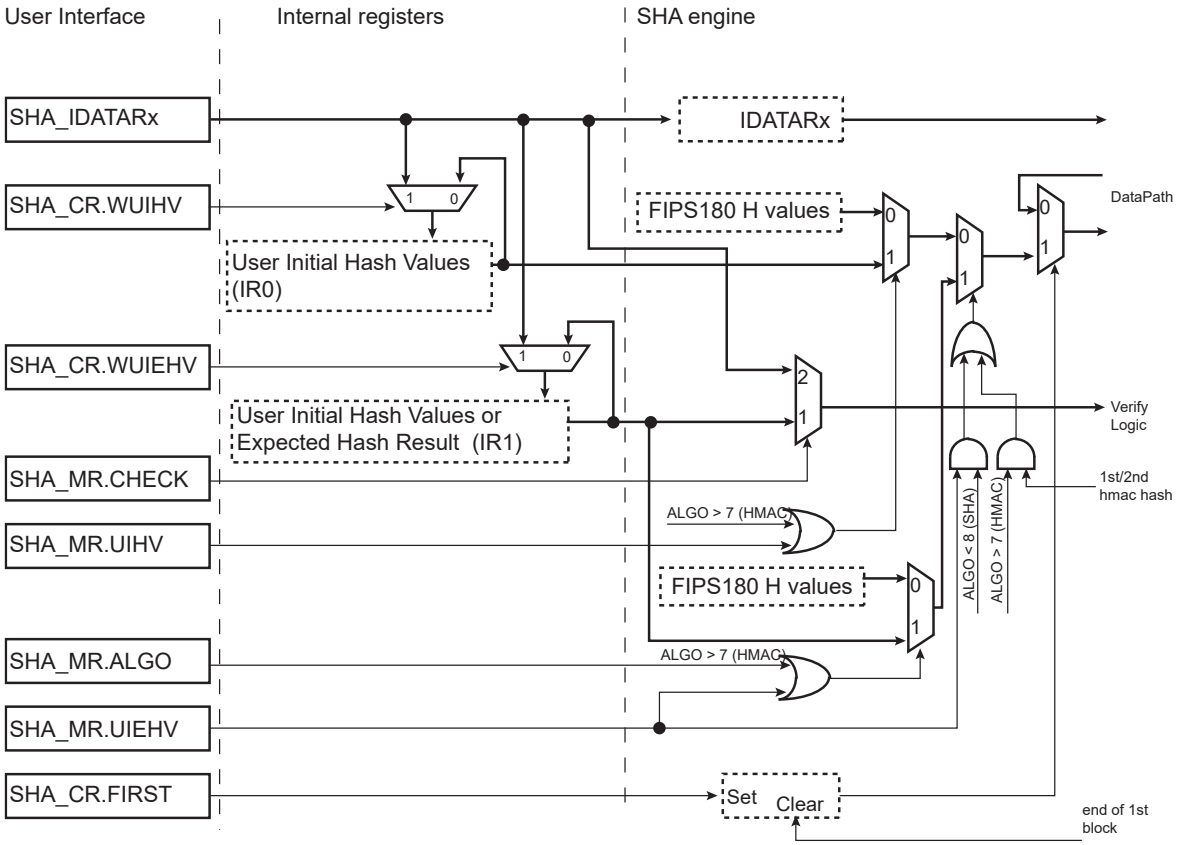
Downloaded from Arrow.com.

To write IR0 or IR1, follow this sequence:

1. Set SHA_CR. WUIHV (IR0) or SHA_CR.WUIEHV (IR1).
2. Write the data in SHA_IDATARx. The number of registers to write depends on the type of data (user initial hash values or expected hash result) and on the type of algorithm selected:
   – SHA_IDATAR0 to SHA_IDATAR4 for data used in algorithms based on SHA1
   – SHA_IDATAR0 to SHA_IDATAR7 for data used in algorithms based on SHA256
   – SHA_IDATAR0 to SHA_IDATAR15 for data used in algorithms based on SHA512
   – SHA_IDATAR0 to SHA_IDATAR6 for expected hash result of algorithms based on SHA224
   – SHA_IDATAR0 to SHA_IDATAR11 for expected hash result of algorithms based on SHA384
3. Clear SHA_CR.WUIHV or SHA_CR.WUIEHV.

IR0 and IR1 are automatically selected for HMAC processing if the field ALGO selects HMAC algorithms. If SHA algorithms are selected, the internal registers are selected if the corresponding UIHV or UIEHV bits are set.

Figure 53-2. User Initial Hash Value and Expected Hash Internal Register Access



53.4.6 Automatic Padding

The SHA module features an automatic padding computation to speed up the execution of the algorithm.

The automatic padding function requires the following information:

- Complete message size in bytes to be written in the MSGSIZE field of the SHA Message Size register (SHA_MSR).
  The size of the message is written at the end of the last block, as required by the FIPS180-2 specification (the size is automatically converted into a bit-size).
- Number of remaining bytes (to write in the SHA_IDATARx) to be written in the BYTCNT field of the SHA Bytes Count register (SHA_BCR).
  Automatic padding occurs when the BYTCNT field reaches 0. At each write in the SHA Input registers, the BYTCNT field value is decreased by the number of bytes written.

Downloaded from Arrow.com.

The BYTCNT field value must be written with the same value as the MSGSIZE field value if the full message is processed. If the message is partially preprocessed and an initial hash value is used, BYTCNT must be written with the remaining bytes to hash while MSGSIZE holds the message size.

To disable the automatic padding feature, the MSGSIZE and BYTCNT fields must be configured with 0.

53.4.7 Automatic Check

The SHA module features an automatic check of the hash result with the expected hash. A check failure can generate an interrupt if configured in the SHA Interrupt Enable register (SHA_IER).

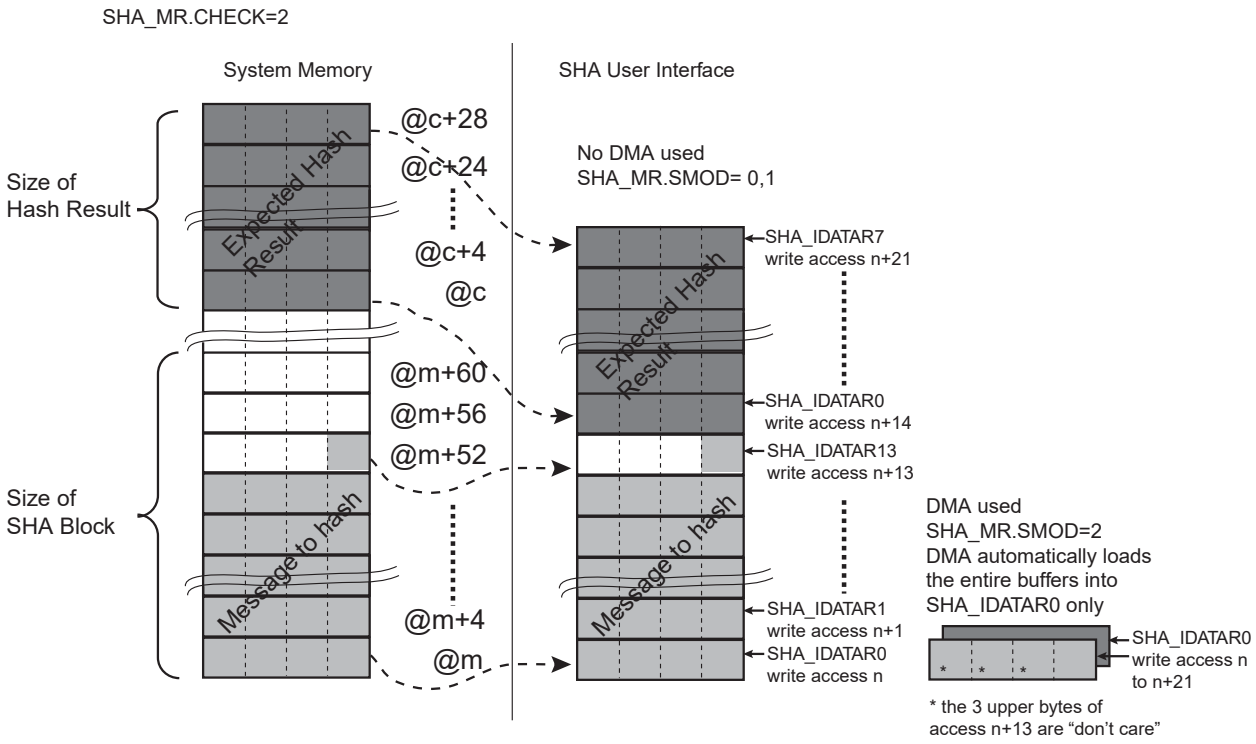
Automatic check requires the automatic padding feature to be enabled (MSGSIZE and BYTCNT fields must be greater than 0).

There are two methods to configure the expected hash result:

- if SHA_MR.CHECK = 1, the expected hash result is read from the internal register (IR1). This method cannot be used when HMAC algorithms is selected because this register is already used to store user initial hash values for the second hash processing. IR1 cannot be read by software.
- If SHA_MR.CHECK = 2, the expected hash result is written in the SHA_IDATARx after the message.

When SHA_MR.CHECK = 2, the method can provide more flexibility of use if a message is stored in system memory together with its expected hash result. A DMA with linked list can be used to ease the transfer of the message and its expected hash result.

Figure 53-3. Message and Expected Hash Result Memory Mapping



The number of 32-bit words of the hash result to check with the expected hash can be selected with SHA_MR.CHKCNT. The status of the check is available in the CHKST field in the SHA Interrupt Status register (SHA_ISR).

An interrupt can be generated (if enabled) when the check is completed. The check occurs several clock cycles after the computation of the requested hash, so the interrupt and the CHECKF bit are set several clock cycles after the DATRDY flag of the SHA_ISR.

53.4.8 Protocol Layers Improved Performances

The SHA can be tightly coupled to the AES module to improve performances when processing protocol layers such as IPsec or OpenSSL.

Downloaded from Arrow.com.

When the AES is configured to be tightly coupled to SHA (AES_MR), SHA must be always configured in Double Buffer mode (SHA_MR.DUALBUFF = 1).

Refer to the section “Advanced Encryption Standard (AES)” for details.

53.4.9 Start Modes

SHA_MR.SMOD is used to select the Hash Processing Start mode.

53.4.9.1 Manual Mode

In Manual mode, the sequence is as follows:

1. Set SHA_IER.DATRDY (Data Ready) , depending on whether an interrupt is required at the end of processing.
2. If the initial hash values differ from the FIPS standard, set SHA_MR.UIHV and/or SHA_MR.UIEHV. If the initial hash values comply with the FIPS180-2 specification, clear SHA_MR.UIHV and/or SHA_MR.UIEHV.
3. If automatic padding is required, configure SHA_MSR.MSGSIZE with the number of bytes of the message, and configure SHA_BCR.BYTCNT with the remaining number of bytes to write. The BYTCNT field must be written with a value different from MSGSIZE field value if the message is preprocessed and completed by using user initial hash values.
   If automatic padding is not required, configure SHA_MSR.MSGSIZE and SHA_BCR.BYTCNT to 0.
4. For the first block of a message, the FIRST command must be set by writing a 1 into the corresponding bit of the Control register (SHA_CR). For the other blocks, there is nothing to write.
5. Write the block to be processed in the SHA_IDATARx.
6. To begin processing, set SHA_CR.START.
7. When processing is completed, the bit DATRDY in the Interrupt Status register (SHA_ISR) rises. If an interrupt has been enabled by setting SHA_IER.DATRDY, the interrupt line of the SHA is activated.
8. Repeat the write procedure for each block, start procedure and wait for the interrupt procedure up to the last block of the entire message. Each time the start procedure is complete, the DATRDY flag is cleared.
9. After the last block is processed (DATRDY flag is set, if an interrupt has been enabled by setting SHA_IER.DATRDY, the interrupt line of the SHA is activated), read the message digest in the Output Data registers. The DATRDY flag is automatically cleared when reading the SHA_IODATARx registers.

53.4.9.2 Auto Mode

In Auto mode, processing starts as soon as the correct number of SHA_IDATARx is written. No action in the SHA_CR is necessary.

53.4.9.3 DMA Mode

The DMA can be used in association with the SHA to perform the algorithm on a complete message without any action by the software during processing.

SHA_MR.SMOD must be configured to 2.

The DMA must be configured with non-incremental addresses.

The start address of any transfer descriptor must be set to point to the SHA_IDATAR0.

The DMA chunk size must be set to transfer, for each trigger request, 16 words of 32 bits.

The FIRST bit of the SHA_CR must be set before starting the DMA when the first block is transferred.

The DMA generates an interrupt when the end of buffer transfer is completed but the SHA processing is still in progress. The end of SHA processing is indicated by the flag DATRDY in the SHA_ISR.

If automatic padding is disabled, the end of SHA processing requires two interrupts to be verified. The DMA end of transfer interrupt must be verified first, then the SHA DATRDY interrupt must be enabled and verified.

If automatic padding is enabled, the end of SHA processing requires only one interrupt to be verified. The DMA end of transfer is not required, so the SHA DATRDY interrupt must be enabled prior to start the DMA and DATRDY interrupt is the only one to be verified. Refer to the figures below.

Downloaded from Arrow.com.

Figure 53-4. Interrupts Processing with DMA

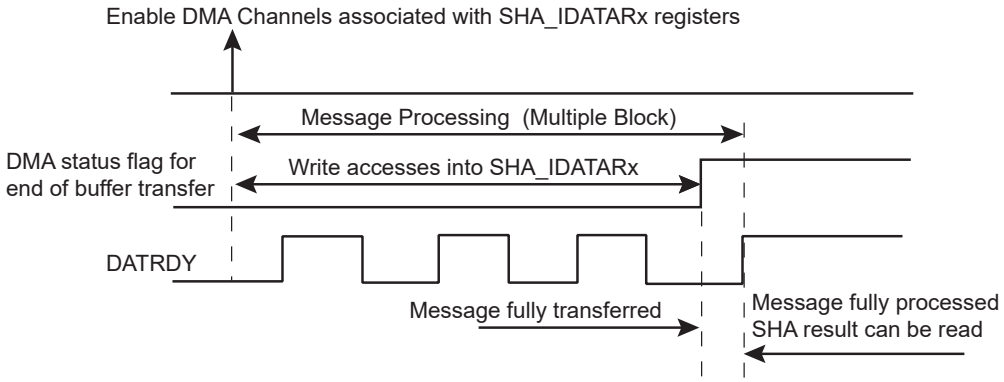
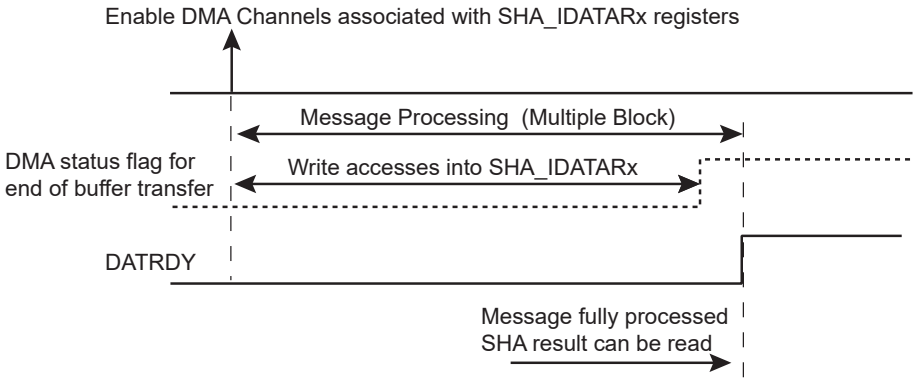


Figure 53-5. Interrupts Processing with DMA and Automatic Padding



53.4.9.4 SHA Register Endianness

In Arm processor-based products, the system bus and processors manipulate data in little-endian form. The SHA interface requires little-endian format words. However, in accordance with the protocol of FIPS 180-2 specification, data is collected, processed and stored by the SHA algorithm in big-endian form.

The following example illustrates how to configure the SHA:

If the first 64 bits of a message (according to FIPS 180-2, i.e., big-endian format) to be processed is 0xcafedeca_01234567, then the SHA_IDATAR0 and SHA_IDATAR1 registers must be written with the following pattern:

- SHA_IDATAR0 = 0xcadefeca
- SHA_IDATAR1 = 0x67452301

In a little-endian system, the message (according to FIPS 180-2) starting with pattern 0xcafedeca_01234567 is stored into memory as follows:

- 0xca stored at initial offset (for example 0x00),
- then 0xfe stored at initial offset + 1 (i.e., 0x01),
- 0xde stored at initial offset + 2 (i.e., 0x02),
- 0xca stored at initial offset + 3 (i.e., 0x03).

If the message is received through a serial-to-parallel communication channel, the first received character is 0xca and it is stored at the first memory location (initial offset). The second byte, 0xfe, is stored at initial offset + 1.

When reading on a 32-bit little-endian system bus, the first word read back from system memory is 0xcadefeca.

When the SHA_IODATARx registers are read, the hash result is organized in little-endian format, allowing system memory storage in the same format as the message.

Taking an example from the FIPS 180-2 specification Appendix B.1, the endian conversion can be observed.

For this example, the 512-bit message is:

Downloaded from Arrow.com.

0x6162638000000000

The peripheral clock of the SHA is monitored by a specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the SHA. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the SHA_WPSR.CGD flag is set, an abnormal condition occurred on the peripheral clock. This flag is not set under normal operating conditions.

The internal sequencer of the SHA is also monitored, and if an abnormal state is detected, the SHA_WPSR.SEQE flag is set. This flag is not set under normal operating conditions.

Software accesses to the SHA are monitored and if an incorrect access is performed, the SHA_WPSR.SWE flag is set. The type of incorrect/abnormal software access is reported in the SHA_WPSR.SWETYP field (see SHA Write Protection Status Register for details), e.g., reading the SHA_ODATARx when the SHA_ISR.DATRDY flag is cleared is an error. SHA_WPSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The CGD, SEQE, SWE and WPVS flags are automatically cleared when SHA_WPSR is read.

If one of these flags is set, the SHA_ISR.SECE flag is set and can trigger an interrupt if SHA_IMR.SECE is ‘1’. SECE is cleared by reading SHA_ISR.

It is possible to configure an action to be performed by SHA as soon as an abnormal event detection occurs. If SHA_WPMR.ACTION > 0, a lock is performed. When a lock occurs, the current processing is ended normally but any new processing is not performed whatever the start mode of operation (see SHA_MR.SMOD).

A locked state of the SHA is unlocked as follows:

1. Read SHA_WPSR.
2. Disable the source of tamper if the tamper is enabled.
3. Write a ‘1’ to SHA_CR.UNLOCK.

It is possible to select the type of event that will lock the SHA in case of abnormal event detection. See SHA_WPMR.ACTION for details.

If SHA_MR.TMPLCK=1 and the tamper pin is active, the SHA is locked whatever the value of the field SHA_WPMR.ACTION.

Downloaded from Arrow.com.

53.5 Register Summary

Offset	Name	Bit Pos.								
0x00	SHA_CR	31:24								UNLOCK
		23:16								
		15:8			WUIEHV	WUIHV				SWRST
		7:0				FIRST				START
0x04	SHA_MR	31:24	CHKCNT[3:0]						CHECK[1:0]	
		23:16								DUALBUFF
		15:8	TMPLCK				ALGO[3:0]			
		7:0	BPE	UIEHV	UIHV	PROCDLY	AOE		SMOD[1:0]	
0x08 ... 0x0F	Reserved									
0x10	SHA_IER	31:24								SECE
		23:16								CHECKF
		15:8								URAD
		7:0								DATRDY
0x14	SHA_IDR	31:24								SECE
		23:16								CHECKF
		15:8								URAD
		7:0								DATRDY
0x18	SHA_IMR	31:24								SECE
		23:16								CHECKF
		15:8								URAD
		7:0								DATRDY
0x1C	SHA_ISR	31:24								SECE
		23:16	CHKST[3:0]							CHECKF
		15:8		URAT[2:0]						URAD
		7:0				WRDY				DATRDY
0x20	SHA_MSR	31:24	MSGSIZE[31:24]							
		23:16	MSGSIZE[23:16]							
		15:8	MSGSIZE[15:8]							
		7:0	MSGSIZE[7:0]							
0x24 ... 0x2F	Reserved									
0x30	SHA_BCR	31:24	BYTCNT[31:24]							
		23:16	BYTCNT[23:16]							
		15:8	BYTCNT[15:8]							
		7:0	BYTCNT[7:0]							
0x34 ... 0x3F	Reserved									
0x40	SHA_IDATAR0	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
...										
0x7C	SHA_IDATAR15	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
0x80	SHA_IODATAR0	31:24	IODATA[31:24]							
		23:16	IODATA[23:16]							
		15:8	IODATA[15:8]							
		7:0	IODATA[7:0]							
...										

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0xBC	SHA_IODATAR15	31:24	IODATA[31:24]							
		23:16	IODATA[23:16]							
		15:8	IODATA[15:8]							
		7:0	IODATA[7:0]							
0xC0 ... 0xE3	Reserved									
0xE4	SHA_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0		ACTION[1:0]		FIRSTE		WPCREN	WPITEN	WPEN
0xE8	SHA_WPSR	31:24	ECLASS				SWETYP[3:0]			
		23:16								
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS

Downloaded from Arrow.com.

53.5.1 SHA Control Register

Name: SHA_CR
Offset: 0x00
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
								UNLOCK
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			WUIEHV	WUIHV				SWRST
Access			W	W				W
Reset			–	–				–

Bit	7	6	5	4	3	2	1	0
				FIRST				START
Access				W				W
Reset				–				–

Bit 24 – UNLOCK Unlock Processing
SHA_WPSR must be cleared before performing the unlock command.

Value	Description
0	No effect.
1	Unlocks the processing in case of abnormal event detection if SHA_WPMR.ACTION > 0.

Bit 13 – WUIEHV Write User Initial or Expected Hash Values

Value	Description
0	SHA_IDATARx accesses are routed to the data registers.
1	SHA_IDATARx accesses are routed to the internal registers (IR1).

Bit 12 – WUIHV Write User Initial Hash Values

Value	Description
0	SHA_IDATARx accesses are routed to the data registers.
1	SHA_IDATARx accesses are routed to the internal registers (IR0).

Bit 8 – SWRST Software Reset

Value	Description
0	No effect.
1	Resets the SHA. A software-triggered hardware reset of the SHA interface is performed.

Bit 4 – FIRST First Block of a Message

Value	Description
0	No effect.
1	Indicates that the next block to process is the first one of a message.

Bit 0 – START Start Processing

Value	Description
0	No effect.

Downloaded from Arrow.com.

Value	Description
1	Starts manual hash algorithm process.

Downloaded from Arrow.com.

53.5.2 SHA Mode Register

Name: SHA_MR
Offset: 0x04
Reset: 0x0000100
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	CHKCNT[3:0]						CHECK[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	23	22	21	20	19	18	17	16
								DUALBUFF
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	TMPLCK				ALGO[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	1

Bit	7	6	5	4	3	2	1	0
	BPE	UIEHV	UIHV	PROCDLY	AOE		SMOD[1:0]	
Access	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Reset	0	0	0	0	0		0	0

Bits 31:28 – CHKCNT[3:0] Check Counter
Number of 32-bit words to check. The value 0 indicates that the number of words to compare will be based on the algorithm selected (5 words for SHA1, 7 words for SHA224, 8 words for SHA256, 12 words for SHA384, 16 words for SHA512).

Bits 25:24 – CHECK[1:0] Hash Check
Values not listed in table must be considered as “reserved”.

Value	Name	Description
0	NO_CHECK	No check is performed
1	CHECK_EHV	Check is performed with expected hash stored in internal expected hash value registers.
2	CHECK_MESSAGE	Check is performed with expected hash provided after the message.

Bit 16 – DUALBUFF Dual Input Buffer

Value	Name	Description
0	INACTIVE	SHA_IDATARx and SHA_IODATARx cannot be written during processing of previous block.
1	ACTIVE	SHA_IDATARx and SHA_IODATARx can be written during processing of previous block when SMOD value = 2. It speeds up the overall runtime of large files.

Bit 15 – TMPLCK Tamper Lock Enable

Value	Description
0	A tamper event has no effect.
1	A tamper event locks the SHA until the tamper root cause is cleared and SHA_CR.UNLOCK is written to 1.

Bits 11:8 – ALGO[3:0] SHA Algorithm
Values not listed in the table must be considered as “reserved”.

Downloaded from Arrow.com.

Value	Name	Description
0	SHA1	SHA1 algorithm processed
1	SHA256	SHA256 algorithm processed
2	SHA384	SHA384 algorithm processed
3	SHA512	SHA512 algorithm processed
4	SHA224	SHA224 algorithm processed
8	HMAC_SHA1	HMAC algorithm with SHA1 Hash processed
9	HMAC_SHA256	HMAC algorithm with SHA256 Hash processed
10	HMAC_SHA384	HMAC algorithm with SHA384 Hash processed
11	HMAC_SHA512	HMAC algorithm with SHA512 Hash processed
12	HMAC_SHA224	HMAC algorithm with SHA224 Hash processed
13	Reserved	–
14	Reserved	–

Bit 7 – BPE Block Processing End
When SMOD=2 and ALGO<5, the SHA_ISR.DATRDY flag rises when each block has been processed.
When SMOD=2 and ALGO>7, the SHA_ISR.DATRDY rises when all blocks except the last one have been processed.

Value	Description
0	BPE must be cleared when a DMA transfers data. When SMOD=2, SHA_ISR.DATRDY flag rises only when the SHA or HMAC processing cycle has completed. No intermediate block processing is reported.
1	When processing small messages, data transfer by software can improve performance compared to DMA. In this case, BPE can be written to 1, forcing the SHA_ISR.DATRDY to rise when a data must be loaded into SHA_IDATARx.

Bit 6 – UIEHV User Initial or Expected Hash Value Registers

Value	Description
0	The SHA algorithm is started with the standard initial values as defined in the FIPS180-2 specification.
1	The SHA algorithm is started with the user initial hash values stored in the internal register 1 (IR1). If HMAC is configured, UIEHV has no effect (i.e. IR1 is always selected).

Bit 5 – UIHV User Initial Hash Values

Value	Description
0	The SHA algorithm is started with the standard initial values as defined in the FIPS180-2 specification.
1	The SHA algorithm is started with the user initial hash values stored in the internal register 0 (IR0). If HMAC is configured, UIHV has no effect (i.e. IR0 is selected).

Bit 4 – PROCDLY Processing Delay
When SHA1 algorithm is processed, runtime period is either 85 or 209 clock cycles.
When SHA256 or SHA224 algorithm is processed, runtime period is either 72 or 194 clock cycles.
When SHA384 or SHA512 algorithm is processed, runtime period is either 88 or 209 clock cycles.

Value	Name	Description
0	SHORTEST	SHA processing runtime is the shortest one
1	LONGEST	SHA processing runtime is the longest one (reduces the SHA bandwidth requirement, reduces the system bus overload)

Bit 3 – AOE Always ON Enable

Value	Description
0	The SHA operates in functional operating modes.
1	As soon as a START command is written, the SHA processes dummy calculations until AOE=0, without software intervention. This can be used to create an additional current consumption when AES is used to encrypt/decrypt.

Bits 1:0 – SMOD[1:0] Start Mode
Values not listed in the table must be considered as “reserved”.
If a DMA transfer is used, configure the SMOD value to 2. See DMA Mode for details.

Downloaded from Arrow.com.

Value	Name	Description
0	MANUAL_START	Manual mode
1	AUTO_START	Auto mode
2	IDATAR0_START	SHA_IDATAR0 access only mode (mandatory when DMA is used)

Downloaded from Arrow.com.

53.5.3 SHA Interrupt Enable Register

Name: SHA_IER
Offset: 0x10
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
								SECE
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
								CHECKF
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								W
Reset								–

Bit 24 – SECE Security and/or Safety Event Interrupt Enable

Bit 16 – CHECKF Check Done Interrupt Enable

Bit 8 – URAD Unspecified Register Access Detection Interrupt Enable

Bit 0 – DATRDY Data Ready Interrupt Enable

Downloaded from Arrow.com.

53.5.4 SHA Interrupt Disable Register

Name: SHA_IDR
Offset: 0x14
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
								SECE
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
								CHECKF
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								W
Reset								–

Bit 24 – SECE Security and/or Safety Event Interrupt Disable

Bit 16 – CHECKF Check Done Interrupt Disable

Bit 8 – URAD Unspecified Register Access Detection Interrupt Disable

Bit 0 – DATRDY Data Ready Interrupt Disable

Downloaded from Arrow.com.

53.5.5 SHA Interrupt Mask Register

Name: SHA_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
								SECE
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
								CHECKF
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
								URAD
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								R
Reset								0

Bit 24 – SECE Security and/or Safety Event Interrupt Mask

Bit 16 – CHECKF Check Done Interrupt Mask

Bit 8 – URAD Unspecified Register Access Detection Interrupt Mask

Bit 0 – DATRDY Data Ready Interrupt Mask

Downloaded from Arrow.com.

53.5.6 SHA Interrupt Status Register

Name: SHA_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
								SECE
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
	CHKST[3:0]							CHECKF
Access	R	R	R	R				R
Reset	0	0	0	0				0

Bit	15	14	13	12	11	10	9	8
		URAT[2:0]						URAD
Access		R	R	R				R
Reset		0	0	0				0

Bit	7	6	5	4	3	2	1	0
				WRDY				DATRDY
Access				R				R
Reset				0				0

Bit 24 – SECE Security and/or Safety Event

Value	Description
0	There is no report in SHA_WPSR.
1	There is a Security and/or Safety Event reported in SHA_WPSR.

Bits 23:20 – CHKST[3:0] Check Status (cleared by writing START or SWRST bits in SHA_CR or by reading SHA_IODATARx)
Value 5 indicates identical hash values (expected hash = hash result). Any other value indicates different hash values.

Bit 16 – CHECKF Check Done Status (cleared by writing START or SWRST bits in SHA_CR or by reading SHA_IODATARx)

Value	Description
0	Hash check has not been computed.
1	Hash check has been computed, status is available in the CHKST bits.

Bits 14:12 – URAT[2:0] Unspecified Register Access Type (cleared by writing a 1 to SWRST bit in SHA_CR)
Only the last Unspecified Register Access Type is available through the URAT field.

Value	Name
0	SHA_IDATAR0 to SHA_IDATAR15 written during data processing in DMA mode (URAD = 1 and URAT = 0 can occur only if DUALBUFF is cleared in SHA_MR)
1	Output Data Register read during data processing
2	SHA_MR written during data processing
3	Write-only register read access

Bit 8 – URAD Unspecified Register Access Detection Status (cleared by writing a 1 to SWRST bit in SHA_CR)

Value	Description
0	No unspecified register access has been detected since the last SWRST.
1	At least one unspecified register access has been detected since the last SWRST.

Downloaded from Arrow.com.

Bit 4 – WRDY Input Data Register Write Ready

Value	Description
0	SHA_IDATAR0 cannot be written
1	SHA_IDATAR0 can be written

Bit 0 – DATRDY Data Ready (cleared by writing a 1 to bit SWRST or START in SHA_CR, or by reading SHA_IODATARx)

Value	Description
0	Output data is not valid.
1	512/1024-bit block process is completed. DATRDY is cleared when one of the following conditions is met: • Bit START in SHA_CR is set. • Bit SWRST in SHA_CR is set. • The hash result is read.

Downloaded from Arrow.com.

53.5.7 SHA Message Size Register

Name: SHA_MSR
Offset: 0x20
Reset: 0x0
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	MSGSIZE[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MSGSIZE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MSGSIZE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MSGSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – MSGSIZE[31:0] Message Size
The size in bytes of the message. When MSGSIZE differs from 0, the SHA appends the corresponding value converted in bits after the padding section, as described in the FIPS180-2 specification.
To disable automatic padding, MSGSIZE field must be written to 0.

Downloaded from Arrow.com.

53.5.8 SHA Bytes Count Register

Name: SHA_BCR
Offset: 0x30
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	BYTCNT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BYTCNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BYTCNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BYTCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BYTCNT[31:0] Remaining Byte Count Before Auto Padding
When the hash processing starts from the beginning of a message (without preprocessed hash part), BYTCNT must be written with the same value as the MSGSIZE. If a part of the message has been already hashed and the hash does not start from the beginning, BYTCNT must be configured with the number of bytes remaining to process before padding section.
When read, provides the size in bytes of message remaining to be written before the automatic padding starts.
BYTCNT field is automatically updated each time a write occurs in the SHA_IDATARx and SHA_IODATARx.
When BYTCNT reaches 0, the MSGSIZE is converted into bit count and appended at the end of the message after the padding as described in the FIPS180-2 specification.
To disable automatic padding, MSGSIZE and BYTCNT fields must be written to 0.

Downloaded from Arrow.com.

53.5.9 SHA Input Data Register x

Name: SHA_IDATARx
Offset: 0x40 + x*0x04 [x=0..15]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	IDATA[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	IDATA[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	IDATA[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	IDATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – IDATA[31:0] Input Data
32-bit Input Data registers load the data block used for hash processing.
These registers are write-only to prevent reading of input data by another application.
SHA_IDATAR0 corresponds to the first word of the block, SHA_IDATAR15 to the last word of the last block in case SHA algorithm is set to SHA1, SHA224, SHA256, or SHA_IODATAR15 to the last word of the block if SHA algorithm is SHA384 or SHA512 (see SHA Input/Output Data Register x).

Downloaded from Arrow.com.

53.5.10 SHA Input/Output Data Register x

Name: SHA_IODATARx
Offset: 0x80 + x*0x04 [x=0..15]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	IODATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	IODATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IODATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IODATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – IODATA[31:0] Input/Output Data
These registers can be used to read the resulting message digest and to write the second part of the message block when the SHA algorithm is SHA-384 or SHA-512.
SHA_IODATAR0 to SHA_IODATAR15 can be written or read but reading these offsets does not return the content of corresponding parts (words) of the message block. Only results from SHA calculation can be read through these registers.
When SHA processing is in progress, these registers return 0x0000.
SHA_IODATAR0 corresponds to the first word of the message digest; SHA_IODATAR4 to the last one in SHA1 mode, SHA_ODATAR6 in SHA224, SHA_IODATAR7 in SHA256, SHA_IODATAR11 in SHA384 or SHA_IODATAR15 in SHA512.
When SHA224 is selected, the content of SHA_ODATAR7 must be ignored.
When SHA384 is selected, the content of SHA_IODATAR12 to SHA_IODATAR15 must be ignored.

Downloaded from Arrow.com.

53.5.11 SHA Write Protection Mode Register

Name: SHA_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		ACTION[1:0]		FIRSTE		WPCREN	WPITEN	WPEN
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x534841	PASSWD	Writing any other value in this field aborts the write operation of the WPEN,WPITEN,WPCREN bits. Always reads as 0.

Bits 6:5 – ACTION[1:0] Action on Abnormal Event Detection

Value	Name	Description
0	REPORT_ONLY	No action (stop or clear key) is performed when one of WPVS,CGD,SEQE, or SWE flag is set.
1	LOCK_WPVS_SWE	If a processing is in progress when the SHA_WPSR.WPVS/SWE event detection occurs, the current processing is ended normally but no other processing is started while a SHA_CR.UNLOCK command is issued.
2	LOCK_CGD_SEQE	If a processing is in progress when the SHA_WPSR.CGD/SEQE event detection occurs, the current processing is ended normally but no other processing is started while a SHA_CR.UNLOCK command is issued.
3	LOCK_ANY_EV	If a processing is in progress when the SHA_WPSR.WPVS/CGD/SEQE/SWE events detection occurs, the current processing is ended normally but no other processing is started while a SHA_CR.UNLOCK command is issued.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in SHA_WPSR.WPVSRC and the last software control error type is reported in SHA_WPSR.SWETYP. The SHA_ISR.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in SHA_WPSR.WPVSRC and only the first software control error type is reported in SHA_WPSR.SWETYP. The SHA_ISR.SECE flag is set at the first error occurrence within a series.

Downloaded from Arrow.com.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x534841 (“SHA” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x534841 (“SHA” in ASCII).

Bit 1 – WPITEN Write Protection Interruption Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x534841 (“SHA” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x534841 (“SHA” in ASCII).

Bit 0 – WPEN Write Protection Configuration Enable
See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on configuration registers if WPKEY corresponds to 0x534841 (“SHA” in ASCII).
1	Enables the write protection on configuration registers if WPKEY corresponds to 0x534841 (“SHA” in ASCII).

Downloaded from Arrow.com.

53.5.12 SHA Write Protection Status Register

Name: SHA_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS				SWETYP[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
					SWE	SEQE	CGD	WPVS
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – ECLASS Software Error Class (cleared on read)
0 (WARNING): An abnormal access that does not affect system functionality
1 (ERROR): An access is performed into key, input data, control registers while the SHA is performing an encryption/decryption or a start is request by software or DMA while the key is not fully configured.

Bits 27:24 – SWETYP[3:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	A write-only register has been read (Warning).
1	WRITE_RO	SHA is enabled and a write access has been performed on a read-only register (Warning).
2	UNDEF_RW	Access to an undefined address (Warning).
3	CTRL_START	SHA is locked and a start command with SHA_CR.START has been performed.
4	AUTO_START	SHA is locked and a tentative automatic start has been performed by writing input data registers (SHA_MR.SMOD>0).
5	BAD_START	SHA is not locked and a start command with SHA_CR.START has been performed whereas Start mode is automatic (SHA_MR.SMOD>0)

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS=1, WPVSRC indicates the register address offset at which a write access has been attempted.
When WPVS=0 and SWE=1, WPVSRC reports the address of the incorrect software access. As soon as WPVS=1, WPVSRC returns the address of the write-protected violation.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of SHA_WPSR.
1	A software error has occurred since the last read of SHA_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Downloaded from Arrow.com.

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of SHA_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of SHA_WPSR. This flag can only be set under abnormal operating conditions.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	The clock monitoring circuitry has not been corrupted since the last read of SHA_WPSR. Under normal operating conditions, this bit is always cleared.
1	The clock monitoring circuitry has been corrupted since the last read of SHA_WPSR. This flag can only be set in case of an abnormal clock signal waveform (glitch).

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protect violation has occurred since the last read of SHA_WPSR.
1	A write protect violation has occurred since the last read of SHA_WPSR. The address offset of the violated register is reported into field WPVSRC.

Downloaded from Arrow.com.

54. Triple Data Encryption Standard (TDES)

54.1 Description

The Triple Data Encryption Standard (TDES) is compliant with the American FIPS (Federal Information Processing Standard) Publication 46-3 specification.

The TDES supports the four different confidentiality modes of operation (ECB, CBC, OFB and CFB), specified in the FIPS (Federal Information Processing Standard) Publication 81 and is compatible with the Peripheral Data Controller channels for all of these modes, minimizing processor intervention for large buffer transfers.

The TDES key can be either loaded by the software or loaded in an invisible manner from the software.

The software can write up to three 64-bit keys, each stored in two 32-bit write-only registers, i.e., Key x Word registers, TDES_KEYxWR0 and TDES_KEYxWR1. For a software-invisible key transfer, the private key bus accesses the private key internal register from the TRNG. The PKRS bit in the Mode register selects either TDES_KEYxWR0/TDES_KEYxWR1 or the private key internal register.

The input data (and initialization vector for some modes) are stored in two corresponding 32-bit write-only registers:

- Input Data registers, TDES_IDATAR0 and TDES_IDATAR1
- Initialization Vector registers, TDES_IVR0 and TDES_IVR1

As soon as the initialization vector, the input data and the keys are configured, the encryption/decryption process may be started. Then the encrypted/decrypted data is ready to be read out on the two 32-bit Output Data registers (TDES_ODATARx) or through the DMA channels.

54.2 Embedded Characteristics

- Supports Single Data Encryption Standard (DES) and Triple Data Encryption Standard (TDES)
- Compliant with FIPS Publication 46-3, Data Encryption Standard (DES)
- 64-bit Cryptographic Key for TDES
- Two-key or Three-key Algorithms for TDES
- 18 Clock Cycles Encryption/Decryption Processing Time for DES
- 50 Clock Cycles Encryption/Decryption Processing Time for TDES
- Supports eXtended Tiny Encryption Algorithm (XTEA)
- 128-bit key for XTEA and Programmable Round Number up to 64
- Supports the Four Standard Modes of Operation specified in the FIPS Publication 81, DES Modes of Operation
  - Electronic Code Book (ECB)
  - Cipher Block Chaining (CBC)
  - Cipher Feedback (CFB)
  - Output Feedback (OFB)
- 8-, 16-, 32- and 64-bit Data Sizes Possible in CFB Mode
- Last Output Data Mode Allowing Optimized Message (Data) Authentication Code (MAC) Generation
- Abnormal Software Access Reports and Automatic Lock
- Abnormal Internal Sequence Detection and Automatic Lock
- Register Write Protection
- Temporary Secured Storage for Keys
- Private Key Bus Access to the Private Key Internal Register Not Readable from any Peripheral or Software
- Connection to DMA Optimizes Data Transfers for all Operating Modes

Downloaded from Arrow.com.

54.3 Product Dependencies

54.3.1 Power Management

The TDES may be clocked through the Power Management Controller (PMC), so the programmer must first configure the PMC to enable the TDES clock.

54.3.2 Interrupt Sources

The TDES interface has an interrupt line connected to the Interrupt Controller. In order to handle interrupts, the Interrupt Controller must be programmed before configuring the TDES.

54.4 Functional Description

The Data Encryption Standard (DES) and the Triple Data Encryption Algorithm (TDES) specify FIPS-approved cryptographic algorithms that can be used to protect electronic data. TDES_MR.TDES is used to select either the single DES or the Triple DES mode.

Encryption (enciphering) converts data to an unintelligible form called ciphertext. Decrypting (deciphering) the ciphertext converts the data back into its original form, called plaintext. TDES_MR.CIPHER is used to choose between encryption and decryption.

A DES is capable of using cryptographic keys of 64 bits to encrypt and decrypt data in blocks of 64 bits. This 64-bit key is defined in the Key 1 registers (TDES_KEY1WRx or private key internal register, only writable from the private key bus).

A TDES key consists of three DES keys, which is also referred to as a key bundle. These three 64-bit keys are defined, respectively, in the Key 1, 2 and 3 Registers (TDES_KEY1WRy, TDES_KEY2WRy and TDES_KEY3WRy or the private key internal register). In Triple DES mode (TDESMOD = 1 in TDES_MR), TDES_MR.KEYMOD is used to choose between a two- and a three-key algorithm, as summarized in the table below.

Table 54-1. TDES Algorithms Summary

Algorithm	Mode	Data Processing Sequence Steps		
		First	Second	Third
Three-key	Encryption	Encryption with Key 1	Decryption with Key 2	Encryption with Key 3
	Decryption	Decryption with Key 3	Encryption with Key 2	Decryption with Key 1
Two-key	Encryption	Encryption with Key 1	Decryption with Key 2	Encryption with Key 1
	Decryption	Decryption with Key 1	Encryption with Key 2	Decryption with Key 1

The input to the encryption processes of the CBC, CFB, and OFB modes includes, in addition to the plaintext, a 64-bit data block called the initialization vector (IV), which must be set in TDES_IVRx. The initialization vector is used in an initial step in the encryption of a message and in the corresponding decryption of the message.

The XTEA algorithm can be used instead of DES/TDES by configuring TDES_MR.TDESMOD with the appropriate value 0x2. An XTEA key consists of a 128-bit key. They are defined in the Key 1 and 2 Registers.

The number of rounds of XTEA is defined in TDES_XTEA_RNDR and can be programmed up to 64 (1 round = 2 Feistel network rounds).

All the start and operating modes of the TDES algorithm can be applied to the XTEA algorithm.

54.4.1 Operating Modes

The TDES supports the following operating modes:

- ECB—Electronic Code Book
- CBC—Cipher Block Chaining
- OFB—Output Feedback

Downloaded from Arrow.com.

- CFB—Cipher Feedback
  - CFB8 (CFB where the length of the data segment is 8 bits)
  - CFB16 (CFB where the length of the data segment is 16 bits)
  - CFB32 (CFB where the length of the data segment is 32 bits)
  - CFB64 (CFB where the length of the data segment is 64 bits)

The data pre-processing, post-processing and data chaining for each mode are automatically performed. Refer to the FIPS Publication 81 for more complete information.

These modes are selected by setting TDES_MR.OPMOD.

In CFB mode, four data sizes are possible (8, 16, 32 and 64 bits), configurable in TDES_MR.CFBS (see TDES Mode Register).

54.4.2 Temporary Secured Storage for Keys

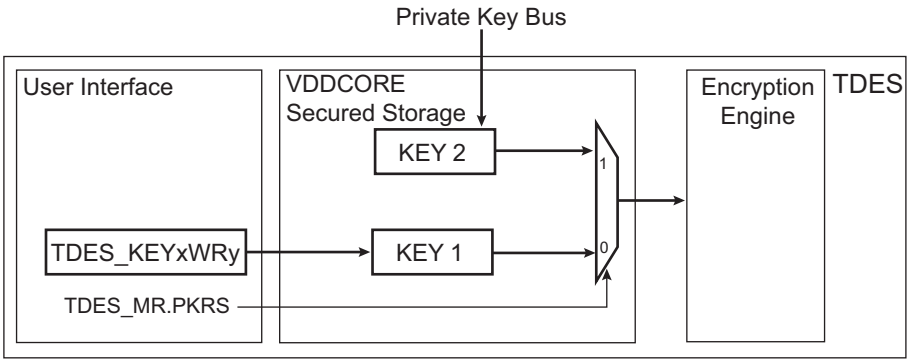
The TDES provides secure storage for two sets of three 64-bit keys. The storage is available while VDDCORE voltage is supplied.

The keys can be only written in TDES internal registers and are not readable. Moreover, the internal registers holding the keys are buried in the overall product logic area during the physical implementation.

One set of keys can be loaded by software by writing the Key Word registers (TDES_KEYxWRy).

One key can be loaded by private key bus only.

Figure 54-1. Temporary Secured Storage for Keys



54.4.3 Start Modes

TDES_MR.SMOD selects the Encryption (or Decryption) start mode.

54.4.3.1 Manual Mode

The sequence is as follows:

1. Write TDES_MR with all required fields, including but not limited to SMOD and OPMOD.
2. Write the 64-bit key(s) in TDES_KEYxWRy or the private key internal register, depending on whether one, two or three keys are required.
3. Write the initialization vector (or counter) in TDES_IVRx.
   Note: TDES_IVRx concern all modes except ECB.
4. Set DATRDY (Data Ready) in the TDES Interrupt Enable register (TDES_IER), depending on whether an interrupt is required or not at the end of processing.
5. Write the data to be encrypted/decrypted in the authorized TDES_IDATARx (see the table below).
   Note: In 32-, 16- and 8-bit CFB modes, writing to TDES_IDATAR1 is not allowed and may lead to processing errors.
6. Set the START bit in the TDES Control Register (TDES_CR) to begin the encryption or decryption process.
7. When the processing completes, DATRDY in the TDES Interrupt Status register (TDES_ISR) rises. If an interrupt has been enabled by setting TDES_IER.DATRDY, the interrupt line of the TDES is activated.
8. When the software reads a TDES_ODATARx, TDES_IER.DATRDY is automatically cleared.

Downloaded from Arrow.com.

Table 54-2. Authorized Input Data Registers

Operating Mode	Input Data Registers to Write
ECB	All
CBC	All
OFB	All
CFB 64-bit	All
CFB 32-bit	TDES_IDATAR0
CFB 16-bit	TDES_IDATAR0
CFB 8-bit	TDES_IDATAR0

54.4.3.2 Auto Mode

The Auto Mode is similar to the Manual Mode, except that as soon as the correct number of TDES_IDATARx is written, processing is automatically started without any action in TDES_CR.

54.4.3.3 DMA Mode

The DMA Controller can be used in association with the TDES to perform an encryption/decryption of a buffer without any action by the software during processing.

TDES_MR.SMOD must be set to 2 and the DMA must be configured with non-incremental addresses.

For all operating modes except CBC-MAC (TDES_MR.LOD=1), 2 DMA channels must be programmed (transmit and receive). In CBC-MAC, only 1 transmit channel must be programmed.

The start address of any transfer descriptor must be set in TDES_IDATAR0.

The DMA chunk size configuration depends on the TDES mode of operation and is listed in the table below.

When writing data to TDES with the first DMA channel, data will be fetched from a memory buffer (source data). It is recommended to configure the size of source data to “words” even for CFB modes. On the contrary, the destination data size depends on the mode of operation. When reading data from the TDES with the second DMA channel, the source data is the data read from TDES and data destination is the memory buffer. In this case, source data size depends on the TDES mode of operation and is listed in the table below.

Table 54-3. DMA Data Transfer Type for the Different Operating Modes

Operating Mode	Chunk Size	Destination/Source Data Transfer Type
ECB	1	Word
CBC	1	Word
OFB	1	Word
CFB 64-bit	1	Word
CFB 32-bit	1	Word
CFB 16-bit	1	Half-word
CFB 8-bit	1	Byte

54.4.4 Last Output Data Mode (CBC-MAC)

This mode is used to generate cryptographic checksums on data (MAC) using a CBC-MAC or a CFB encryption algorithm (refer to *FIPS Publication 81 Appendix F*).

The CMAC algorithm is a variant of CBC-MAC with post-processing requiring one-block encryption in ECB mode. Thus CBC-MAC is useful to accelerate CMAC.

Downloaded from Arrow.com.

After each end of encryption/decryption, the output data is available either on the output data registers for Manual and Auto modes or at the address specified in the receive buffer pointer for DMA mode (See Last Output Data Mode Behavior versus Start Modes).

TDES_MR.LOD can be used to retrieve only the last data of several encryption/decryption processes.

This data is only available in TDES_ODATARx.

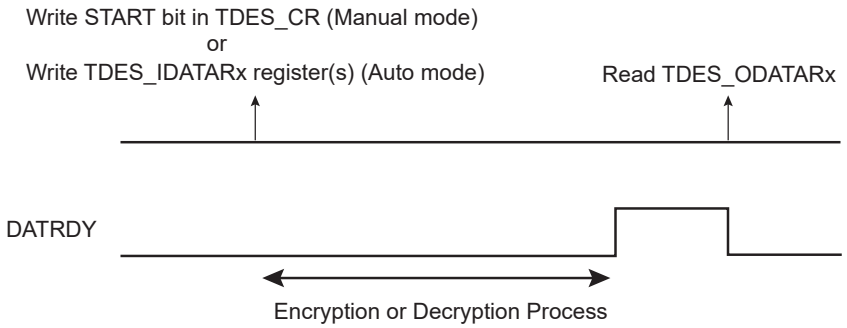
Therefore, there is no need to define a read buffer in DMA mode.

54.4.4.1 Manual and Auto Modes

54.4.4.1.1 TDES_MR.LOD = 0

The DATRDY flag is cleared when at least one TDES_ODATARx is read. See the figure below.

Figure 54-2. Manual and Auto Modes with LOD = 0

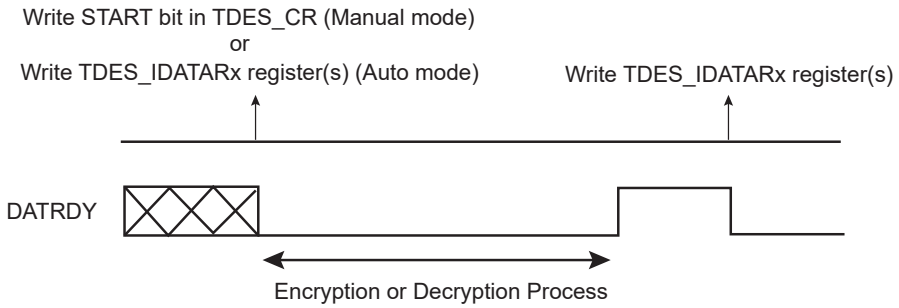


If the user does not want to read TDES_ODATARx between each encryption/decryption, the DATRDY flag will not be cleared. If the DATRDY flag is not cleared, the user will not be informed of the end of the encryptions/decryptions that follow.

54.4.4.1.2 TDES_MR.LOD = 1

The DATRDY flag is cleared when at least one TDES_IDATARx is written, before the start of a new transfer. See the figure below. No further TDES_ODATARx reads are necessary between consecutive encryptions/decryptions.

Figure 54-3. Manual and Auto Modes with LOD = 1



54.4.4.2 DMA Mode

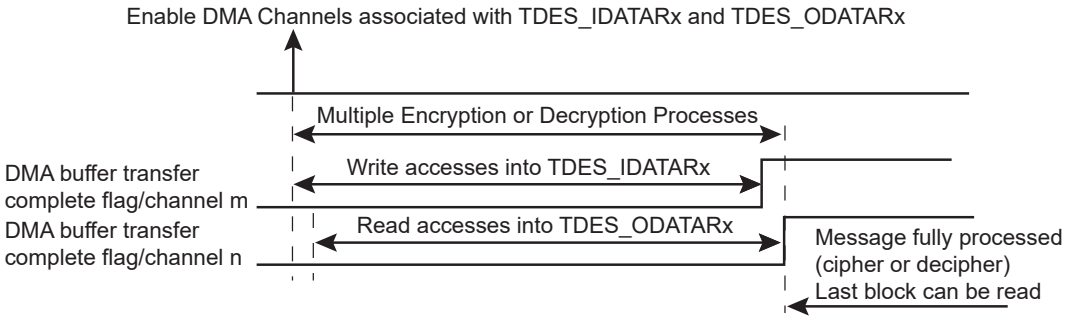
54.4.4.2.1 TDES_MR.LOD = 0

This mode may be used for all TDES operating modes except CBC-MAC where LOD = 1 mode is recommended.

The end of the encryption/decryption is indicated by the end of DMA transfer associated to TDES_ODATARx (see the figure below). Two DMA channels are required: one for writing message blocks to TDES_IDATARx and one to obtain the result from TDES_ODATARx.

Downloaded from Arrow.com.

Figure 54-4. DMA Transfer with LOD = 0



54.4.4.2.2 TDES_MR.LOD = 1

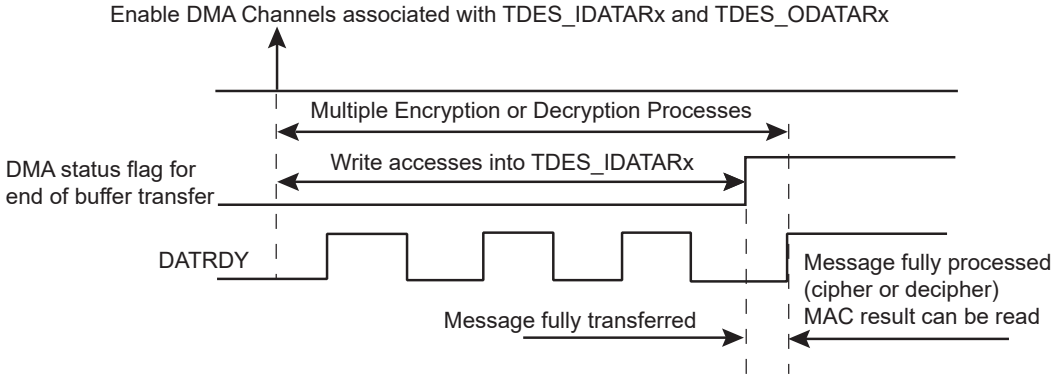
This mode is optimized to process the TDES CBC-MAC operating mode.

