i.MX21
Applications Processor

Reference Manual

Document Number: MC9328MX21RM
Rev. 3
04/2007

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

***For Literature Requests Only*:**
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-521-6274 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Contents – *Concise*

Part 1: Device Introduction

Part 2: Core Technology

Part 3: System Control

Part 4: Memory Interfaces

Part 5: InterChip Connectivity

Part 6: Peripherals

Part 7: Connectivity and Expansion

Contents – *Full*

Part 1: Device Introduction

Chapter 1
Introduction

Chapter 2
Signal Descriptions and Pin Assignments

Chapter 3
Memory Map

Part 2: Core Technology

Chapter 4
ARM9 Platform

Chapter 5
ARM926EJ-S Interrupt Controller (AITC)

Chapter 6
Phase-Locked Loop (PLL), Clock and Reset Controller

Chapter 7
AHB-Lite IP Interface (AIPI) Module

Part 3: System Control

Chapter 8
System Control

Chapter 9
Internal ROM, System Boot Manager

Chapter 10
Multi-layer AHB Crossbar Switch (MAX)

Chapter 11
JTAG Controller

Chapter 12
Watchdog Timer Module (WDOG)

Chapter 13
Real-Time Clock (RTC)

Chapter 14
General-Purpose Timers (GPT)

Chapter 15
General-Purpose I/O (GPIO)

Chapter 16
Pulse-Width Modulator (PWM)

Part 4: Memory Interfaces

Chapter 17
SDRAM Memory Controller

Chapter 18
Direct Memory Access Controller (DMAC)

Chapter 19
NAND Flash Memory Controller

Chapter 20
External Interface Module (EIM)

Chapter 21
Bus Master Interface (BMI)

Part 5: InterChip Connectivity

Chapter 22
I²C Module

Chapter 23
Configurable Serial Peripheral Interface (CSPI)

Chapter 24
Synchronous Serial Interface (SSI)

Part 6: Peripherals

Chapter 25
CMOS Sensor Interface (CSI)

Chapter 26
Liquid Crystal Display Controller (LCDC)

Chapter 27
Smart Liquid Crystal Display Controller (SLCDC)

Chapter 28
enhanced Multimedia Accelerator (eMMA)

Chapter 29
MultimediaCard/Secure Digital Host Controller (MMC/SDHC)

Chapter 30
Digital Audio Mux (AUDMUX)

Part 7: Connectivity and Expansion

Chapter 31
Universal Asynchronous Receiver/Transmitters (UART) Modules

Chapter 32
Universal Serial Bus On-The-Go (USB OTG)

Chapter 33
PCMCIA/CF Interface

Chapter 34
Keypad Port (KPP)

Chapter 35
Fast InfraRed Interface (FIRI) Module

Chapter 36
1-Wire Interface (1-Wire®)

About This Book

This reference manual describes the features and operation of the i.MX21 microprocessor, the seventh generation of the DragonBall family of products. It provides the details of how to initialize, configure, and program the i.MX21. The manual presumes basic knowledge of ARM926EJ-S™ architecture.

Audience

The i.MX21 reference manual is intended to provide a design engineer with the necessary data to successfully integrate the i.MX21 into a wide variety of applications. It is assumed that the reader has a good working knowledge of the ARM926EJ-S processor. For programming information about the ARM926EJ-S processor, see the documents listed in the Suggested Reading section of this preface.

Organization

The i.MX21 reference manual is organized into 8 parts, consisting of 37 chapters that cover the operation and programming of the i.MX21 device. Summaries of the chapters follow.

Part 1 Device Introduction

Chapter 1, “Introduction,”
: This chapter contains an high-level summary of the i.MX21 device and includes device feature list, overview of system modules, and system block diagrams.

Chapter 2, “Signal Descriptions and Pin Assignments,”
: This chapter contains listings of the i.MX21 input and output signals, organized into functional groups.

Chapter 3, “Memory Map,”
: The Memory Map chapter provides a complete listing of all of the register, internal and external module memory assignments.

Part 2 Core Technology

Chapter 4, “ARM9 Platform,”
: A brief summary of the ARM9™ platform and its operational features is covered in this chapter.

Chapter 5, “ARM926EJ-S Interrupt Controller (AITC),”
: This chapter describes the operation of the 32-bit peripheral which collects interrupt requests from up to 64 sources and provides an interface to the ARM926EJ-S core.

Chapter 6, “Phase-Locked Loop (PLL), Clock and Reset Controller,”
: This chapter provides detailed information about the operation and programming of the clock generation module as well as the recommended circuit schematics for external clock circuits. It also describes and provides programming information about the operation of the power control module and the system power states.

Chapter 7, “AHB-Lite IP Interface (AIPI) Module,”
: This chapter provides an overview of the two AHB-Lite to IP bus interface (AIPI) modules. The AIPIAHB-Lite IP interface module (AIPI Ver1.0) acts as an interface between the ARM® advanced high-performance bus “lite” (AHB-Lite) and lower bandwidth peripherals.

Part 3 System Control

Chapter 8, “System Control,”

This chapter describes the operation of and programming models for the system multiplex control, peripheral control, ID register, and I/O drive control registers.

Chapter 9, “Internal ROM, System Boot Manager,”

This chapter describes the system boot up sequence of the i.MX21 microprocessor. The operation of bootstrap models is described in detail. This chapter also provides the programming information necessary to allow a system to initialize a target system and download a program or data to the target system’s RAM using the UART controller.

Chapter 10, “Multi-layer AHB Crossbar Switch (MAX),”

This chapter provides an overview of the generic MAX (Multi-Layer AHB Crossbar Switch) which allows concurrent support of up to 8 simultaneous connections between 6 master ports and 4 slave ports.

Chapter 11, “JTAG Controller,”

The operation and configuration of the JTAG controller module is described in this chapter. The JTAG controller supports debug access to ARM926 core, i.MX21 BIST test (excluding ARM926 platform BIST, Boot ROM and VectorRAM bist) and tristate enable of the I/O pads.

Chapter 12, “Watchdog Timer Module (WDOG),”

The operation of the watchdog timer module is described in this chapter. It includes information of how the watchdog timer protects against system failures by providing a method of escaping from unexpected events or programming errors.

Chapter 13, “Real-Time Clock (RTC),”

This chapter describes the operation of the real-time clock module