The user must first wait for the DMA buffer transfer complete flag, then for the flag DATRDY to rise to ensure that the encryption/decryption is completed (see the figure below).

The DMA receive channel must not be used. Prior to reading the CBC-MAC result, TDES_MR.SMOD must be written to 0. To restart a CBC-MAC on a new buffer, TDES_MR.SMOD must be written to 2.

The output data is only available on TDES_ODATARx.

Figure 54-5. DMA Transfer with LOD = 1



The table below summarizes the different cases.

Table 54-4. Last Output Data Mode Behavior versus Start Modes

Sequence	Manual and Auto Modes		DMA Transfer	
	LOD = 0	LOD = 1	LOD = 0	LOD = 1
DATRDY Flag Clearing Condition (1)	At least one TDES_ODATARx must be read	At least one TDES_IDATARx must be written	Not used	Managed by the DMA
End of Encryption/ Decryption	DATRDY	DATRDY	2 DMA Buffer transfer complete flags (channel m and channel n)	DMA buffer transfer complete flag, then TDES DATRDY flag
Encrypted/ Decrypted Data Result Location	In TDES_ODATARx	In TDES_ODATARx	Not available	In TDES_ODATARx

Note: Depending on the mode, there are other ways of clearing the DATRDY flag. See TDES Interrupt Status Register.

Downloaded from Arrow.com.

WARNING In DMA mode, reading to TDES_ODATARx before the last data transfer may lead to unpredictable results.

54.4.5 Security Features

54.4.5.1 Private Key Bus

The TDES provides secure key transfer that requires a transfer command only, thus avoiding any manipulation of the key by software.

The TDES features a set of private key internal registers that can be accessed only through the dedicated private key bus from the TRNG.

The private key internal registers cannot be read from any peripheral or from software.

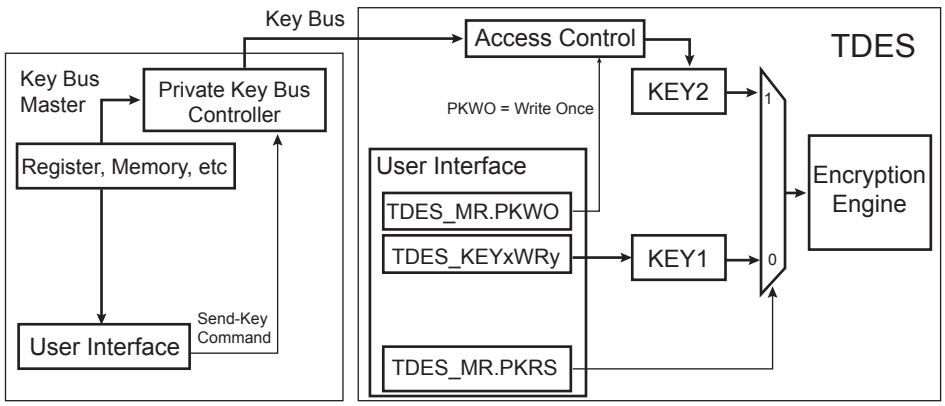
The TDES key used by the encryption/decryption engine is either the private key internal registers content or the internal key registers loaded via the TDES_KEYxWRy.

To select the private key internal registers as the source of the TDES key, TDES_MR.PKRS must be written to '1'.

To write the private key internal registers, the software must:

1. Write a '1' in TDES_MR.PKRS.
2. Trigger the key transfer over the private key bus from the KEY_BUS_MASTERS key bus master.
3. Wait for completion of the transfer signaled in the KEY_BUS_MASTERS status register.
4. Check for any access violation in TDES_WPSR.PKRPVS.

Figure 54-6. Key Selection



While TDES_MR.PKWO=0, it is possible to write the private key internal registers as many times as required.

As soon as the bit TDES_MR.PKWO=1, the next write sequence on private key internal registers is the last one. Any additional write sequence in the private key internal registers has no effect, thus providing write-protection of these registers. A hardware reset is the only way to exit from the write-protected state.

54.4.5.2 Unspecified Register Access Detection

When an unspecified register access occurs, TDES_ISR.URAD is set. Its source is then reported in TDES_ISR.URAT. Only the last unspecified register access is available through TDES_ISR.URAT.

Several kinds of unspecified register accesses can occur:

- TDES_IDATARx written during the data processing in DMA mode
- TDES_ODATARx read during the data processing
- TDES_MR written during the data processing
- Write-only register read access

URAD and URAT can only be reset by TDES_CR.SWRST.

Downloaded from Arrow.com.

54.4.5.3 Clearing Key on Tamper Event

On a tamper detection event on WKUP1..8 pins, an immediate clear of the key (internal registers) can be performed if TDES_MR.TAMPCLR=1. For configuration details, refer to section Real-Time Clock (RTC).

54.4.5.4 Register Write Protection

To prevent any single software error from corrupting TDES behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the TDES Write Protection Mode Register (TDES_WPMR).

If a write access to the protected registers is detected, the WPVS flag in the TDES Write Protection Status Register (TDES_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS flag is automatically cleared by reading TDES_WPSR.

The following register can be write-protected when WPEN is set:

- TDES Mode Register
- TDES Key 1 Word Register x
- TDES Key 2 Word Register x
- TDES Key 3 Word Register x
- TDES Initialization Vector Register x
- TDES XTEA Rounds Register

The following registers can be write-protected when WPITEN is set:

- TDES Interrupt Enable Register
- TDES Interrupt Disable Register

The following register can be write-protected when WPCREN is set:

- TDES Control Register

54.4.5.5 Security and Safety Analysis and Reports

Several types of checks are performed when the TDES is enabled.

The peripheral clock of the TDES is monitored by specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the TDES. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the flag TDES_WPSR.CGD is set, an abnormal condition occurred on the peripheral clock. This flag is not set under normal operating conditions.

The internal sequencer of the TDES is also monitored and if an abnormal state is detected, the flag TDES_WPSR.SEQE is set. This flag is not set under normal operating conditions.

The software accesses to the TDES are monitored and if an incorrect access is performed, the flag TDES_WPSR.SWE is set. The type of incorrect/abnormal software access is reported in the TDES_WPSR.SWETYP field (see TDES Write Protection Status Register for details). e.g., writing the TDES_ODATARx is an error, as well as reading the TDES_IDATARx, when the TDES_ISR.DATRDY flag is cleared. TDES_WPSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The flags CGD, SEQE, SWE and WPVS are automatically cleared when TDES_WPSR is read.

If one of these flags is set, the flag TDES_ISR.SECE is set and can trigger an interrupt if the TDES_IMR.SECE bit is ‘1’. SECE is cleared by reading TDES_ISR.

It is possible to configure an action to be performed by the TDES as soon as an abnormal event detection occurs. If the field TDES_WPMR.ACTION is greater than 0, either a lock is performed or a lock and immediate clear of TDES_KEYxWRy. If a lock is performed, the current processing is ended normally but any new processing is not performed regardless of the start mode of operation (see TDES_MR.SMOD).

A locked state of the TDES is unlocked as follows:

1. Read the TDES_WPSR.
2. Disable the source of tamper if the tamper is enabled to perform a clear of the key.
3. Write a ‘1’ to TDES_CR.UNLOCK.

Downloaded from Arrow.com.

It is possible to select the type of event that will lock the TDES in case of abnormal event detection. See TDES_WPMR.ACTION for details.

If the TDES_MR.TMPCLR=1 and the tamper pin is active, the TDES is locked.

Downloaded from Arrow.com.

54.5 Register Summary

Offset	Name	Bit Pos.								
0x00	TDES_CR	31:24								UNLOCK
		23:16								
		15:8								SWRST
		7:0								START
0x04	TDES_MR	31:24	TAMPCLR							
		23:16							CFBS[1:0]	
		15:8	LOD		OPMOD[1:0]				SMOD[1:0]	
		7:0	PKRS	PKWO		KEYMOD		TDESMOD[1:0]		CIPHER
0x08 ... 0x0F	Reserved									
0x10	TDES_IER	31:24								
		23:16								SECE
		15:8								URAD
		7:0								DATRDY
0x14	TDES_IDR	31:24								
		23:16								SECE
		15:8								URAD
		7:0								DATRDY
0x18	TDES_IMR	31:24								
		23:16								SECE
		15:8								URAD
		7:0								DATRDY
0x1C	TDES_ISR	31:24								
		23:16								SECE
		15:8			URAT[1:0]					URAD
		7:0								DATRDY
0x20	TDES_KEY1WR0	31:24	KEY1W[31:24]							
		23:16	KEY1W[23:16]							
		15:8	KEY1W[15:8]							
		7:0	KEY1W[7:0]							
0x24	TDES_KEY1WR1	31:24	KEY1W[31:24]							
		23:16	KEY1W[23:16]							
		15:8	KEY1W[15:8]							
		7:0	KEY1W[7:0]							
0x28	TDES_KEY2WR0	31:24	KEY2W[31:24]							
		23:16	KEY2W[23:16]							
		15:8	KEY2W[15:8]							
		7:0	KEY2W[7:0]							
0x2C	TDES_KEY2WR1	31:24	KEY2W[31:24]							
		23:16	KEY2W[23:16]							
		15:8	KEY2W[15:8]							
		7:0	KEY2W[7:0]							
0x30	TDES_KEY3WR0	31:24	KEY3W[31:24]							
		23:16	KEY3W[23:16]							
		15:8	KEY3W[15:8]							
		7:0	KEY3W[7:0]							
0x34	TDES_KEY3WR1	31:24	KEY3W[31:24]							
		23:16	KEY3W[23:16]							
		15:8	KEY3W[15:8]							
		7:0	KEY3W[7:0]							
0x38 ... 0x3F	Reserved									

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.									
0x40	TDES_IDATAR0	31:24	IDATA[31:24]								
		23:16	IDATA[23:16]								
		15:8	IDATA[15:8]								
		7:0	IDATA[7:0]								
0x44	TDES_IDATAR1	31:24	IDATA[31:24]								
		23:16	IDATA[23:16]								
		15:8	IDATA[15:8]								
		7:0	IDATA[7:0]								
0x48 ... 0x4F	Reserved										
0x50	TDES_ODATAR0	31:24	ODATA[31:24]								
		23:16	ODATA[23:16]								
		15:8	ODATA[15:8]								
		7:0	ODATA[7:0]								
0x54	TDES_ODATAR1	31:24	ODATA[31:24]								
		23:16	ODATA[23:16]								
		15:8	ODATA[15:8]								
		7:0	ODATA[7:0]								
0x58 ... 0x5F	Reserved										
0x60	TDES_IVR0	31:24	IV[31:24]								
		23:16	IV[23:16]								
		15:8	IV[15:8]								
		7:0	IV[7:0]								
0x64	TDES_IVR1	31:24	IV[31:24]								
		23:16	IV[23:16]								
		15:8	IV[15:8]								
		7:0	IV[7:0]								
0x68 ... 0x6F	Reserved										
0x70	TDES_XTEA_RNDR	31:24									
		23:16									
		15:8									
		7:0			XTEA_RNDS[5:0]						
0x74 ... 0xE3	Reserved										
0xE4	TDES_WPMR	31:24	WPKEY[23:16]								
		23:16	WPKEY[15:8]								
		15:8	WPKEY[7:0]								
		7:0	ACTION[2:0]			FIRSTE		WPCREN	WPITEN	WPEN	
0xE8	TDES_WPSR	31:24	ECLASS				SWETYP[3:0]				
		23:16	WPVSRC[15:8]								
		15:8	WPVSRC[7:0]								
		7:0				PKRPVS	SWE	SEQE	CGD	WPVS	

Downloaded from Arrow.com.

54.5.1 TDES Control Register

Name: TDES_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the TDES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								UNLOCK
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								SWRST
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								START
Access								W
Reset								–

Bit 24 – UNLOCK Unlock Processing

Value	Description
0	No effect.
1	Unlocks the processing in case of abnormal event detection if TDES_WPMR.ACTION > 0.

Bit 8 – SWRST Software Reset

Value	Description
0	No effect
1	Resets the TDES. A software-triggered reset of the TDES interface is performed.

Bit 0 – START Start Processing

Value	Description
0	No effect
1	Starts Manual encryption/decryption process.

Downloaded from Arrow.com.

54.5.2 TDES Mode Register

Name: TDES_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TDES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	TAMPCLR							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
							CFBS[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	LOD		OPMOD[1:0]				SMOD[1:0]	
Access	R/W		R/W	R/W			R/W	R/W
Reset	0		0	0			0	0

Bit	7	6	5	4	3	2	1	0
	PKRS	PKWO		KEYMOD		TDESMOD[1:0]		CIPHER
Access	R/W	R/W		R/W		R/W	R/W	R/W
Reset	0	0		0		0	0	0

Bit 31 – TAMPCLR Tamper Pin Clear Key Enable

Value	Description
0	A tamper detection event has no effect on TDES_KEYxWRy.
1	A tamper detection event immediately clears TDES_KEYxWRy.

Bits 17:16 – CFBS[1:0] Cipher Feedback Data Size

Value	Name	Description
0	SIZE_64BIT	64-bit
1	SIZE_32BIT	32-bit
2	SIZE_16BIT	16-bit
3	SIZE_8BIT	8-bit

Bit 15 – LOD Last Output Data Mode

WARNING In DMA mode, reading to TDES_ODATARx before the last data encryption/decryption process may lead to unpredictable result.

Value	Description
0	No effect. After each end of encryption/decryption, the output data is available either on TDES_ODATARx (Manual and Auto modes) . In Manual and Auto modes, the DATRDY flag is cleared when at least one of the TDES_ODATARx is read.

Downloaded from Arrow.com.

Value	Description
1	The DATRDY flag is cleared when at least one of the Input Data Registers is written. No further TDES_ODATARx reads are necessary between consecutive encryptions/decryptions (see Last Output Data Mode).

Bits 13:12 – OPMOD[1:0] Operating Mode
For CBC-MAC operating mode, set OPMOD to CBC and LOD to 1.

Value	Name	Description
0	ECB	Electronic Code Book mode
1	CBC	Cipher Block Chaining mode
2	OFB	Output Feedback mode
3	CFB	Cipher Feedback mode

Bits 9:8 – SMOD[1:0] Start Mode
If a DMA transfer is used, 0x2 must be configured. See DMA Mode for more details.

Value	Name	Description
0	MANUAL_START	Manual Mode
1	AUTO_START	Auto Mode
2	IDATAR0_START	TDES_IDATAR0 accesses only Auto Mode

Bit 7 – PKRS Private Key Internal Register Select

Value	Description
0	The keys used by the TDES are in the TDES_KEY1WRx, TDES_KEY2WRx and TDES_KEY3WRx registers.
1	The keys used by the TDES are the in the private key internal register written through the private key bus.

Bit 6 – PKWO Private Key Write Once

Value	Description
0	The private key internal register can be written multiple times through the private key bus.
1	The private key internal register can be written only once through the private key bus until hardware reset.

Bit 4 – KEYMOD Key Mode

Value	Description
0	Three-key algorithm is selected.
1	Two-key algorithm is selected. There is no need to write TDES_KEY3WRy (or private key internal register with more than 128 bits).

Bits 2:1 – TDESMOD[1:0] ALGORITHM Mode
Values which are not listed in the table must be considered as “reserved”.

Value	Name	Description
0	SINGLE_DES	Single DES processing using TDES_KEY1WRy.
1	TRIPLE_DES	Triple DES processing using TDES_KEY1WRy, TDES_KEY2WRy and TDES_KEY3WRy .
2	XTEA	XTEA processing using TDES_KEY1WRy and TDES_KEY2WRy.

Bit 0 – CIPHER Processing Mode

Value	Name	Description
0	DECRYPT	Decrypts data.
1	ENCRYPT	Encrypts data.

Downloaded from Arrow.com.

54.5.3 TDES Interrupt Enable Register

Name: TDES_IER
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TDES Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
Access								W
Reset								–

Bit 16 – SECE Security and/or Safety Event Interrupt Enable

Bit 8 – URAD Unspecified Register Access Detection Interrupt Enable

Bit 0 – DATRDY Data Ready Interrupt Enable

Downloaded from Arrow.com.

54.5.4 TDES Interrupt Disable Register

Name: TDES_IDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TDES Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								SECE
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								W
Reset								–

Bit 16 – SECE Security and/or Safety Event Interrupt Disable

Bit 8 – URAD Unspecified Register Access Detection Interrupt Disable

Bit 0 – DATRDY Data Ready Interrupt Disable

Downloaded from Arrow.com.

54.5.5 TDES Interrupt Mask Register

Name: TDES_IMR
Offset: 0x18
Reset: 0x000000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								SECE
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
								URAD
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								R
Reset								0

Bit 16 – SECE Security and/or Safety Event Interrupt Mask

Bit 8 – URAD Unspecified Register Access Detection Interrupt Mask

Bit 0 – DATRDY Data Ready Interrupt Mask

Downloaded from Arrow.com.

54.5.6 TDES Interrupt Status Register

Name: TDES_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								SECE
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
			URAT[1:0]					URAD
Access			R	R				R
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								R
Reset								0

Bit 16 – SECE Security and/or Safety Event Interrupt Mask

Value	Description
0	There is no security report in TDES_WPSR.
1	One security flag is set in TDES_WPSR.

Bits 13:12 – URAT[1:0] Unspecified Register Access (cleared by setting bit TDES_CR.SWRST)
Only the last Unspecified Register Access Type is available through the URAT field.

Value	Name	Description
0	IDR_WR_PROCESSING	TDES_IDATAR written during data processing when SMOD = 0x2 mode.
1	ODR_RD_PROCESSING	TDES_ODATAR read during data processing.
2	MR_WR_PROCESSING	TDES_MR written during data processing.
3	WOR_RD_ACCESS	Write-only register read access.

Bit 8 – URAD Unspecified Register Access Detection Status (cleared by setting TDES_CR.SWRST)

Value	Description
0	No unspecified register access has been detected since the last write of TDES_CR.SWRST.
1	At least one unspecified register access has been detected since the last write of TDES_CR.SWRST.

Bit 0 – DATRDY Data Ready (cleared by setting TDES_CR.START or TDES_CR.SWRST, or by reading TDES_ODATARx)
If TDES_MR.LOD = 1: In Manual and Auto modes, the DATRDY flag can also be cleared by writing at least one TDES_IDATARx.

Value	Description
0	Output data is not valid.
1	Encryption or decryption process is completed.

Downloaded from Arrow.com.

54.5.7 TDES Key 1 Word Register y

Name: TDES_KEY1WRy
Offset: 0x20 + y*0x04 [y=0..1]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the TDES Write Protection Mode Register.

Immediately cleared on tamper detection event if TDES_MR.TAMPCLR=1.

Bit	31	30	29	28	27	26	25	24
	KEY1W[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	KEY1W[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	KEY1W[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	KEY1W[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – KEY1W[31:0] Key 1 Word
The two 32-bit Key 1 Word registers are used to set the 64-bit cryptographic key used for encryption/decryption.
TDES_KEY1WR0.KEY1W refers to the first word of the key and TDES_KEY1WR1.KEY1W to the last one.
These registers are write-only to prevent the key from being read by another application.
In XTEA mode, the key is defined on 128 bits. These registers contain the 64 LSB bits of the encryption/decryption key.
TDES_KEY1WRy registers are not used if the private key internal register is selected instead by writing a 1 to TDES_MR.PKRS.

Downloaded from Arrow.com.

54.5.8 TDES Key 2 Word Register y

Name: TDES_KEY2WRy
Offset: 0x28 + y*0x04 [y=0..1]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the TDES Write Protection Mode Register.

Immediately cleared on tamper detection event if TDES_MR.TAMPCLR=1.

Bit	31	30	29	28	27	26	25	24
	KEY2W[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	KEY2W[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	KEY2W[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	KEY2W[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – KEY2W[31:0] Key 2 Word
The two 32-bit Key 2 Word registers are used to set the 64-bit cryptographic key used for encryption/decryption.
TDES_KEY2WR0.KEY2W refers to the first word of the key and TDES_KEY2W1.KEY2W to the last one.
These registers are write-only to prevent the key from being read by another application.
TDES_KEY2WRx registers are not used in DES mode.
In XTEA mode, the key is defined on 128 bits. These registers contain the 64 MSB bits of the encryption/decryption key.
TDES_KEY2WRy registers are not used if the private key internal register is selected instead by writing a 1 to TDES_MR.PKRS.

Downloaded from Arrow.com.

54.5.9 TDES Key 3 Word Register y

Name: TDES_KEY3WRy
Offset: 0x30 + y*0x04 [y=0..1]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the TDES Write Protection Mode Register.

Immediately cleared on tamper detection event if TDES_MR.TAMPCLR=1.

Bit	31	30	29	28	27	26	25	24
	KEY3W[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	KEY3W[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	KEY3W[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	KEY3W[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – KEY3W[31:0] Key 3 Word
The two 32-bit Key 3 Word registers are used to set the 64-bit cryptographic key used for encryption/decryption. TDES_KEY3WR0.KEY3W refers to the first word of the key and TDES_KEY3WR1.KEY3W to the last one. These registers are write-only to prevent the key from being read by another application.
TDES_KEY3WRx registers are not used in DES mode, TDES with two-key algorithm selected and XTEA mode.
TDES_KEY3WRy registers are not used if the private key internal register is selected by writing a 1 to TDES_MR.PKRS.

Downloaded from Arrow.com.

54.5.10 TDES Input Data Register x

Name: TDES_IDATARx
Offset: 0x40 + x*0x04 [x=0..1]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	IDATA[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	IDATA[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	IDATA[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	IDATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – IDATA[31:0] Input Data
The two 32-bit TDES_IDATARx are used to set the 64-bit data block used for encryption/decryption.
TDES_IDATAR0.IDATA refers to the first word of the data to be encrypted/decrypted, and TDES_IDATAR1.IDATA to the last one.
These registers are write-only to prevent the input data from being read by another application.

Downloaded from Arrow.com.

54.5.11 TDES Output Data Register x

Name: TDES_ODATARx
Offset: 0x50 + x*0x04 [x=0..1]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ODATA[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ODATA[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ODATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ODATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ODATA[31:0] Output Data
The two 32-bit TDES_ODATARx contain the 64-bit data block which has been encrypted/decrypted.
TDES_ODATAR0.ODATA refers to the first word, TDES_ODATAR1.ODATA to the last one.

Downloaded from Arrow.com.

54.5.12 TDES Initialization Vector Register x

Name: TDES_IVRx
Offset: 0x60 + x*0x04 [x=0..1]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the TDES Write Protection Mode Register.

These registers are write-only to prevent the Initialization Vector from being read by another application.

These registers are not used for the ECB mode and must not be written.

Bit	31	30	29	28	27	26	25	24
	IV[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	IV[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	IV[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	IV[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – IV[31:0] Initialization Vector
The two 32-bit TDES_IVRx are used to set the 64-bit initialization vector data block, which is used by some modes of operation as an additional initial input.
TDES_IVR1.IV refers to the first word of the Initialization Vector, TDES_IVR2.IV to the last one.

Downloaded from Arrow.com.

54.5.13 TDES XTEA Rounds Register

Name: TDES_XTEA_RNDR
Offset: 0x70
Reset: 0x000000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TDES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			XTEA_RNDS[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:0 – XTEA_RNDS[5:0] Number of Rounds
This 6-bit field is used to define the number of complete rounds (1 complete round = 2 Feistel rounds) processed in XTEA algorithm.
The value of XTEA_RNDS has no effect if TDES_MR.TDESMOD is set to 0x0 or 0x1.
0x00 corresponds to 1 complete round, 0x01 corresponds to 2 complete rounds, etc.

Downloaded from Arrow.com.

54.5.14 TDES Write Protection Mode Register

Name: TDES_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

See Register Write Protection for the list of registers that can be write-protected.

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ACTION[2:0]			FIRSTE		WPCREN	WPITEN	WPEN
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x444553	PASSWD	Writing any other value in this field aborts the write operation of bits WPEN, WPITEN and WPCREN. Always reads as 0.

Bits 7:5 – ACTION[2:0] Action on Abnormal Event Detection

Value	Name	Description
0	REPORT_ONLY	No action (stop or clear key) is performed when one of PKRPVS, WPVS, CGD, SEQE, or SWE flags are set.
1	LOCK_PKRPVS_WPVS_SWE	If a processing is in progress when the TDES_WPSR.PKRPVS/WPVS/SWE event detection occurs, the current processing is ended normally but no other processing is started while a TDES_CR.UNLOCK command is issued.
2	LOCK_CGD_SEQE	If a processing is in progress when the TDES_WPSR.CGD/SEQE event detection occurs, the current processing is ended normally but no other processing is started while a TDES_CR.UNLOCK command is issued.
3	LOCK_ANY_EV	If a processing is in progress when the TDES_WPSR.PKRPVS/WPVS/CGD/SEQE/SWE events detection occurs, the current processing is ended normally but no other processing is started while a TDES_CR.UNLOCK command is issued.
4	CLEAR_PKRPVS_WPVS_SWE	If a processing is in progress when the TDES_WPSR.PKRPVS/WPVS/SWE events detection occurs, the current processing is ended normally but no other processing is started while a TDES_CR.UNLOCK command is issued. Moreover, TDES_KEYxWRy are immediately cleared.

Downloaded from Arrow.com.

Value	Name	Description
5	CLEAR_CGD_SEQE	If a processing is in progress when the TDES_WPSR.CGD/SEQE events detection occurs, the current processing is ended normally but no other processing is started while a TDES_CR.UNLOCK command is issued. Moreover, TDES_KEYxWRy are immediately cleared.
6	CLEAR_ANY_EV	If a processing is in progress when the TDES_WPSR.PKRPVS/ WPVS/CGD/SEQE/SWE events detection occurs, the current processing is ended normally but no other processing is started while a TDES_CR.UNLOCK command is issued. Moreover, TDES_KEYxWRy are immediately cleared.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in TDES_WPSR.WPVSRC and the last software control error type is reported in TDES_WPSR.SWETYP. The TDES_ISR.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in TDES_WPSR.WPVSRC and only the first software control error type is reported in TDES_WPSR.SWETYP. The TDES_ISR.SECE flag is set at the first error occurrence within a series.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to0x444553 (“DES” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x444553 (“DES” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x444553 (“DES” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x444553 (“DES” in ASCII).

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x444553 (“DES” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x444553 (“DES” in ASCII).

Downloaded from Arrow.com.

54.5.15 TDES Write Protection Status Register

Name: TDES_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ECLASS				SWETYP[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				PKRPVS	SWE	SEQE	CGD	WPVS
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 31 – ECLASS Software Error Class (cleared on read)

Value	Name	Description
0	WARNING	An abnormal access that does not affect system functionality.
1	ERROR	An access is performed into key, input data, control registers while the TDES is performing an encryption/decryption or a start is request by software or DMA while the key is not fully configured.

Bits 27:24 – SWETYP[3:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	A write-only register has been read (Warning).
1	WRITE_RO	TDES is enabled and a write access has been performed on a read-only register (Warning).
2	UNDEF_RW	Access to an undefined address (Warning).
3	CTRL_START	Abnormal use of TDES_CR.START command when DMA access is configured.
4	WEIRD_ACTION	A key write, init value write, output data read, Mode register write, private key bus access or XTEA round register has been performed while a current processing is in progress (abnormal).
5	INCOMPLETE_KEY	A tentative of start is required while the keys are not fully loaded into TDES_KEYxWRy.

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source (cleared on read)
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.
When WPVS=0 and SWE=1, WPVSRC reports the address of the incorrect software access. As soon as WPVS=1, WPVSRC returns the address of the write-protected violation.

Bit 4 – PKRPVS Private Key Register Protection Violation Status (cleared on read)

Value	Description
0	No private key internal register access violation has occurred since the last read of TDES_WPSR.
1	A private key internal register access violation has occurred since the last read of TDES_WPSR.

Downloaded from Arrow.com.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of TDES_WPSR.
1	A software error has occurred since the last read of TDES_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of TDES_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of TDES_WPSR. This flag is set under abnormal operating conditions.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	The clock monitoring circuitry has not been corrupted since the last read of TDES_WPSR. Under normal operating conditions, this bit is always cleared.
1	The clock monitoring circuitry has been corrupted since the last read of TDES_WPSR. This flag is set in case of abnormal clock signal waveform (glitch).

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protection violation has occurred since the last read of TDES_WPSR.
1	A write protection violation has occurred since the last read of TDES_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

55. True Random Number Generator (TRNG)

55.1 Description

The True Random Number Generator (TRNG) passes the American *NIST Special Publication 800-22 (A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications)* and the *Diehard Suite of Tests*.

The TRNG may be used as an entropy source for seeding an NIST approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3.

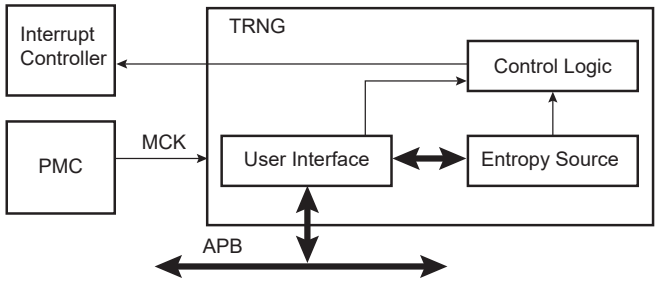
The TRNG is fully designed with digital cells, and under the specified operating conditions, external factors such as temperature, humidity, etc. affect TRNG ageing in the same manner as all other digital peripherals (CPU core, bus matrix, etc.) of the product.

55.2 Embedded Characteristics

- Passes *NIST Special Publication 800-22 Test Suite*
- Passes *Diehard Suite of Tests*
- May be Used as Entropy Source for seeding a NIST-approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3
- Provides a 32-bit Random Number Every 84 Clock Cycles
- Abnormal Software Access Reports
- Register Write Protection
- Private Key Bus Interface to generate cryptographic keys not readable from any Peripheral nor from software

55.3 Block Diagram

Figure 55-1. TRNG Block Diagram



55.4 Product Dependencies

55.4.1 Power Management

The TRNG interface may be clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the TRNG user interface clock. The user interface clock is independent from any clock that may be used in the entropy source logic circuitry. The source of entropy can be enabled before enabling the user interface clock.

55.4.2 Interrupt Sources

The TRNG interface has an interrupt line connected to the Interrupt Controller. In order to handle interrupts, the Interrupt Controller must be programmed before configuring the TRNG.

Downloaded from Arrow.com.

55.5 Functional Description

As soon as the TRNG is enabled in the Control register (TRNG_CR), the generator provides one 32-bit random value every 84 clock cycles.

A sequence of random values can be generated by the TRNG and a random value can be directly loaded through the private key bus into specific private key internal registers of the private key bus slaves (e.g. AES, etc.) with no possibility of reading these keys from the processor and software from system bus. This is done by writing the Private Key Bus Control register (TRNG_PKBCR) with the appropriate destination and length of the key to be generated.

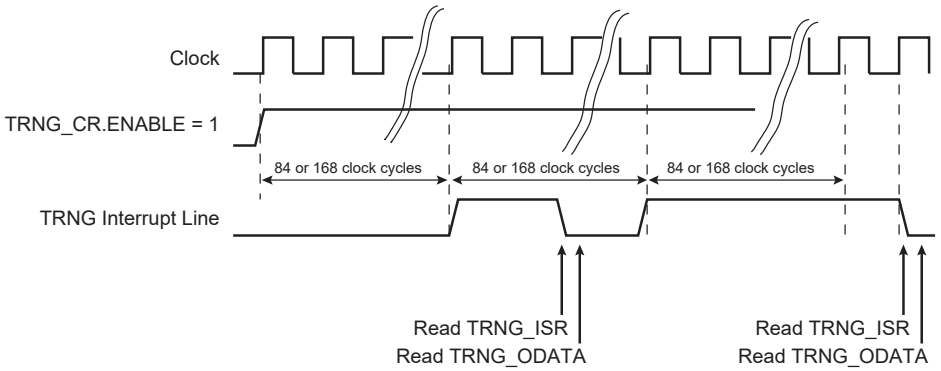
By writing a ‘1’ to the HALFR bit in the Mode register (TRNG_MR), the random values are provided every 168 cycles instead of every 84 cycles. HALFR must be written to ‘1’ when the TRNG peripheral clock frequency is above 100 MHz.

The TRNG interrupt line can be enabled in the Interrupt Enable register (TRNG_IER), and disabled in the Interrupt Disable register (TRNG_IDR). This interrupt is set when a new random value is available or when a transfer over the private key bus is complete and is cleared when the Status register (TRNG_ISR) is read. The flag TRNG_ISR.DATRDY is set when the random data is ready to be read out on the 32-bit Output Data register (TRNG_ODATA). The flag TRNG_ISR.EOTPKB is set when the transfer through the private key bus is complete.

55.5.1 Normal Operating Mode

The normal mode of operation checks that the flag in TRNG_ISR equals ‘1’ before reading TRNG_ODATA when a 32-bit random value is required by the software application.

Figure 55-2. TRNG Data Generation Sequence



55.5.2 Key Bus Operating Mode

After a write to the Private Key Bus Control register of the key bus destination slave KBSLAVE, key ID KID and key length KLENGTH, the software:

- waits for the end of transfer of the key indicated by the TRNG_ISR.EOTPKB flag being read at ‘1’, optionally after a TRNG interrupt,
- checks for any key bus access violation in the selected private key bus destination slave status register,
- uses the private key bus destination slave or launches any other private key bus transfer.

Downloaded from Arrow.com.

Figure 55-3. TRNG Private Key Bus

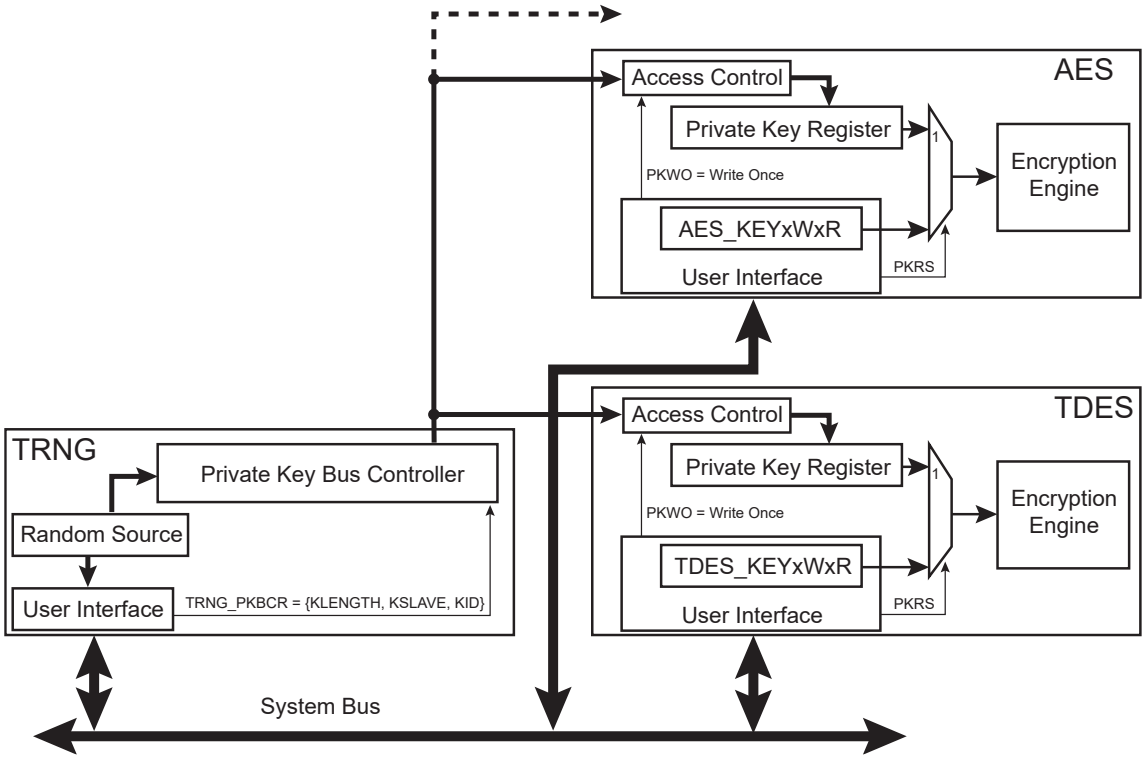
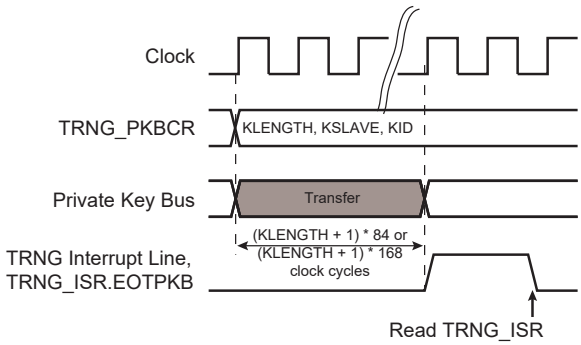


Figure 55-4. TRNG Private Key Bus Transfer



55.5.3 Register Write Protection

To prevent any single software error from corrupting TRNG behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the TRNG Write Protection Mode Register (TRNG_WPMR).

If a write access to the protected registers is detected, the WPVS flag in the TRNG Write Protection Status Register (TRNG_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS flag is automatically cleared by reading TRNG_WPSR.

The following register can be write-protected when WPEN is set:

- TRNG Mode Register

The following registers can be write-protected when WPITEN is set:

- TRNG Interrupt Enable Register
- TRNG Interrupt Disable Register

The following registers can be write-protected when WPCREN is set:

Downloaded from Arrow.com.

- TRNG Control Register
- TRNG Private Key Bus Control Register

55.5.4 Security and Safety Analysis and Reports

Several type of checks are performed when the TRNG is enabled.

The peripheral clock of the TRNG is monitored by specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the TRNG. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the flag TRNG_WPSR.CGD is set, an abnormal condition occurred on the peripheral clock. This flag is not set under normal operating conditions.

The internal sequencer of the TRNG is also monitored and if an abnormal state is detected, the flag TRNG_WPSR.SEQE is set. This flag is not set under normal operating conditions.

The software accesses to the TRNG are monitored and if an incorrect access is performed, the flag TRNG_WPSR.SWE is set. The type of incorrect/abnormal software access is reported in the TRNG_WPSR.SWETYP field (see TRNG Write Protection Status Register for details). e.g., reading the TRNG_ODATA when the TRNG is disabled is an error, as well as reading the TRNG_ODATA, when the TRNG_ISR.DATRDY flag is cleared. TRNG_WPSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The flags CGD, SEQE, SWE and WPVS are automatically cleared when TRNG_WPSR is read.

If one of these flags is set, the flag TRNG_ISR.SECE is set and can trigger an interrupt if the TRNG_IMR.SECE bit is ‘1’. SECE is cleared by reading TRNG_ISR.

Downloaded from Arrow.com.

55.6 Register Summary

Offset	Name	Bit Pos.								
0x00	TRNG_CR	31:24	WAKEY[23:16]							
		23:16	WAKEY[15:8]							
		15:8	WAKEY[7:0]							
		7:0								ENABLE
0x04	TRNG_MR	31:24								
		23:16								
		15:8								
		7:0								HALFR
0x08	TRNG_PKBCR	31:24	WAKEY[15:8]							
		23:16	WAKEY[7:0]							
		15:8	KLENGTH[7:0]							
		7:0			KSLAVE[1:0]					KID
0x0C ... 0x0F	Reserved									
0x10	TRNG_IER	31:24								
		23:16								
		15:8								
		7:0						EOTPKB	SECE	DATRDY
0x14	TRNG_IDR	31:24								
		23:16								
		15:8								
		7:0						EOTPKB	SECE	DATRDY
0x18	TRNG_IMR	31:24								
		23:16								
		15:8								
		7:0						EOTPKB	SECE	DATRDY
0x1C	TRNG_ISR	31:24								
		23:16								
		15:8								
		7:0						EOTPKB	SECE	DATRDY
0x20 ... 0x4F	Reserved									
0x50	TRNG_ODATA	31:24	ODATA[31:24]							
		23:16	ODATA[23:16]							
		15:8	ODATA[15:8]							
		7:0	ODATA[7:0]							
0x54 ... 0xE3	Reserved									
0xE4	TRNG_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0				FIRSTE		WPCREN	WPITEN	WPEN
0xE8	TRNG_WPSR	31:24	ECLASS				SWETYP[3:0]			
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS

Downloaded from Arrow.com.

55.6.1 TRNG Control Register

Name: TRNG_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the TRNG Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	WAKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	WAKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	WAKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
								ENABLE
Access								W
Reset								–

Bits 31:8 – WAKEY[23:0] Register Write Access Key

Value	Name	Description
0x524E47	PASSWD	Writing any other value in this field aborts the write operation.

Bit 0 – ENABLE Enable TRNG to Provide Random Values

Value	Description
0	Disables the TRNG if 0x524E47 (“RNG” in ASCII) is written in WAKEY field at the same time.
1	Enables the TRNG if 0x524E47 (“RNG” in ASCII) is written in WAKEY field at the same time.

Downloaded from Arrow.com.

55.6.2 TRNG Mode Register

Name: TRNG_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TRNG Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								HALFR
Access								R/W
Reset								0

Bit 0 – HALFR Half Rate Enable

Value	Description
0	Maximum stream rate provided.
1	Half maximum stream rate provided if the peripheral clock frequency is above 100 MHz.

Downloaded from Arrow.com.

55.6.3 TRNG Private Key Bus Control Register

Name: TRNG_PKBCR
Offset: 0x08
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the TRNG Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	WAKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	WAKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	KLENGTH[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
			KSLAVE[1:0]					KID
Access			W	W				W
Reset			–	–				–

Bits 31:16 – WAKEY[15:0] Register Write Access Key

Value	Name	Description
0x524B	PASSWD	Writing any other value in this field aborts the write operation.

Bits 15:8 – KLENGTH[7:0] Key Length
Length-1 in 32-bit words of the key(s) to be directly loaded from the TRNG into the private key internal registers of the private key bus slave KSLAVE.
Example: for one 64-bit key to be loaded, KLENGTH must be written to 1. For 128-bit keys, KLENGTH must be written to 3.

Bits 5:4 – KSLAVE[1:0] Key Bus Slave
Private key bus slave to be loaded from the TRNG.

Value	Name	Description
0	TDES_ID	TDES selection
1	AES_ID	AES selection
2	Reserved_ID	Reserved selection
3	OTPC_ID	OTPC selection

Bit 0 – KID Key ID (Must always be written to 0)

Downloaded from Arrow.com.

55.6.4 TRNG Interrupt Enable Register

Name: TRNG_IER
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TRNG Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						EOTPKB	SECE	DATRDY
Access						W	W	W
Reset						–	–	–

Bit 2 – EOTPKB End Of Transfer on Private Key Bus Interrupt Enable

Value	Description
0	No effect.
1	Enables the corresponding interrupt.

Bit 1 – SECE Security and/or Safety Event Interrupt Enable

Value	Description
0	No effect.
1	Enables the corresponding interrupt.

Bit 0 – DATRDY Data Ready Interrupt Enable

Value	Description
0	No effect.
1	Enables the corresponding interrupt.

Downloaded from Arrow.com.

55.6.5 TRNG Interrupt Disable Register

Name: TRNG_IDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TRNG Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						EOTPKB	SECE	DATRDY
Access						W	W	W
Reset						–	–	–

Bit 2 – EOTPKB End Of Transfer on Private Key Bus Interrupt Disable

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

Bit 1 – SECE Security and/or Safety Event Interrupt Disable

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

Bit 0 – DATRDY Data Ready Interrupt Disable

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

Downloaded from Arrow.com.

55.6.6 TRNG Interrupt Mask Register

Name: TRNG_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						EOTPKB	SECE	DATRDY
Access						R	R	R
Reset						0	0	0

Bit 2 – EOTPKB End Of Transfer on Private Key Bus Interrupt Mask

Value	Description
0	The corresponding interrupt is not enabled.
1	The corresponding interrupt is enabled.

Bit 1 – SECE Security and/or Safety Event Interrupt Mask

Value	Description
0	The corresponding interrupt is not enabled.
1	The corresponding interrupt is enabled

Bit 0 – DATRDY Data Ready Interrupt Mask

Value	Description
0	The corresponding interrupt is not enabled.
1	The corresponding interrupt is enabled.

Downloaded from Arrow.com.

55.6.7 TRNG Interrupt Status Register

Name: TRNG_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						EOTPKB	SECE	DATRDY
Access						R	R	R
Reset						0	0	0

Bit 2 – EOTPKB End Of Transfer on Private Key Bus (cleared on read)

Value	Description
0	No private key bus transfer has ended since the last read of the Interrupt Status Register.
1	The private key bus transfer has ended.

Bit 1 – SECE Security and/or Safety Event (cleared on read)

Value	Description
0	No security or safety event occurred.
1	One or more safety or security event occurred since the last read of TRNG_ISR. For details on the event, see TRNG Write Protection Status Register.

Bit 0 – DATRDY Data Ready (cleared on read)

Value	Description
0	Output data is not valid or TRNG is disabled.
1	New random value is completed since the last read of TRNG_ODATA.

Downloaded from Arrow.com.

55.6.8 TRNG Output Data Register

Name: TRNG_ODATA
Offset: 0x50
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ODATA[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ODATA[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ODATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ODATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ODATA[31:0] Output Data
The 32-bit Output Data register contains the 32-bit random data.

Downloaded from Arrow.com.

55.6.9 TRNG Write Protection Mode Register

Name: TRNG_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				FIRSTE		WPCREN	WPITEN	WPEN
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x524E47	PASSWD	Writing any other value in this field aborts the write operation of bits WPEN, WPITEN and WPCREN. Always reads as 0.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in TRNG_WPSR.WPVSRC and the last software control error type is reported in TRNG_WPSR.SWETYP. The TRNG_ISR.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in TRNG_WPSR.WPVSRC and only the first software control error type is reported in TRNG_WPSR.SWETYP. The TRNG_ISR.SECE flag is set at the first error occurrence within a series.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x524E47 (“RNG” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x524E47 (“RNG” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x524E47 (“RNG” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x524E47 (“RNG” in ASCII).

Downloaded from Arrow.com.

Bit 0 – WPEN Write Protection Enable
See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x524E47 (“RNG” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x524E47 (“RNG” in ASCII).

Downloaded from Arrow.com.

55.6.10 TRNG Write Protection Status Register

Name: TRNG_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS				SWETYP[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SWE	SEQE	CGD	WPVS
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – ECLASS Software Error Class (cleared on read)

Value	Name	Description
0	WARNING	An abnormal access that does not affect system functionality.
1	ERROR	Reading TRNG_ODATA when TRNG is disabled or used for private key bus transfer does not provide a random value. Writing to the PKB_CTRL register while a private key bus transfer is ongoing does not launch a new private key bus transfer.

Bits 27:24 – SWETYP[3:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	TRNG is enabled and a write-only register has been read (Warning).
1	WRITE_RO	TRNG is enabled and a write access has been performed on a read-only register (Warning).
2	UNDEF_RW	Access to an undefined address.
3	TRNG_DIS	The TRNG_ODATA register has been read when TRNG is disabled or used for private key bus transfer (Error).
4	PKB_BUSY	A write access to the PKB_CTRL register has been attempted during a private key bus transfer (Error).

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source (cleared on read)
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.
When WPVS=0 and SWE=1, WPVSRC reports the address of the incorrect software access. As soon as WPVS=1, WPVSRC returns the address of the write-protected violation.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of TRNG_WPSR.

Downloaded from Arrow.com.

Value	Description
1	A software error has occurred since the last read of TRNG_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of TRNG_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of TRNG_WPSR. This flag is set under abnormal operating conditions.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	No clock glitch has occurred since the last read of TRNG_WPSR. Under normal operating conditions, this bit is always cleared.
1	A clock glitch has occurred since the last read of TRNG_WPSR. This flag is set in case of abnormal clock signal waveform (glitch).

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protection violation has occurred since the last read of TRNG_WPSR.
1	A write protection violation has occurred since the last read of TRNG_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

56. Analog-to-Digital Controller (ADC)

56.1 Description

The ADC is based on a 12-bit Analog-to-Digital Converter (ADC) managed by an ADC Controller providing enhanced resolution up to 16 bits. See 56.3 Block Diagram. It also integrates a 12-to-1 analog multiplexer, making possible the analog-to-digital conversions of 12 analog lines. The conversions extend from the voltage on pin ADVREFN to the voltage carried on pin ADVREFP.

Conversion results are reported in a common register for all channels, as well as in a channel-dedicated register.

The 13-bit, 14-bit, 15-bit and 16-bit resolution modes are obtained by averaging multiple samples to decrease quantization noise. For the 13-bit mode, 4 samples are used, which gives a real sample rate of 1/4 of the actual sample frequency. For the 14-bit mode, 16 samples are used, giving a real sample rate of 1/16 of the actual sample frequency. For the 15-bit and 16-bit modes, respectively 64 and 256 samples are used, giving a real sample rate of respectively 1/64 and 1/256 of the actual sample frequency. This arrangement allows conversion speed to be traded off against for better accuracy.

The software trigger, external trigger on rising edge of the ADTRG pin or internal triggers from Timer Counter output(s) are configurable.

The comparison circuitry allows automatic detection of values below a threshold, higher than a threshold, in a given range or outside the range, thresholds and ranges being fully configurable.

The ADC Controller internal fault output is directly connected to the PWM fault input. This input can be asserted by means of comparison circuitry to immediately put the PWM output in a safe state (pure combinational path).

The ADC also integrates a Sleep mode and a conversion sequencer and connects with a DMA channel. These features reduce both power consumption and processor intervention.

This ADC has a selectable single-ended, pseudo-differential or fully differential input.

This ADC Controller includes a Resistive Touchscreen Controller. It supports 4-wire and 5-wire technologies.

56.2 Embedded Characteristics

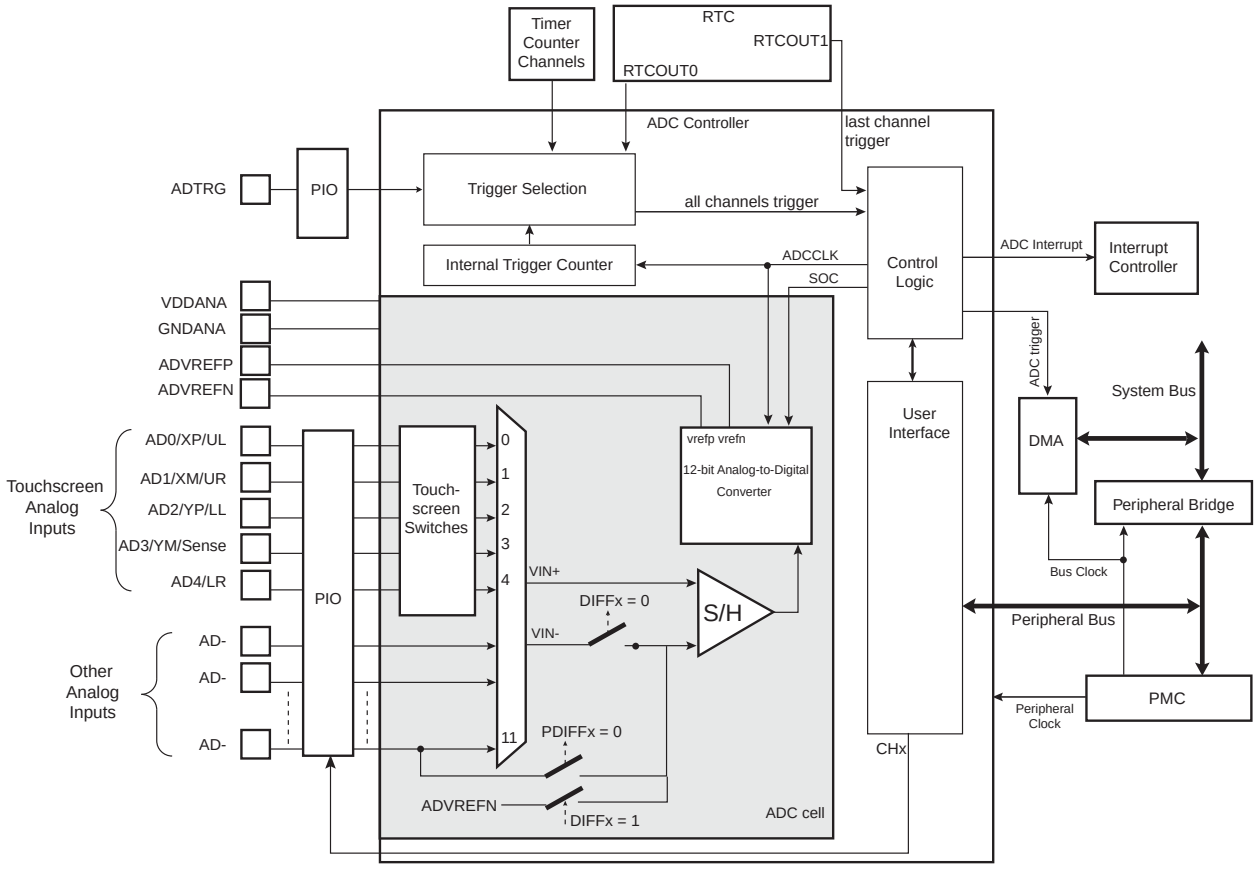
- 12-bit Resolution with Enhanced Mode up to 16 bits
- 1 MSps Conversion Rate
- Digital Averaging Function providing Enhanced Resolution Mode up to 16 bits
- Wide Range of Power Supply Operation
- Selectable Single-Ended, Pseudo-Differential or Differential Input Voltage
- Digital correction of offset and gain errors
- Resistive 4-wire and 5-wire Touchscreen Controller
  - Position and pressure measurement for 4-wire screens
  - Position measurement for 5-wire screens
  - Average of up to 8 measures for noise filtering
- Programmable Pen Detection Sensitivity
- Integrated Multiplexer Offering Up to 12 Independent Analog Inputs
- Individual Enable and Disable of Each Channel
- Hardware or Software Trigger from:
  - External trigger pin
  - Timer counter outputs (corresponding TIOA trigger)
  - ADC internal trigger counter
  - Trigger on pen contact detection
  - PWM event line

Downloaded from Arrow.com.

- Drive of PWM Fault Input
- DMA Support
- Two Sleep Modes (Automatic Wakeup on Trigger)
  - Lowest power consumption (voltage reference OFF between conversions)
  - Fast wakeup time response on trigger event (voltage reference ON between conversions)
- Channel Sequence Customizing
- Automatic Window Comparison of Converted Values
- Asynchronous Partial Wakeup (SleepWalking) on External Trigger
- Register Write Protection

56.3 Block Diagram

Figure 56-1. ADC Block Diagram



56.4 Signal Description

Table 56-1. ADC Pin Description

Pin Name	Description
VDDANA	Analog power supply
ADVREFP	Reference voltage
ADVREFN	Reference voltage
AD0–AD11	Analog input channels

Downloaded from Arrow.com.

.....continued	
Pin Name	**Description**
ADTRG	External trigger

56.5 Product Dependencies

56.5.1 Power Management

The ADC Controller is not continuously clocked. The programmer must first enable the ADC Controller peripheral clock in the Power Management Controller (PMC) before using the ADC Controller. However, if the application does not require ADC operations, the ADC Controller clock can be stopped when not needed and restarted when necessary. Configuring the ADC Controller does not require the ADC Controller clock to be enabled.

56.5.2 Interrupt Sources

The ADC interrupt line is connected on one of the internal sources of the Interrupt Controller. Using the ADC interrupt requires the interrupt controller to be programmed first.

56.5.3 I/O Lines

The digital inputs ADx and ADTRG are multiplexed with digital functions on the I/O lines.

ADx inputs are selected as inputs of the ADCC when writing a one in the corresponding ADC_CHER.CHx bit and the associated I/O is automatically turned in Analog mode.

56.5.4 Hardware Triggers

The ADC can use internal signals to start conversions. See the ADC_MR.TRGSEL field description in 56.7.2 ADC_MR for exact wiring of internal triggers.

56.5.5 Fault Output

The ADC Controller has the FAULT output connected to the FAULT input of PWM. See 56.6.18 Fault Event and section “Pulse Width Modulation Controller (PWM)”.

56.6 Functional Description

56.6.1 Analog-to-Digital Conversion

Once the programmed startup time (ADC_MR.STARTUP) has elapsed, ADC conversions are sequenced by three operating times:

- Tracking time—the time for the ADC to charge its input sampling capacitor to the input voltage. When several channels are converted consecutively, the inherent tracking time is 6 ADC clock cycles. However, the tracking time can be increased using the TRACKTIM field in the Mode register (ADC_MR) and the TRACKX4 bit in the Extended Mode register (ADC_EMR).
- ADC inherent conversion time—the time for the ADC to convert the sampled analog voltage. This time is constant and is defined from start of conversion to end of conversion.
- Channel conversion period—the effective time between the end of the current channel conversion and the end of the next channel conversion.

Downloaded from Arrow.com.

Figure 56-2. Sequence of Consecutive ADC Conversions with TRACKTIM = 0

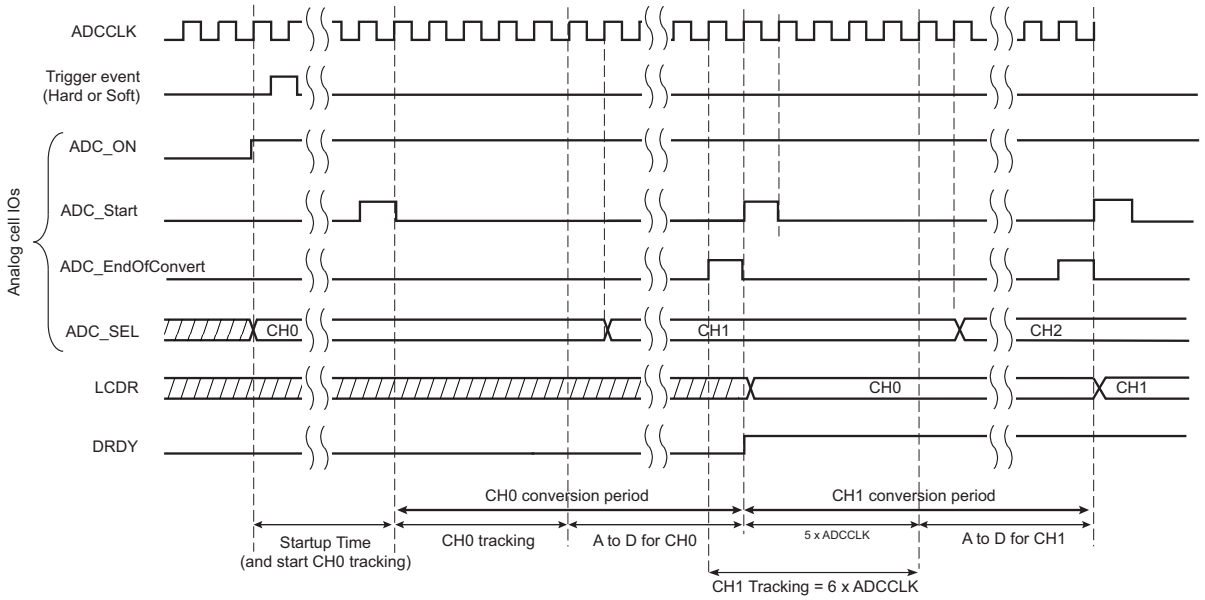
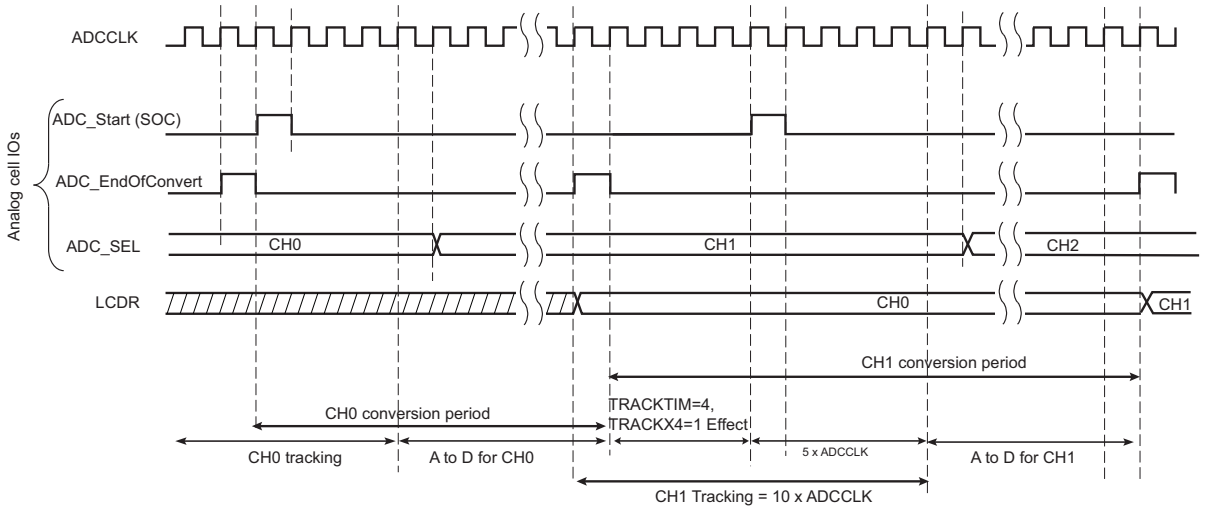


Figure 56-3. Sequence of Consecutive ADC Conversions with TRACKTIM = 4 and TRACK4X = 1



56.6.2 ADC Clock

The ADC uses the ADC clock (ADCCLK) to perform conversions. The ADC clock frequency is selected in the ADC_MR.PRESCAL.

To generate the ADC clock, the prescaler has two clock sources: the peripheral clock and the GCLK clock. This clock source is selected using the SRCCLK bit in the Extended Mode register (ADC_EMR).

If GCLK is selected as a source clock, the ADC clock frequency is independent of the processor/bus clock. At reset, the peripheral clock is selected.

If the ADC_EMR.SRCCLK bit is cleared, the prescaler clock (presc_clk) is driven by peripheral_clock. If the ADC_EMR.SRCCLK bit is set, the prescaler clock is driven by GCLK. The ADC clock frequency is between fpresc_clk/2, if PRESCAL is 0, and fpresc_clk/512, if PRESCAL is set to 255 (0xFF).

PRESCAL must be programmed to provide the ADC clock frequency parameter provided in the “Electrical Characteristics” section.

Downloaded from Arrow.com.

56.6.3 ADC Reference Voltage

The voltage reference input of the ADC is the ADVREFP pin and the negative reference voltage is ADVREFN. Refer to the section “Electrical Characteristics”.

56.6.4 Conversion Resolution

The ADC has a native resolution of 12 bits.

The ADC Controller provides enhanced resolution up to 16 bits by means of digital averaging.

If ADTRG is asynchronous to the ADC peripheral clock, the internal resynchronization introduces a jitter of 1 peripheral clock. This jitter may reduce the resolution of the converted signal.

The same applies when using the independent clock (ADC_MR.SRCCLK = 1), if the provided clock is asynchronous to ADC peripheral clock.

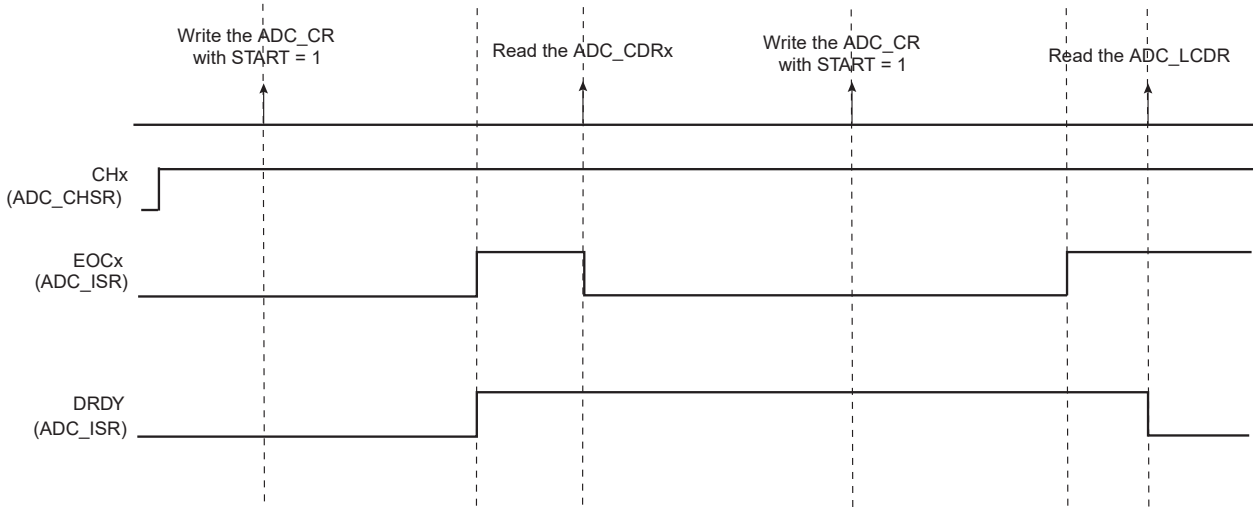
56.6.5 Conversion Results

When a conversion is completed, the resulting digital value is stored in the Channel Data register (ADC_CDRx) of the current channel and in the Last Converted Data register (ADC_LCDR). By setting the TAG option in ADC_EMR, ADC_LCDR presents the channel number associated with the last converted data in the CHNB field.

When a conversion is completed, the channel EOC bit and the DRDY bit in the Interrupt Status register (ADC_ISR) are set. In the case of a connected DMA channel, DRDY rising triggers a data request. In any case, either EOC and DRDY can trigger an interrupt.

Reading one of the ADC_CDRx clears the corresponding EOC bit. Reading ADC_LCDR clears the DRDY bit.

Figure 56-4. EOCx and DRDY Flag Behavior



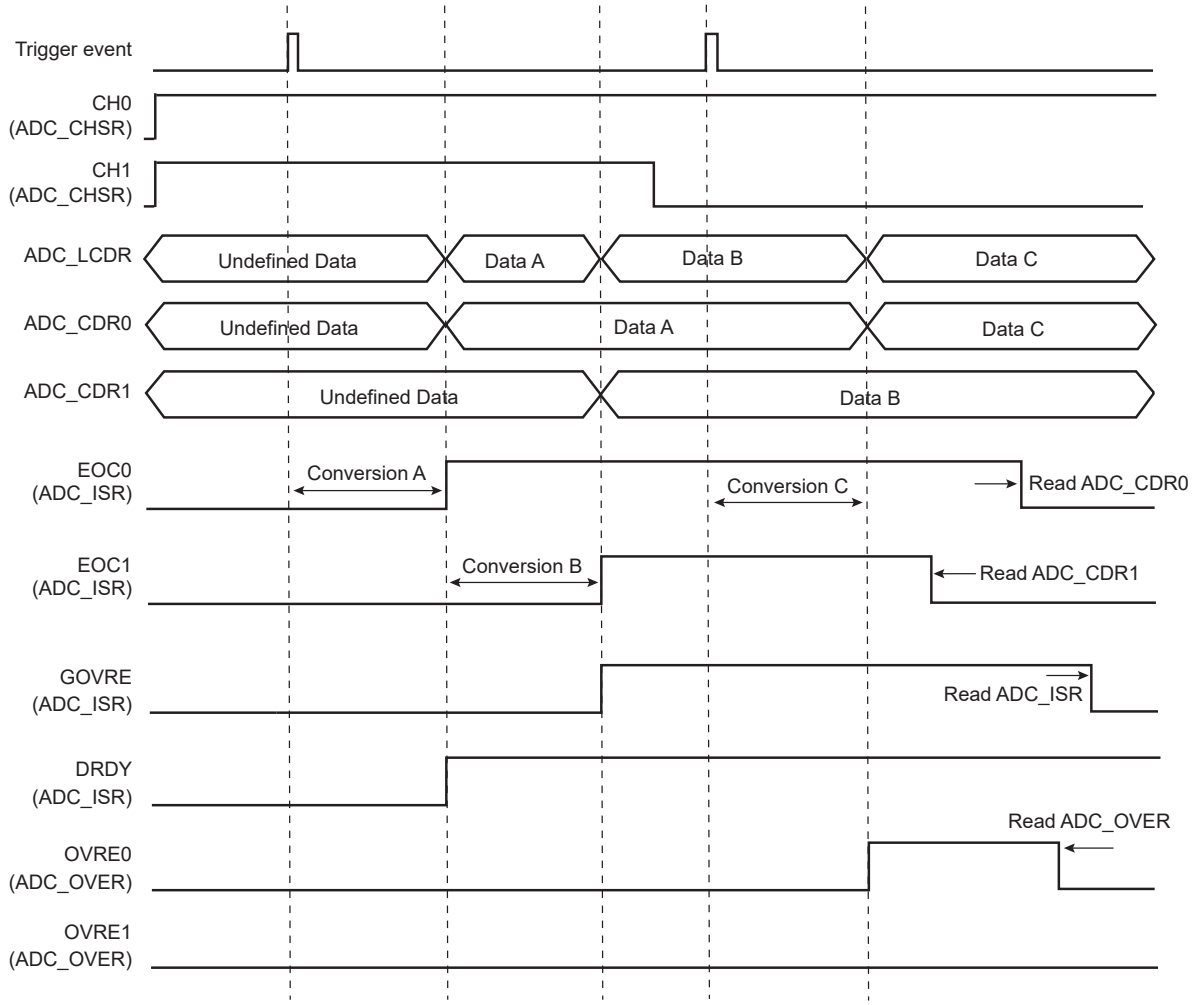
If ADC_CDR is not read before further incoming data is converted, the corresponding OVREx flag is set in the Overrun Status register (ADC_OVER).

If new data is converted when DRDY is high, the ADC_ISR.GOVRE bit is set.

The OVREx flag is automatically cleared when ADC_OVER is read, and the GOVRE flag is automatically cleared when ADC_ISR is read.

Downloaded from Arrow.com.

Figure 56-5. EOCx, OVREx and GOVREx Flag Behavior



WARNING If the corresponding channel is disabled during a conversion or if it is disabled and then reenabled during a conversion, its associated data and corresponding EOCx and GOVRE flags in ADC_ISR and OVREx flags in ADC_OVER are unpredictable.

56.6.6 Conversion Results Format

The conversion results can be signed (2's complement) or unsigned depending on the value of the ADC_EMR.SIGNMODE field.

If conversion results are signed and resolution is less than 16 bits, the sign is extended up to the bit 15 (e.g., 0xF43 for 12-bit resolution is read as 0xFF43, and 0x467 is read as 0x0467).

56.6.7 Conversion Triggers

Conversions of the active analog channels are started with a software or hardware trigger. The software trigger is provided by writing the Control register (ADC_CR) with the START bit at 1 and ADC_TRGR.TRGMOD=0.

The list of external/internal events is provided in 56.7.2 ADC_MR. The hardware trigger is selected using the ADC_MR.TRGSEL field. The selected hardware trigger is enabled if TRGMOD = 1, 2 or 3 in the Trigger register (ADC_TRGR). In these modes, the software trigger is disabled (writing ADC_CR.START=1 has no effect).

Downloaded from Arrow.com.

The ADC also provides a dual trigger mode (ADC_LCTMR.DUALTRIG = 1) in which the higher index channel can be sampled at a rhythm different from the other channels. The trigger of the last channel is generated by the RTC. See 56.6.12 Last Channel Specific Measurement Trigger.

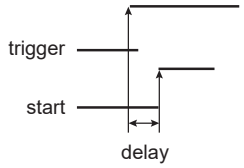
The ADC_TRGR.TRGMOD field selects the hardware trigger from the following:

- Any edge, either rising or falling or both, detected on the external trigger pin ADTRG or internal triggers
- The Pen Detect, depending on how the PENDET bit is set in the Touchscreen Mode register (ADC_TSMR)
- A continuous trigger, meaning the ADC Controller restarts the next sequence as soon as it finishes the current one
- A periodic trigger, which is defined by programming the ADC_TRGR.TRGPER field

The minimum time between two consecutive trigger events must be strictly greater than the duration time of the longest conversion sequence according to configuration of registers ADC_MR, ADC_CHSR, ADC_SEQRx, and ADC_TSMR.

If a hardware trigger is selected, the start of a conversion is triggered after a delay starting at each rising edge of the selected signal. Due to asynchronous handling, the delay may vary in a range of two peripheral clock periods to one ADC clock period. This delay introduces sampling jitter in the A/D conversion process and may therefore degrade the conversion performance (e.g., SNR, THD).

Figure 56-6. Hardware Trigger Delay



If one of the TIOA outputs is selected, the corresponding Timer Counter channel must be programmed in Waveform mode.

Only one start command is necessary to initiate a conversion sequence on all the enabled channels. The ADC hardware logic automatically performs the conversions on the active channels, then waits for a new request. The Channel Enable (ADC_CHER) and Channel Disable (ADC_CHDR) registers enable the analog channels to be enabled or disabled independently.

If the ADC is used with a DMA, only the transfers of converted data from enabled channels are performed and the resulting data buffers should be interpreted accordingly.

56.6.8 Sleep Mode and Conversion Sequencer

The ADC Sleep mode maximizes power saving by automatically deactivating the ADC when it is not being used for conversions. Sleep mode is selected by setting the ADC_MR.SLEEP bit.

Sleep mode is managed by a conversion sequencer, which automatically processes the conversions of all channels at lowest power consumption.

This mode can be used when the minimum period of time between two successive trigger events is greater than the startup period of the ADC. Refer to section “Electrical Characteristics”.

When a start conversion request occurs, the ADC is automatically activated. As the analog cell requires a startup time, the logic waits during this time and starts the conversion on the enabled channels. When all conversions are complete, the ADC is deactivated until the next trigger. Events triggered during the sequence are ignored.

The conversion sequencer allows automatic processing with minimum processor intervention and optimized power consumption. Conversion sequences can be performed periodically using the internal timer (ADC_TRGR) or the PWM event line. The periodic acquisition of several samples can be processed automatically without any intervention of the processor via the DMA.

The sequence can be customized by programming the Sequence Channel registers ADC_SEQR1 and ADC_SEQR2 and setting the USEQ bit of the Mode register (ADC_MR). The user can choose a specific order of channels and can program up to 12 conversions by sequence. The user is free to create a personal sequence by writing channel numbers in ADC_SEQR1 and ADC_SEQR2. Not only can channel numbers be written in any sequence, channel numbers can be repeated several times. When the ADC_MR.USEQ bit is set, the ADC_SEQR1.USCHx and

Downloaded from Arrow.com.

ADC_SEQR2.USCHx fields are used to define the sequence. Only enabled USCHx fields will be part of the sequence. Each USCHx field has a corresponding enable, CHx-1, in ADC_CHER.

If all ADC channels (i.e., 12) are used on an application board, there is no restriction of usage of the user sequence. However, if some ADC channels are not enabled for conversion but rather used as pure digital inputs, the respective indexes of these channels cannot be used in the user sequence fields (see ADC_SEQRx). For example, if channel 4 is disabled (ADC_CHSR[4] = 0), ADC_SEQRx fields USCH1 up to USCH12 must not contain the value 4. Thus the length of the user sequence may be limited by this behavior.

As an example, if only four channels over 12 (CH0 up to CH3) are selected for ADC conversions, the user sequence length cannot exceed four channels. Each trigger event may launch up to four successive conversions of any combination of channels 0 up to 3 but no more (i.e., in this case the sequence CH0, CH0, CH1, CH1, CH1 is impossible).

A sequence that repeats the same channel several times requires more enabled channels than channels actually used for conversion. For example, the sequence CH0, CH0, CH1, CH1 requires four enabled channels (four free channels on application boards) whereas only CH0, CH1 are really converted.

Note: The reference voltage pins always remain connected in Normal mode as in Sleep mode.

56.6.9 Comparison Window

The ADC Controller features automatic comparison functions. It compares converted values to a low threshold, a high threshold or both, depending on the value of the ADC_EMR.CMPMODE field. The comparison can be done on all channels or only on the channel specified in the ADC_EMR.CMPSEL field. To compare all channels, the ADC_EMR.CMPALL bit must be set.

If set, the ADC_EMR.CMPTYPE bit can be used to discard all conversion results that do not match the comparison conditions. Once a conversion result matches the comparison conditions, all the subsequent conversion results are stored in ADC_LCDR (even if these results do not meet the comparison conditions). Setting the ADC_CR.CMPRST bit immediately stops the conversion result storage until the next comparison match.

If the ADC_EMR.CMPTYPE bit is cleared, all conversions are stored in ADC_LCDR. Only the conversions that match the comparison conditions trigger the ADC_ISR.COMPE flag.

Moreover, a filtering option can be set by writing the number of consecutive comparison matches needed to raise the flag. This number can be written and read in the ADC_EMR.CMPFILTER field. The filtering option is dedicated to reinforcing the detection of an analog signal overpassing a predefined threshold. The filter is cleared as soon as ADC_ISR is read, so this filtering function must be used with peripheral DMA controller and works only when using Interrupt mode (no polling).

The flag can be read on the ADC_ISR.COMPE bit and can trigger an interrupt.

The high threshold and the low threshold can be read/write in the Compare Window register (ADC_CWR).

Depending on the sign of the conversion, chosen with the ADC_EMR.SIGNMODE field, the high threshold and low threshold values must be signed or unsigned to maintain consistency during the comparison. If the conversion is signed, both thresholds must also be signed; if the conversion is unsigned, both thresholds must be unsigned. If comparison occurs on all channels, the ADC_EMR.SIGNMODE field must be set to ALL_UNSIGNED or ALL_SIGNED and the thresholds must be set accordingly.

56.6.10 Pseudo-differential, Differential and Single-ended Input Modes

56.6.10.1 Input-output Transfer Functions

The ADC can be configured to operate in the following input voltage modes:

- Single-ended—ADC_CCR.DIFFx = 0 and ADC_PDR.PDIFFx bit = 0. This is the default mode after a reset.
- Differential—ADC_CCR.DIFFx = 1 and ADC_PDR.PDIFFx bit = 0 (see the figure below). In Differential mode, the ADC requires differential input signals having a VDD/2 common mode voltage (refer to section “Electrical Characteristics”).
-

The following equations give the unsigned ADC input-output transfer function in each mode(1). With signed conversions (see field ADC_EMR.SIGNMODE), subtract 2047 from the ADC_LCDR.DATA value given below. Note that Single-ended mode introduces an x2 gain compared to Pseudo-differential mode.

Downloaded from Arrow.com.

In the formula, REFP = VREFP, REFN = VREFN.

Single-ended mode:

$$\text{ADC_LCDR.LDATA} = \frac{\text{ADx} - \text{REFN}}{\text{REFP} - \text{REFN}} \times 2^{12}$$

Differential mode:

$$\text{ADC_LCDR.LDATA} = \left(1 + \frac{\text{ADx} - \text{ADx+1}}{\text{REFP} - \text{REFN}}\right) \times 2^{11}$$

Pseudo-differential mode:

$$\text{ADC_LCDR.LDATA} = \left(1 + \frac{\text{ADx} - \text{AD11}}{\text{REFP} - \text{REFN}}\right) \times 2^{11}$$

Note: 1. Equations assume ADC_EMR.OSR = 1

If the ADC_MR.ANACH bit is set, the ADC can manage both differential channels and single-ended channels. If the ADC_MR.ANACH bit is cleared, the parameters defined in ADC_CCR are applied to all channels.

The following tables give the internal positive and negative ADC inputs assignment with respect to the programmed mode (ADC_CCR.DIFFx and ADC_PDR.PDIFFx).

For example, if Differential mode is required on channel 0, input pins AD0 and AD1 are used. In this case, only channel 0 must be enabled by writing a 1 to ADC_CHER.CH0.

Table 56-2. Input Pins and Channel Numbers in Single-ended and Differential Modes

Internal ADC Inputs (VIN+, VIN-)		Channel Number	
Single-ended Mode	Differential Mode	Single-ended Mode	Differential Mode
AD0, ADVREFN	AD0, AD1	CH0	CH0
AD1, ADVREFN	–	CH1	
AD2, ADVREFN	AD2, AD3	CH2	CH2
AD3, ADVREFN	–	CH3	
AD4, ADVREFN	AD4, AD5	CH4	CH4
AD5, ADVREFN	–	CH5	
AD6, ADVREFN	AD6, AD7	CH6	CH6
AD7, ADVREFN	–	CH7	
AD8, ADVREFN	AD8, AD9	CH8	CH8
AD9, ADVREFN	–	CH9	
AD10, ADVREFN	AD10, AD11	CH10	CH10
AD11, ADVREFN	–	CH11	

In Pseudo-differential mode, inputs are managed by a 12/2:1-channel analog multiplexer. See the table below.

Table 56-3. Input Pins and Channel Numbers in Pseudo-differential Mode

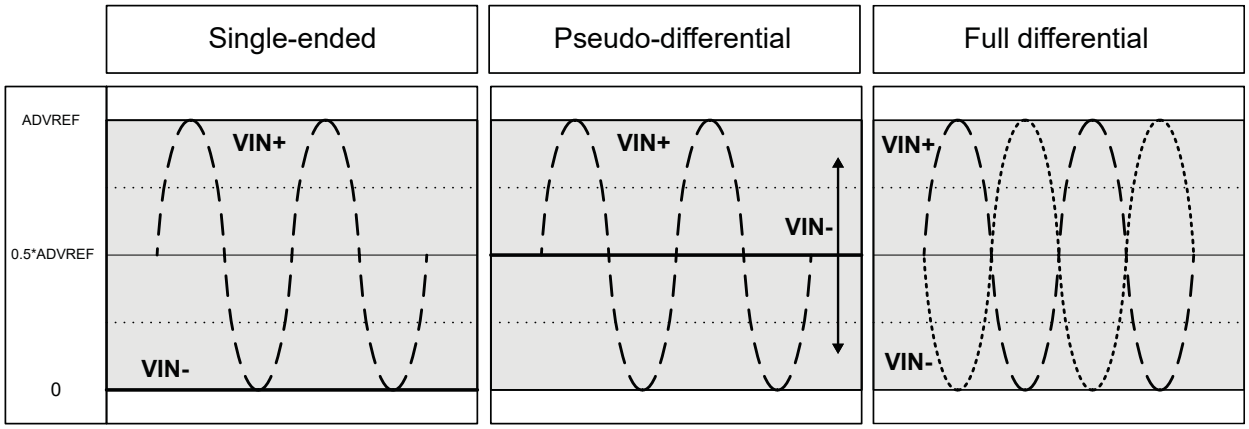
Internal ADC Inputs (VIN+, VIN-)	Channel Number
AD0, AD11	CH0
AD1, AD11	CH1
AD2, AD11	CH2
AD3, AD11	CH3

Downloaded from Arrow.com.

.....continued

Internal ADC Inputs (VIN+, VIN-)	Channel Number
AD4, AD11	CH4
AD5, AD11	CH5
AD6, AD11	CH6
AD7, AD11	CH7
AD8, AD11	CH8
AD9, AD11	CH9
AD10, AD11	CH10
AD11, AD11	CH11

Figure 56-7. Analog Full Scale Ranges in Single-Ended/Pseudo-differential/Differential Applications



56.6.11 ADC Timings

The ADC startup time is programmed through the ADC_MR.STARTUP field. Refer to the “Electrical Characteristics” section.

The ADC Controller provides an inherent tracking time of six ADC clock cycles.

A minimal tracking time is necessary for the ADC to guarantee the best converted final value between two conversions. The tracking time can be adjusted to accommodate a range of source impedances. If more than six ADC clock cycles are required, the tracking time can be increased using the ADC_MR.TRACKTIM field and the ADC_EMR.TRACKX4 bit.

⚠WARNING No input buffer amplifier to isolate the source is included in the ADC. Refer to the section "Electrical Characteristics".

56.6.12 Last Channel Specific Measurement Trigger

The last channel (higher index available) embeds a specific mode allowing a measurement trigger period which differs from other active channels. This allows efficient management of the conversions especially if the channel is driven by a device with a variation of a different frequency from other converted channels (for example, but not limited to, temperature sensor).

The last channel can be sampled in different ways through the ADC Controller. The different methods of sampling depend on the ADC_TRGR.TRGMOD configuration field and on the ADC_CHSR.CH11 bit.

The last channel conversion can be triggered like the other channels by enabling the ADC_CHER.CH11 bit.

Downloaded from Arrow.com.

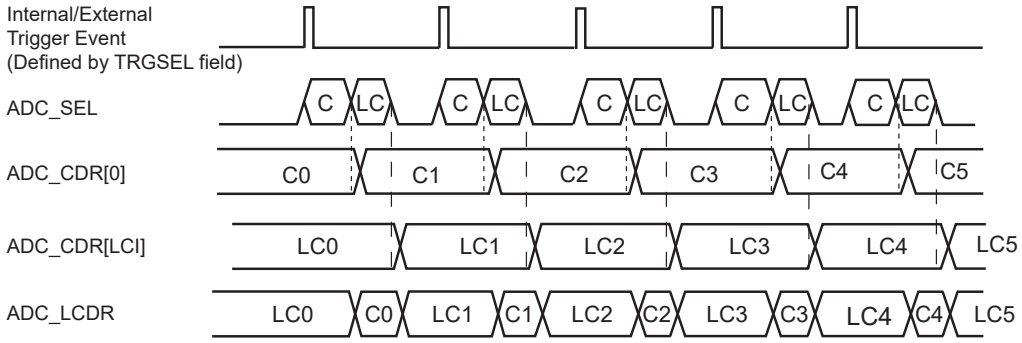
The manual start can only be performed if field TRGMOD = 0. When the ADC_CR.START bit is set, the last channel conversion is scheduled together with the other enabled channels (if any). The result of the conversion is placed in the ADC_CDR11 register, and the associated ADC_ISR.EOC11 flag is set.

If the last channel is enabled in the Channel Status register (ADC_CHSR), ADC_LCTMR.DUALTRIG is cleared and field TRGMOD = 1, 2, 3, 5, the last channel is periodically converted together with the other enabled channels and the result is placed in the ADC_LCDR and ADC_CDR11 registers. Thus the last channel conversion result is part of the DMA Controller buffer (see the following figure).

When the conversion result matches the conditions defined in ADC_LCTMR and ADC_LCCWR, the ADC_ISR.LCCHG flag is set.

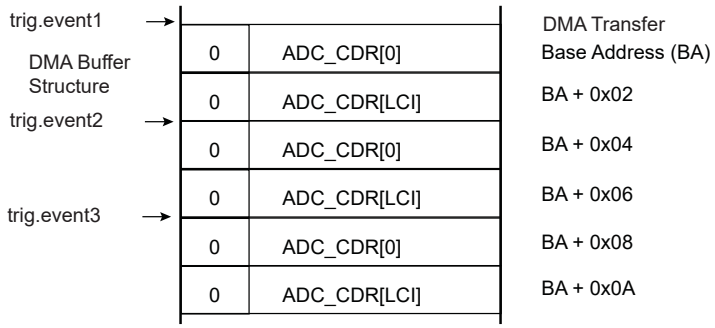
Figure 56-8. Same Trigger for All Channels (ADC_CHSR[LCI] = 1 and ADC_TRGR.TRGMOD = 1, 2, 3, 5)

ADC_LCTMR.DUALTRIG = 1



Notes: ADC_SEL: Command to the ADC analog cell
Cx: All ADC channel values except the last channel (highest index)
LCx: Last channel value
LCI: Last channel index

Assuming ADC_CHSR[0] = 1 and ADC_CHSR[LCI] = 1



If the last channel is driven by a device with a slower variation compared to other channels (temperature sensor for example), the channel can be enabled/disabled at any time. However, this may not be optimal for downstream processing.

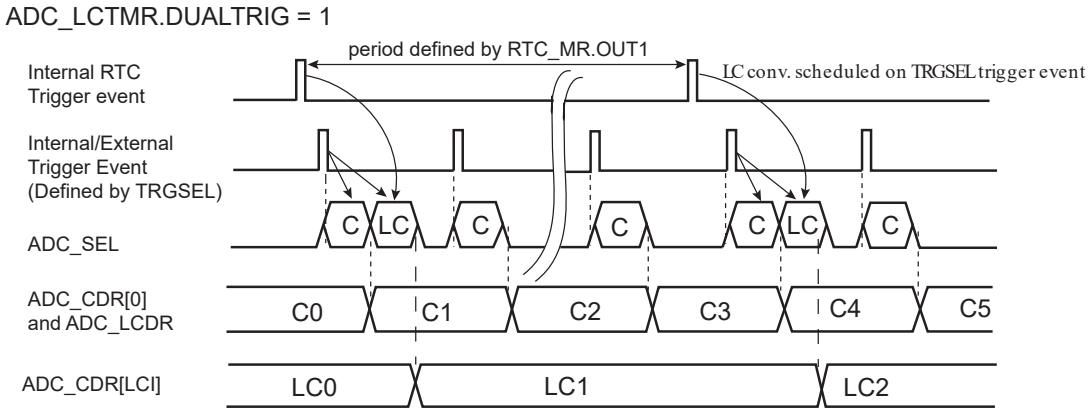
The ADC Controller allows a different way of triggering the measurement when DUALTRIG is set in the Last Channel Trigger Mode register (ADC_LCTMR) but CH11 is not set in ADC_CHSR.

Under these conditions, the last channel conversion is triggered with a period defined by the field RTC_MR.OUTx (see 56.3 Block Diagram for the value of 'x') while other channels are still active and triggered by internal/external triggers. The RTC event is processed on the next internal/external trigger event, as shown in the following figure. The internal/external trigger for other channels is selected through the ADC_MR.TRGSEL field.

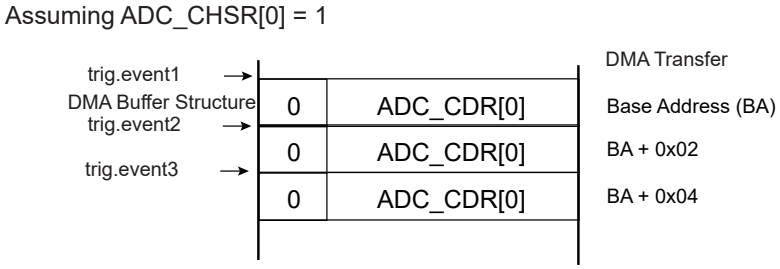
When DUALTRIG = 1, the result of each conversion of channel 11 is only uploaded in the ADC_CDR11 register and not in ADC_LCDR (see the following figure). Therefore, there is no change in the structure of the peripheral DMA controller buffer due to the conversion of the last channel: only the enabled channels are kept in the buffer. The end of conversion of the last channel is reported by the ADC_ISR.EOC11 flag.

Downloaded from Arrow.com.

Figure 56-9. Independent Trigger Measurement for Last Channel (ADC_CHSR[LCI] = 0 and ADC_TRGR.TRGMOD = 1, 2, 3, 5)



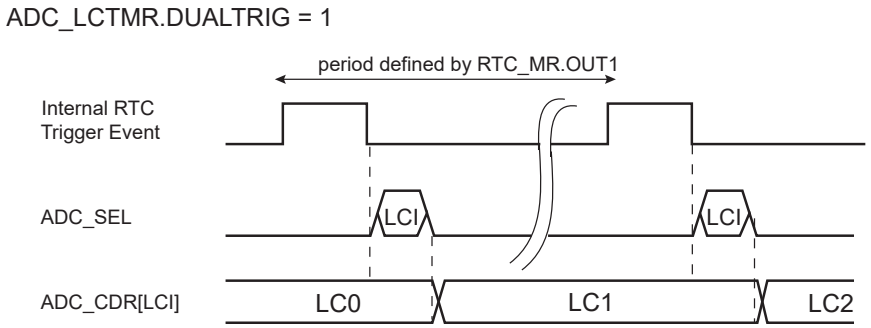
Notes: ADC_SEL: Command to the ADC analog cell
Cx: All ADC channel values except the last channel (highest index)
LCx: Last channel value
LCI: Last channel index



If DUALTRIG = 1 and field ADC_TRGR.TRGMOD = 0 and none of the channels are enabled in ADC_CHSR (ADC_CHSR = 0), then only channel 11 is converted at a rate defined by the trigger event signal that can be configured in RTC_MR.OUT1 (see the following figure).

This mode of operation, when combined with the Sleep mode operation of the ADC Controller, provides a low-power mode for last channel measure. This assumes there is no other ADC conversion to schedule at a high sampling rate or no other channel to convert.

Figure 56-10. Only Last Channel Measurement Triggered at Low Speed (ADC_CHSR[LCI] = 0 and ADC_TRGR.TRGMOD = 0)



Notes: ADC_SEL: Command to the ADC analog cell
LCx: Last channel value
LCI: Last channel index

Downloaded from Arrow.com.

56.6.13 Enhanced Resolution Mode and Digital Averaging Function

56.6.13.1 Enhanced Resolution Mode

The Enhanced Resolution mode is enabled if the OSR field is configured to 1, 2, 3 or 4 in ADC_EMR. The enhancement is based on a digital averaging function.

There is no averaging on the last index channel if the measure is triggered by an RTC event.

In this mode, the ADC Controller will trade off conversion speed against accuracy by averaging multiple samples, thus providing a digital low-pass filter function.

The selected oversampling ratio applies to all enabled channels when triggered by an RTC event.

$$\text{ADC_LCDR.LDATA} = \frac{1}{M} \times \sum_{k=0}^{k=N-1} \text{ADC}(k)$$

where N and M are given in the table below.

Table 56-4. Digital Averaging Function Configuration versus OSR Values

ADC_EMR.OSR Value	ADC_LCDR.LDATA Length	N Value	M Value	Full Scale Value	Maximum Value
0	12 bits	1	1	4095	4095
1	13 bits	4	2	8191	8190
2	14 bits	16	4	16383	16381
3	15 bits	64	8	32767	32761
4	16 bits	256	16	65535	65521

The average result is valid in ADC_CDRx (x corresponds to the index of the channel) only if the ADC_ISR.EOCn flag is set and if the ADC_OVER.OVREn flag is cleared. The average result for all channels is valid in ADC_LCDR only if the ADC_ISR.DRDY bit is set and the ADC_ISR.GOVRE bit is cleared.

Note that ADC_CDRs are not buffered. Therefore, when an averaging sequence is ongoing, the value in these registers changes after each averaging sample. However, overrun flags in ADC_OVER rise as soon as the first sample of an averaging sequence is received. Thus the previous averaged value is not read, even if the new averaged value is not ready.

Consequently, when an overrun flag rises in ADC_OVER, it means that the previous unread data is lost but it does not mean that this data has been overwritten by the new averaged value as the averaging sequence concerning this channel can still be ongoing.

When an oversampling is performed, the maximum value that can be read on ADC_CDRx or ADC_LCDR is not the full-scale value, even if the maximum voltage is supplied on the analog input. See table above.

56.6.13.2 Averaging Function versus Trigger Events

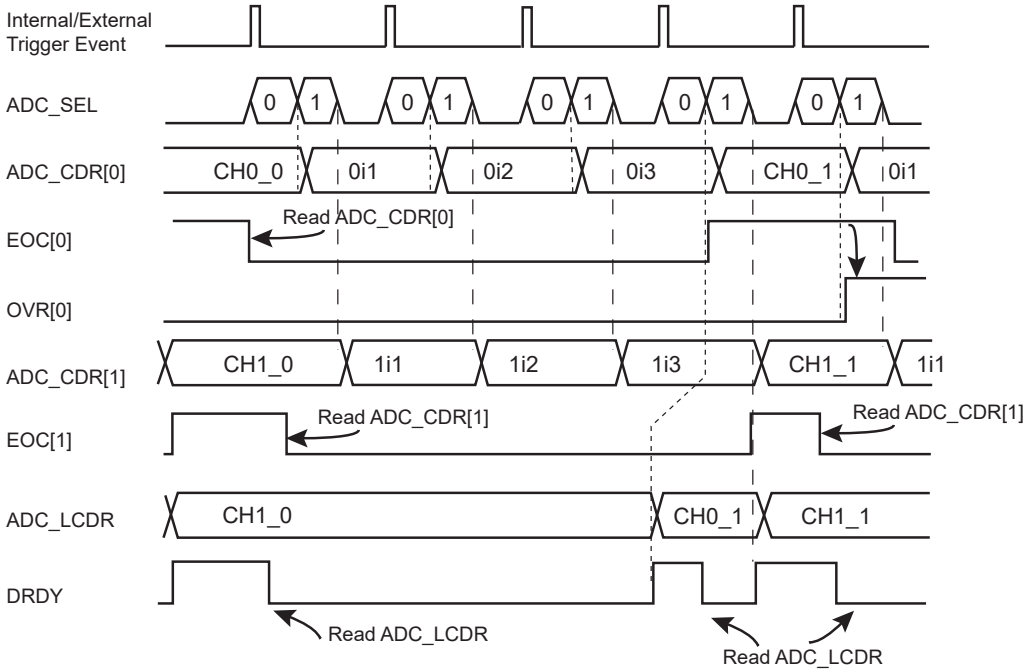
The samples can be defined in different ways for the averaging function depending on the configuration of the ADC_EMR.ASTE bit and the ADC_MR.USEQ bit.

When USEQ = 0, there are two possible ways to generate the averaging through the trigger event. If ADC_EMR.ASTE = 0, every trigger event generates one sample for each enabled channel, as described in the following figure. Therefore, four trigger events are required to obtain the result of averaging if OSR = 1.

Downloaded from Arrow.com.

Figure 56-11. Digital Averaging Function Waveforms Over Multiple Trigger Events

ADC_EMR.OSR = 1, ASTE = 0, ADC_CHSR[1:0] = 0x3 and ADC_MR.USEQ = 0



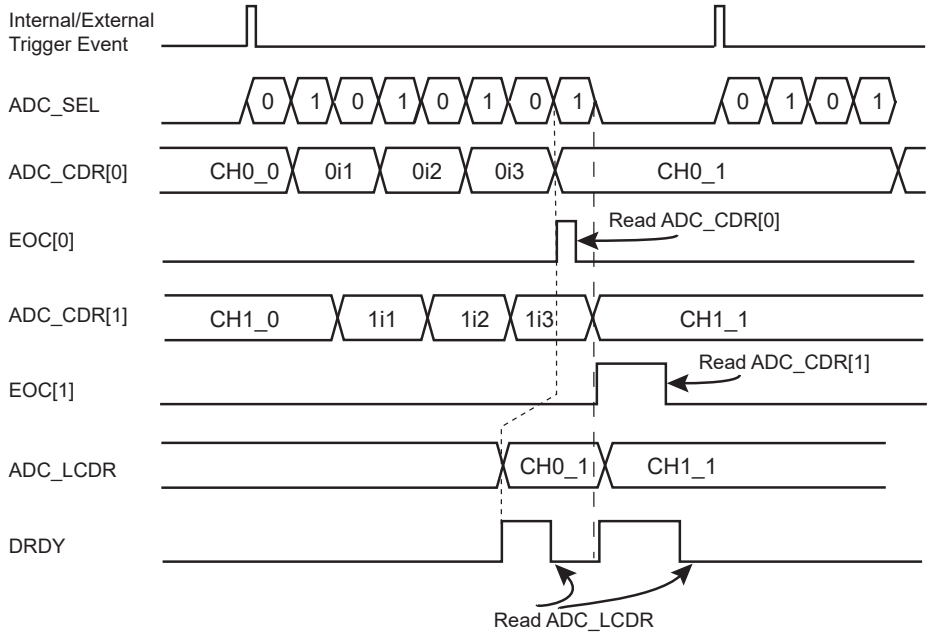
Note: ADC_SEL: Command to the ADC analog cell
0i1, 0i2, 0i3, 1i1, 1i2, 1i3 are intermediate results and CH0_0, CH0_1, CH1_0 and CH1_1 are final results of average function.

If ADC_EMR.ASTE = 1 and ADC_MR.USEQ = 0, the sequence to be converted, defined in ADC_CHSR, is automatically repeated n times (where n corresponds to the oversampling ratio defined in the ADC_EMR.OSR field). As a result, only one trigger is required to obtain the result of the averaging function as described in the following figure.

Downloaded from Arrow.com.

Figure 56-12. Digital Averaging Function Waveforms on a Single Trigger Event

ADC_EMR.OSR = 1, ASTE = 1, ADC_CHSR[1:0] = 0x3 and ADC_MR.USEQ = 0



Note: ADC_SEL: Command to the ADC analog cell
0i1, 0i2, 0i3, 1i1, 1i2, 1i3 are intermediate results and CH0_0, CH0_1, CH1_0 and CH1_1 are final results of average function.

When USEQ = 1, the user can define the channel sequence to be converted by configuring ADC_SEQRx and ADC_CHER so that channels are not interleaved during the averaging period. Under these conditions, a sample is defined for each end of conversion as described in the figure below.

When USEQ = 1 and ASTE = 1, OSR can be only configured to 1. Up to three channels can be converted in this mode. The averaging result will be placed in the corresponding ADC_CDRx and in ADC_LCDR for each trigger event. The ADC real sample rate remains the maximum ADC sample rate divided by 4.

It is important that the user sequence follows a specific pattern. The user sequence must be programmed in such a way that it generates a stream of conversion, where a same channel is successively converted.

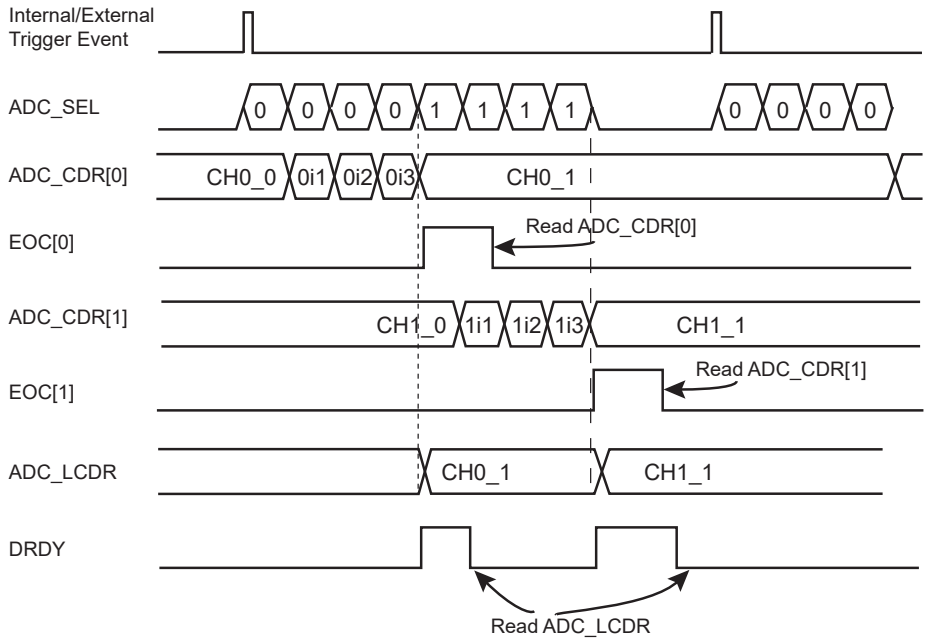
Table 56-5. Example Sequence Configurations (USEQ = 1, ASTE = 1, OSR = 1)

Register	Number of Channels Non-interleaved Averaging - Register Value		
	1 (e.g., CH0)	2 (e.g., CH0, CH1)	3 (e.g., CH0, CH1, CH2)
ADC_CHSR	0x0000_000F	0x0000_00FF	0x0000_0FFF
ADC_SEQR1	0x0000_0000	0x1111_0000	0x1111_0000
ADC_SEQR2	0x0000_0000	0x0000_0000	0x0000_2222

Downloaded from Arrow.com.

Figure 56-13. Digital Averaging Function Waveforms on a Single Trigger Event, Non-interleaved

ADC_EMR.OSR = 1, ASTE = 1, ADC_CHSR[7:0] = 0xFF and ADC_MR.USEQ = 1
ADC_SEQR1 = 0x1111_0000



Note: ADC_SEL: Command to the ADC analog cell
0i1, 0i2, 0i3, 1i1, 1i2, 1i3 are intermediate results and CH0_0, CH0_1, CH1_0 and CH1_1 are final results of average function.

56.6.14 Automatic Error Correction

The ADC features automatic error correction of conversion results. Offset and gain error corrections are available. The correction can be enabled for each channel and correction values (offset and gain) are .

To enable error correction, the corresponding ECORRx bit must be set in the Channel Error Correction register (ADC_CECR). The offset and gain values used to compensate the results are .

The error correction for channels used with the touchscreen is available in the ADC Touchscreen Correction Values register (ADC_TSCVR).

The ADC_EMR.ADCMODE field is used to configure a running mode of the ADC Normal mode, Offset Error mode, or Gain Error mode (see 56.7.16 ADC_EMR). ADCMODE uses 3 internal references to be measured and to extract the offset and gain error from 3 point-measurement codes. If some references already exist on the final application connected to some input channel ADx, they can be used as a replacement of the ADCMODE to generate the 2 or 3 points of calibration and used to extract the GAINCORR and OFFSETCORR.

After a reset, the running mode of the ADC is Normal mode. Offset Error mode and Gain Error mode are used to determine values of offset compensation and gain compensation, respectively, to apply to conversion results. The table below provides formulas to obtain the compensation values, with:

- OFFSETCORR—the Offset Correction value. OFFSETCORR is a signed value.
- GAINCORR—the Gain Correction value
- GCi—the intermediate Gain Compensation value
- Gs—the value 15
- ConvValue—the value converted by the ADC (as returned in ADC_LCDR or ADC_CDR)
- Resolution—the resolution used to process the conversion (either RESOLUTION, RESOLUTION+1, RESOLUTION+3, RESOLUTION+4 RESOLUTION+2, RESOLUTION+3, RESOLUTION+4).

Downloaded from Arrow.com.

Table 56-6. ADC Running Modes

ADC_EMR.ADCMODE	Mode	Description
0	Normal	Normal mode of operation to perform conversions
1	Offset Error	For unsigned conversions: OFFSETCORR = ConvValue − $2^{(\text{Resolution}} - 1)$
		For signed conversions: OFFSETCORR = ConvValue
2	Gain Error	GCi = ConvValue
3		$\text{GAINCORR} = \frac{3584}{\text{GCi} - \text{ConvValue}} \times 2^{(\text{Gs})}$

The final conversion result after error correction is obtained using the following formula:

$$\text{Corrected Data} = (\text{Converted Data} + \text{OFFSETCORR}) \times \frac{\text{GAINCORR}}{2^{(\text{Gs})}}$$

56.6.15 Touchscreen

56.6.15.1 Touchscreen Mode

The ADC_TSMR.TSMODE parameter is used to enable/disable the touchscreen functionality, to select the type of screen (4-wire or 5-wire) and, in the case of a 4-wire screen, to activate (or not) the pressure measurement.

In 4-wire mode, channels 0, 1, 2 and 3 must not be used for classic ADC conversions. Likewise, in 5-wire mode, channels 0, 1, 2, 3, and 4 must not be used for classic ADC conversions.

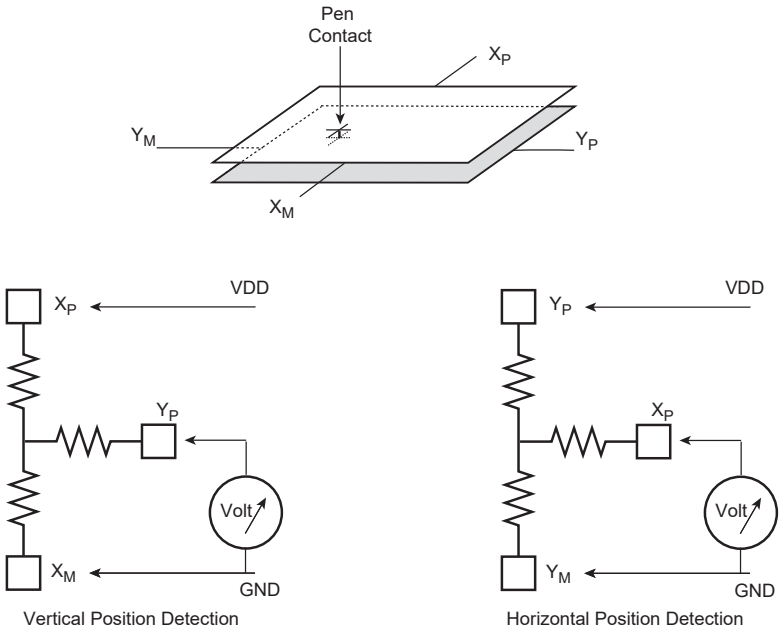
56.6.15.2 4-wire Resistive Touchscreen Principles

A resistive touchscreen is based on two resistive films, each one being fitted with a pair of electrodes, placed at the top and bottom on one film, and on the right and left on the other. In between, there is a layer acting as an insulator, but also enables contact when you press the screen. This is illustrated in the following figure.

The ADC Controller can perform the following tasks without external components:

- position measurement
- pressure measurement
- pen detection

Figure 56-14. Touchscreen Position Measurement



Downloaded from Arrow.com.

56.6.15.3 4-wire Position Measurement Method

As shown in the above figure, to detect the position of a contact, a supply is first applied from top to bottom. Due to the linear resistance of the film, there is a voltage gradient from top to bottom. When a contact is performed on the screen, the voltage propagates at the point the two surfaces come into contact. If the input impedance on the right and left electrodes is high enough, the film intrinsic resistor does not affect this voltage.

For the horizontal direction, the same method is used, but by applying supply from left to right. The range depends on the supply voltage and on the loss in the switches that connect to the top and bottom electrodes.

In an ideal world (linear, with no loss through switches), the horizontal position is equal to:

VY_M / VDD or VY_P / VDD.

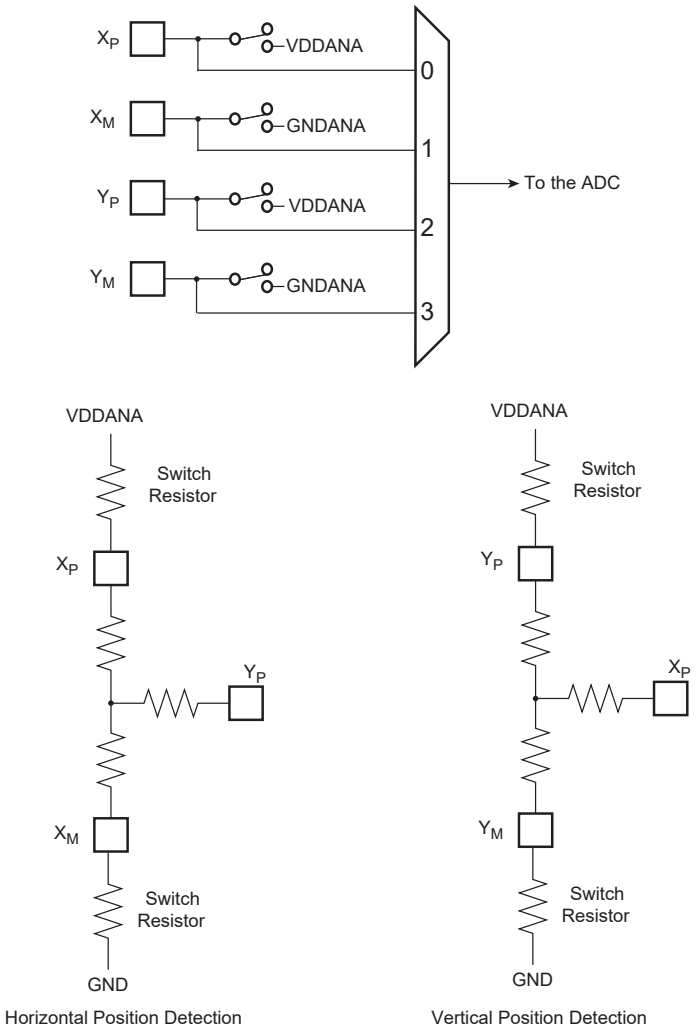
The implementation with on-chip power switches is shown in the figure below. The voltage measurement at the output of the switch compensates for the switches loss.

It is possible to correct for switch loss by performing the operation:

$[VY_P - VX_M] / [VX_P - VX_M]$.

This requires additional measurements, as shown in the figure below.

Figure 56-15. Touchscreen Switches Implementation



56.6.15.4 4-wire Pressure Measurement Method

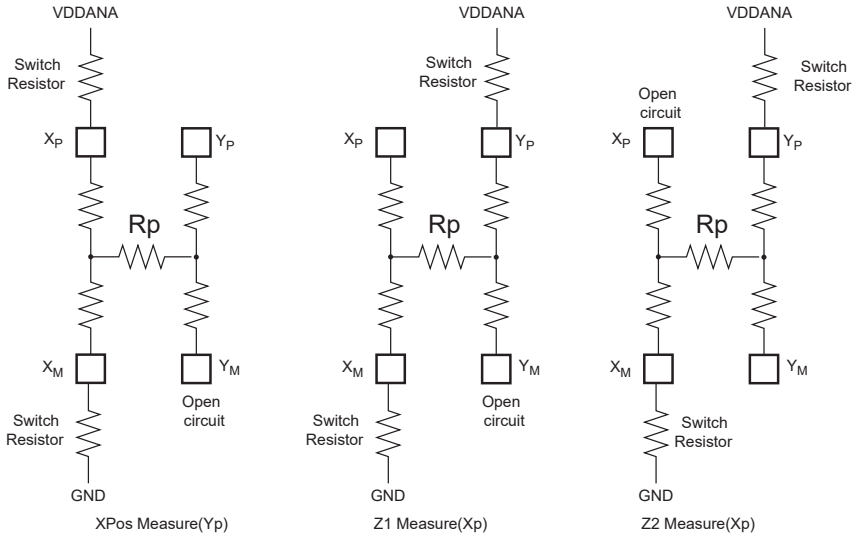
The method to measure the pressure (Rp) applied to the touchscreen is based on the known resistance of the X-Panel resistance (Rxp).

Downloaded from Arrow.com.

Three conversions (Xpos,Z1,Z2) are necessary to determine the value of Rp (Zaxis resistance).

Rp = Rxp × (Xpos/1024) × [(Z2/Z1)-1]

Figure 56-16. Pressure Measurement



56.6.15.5 5-wire Resistive Touchscreen Principles

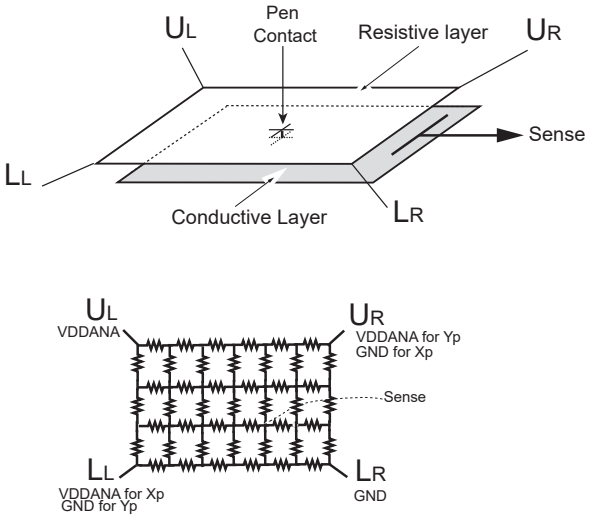
To make a 5-wire touchscreen, a resistive layer with a contact point at each corner and a conductive layer are used.

The 5-wire touchscreen differs from the 4-wire type mainly in that the voltage gradient is applied only to one layer, the resistive layer, while the other layer is the sense layer for both measurements.

The measurement of the X position is obtained by biasing the upper left corner and lower left corner to VDDANA and the upper right corner and lower right to ground.

To measure along the Y axis, bias the upper left corner and upper right corner to VDDANA and bias the lower left corner and lower right corner to ground.

Figure 56-17. 5-Wire Principle



56.6.15.6 5-wire Position Measurement Method

In an application only monitoring clicks, 100 points per second is typically needed. For handwriting or motion detection, the number of measurements to consider is approximately 200 points per second. This must take into account that multiple measurements are included (over sampling, filtering) to compute the correct point.

Downloaded from Arrow.com.

The 5-wire touchscreen panel works by applying a voltage at the corners of the resistive layer and measuring the vertical or horizontal resistive network with the sense input. The ADC converts the voltage measured at the point the panel is touched.

A measurement of the Y position of the pointing device is made by:

- Connecting Upper left (UL) and upper right (UR) corners to VDDANA
- Connecting Lower left (LL) and lower right (LR) corners to ground.

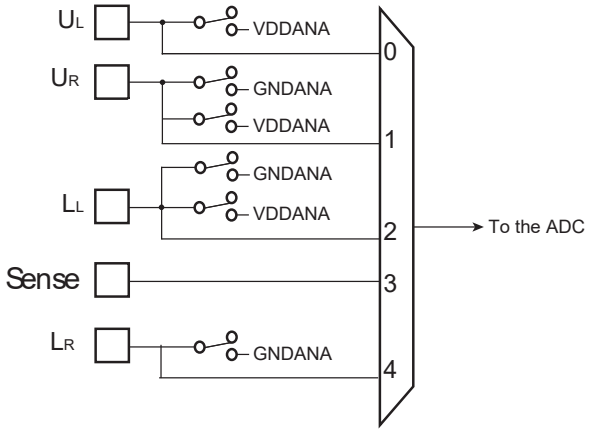
The voltage measured is determined by the voltage divider developed at the point of touch (Y position) and the SENSE input is converted by ADC.

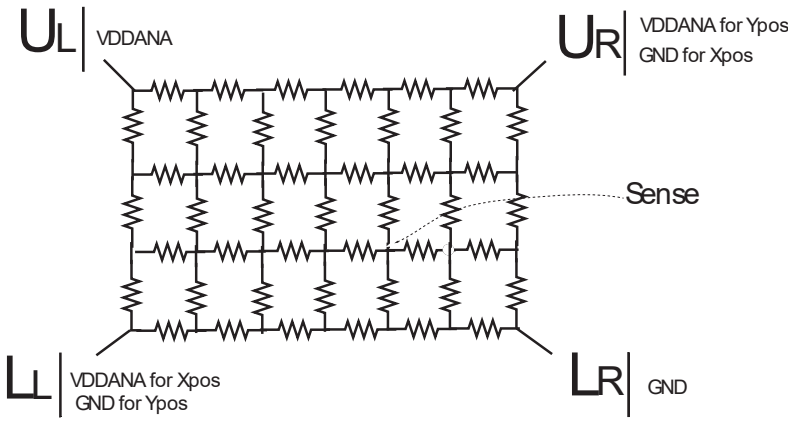
A measurement of the X position of the pointing device is made by:

- Connecting the upper left (UL) and lower left (LL) corners to ground
- Connecting the upper right and lower right corners to VDDANA.

The voltage measured is determined by the voltage divider developed at the point of touch (X position) and the SENSE input is converted by ADC.

Figure 56-18. Touchscreen Switches Implementation





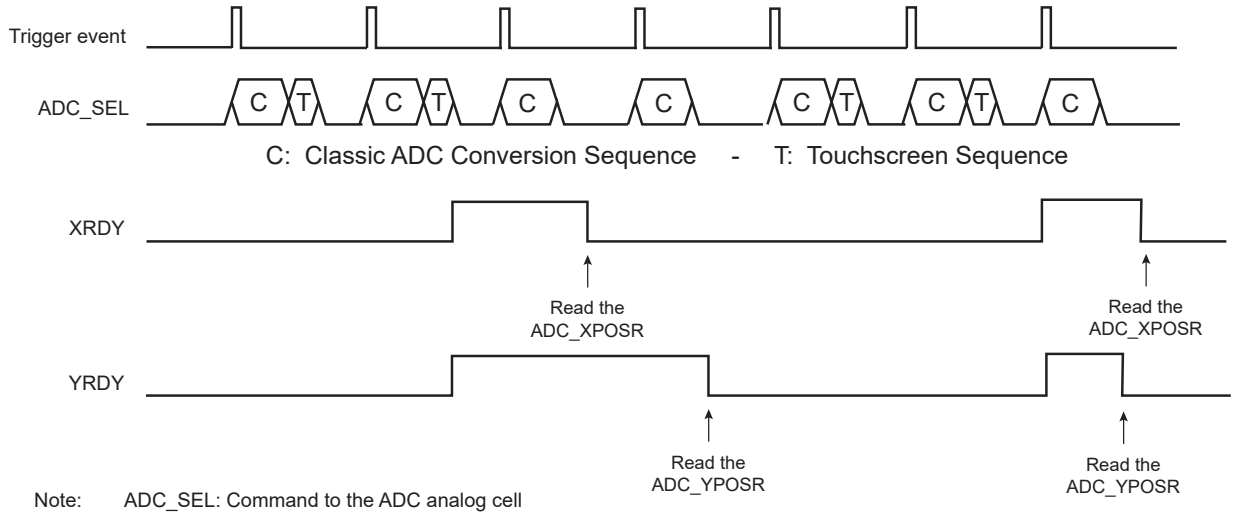
56.6.15.7 Sequence and Noise Filtering

The ADC Controller can manage ADC conversions and touchscreen measurement. On each trigger event the sequence of ADC conversions is performed as described in 56.6.8 Sleep Mode and Conversion Sequencer. The touchscreen measure frequency can be specified in number of trigger events by writing the ADC_TSMR.TSFREQ parameter. An internal counter counts triggers up to TSFREQ, and every time it rolls out, a touchscreen sequence is appended to the classic ADC conversion sequence (see figure below).

Downloaded from Arrow.com.

Additionally the user can average multiple touchscreen measures by writing the ADC_TSMR.TSAV parameter. This can be 1, 2, 4 or 8 measures performed on consecutive triggers as illustrated in the figure below. Consequently, the ADC_TSMR.TSFREQ parameter must be greater than or equal to the ADC_TSMR.TSAV parameter.

Figure 56-19. Insertion of Touchscreen Sequences (TSFREQ = 2; TSAV = 1)



56.6.15.8 Measured Values, Registers and Flags

As soon as the controller finishes the Touchscreen sequence, XRDY, YRDY and PRDY are set and can generate an interrupt. These flags can be read in the Interrupt Status register (ADC_ISR). They are reset independently by reading in the ADC Touchscreen X Position register (ADC_XPOSR), the ADC Touchscreen Y Position register (ADC_YPOSR) and the ADC Touchscreen Pressure register (ADC_PRESSR).

ADC_XPOSR presents XPOS (VX - VXmin) on its LSB and XSCALE (VXMAX - VXmin) aligned on the 16th bit.

ADC_YPOSR presents YPOS (VY - VYmin) on its LSB and YSCALE (VYMAX - VYmin) aligned on the 16th bit.

To improve the quality of the measure, the user must calculate XPOS/XSCALE and YPOS/YSCALE.

VXMAX, VXmin, VYMAX, and VYmin are measured at the first startup of the controller. These values can change during use, so it can be necessary to refresh them. Refresh can be done by writing ‘1’ in the ADC_CR.TSCALIB field.

ADC_PRESSR presents Z1 on its LSB and Z2 aligned on the 16th bit. See 56.6.15.4 4-wire Pressure Measurement Method.

56.6.15.9 Pen Detect Method

When there is no contact, it is not necessary to perform a conversion. However, it is important to detect a contact by keeping the power consumption as low as possible.

The implementation polarizes one panel by closing the switch on (X_P/U_L) and ties the horizontal panel by an embedded resistor connected to Y_M / Sense. This resistor is enabled by a fifth switch. Since there is no contact, no current is flowing and there is no related power consumption. As soon as a contact occurs, a current is flowing in the Touchscreen and a Schmitt trigger detects the voltage in the resistor.

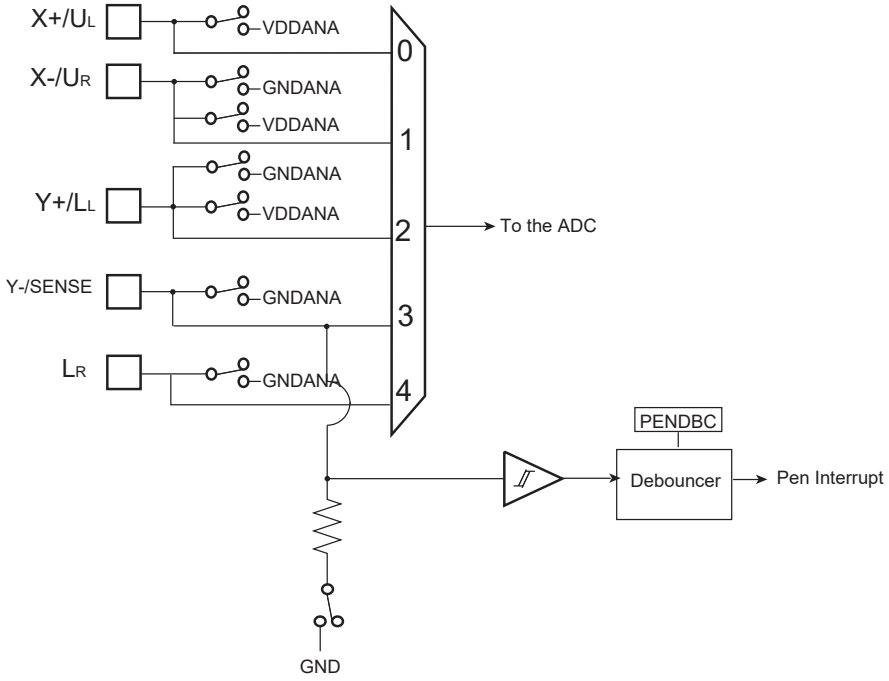
The Touchscreen Interrupt configuration is entered by programming the ADC_TSMR.PENDET bit. If this bit is written at 1, the controller samples the pen contact state when it is not converting and waiting for a trigger.

To complete the circuit, a programmable debouncer is placed at the output of the Schmitt trigger. This debouncer is programmable up to 2^{15} ADC clock periods. The debouncer length can be selected by programming the ADC_TSMR.PENDBC field.

Due to the analog switch’s structure, the debouncer circuitry is only active when no conversion (touchscreen or classic ADC channels) is in progress. Thus, if the time between the end of a conversion sequence and the arrival of the next trigger event is lower than the debouncing time configured on ADC_TSMR.PENDBC, the debouncer will not detect any contact.

Downloaded from Arrow.com.

Figure 56-20. Touchscreen Pen Detect



The touchscreen pen detect can be used to generate an ADC interrupt to wake up the system. The pen detect generates two types of status, reported in ADC_ISR:

- the ADC_ISR.PEN bit is set as soon as a contact exceeds the debouncing time as defined by the ADC_TSMR.PENDBC field and remains set until ADC_ISR is read.
- the ADC_ISR.NOPEN bit is set as soon as no current flows for a time over the debouncing time as defined by PENDBC and remains set until ADC_ISR is read.

Both bits are automatically cleared as soon as ADC_ISR is read, and can generate an interrupt by writing ADC_IER.

Moreover, the rising of either one of them clears the other, they cannot be set at the same time.

The ADC_ISR.PENS bit shows the current status of the pen contact.

56.6.16 Asynchronous and Partial Wakeup (SleepWalking)

This operating mode is a means of data preprocessing that qualifies an incoming event, thus allowing the ADC to decide whether or not to wake up the system. Asynchronous and partial wakeup is mainly used when the system is in Wait mode (refer to the Power Management Controller (PMC) section for further details). It can also be enabled when the system is fully running.

Once the Asynchronous and partial wakeup mode is enabled, no access must be performed in the ADC before a wakeup is performed by the ADC.

When the Asynchronous and partial wakeup mode is enabled for the ADC (refer to the Power Management Controller (PMC) section), the PMC decodes a clock request from the ADC. The clock request is generated as soon as a trigger event occurs. Only a trigger from RTC or ADTRG pin can be used in partial wakeup mode. The selection between RTC or ADTRG pin is performed through the ADC_MR.TRGSEL field.

If the system is in Wait mode (processor and peripheral clocks switched off), the PMC restarts the fast RC oscillator and provides the clock only to the ADC.

To perform a conversion at regular intervals with RTC trigger, the RTC must be configured with the following settings: RTC_MR.OUT0=7 and RTC_MR.THIGH=7. The period of the trigger can be defined in RTC_MR.TPERIOD.

To trigger a conversion using the ADTRG pin, the minimum high level duration of the ADTRG signal must be greater than 2 clock periods of the fast RC oscillator. The maximum duration of the high level must be limited to the amount of startup and conversion time.

Downloaded from Arrow.com.

As soon as the clock is provided by the PMC, the ADC processes the conversions and compares the converted values with the ADC_CWR.LOWTHRES and ADC_CWR.HIGHTHRES field values.

The ADC instructs the PMC to disable the clock if the converted value does not meet the conditions defined by the ADC_CWR.LOWTHRES and ADC_CWR.HIGHTHRES field values.

If the converted value meets the conditions, the ADC instructs the PMC to exit the full system from Wait mode.

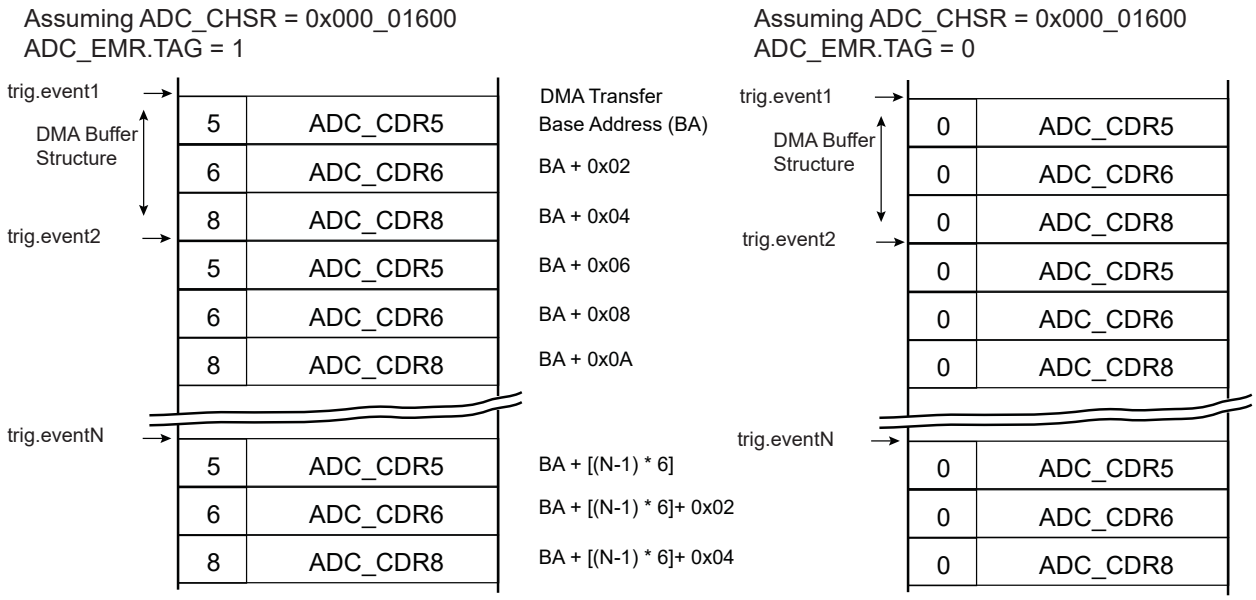
If the processor and peripherals are running, the ADC can be configured in Asynchronous and partial wakeup mode by enabling PMC_SLPWK_ER (refer to the Power Management Controller (PMC) section). When a trigger event occurs, the ADC requests the clock from the PMC and the comparison is performed. If there is a comparison match, the ADC continues to request the clock. If there is no match, the clock is switched off for the ADC only, until a new trigger event is detected.

It is recommended to write a '1' to the ADC_MR.SLEEP bit to reduce the power consumption of the ADC analog part when the system is waiting for a trigger event.

56.6.17 Buffer Structure

The DMA read channel is triggered each time a new data is stored in ADC_LCDR. The same data structure is repeatedly stored in ADC_LCDR each time a trigger event occurs. Depending on user mode of operation (ADC_MR, ADC_CHSR, ADC_SEQR1, ADC_SEQR2, ADC_TSMR) the structure differs. Each data read to DMA buffer, carried on a half-word (16-bit), consists of last converted data right-aligned and when the ADC_EMR.TAG is set, the four most significant bits are carrying the channel number thus allowing an easier postprocessing in the DMA buffer or better checking the DMA buffer integrity.

Figure 56-21. Buffer Structure



As soon as touchscreen conversions are required, the pen detection function can help the postprocessing of the buffer. See 56.6.17.4 Pen Detection Status.

56.6.17.1 Classic ADC Channels Only (Touchscreen Disabled)

When no touchscreen conversion is required (i.e., ADC_TSMR.TSMODE = 0), the data structure within the buffer is defined by ADC_MR, ADC_CHSR, ADC_SEQRx. See figure Buffer Structure.

If the user sequence is not used (i.e., ADC_MR.USEQ is cleared) then only the value of ADC_CHSR defines the data structure. For each trigger event, enabled channels will be consecutively stored in ADC_LCDR and automatically read to the buffer.

When the user sequence is configured (i.e., ADC_MR.USEQ is set) not only does ADC_CHSR modify the data structure of the buffer, but ADC_SEQRx registers may modify the data structure of the buffer as well.

Downloaded from Arrow.com.

56.6.17.2 Touchscreen Channels Only

When only touchscreen conversions are required (i.e., TSMODE ≠ 0 in ADC_TSMR and ADC_CHSR equals 0), the structure of data within the buffer is defined by ADC_TSMR.

When TSMODE = 1 or 3, each trigger event adds two half-words in the buffer (assuming TSAV = 0), first half-word being ADC_XPOSR.XPOS, then ADC_YPOSR.YPOS. If TSAV/TSFREQ ≠ 0, the data structure remains unchanged. Not all trigger events add data to the buffer.

When TSMODE = 2, each trigger event adds four half-words to the buffer (assuming TSAV = 0), first half-word being ADC_XPOSR.XPOS, followed by ADC_YPOSR.YPOS and finally ADC_PRESSR.Z1, followed by ADC_PRESSR.Z2.

When ADC_EMR.TAG is set, the CHNB field (four most significant bits of ADC_LCDR) is cleared when ADC_XPOSR.XPOS is transmitted and set when ADC_YPOSR.YPOS is transmitted, allowing an easier post-processing of the buffer or a better checking of the buffer integrity. In case 4-wire with Pressure mode is selected, the Z1 value is transmitted to the buffer along with tag set to 2 and Z2 is tagged with value 3.

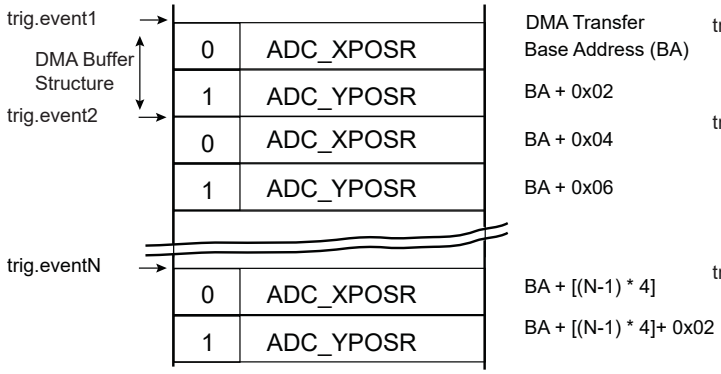
XSCALE and YSCALE (calibration values) are not transmitted to the buffer because they are supposed to be constant and moreover only measured at the very first startup of the controller or upon user request.

There is no change in buffer structure whatever the value of the PENDET.ADC_TSMR bit configuration, but it is recommended to use the pen detection function for buffer postprocessing (see 56.6.17.4 Pen Detection Status).

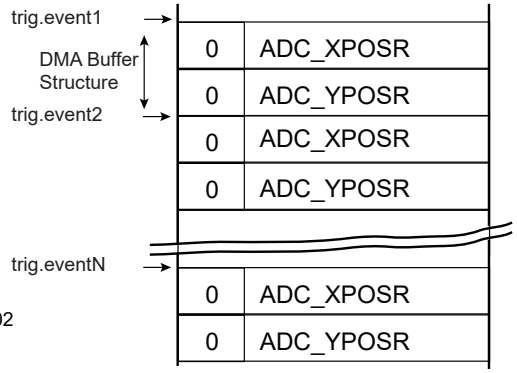
Downloaded from Arrow.com.

Figure 56-22. Buffer Structure When Only Touchscreen Channels are Enabled

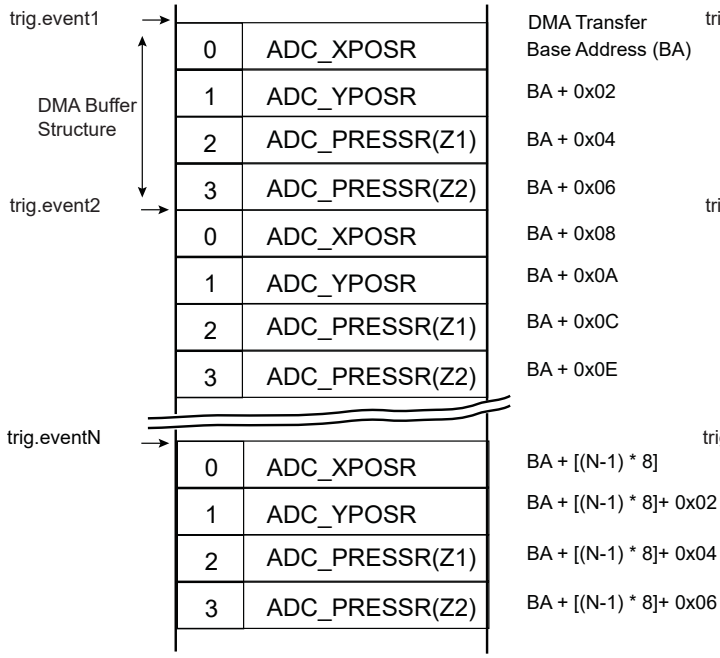
Assuming ADC_TSMR.TSMOD = 1 or 3
ADC_TSMR.TSAV = 0
ADC_CHSR = 0x000_00000, ADC_EMR.TAG = 1



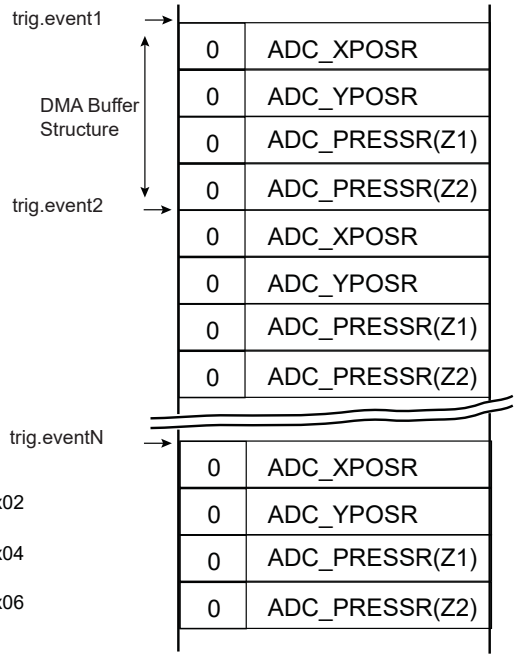
Assuming ADC_TSMR.TSMOD = 1 or 3
ADC_TSMR.TSAV = 0
ADC_CHSR = 0x000_00000, ADC_EMR.TAG = 0



Assuming ADC_TSMR.TSMOD = 2
ADC_TSMR.TSAV = 0
ADC_CHSR = 0x000_00000, ADC_EMR.TAG = 1



Assuming ADC_TSMR.TSMOD = 2
ADC_TSMR.TSAV = 0
ADC_CHSR = 0x000_00000, ADC_EMR.TAG = 0



56.6.17.3 Interleaved Channels

When both classic ADC channels (CH4/CH5 up to CH12 are set in ADC_CHSR) and touchscreen conversions are required (TSMODE ≠ 0 in ADC_TSMR), the structure of the buffer differs according to the ADC_TSMR.TSAV and ADC_TSMR.TSFREQ values.

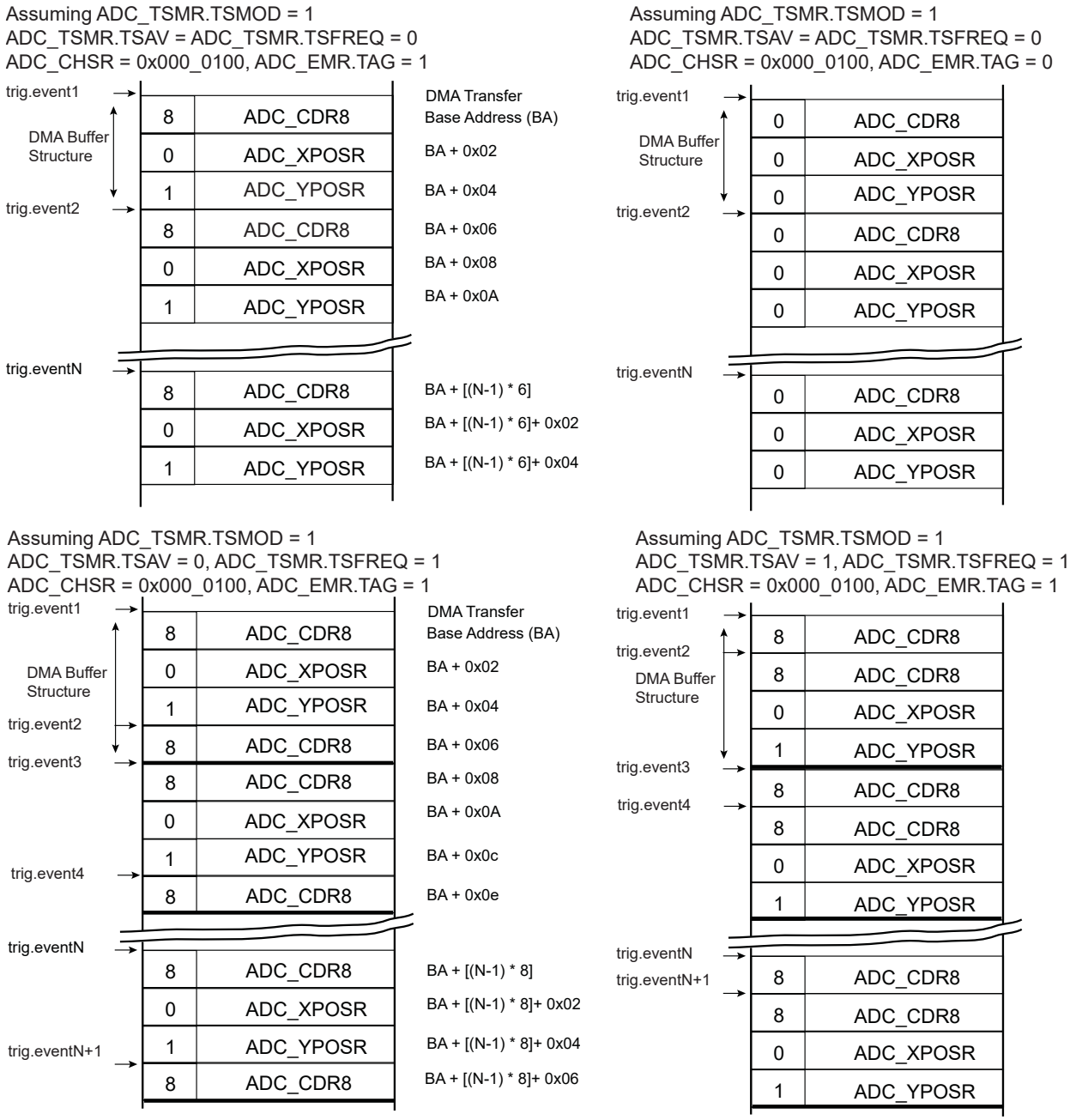
If TSFREQ ≠ 0, not all events generate touchscreen conversions, therefore the buffer structure is based on 2^{TSFREQ} trigger events. Given a TSFREQ value, the location of touchscreen conversion results depends on TSAV value.

When TSFREQ = 0, TSAV must equal 0.

There is no change in buffer structure whatever the value of the ADC_TSMR.PENDET bit configuration, but it is recommended to use the pen detection function for buffer post-processing (see 56.6.17.4 Pen Detection Status).

Downloaded from Arrow.com.

Figure 56-23. Buffer Structure When Classic ADC and Touchscreen Channels are Interleaved



56.6.17.4 Pen Detection Status

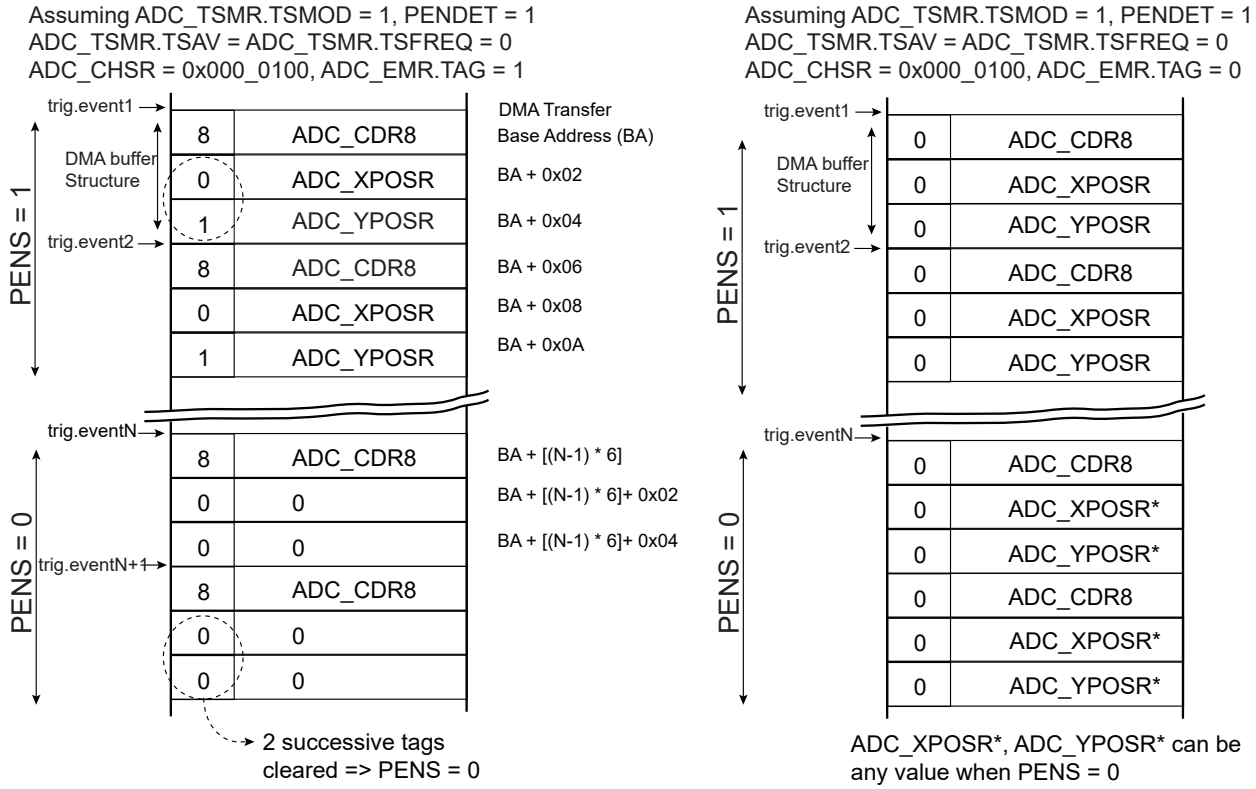
If the pen detection measure is enabled (ADC_TSMR.PENDET is set), the XPOS, YPOS, Z1, Z2 values transmitted to the buffer through ADC_LCDR are cleared (including the CHNB field), if the ADC_ISR.PENS flag is 0. When the ADC_ISR.PENS flag is set, XPOS, YPOS, Z1, Z2 are normally transmitted.

Therefore, using pen detection together with tag function eases the post-processing of the buffer, especially to determine which touchscreen converted values correspond to a period of time when the pen was in contact with the screen.

When the pen detection is disabled or the tag function is disabled, XPOS, YPOS, Z1, Z2 are normally transmitted without tag and no relationship can be found with pen status, thus post-processing may not be easy.

Downloaded from Arrow.com.

Figure 56-24. Buffer Structure With and Without Pen Detection Enabled



56.6.18 Fault Event

The ADC Controller internal fault output is directly connected to the PWM fault input. The fault event may be asserted depending on the configuration of ADC_EMR, ADC_CWR, ADC_LCMR and ADC_LCCWR and converted values.

Two types of comparison can trigger a comparison event (fault output pulse):

- The first comparison type is based on ADC_LCCWR settings, i.e., on all converted channels except the last one;
- The second comparison type is linked to the last channel.

As an example, overcurrent and temperature exceeding limits can trigger a fault to PWM.

When the comparison event occurs, the ADC fault output generates a pulse of one peripheral clock cycle to the PWM fault input. This fault line can be enabled or disabled within PWM. Should it be activated and asserted by the ADC Controller, the PWM outputs are immediately placed in a safe state (pure combinational path). Note that the ADC fault output connected to the PWM is not the COMPE bit. Thus the Fault mode (FMOD) within the PWM configuration must be FMOD = 1.

56.6.19 Register Write Protection

To prevent any single software error from corrupting ADC behavior, certain registers in the address space can be write-protected by setting the bits WPEN and WPITEN in the "ADC Write Protection Mode Register" (ADC_WPMR).

If a write access to the protected registers is detected, the WPVS flag in the "ADC Write Protection Status Register" (ADC_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS flag is automatically reset by reading ADC_WPSR.

The following registers are write-protected when ADC_WPMR.WPEN is set:

- ADC Mode Register
- ADC Channel Sequence 1 Register
- ADC Channel Sequence 2 Register

Downloaded from Arrow.com.

- ADC Channel Enable Register
- ADC Channel Disable Register
- ADC Last Channel Trigger Mode Register
- ADC Last Channel Compare Window Register
- ADC Extended Mode Register
- ADC Compare Window Register
- ADC Channel Configuration Register
- ADC Analog Control Register
- ADC Pseudo-Differential Register
- ADC_Touchscreen Mode Register
- ADC Trigger Register
- ADC Correction Values Register
- ADC Channel Error Correction Register
- ADC Touchscreen Correction Values Register

The following registers are write-protected when ADC_WPMR.WPITEN is set:

- ADC Interrupt Enable Register
- ADC Interrupt Disable Register

The following register is write-protected when ADC_WPMR.WPCTEN is set:

- ADC Control Register

Downloaded from Arrow.com.

56.7 Register Summary

Offset	Name	Bit Pos.								
0x00	ADC_CR	31:24								
		23:16								
		15:8								
		7:0				CMPRST		TSCALIB	START	SWRST
0x04	ADC_MR	31:24	USEQ	MAXSPEED	TRANSFER[1:0]		TRACKTIM[3:0]			
		23:16	ANACH				STARTUP[3:0]			
		15:8	PRESCAL[7:0]							
		7:0		FWUP	SLEEP		TRGSEL[2:0]			
0x08	ADC_SEQR1	31:24	USCH8[3:0]				USCH7[3:0]			
		23:16	USCH6[3:0]				USCH5[3:0]			
		15:8	USCH4[3:0]				USCH3[3:0]			
		7:0	USCH2[3:0]				USCH1[3:0]			
0x0C	ADC_SEQR2	31:24								
		23:16								
		15:8					USCH11[3:0]			
		7:0	USCH10[3:0]				USCH9[3:0]			
0x10	ADC_CHER	31:24								
		23:16								
		15:8					CH11	CH10	CH9	CH8
		7:0	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
0x14	ADC_CHDR	31:24								
		23:16								
		15:8					CH11	CH10	CH9	CH8
		7:0	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
0x18	ADC_CHSR	31:24								
		23:16								
		15:8					CH11	CH10	CH9	CH8
		7:0	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
0x1C ... 0x1F	Reserved									
0x20	ADC_LCDR	31:24				CHNBOSR[4:0]				
		23:16								
		15:8	LDATA[15:8]							
		7:0	LDATA[7:0]							
0x24	ADC_IER	31:24		NOPEN	PEN			COMPE	GOVRE	DRDY
		23:16		PRDY	YRDY	XRDY	LCCHG			
		15:8					EOC11	EOC10	EOC9	EOC8
		7:0	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
0x28	ADC_IDR	31:24		NOPEN	PEN			COMPE	GOVRE	DRDY
		23:16		PRDY	YRDY	XRDY	LCCHG			
		15:8					EOC11	EOC10	EOC9	EOC8
		7:0	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
0x2C	ADC_IMR	31:24		NOPEN	PEN			COMPE	GOVRE	DRDY
		23:16		PRDY	YRDY	XRDY	LCCHG			
		15:8					EOC11	EOC10	EOC9	EOC8
		7:0	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
0x30	ADC_ISR	31:24	PENS	NOPEN	PEN			COMPE	GOVRE	DRDY
		23:16		PRDY	YRDY	XRDY	LCCHG			
		15:8					EOC11	EOC10	EOC9	EOC8
		7:0	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
0x34	ADC_LCTMR	31:24								
		23:16								
		15:8								
		7:0			CMPMOD[1:0]					DUALTRIG

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0x38	ADC_LCCWR	31:24					HIGHTHRES[11:8]			
		23:16	HIGHTHRES[7:0]							
		15:8					LOWTHRES[11:8]			
		7:0	LOWTHRES[7:0]							
0x3C	ADC_OVER	31:24								
		23:16								
		15:8					OVRE11	OVRE10	OVRE9	OVRE8
		7:0	OVRE7	OVRE6	OVRE5	OVRE4	OVRE3	OVRE2	OVRE1	OVRE0
0x40	ADC_EMR	31:24			ADCMODE[1:0]			SIGNMODE[1:0]		TAG
		23:16		TRACKX4	SRCCLK	ASTE		OSR[2:0]		
		15:8			CMPFILTER[1:0]				CMPALL	
		7:0	CMPSEL[3:0]					CMPTYPE	CMPMODE[1:0]	
0x44	ADC_CWR	31:24	HIGHTHRES[15:8]							
		23:16	HIGHTHRES[7:0]							
		15:8	LOWTHRES[15:8]							
		7:0	LOWTHRES[7:0]							
0x48 ... 0x4B	Reserved									
0x4C	ADC_CCR	31:24					DIFF11	DIFF10	DIFF9	DIFF8
		23:16	DIFF7	DIFF6	DIFF5	DIFF4	DIFF3	DIFF2	DIFF1	DIFF0
		15:8								
		7:0								
0x50	ADC_CDR0	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
...										
0x7C	ADC_CDR11	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x80 ... 0x93	Reserved									
0x94	ADC_ACR	31:24								
		23:16								
		15:8							IBCTL[1:0]	
		7:0							PENDETSENS[1:0]	
0x98 ... 0x9F	Reserved									
0xA0	ADC_PDR	31:24								
		23:16								
		15:8	PDIFF15	PDIFF14	PDIFF13	PDIFF12	PDIFF11	PDIFF10	PDIFF9	PDIFF8
		7:0	PDIFF7	PDIFF6	PDIFF5	PDIFF4	PDIFF3	PDIFF2	PDIFF1	PDIFF0
0xA4 ... 0xAF	Reserved									
0xB0	ADC_TSMR	31:24	PENDBC[3:0]							PENDET
		23:16		NOTSDMA			TSSCTIM[3:0]			
		15:8					TSFREQ[3:0]			
		7:0			TSAV[1:0]				TSMODE[1:0]	
0xB4	ADC_XPOSR	31:24					XSCALE[11:8]			
		23:16	XSCALE[7:0]							
		15:8					XPOS[11:8]			
		7:0	XPOS[7:0]							

Downloaded from Arrow.com.

.....continued

Offset	Name	Bit Pos.								
0xB8	ADC_YPOSR	31:24					YSCALE[11:8]			
		23:16	YSCALE[7:0]							
		15:8					YPOS[11:8]			
		7:0	YPOS[7:0]							
0xBC	ADC_PRESSR	31:24					Z2[11:8]			
		23:16	Z2[7:0]							
		15:8					Z1[11:8]			
		7:0	Z1[7:0]							
0xC0	ADC_TRGR	31:24	TRGPER[15:8]							
		23:16	TRGPER[7:0]							
		15:8								
		7:0						TRGMOD[2:0]		
0xC4 ... 0xD3	Reserved									
0xD4	ADC_CVR	31:24	GAINCORR[15:8]							
		23:16	GAINCORR[7:0]							
		15:8	OFFSETCORR[15:8]							
		7:0	OFFSETCORR[7:0]							
0xD8	ADC_CECR	31:24								
		23:16								
		15:8					ECORR11	ECORR10	ECORR9	ECORR8
		7:0	ECORR7	ECORR6	ECORR5	ECORR4	ECORR3	ECORR2	ECORR1	ECORR0
0xDC	ADC_TSCVR	31:24	TSGAINCORR[15:8]							
		23:16	TSGAINCORR[7:0]							
		15:8	TSOFFSETCORR[15:8]							
		7:0	TSOFFSETCORR[7:0]							
0xE0 ... 0xE3	Reserved									
0xE4	ADC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCTEN	WPITEN	WPEN
0xE8	ADC_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

Downloaded from Arrow.com.

56.7.1 ADC Control Register

Name: ADC_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCTEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				CMPRST		TSCALIB	START	SWRST
Access				W		W	W	W
Reset				–		–	–	–

Bit 4 – CMPRST Comparison Restart

Value	Description
0	No effect.
1	Stops the conversion result storage until the next comparison match.

Bit 2 – TSCALIB Touchscreen Calibration
If conversion is in progress, the calibration sequence starts at the beginning of a new conversion sequence. If no conversion is in progress, the calibration sequence starts at the second conversion sequence located after the TSCALIB command (Sleep mode, waiting for a trigger event).
TSCALIB measurement sequence does not affect the Last Converted Data register (ADC_LCDR).

Value	Description
0	No effect.
1	Programs screen calibration (VDD/GND measurement)

Bit 1 – START Start Conversion

Value	Description
0	No effect.
1	Triggers a single sequence of analog-to-digital conversions if ADC_TRGR.TRGMOD=0.

Bit 0 – SWRST Software Reset

Value	Description
0	No effect.
1	Resets the ADC.

Downloaded from Arrow.com.

56.7.2 ADC Mode Register

Name: ADC_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	USEQ	MAXSPEED	TRANSFER[1:0]		TRACKTIM[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ANACH				STARTUP[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PRESCAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		FWUP	SLEEP		TRGSEL[2:0]			
Access		R/W	R/W		R/W	R/W	R/W	
Reset		0	0		0	0	0	

Bit 31 – USEQ User Sequence Enable

Value	Name	Description
0	NUM_ORDER	Normal mode: The controller converts channels in a simple numeric order depending only on the channel index.
1	REG_ORDER	User Sequence mode: The sequence respects what is defined in ADC_SEQR1 and ADC_SEQR2 registers and can be used to convert the same channel several times.

Bit 30 – MAXSPEED Maximum Sampling Rate Enable in Freerun Mode
This bit should always be set to 1.

Bits 29:28 – TRANSFER[1:0] Transfer Time
The TRANSFER field must be set to 2 to guarantee the optimal transfer time.

Bits 27:24 – TRACKTIM[3:0] Tracking Time

ADC_EMR.TRACK4X	TRACKTIM		
	0 to 3	4 to 14	15
0	$6 \times t_{ADCCLK}$		$7 \times t_{ADCCLK}$
1	$6 \times t_{ADCCLK}$	$([4 \times (TRACKTIM + 1)] - 10) \times t_{ADCCLK}$	

Bit 23 – ANACH Analog Change

Value	Name	Description
0	NONE	No analog change on channel switching: DIFF0 is used for all channels.
1	ALLOWED	Allows different analog settings for each channel. .

Bits 19:16 – STARTUP[3:0] Startup Time

Downloaded from Arrow.com.

Value	Name	Description
0	SUT0	0 periods of ADCCLK
1	SUT8	8 periods of ADCCLK
2	SUT16	16 periods of ADCCLK
3	SUT24	24 periods of ADCCLK
4	SUT64	64 periods of ADCCLK
5	SUT80	80 periods of ADCCLK
6	SUT96	96 periods of ADCCLK
7	SUT112	112 periods of ADCCLK
8	SUT512	512 periods of ADCCLK
9	SUT576	576 periods of ADCCLK
10	SUT640	640 periods of ADCCLK
11	SUT704	704 periods of ADCCLK
12	SUT768	768 periods of ADCCLK
13	SUT832	832 periods of ADCCLK
14	SUT896	896 periods of ADCCLK
15	SUT960	960 periods of ADCCLK

Bits 15:8 – PRESCAL[7:0] Prescaler Rate Selection
PRESCAL = ($f_{peripheral\ clock}$ / (2 × f_{ADCCLK})) – 1.

Bit 6 – FWUP Fast Wakeup

Value	Name	Description
0	OFF	If SLEEP is 1, then both ADC core and reference voltage circuitry are OFF between conversions
1	ON	If SLEEP is 1, then Fast Wakeup Sleep mode: The voltage reference is ON between conversions and ADC core is OFF

Bit 5 – SLEEP Sleep Mode

Value	Name	Description
0	NORMAL	Normal Mode: The ADC core and reference voltage circuitry are kept ON between conversions.
1	SLEEP	Sleep Mode: The wakeup time can be modified by programming the FWUP bit.

Bits 3:1 – TRGSEL[2:0] Trigger Selection
The trigger selection can be performed only if ADC_TRGR.TRGMOD = 1, 2 or 3.

Value	Name	Description
0	ADC_TRIG0	ADTRG
1	ADC_TRIG1	TIOA0
2	ADC_TRIG2	TIOA1
3	ADC_TRIG3	TIOA2
4	ADC_TRIG4	Reserved
5	ADC_TRIG5	Reserved
6	ADC_TRIG6	Reserved
7	ADC_TRIG7	Reserved

Downloaded from Arrow.com.

56.7.3 ADC Channel Sequence 1 Register

Name: ADC_SEQR1
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	USCH8[3:0]				USCH7[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	USCH6[3:0]				USCH5[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	USCH4[3:0]				USCH3[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	USCH2[3:0]				USCH1[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0:3, 4:7, 8:11, 12:15, 16:19, 20:23, 24:27, 28:31 – USCHx User Sequence Number x
This register can be used only if the ADC_MR.USEQ field is set to ‘1’.
Any USCHx field is processed only if the ADC_CHSR.CHx-1 bit reads logical ‘1’, else any value written in USCHx does not add the corresponding channel in the conversion sequence.
Configuring the same value in different fields leads to multiple samples of the same channel during the conversion sequence. This can be done consecutively, or not, according to user needs.
Example: for each trigger event, to obtain the “CH3 CH1 CH0 CH4 CH4” conversion sequence, use the following settings:
ADC_SEQR1.USCH1=3, ADC_CHSR.CH0=1
ADC_SEQR1.USCH2=1, ADC_CHSR.CH1=1
ADC_SEQR1.USCH3=0, ADC_CHSR.CH2=1
ADC_SEQR1.USCH4=4, ADC_CHSR.CH3=1
ADC_SEQR1.USCH5=4, ADC_CHSR.CH4=1

Downloaded from Arrow.com.

56.7.4 ADC Channel Sequence 2 Register

Name: ADC_SEQR2
Offset: 0x0C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					USCH11[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	USCH10[3:0]				USCH9[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0:3, 4:7, 8:11 – USCHx User Sequence Number x
This register can be used only if the ADC_MR.USEQ field is set to ‘1’.
Any USCHx field is processed only if the ADC_CHSR.CHx-1 bit reads logical ‘1’, else any value written in USCHx does not add the corresponding channel in the conversion sequence.
Configuring the same value in different fields leads to multiple samples of the same channel during the conversion sequence. This can be done consecutively, or not, according to user needs.

Downloaded from Arrow.com.

56.7.5 ADC Channel Enable Register

Name: ADC_CHER
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					CH11	CH10	CH9	CH8
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – CHx Channel x Enable
If ADC_MR.USEQ = 1, CHx corresponds to the enable of sequence number x+1 described in ADC_SEQR1 and ADC_SEQR2 (e.g. CH0 enables sequence number USCH1).

Value	Description
0	No effect.
1	Enables the corresponding channel.

Downloaded from Arrow.com.

56.7.6 ADC Channel Disable Register

Name: ADC_CHDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					CH11	CH10	CH9	CH8
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – CHx Channel x Disable

WARNING If the corresponding channel is disabled during a conversion or if it is disabled and then reenabled during a conversion, its associated data and corresponding EOCx and GOVRE flags in ADC_ISR and OVREx flags in ADC_OVER are unpredictable

Value	Description
0	No effect.
1	Disables the corresponding channel.

Downloaded from Arrow.com.

56.7.7 ADC Channel Status Register

Name: ADC_CHSR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					CH11	CH10	CH9	CH8
Access					W	W	W	W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – CHx Channel x Status

Value	Description
0	The corresponding channel (or part of sequence, see ADC_SEQyR.USCHx field) is disabled.
1	The corresponding channel (or part of sequence, see ADC_SEQyR.USCHx field) is enabled. As an example, when ADC_MR.USEQ=1 and ADC_CHSR.CH2=1, the channel configured in ADC_SEQ1R.USCH3 is part of the sequence of conversions.

Downloaded from Arrow.com.

56.7.8 ADC Last Converted Data Register

Name: ADC_LCDR
Offset: 0x20
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
				CHNBOSR[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LDATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 28:24 – CHNBOSR[4:0] Channel Number in Oversampling Mode
Indicates the last converted channel when the ADC_EMR.TAG bit is set and the ADC_EMR0.OSR field is not equal to 0. If the ADC_EMR.TAG bit is not set, CHNBOSR = 0.

Bits 15:0 – LDATA[15:0] Last Data Converted
The analog-to-digital conversion data is placed into this register at the end of a conversion and remains until a new conversion is completed.
If OSR = 0 and TAG = 1 in ADC_EMR, the 4 MSBs of LDATA carry the channel number to obtain a packed system memory buffer made of 1 converted data stored in a halfword (16-bit) instead of 1 converted data in a 32-bit word, thus dividing by 2 the size of the memory buffer.

Downloaded from Arrow.com.

56.7.9 ADC Interrupt Enable Register

Name: ADC_IER
Offset: 0x24
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the ADC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
		NOPEN	PEN			COMPE	GOVRE	DRDY
Access		W	W			W	W	W
Reset		–	–			–	–	–

Bit	23	22	21	20	19	18	17	16
		PRDY	YRDY	XRDY	LCCHG			
Access		W	W	W	W			
Reset		–	–	–	–			

Bit	15	14	13	12	11	10	9	8
					EOC11	EOC10	EOC9	EOC8
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 30 – NOPEN No Pen Contact Interrupt Enable

Bit 29 – PEN Pen Contact Interrupt Enable

Bit 26 – COMPE Comparison Event Interrupt Enable

Bit 25 – GOVRE General Overrun Error Interrupt Enable

Bit 24 – DRDY Data Ready Interrupt Enable

Bit 22 – PRDY Touchscreen Measure Pressure Ready Interrupt Enable

Bit 21 – YRDY Touchscreen Measure YPOS Ready Interrupt Enable

Bit 20 – XRDY Touchscreen Measure XPOS Ready Interrupt Enable

Bit 19 – LCCHG Last Channel Change Interrupt Enable

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – EOCx End of Conversion Interrupt Enable x

Downloaded from Arrow.com.

56.7.10 ADC Interrupt Disable Register

Name: ADC_IDR
Offset: 0x28
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the ADC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
		NOPEN	PEN			COMPE	GOVRE	DRDY
Access		W	W			W	W	W
Reset		–	–			–	–	–

Bit	23	22	21	20	19	18	17	16
		PRDY	YRDY	XRDY	LCCHG			
Access		W	W	W	W			
Reset		–	–	–	–			

Bit	15	14	13	12	11	10	9	8
					EOC11	EOC10	EOC9	EOC8
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 30 – NOPEN No Pen Contact Interrupt Disable

Bit 29 – PEN Pen Contact Interrupt Disable

Bit 26 – COMPE Comparison Event Interrupt Disable

Bit 25 – GOVRE General Overrun Error Interrupt Disable

Bit 24 – DRDY Data Ready Interrupt Disable

Bit 22 – PRDY Touchscreen Measure Pressure Ready Interrupt Disable

Bit 21 – YRDY Touchscreen Measure YPOS Ready Interrupt Disable

Bit 20 – XRDY Touchscreen Measure XPOS Ready Interrupt Disable

Bit 19 – LCCHG Last Channel Change Interrupt Disable

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – EOCx End of Conversion Interrupt Disable x

Downloaded from Arrow.com.

56.7.11 ADC Interrupt Mask Register

Name: ADC_IMR
Offset: 0x2C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
		NOPEN	PEN			COMPE	GOVRE	DRDY
Access		R	R			R	R	R
Reset		0	0			0	0	0

Bit	23	22	21	20	19	18	17	16
		PRDY	YRDY	XRDY	LCCHG			
Access		R	R	R	R			
Reset		0	0	0	0			

Bit	15	14	13	12	11	10	9	8
					EOC11	EOC10	EOC9	EOC8
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 30 – NOPEN No Pen Contact Interrupt Mask

Bit 29 – PEN Pen Contact Interrupt Mask

Bit 26 – COMPE Comparison Event Interrupt Mask

Bit 25 – GOVRE General Overrun Error Interrupt Mask

Bit 24 – DRDY Data Ready Interrupt Mask

Bit 22 – PRDY Touchscreen Measure Pressure Ready Interrupt Mask

Bit 21 – YRDY Touchscreen Measure YPOS Ready Interrupt Mask

Bit 20 – XRDY Touchscreen Measure XPOS Ready Interrupt Mask

Bit 19 – LCCHG Last Channel Change Interrupt Disable

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – EOCx End of Conversion Interrupt Mask x

Downloaded from Arrow.com.

56.7.12 ADC Interrupt Status Register

Name: ADC_ISR
Offset: 0x30
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	PENS	NOPEN	PEN			COMPE	GOVRE	DRDY
Access	R	R	R			R	R	R
Reset	0	0	0			0	0	0

Bit	23	22	21	20	19	18	17	16
		PRDY	YRDY	XRDY	LCCHG			
Access		R	R	R	R			
Reset		0	0	0	0			

Bit	15	14	13	12	11	10	9	8
					EOC11	EOC10	EOC9	EOC8
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 31 – PENS Pen Detect Status

Value	Description
0	The pen does not press the screen.
1	The pen presses the screen. Note: PENS is not a source of interruption.

Bit 30 – NOPEN No Pen Contact (cleared on read)

Value	Description
0	No loss of pen contact since the last read of ADC_ISR.
1	At least one loss of pen contact since the last read of ADC_ISR.

Bit 29 – PEN Pen contact (cleared on read)

Value	Description
0	No pen contact since the last read of ADC_ISR.
1	At least one pen contact since the last read of ADC_ISR.

Bit 26 – COMPE Comparison Event (cleared on read)

Value	Description
0	No comparison event since the last read of ADC_ISR.
1	At least one comparison event (defined in ADC_EMR and ADC_CWR) has occurred since the last read of ADC_ISR.

Bit 25 – GOVRE General Overrun Error (cleared on read)

Value	Description
0	No general overrun error occurred since the last read of ADC_ISR.
1	At least one general overrun error has occurred since the last read of ADC_ISR.

Bit 24 – DRDY Data Ready (automatically set / cleared)

Downloaded from Arrow.com.

Value	Description
0	No data has been converted since the last read of ADC_LCDR.
1	At least one data has been converted and is available in ADC_LCDR.

Bit 22 – PRDY Touchscreen Pressure Measure Ready (cleared on read)

Value	Description
0	No measure has been performed since the last read of ADC_PRESSR.
1	At least one measure has been performed since the last read of ADC_ISR.

Bit 21 – YRDY Touchscreen YPOS Measure Ready (cleared on read)

Value	Description
0	No measure has been performed since the last read of ADC_YPOSR.
1	At least one measure has been performed since the last read of ADC_ISR.

Bit 20 – XRDY Touchscreen XPOS Measure Ready (cleared on read)

Value	Description
0	No measure has been performed since the last read of ADC_XPOSR.
1	At least one measure has been performed since the last read of ADC_ISR.

Bit 19 – LCCHG Last Channel Change (cleared on read)

Value	Description
0	There is no comparison match (defined in the Last Channel Compare Window register (ADC_LCCWR) since the last read of ADC_ISR.
1	The converted value reported on ADC_CDR11 has changed since the last read of ADC_ISR, according to what is defined in ADC_LCTMR and ADC_LCCWR.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – EOCx End of Conversion x (automatically set / cleared)

Value	Description
0	The corresponding analog channel is disabled, or the conversion is not finished. This flag is cleared when reading the corresponding ADC_CDRx registers.
1	The corresponding analog channel is enabled and conversion is complete.

Downloaded from Arrow.com.

56.7.13 ADC Last Channel Trigger Mode Register

Name: ADC_LCTMR
Offset: 0x34
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			CMPMOD[1:0]					DUALTRIG
Access			R/W	R/W				R/W
Reset			0	0				0

Bits 5:4 – CMPMOD[1:0] Last Channel Comparison Mode

Value	Name	Description
0	LOW	Generates the ADC_ISR.LCCHG flag when the converted data is lower than the low threshold of the window.
1	HIGH	Generates the ADC_ISR.LCCHG flag when the converted data is higher than the high threshold of the window.
2	IN	Generates the ADC_ISR.LCCHG flag when the converted data is in the comparison window.
3	OUT	Generates the ADC_ISR.LCCHG flag when the converted data is out of the comparison window.

Bit 0 – DUALTRIG Dual Trigger ON

Value	Description
0	All channels are triggered by event defined by ADC_MR.TRGSEL.
1	Last channel (higher index) trigger period is defined by RTC_MR.OUT1.

Downloaded from Arrow.com.

56.7.14 ADC Last Channel Compare Window Register

Name: ADC_LCCWR
Offset: 0x38
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
					HIGHTHRES[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HIGHTHRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					LOWTHRES[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LOWTHRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – HIGHTHRES[11:0] High Threshold
High threshold associated to compare settings of ADC_LCTMR.

Bits 11:0 – LOWTHRES[11:0] Low Threshold
Low threshold associated to compare settings of ADC_LCTMR.

Downloaded from Arrow.com.

56.7.15 ADC Overrun Status Register

Name: ADC_OVER
Offset: 0x3C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					OVRE11	OVRE10	OVRE9	OVRE8
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OVRE7	OVRE6	OVRE5	OVRE4	OVRE3	OVRE2	OVRE1	OVRE0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – OVREx Overrun Error x

Value	Description
0	No overrun error on the corresponding channel since the last read of ADC_OVER.
1	An overrun error has occurred on the corresponding channel since the last read of ADC_OVER. **Note:** An overrun error does not always mean that the unread data has been replaced by a new valid data. See 56.6.13 Enhanced Resolution Mode and Digital Averaging Function for details.

Downloaded from Arrow.com.

56.7.16 ADC Extended Mode Register

Name: ADC_EMR
Offset: 0x40
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
			ADCMODE[1:0]			SIGNMODE[1:0]		TAG
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
		TRACKX4	SRCCLK	ASTE		OSR[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
			CMPFILTER[1:0]				CMPALL	
Access			R/W	R/W			R/W	
Reset			0	0			0	

Bit	7	6	5	4	3	2	1	0
	CMPSEL[3:0]					CMPTYPE	CMPMODE[1:0]	
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bits 29:28 – ADCMODE[1:0] ADC Running Mode
See 56.6.14 Automatic Error Correction for details on ADC running mode.

Value	Name	Description
0	NORMAL	Normal mode of operation.
1	OFFSET_ERROR	Offset Error mode to measure the offset error. See table ADC Running Modes.
2	GAIN_ERROR_HIGH	Gain Error mode to measure the gain error. See table ADC Running Modes.
3	GAIN_ERROR_LOW	Gain Error mode to measure the gain error. See table ADC Running Modes.

Bits 26:25 – SIGNMODE[1:0] Sign Mode
If conversion results are signed and resolution is below 16 bits, the sign is extended up to the bit 15 (for example, 0xF43 for 12-bit resolution will be read as 0xFF43 and 0x467 will be read as 0x0467). See 56.6.6 Conversion Results Format.

Value	Name	Description
0	SE_UNSG_DF_SIGN	Single-Ended channels: Unsigned conversions. Pseudo-differential channels and Differential channels: Signed conversions.
1	SE_SIGN_DF_UNSG	Single-Ended channels: Signed conversions. Pseudo-differential channels and Differential channels: Unsigned conversions.
2	ALL_UNSIGNED	All channels: Unsigned conversions.
3	ALL_SIGNED	All channels: Signed conversions.

Bit 24 – TAG Tag of ADC_LCDR

Value	Description
0	Sets ADC_LCDR.CHNB field to zero.
1	Appends the channel number to the conversion result in ADC_LCDR.

Bit 22 – TRACKX4 Tracking Time x4

Downloaded from Arrow.com.

Value	Description
0	The ADC_MR.TRACKTIM field effect is multiplied by 1.
1	The ADC_MR.TRACKTIM field effect is multiplied by 4.

Bit 21 – SRCCLK External Clock Selection
0 (PERIPH_CLK): The peripheral clock is the source for the ADC prescaler.
1 (GCLK): GCLK is the source clock for the ADC prescaler, thus the ADC clock can be independent of the core/peripheral clock.

Bit 20 – ASTE Averaging on Single Trigger Event

Value	Name	Description
0	MULTI_TRIG_AVERAGE	The average requests several trigger events.
1	SINGLE_TRIG_AVERAGE	The average requests only one trigger event.

Bits 18:16 – OSR[2:0] Over Sampling Rate

Value	Name	Description
0	NO_AVERAGE	No averaging. ADC sample rate is maximum.
1	OSR4	1-bit enhanced resolution by averaging. ADC sample rate divided by 4.
2	OSR16	2-bit enhanced resolution by averaging. ADC sample rate divided by 16.
3	OSR64	3-bit enhanced resolution by averaging. ADC sample rate divided by 64.
4	OSR256	4-bit enhanced resolution by averaging. ADC sample rate divided by 256.

Bits 13:12 – CMPFILTER[1:0] Compare Event Filtering
Number of consecutive compare events necessary to raise the flag = CMPFILTER+1
When programmed to 0, the flag rises as soon as an event occurs.
See 56.6.9 Comparison Window when using the filtering option (CMPFILTER > 0).

Bit 9 – CMPALL Compare All Channels

Value	Description
0	Only channel indicated in CMPSEL field is compared.
1	All channels are compared.

Bits 7:4 – CMPSEL[3:0] Comparison Selected Channel
If CMPALL = 0: CMPSEL indicates which channel has to be compared.
If CMPALL = 1: No effect.

Bit 2 – CMPTYPE Comparison Type

Value	Name	Description
0	FLAG_ONLY	Any conversion is performed and comparison function drives the ADC_ISR.COMPE flag.
1	START_CONDITION	Comparison conditions must be met to start the storage of all conversions until the ADC_CR.CMPRST bit is set.

Bits 1:0 – CMPMODE[1:0] Comparison Mode

Value	Name	Description
0	LOW	When the converted data is lower than the low threshold of the window, generates the ADC_ISR.COMPE flag or, in Partial Wakeup mode, defines the conditions to exit system from Wait mode.
1	HIGH	When the converted data is higher than the high threshold of the window, generates the ADC_ISR.COMPE flag or, in Partial Wakeup mode, defines the conditions to exit system from Wait mode.
2	IN	When the converted data is in the comparison window, generates the ADC_ISR.COMPE flag or, in Partial Wakeup mode, defines the conditions to exit system from Wait mode.
3	OUT	When the converted data is out of the comparison window, generates the ADC_ISR.COMPE flag or, in Partial Wakeup mode, defines the conditions to exit system from Wait mode.

Downloaded from Arrow.com.

56.7.17 ADC Compare Window Register

Name: ADC_CWR
Offset: 0x44
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	HIGHTHRES[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HIGHTHRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LOWTHRES[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LOWTHRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – HIGHTHRES[15:0] High Threshold
High threshold associated to compare settings of ADC_EMR.

Bits 15:0 – LOWTHRES[15:0] Low Threshold
Low threshold associated to compare settings of ADC_EMR.

Downloaded from Arrow.com.

56.7.18 Channel Configuration Register

Name: ADC_CCR
Offset: 0x4C
Reset: 0
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
					DIFF11	DIFF10	DIFF9	DIFF8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIFF7	DIFF6	DIFF5	DIFF4	DIFF3	DIFF2	DIFF1	DIFF0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27 – DIFFx Differential Inputs for Channel x

Value	Description
0	Corresponding channel is set in Single-ended mode.
1	Corresponding channel is set in Differential mode.

Downloaded from Arrow.com.

56.7.19 ADC Channel Data Register

Name: ADC_CDRx
Offset: 0x50 + x*0x04 [x=0..11]
Reset: 0
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – DATA[15:0] Converted Data
The analog-to-digital conversion data is placed into this register at the end of a conversion and remains until a new conversion is completed. ADC_CDRx is only loaded if the corresponding analog channel is enabled.

Downloaded from Arrow.com.

56.7.20 ADC Analog Control Register

Name: ADC_ACR
Offset: 0x94
Reset: 0x00001200
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

By default, bits 12 and 13 are set to 1 and 0, respectively, and must not be modified.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access							IBCTL[1:0]	
Reset							1	0

Bit	7	6	5	4	3	2	1	0
Access							PENDETSENS[1:0]	
Reset							0	0

Bits 9:8 – IBCTL[1:0] ADC Bias Current Control
Adapts performance versus power consumption. Refer to the “Electrical Characteristics” section for further details.

Bits 1:0 – PENDETSENS[1:0] Pen Detection Sensitivity
Modifies the pen detection input pull-up resistor value. Refer to the “Electrical Characteristics” section for further details.

Downloaded from Arrow.com.

56.7.21 ADC Pseudo-Differential Register

Name: ADC_PDR
Offset: 0xA0
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PDIFF15	PDIFF14	PDIFF13	PDIFF12	PDIFF11	PDIFF10	PDIFF9	PDIFF8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PDIFF7	PDIFF6	PDIFF5	PDIFF4	PDIFF3	PDIFF2	PDIFF1	PDIFF0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – PDIFFx Pseudo-Differential Inputs for Channel x

Value	Description
0	The channel is configured as defined by the ADC_CCR.DIFFx bit.
1	The channel is configured in Pseudo-differential mode.

Downloaded from Arrow.com.

56.7.22 ADC Touchscreen Mode Register

Name: ADC_TSMR
Offset: 0xB0
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	PENDBC[3:0]							PENDET
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit	23	22	21	20	19	18	17	16
		NOTSDMA			TSSCTIM[3:0]			
Access		R/W			R/W	R/W	R/W	R/W
Reset		0			0	0	0	0

Bit	15	14	13	12	11	10	9	8
					TSFREQ[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
			TSAV[1:0]				TSMODE[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 31:28 – PENDBC[3:0] Pen Detect Debouncing Period
Debouncing period = 2^{PENDBC} ADCCLK periods.

Bit 24 – PENDET Pen Contact Detection Enable
When PENDET = 1, XPOS, YPOS, Z1, Z2 values of ADC_XPOSR, ADC_YPOSR, ADC_PRESSR are automatically cleared when ADC_ISR.PENS = 0.

Value	Description
0	Pen contact detection disabled.
1	Pen contact detection enabled.

Bit 22 – NOTSDMA No TouchScreen DMA

Value	Description
0	XPOS, YPOS, Z1, Z2 are transmitted in ADC_LCDR.
1	XPOS, YPOS, Z1, Z2 are never transmitted in ADC_LCDR, therefore the buffer does not contains touchscreen values.

Bits 19:16 – TSSCTIM[3:0] Touchscreen Switches Closure Time
Defines closure time of analog switches necessary to establish the measurement conditions.
The closure time is:
Switch Closure Time = (TSSCTIM × 4) ADCCLK periods.

Bits 11:8 – TSFREQ[3:0] Touchscreen Frequency
Defines the touchscreen frequency compared to the trigger frequency.
TSFREQ must be greater or equal to TSAV.
The touchscreen frequency is:
Touchscreen Frequency = Trigger Frequency / 2^{TSFREQ}

Bits 5:4 – TSAV[1:0] Touchscreen Average

Downloaded from Arrow.com.

Value	Name	Description
0	NO_FILTER	No Filtering. Only one ADC conversion per measure
1	AVG2CONV	Averages 2 ADC conversions
2	AVG4CONV	Averages 4 ADC conversions
3	AVG8CONV	Averages 8 ADC conversions

Bits 1:0 – TSMODE[1:0] Touchscreen Mode
When TSMOD equals 01 or 10 (i.e., 4-wire mode), channels 0, 1, 2 and 3 must not be used for classic ADC conversions. When TSMOD equals 11 (i.e., 5-wire mode), channels 0, 1, 2, 3, and 4 must not be used.

Value	Name	Description
0	NONE	No Touchscreen
1	4_WIRE_NO_PM	4-wire Touchscreen without pressure measurement
2	4_WIRE	4-wire Touchscreen with pressure measurement
3	5_WIRE	5-wire Touchscreen

Downloaded from Arrow.com.

56.7.23 ADC Touchscreen X Position Register

Name: ADC_XPOSR
Offset: 0xB4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
					XSCALE[11:8]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	XSCALE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					XPOS[11:8]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	XPOS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – XSCALE[11:0] Scale of XPOS
Indicates the max value that XPOS can reach. This value should be close to 2^{12}.

Bits 11:0 – XPOS[11:0] X Position
The position measured is stored here. If XPOS = 0 or XPOS = XSIZE, the pen is on the border.
When pen detection is enabled (ADC_TSMR.PENDET set to '1'), XPOS is tied to 0 while there is no detection of contact on the touchscreen (i.e., when the ADC_ISR.PENS bit is cleared).

Downloaded from Arrow.com.

56.7.24 ADC Touchscreen Y Position Register

Name: ADC_YPOSR
Offset: 0xB8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
					YSCALE[11:8]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	YSCALE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					YPOS[11:8]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	YPOS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – YSCALE[11:0] Scale of YPOS
Indicates the max value that YPOS can reach. This value should be close to 2^{12}.

Bits 11:0 – YPOS[11:0] Y Position
The position measured is stored here. If YPOS = 0 or YPOS = YSIZE, the pen is on the border.
When pen detection is enabled (ADC_TSMR.PENDET set to '1'), YPOS is tied to 0 while there is no detection of contact on the touchscreen (i.e., when the ADC_ISR.PENS bit is cleared).

Downloaded from Arrow.com.

56.7.25 ADC Touchscreen Pressure Register

Name: ADC_PRESSR
Offset: 0xBC
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
					Z2[11:8]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	Z2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					Z1[11:8]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	Z1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – Z2[11:0] Data of Z2 Measurement
Data Z2 necessary to calculate pen pressure.
When pen detection is enabled (ADC_TSMR.PENDET set to ‘1’), Z2 is tied to 0 while there is no detection of contact on the touchscreen (i.e., when the ADC_ISR.PENS bit is cleared).
Note: These two values are unavailable if ADC_TSMR.TSMODE is not set to 2.

Bits 11:0 – Z1[11:0] Data of Z1 Measurement
Data Z1 necessary to calculate pen pressure.
When pen detection is enabled (ADC_TSMR.PENDET set to ‘1’), Z1 is tied to 0 while there is no detection of contact on the touchscreen (i.e., when the ADC_ISR.PENS bit is cleared).

Downloaded from Arrow.com.

56.7.26 ADC Trigger Register

Name: ADC_TRGR
Offset: 0xC0
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	TRGPER[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TRGPER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						TRGMOD[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:16 – TRGPER[15:0] Trigger Period
Effective only if TRGMOD defines a periodic trigger.
Defines the periodic trigger period, with the following equation:
Trigger Period = (TRGPER + 1) / ADCCLK
The minimum time between two consecutive trigger events must be strictly greater than the duration time of the longest conversion sequence depending on the configuration of registers ADC_MR, ADC_CHSR, ADC_SEQRx, ADC_TSMR.
When TRGMOD is set to pen detect trigger (i.e., 100) and averaging is used (i.e., field TSAV ≠ 0 in ADC_TSMR) only one measure is performed. Thus, XRDY, YRDY, PRDY, DRDY will not rise on pen contact trigger. To achieve measurement, several triggers must be provided either by software or by setting the TRGMOD on continuous trigger (i.e., 110) until flags rise.

Bits 2:0 – TRGMOD[2:0] Trigger Mode

Value	Name	Description
0	NO_TRIGGER	No hardware trigger enabled, only software trigger can start conversions
1	EXT_TRIG_RISE	Rising edge of the selected hardware trigger event, defined in ADC_MR.TRGSEL
2	EXT_TRIG_FALL	Falling edge of the selected hardware trigger event
3	EXT_TRIG_ANY	Any edge of the selected hardware trigger event
4	PEN_TRIG	Pen Detect Trigger (shall be selected only if PENDET is set and TSMODE > 0)
5	PERIOD_TRIG	ADC internal hardware periodic trigger (see field TRGPER)
6	CONTINUOUS	Continuous mode

Downloaded from Arrow.com.

56.7.27 ADC Correction Values Register

Name: ADC_CVR
Offset: 0xD4
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	GAINCORR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GAINCORR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OFFSETCORR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OFFSETCORR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – GAINCORR[15:0] Gain Correction
Gain correction to apply on converted data. Only bits 0 to 15 are relevant (other bits are ignored and read as 0).

Bits 15:0 – OFFSETCORR[15:0] Offset Correction
Offset correction to apply on converted data. The offset is signed (2's complement), only bits 0 to 11 are relevant (other bits are ignored and read as 0).

Downloaded from Arrow.com.

56.7.28 ADC Channel Error Correction Register

Name: ADC_CECR
Offset: 0xD8
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					ECORR11	ECORR10	ECORR9	ECORR8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ECORR7	ECORR6	ECORR5	ECORR4	ECORR3	ECORR2	ECORR1	ECORR0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – ECORRx Error Correction Enable for channel x

Value	Description
0	Automatic error correction is disabled for channel x.
1	Automatic error correction is enabled for channel x.

Downloaded from Arrow.com.

56.7.29 ADC Touchscreen Correction Values Register

Name: ADC_TSCVR
Offset: 0xDC
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	TSGAINCORR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TSGAINCORR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TSOFFSETCORR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TSOFFSETCORR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – TSGAINCORR[15:0] Touchscreen Gain Correction
Gain correction to apply on converted data for the touchscreen channels. Only bits 0 to 15 are relevant (other bits are ignored and read as 0).

Bits 15:0 – TSOFFSETCORR[15:0] Touchscreen Offset Correction
Offset correction to apply on converted data for the touchscreen channels. The offset is signed (2's complement), only bits 0 to 11 are relevant (other bits are ignored and read as 0).

Downloaded from Arrow.com.

56.7.30 ADC Write Protection Mode Register

Name: ADC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						WPCTEN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x414443	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0

Bit 2 – WPCTEN Write Protection Control Enable
See 56.6.19 Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables the write protection on control registers if WPKEY corresponds to 0x414443 (“ADC” in ASCII).
1	Enables the write protection on control registers if WPKEY corresponds to 0x414443 (“ADC” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable
See 56.6.19 Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x414443 (“ADC” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x414443 (“ADC” in ASCII).

Bit 0 – WPEN Write Protection Enable
See 56.6.19 Register Write Protection for the list of write-protected registers.

Value	Description
0	Disables the write protection if WPKEY value corresponds to 0x414443 (“ADC” in ASCII).
1	Enables the write protection if WPKEY value corresponds to 0x414443 (“ADC” in ASCII).

Downloaded from Arrow.com.

56.7.31 ADC Write Protection Status Register

Name: ADC_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of ADC_WPSR.
1	A write protection violation has occurred since the last read of ADC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Downloaded from Arrow.com.

57. Electrical Characteristics

57.1 Electrical Parameters Usage

Tables in sections 57.5 I/O Characteristics, 57.6 Digital Peripheral Characteristics 57.7 Analog Peripheral Characteristics, 57.8 Power Consumption in Active Mode and 57.9 Operation and Power Consumption in Low-Power Modes define the limiting values for several electrical parameters.

- Unless otherwise noted, these values are valid over the junction temperature range T_J= [-40°C + 125°C].
- These limits may be affected by the board on which the device is mounted. In particular, noisy supply and ground conditions must be avoided and care must be taken to provide:
  - a PCB with a low-impedance ground plane. A single unbroken ground plane is a minimum requirement.
  - low-impedance decoupling of the device power supply inputs. A 10 nF to 220 nF Ceramic X7R (or X5R) capacitor placed very close to each power supply input is a minimum requirement. See specific recommendations regarding analog pins or functions in the corresponding sections. To reduce any potential electromagnetic compatibility (EMC) related issues, it is good practice to double this decoupling capacitor whenever possible with a high frequency one, e.g. one 100 pF (C0G or NP0) per power supply input.
  - low-impedance power supply decoupling of external components. This recommendation aims at avoiding large current spikes flowing into the PCB ground and power planes.
- In addition, although the device is specified with wide operating supply ranges on most of its supply inputs (e.g. 1.7V to 3.6V), large and fast supply variations may lead to unpredictable device behavior including, but not limited to, out-of-specification operation. Therefore, in addition to maintaining the power supply inputs within their specified ranges, it is mandatory to keep the power supply variations within the limits shown in the following table during the device operation.

Table 57-1. Maximum Power Supply Variations

Symbol	Parameter	Conditions	Min	Max	Unit
V_N	Peak-to-peak ripple and noise voltage	Applies to: VDDCORE, VDDBU, VDDIOP0, VDDIOP1, VDDQSPI, VDDNF, VDDIOM	–	3	% V_{DC}(1)
		Applies to: VDDIN33, VDDANA	–	1	% V_{DC}(1)
SR	Slewrate of power supply variations	$\Delta V \le 5\%\ V_{DC_MIN}$(2)(3)	–	±50	V/ms
		$\Delta V \le 10\%\ V_{DC_MIN}$	–	±10	
		$\Delta V > 10\%\ V_{DC_MIN}$	–	±1	

Note:

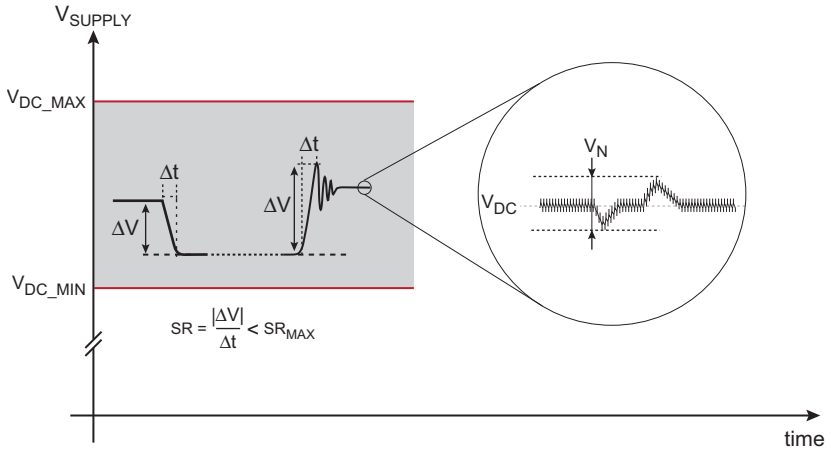
1. V_{DC} is the power supply DC value.
2. V_{DC_MIN} is the minimum operating voltage of the supply input as described in table Recommended Operating Conditions on Power Supply Inputs.
3. ΔV is the variation amplitude. The slewrate specification applies when $\Delta V \ge V_N$.

The following examples and figure illustrate the above table:

- When working with VDDIOP0 = 3.3V, a maximum power supply ripple and noise voltage of 99 mV peak-to-peak (3% of 3.3V) must be respected.
- When working with VDDIN33 = 3.3V, a maximum power supply ripple and noise voltage of 33 mV peak-to-peak (1% of 3.3V) must be respected.

Downloaded from Arrow.com.

Figure 57-1. Maximum Power Supply Variation



57.2 Absolute Maximum Ratings

Table 57-2. Absolute Maximum Ratings

Storage Temperature	-60°C to +150°C	**Note:** Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. **Exposure to absolute maximum rating conditions for extended periods may affect device reliability.**
Voltage Difference between two ground pins (among GND, GNDANA and GNDIN33)	±50 mV	
Voltage on Power Supply Inputs with respect to ground pins:		
VDDCORE	-0.3V to 1.4V	
VDDIOM, VDDIOP0, VDDIOP1	-0.3V to 4.0V	
VDDANA, VDDIN33	-0.3V to 4.0V	
VDDNF, VDDQSPI, VDDBU	-0.3V to 4.0V	
Voltage on Digital or Analog Input Pins with respect to ground	-0.3V to 4.0V	
Injected Current into any input pin	± 1 mA	
Total Injected Current in all input pins of a common power supply pair	± 10 mA	
Maximum DC Output Current:		
On all I/O lines into one power rail	100 mA	
Per output pin	25 mA	

Note: All I/O pins are internally clamped to their respective VDD and GND rails as defined in the Pin Description table, e.g. for PA2, this corresponds to VDDIOP0 and GND.

Downloaded from Arrow.com.

57.3 Recommended Operating Conditions

Table 57-3. Recommended Operating Conditions on Power Supply Inputs

Power Input	Parameter	Conditions	Min	Max	Unit
VDDCORE	Core logic power supply (including CPU, memories and peripherals)	$f_{CPU} \le 420$ MHz, $f_{MCK} \le 140$ MHz $f_{CPU} \le 600$ MHz, $f_{MCK} \le 200$ MHz	1.02 1.12	1.21 1.21	V
VDDIOM	SDRAM I/O lines power supply	SDR-SDRAM	3.00	3.60	V
		LPDDR-SDRAM, LPSDR-SDRAM, DDR2-SDRAM	1.70	1.90	
VDDNF	NAND Flash I/O lines power supply	–	1.70	3.60	V
VDDQSPI	VDDQSPI I/O lines power supply	–	1.70	3.60	V
VDDANA(1)	VDDANA I/O lines, A/D converter, OTP memory power supply	–	3.00	3.60	V
VDDIN33(1)	2.5V regulator power input, USB interface I/O lines power supply	–	3.00	3.60	V
VDDIOP0	VDDIOP0 I/O lines power supply	–	1.70	3.60	V
VDDIOP1	VDDIOP1 I/O lines power supply	–	1.70	3.60	V
VDDBU	Backup domain power supply	–	1.60	3.60	V
t_{R_VDD}	Power supply slope at power-up	Applies to any of the power supply inputs listed above	0.2	20	mV/µs
t_{F_VDD}	Power supply slope at power-down	Applies to any of the power supply inputs listed above	-20	-1	mV/µs

Note:

1. VDDANA and VDDIN33 are powered from one single power source so that : V(VDDANA,VDDIN33) ≤ ±50mV.

Table 57-4. Recommended Operating Conditions on Input Pins (see Note 1)

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IN}	Input line voltage range on inputs(2)(3)	–	-0.3V	V_{DD} + 0.3V	V
I_{IN}	DC current injection on inputs(4)	–	–	±0.2	mA
I_{TOT_INJ}	Total current injection per power rail or per ground rail(5)	–	–	±2	mA

Note:

1. In this table, VDD refers to the voltage of the associated power rail of the I/O line, as defined in the Pin Description table, e.g., for PA2, VDD refers to VDDIOP0.
2. Input voltages $V_{IN} \le 0$V or $V_{IN} \ge$ VDD lead to negative or positive current injection on inputs.
3. For analog inputs (PB6..PB17), input voltages $V_{IN} \ge \min(V_{DDANA}, V_{ADVREFP})$ lead to saturated A/D conversion to 0xFFF.
4. Current injection on A/D converter analog inputs (PB6..17) may degrade the analog performance of the corresponding channel or the analog performance of other analog channels.
5. Corresponds to the sum of the positive currents into one power rail and respectively to the sum of the negative currents into one ground rail as defined in the Pin Description table.

Downloaded from Arrow.com.

Table 57-5. Recommended Thermal Operating Conditions

Symbol	Parameter	Conditions		Min	Max	Unit
T_A	Ambient temperature range	–		-40	105	°C
T_J	Junction temperature range	–		-40	125	
R_{JA}	Junction-to-ambient thermal resistance	BGA228		–	40	°C/W
P_D	Allowable power dissipation	BGA228	T_A = 70°C	–	1.3	W
			T_A = 85°C	–	1.0	
			T_A = 105°C	–	0.5	

Table 57-6. Recommended Operating Conditions on Internal Clocks

Symbol	Parameter	Conditions	Min	Max	Unit
f_{CPU_CLK}	Processor clock (CPU_CLK) frequency	V_{DDCORE} ≥ 1.12V	–	600	MHz
		V_{DDCORE} ≥ 1.02V	–	420	
f_{MCK}	Master clock (MCK) frequency	V_{DDCORE} ≥ 1.12V	–	200	MHz
		V_{DDCORE} ≥ 1.02V	–	140	

Table 57-7. Recommended Operating Conditions on SDRAM Interface

Symbol	Parameter	Conditions	Min	Max	Unit
f_{SDRAM_CLK}	SDRAM clock frequency	V_{DDCORE} ≥ 1.12V			
		16-bit SDR-SDRAM	–	200	MHz
		16-bit LPSDR-SDRAM	–	166	
		16-bit DDR2-SDRAM	125	200	
		16-bit LPDDR-SDRAM	–	200	
		V_{DDCORE} ≥ 1.12V			
		32-bit SDR-SDRAM	–	180	MHz
		32-bit LPSDR-SDRAM	–	158	
		V_{DDCORE} ≥ 1.02V			
		16-bit SDR-SDRAM	–	140	MHz
		16-bit LPSDR-SDRAM	–	140	
		16-bit DDR2-SDRAM	125	140	
		16-bit LPDDR-SDRAM	–	140	

57.4 Recommended Power Supply Sequencing

In the following sections, various recommended power sequences are described. Operating the device outside this scope may lead to unpredictable behavior. In addition, in all modes other than the MPU Backup mode, every power supply input must be powered to operate the device.

Downloaded from Arrow.com.

57.4.1 Power-up and Power-down

At power-up, from a power supply sequencing perspective, the SAM9X60 power supply inputs are categorized into three independent groups:

- VDDCORE
- VDDIN33
- Periphery group containing all other power supply inputs except VDDBU

The figure below shows the recommended power-up sequence. Note the following:

- VDDBU
  - When supplied from a precharged storage element (battery or supercapacitor), VDDBU is an always-on supply input and is therefore not part of the power supply sequencing.
  - When no storage element is used on VDDBU in the application, VDDBU must be tied to VDDIN33.
  - When a supercapacitor is used in the application to power VDDBU during Backup mode, this element must be isolated from VDDBU during its (slow) charge, so that VDDBU closely follows VDDIN33. In table Power-up Timing Specification, the parameter t_0 limits the delay to establish VDDBU after VDDIN33.
- VDDOUT25 is the output of the internal 2.5V VDDOUT25 regulator, therefore, there is no power supply requirement on this pin. VDDOUT25 is automatically started at VDDIN33 rise when VDDIN33 is above its Power-On-Reset threshold.

Figure 57-2. Recommended Power Sequence at Power-up

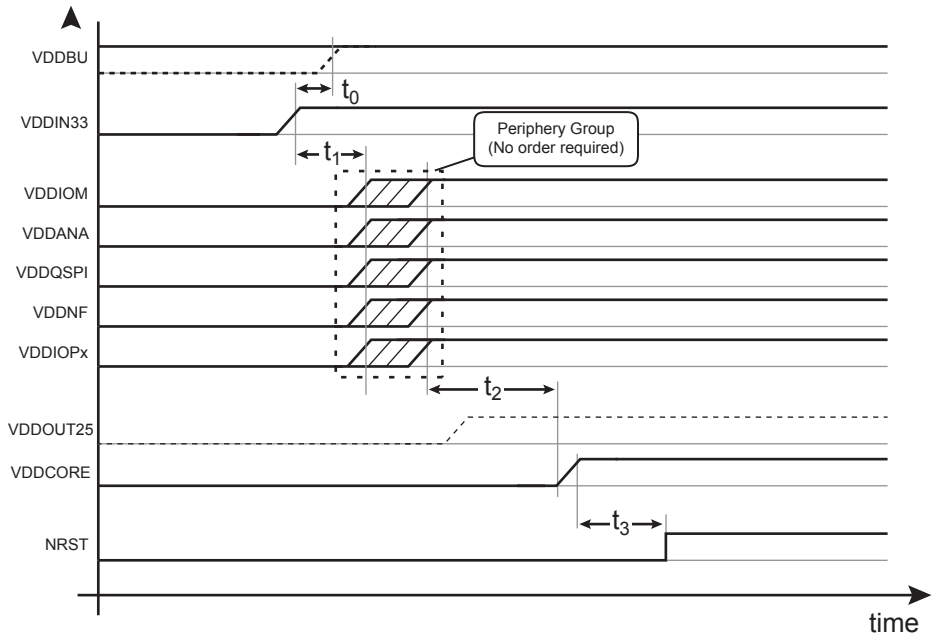


Table 57-8. Power-up Timing Requirements

Symbol	Parameter	Conditions[1]	Min	Max	Unit
t_0	VDDBU Delay	Delay from VDDIN33 established to VDDBU established	–	0.2	ms
t_1	VDDIN33 to Periphery Group delay	Delay from VDDIN33 established to the first Periphery Group supply established	0	–	ms
t_2	Periphery Group to VDDCORE delay	Delay from the last Periphery Group established supply to VDDCORE supply turn-on	0	–	ms
t_3	Reset Delay at power-up	From VDDCORE established to NRST high.	8	–	ms

Downloaded from Arrow.com.

Note:

1. The term "established" refers to a power supply established to 90% of its final value.

The following figure shows the SAM9X60 power-down sequence that starts by asserting the NRST line to 0. Once NRST is asserted, the supply inputs can be immediately shut down without any specific timing or order. VDDBU may not be shut down if the application uses a backup storage element on this supply input.

Figure 57-3. Recommended Power Sequence at Power-down

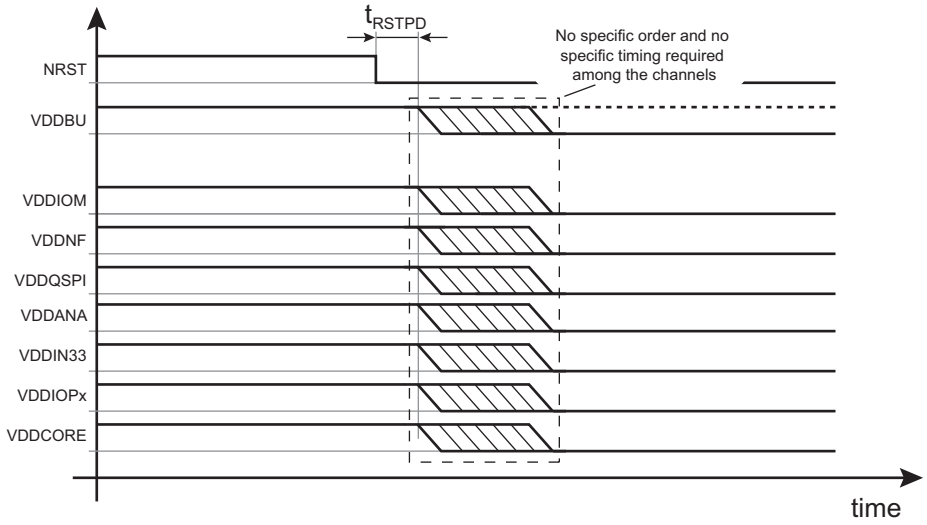


Table 57-9. Power-down Timing Requirements

Symbol	Parameter	Conditions	Min	Max	Unit
t_{RSTPD}	NRST delay at power-down	Delay from NRST low to the first supply out of its operating range	0	–	ms

57.4.2 Backup Mode Entry and Wake-up

The following figure shows the recommended power-down sequence to place the SAM9X60 in Backup mode. The SHDN signal, output of Shutdown Controller (SHDWC), signals the shutdown request to the external power supply. This output is supplied by VDDBU that is present in Backup mode. In a similar way to the power-down sequence, the NRST signal must be asserted prior to turning the SAM9X60 power supplies off.

Downloaded from Arrow.com.

Figure 57-4. Recommended Power Sequence at Backup Mode Entry

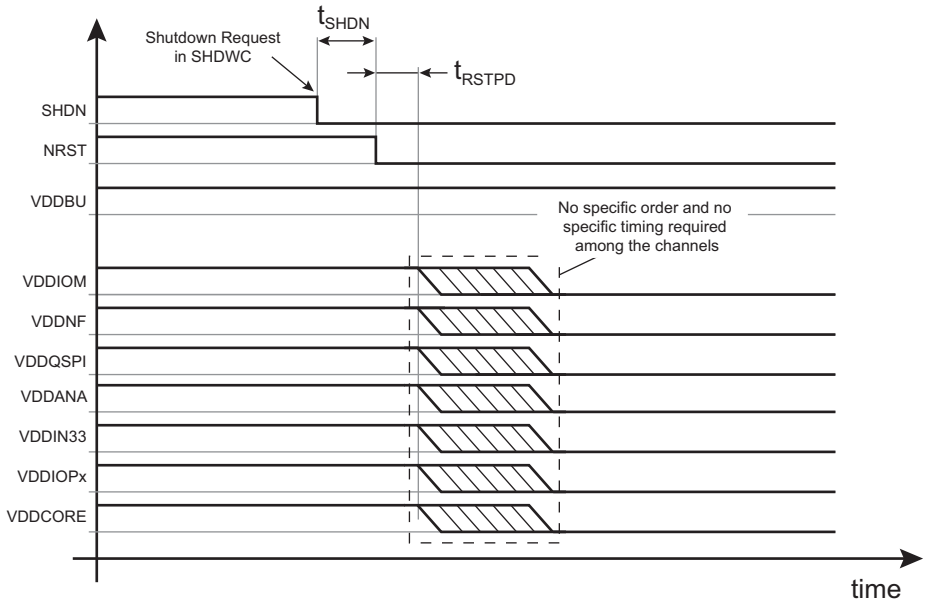


Table 57-10. Backup Mode Entry Timing Requirements

Symbol	Parameter	Conditions[1]	Min	Max	Unit
t_{SHDN}	SHDN Delay at Backup mode entry	Delay from SHDN low to NRST low	0	–	ms
t_{RSTPD}	NRST Delay at power-down	Delay from NRST low to the first supply out of its operating range	0	–	ms

The following figure shows the recommended power-up sequence to wake up SAM9X60 from Backup mode. Upon a wake-up event (WKUP pin, RTC alarm, etc.), the Shutdown Controller automatically toggles its SHDN output back to VDDBU to request the external power supply to restart. This power-up sequence is the same as the one shown in figure Recommended Power Sequence at Power-up. In particular, the supply group's definition is the same.

Downloaded from Arrow.com.

Figure 57-5. Recommended Power Sequence at Wake-up from Backup Mode

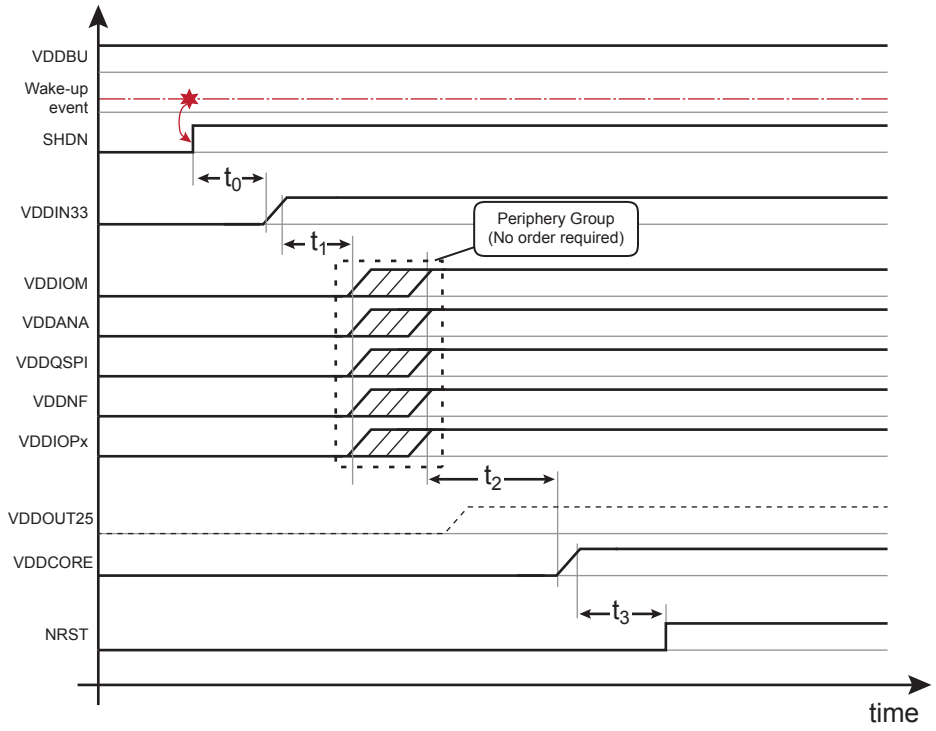


Table 57-11. Wake-up from Backup Mode Timing Requirements

Symbol	Parameter	Conditions	Min	Max	Unit
t_0	VDDIN33 delay	Delay from SHDN high to VDDIN33 turn-on	0	–	ms
t_1	VDDIN33 to periphery group delay	Delay from VDDIN33 established to the first periphery group supply established	0	–	ms
t_2	Periphery group to VDDCORE delay	Delay from the last periphery group established supply to VDDCORE supply turn-on	0	–	ms
t_3	Reset delay at power-up	From VDDCORE established to NRST high	8	–	ms

Note:

1. The term "established" refers to a power supply established to 90% of its final value.

57.5 I/O Characteristics

The device features two types of Input/Output (I/O) circuits:

- GPIO
- DDRIO

Unless otherwise specified:

- The following specifications apply to:
  - the GPIO type
  - the DDRIO type when these I/Os are not driven by the MPDDRC
- V_{DD} refers to the voltage of the associated power rail of the I/O line, as defined in the Pin Description table, e.g., for PA2, V_{DD} refers to the voltage applied on VDDIOP0.

The DRIVE and SLEWRATE controls for the GPIO type are set in PIO_DRIVE and PIO_SLEWR registers, respectively. The DRIVE control for the DDRIO type is set in the SFR_CCFG_EBICSA register.

Downloaded from Arrow.com.

57.5.1 I/O DC Characteristics

Table 57-12. Input DC Characteristics

Symbol	Parameter	Conditions		Min	Max	Unit
V_{IL}	Low-level input voltage(1)	–		–	0.3 x V_{DD}	V
V_{IH}	High-level input voltage(1)	–		0.7 x V_{DD}	–	V
I_{IH}	Input-high input leakage current(1)	Pull-down resistor disabled; V_{IN} = V_{DD} = 3.6V	$T_J \le 30°C$	-50	50	nA
			$T_J \le 125°C$	-150	150	
I_{IL}	Input-low input leakage current(1)	Pull-up resistor disabled; V_{IN} = 0V, V_{DD} = 3.6V	$T_J \le 30°C$	-50	50	nA
			$T_J \le 125°C$	-150	150	
R_{PULL}	Programmable pull-up or pull-down resistor	Digital input mode		60	140	kΩ
C_{IN}	Input capacitance(1)	–		–	4	pF
V_{hys}	Input hysteresis(1)	–		0.15	–	V

Note:

1. Characterization and/or simulation data.

Table 57-13. Output DC Characteristics (1.7V < V_{DD} < 1.9V)

Symbol	Parameter	I/O Type	Conditions	Min	Max	Unit
I_{OL} or I_{OH}	Low-level or high-level output current	Any	I_{OL}: V_{OL} = 0.2 x V_{DD} I_{OH}: V_{OH} = 0.8 x V_{DD}	–	–	–
		GPIO	Drive = 0, Slewrate = 0	–	3	mA
			Drive = 1, Slewrate = 0		4	
			Drive = 0, Slewrate = 1		1	
			Drive = 1, Slewrate = 1		2	
		DDRIO	Drive = 0	–	3	
			Drive = 1		4	

Table 57-14. Output DC Characteristics (3.0V < V_{DD} < 3.6V)

Symbol	Parameter	I/O Type	Conditions	Min	Max	Unit
I_{OL} or I_{OH}	Low-level/high-level output current	Any	I_{OL}: V_{OL} = 0.2 x V_{DD} I_{OH}: V_{OH} = 0.8 x V_{DD}	–	–	–
		GPIO	Drive = 0, Slewrate = 0	–	9	mA
			Drive = 1, Slewrate = 0		10	
			Drive = 0, Slewrate = 1		3	
			Drive = 1, Slewrate = 1		6	
		DDRIO	Drive = 0	–	8	
			Drive = 1		10	

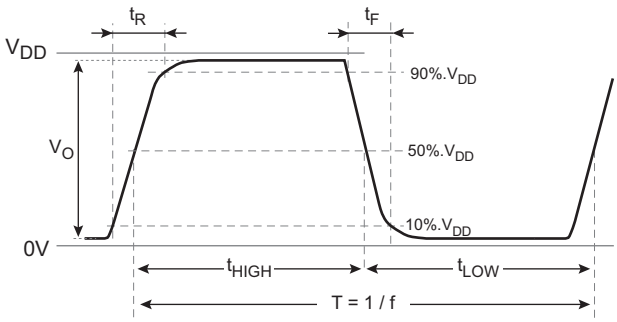
Downloaded from Arrow.com.

57.5.2 I/O AC Characteristics

57.5.2.1 Output Driver AC Characteristics

The timing definitions necessary to specify the maximum operating frequency of an output driver are given in the following figure.

Figure 57-6. Timing Definitions of a Digital Output Signal



T: Period of the digital output signal

f = 1 / T: Frequency of the digital output signal

t_{HIGH}: Time during which the output waveform is greater than V_{DD} / 2

t_{LOW} = T - t_{HIGH}: Time during which the output waveform is less than V_{DD} / 2

d = t_{HIGH} / T: Output waveform duty cycle

V_O: Output waveform amplitude

In tables Output Driver AC Characteristics (1.7V < VDD < 1.9V) and Output Driver AC Characteristics (3.0V < VDD < 3.6V), the maximum operating frequency f_{MAX} ensures that the driver's output waveform fulfills all the following characteristics:

- t_R < 0.75 / f_{MAX} and t_F < 0.75 / f_{MAX}
- d: the duty cycle of the output waveform is between 45% and 55%
- V_O: the output waveform amplitude is greater than 95% VDD

The f_{MAX} parameter indicates the speed limit of an output driver across various operating conditions: supply voltage range, load capacitance, drive strength and slewrate programming. The effective maximum output frequency of a specific output line may be limited by the peripheral that drives this line.

Tables Output Driver AC Characteristics (1.7V < VDD < 1.9V) and Output Driver AC Characteristics (3.0V < VDD < 3.6V) give the AC output characteristics of the output drivers in the following conditions:

- Output load: 10 pF capacitor to ground
- Two VDD ranges:
  - 1.7V < VDD < 1.9V
  - 3.0V < VDD < 3.6V
- Two Drive settings: 0 and 1
- Two Slewrate settings for the GPIO type: 0 and 1. The DDRIO drivers do not have slewrate settings and rather use an autocalibration setting. See 6.2.4.1 DDR/SDR I/O Calibration.

For the GPIO drivers, the table below gives the recommended Drive and Slewrate settings depending on the output switching frequency and the two commonly used VDD ranges (1.8V and 3.3V). Other settings are possible but they may lead to excessively fast rise and fall times (t_R, t_F), with a potentially negative impact on the electromagnetic emissions of the application.

For the DDRIO drivers, it is recommended to use Drive = 0 if the VDD range is 3.0V to 3.6V, and Drive = 1 if the VDD range is 1.7V to 1.9V.

Downloaded from Arrow.com.

Table 57-15. Recommended GPIO Drive and Slewrate Settings versus GPIO Use Case

VDD Range	Low-speed $f_{GPIO} \leq 50$ MHz(1)	High-speed 50 MHz $\leq f_{GPIO} \leq 150$ MHz(1)
1.7V – 1.9V	Drive = 1, Slewrate = 1	Drive = 1, Slewrate = 0
3.0V – 3.6V	Drive = 0, Slewrate = 1	Drive = 0, Slewrate = 0

Note:

1. This is an indicative value. Refer to tables Output Driver AC Characteristics (1.7V < VDD < 1.9V) and Output Driver AC Characteristics (3.0V < VDD < 3.6V) for accurate maximum frequency specifications.

Table 57-16. Output Driver AC Characteristics (1.7V < V_{DD} < 1.9V)

Symbol	Parameter	I/O Type	Conditions	Min	Max	Unit
t_R or t_F	Rise or fall time(1)(4)	Any	C_L = 10 pF	–	–	–
		GPIO	Drive = 0, Slewrate = 0	0.9	4.0	ns
			Drive = 1, Slewrate = 0	0.8	2.8	
			Drive = 0, Slewrate = 1	3.0	12.0	
			Drive = 1, Slewrate = 1	1.6	5.6	
		DDRIO	Drive = 0	1.2	3.1	
			Drive = 1	1.0	2.6	
f_{MAX}	Maximum frequency(2)(3)(4)	Any	C_L = 10 pF	–	–	–
		GPIO	Drive = 0, Slewrate = 0	90	–	MHz
			Drive = 1, Slewrate = 0	130	–	
			Drive = 0, Slewrate = 1	25	–	
			Drive = 1, Slewrate = 1	50	–	
		DDRIO	Drive = 0	90	–	
			Drive = 1	100	–	

Note:

1. Measured between V_O = 10% V_{DD} and V_O= 90% V_{DD}.
2. f_{MAX} = 0.75 / (t_R+t_F). Frequency numbers are rounded for legibility.
3. f_{MAX} may be limited by the peripheral that drives the I/O line.
4. Characterization and/or simulation data.

Table 57-17. Output Driver AC Characteristics (3.0V < V_{DD} < 3.6V)

Symbol	Parameter	I/O Type	Conditions	Min	Max	Unit
t_R or t_F	Rise or fall time(1)(4)	Any	C_L = 10 pF	–	–	–
		GPIO	Drive = 0, Slewrate = 0	0.9	2.0	ns
			Drive = 1, Slewrate = 0	0.8	1.8	
			Drive = 0, Slewrate = 1	3.0	6.6	
			Drive = 1, Slewrate = 1	1.6	3.3	
		DDRIO	Drive = 0	0.9	2.0	
			Drive = 1	0.8	1.7	

Downloaded from Arrow.com.

.....continued

Symbol	Parameter	I/O Type	Conditions	Min	Max	Unit
f_{MAX}	Maximum frequency(2)(3)(4)	Any	C_L = 10 pF	–	–	–
		GPIO	Drive = 0, Slewrate = 0	150	–	MHz
			Drive = 1, Slewrate = 0	170	–	
			Drive = 0, Slewrate = 1	50	–	
			Drive = 1, Slewrate = 1	75	–	
		DDRIO	Drive = 0	140	–	
			Drive = 1	160	–	

Note:

1. Measured between V_O = 10% V_{DD} and V_O= 90% V_{DD}
2. f_{MAX} = 0.75 / (t_R+t_F). Frequency numbers are rounded for legibility.
3. f_{MAX} may be limited by the peripheral that drives the I/O line.
4. Characterization and/or simulation data.

57.5.2.2 Input AC Characteristics

The following table provides the input requirements of the signals to be connected to I/O lines when they are configured as digital inputs. In particular, these values apply when the **XIN** input is used as a clock input of the device (main crystal oscillator set in Bypass mode). They do not apply for the **XIN32** input which is designed for slow signals with frequencies up to **50 kHz**. Parameters V_{IL} and V_{IH} defined in table Input DC Characteristics apply.

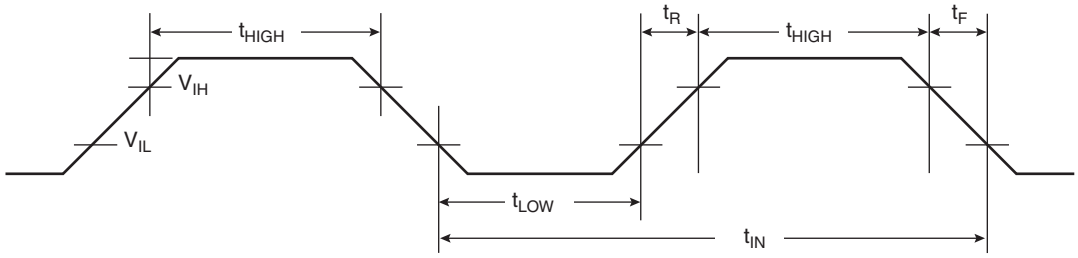
Table 57-18. Input AC Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{IN}	Input frequency(1)	XOUT connected to ground	–	–	50	MHz
t_{IN}	Input period	–	20	–	–	ns
t_{HIGH}	Time at high level	–	8	–	–	ns
t_{LOW}	Time at low level	–	8	–	–	ns
t_R	Rise time	–	–	–	2.2	ns
t_F	Fall time	–	–	–	2.2	ns

Note:

1. The maximum input frequency may be limited by the peripheral receiving this signal.

Figure 57-7. Digital Input AC Characteristics



Downloaded from Arrow.com.

57.6 Digital Peripheral Characteristics

57.6.1 QSPI Characteristics

Figure 57-8. QSPI Master Mode with (CPOL= NCPHA = 0) or (CPOL= NCPHA= 1) and Single Data Rate

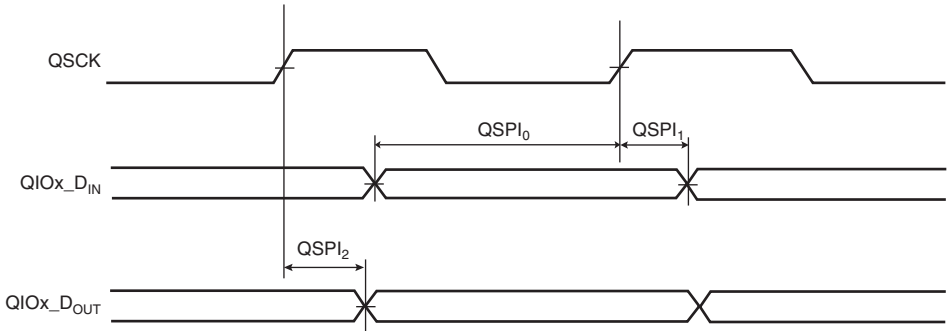


Figure 57-9. QSPI Master Mode with (CPOL = 0 and NCPHA=1) or (CPOL=1 and NCPHA= 0) and Single Data Rate

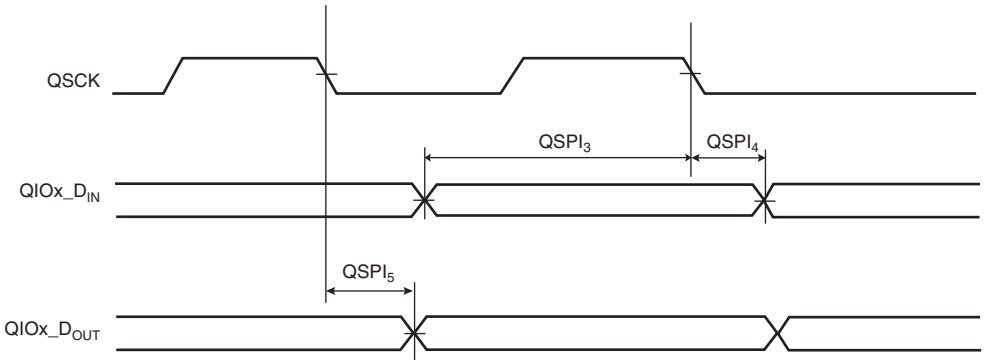
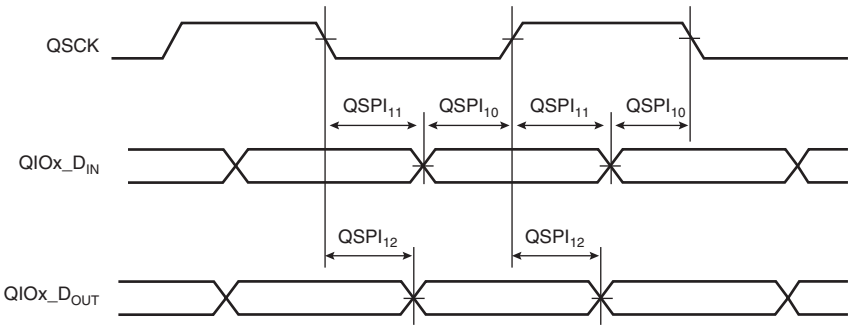


Figure 57-10. QSPI Master Mode in Double Data Rate mode



57.6.1.1 QSPI Supported Operation

The following table summarizes the supported operating modes (Single Data Rate, Double Data Rate, or none) of the QSPI peripheral as a function of the ratios between CPU_CLK, MCK, QSPI 2x MCK and QSCK frequencies. Frequencies are expressed in MHz.

Downloaded from Arrow.com.

Table 57-19. QSPI Operation

CPU_CLK	MDIV	MCK	STH / WCS	QSPI 2x MCK	QSCK	Supported Operation
600	3	200	STH	400	50	SDR and DDR
	4	150	STH	300	75	SDR and DDR
540	3	180	STH	360	60	SDR and DDR
	4	135	STH	270	67.5	SDR and DDR
420	3	140	WCS	280	70	SDR and DDR
	4	105	WCS	210	52.5	SDR and DDR

Note: "–" means the operation is not supported.

57.6.1.2 Maximum QSPI Frequency

The following formulas give the maximum QSPI frequency in Master Read and Write modes.

- Master Write in Single Data Rate Mode
- Master Read in Single Data Rate Mode

$$f_{\mathrm{QSCK}}\mathrm{max} = \frac{1}{QSPI_0(\mathrm{or}\ QSPI_3\ \mathrm{or}\ QSPI_{10}) + t_{\mathrm{VALID}}}$$

t_{VALID} is the slave time response to output data after detecting a QSCK edge.

For a QSPI slave device with t_{VALID} (or t_V) = 12 ns, with $QSPI_3$ = 1.1 ns, f_{QSCK}max = 76 MHz.

For a QSPI Flash memory device with t_{VALID} (or t_V) = 6 ns, and $QSPI_3$ = 1.1 ns, the formula returns a value of 141 MHz. In worst case conditions, this exceeds 70 MHz, which is the maximum allowed frequency of the QSPI master. In this case, the limitation is due to the controller and not to the slave.

57.6.1.3 QSPI Timings

Timings are given in the following domains:

- 1.8V domain: VDDIO from 1.7V to 1.95V, maximum external capacitor = 15 pF, DRV= 1, SR = 0
- 3.3V domain: VDDIO from 3.00V to 3.6V, maximum external capacitor = 15 pF, DRV= 0, SR = 0

Table 57-20. QSPI Timings in Single Data Rate Mode(1)

Symbol	Parameter	Conditions	Min	Max	Unit
$QSPI_0$	QIOx data in to QSCK rising edge (input setup time)	3.3V domain	1.2	–	ns
		1.8V domain	1.1	–	ns
$QSPI_1$	QIOx data in to QSCK rising edge (input hold time)	3.3V domain	0.2	–	ns
		1.8V domain	0.2	–	ns
$QSPI_2$	QSCK rising edge to QIOx data out valid	3.3V domain	0	1.4	ns
		1.8V domain	0	1.9	ns
$QSPI_3$	QIOx data in to QSCK falling edge (input setup time)	3.3V domain	1.1	–	ns
		1.8V domain	1.1	–	ns
$QSPI_4$	QIOx data in to QSCK falling edge (input hold time)	3.3V domain	0.5	–	ns
		1.8V domain	0.4	–	ns
$QSPI_5$	QSCK falling edge to QIOx data out valid	3.3V domain	0	1.3	ns
		1.8V domain	0	1.6	ns

Downloaded from Arrow.com.

Note:

1. The data provided in this table are extracted from circuit simulation results.

In the following table:

- t_{HCLK} is the HCLK period.
- k is 0.25 if $f_{MCK} = f_{QSCK}$ and k=0.5 when $f_{MCK} \geq f_{QSCK}$

Table 57-21. QSPI Timings in Double Data Rate Mode(1)

Symbol	Parameter	Conditions	Min	Max	Unit
$QSPI_{10}$	QIOx data in to QSCK edge (rising or falling, input setup time)	3.3V domain	1.2	–	ns
		1.8V domain	1.4	–	ns
$QSPI_{11}$	QIOx data in to QSCK edge (rising or falling, input hold time)	3.3V domain	0.5	–	ns
		1.8V domain	0.4	–	ns
$QSPI_{12}$	QSCK edge (rising or falling) to QIOx data out valid	3.3V domain	k x T_{HCLK} -0.6	k x T_{HCLK} + 1.5	ns
		1.8V domain	k x T_{HCLK} - 0.8	k x T_{HCLK} + 2.0	ns

Note:

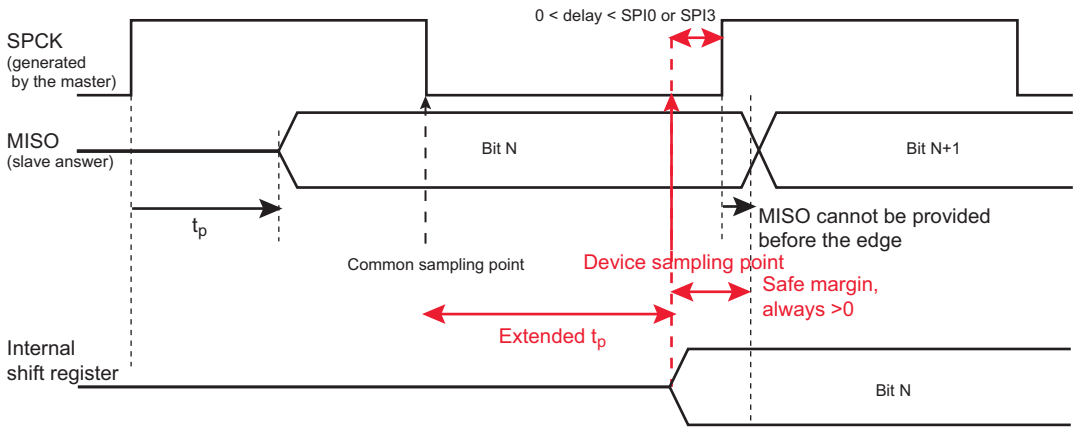
1. The data provided in this table are extracted from circuit simulation results.

57.6.2 FLEXCOM Characteristics

In Figure 57-12 and Figure 57-13, the MOSI line shifting edge is represented with a hold time equal to 0. However, it is important to note that for this device, the MISO line is sampled prior to the MOSI line shifting edge. As shown in Figure 57-11, the device sampling point extends the propagation delay (t_p) for slave and routing delays to more than half the SPI clock period, whereas the common sampling point allows only less than half the SPI clock period.

As an example, an SPI Slave working in Mode 0 can be safely driven if the SPI Master is configured in Mode 0.

Figure 57-11. FLEXCOM in SPI Mode: MISO Capture in Master Mode



Downloaded from Arrow.com.

Figure 57-12. FLEXCOM in SPI Master Mode 1 and 2

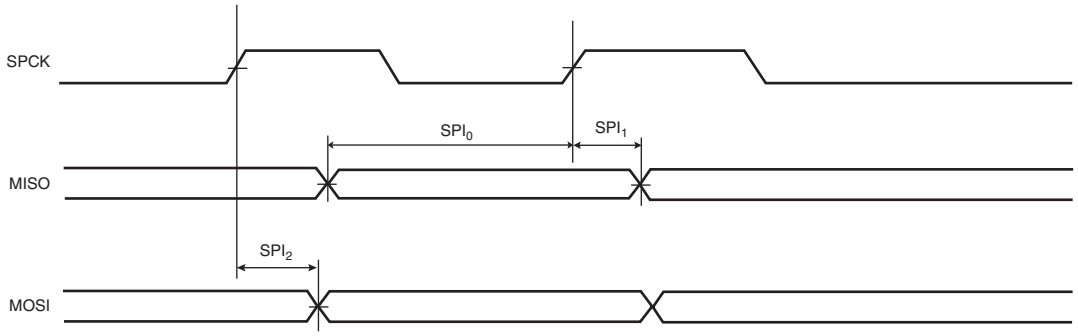


Figure 57-13. FLEXCOM in SPI Master Mode 0 and 3

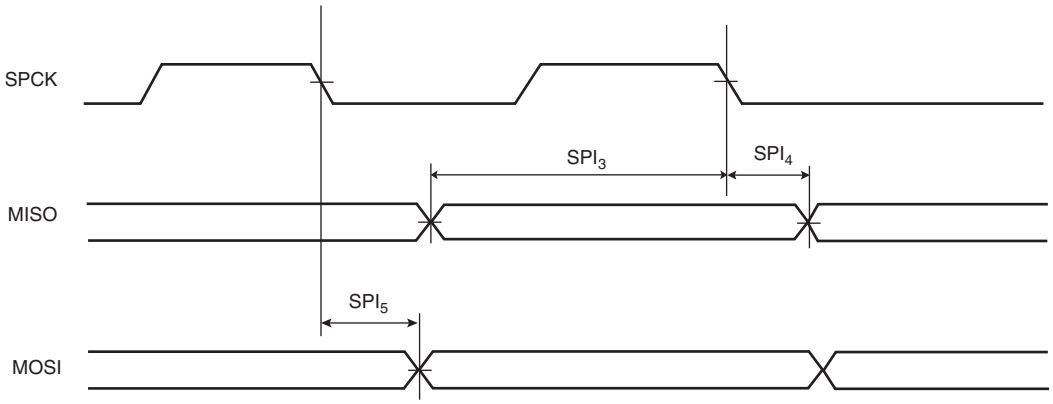
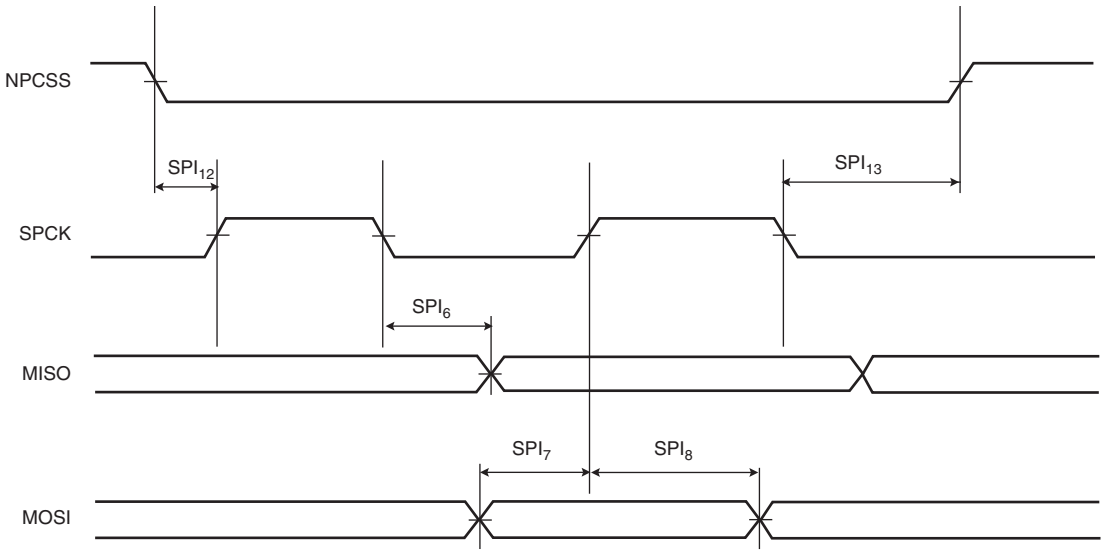


Figure 57-14. FLEXCOM in SPI Slave Mode 0 and 3



Downloaded from Arrow.com.

Figure 57-15. FLEXCOM in SPI Slave Mode 1 and 2

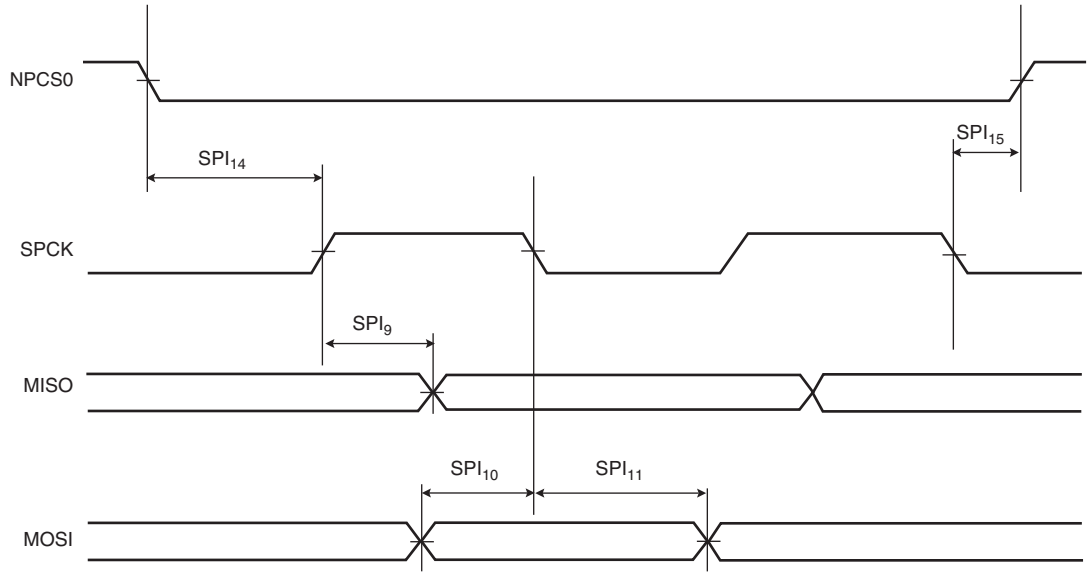
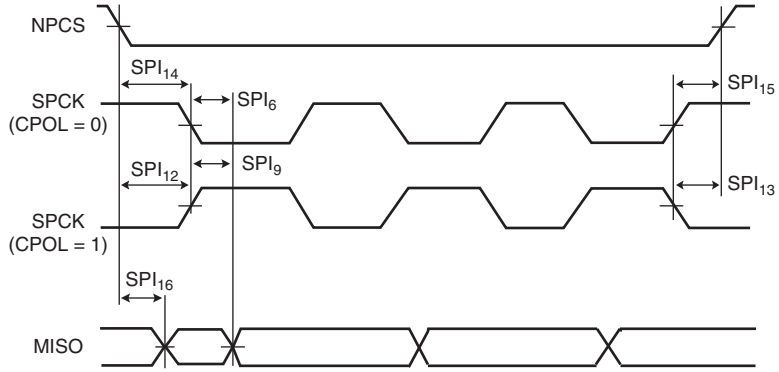


Figure 57-16. FLEXCOM in SPI Slave - NPCS Timings



57.6.2.1 Maximum FLEXCOM SPI Frequency

The following formulas give the maximum SPI frequency in Master Read and Write modes and in Slave Read and Write modes.

57.6.2.1.1 Master Write Mode

The SPI sends data to a slave device only, e.g. an LCD. The limit is given by SPI_2 (or SPI_5) timing. Since it gives a maximum frequency above the output driver maximum frequency (see 57.5.2 I/O AC Characteristics), the max SPI frequency is the one from the output driver.

57.6.2.1.2 Master Read Mode

$$f_{SPCK}\max = \frac{1}{SPI_0(\text{or } SPI_3) + t_{valid}}$$

t_{valid} is the slave time response to output data after detecting an SPCK edge.

For a nonvolatile memory with t_{VALID} (or t_v) = 5 ns, using SPI_3 at 3.3V, f_{SPCK}max = 56 MHz.

57.6.2.1.3 Slave Read Mode

In Slave mode, SPCK is the input clock for the SPI. The max SPCK frequency is given by setup and hold timings SPI_7/SPI_8(or SPI_{10}/SPI_{11}). Since this gives a frequency well above the output driver maximum frequency, the limit in Slave Read mode is given by the SPCK output driver.

57.6.2.1.4 Slave Write Mode

$$f_{SPCK}\max = \frac{1}{2x(SPI_{6max}(\text{or } SPI_{9max}) + t_{setup})}$$

Downloaded from Arrow.com.

t_{setup} is the setup time from the master before sampling data.

57.6.2.2 FLEXCOM SPI Timings

The timings in the table below are given in the following domains:

- 1.8V domain: VDDIO from 1.7V to 1.95V, maximum external capacitor = 15 pF, DRV = 1, SR = 1
- 3.3V domain: VDDIO from 3.00V to 3.6V, maximum external capacitor = 15 pF, DRV = 0, SR = 1

Table 57-22. FLEXCOM SPI Timings(1)

Symbol	Parameter	Conditions	Min	Max	Unit
Master Mode					
SPI_0	MISO setup time before SPCK rises	3.3V domain	16	–	ns
		1.8V domain	18	–	ns
SPI_1	MISO hold time after SPCK rises	3.3V domain	0	–	ns
		1.8V domain	0	–	ns
SPI_2	SPCK rising to MOSI delay	3.3V domain	0	7.3	ns
		1.8V domain	0	7.3	ns
SPI_3	MISO setup time before SPCK falls	3.3V domain	17.9	–	ns
		1.8V domain	18.2	–	ns
SPI_4	MISO hold time after SPCK falls	3.3V domain	0	–	ns
		1.8V domain	0	–	ns
SPI_5	SPCK falling to MOSI delay	3.3V domain	0	9.2	ns
		1.8V domain	0	7.5	ns
Slave Mode					
SPI_6	SPCK falling to MISO delay	3.3V domain	2.9	13.6	ns
		1.8V domain	3.5	14.3	ns
SPI_7	MOSI setup time before SPCK rises	3.3V domain	1.5	–	ns
		1.8V domain	1.6	–	ns
SPI_8	MOSI hold time after SPCK rises	3.3V domain	0.6	–	ns
		1.8V domain	0.8	–	ns
SPI_9	SPCK rising to MISO delay	3.3V domain	3.1	14.1	ns
		1.8V domain	3.6	14.7	ns
SPI_{10}	MOSI setup time before SPCK falls (slave)	3.3V domain	1.5	–	ns
		1.8V domain	1.6	–	ns
SPI_{11}	MOSI hold time after SPCK falls (slave)	3.3V domain	0.6	–	ns
		1.8V domain	0.8	–	ns
SPI_{12}	NPCS setup to SPCK rising (slave)	3.3V domain	5.7	–	ns
		1.8V domain	6.1	–	ns
SPI_{13}	NPCS hold after SPCK falling (slave)	3.3V domain	0	–	ns
		1.8V domain	0	–	ns

Downloaded from Arrow.com.

.....continued					
Symbol	**Parameter**	**Conditions**	**Min**	**Max**	**Unit**
SPI_{14}	NPCS setup to SPCK falling (slave)	3.3V domain	6.1	–	ns
		1.8V domain	6.4	–	ns
SPI_{15}	NPCS hold after SPCK falling (slave)	3.3V domain	0	–	ns
		1.8V domain	0	–	ns
SPI_{16}	NPCS falling to MISO valid (slave)	3.3V domain	20.5	–	ns
		1.8V domain	21.4	–	ns

Note:

1. The data provided in this table are extracted from circuit simulation results.

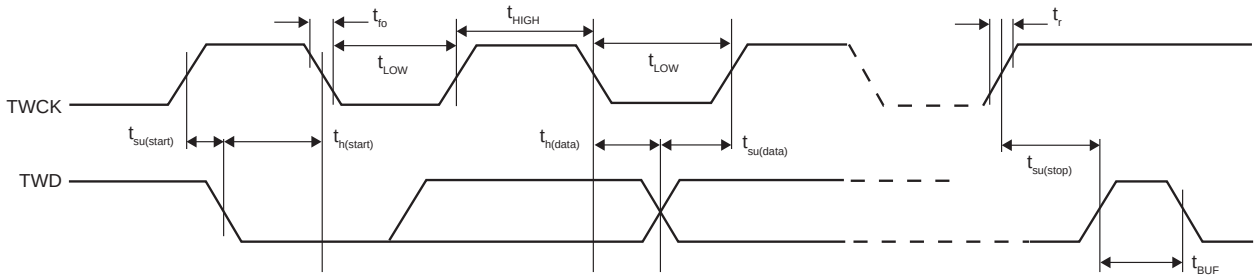
57.6.2.3 USART in Asynchronous Modes

In Asynchronous modes, the maximum baud rate that can be achieved is MCK / 8, if the bit USART_MR.OVER=1.

Example: if MCK = 200 MHz, the baud rate is 25 Mbit/s.

57.6.2.4 TWI Timings

Figure 57-17. Two-wire Serial Bus Timing



The table below describes the requirements for devices connected to the Two-wire Serial Bus.

Table 57-23. Two-wire Serial Bus Requirements

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IL}	Input Low-voltage	–	-0.3	0.3 × V_{DDIO}	V
V_{IH}	Input High-voltage	–	0.7 × V_{DDIO}	V_{CC} + 0.3	V
V_{hys}	Hysteresis of Schmitt Trigger Inputs	–	0.150	–	V
V_{OL}	Output Low-voltage	3 mA sink current	–	0.4	V
t_r	Rise Time for both TWD and TWCK	–	20 + 0.1C_b(2)	300	ns
t_{fo}	Output Fall Time from V_{IHmin} to V_{ILmax}	10 pF < C_b < 400 pF (see the figure above)	20 + 0.1C_b(2)	250	ns
C_i(1)	Capacitance for each I/O Pin	–	–	10	pF
f_{TWCK}	TWCK Clock Frequency	–	0	400	kHz
Rp	Value of Pull-up Resistor	f_{TWCK}≤ 100 kHz	(V_{DDIO} - 0.4V) ÷ 3mA	1000ns ÷ C_b	Ω
		f_{TWCK} > 100 kHz	(V_{DDIO} - 0.4V) ÷ 3mA	300ns ÷ C_b	Ω
t_{LOW}	Low Period of the TWCK Clock	f_{TWCK}≤ 100 kHz	(3)	–	µs
		f_{TWCK} > 100 kHz	(3)	–	µs

Downloaded from Arrow.com.

.....continued

Symbol	Parameter	Conditions	Min	Max	Unit
t_{HIGH}	High Period of the TWCK Clock	f_{TWCK}≤ 100 kHz	(4)	–	µs
		f_{TWCK} > 100 kHz	(4)	–	µs
$t_{h(start)}$	Hold Time (repeated) START condition	f_{TWCK}≤ 100 kHz	t_{HIGH}	–	µs
		f_{TWCK} > 100 kHz	t_{HIGH}	–	µs
$t_{su(start)}$	Setup Time for a Repeated START condition	f_{TWCK}≤ 100 kHz	t_{HIGH}	–	µs
		f_{TWCK} > 100 kHz	t_{HIGH}	–	µs
$t_{h(data)}$	Data Hold Time	f_{TWCK}≤ 100 kHz	0	(HOLD + 3) × $t_{peripheral\ clock}$	µs
		f_{TWCK} > 100 kHz	0	(HOLD + 3) × $t_{peripheral\ clock}$	µs
$t_{su(data)}$	Data Setup Time	f_{TWCK}≤ 100 kHz	t_{LOW} - (HOLD + 3) × $t_{peripheral\ clock}$	–	ns
		f_{TWCK} > 100 kHz	t_{LOW} - (HOLD + 3) × $t_{peripheral\ clock}$	–	ns
$t_{su(stop)}$	Setup time for STOP condition	f_{TWCK}≤ 100 kHz	t_{HIGH}	–	µs
		f_{TWCK} > 100 kHz	t_{HIGH}	–	µs
t_{BUF}	Bus free time between a STOP and START condition	f_{TWCK}≤ 100 kHz	t_{LOW}	–	µs
		f_{TWCK} > 100 kHz	t_{LOW}	–	µs

Note:

1. Required only for f_{TWCK} > 100 kHz
2. C_B = capacitance of one bus line in pF. Per I2C Standard, C_b Max = 400 pF
3. The TWCK low period is defined as follows: $t_{LOW} = ((CLDIV \times 2^{CKDIV}) + 3) \times t_{MCK}$
4. The TWCK high period is defined as follows: $t_{HIGH} = ((CHDIV \times 2^{CKDIV}) + 3) \times t_{MCK}$

57.6.3 SDMMC Characteristics

The Secure Digital Multimedia Card Controller (SDMMC) complies with the following specifications:

- SD Host Controller Standard Specification Version 3.00
- Embedded MultiMedia Card Specification (e.MMC) Version 4.51
  - e.MMC Default Speed (Maximum SDCLK Frequency = 26 MHz)
  - e.MMC High Speed (Maximum SDCLK Frequency = 52 MHz)
  - e.MMC High Speed DDR (Maximum SDCLK Frequency = 52 MHz)
- SD Memory Card Specification Version 3.00
- SDIO Specification Version 3.00
  - SD/SDIO Default Speed (Maximum SDCLK Frequency = 25 MHz)
  - SD/SDIO High Speed (Maximum SDCLK Frequency = 50 MHz)

57.6.4 MPDDRC and SDRAMC Characteristics

The MPDDRC and SDRAMC are respectively compliant with the following JEDEC standards:

- DDR2-SDRAM: JESD79-2F with operating frequencies up to 200 MHz
- LPDDR-SDRAM: JESD209B with operating frequencies up to 200 MHz
- SDR-SDRAM: PC100 and PC133 with frequencies up to 200 MHz
- LPSDR-SDRAM : PC100 and PC133 with low-power extensions and frequencies up to 166 MHz

The physical interface (PCB layout) between the processor and its memory has a major impact on signal integrity. Microchip provides IBIS models of the SAM9X60 device and strongly recommends to verify this processor-memory

Downloaded from Arrow.com.

interface on a PCB simulation tool. For design guidance, refer to the *SAM9X60 hardware design guideline application note*.

The following table gives the recommended settings on the MPDDRC_RD_DATA_PATH.SHIFT_SAMPLING or SDRAMC_MDR.SHIFT_SAMPLING field depending on the memory type and on its operating frequency.

Table 57-24. SHIFT_SAMPLING Settings

SDRAM Type	SDRAM Clock Frequency	SHIFT_SAMPLING
DDR2-SDRAM	125 MHz ≤ f_{SDRAM_CLK} ≤ 200 MHz	2
LPDDR-SDRAM	f_{SDRAM_CLK} ≤ 200 MHz	2
SDR-SDRAM	f_{SDRAM_CLK} ≤ 200 MHz	3
LPSDR-SDRAM	f_{SDRAM_CLK} ≤ 166 MHz	3

57.6.5 SMC Timings

Timings are given in the following domains:

- 1.8V domain: VDDIO from 1.7V to 1.95V, maximum external capacitor = 15 pF, DRV= 1, SR = 0
- 3.3V domain: VDDIO from 3.00V to 3.6V, maximum external capacitor = 15 pF, DRV= 0, SR = 0

Timings are given assuming a capacitance load on data, control and address output lines.

In the following tables, t_{CPMCK} is MCK period.

57.6.5.1 Read Timings

Table 57-25. SMC Read Signals - NRD-controlled (READ_MODE = 1)(1)

Symbol	Parameter	Conditions	Min	Max	Unit
NO HOLD SETTINGS (NRD_HOLD = 0)					
SMC_1	Data setup before NRD high	3.3V domain	16	–	ns
		1.8V domain	18	–	ns
SMC_2	Data hold after NRD high	3.3V domain	0	–	ns
		1.8V domain	0	–	ns
HOLD SETTINGS (NRD_HOLD ≠ 0)					
SMC_3	Data setup before NRD high	3.3V domain	12.8	–	ns
		1.8V domain	15.1	–	ns
SMC_4	Data hold after NRD high	3.3V domain	0	–	ns
		1.8V domain	0	–	ns
HOLD or NO HOLD SETTINGS (NRD_HOLD ≠ 0, NRD_HOLD = 0)					
SMC_5	A0–A22 valid before NRD high	3.3V domain	(NRD_SETUP + NRD_PULSE) × t_{CPMCK} - 10	–	ns
		1.8V domain	(NRD_SETUP + NRD_PULSE) × t_{CPMCK} - 8.7	–	ns
SMC_6	NCS low before NRD high	3.3V domain	(NRD_SETUP + NRD_PULSE - NCS_RD_SETUP) × t_{CPMCK} - 9.9	–	ns
		1.8V domain	(NRD_SETUP + NRD_PULSE - NCS_RD_SETUP) × t_{CPMCK} - 8.5	–	ns

Downloaded from Arrow.com.

.....continued

Symbol	Parameter	Conditions	Min	Max	Unit
SMC_7	NRD pulse width	3.3V domain	NRD_PULSE × t_{CPMCK} - 0.1	–	ns
		1.8V domain	NRD_PULSE × t_{CPMCK} - 0.2	–	ns

Note:

1. The data provided in this table are extracted from circuit simulation results.

Table 57-26. SMC Read Signals - NCS-controlled (READ_MODE= 0)(1)

Symbol	Parameter	Conditions	Min	Max	Unit
NO HOLD SETTINGS (NCS_RD_HOLD = 0)					
SMC_8	Data setup before NCS high	3.3V domain	24.3	–	ns
		1.8V domain	24.9	–	ns
SMC_9	Data hold after NCS high	3.3V domain	0	–	ns
		1.8V domain	0	–	ns
HOLD SETTINGS (NCS_RD_HOLD ≠ 0)					
SMC_{10}	Data setup before NCS high	3.3V domain	21.9	–	ns
		1.8V domain	22.6	–	ns
SMC_{11}	Data hold after NCS high	3.3V domain	0	–	ns
		1.8V domain	0	–	ns
HOLD or NO HOLD SETTINGS (NCS_RD_HOLD ≠ 0, NCS_RD_HOLD = 0)					
SMC_{12}	A0–A22 Valid before NCS High	3.3V domain	(NCS_RD_SETUP + NCS_RD_PULSE) × t_{CPMCK} - 8.6	–	ns
		1.8V domain	(NCS_RD_SETUP + NCS_RD_PULSE) × t_{CPMCK} - 7.2	–	ns
SMC_{13}	NRD low before NCS High	3.3V domain	(NCS_RD_SETUP + NCS_RD_PULSE - NRD_SETUP) × t_{CPMCK} - 0.4	–	ns
		1.8V domain	(NCS_RD_SETUP + NCS_RD_PULSE - NRD_SETUP) × t_{CPMCK} - 0.5	–	ns
SMC_{14}	NCS pulse width	3.3V domain	NCS_RD_PULSE × t_{CPMCK} - 2.8	–	ns
		1.8V domain	NCS_RD_PULSE × t_{CPMCK} - 1.5	–	ns

Note:

1. The data provided in this table are extracted from circuit simulation results.

57.6.5.2 Write Timings

Table 57-27. SMC Write Signals - NWE-controlled (WRITE_MODE = 1)

Symbol	Parameter	Conditions	Min	Max	Unit
HOLD or NO HOLD SETTINGS (NWE_HOLD ≠ 0, NWE_HOLD = 0)					
SMC_{15}	Data out valid before NWE high	3.3V domain	NWE_PULSE × t_{CPMCK} - 10.4	–	ns
		1.8V domain	NWE_PULSE × t_{CPMCK} - 9.4	–	ns

Downloaded from Arrow.com.

.....continued

Symbol	Parameter	Conditions	Min	Max	Unit
SMC_{16}	NWE pulse width	3.3V domain	NWE_PULSE × t_{CPMCK} - 2	–	ns
		1.8V domain	NWE_PULSE × t_{CPMCK} - 0.3	–	ns
SMC_{17}	A0–A22 valid before NWE low	3.3V domain	NWE_SETUP × t_{CPMCK} - 10.5	–	ns
		1.8V domain	NWE_SETUP × t_{CPMCK} - 9.2	–	ns
SMC_{18}	NCS low before NWE high	3.3V domain	(NWE_SETUP - NCS_RD_SETUP + NWE_PULSE) × t_{CPMCK} - 10.3	–	ns
		1.8V domain	(NWE_SETUP - NCS_RD_SETUP + NWE_PULSE) × t_{CPMCK} - 8.8	–	ns
HOLD SETTINGS (NWE_HOLD ≠ 0)					
SMC_{19}	NWE high to data OUT, NBS0/A0 NBS1, A1 - A23 change	3.3V domain	NWE_HOLD × t_{CPMCK} - 4.5	–	ns
		1.8V domain	NWE_HOLD × t_{CPMCK} - 5.0	–	ns
SMC_{20}	NWE high to NCS inactive(1)	3.3V domain	(NWE_HOLD - NCS_WR_HOLD) × t_{CPMCK} - 3.3	–	ns
		1.8V domain	(NWE_HOLD - NCS_WR_HOLD) × t_{CPMCK} - 3.6	–	ns
NO HOLD SETTINGS (NWE_HOLD = 0)					
SMC_{21}	NWE high to data OUT, NBS0/A0 NBS1, A1 - A23, NCS change(1)	3.3V domain	- 4.7	–	ns
		1.8V domain	- 5.0	–	ns

Note:

1. Hold length = Total cycle duration - setup duration - pulse duration. "hold length" is for "NCS_WR_HOLD length" or "NWE_HOLD length".

Table 57-28. SMC Write Signals - NCS-Controlled (WRITE_MODE = 0)

Symbol	Parameter	Conditions	Min	Max	Unit
SMC_{22}	Data out valid before NCS high	3.3V domain	NCS_WR_PULSE × t_{CPMCK} - 9.3	–	ns
		1.8V domain	NCS_WR_PULSE × t_{CPMCK} - 7.8	–	ns
SMC_{23}	NCS pulse width	3.3V domain	NCS_WR_PULSE × t_{CPMCK} - 2.8	–	ns
		1.8V domain	NCS_WR_PULSE × t_{CPMCK} - 1.5	–	ns
SMC_{24}	A0–A22 valid before NCS low	3.3V domain	NCS_WR_SETUP × t_{CPMCK} - 9.3	–	ns
		1.8V domain	NCS_WR_SETUP × t_{CPMCK} - 7.9	–	ns
SMC_{25}	NWE low before NCS high	3.3V domain	(NCS_WR_SETUP - NWE_SETUP + NCS_PULSE) × t_{CPMCK} - 6.4	–	ns
		1.8V domain	(NCS_WR_SETUP - NWE_SETUP + NCS_PULSE) × t_{CPMCK} - 4.8	–	ns
SMC_{26}	NCS high to data OUT, A0 - A25 change	3.3V domain	NCS_WR_HOLD × t_{CPMCK} - 8.6	–	ns
		1.8V domain	NCS_WR_HOLD × t_{CPMCK} - 9.1	–	ns

Downloaded from Arrow.com.

.....continued					
Symbol	**Parameter**	**Conditions**	**Min**	**Max**	**Unit**
SMC_{27}	NCS high to NWE Inactive	3.3V domain	(NCS_WR_HOLD - NWE_HOLD) × t_{CPMCK} - 8.6	–	ns
		1.8V domain	(NCS_WR_HOLD - NWE_HOLD) × t_{CPMCK} - 8.5	–	ns

Figure 57-18. SMC Timings - NCS-controlled Read and Write

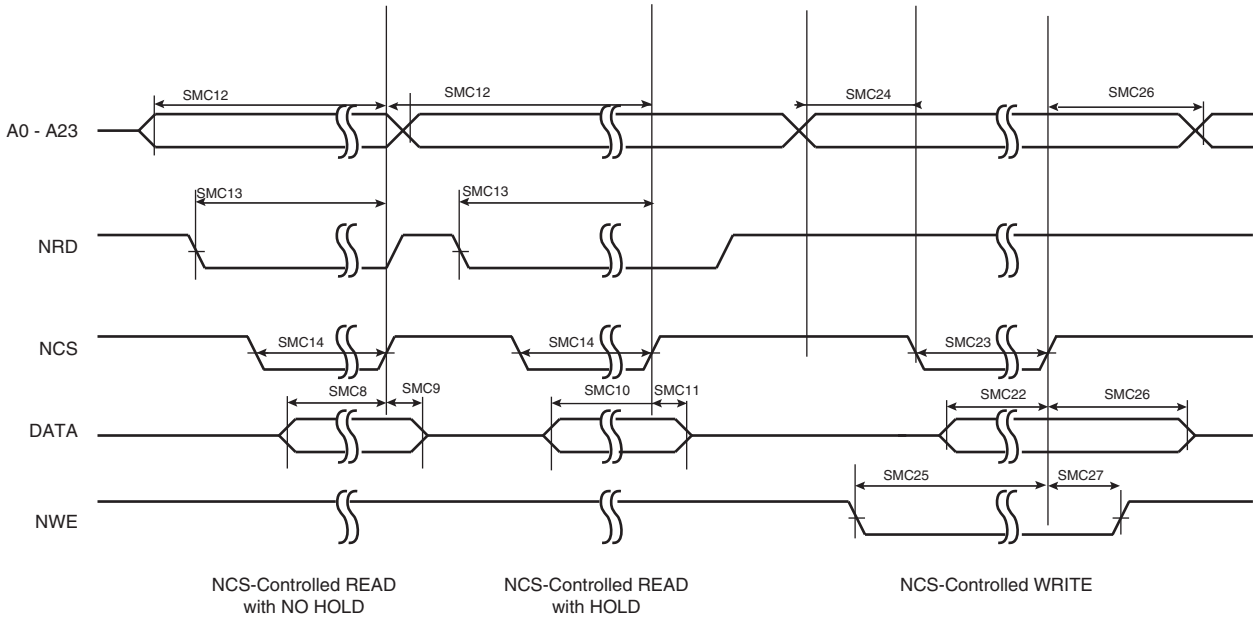
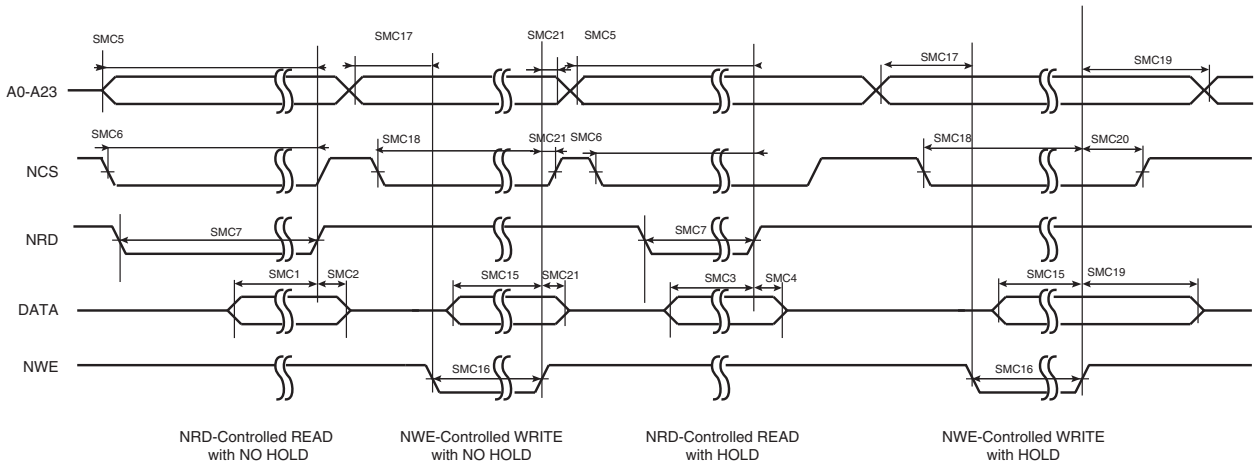


Figure 57-19. SMC Timings - NRD-controlled Read and NEW-controlled Write



57.6.6 SSC Timings

Timings are given in the following domains:

- 1.8V domain: VDDIO from 1.7V to 1.95V, maximum external capacitor = 15 pF, DRV = 1, SR = 1
- 3.3V domain: VDDIO from 3.00V to 3.6V, maximum external capacitor = 15 pF, DRV = 0, SR = 1

Downloaded from Arrow.com.

Figure 57-20. SSC Transmitter, TK and TF in Output

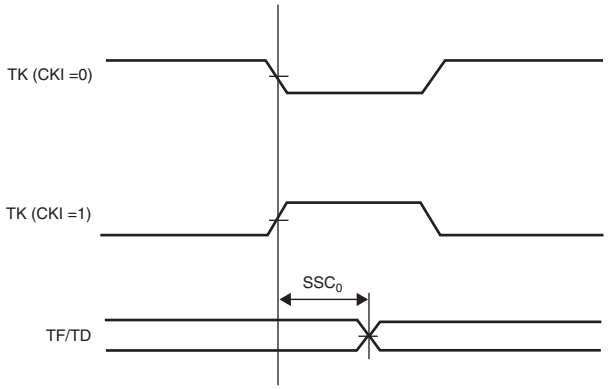


Figure 57-21. SSC Transmitter, TK in Input and TF in Output

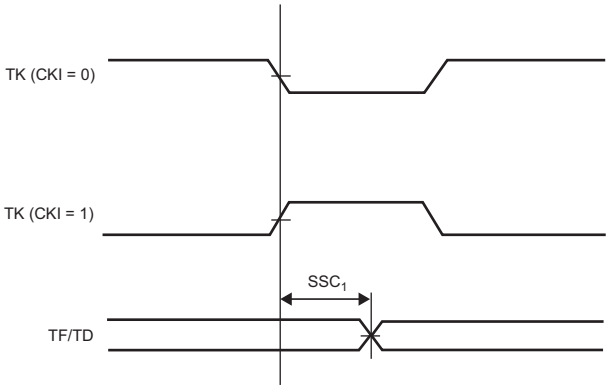
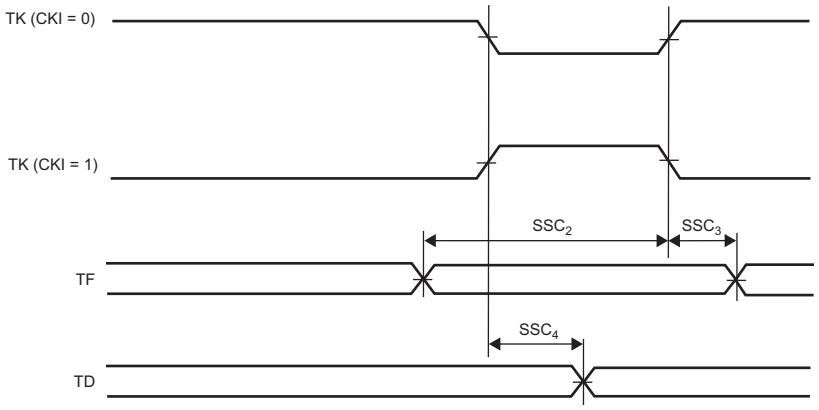


Figure 57-22. SSC Transmitter, TK in Output and TF in Input



Downloaded from Arrow.com.

Figure 57-23. SSC Transmitter, TK and TF in Input

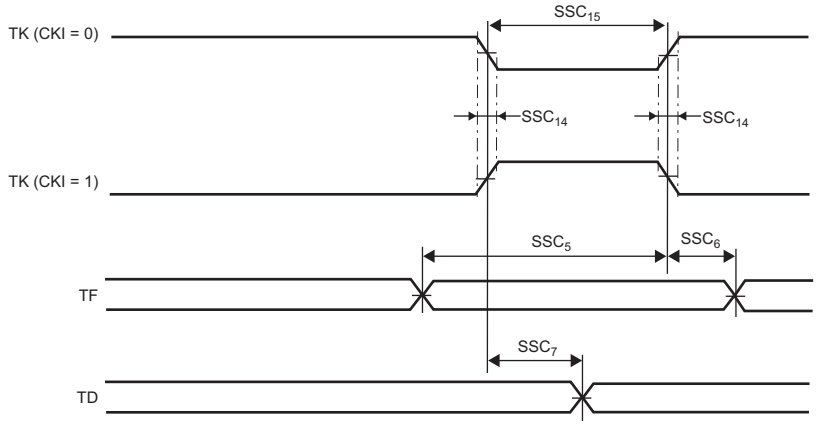


Figure 57-24. SSC Receiver RK and RF in Input

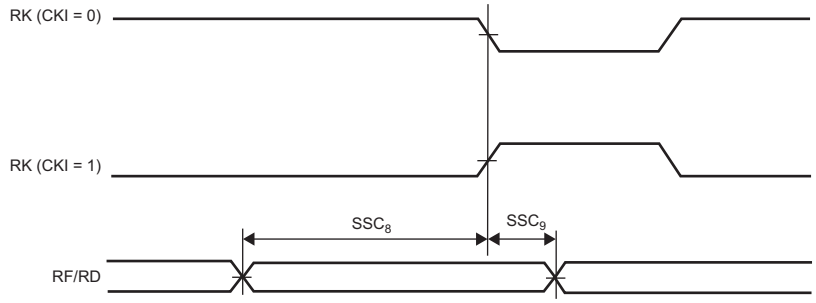
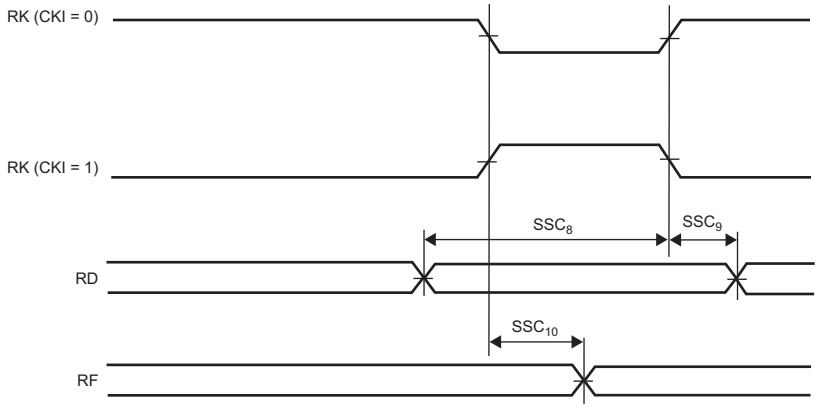


Figure 57-25. SSC Receiver, RK in Input and RF in Output



Downloaded from Arrow.com.

Figure 57-26. SSC Receiver, RK and RF in Output

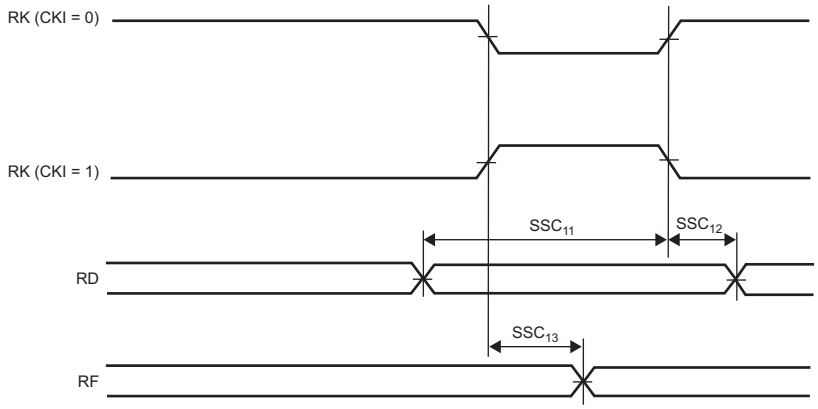


Figure 57-27. SSC Receiver, RK in Output and RF in Input

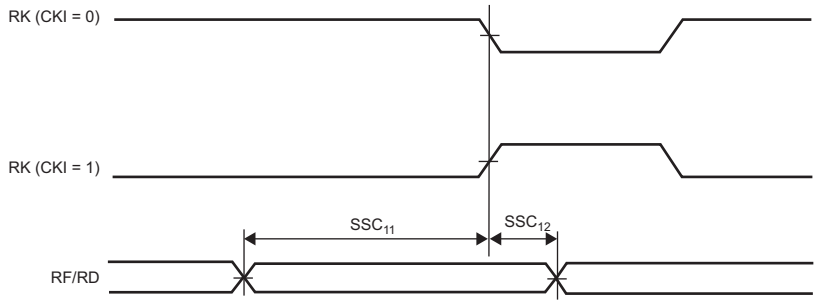


Table 57-29. SSC Timings[2]

Symbol	Power supply / Parameter	Conditions	1.8V		3.3V		Unit
			Min	Max	Min	Max	
Transmitter							
SSC_0	TK edge to TF/TD (TK output, TF output)	–	0	3.8	0	5.5	ns
SSC_1	TK edge to TF/TD (TK input, TF output)	–	3.1	14.4	2.5	13.8	ns
SSC_2	TF setup time before TK edge (TK output)	–	12.8	–	12.3	–	ns
SSC_3	TF hold time after TK edge (TK output)	–	0	–	0	–	ns
SSC_4	TK edge to TF/TD (TK output, TF input)	–	0	3.8	0	5.5	ns
		STTDLY = 0 START = 4, 5 or 7	2 × t_{CPMCK}	3.8 + (2 × t_{CPMCK})	2 × t_{CPMCK}	5.5 + (2 × t_{CPMCK})	
SSC_5	TF setup time before TK edge (TK input)	–	0	–	0	–	ns
SSC_6	TF hold time after TK edge (TK input)	–	t_{CPMCK}	–	t_{CPMCK}	–	ns
SSC_7	TK edge to TF/TD (TK input, TF input)	–	3.1	14.3	2.5	13.8	ns
		STTDLY = 0 START = 4, 5 or 7	3.1 + (3 × t_{CPMCK})	14.3 + (3 × t_{CPMCK})	2.5 + (3 × t_{CPMCK})	13.8 + (3 × t_{CPMCK})	
SSC_{14}	TK low or high time[1]	$V_{TK} > V_{IH}$ or $V_{TK} < V_{IL}$	3 x t_{CPMCK} + 1.5	–	3 x t_{CPMCK} + 1.7	–	ns
SSC_{15}	TK rise time or fall time[1]	10% to 90%	–	10	–	10	ns
Receiver							
SSC_8	RF/RD setup time before RK edge (RK input)	–	0	–	0	–	ns

Downloaded from Arrow.com.

.....continued

Symbol	Power supply / Parameter	Conditions	1.8V Min	1.8V Max	3.3V Min	3.3V Max	Unit
SSC_9	RF/RD hold time after RK edge (RK input)	–	t_{CPMCK}	–	t_{CPMCK}	–	ns
SSC_{10}	RK edge to RF (RK input)	–	3.5	15.8	3.0	15.2	ns
SSC_{11}	RF/RD setup time before RK edge (RK output)	–	13.3 - t_{CPMCK}	–	12.8 - t_{CPMCK}	–	ns
SSC_{12}	RF/RD hold time after RK edge (RK output)	–	t_{CPMCK}	–	t_{CPMCK}	–	ns
SSC_{13}	RK edge to RF (RK output)	–	0	4.9	0	6.5	ns

Note:

1. SSC_{14} and SSC_{15} apply to RK when RK is selected instead of TK (SSC_TCMR.CKS = RK).
2. The data provided in this table are extracted from circuit simulation results.

57.6.7 I2SMCC Timings

Timings are given in the following domains:

- 1.8V domain: VDDIO from 1.7V to 1.95V, maximum external capacitor = 15 pF, DRV = 1, SR = 1
- 3.3V domain: VDDIO from 3.00V to 3.6V, maximum external capacitor = 15 pF, DRV = 0, SR = 1

Figure 57-28. I2SMCC Timing Diagram in Master Mode

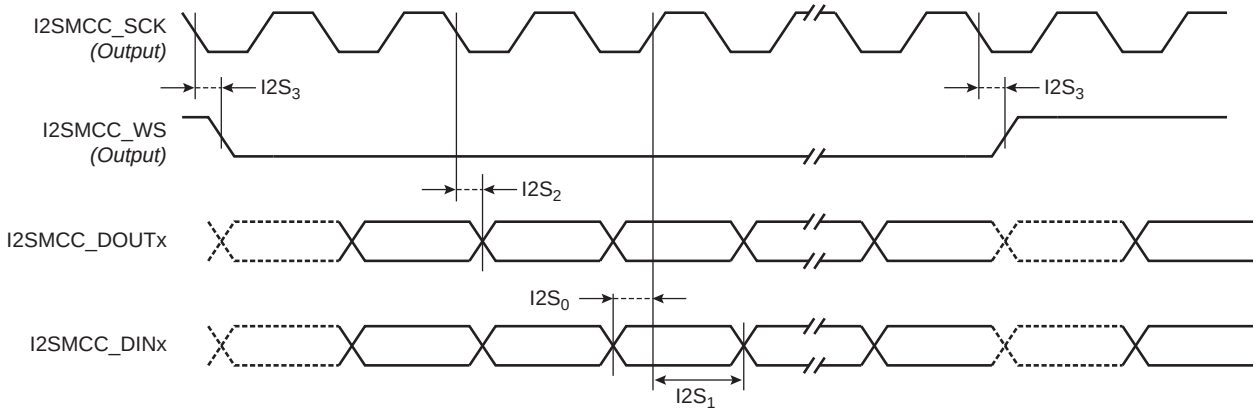
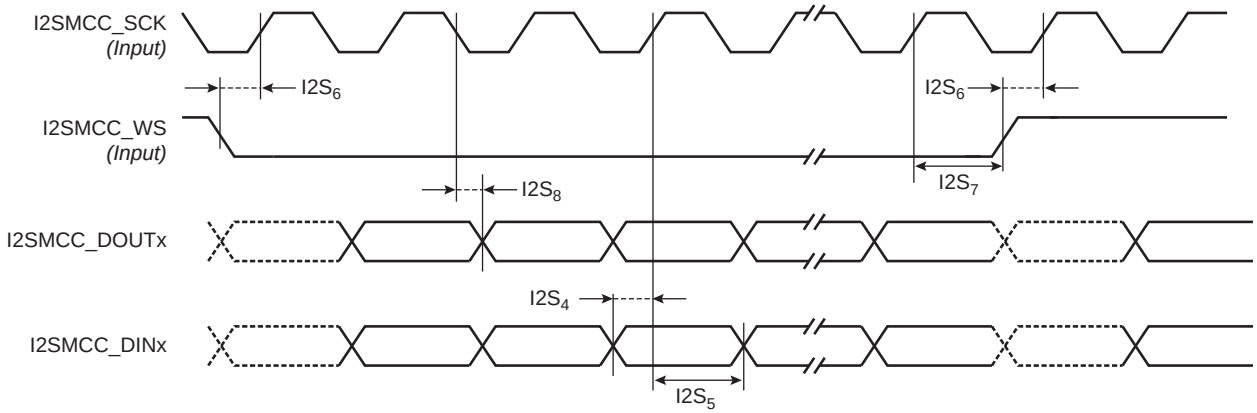


Figure 57-29. I2SMCC Timing Diagram in Slave Mode



Downloaded from Arrow.com.

Table 57-30. I2SMCC Timings[1]

Symbol	Parameter	Conditions	Min	Max	Unit
	Master Mode				
$I2S_0$	I2SMCC_DINx setup time before I2SMCC_SCK rises	3.3V domain	10.5	–	ns
		1.8V domain	12.7	–	
$I2S_1$	I2SMCC_DINx hold time after I2SMCC_SCK rises	3.3V domain	0	–	ns
		1.8V domain	0	–	
$I2S_2$	I2SMCC_SCK falling to I2SMCC_DOUTx delay	3.3V domain	0	3.4	ns
		1.8V domain	0	1.7	
$I2S_3$	I2SMCC_SCK falling to I2SMCC_WS delay	3.3V domain	0	3.5	ns
		1.8V domain	0	1.9	
	Slave Mode				
$I2S_4$	I2SMCC_DINx setup time before I2SMCC_SCK rises	3.3V domain	0.1	–	ns
		1.8V domain	0.2	–	
$I2S_5$	I2SMCC_DINx hold time after I2SMCC_SCK rises	3.3V domain	2.1	–	ns
		1.8V domain	2.3	–	
$I2S_6$	I2SMCC_WS setup time before I2SMCC_SCK rises	3.3V domain	2.2	–	ns
		1.8V domain	2.5	–	
$I2S_7$	I2SMCC_WS hold time after I2SMCC_SCK rises	3.3V domain	1.3	–	ns
		1.8V domain	1.4	–	
$I2S_8$	I2SMCC_SCK falling to I2SMCC_DOUTx delay	3.3V domain	1.9	12.0	ns
		1.8V domain	2.5	12.7	

Note:

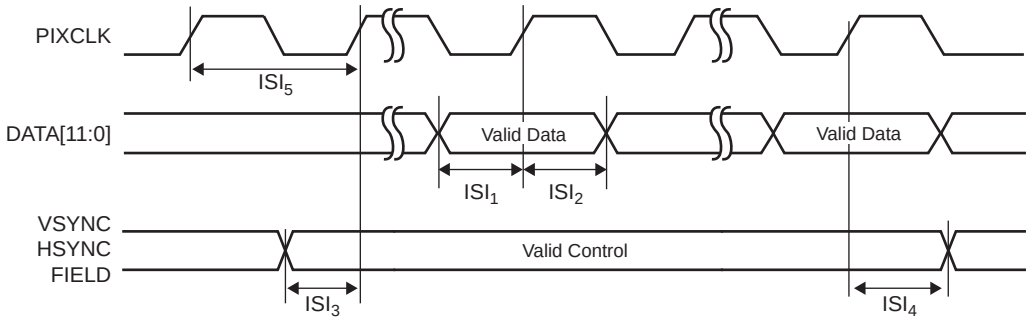
1. The data provided in this table are extracted from circuit simulation results.

57.6.8 ISI Timings

Timings are given in the following domains:

- 1.8V domain: VDDIO from 1.7V to 1.95V, maximum external capacitor = 15 pF, DRV = 1, SR = 1
- 3.3V domain: VDDIO from 3.00V to 3.6V, maximum external capacitor = 15 pF, DRV = 0, SR = 1

Figure 57-30. ISI Timing Diagram



Downloaded from Arrow.com.

Table 57-31. ISI Timings[1]

Symbol	Parameter	Conditions	Min	Max	Unit
ISI_1	DATA setup time before PIXCLK rises	3.3V domain	2.5	–	ns
		1.8V domain	2.8	–	
ISI_2	DATA hold time after PIXCLK rises	3.3V domain	0	–	ns
		1.8V domain	0	–	
ISI_3	VSYNC / HSYNC / FIELD setup time before PIXCLK rises	3.3V domain	2.5	–	ns
		1.8V domain	2.8	–	
ISI_4	VSYNC / HSYNC / FIELD hold time after PIXCLK rises	3.3V domain	0	–	ns
		1.8V domain	0	–	
ISI_5	PIXCLK frequency	3.3V domain	–	75	MHz
		1.8V domain	–	75	

Note:

1. The data provided in this table are extracted from circuit simulation results.

57.6.9 EMAC Timings

Timings are given in the following conditions:

- VDDIO from 3.00V to 3.6V, maximum external capacitor = 15 pF, DRV = 0, SR = 0

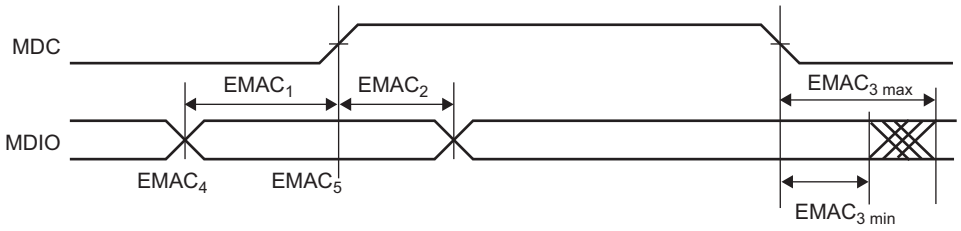
Table 57-32. Ethernet MAC Signals Relative to MDC

Symbol	Parameter	Min	Max	Unit
f_{MDC}	MDC clock frequency[1]	–	10	MHz
$EMAC_1$	Setup for MDIO from MDC rising	10	–	ns
$EMAC_2$	Hold for MDIO from MDC rising	10	–	ns
$EMAC_3$	MDIO toggling from MDC rising[2]	0	300	ns

Note:

1. The Ethernet PHY should be selected to comply with f_{MDC} = MCK/64.
   If MCK = 200 MHz, f_{MDC} min = 200/64= 3.125 MHz.
2. For Ethernet MAC output signals, minimum and maximum access times are defined. The minimum access time is the time between the MDC rising edge and the signal change. The maximum access timing is the time between the MDC rising edge and the signal stabilization. The figure below illustrates minimum and maximum access times for $EMAC_3$.

Figure 57-31. Minimum and Maximum Access Times of Ethernet MAC Output Signals



Downloaded from Arrow.com.

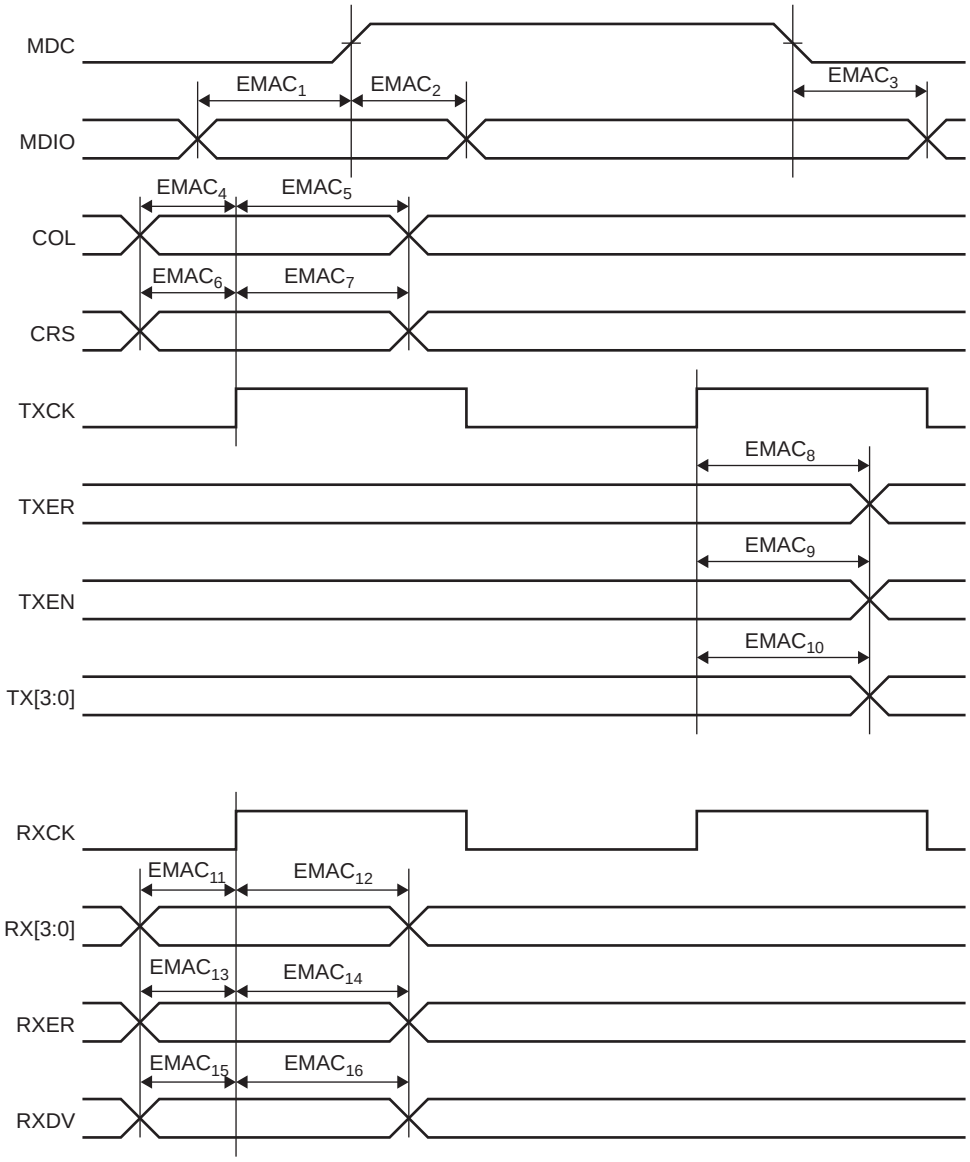
57.6.9.1 Ethernet MAC MII Mode

Table 57-33. Ethernet MAC MII Specific Signals

Symbol	Parameter	Min	Max	Unit
$EMAC_4$	Setup for COL from TXCK rising	10	–	ns
$EMAC_5$	Hold for COL from TXCK rising	10	–	ns
$EMAC_6$	Setup for CRS from TXCK rising	10	–	ns
$EMAC_7$	Hold for CRS from TXCK rising	10	–	ns
$EMAC_8$	TXER toggling from TXCK rising	10	25	ns
$EMAC_9$	TXEN toggling from TXCK rising	10	25	ns
$EMAC_{10}$	TX[3:0] toggling from TXCK rising	10	25	ns
$EMAC_{11}$	Setup for RX[3:0] from RXCK	10	–	ns
$EMAC_{12}$	Hold for RX[3:0] from RXCK	10	–	ns
$EMAC_{13}$	Setup for RXER from RXCK	10	–	ns
$EMAC_{14}$	Hold for RXER from RXCK	10	–	ns
$EMAC_{15}$	Setup for RXDV from RXCK	10	–	ns
$EMAC_{16}$	Hold for RXDV from RXCK	10	–	ns

Downloaded from Arrow.com.

Figure 57-32. Ethernet MAC MII Mode



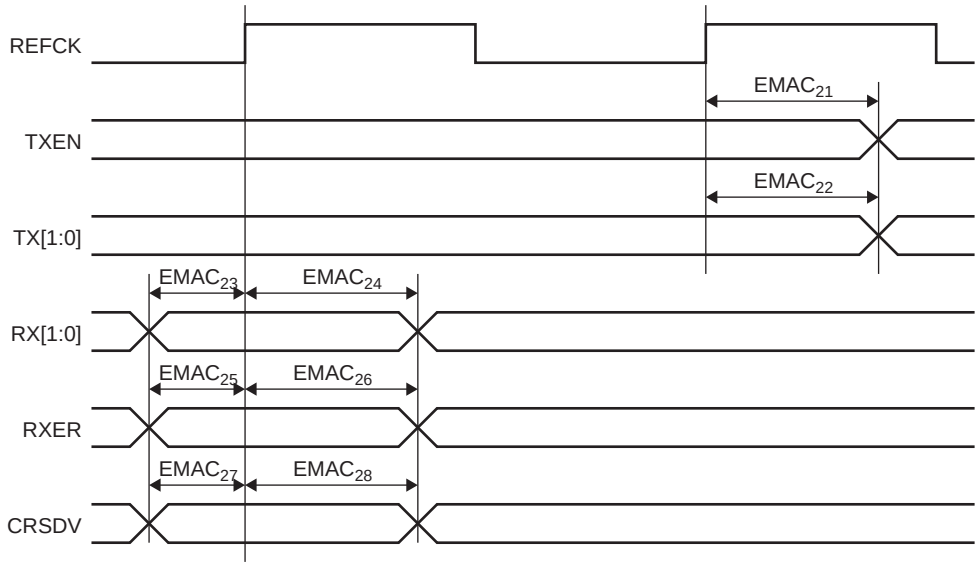
57.6.9.2 Ethernet MAC RMII Mode

Table 57-34. Ethernet MAC RMII Mode

Symbol	Parameter	Min	Max	Unit
$EMAC_{21}$	TXEN toggling from REFCK rising	2	16	ns
$EMAC_{22}$	TX[1:0] toggling from REFCK rising	2	16	ns
$EMAC_{23}$	Setup for RX[1:0] from REFCK rising	4	–	ns
$EMAC_{24}$	Hold for RX[1:0] from REFCK rising	2	–	ns
$EMAC_{25}$	Setup for RXER from REFCK rising	4	–	ns
$EMAC_{26}$	Hold for RXER from REFCK rising	2	–	ns
$EMAC_{27}$	Setup for CRSDV from REFCK rising	4	–	ns
$EMAC_{28}$	Hold for CRSDV from REFCK rising	2	–	ns

Downloaded from Arrow.com.

Figure 57-33. Ethernet MAC RMII Timings



57.7 Analog Peripheral Characteristics

57.7.1 VDDOUT25 Voltage Regulator

Table 57-35. VDDOUT25 Voltage Regulator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)	–	3.0	–	3.6	V
V_{START}	Input startup voltage	–	V_{IT+} POR VDDIN33			V
V_{STOP}	Input shutdown voltage	–	V_{IT-} POR VDDIN33			V
t_{START}	Startup time(1)	From VDDIN33 > V_{START} to VDDOUT25 set to 95% of its final value	–	–	6	ms
I_{LOAD_EXT}	External DC output current(2)	–	–	–	1	mA
$V_{DDOUT25}$	VDDOUT25 accuracy	–	2.45	–	2.55	V
I_{INRUSH}	Inrush current(3)(1)	I_{LOAD} = 0	–	–	100	mA
C_{IN}	Input decoupling capacitor(4)	–	2.2	–	–	µF
C_{OUT}	Stable output capacitor range(5)	Capacitance	1	2.2	2.7	µF
		ESR	0.01	–	0.3	Ω
R_{DIS}	Output discharge resistor value	Regulator off	100	–	300	Ω

Downloaded from Arrow.com.

Note:

1. Characterization data
2. This regulator is designed to supply SAM9X60 internal circuits (PLLs, oscillators, etc.). To supply external components, only DC current is permissible (e.g. a resistive divider).
3. Input current when charging the external output capacitor C_{OUT}.
4. A X5R or X7R ceramic capacitor connected between VDDIN33 and the device's closest GND pin is a minimum requirement to reduce the inrush current and maximize the regulator's performances.
5. To ensure stability, an external X5R or X7R ceramic output capacitor, C_{OUT}, must be connected between the VDDOUT25 and the device's closest GND pin.

57.7.2 VDDCORE Power-On-Reset

Table 57-36. Core Power Supply Power-On-Reset Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDCORE}	Supply voltage range (VDDCORE)	–	0.5	–	1.21	V
V_{IT-}	Negative-going input threshold voltage (VDDCORE)	Measured with a -10V/s ramp rate	0.68	–	0.92	V
V_{IT+}	Positive-going input threshold voltage (VDDCORE)	Measured with a +10V/s ramp rate	0.70	–	0.99	V
V_{hys}	Hysteresis voltage(1)(2)	–	20	–	60	mV
t_{RES}	Reset Time(2)	–	1	–	5	ms
t_{DET-}	V_{IT-} detection propagation time(2)	100 mV threshold overdrive	–	–	10	µs

Note:

1. $V_{hys} = V_{IT+} - V_{IT-}$
2. Characterization data

57.7.3 VDDIN33 Power-On-Reset

Table 57-37. VDDIN33 POR Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)	–	–	–	3.6	V
V_{IT-}	Negative-going input threshold voltage (VDDIN33)	Measured with a -10V/s ramp rate	2.40	–	2.60	V
V_{IT+}	Positive-going input threshold voltage (VDDIN33)	Measured with a +10V/s ramp rate	2.44	–	2.65	V
V_{hys}	Hysteresis voltage(1)(2)	–	40	–	60	mV
t_{RES}	Reset time(2)	–	1	–	6	ms
t_{DET-}	V_{IT-} detection propagation time(2)	100 mV threshold overdrive	–	–	30	µs

Note:

1. $V_{hys} = V_{IT+} - V_{IT-}$
2. Characterization data

Downloaded from Arrow.com.

57.7.4 VDDBU Power-On-Reset

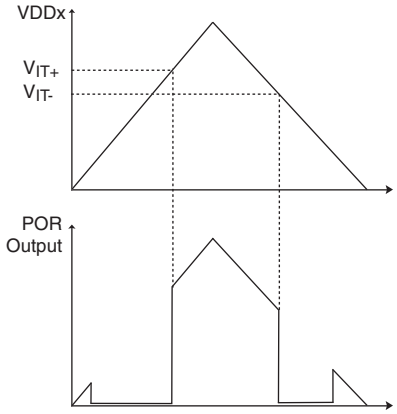
Table 57-38. VDDBU POR Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDBU}	Supply voltage range (VDDBU)	–	–	–	3.6	V
V_{IT-}	Negative-going input threshold voltage (VDDBU)	Measured with a -10V/s ramp rate	1.38	–	1.51	V
V_{IT+}	Positive-going input threshold voltage (VDDBU)	Measured with a +10V/s ramp rate	1.42	–	1.55	V
V_{hys}	Hysteresis voltage(1)(2)	–	25	–	45	mV
t_{RES}	Reset time(2)	–	1.2	–	3.3	ms
t_{DET-}	V_{IT-} detection propagation time(2)	100 mV threshold overdrive	–	–	6	µs

Note:

1. $V_{hys} = V_{IT+} - V_{IT-}$
2. Characterization data

Figure 57-34. VDDIN33/VDDCORE/VDDBU Power-On Reset Characteristics



57.7.5 Slow RC Oscillator

Table 57-39. Slow RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDBU}	Supply voltage range (VDDBU)(1)	–	V_{IT-}	–	3.6	V
f_{SLOWRC}	Frequency accuracy	VDDBU = 3.3V, T_J = 0 to +50°C	31	32	33	kHz
		VDDBU = V_{IT-} to 3.6V T_J = -40 to +125°C (2)	29	–	35	

Note:

1. In this table, V_{IT-} corresponds to the negative-going input threshold voltage of the VDDBU POR (see table VDDBU POR Characteristics). Operation of the device backup section, and in particular of the Slow RC oscillator, is granted down to the VDDBU POR V_{IT-} threshold.
2. Characterization data

Downloaded from Arrow.com.

57.7.6 Main RC Oscillator

Table 57-40. Main RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)(1)	–	3.0	–	3.6	V
I_{DDIN33}	Current consumption (VDDIN33)	–	–	–	200	µA
t_{START}	Startup time(2)	–	–	–	15	µs
f_0	Nominal output frequency	–	–	12	–	MHz
f_{ACC}	Output frequency accuracy	0°C < T_J < +50°C -20°C < T_J < +70°C(2) -40°C < T_J < +125°C(2)	–	–	±2 ±3 ±5	%
df/dV	Output frequency drift with VDDIN33(2)	V_{DDIN33} : 3.0V to 3.6V	–	–	0.01	%/V
f_{STEP}	Frequency trimming step size	–	–	50	–	kHz
Duty	Output duty cycle	–	45	50	55	%

Note:

1. This oscillator is powered by the 2.5V regulated output of the VDDOUT25 regulator, which is supplied from VDDIN33.
2. Characterization data

57.7.7 32.768 kHz Crystal Oscillator

Table 57-41. 32.768 kHz Crystal Oscillator Characteristics

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{DDBU}	Supply voltage range (VDDBU)	–		V_{IT-}(1)	–	3.6	V
I_{VDDBU}	Current consumption (VDDBU)	–		–	–	3.1	µA
t_{START}	Startup time	ESR < 50 kΩ	C_M = 0.6 fF	–	–	3.7	s
			C_M = 3 fF	–	–	0.8	
		ESR < 90 kΩ	C_M = 0.6 fF	–	–	5.0	
			C_M = 3 fF	–	–	1.0	
f_{OSC}	Operating frequency	–		32.7	–	32.8	kHz
Duty	Duty cycle	–		40	50	60	%
$C_{PARA32K}$	Internal parasitic capacitance	Between XIN32 and XOUT32		–	4	–	pF

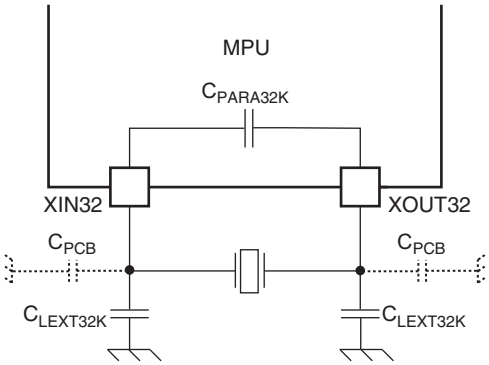
Note:

1. In this table, V_{IT-} corresponds to the negative-going input threshold voltage of VDDBU POR (see table VDDBU POR Characteristics). Operation of the device backup section is granted down to the VDDBU POR V_{IT-} threshold.

The 32.768 crystal oscillator supports a Bypass mode. Refer to the section 57.5.2.2 Input AC Characteristics.

Downloaded from Arrow.com.

Figure 57-35. 32.768 kHz Crystal Oscillator



$C_{LEXT32K} = 2x(C_{CRYSTAL} – C_{PARA32K} – C_{PCB} / 2)$

where C_{PCB} is the single-ended (ground-referenced) parasitic capacitance of the printed circuit board (PCB) on XIN32 and XOUT32 tracks. As an example, if the crystal is specified for a 12.5 pF load, with C_{PCB}=1 pF (on XIN32 and on XOUT32), $C_{LEXT32K}$ = 2 x (12.5 - 1.7 - 0.5) = 20.6 pF.

The table below summarizes recommendations for 32.768 kHz crystal selection.

Table 57-42. Recommended 32.768 kHz Crystal Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$C_{CRYSTAL}$	Crystal load capacitance	As specified by the crystal manufacturer	6	–	12.5	pF
R_S	Equivalent series resistor	–	–	50	90	kΩ
C_M	Motional capacitance	–	0.6	–	3	fF
C_{SHUNT}	Shunt capacitance	–	0.6	–	2	pF
P_{ON}	Drive level	–	0.2	–	–	µW

57.7.8 Main Crystal Oscillator

Table 57-43. Main Crystal Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)(1)	–	3.0	–	3.6	V
$I_{VDDIN33}$	Current consumption (VDDIN33)(2)	–	–	–	2.6	mA
t_{START}	Startup time	–	–	–	10	ms
f_{OSC}	Operating frequency range	–	12	–	48	MHz
Duty	Duty cycle	–	40	50	60	%
C_{PARA}	Parasitic load capacitance	On XIN and XOUT	–	1	–	pF

Note:

1. This oscillator is powered by the 2.5V regulated output of the VDDOUT25 regulator, which is supplied from VDDIN33.
2. Characterization data

The Main crystal oscillator supports a Bypass mode. Refer to the section 57.5.2.2 Input AC Characteristics.

Three sets of crystal characteristics are supported with this oscillator corresponding to three operating frequency ranges:

- Set 1: Crystal frequency is between 12 and 20 MHz. See Table 57-44.
- Set 2: Crystal frequency is between 20 and 30 MHz. See Table 57-45.

Downloaded from Arrow.com.

- Set 3: Crystal frequency is between 30 and 48 MHz. See Table 57-46.

When choosing a crystal, one and only one of these sets of characteristics must be completely satisfied.

Table 57-44. Recommended Crystal Characteristics (Set 1)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_0	Fundamental frequency	–	12	–	20	MHz
$C_{CRYSTAL}$	Load capacitance	–	6	–	17.5	pF
C_{SHUNT}	Shunt capacitance	–	–	–	3	pF
ESR	Equivalent series resistance	–	–	–	100	Ω
C_M	Motional capacitance	–	1.3	–	3.2	fF
P_{ON}	Drive level	–	–	–	150	µW

Table 57-45. Recommended Crystal Characteristics (Set 2)

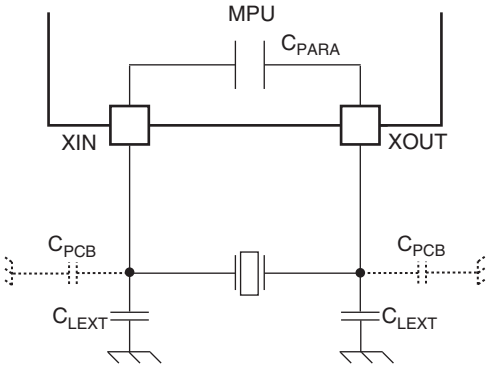
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_0	Fundamental frequency	–	20	–	30	MHz
$C_{CRYSTAL}$	Load capacitance	–	6	–	12.5	pF
C_{SHUNT}	Shunt capacitance	–	–	–	3	pF
ESR	Equivalent series resistance	–	–	–	100	Ω
C_M	Motional capacitance	–	1.3	–	3.2	fF
P_{ON}	Drive level	–	–	–	300	µW

Table 57-46. Recommended Crystal Characteristics (Set 3)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_0	Fundamental frequency	–	30	–	48	MHz
$C_{CRYSTAL}$	Load capacitance	–	6	–	10	pF
C_{SHUNT}	Shunt capacitance	With ESR_{MAX} = 60Ω	–	–	3	pF
		With ESR_{MAX} = 80Ω			1	
ESR	Equivalent series resistance	With C_{SHUNT_MAX} = 1pF	–	–	80	Ω
		With C_{SHUNT_MAX} = 3pF			60	
C_M	Motional capacitance	–	1.3	–	3.2	fF
P_{ON}	Drive level	–	–	–	400	µW

Downloaded from Arrow.com.

Figure 57-36. Main Crystal Oscillator Schematic



C_{LEXT} = 2 x ($C_{CRYSTAL}$ – C_{PARA} – C_{PCB} / 2).

where C_{PCB} is the single-ended (ground-referenced) parasitic capacitance of the printed circuit board (PCB) on XIN and XOUT tracks. As an example, if the crystal is specified for a 12.5 pF load, with C_{PCB}=1 pF (on XIN and on XOUT), C_{LEXT} = 2 x (12.5 - 1.6 - 0.5) = 20.8 pF.

57.7.9 Crystal Oscillator Design Considerations

When choosing a crystal for the 32.768 kHz Crystal Oscillator or for the Main Crystal Oscillator, several parameters must be taken into account. Important parameters are as follows:

- Crystal Load Capacitance: the total capacitance loading the crystal, including the oscillator's internal parasitics and the PCB parasitics, must match the load capacitance for which the crystal's frequency is specified. Any mismatch in the load capacitance with respect to the crystal's specification leads to inaccurate oscillation frequency.
- Crystal Drive Level: use only crystals with specified drive levels greater than the minimum recommended value. Applications that do not respect this criterion may damage the crystal.
- Crystal Equivalent Series Resistance (ESR): use only crystals with a specified ESR lower than the maximum specified value. In applications where this criterion is not respected, the crystal oscillator may not start.
- Crystal Shunt Capacitance: use only crystals with a specified shunt capacitance lower than the maximum specified value. In applications where this criterion is not respected, the crystal oscillator may not start.
- PCB Layout Considerations: to minimize inductive and capacitive parasitics associated with XIN, XOUT, XIN32, XOUT32 traces, it is recommended to minimize as much as possible their routing length. These traces must be kept away from noisy switching signals (clock, data, PWM, etc.). A good practice is to shield them with a quiet ground to avoid coupling to neighboring signals.

57.7.10 PLL Characteristics

Table 57-47. PLLA Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)(1)	–	3.0	–	3.6	V
V_{DDCORE}	Supply voltage range (VDDCORE)	–	1.02	–	1.21	V
$I_{VDDIN33}$	Current consumption (VDDIN33)	$f_{COREPLLCK}$ = 1.2 GHz	–	–	3.0	mA
$I_{VDDCORE}$	Current consumption (VDDCORE)		–	–	3.5	mA
t_{START}	Startup time(2)	–	–	–	50	µs
f_{IN}	Input frequency range	(3)	12	–	48	MHz
f_{PLLACK}	Output frequency range (PLLACK)	–	–	–	600	
$f_{COREPLLCK}$	COREPLLCK frequency range	–	600	–	1200	

Downloaded from Arrow.com.

Note:

1. This PLL is powered by the 2.5V regulated output of the VDDOUT25 regulator, which is supplied from VDDIN33.
2. Characterization data
3. For optimal setting of the PLLA, set the register PMC_PLL_ACR to the value 0x00020010.

Table 57-48. UPLL Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)(1)	–	3.0	–	3.6	V
V_{DDCORE}	Supply voltage range (VDDCORE)	–	1.02	–	1.21	V
$I_{VDDIN33}$	Current consumption (VDDIN33)	$f_{COREPLLCK}$ = 960 MHz	–	–	2.4	mA
$I_{VDDCORE}$	Current consumption (VDDCORE)		–	–	2.8	mA
t_{START}	Startup time(3)(5)	–	–	–	150	µs
f_{IN}	Input frequency range(2)	(6)	12	–	48	MHz
$f_{COREPLLCK}$	COREPLLCK frequency range	–	600	–	1200	
f_{OUT}	Output frequency range(4)	–	$f_{COREPLLCK}$ / 2			

Note:

1. This PLL is powered by an internal dedicated voltage regulator, supplied from VDDIN33, that must be started by software before enabling this PLL. Refer to 27. Clock Generator.
2. Only 12, 16, 24 or 48 MHz input frequencies are authorized to support USB-related features of the bootloader program in ROM.
3. Covers the startup time of the PLL and of its dedicated voltage regulator.
4. The post divider is hardwired in a divide-by-2 configuration.
5. Characterization data
6. For optimal setting of the PLLUTMI, set the PMC_PLL_ACR as follows:
   PMC_PLL_ACR = 0x09023010 for f_{IN} = [12 MHz, 18 MHz[
   PMC_PLL_ACR = 0x12023010 for f_{IN} = [18 MHz, 32 MHz[
   PMC_PLL_ACR = 0x1B023010 for f_{IN} = [32 MHz, 48 MHz]

57.7.11 12-bit ADC Characteristics

Table 57-49. ADC Power Supply and Voltage Reference Input Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDANA}	Supply voltage range (VDDANA)	–	3.0	–	3.6	V
I_{VDDANA}	Current consumption on VDDANA	Low Speed – $f_S \le$ 500 kS/s ADC_ACR.IBCTL = $(00)_2$	–	0.7	1.0	mA
		Full Speed – $f_S \le$ 1 MS/s ADC_ACR.IBCTL = $(01)_2$	–	1.2	1.7	
$V_{ADVREFP}$	ADVREFP input voltage range(1)	–	2.4	–	V_{DDANA}	V
$R_{ADVREFP}$	ADVREFP input resistance to ground(2)	ADC on	7.2	–	12	kΩ
		ADC off	1	–	–	MΩ
$C_{ADVREFP}$	Recommended decoupling capacitor on ADVREFP	–	1	–	–	µF

Downloaded from Arrow.com.

Note:

1. ADVREFN pin must be connected to the PCB ground plane.
2. Simulation data

Table 57-50. ADC Timing Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{CKADC}	ADC clock frequency	ADC_ACR.IBCTL = $(00)_2$	0.1	–	10	MHz
		ADC_ACR.IBCTL = $(01)_2$	0.2	–	20	
t_{CONV}	ADC conversion time(1)	–	20	–	–	t_{CKADC}
f_S	Sampling rate(2)	ADC_ACR.IBCTL = $(00)_2$	–	–	0.5	MS/s
		ADC_ACR.IBCTL = $(01)_2$	–	–	1	
t_{START}	Startup time(4)	Off to on	–	–	5	µs
t_{TRACK}	Track and hold time(3)	–	300	–	–	ns

Note:

1. $t_{CONV} = t_{CH} + t_{TRACK} + 14 \times t_{CKADC}$ with $t_{CKADC} = 1 / f_{CKADC}$. $t_{CH} = 0$ when the ADC operates in the same input mode (single-ended, pseudo-differential or differential) for the current conversion than for the previous one. $t_{CH} = 2$ when the ADC input mode is changed to perform the current conversion.
2. $f_S = 1 / t_{CONV}$.
3. See section Track and Hold Time versus Source Impedance – Sampling Rate.
4. Characterization data

Table 57-51. ADC Analog Input Characteristics

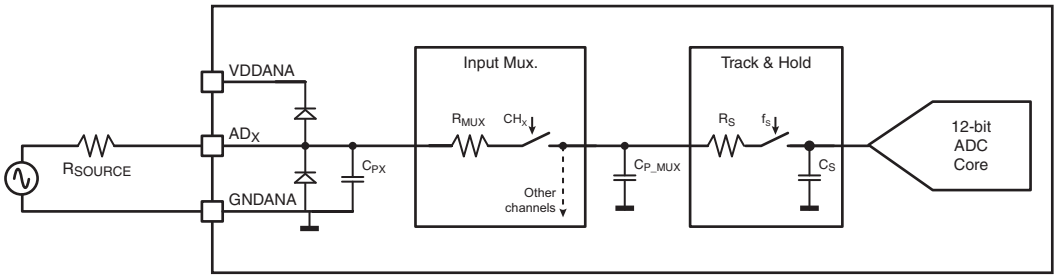
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{FS}	Analog input full scale range(1)	ADC_CCR.DIFFx = 0	0	–	V_{ADVREF}	V
		ADC_CCR.DIFFx = 1	$-V_{ADVREF}$	–	V_{ADVREF}	
V_{INCM}	Common mode input range in Differential Input mode(2)	ADC_CCR.DIFFx = 1	0.4 x V_{DDANA}	–	0.6 x V_{DDANA}	V
C_S	ADC sampling capacitance(5)	–	–	–	3	pF
C_{P_ADx}	ADx input parasitic capacitance(3)(5)	ADx pin configured as analog input	–	–	7	
R_{ON}	Internal series resistor(3)	–	–	–	2	kΩ
Z_{IN}	Common mode input impedance(4)(5)	On ADx pin	$1 / (f_S.C_S)$	–	–	Ω

Note:

1. $V_{FS} = (V_{ADx} - V_{GNDANA})$ in Single-ended mode, $V_{FS} = (V_{ADx} - V_{AD11})$ in Pseudo-differential mode, and $V_{FS} = (V_{ADx} - V_{ADx+1})$ in Differential mode.
2. $V_{INCM} = (V_{ADx} + V_{ADx+1}) / 2$.
3. With respect to the equivalent model of figure Equivalent Model of the Acquisition Path.
4. Assuming conversion on one single channel.
5. Simulation data

Downloaded from Arrow.com.

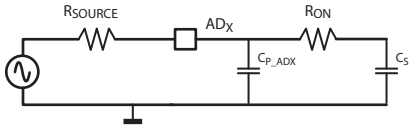
Figure 57-37. Acquisition Path Block Diagram



For tracking time calculation, during the sampling phase of the converter, this acquisition path can be reduced to the equivalent model provided in the following figure, where:

- $R_{ON} = R_{MUX} + R_S$
- $C_{P_ADX} = C_{PX} + C_{P_MUX}$

Figure 57-38. Equivalent Model of the Acquisition Path



See section Track and Hold Time versus Source Impedance – Sampling Rate for further details on how to use this model.

In the following table, unless otherwise specified, the specifications are given for two speed operating ranges.

- Source resistance = 50 Ω
- ADC_EMR.OSR<2:0> = $(000)_2$
- Low-speed
  - f_{CKADC} = 10 MHz, f_S = 500 kS/s
  - ADC_ACR.IBCTL = $(00)_2$
- High-speed
  - f_{CKADC} = 20 MHz, f_S = 1 MS/s
  - ADC_ACR.IBCTL = $(01)_2$

Table 57-52. Static Performance Characteristics(1)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
RES_{ADC}	Native ADC resolution	–	–	12	–	Bit
INL	Integral non-linearity	Single mode	-3	–	+3	LSB
		Differential mode	-2	–	+2	
DNL	Differential non-linearity	Single mode	-2	–	+2	
		Differential mode	-1		+1	
OE	Offset error(2)	–	-4	–	4	
GE	Gain error(2)	–	-4	–	4	

WARNING ADC channels should be selected knowing that toggling the QSPI I/Os may decrease ADC channel 4 and 0 performances.

Downloaded from Arrow.com.

Note:

1. In this table, errors are expressed in LSB where:
   – LSB = V_{ADVREF} / 2^{12} in Single-ended mode (ADC_CCR.DIFFx = 0 and ADC_PDR.PDIFFx = 0)
   – LSB = V_{ADVREF} / 2^{11} in Differential or Pseudo-differential mode (ADC_CCR.DIFFx = 1)
2. Error with respect to the best fit line method.

57.7.11.1 Track and Hold Time versus Source Impedance – Sampling Rate

Referring to figure Equivalent Model of the Acquisition Path, during its tracking phase, the 12-bit ADC charges its sampling capacitor C_S through various serial resistors modeled as R_{SOURCE} (source output resistor) and R_{ON} (multiplexer series resistor and sampling switch series resistor). In case of high output source resistance (e.g. a low power resistive divider), the tracking time must be increased to ensure full settling of the sampling capacitor voltage. Of course, programming a long tracking time may impact the sampling frequency (f_S). The following formula gives the minimum tracking time that ensures a 12-bit accurate settling:

$t_{TRACK} \geq 8 \times (R_{SOURCE} + R_{ON}) \times C_S$

The ADC Controller (ADCC) counts the tracking time in ADC clock cycles (t_{CKADC}). This time can be adjusted between 6 and 54 cycles by means of the fields ADC_MR.TRACKTIM and ADC_EMR.TRACKX4. At maximum ADC clock frequency (20 MHz), the maximum tracking time that can be programmed is 2.7 µs. This limits 12-bit accurate sampling to sources having R_{SOURCE} in the 100 kΩ range. To overcome this limitation, the ADC clock frequency can be decreased.

The following two examples show typical use cases of tracking time and sampling frequency calculation.

Example 1: Calculated tracking time is lower than the default (minimum) tracking time.

- Assuming: f_{CKADC} = 8 MHz (t_{CKADC} = 125 ns), R_{SOURCE} = 10 kΩ
- The minimum required track time is: t_{TRACK} = 8 x (10 kΩ + 2 kΩ) x 3pF = 288 ns
- t_{TRACK} is less than the minimum tracking time (6 x t_{CKADC} = 750 ns): set TRACKTIM = 0 and TRACKX4 = 0
- The real tracking time is: 6 x t_{CKADC} (750 ns) and the conversion time is t_{CONV} = t_{TRACK} + 14 x t_{CKADC}= 20 x t_{CKADC}
- The sampling rate is: f_S = 8 MHz / 20 = 400 kS/s
- The maximum allowable source resistance is: R_{SOURCE_MAX} = (6 x t_{CKADC}) / (3pF x 8) - 2 kΩ= 29.25 kΩ

Example 2: Calculated tracking time is greater than the default (minimum) tracking time.

- Assuming: f_{CKADC} = 20 MHz (t_{CKADC} = 50 ns), R_{SOURCE} = 10 kΩ
- The minimum required track time is: t_{TRACK} = 8 x (10 kΩ + 8 kΩ) x 3 pF= 432 ns
- t_{TRACK} is greater than the minimum tracking time (6 x t_{CKADC} = 300 ns): set TRACKTIM=4 and TRACKX4=1
- The real track time is: (4 x (4 + 1) - 10) = 10 x t_{CK_ADC} = 500 ns
- The conversion time is: t_{CONV} = t_{TRACK} + 14 x t_{CKADC}= 24 x t_{CKADC}
- The sampling rate is: f_S = 20 MHz / 24 = 833.3 kS/s
- The maximum allowable source resistance is: R_{SOURCE_MAX} = (10 x t_{CKADC}) / (3 pF x 8) - 8 kΩ = 12.8 kΩ

57.7.12 HS USB Transceiver Characteristics

The device complies with all voltage, power, and timing characteristics and specifications as set forth in the USB 2.0 Specification. Refer to the USB 2.0 Specification for more information.

Table 57-53. USB 2.0 Transceiver Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{VDDIN33}$	Supply voltage range (VDDIN33)	–	3.0	–	3.6	V
$I_{VDDIN33}$	HS transceiver current consumption (VDDIN33)(1)	Bias circuits enabled only	–	–	15	mA
		HS	–	–	20	
		LS / FS - 0m cable(3)	–	–	5	
		LS / FS - 5m cable(3)	–	–	30	

Downloaded from Arrow.com.

Note:

1. Current consumption is per USB port.
2. Including 1 mA due to pull-up/pull-down current.

57.8 Power Consumption in Active Mode

57.8.1 Processor Power Consumption in Active Mode

The table below gives the processor power consumption in the following conditions:

- f_{CPU_CLK} = from 200 MHz to 600 MHz
- f_{MCK} = 200 MHz
- L1 caches enabled
- The ARM926EJ-S core executes a Dhrystone benchmark from the (internal) SRAM0
- Code compiled with speed optimization
- Peripheral clocks disabled
- Current measured as per the figure below

Figure 57-39. Current Measurement on VDDCORE

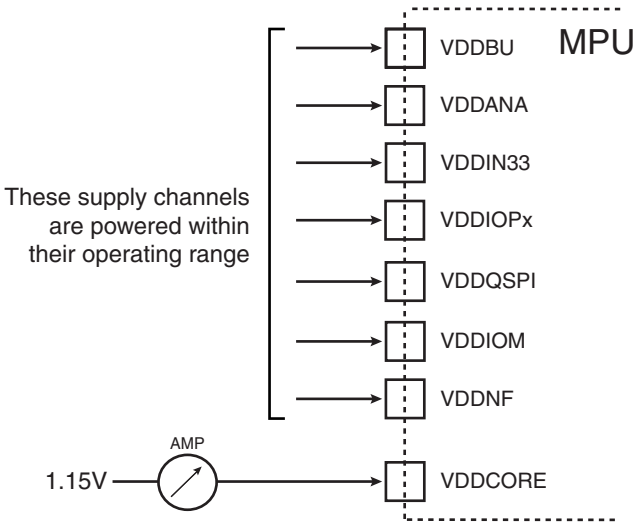
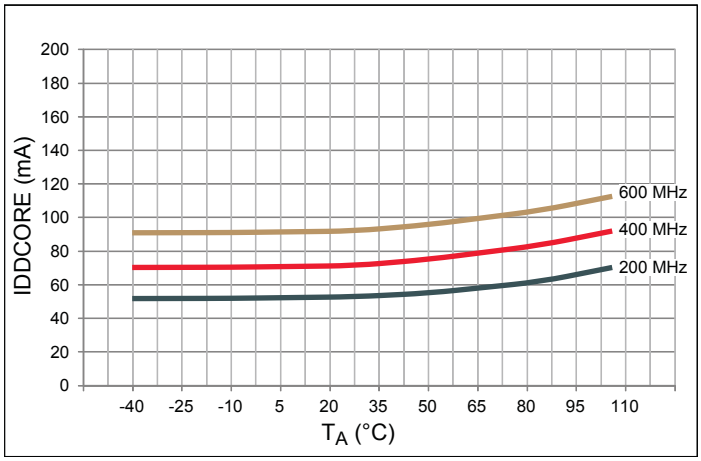


Table 57-54. Processor Power Consumption Running a Dhrystone Benchmark from SRAM0

f_{CPU_CLK}	IDDCORE vs Ambient Temperature (T_A)						Unit
	-40°C	-10°C	25°C	50°C	85°C	105°C	
200 MHz	52.8	53.1	54.2	56.1	62.2	69.7	mA
400 MHz	73.7	74.0	75.2	76.9	83.5	90.9	
600 MHz	93.2	93.4	94.5	96.5	103.0	110.3	

Downloaded from Arrow.com.

Figure 57-40. Processor Current Consumption Running a Dhrystone Benchmark from SRAM0



57.8.2 System Power Consumption in Applicative Use Cases

Table Table 57-56 gives the processor power consumption in the following conditions:

- f_{CPU_CLK} = 600 MHz
- f_{MCK} = 200 MHz
- I & D caches enabled
- External SDRAM memory: 16-bit DDR2 operating at 200 MHz
- Uses cases are run on Linux®
- Current consumptions are measured according to figure Current Measurement Schematic for Applicative Use Cases. Note that external component current consumptions are not counted. Either the 24-bit display or the 8-bit image sensor is connected to the SAM9X60, but not both.

Table 57-55. Use Case Definition

Use Case	Description of Test
1	After Linux boot root prompt
2	Video playback from USB mass storage (mp4 format, 480x272 @ 24fps, bitrate = 369 kbps)
3	Waveform (WAV) audio file decoding and playback on I2S. WAV file on USB mass storage
4	SAM9X60 running as iPerf server
5	Run Bonnie++ on USB mass storage
6	SAM9X60 running 2D graphic benchmark
7	Image sensor video recording with QVGA resolution - 10 fps

Downloaded from Arrow.com.

Figure 57-41. Current Measurement for Applicative Use Cases

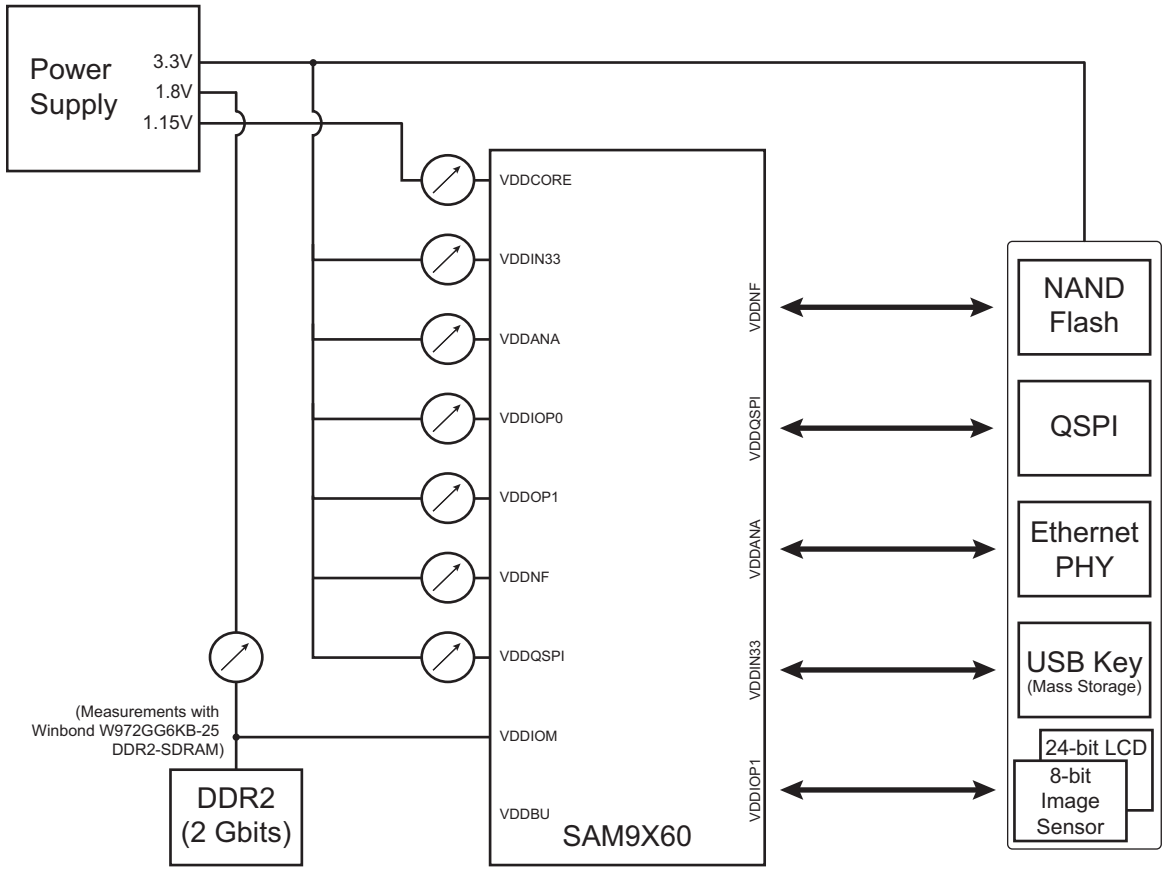


Table 57-56. Power Consumption in Applicative Use Cases

Use Case	VDDCORE (1.15V)	VDDIOM (1.8V)	VDDIN33 (3.3V)	VDDNF (1.8V)	VDDPIOP0 (3.3V)	VDDPIOP1 (3.3V)	Unit	Total Power	Unit
1	60.25	35.7	10	3.9	0.17	0.011	mA	174	mW
2	101	72.7	15.8	3.5	0.71	8.7		337	
3	64	41.4	15.8	3.75	0.75	0.003		209	
4	77.5	92.7	10	3.55	0.168	0.006		296	
5	102	60	19.8	3.2	0.183	0.006		297	
6	98	97	10	3.4	0.71	8.5		357	
7	101	61.1	15.8	3.1	0.84	26		372	

57.9 Operation and Power Consumption in Low-Power Modes

The SAM9X60 features four low-power modes summarized in the following table. A detailed description of each mode is given in the following sections.

Downloaded from Arrow.com.

Table 57-57. Low-power Modes Summary

		Low-power Mode			
		Backup Mode	ULP1	ULP0	Idle
VDDBU power		✓	✓	✓	✓
VDDCORE power		✗	✓	✓	✓
SDRAM or DDR power		✗	✓	✓	✓
I/O power		✗	✓	✓	✓
DDR Self-refresh		–	✓	✓	✗
Main crystal oscillator		–	✗	✗	✓
Main RC oscillator		–	✗	✗	✓
Main clock (MAINCK)	Source (MOSCSEL)	–	Main RC osc.	Main RC osc.	User-defined
	Status	–	✗	✗	✓
Master clock (MCK)	Source (CSS)	–	MAINCK	MD_SLCK (divided by 64)	User-defined (PLLA, etc.)
	Status	–	✗	✓	✓
CPU clock (HCLK)		–	✗	✓	✗
ARM926EJ-S state		–	Clocks stopped	WFI	WFI
Mode entry		Use SHDWC to shutdown all power supplies except VDDBU	Context saved in DDR DDR in Self-refresh ULP1 bit	Context saved in DDR DDR in Self-refresh WFI	WFI
Wake-up	Sources	RTC alarm WKUP0 pin	RTC / RTT alarm USB resume Wake-On-LAN WKUP1..13	Any not masked interrupt	Any not masked interrupt
	Event sampling clock	MD_SLCK	Asynchronous	MD_SLCK	User-defined (MCK, periph_clk, etc.)
Master clock (MCK) at wake-up		MAINCK (defaults to Main RC osc.)	MAINCK (configured to Main RC osc.)	MD_SLCK	User-defined
Power consumption		< 50 µA @ 85°C on VDDBU	< 4 mA @ 85°C on VDDCORE	< 6 mA @ 85°C on VDDCORE	–
Wake-up time (time to fetch first instruction)		–	< 15 µs	(2)	A few HCLK cycles

Downloaded from Arrow.com.

Note:

1. ✓/✗ means either powered/unpowered for a power source, on/off for an oscillator, active/inactive for a clock signal, or in/out of Self-refresh mode for the external SDRAM.
2. This value depends on the frequency selected for ULP0.

57.9.1 Backup Mode

57.9.1.1 Operation

The Backup mode is designed to serve prolonged power-down periods of the processor. In this mode, only the backup area of the device powered by VDDBU is maintained (RTC, GPBR, SHDWC, BSCR). This mode is entered by shutting down all the power rails of the device except VDDBU. It is good practice to use the SHDN output of the Shutdown Controller (SHDWC) for this purpose.

The SAM9X60 exits Backup mode when an active wake-up event is triggered by the Shutdown Controller (SHDWC). Upon this wake-up event, the SHDWC automatically toggles its SHDN output back to VDDBU and this signal typically helps to restart all the power supply channels at board level.

The wake-up events to exit Backup mode are:

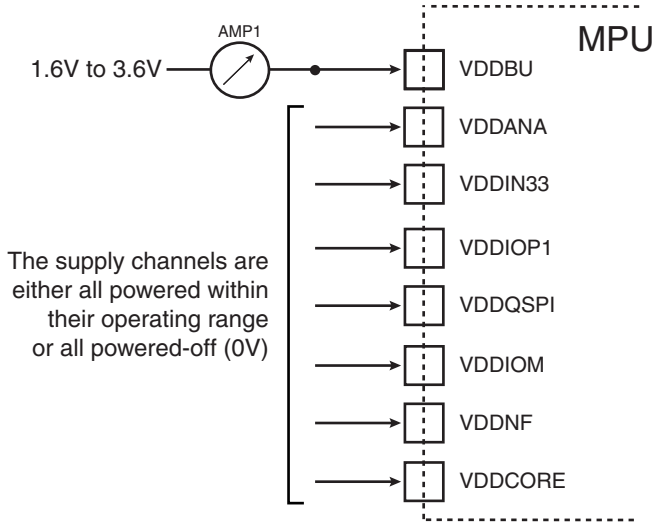
- WKUP0 pin (level transition, configurable debouncing)
- RTC alarm

57.9.1.2 Power Consumption

Backup mode configuration and measurements are defined as follows:

- POR backup on VDDBU is enabled
- RTC running
- Force wake-up (FWUP) enabled
- Current measurement as per the following figure

Figure 57-42. Measurement on VDDBU



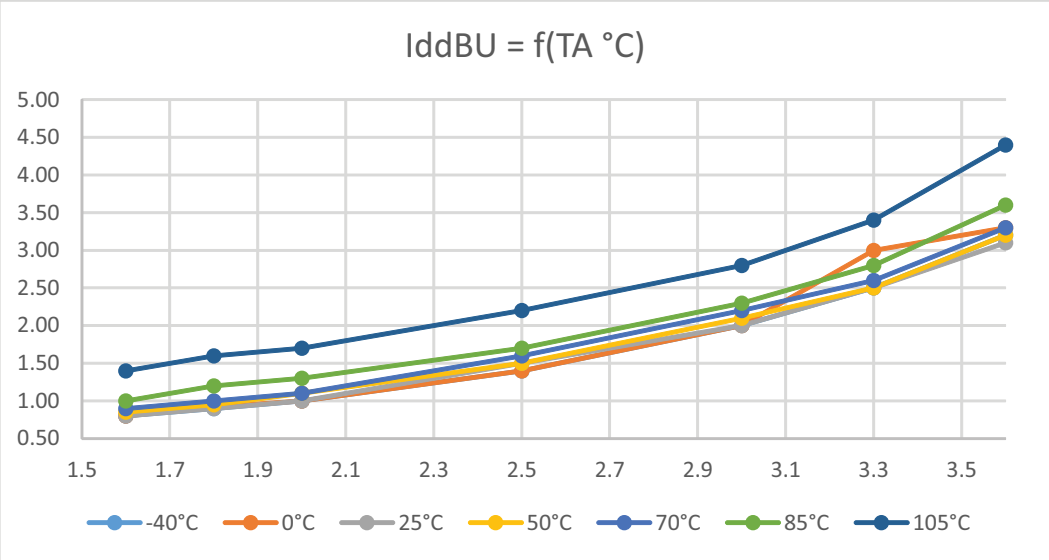
Note: Measurement done on typical process.

Downloaded from Arrow.com.

Table 57-58. Typical Current Consumption In Backup Mode

VDDBU (V)	-40°C	0°C	25°C	50°C	70°C	85°C	105°C	Unit
3.6	3.20	3.30	3.10	3.20	3.30	3.60	4.40	µA
3.3	2.50	3.00	2.50	2.50	2.60	2.80	3.40	
3	2.00	2.00	2.00	2.10	2.20	2.30	2.80	
2.5	1.40	1.40	1.50	1.50	1.60	1.70	2.20	
2	1.00	1.00	1.00	1.10	1.10	1.30	1.70	
1.8	0.90	0.92	0.90	0.95	1.00	1.20	1.60	
1.6	0.80	0.80	0.80	0.85	0.90	1.00	1.40	

Figure 57-43. Typical Current Consumption in Backup Mode



57.9.2 ULP0, ULP1 and Idle Modes

- In these three low-power modes, the SAM9X60 power supplies are all applied within their operating range. The power reduction is achieved by reducing the frequency or even stopping the clock signals of the processor and/or its peripherals.
- In Idle mode, only the processor clock is affected when in ULP0 and ULP1 modes; the clocks feeding both the processor and its peripherals are slowed down.
- In ULP0 or ULP1 mode, the SAM9X60 is placed in Retention mode and is able to resume on wake-up events (any interrupt or hardware event). This mode is a combination of the Wait for Interrupt mode of the ARM core and the system clocks frequency reduction or shut-off.

A detailed description of each mode is given in the following sections and is followed by a power consumption section dedicated to those modes.

57.9.2.1 ULP0 Operation

The ULP0 mode maintains very low frequency clocks (MCK, CPU_CLK) in the system to wake up on any interrupt. The selection of the clock frequency depends on the current consumption target versus the desired wake-up time. The higher the frequency, the higher the power consumption and the faster the wake-up time.

The sequence to enter ULP0 mode is detailed below. The code used to enter this mode must be executed out of the internal SRAM0.

1. Set the DDR to Self-refresh mode.

Downloaded from Arrow.com.

2. Disable PMC_SCDR.DDRCLK.
3. Set the interrupts to wake up the system.
4. Disable all peripheral clocks.
5. Set the I/Os to an appropriate state and disable the USB transceivers (refer to 23. Special Function Registers (SFR)).
6. Switch the system clock to Slow Clock.
7. Set the SRAM memories to Light Sleep mode in SFR_LS.
8. Disable the PLLs, the main oscillator and the 12 MHz RC oscillator.
9. Enter the Wait for Interrupt mode.

Wake-up from ULP0 mode is triggered by any enabled interrupt. When resuming, the software reconfigures the system (oscillator, PLL, etc.) in the same state as before WFI. Disable Light Sleep mode for SRAM memories in the SFR_LS register prior to returning to full speed operation.

57.9.2.2 ULP1 Operation

Unlike the ULP0 mode, all the clocks are off in the ULP1 mode, and the number of wake-up sources is limited to the list below:

- WKUP1..13 pins (level transition, configurable debouncing)
- RTC/RTT alarm
- USB Resume from Suspend mode
- Wake-On-LAN events from EMAC peripherals

The sequence to enter ULP1 mode is detailed below. The code used to enter this mode must be executed out of the internal SRAM0.

1. Set the DDR to Self-refresh mode.
2. Disable PMC_SCDR.DDRCLK.
3. Set the events to enable a system wake-up.
4. Disable all peripheral clocks.
5. Set the I/Os to an appropriate state and disable the USB transceivers (refer to 23. Special Function Registers (SFR)).
6. Switch the system clock to the 12 MHz RC oscillator.
7. Set the SRAM memories to Light Sleep mode in SFR_LS.
8. Disable the PLLs and the main oscillator.
9. Enter ULP1 mode by setting the ULP1 bit in CKGR_MOR.

When resuming, the software reconfigures the system (oscillator, PLL, etc.) in the same state as before entering ULP1. In particular, the SRAM memories Light Sleep mode must be disabled as soon as possible in the wake-up process.

57.9.2.3 Idle Mode Operation

The purpose of Idle mode is to optimize power consumption of the device versus response time. In this mode, only the processor clock is stopped. The peripheral clocks, including the DDR controller clock, can be enabled. The current consumption in this mode is application-dependent.

This mode is entered via the Wait for Interrupt (WFI) instruction and CPU_CLK disabling.

The processor can be awakened from an interrupt. The system resumes where it was before entering WFI mode.

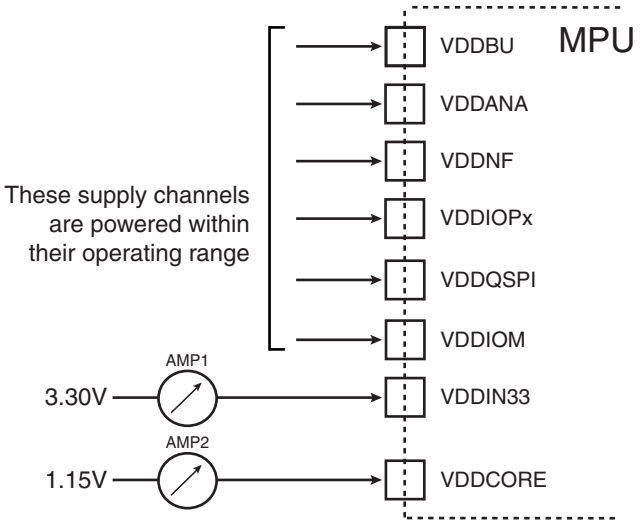
57.9.2.4 Power Consumption in ULP0, ULP1 and Idle Modes

- Parts are from typical process.
- All power supply inputs are powered within their operating range and in particular VDDCORE = 1.15V
- There is no consumption on the device's I/Os. To reach the best possible power consumption figures, it is of prime importance to set each I/O of the device to an appropriate state (pull-up/pull-down,etc..) with respect to the external components connected to these I/Os.
- USB tranceivers are disabled.
- All peripheral clocks are disabled.

Downloaded from Arrow.com.

- Current measurement as per the following figure

Figure 57-44. Current Measurement in ULP0, ULP1 and Idle Modes



All measurements below are based on typical processes.

Table 57-59. Typical Current Consumption on VDDCORE in ULP0, ULP1 and Idle Modes

Low Power Mode\T_A (°C)	-40°C	0°C	25°C	50°C	70°C	85°C	105°C	Unit	Wakeup Time	Unit
Idle (MCK=200 MHz)	46.9	47.2	47.4	48.6	50	52	56.3	mA	0.5	µs
ULP1	0.1	0.2	0.43	1.1	2.3	3.8	7.2		12	
ULP0 12 MHz	1.2	1.35	1.75	2.7	4.4	6.5	11		12	
ULP0 750 KHz	0.24	0.38	0.76	1.75	3.4	5.5	10.1		170	
ULP0 187 KHz	0.17	0.32	0.68	1.7	3.3	5.4	9.9		720	
ULP0 32 KHz	0.1	0.24	0.6	1.6	3.3	5.4	9.9		4200	

Table 57-60. Typical Current Consumption on VDDIN33 in ULP0, ULP1 and Idle Modes

Low Power Mode\T_A (°C)	-40°C	0°C	25°C	50°C	70°C	85°C	105°C	Unit
Idle (MCK=200 MHz)	2.2	2.2	2.1	2.1	2.1	2.1	2.1	mA
ULP1	70	70	72	73	74	75	77	µA
ULP0 12 MHz	188	192	195	198	201	203	207	
ULP0 750 KHz	188	192	195	198	201	203	207	
ULP0 187 KHz	188	192	195	198	201	203	207	
ULP0 32 KHz	69	70	71	73	74	75	77	

The following figures show the power consumption of the SAM9X60 in ULP0, ULP1 and Idle modes. These figures account for the current consumption in VDDCORE and VDDIN33: Power (mW) = I_{DDIN33} x 3.3V + I_{DDCORE} x 1.15V with I_{DDIN33} and + I_{DDCORE} as measured in the above table.

Downloaded from Arrow.com.

Figure 57-45. Typical Power Consumption in Idle Mode

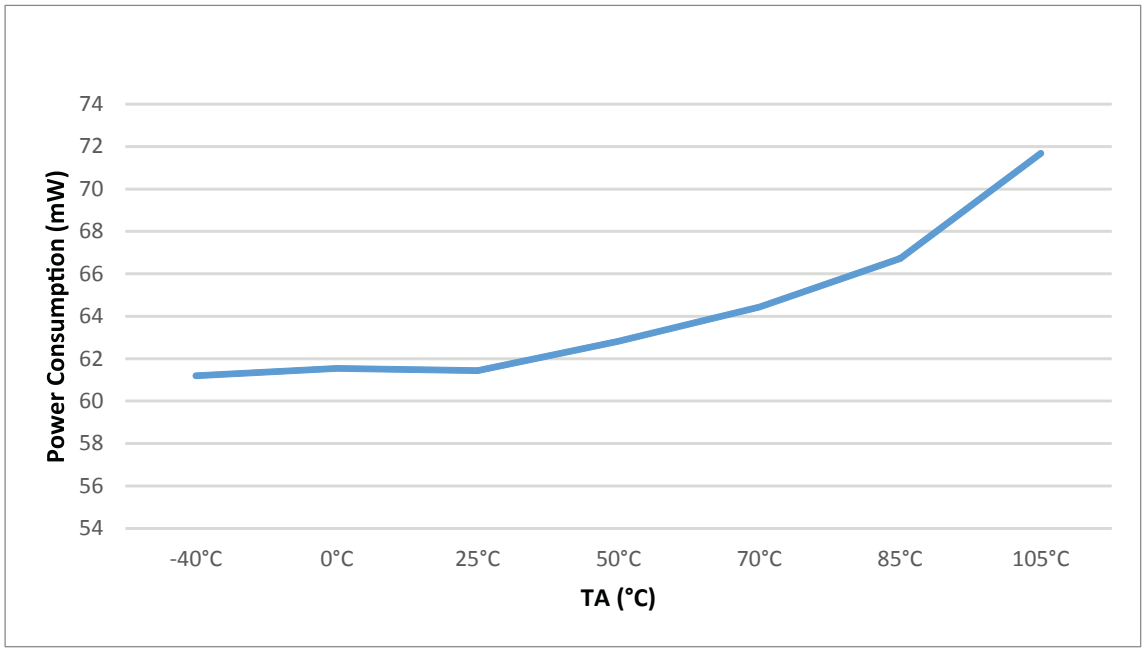
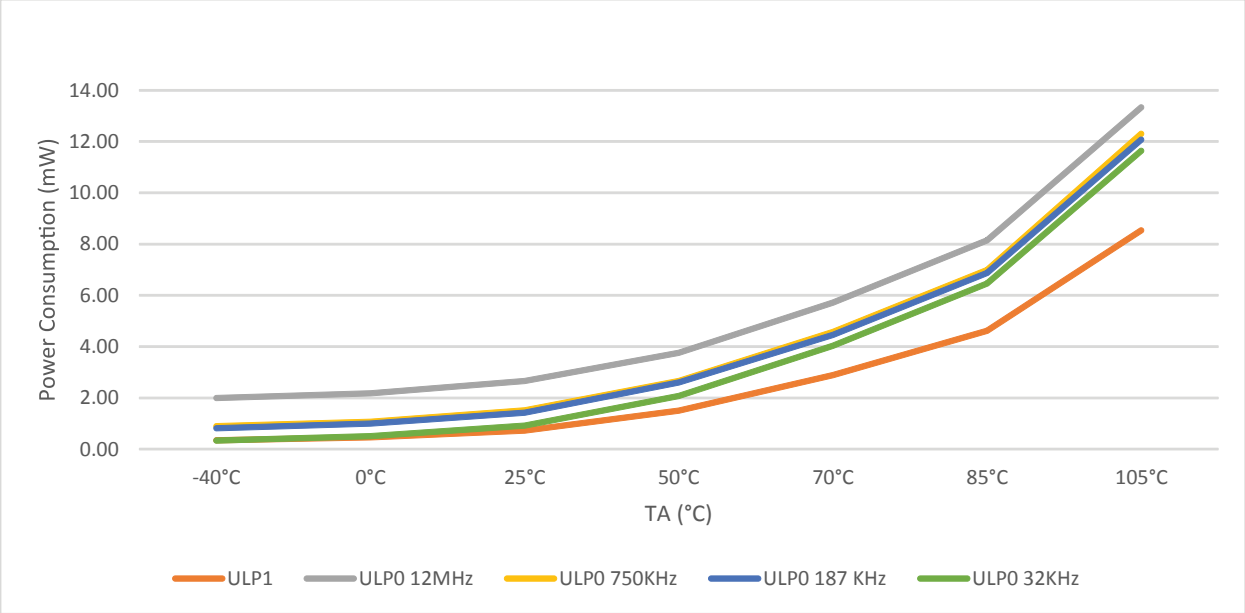


Table 57-61. Typical Power Consumption in ULP0, ULP1 and Idle Modes

Low Power Mode \T_A (°C)	-40°C	0°C	25°C	50°C	70°C	85°C	105°C	Unit
Idle Mode (MCK=200 MHz)	61.195	61.54	61.44	62.82	64.43	66.73	71.675	mW
ULP1	0.35	0.46	0.73	1.51	2.89	4.62	8.53	
ULP0 12 MHz	2.00	2.19	2.66	3.76	5.72	8.14	13.33	
ULP0 750 KHz	0.90	1.07	1.52	2.67	4.57	6.99	12.30	
ULP0 187 KHz	0.82	1.00	1.43	2.61	4.46	6.88	12.07	
ULP0 32 KHz	0.34	0.51	0.92	2.08	4.04	6.46	11.64	

Downloaded from Arrow.com.

Figure 57-46. Typical Power Consumption in ULP Mode 0 and ULP Mode 1



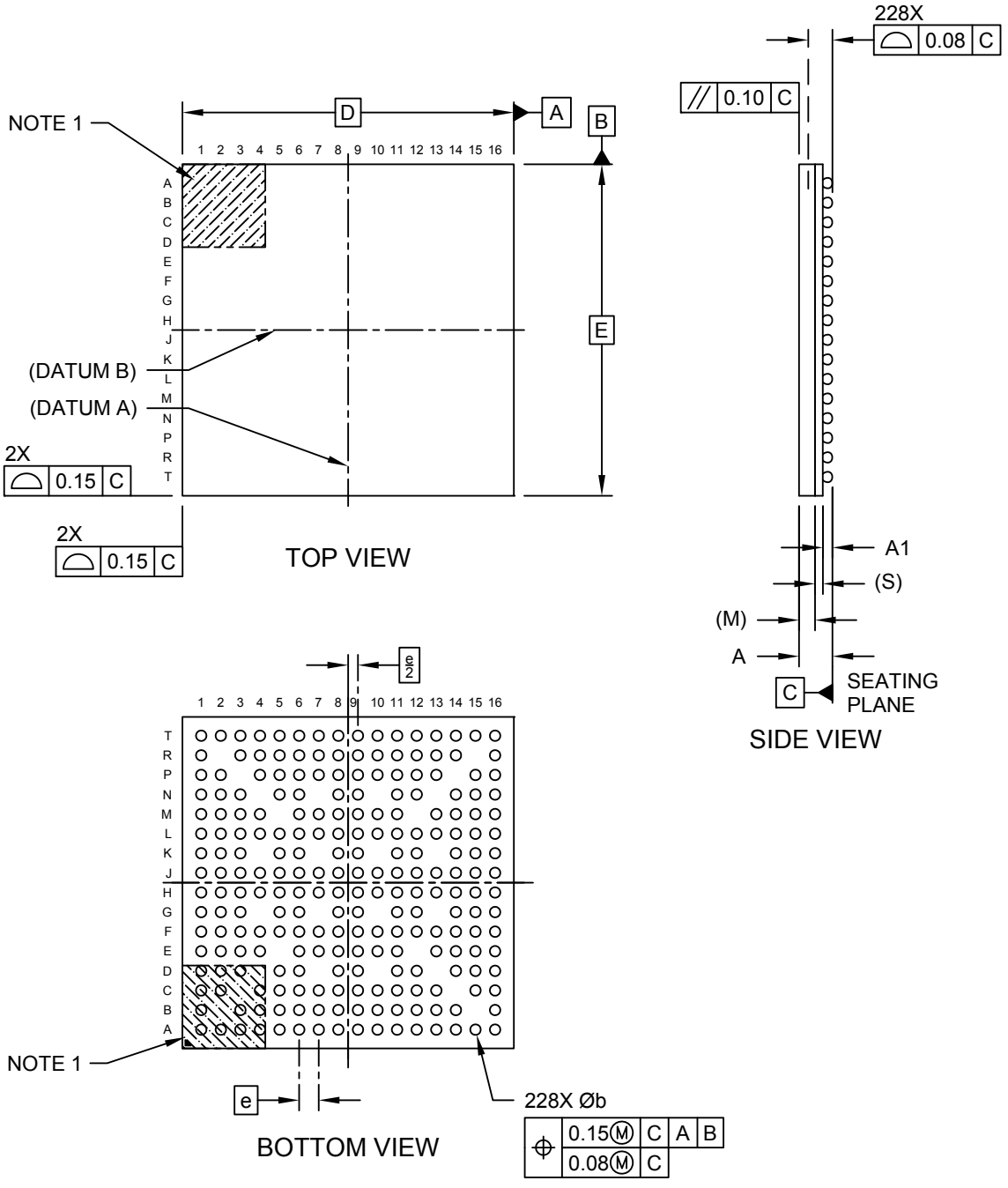
Downloaded from Arrow.com.

58. Mechanical Characteristics

58.1 228-ball TFBGA Mechanical Characteristics

228-Ball Thin, Fine Pitch Ball Grid Array (DWB) - 11x11 mm Body [TFBGA]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

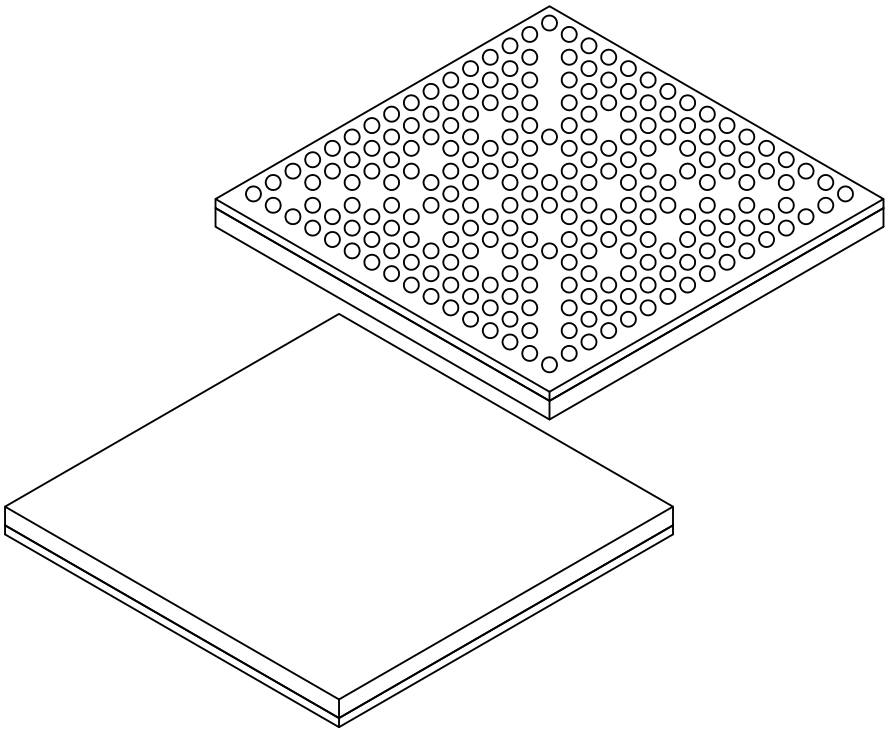


Microchip Technology Drawing C04-21484 Rev B Sheet 1 of 2

Downloaded from Arrow.com.

228-Ball Thin, Fine Pitch Ball Grid Array (DWB) - 11x11 mm Body [TFBGA]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N		228	
Pitch	e		0.65 BSC	
Overall Height	A	-	-	1.20
Ball Height	A1	0.22	0.27	0.32
Substrate Thickness	S		0.26 REF	
Mold Cap Thickness	M		0.53 REF	
Overall Length	D		11.00 BSC	
Overall Width	E		11.00 BSC	
Terminal Diameter	b	0.32	0.37	0.42

Notes:

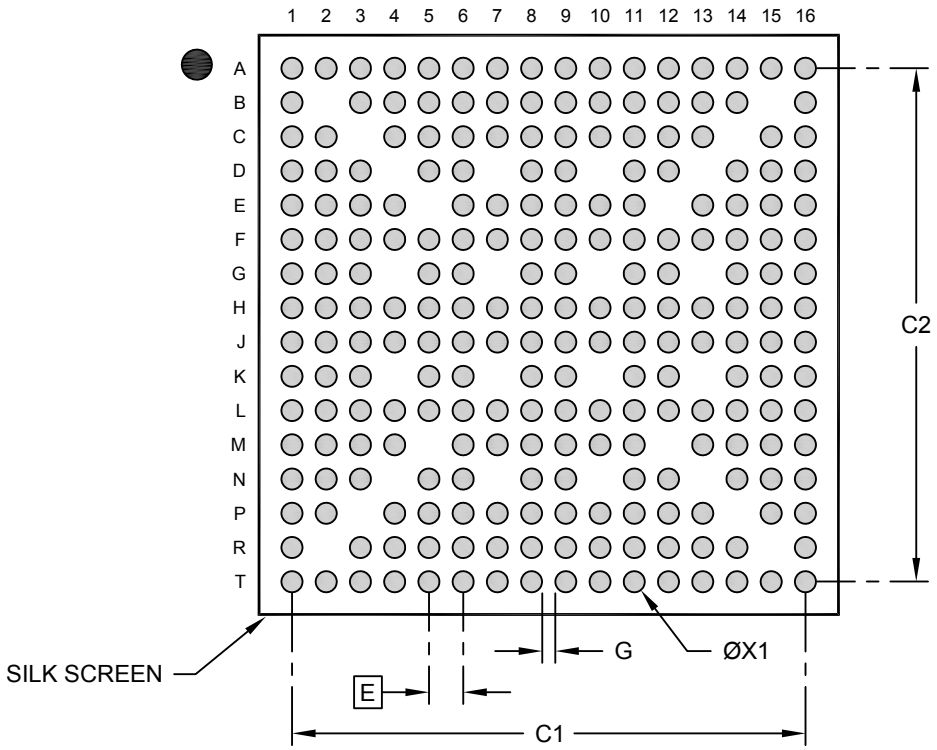
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-21484 Rev B Sheet 1 of 2

Downloaded from Arrow.com.

228-Ball Thin, Fine Pitch Ball Grid Array (DWB) - 11x11 mm Body [TFBGA]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Contact Pad Spacing	C1		9.75	
Contact Pad Spacing	C2		9.75	
Contact Pad Diameter (X228)	X1			0.42
Contact Pad to Contact Pad	G1	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-23484 Rev B

Downloaded from Arrow.com.

Table 58-1. 228-ball TFBGA Package Characteristics

Moisture Sensitivity Level	3

Table 58-2. Device and 228-ball TFBGA Package Weight

247	mg

Table 58-3. Package Reference

JEDEC Drawing Reference	NA
J-STD-609 Classification	e8

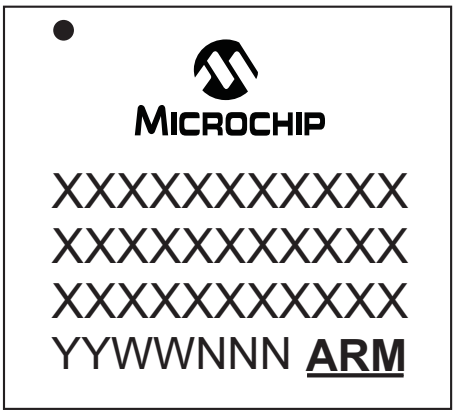
Table 58-4. 228-ball TFBGA Package Information

Ball Land	0.40 mm
Nominal Ball Diameter	0.35 mm
Solder Mask Opening	0.30 mm
Solder Mask Definition	NSMD
Solder	SAC105

Downloaded from Arrow.com.

59. Marking

Top marking follows the scheme below:



with possible values:

Line	Description	Values
1	Company logo	Microchip logo
2	Company name	Microchip
3	Device name	SAM9X60
4	Temperature code / Packaging code	V / DWB
5	Not used	–
6	Lot traceability, ARM logo	YYWWNNN ARM

Downloaded from Arrow.com.

60. Ordering Information

Ordering Code	Package	Carrier Type	Operating Temperature Range
SAM9X60-V/DWB	TFBGA228	Tray	-40°C to +105°C
SAM9X60T-V/DWB	TFBGA228	Tape and reel	-40°C to +105°C

Downloaded from Arrow.com.

61. Revision History

61.1 DS60001579B - 02/2020

Section	Changes
All sections	Changed order of bits (now from MSB to LSB) in Register Summary tables
Safety and Security Features	Removed "Classical Advanced Software Crypto Library (CASCL)" from Security Features table
Peripherals	Updated FLEXCOM Features table
Boot Strategies	Updated NAND Flash Boot: NAND Flash Detection introduction
OTP Memory Controller (OTPC)	OTPC_CR: updated READ bit description
Clock Generator	Updated Main Clock (MAINCK) Block Diagram
Power Management Controller (PMC)	Updated: • General Clock Distribution Block Diagram • List of write-protected registers in Register Write Protection PMC_CPU_CKR: corrected MDIV field width
External Bus Interface (EBI)	Updated Multi-Port DDR and SDRAM Controllers
AHB Multiport DDR-SDRAM Controller (MPDDRC)	DDR2-SDRAM Initialization: added TRFC constraint content Monitor: added monitor use example Updated: • Quality Of Service Arbitration • Description • Change Frequency During Self-Refresh Mode with Low-power DDR-SDRAM Devices
Flexible Serial Communication Controller (FLEXCOM)	Added introduction in USART FIFO Mode Register. USART FIFO Level Register, USART FIFO Interrupt Mask Register, SPI Status Register, SPI FIFO Mode Register, SPI FIFO Level Register, TWI FIFO Mode Register, TWI FIFO Level Register, TWI FIFO Status Register and TWI FIFO Interrupt Mask Register. Updated: • Figure TWI Read Operation with Multiple Data Bytes + Write Operation with Multiple Data Bytes (Sr) • TWI Clock Waveform Generator Register (CLDIV and CHDIV bit descriptions) • Modes of Operation, TWI Compatibility with I2C Standard and TWI/SMBus Characteristics (Fast Mode Plus and High Speed Modes)
Image Sensor Interface (ISI)	Power Management: added note
Timer Counter (TC)	Embedded Characteristics: corrected number of channels Block Diagram: added Note 2 TC_SRx: corrected SECE bit position
Advanced Encryption Standard (AES)	Updated: • Operating Modes • AES_MR (OPMOD field description) • AES_IVRx (IV field description)

Downloaded from Arrow.com.

.....continued	
Section	**Changes**
Secure Hash Algorithm (SHA)	SHA_MR: updated ALGO bit field description
True Random Number Generator (TRNG)	TRNG_PKBCR: updated KSLAVE field description
Electrical Characteristics	Updated: • Electrical Parameters Usage • Recommended Power Supply Sequencing introduction • ULP0 Operation and ULP1 Operation sequences of operation • Table Main RC Oscillator Characteristics • Figure Current Measurement for Applicative Use Cases
Mechanical Characteristics	Updated 228-ball TFBGA Mechanical Characteristics
Marking	Updated Line 4 description

61.2 DS60001579A - 10/2019

Changes
First issue.

Downloaded from Arrow.com.

Table of Contents

Introduction....1

Features....1

1. Configuration Summary....3

2. Block Diagram....4

3. Signal Description....5

4. Safety and Security Features....10
   - 4.1. Design for Safety and IEC60730 Class B Certification....10
   - 4.2. Design for Security....10
   - 4.3. Safety and IEC 60730 Features....10
   - 4.4. Security Features....11

5. Package and Pinout....14
   - 5.1. Packages....14
   - 5.2. Pinout....14

6. Memories....27
   - 6.1. Embedded Memories....28
   - 6.2. External Memory....28

7. System Controller....32
   - 7.1. Power-On Reset....34

8. Peripherals....35
   - 8.1. Peripheral Mapping....35
   - 8.2. Peripheral Identifiers....35
   - 8.3. FLEXCOM Features....37
   - 8.4. Peripheral Signal Multiplexing on I/O Lines....37

9. ARM926EJ-S Processor....39

10. Debug and Test....40
    - 10.1. Description....40
    - 10.2. Embedded Characteristics....40
    - 10.3. Block Diagram....41
    - 10.4. Application Examples....41
    - 10.5. Debug and Test Pin Description....43
    - 10.6. Functional Description....43

11. Boot Strategies....46
    - 11.1. Description....46
    - 11.2. Flow Diagram....46
    - 11.3. Chip Setup....47
    - 11.4. Boot Configuration....47
    - 11.5. SAM-BA Monitor....74

Downloaded from Arrow.com.

12. System Controller Write Protection (SYSCWP)..... 79
12.1. Functional Description..... 79
12.2. Register Summary..... 81

13. General Purpose Backup Registers (GPBR)..... 84
13.1. Description..... 84
13.2. Embedded Characteristics..... 84
13.3. Register Summary..... 85

14. Watchdog Timer (WDT)..... 89
14.1. Description..... 89
14.2. Embedded Characteristics..... 89
14.3. Block Diagram..... 89
14.4. Functional Description..... 89
14.5. Register Summary..... 92

15. Reset Controller (RSTC)..... 103
15.1. Description..... 103
15.2. Embedded Characteristics..... 103
15.3. Block Diagram..... 103
15.4. Functional Description..... 104
15.5. Register Summary..... 111

16. Real-time Timer (RTT)..... 117
16.1. Description..... 117
16.2. Embedded Characteristics..... 117
16.3. Block Diagram..... 117
16.4. Functional Description..... 118
16.5. Register Summary..... 120

17. Real-time Clock (RTC)..... 128
17.1. Description..... 128
17.2. Embedded Characteristics..... 128
17.3. Block Diagram..... 128
17.4. Product Dependencies..... 129
17.5. Functional Description..... 129
17.6. Register Summary..... 140

18. Shutdown Controller (SHDWC)..... 172
18.1. Description..... 172
18.2. Embedded Characteristics..... 172
18.3. Block Diagram..... 172
18.4. I/O Lines Description..... 172
18.5. Product Dependencies..... 173
18.6. Functional Description..... 173
18.7. Register Summary..... 174

19. Periodic Interval Timer (PIT)..... 179
19.1. Description..... 179

Downloaded from Arrow.com.

19.2. Embedded Characteristics 179
19.3. Block Diagram 179
19.4. Functional Description 179
19.5. Register Summary 181

20. 64-bit Periodic Interval Timer (PIT64B) 186
20.1. Description 186
20.2. Embedded Characteristics 186
20.3. Block Diagram 186
20.4. Product Dependencies 186
20.5. Functional Description 187
20.6. Register Summary 191

21. Debug Unit (DBGU) 205
21.1. Description 205
21.2. Embedded Characteristics 205
21.3. Block Diagram 206
21.4. Product Dependencies 206
21.5. Functional Description 207
21.6. Register Summary 214

22. OTP Memory Controller (OTPC) 232
22.1. Description 232
22.2. Embedded Characteristics 232
22.3. Block Diagram 233
22.4. Functional Description 233
22.5. Register Summary 242

23. Special Function Registers (SFR) 273
23.1. Description 273
23.2. Embedded Characteristics 273
23.3. Register Summary 274

24. Bus Matrix (MATRIX) 286
24.1. Description 286
24.2. Embedded Characteristics 288
24.3. Memory Mapping 288
24.4. Special Bus Granting Techniques 288
24.5. No Default Master 289
24.6. Last Access Master 289
24.7. Fixed Default Master 289
24.8. Arbitration 289
24.9. Register Write Protection 292
24.10. Register Summary 293

25. Advanced Interrupt Controller (AIC) 310
25.1. Description 310
25.2. Embedded Characteristics 310
25.3. Block Diagram 311

Downloaded from Arrow.com.

25.4. Application Block Diagram....311
25.5. AIC Detailed Block Diagram....311
25.6. I/O Line Description....312
25.7. Product Dependencies....312
25.8. Functional Description....312
25.9. Register Summary....321

26. Slow Clock Controller (SCKC)....351
26.1. Description....351
26.2. Embedded Characteristics....351
26.3. Block Diagram....351
26.4. Functional Description....351
26.5. Register Summary....353

27. Clock Generator....355
27.1. Description....355
27.2. Embedded Characteristics....355
27.3. Block Diagram....356
27.4. Slow Clock....356
27.5. Main Clock....357
27.6. PLL Controls....360

28. Power Management Controller (PMC)....364
28.1. Description....364
28.2. Embedded Characteristics....364
28.3. Block Diagram....365
28.4. Processor Clock Controller....365
28.5. USB Clock Controller....366
28.6. Free-running Processor Clock....366
28.7. Peripheral and Generic Clock Controller....366
28.8. Programmable Clock Output Controller....367
28.9. Ultra-Low Power Mode and Fast Startup....367
28.10. Main Crystal Oscillator Failure Detection....368
28.11. 32.768 kHz Crystal Oscillator Frequency Monitor....369
28.12. MCK Frequency Monitor....369
28.13. Recommended Programming Sequence....370
28.14. Clock Switching Details....371
28.15. Register Write Protection....371
28.16. Register Summary....373

29. Parallel Input/Output Controller (PIO)....415
29.1. Description....415
29.2. Embedded Characteristics....415
29.3. Block Diagram....416
29.4. Product Dependencies....416
29.5. Functional Description....417
29.6. Register Summary....426

30. External Bus Interface (EBI)....480

Downloaded from Arrow.com.

30.1. Description ... 480
30.2. Embedded Characteristics ... 480
30.3. EBI Block Diagram ... 481
30.4. I/O Lines Description ... 482
30.5. Application Examples ... 483

31. AHB Multiport DDR-SDRAM Controller (MPDDRC) ... 498
31.1. Description ... 498
31.2. Embedded Characteristics ... 498
31.3. Block Diagram ... 499
31.4. Product Dependencies, Initialization Sequence ... 499
31.5. Functional Description ... 502
31.6. Software Interface/SDRAM Organization, Address Mapping ... 516
31.7. Register Summary ... 520

32. SDRAM Controller (SDRAMC) ... 564
32.1. Description ... 564
32.2. Embedded Characteristics ... 564
32.3. Signal Description ... 564
32.4. Software Interface/SDRAM Organization, Address Mapping ... 565
32.5. Product Dependencies ... 567
32.6. Functional Description ... 568
32.7. Register Summary ... 577

33. Static Memory Controller (SMC) ... 597
33.1. Description ... 597
33.2. Embedded Characteristics ... 597
33.3. I/O Lines Description ... 597
33.4. Interrupt Source ... 598
33.5. Multiplexed Signals ... 598
33.6. Application Example ... 598
33.7. Product Dependencies ... 599
33.8. External Memory Mapping ... 599
33.9. Connection to External Devices ... 599
33.10. Standard Read and Write Protocols ... 602
33.11. Automatic Wait States ... 609
33.12. Data Float Wait States ... 613
33.13. External Wait ... 617
33.14. Slow Clock Mode ... 621
33.15. Asynchronous Page Mode ... 621
33.16. Register Write Protection ... 623
33.17. Security and Safety Analysis and Reports ... 624
33.18. Scrambling/Unscrambling Function ... 624
33.19. Clearing Scrambling Keys on a Tamper Event ... 624
33.20. Register Summary ... 625

34. Programmable Multibit Error Correction Code Controller (PMECC) ... 639
34.1. Description ... 639
34.2. Embedded Characteristics ... 639

Downloaded from Arrow.com.

34.3. Block Diagram....640
34.4. Functional Description....640
34.5. Software Implementation....646
34.6. Register Summary....650

35. Programmable Multibit ECC Error Location Controller (PMERRLOC)....666
35.1. Description....666
35.2. Embedded Characteristics....666
35.3. Block Diagram....666
35.4. Functional Description....666
35.5. Register Summary....668

36. DMA Controller (XDMAC)....682
36.1. Description....682
36.2. Embedded Characteristics....682
36.3. Block Diagram....683
36.4. DMA Controller Peripheral Connections....683
36.5. Functional Description....685
36.6. Linked List Descriptor Operation....689
36.7. XDMAC Maintenance Software Operations....692
36.8. XDMAC Software Requirements....692
36.9. Register Summary....694

37. LCD Controller (LCDC)....749
37.1. Description....749
37.2. Embedded Characteristics....749
37.3. Block Diagram....750
37.4. I/O Lines Description....750
37.5. Product Dependencies....751
37.6. Functional Description....751
37.7. Register Summary....780

38. 2D Graphics Engine (GFX2D)....952
38.1. Description....952
38.2. Embedded Characteristics....952
38.3. Block Diagram....952
38.4. Functional Description....952
38.5. Register Summary....993

39. Ethernet MAC 10/100 (EMAC)....1016
39.1. Description....1016
39.2. Embedded Characteristics....1016
39.3. Block Diagram....1017
39.4. Functional Description....1017
39.5. Programming Interface....1028
39.6. Register Summary....1030

40. USB High Speed Device Port (UDPHS)....1088
40.1. Description....1088

Downloaded from Arrow.com.

40.2. Embedded Characteristics 1088
40.3. Block Diagram 1089
40.4. Typical Connection 1090
40.5. Product Dependencies 1090
40.6. Functional Description 1090
40.7. Register Summary 1111

41. USB Host High Speed Port (UHPHS) 1171
41.1. Description 1171
41.2. Embedded Characteristics 1171
41.3. Block Diagram 1171
41.4. Typical Connection 1172
41.5. Product Dependencies 1172
41.6. Functional Description 1174
41.7. Register Summary 1176

42. Audio Class D Amplifier (CLASSD) 1201
42.1. Description 1201
42.2. Embedded Characteristics 1201
42.3. Block Diagram 1202
42.4. Pin Name List 1202
42.5. Product Dependencies 1203
42.6. Functional Description 1203
42.7. Register Summary 1216

43. Inter-IC Sound Multi-Channel Controller (I2SMCC) 1229
43.1. Description 1229
43.2. Embedded Characteristics 1229
43.3. Block Diagram 1230
43.4. I/O Lines Description 1230
43.5. Product Dependencies 1230
43.6. Functional Description 1231
43.7. Register Summary 1241

44. Synchronous Serial Controller (SSC) 1263
44.1. Description 1263
44.2. Embedded Characteristics 1263
44.3. Block Diagram 1264
44.4. Application Block Diagram 1264
44.5. SSC Application Examples 1264
44.6. Pin Name List 1266
44.7. Product Dependencies 1266
44.8. Functional Description 1267
44.9. Register Summary 1277

45. Flexible Serial Communication Controller (FLEXCOM) 1305
45.1. Description 1305
45.2. Embedded Characteristics 1306
45.3. Block Diagram 1308

Downloaded from Arrow.com.

45.4. I/O Lines Description....1309
45.5. Product Dependencies....1309
45.6. Register Accesses....1309
45.7. USART Functional Description....1309
45.8. SPI Functional Description....1360
45.9. TWI Functional Description....1379
45.10. Register Summary....1421

46. Quad Serial Peripheral Interface (QSPI)....1569
46.1. Description....1569
46.2. Embedded Characteristics....1569
46.3. Block Diagram....1570
46.4. Signal Description....1570
46.5. Product Dependencies....1570
46.6. Functional Description....1571
46.7. Register Summary....1588

47. Secure Digital MultiMedia Card Controller (SDMMC)....1610
47.1. Description....1610
47.2. Embedded Characteristics....1610
47.3. Reference Documents....1610
47.4. Block Diagram....1611
47.5. Application Block Diagram....1612
47.6. Pin Name List....1612
47.7. Product Dependencies....1612
47.8. SD/SDIO Operating Mode....1613
47.9. e.MMC Operating Mode....1613
47.10. Register Summary....1616

48. Image Sensor Interface (ISI)....1693
48.1. Description....1693
48.2. Embedded Characteristics....1694
48.3. Block Diagram....1694
48.4. Product Dependencies....1694
48.5. Functional Description....1695
48.6. Register Summary....1703

49. Controller Area Network (CAN)....1737
49.1. Description....1737
49.2. Embedded Characteristics....1737
49.3. Block Diagram....1738
49.4. Application Block Diagram....1738
49.5. I/O Lines Description....1738
49.6. Product Dependencies....1739
49.7. CAN Controller Features....1739
49.8. Functional Description....1748
49.9. Register Summary....1759

50. Timer Counter (TC)....1797

Downloaded from Arrow.com.

50.1. Description ... 1797
50.2. Embedded Characteristics ... 1797
50.3. Block Diagram ... 1798
50.4. Pin List ... 1799
50.5. Product Dependencies ... 1799
50.6. Functional Description ... 1799
50.7. Register Summary ... 1822

51. Pulse Width Modulation Controller (PWM) ... 1862
51.1. Description ... 1862
51.2. Embedded Characteristics ... 1862
51.3. Block Diagram ... 1863
51.4. I/O Lines Description ... 1863
51.5. Product Dependencies ... 1863
51.6. Functional Description ... 1864
51.7. Register Summary ... 1871

52. Advanced Encryption Standard (AES) ... 1889
52.1. Description ... 1889
52.2. Embedded Characteristics ... 1889
52.3. Product Dependencies ... 1890
52.4. Functional Description ... 1890
52.5. Register Summary ... 1911

53. Secure Hash Algorithm (SHA) ... 1943
53.1. Description ... 1943
53.2. Embedded Characteristics ... 1943
53.3. Product Dependencies ... 1943
53.4. Functional Description ... 1943
53.5. Register Summary ... 1952

54. Triple Data Encryption Standard (TDES) ... 1972
54.1. Description ... 1972
54.2. Embedded Characteristics ... 1972
54.3. Product Dependencies ... 1973
54.4. Functional Description ... 1973
54.5. Register Summary ... 1981

55. True Random Number Generator (TRNG) ... 2001
55.1. Description ... 2001
55.2. Embedded Characteristics ... 2001
55.3. Block Diagram ... 2001
55.4. Product Dependencies ... 2001
55.5. Functional Description ... 2002
55.6. Register Summary ... 2005

56. Analog-to-Digital Controller (ADC) ... 2018
56.1. Description ... 2018
56.2. Embedded Characteristics ... 2018

Downloaded from Arrow.com.

56.3. Block Diagram....2019
56.4. Signal Description....2019
56.5. Product Dependencies....2020
56.6. Functional Description....2020
56.7. Register Summary....2046

57. Electrical Characteristics....2084
57.1. Electrical Parameters Usage....2084
57.2. Absolute Maximum Ratings....2085
57.3. Recommended Operating Conditions....2086
57.4. Recommended Power Supply Sequencing....2087
57.5. I/O Characteristics....2091
57.6. Digital Peripheral Characteristics....2096
57.7. Analog Peripheral Characteristics....2116
57.8. Power Consumption in Active Mode....2127
57.9. Operation and Power Consumption in Low-Power Modes....2129

58. Mechanical Characteristics....2137
58.1. 228-ball TFBGA Mechanical Characteristics....2137

59. Marking....2141

60. Ordering Information....2142

61. Revision History....2143
61.1. DS60001579B - 02/2020....2143
61.2. DS60001579A - 10/2019....2144

The Microchip Website....2155

Product Change Notification Service....2155

Customer Support....2155

Product Identification System....2156

Microchip Devices Code Protection Feature....2156

Legal Notice....2157

Trademarks....2157

Quality Management System....2157

Worldwide Sales and Service....2158

Downloaded from Arrow.com.

The Microchip Website

Microchip provides online support via our website at http://www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to http://www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

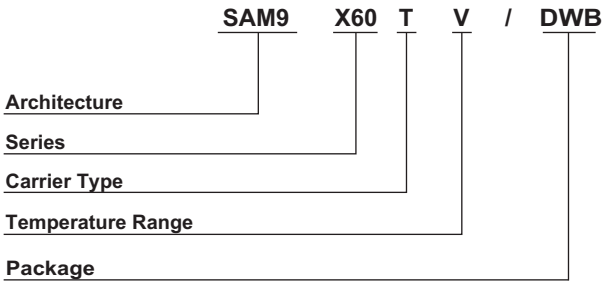
Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: http://www.microchip.com/support

Downloaded from Arrow.com.

Product Identification System

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



Architecture:	SAM9	= ARM926EJ-S Arm Thumb microprocessor
Series:	X60	
Carrier Type:	Blank	= Standard packaging (tray)
	T	= Tape and Reel
Temperature Range:	V	= -40°C to +105°C (Extended Industrial)
Package:	DWB	= TFBGA228

Example:

- SAM9X60T-V/DWB = ARM926EJ-S Arm Thumb microprocessor, Tape and Reel, Extended Industrial temperature, 228-ball TFBGA package

Note:

1. Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.
2. Small form-factor packaging options may be available. Please check http://www.microchip.com/packaging for small-form factor package availability, or contact your local Sales Office.

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Downloaded from Arrow.com.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TempTrackr, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2020, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-5568-4

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit http://www.microchip.com/quality.

Downloaded from Arrow.com.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: http://www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4450-2828 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820

Downloaded from Arrow.com.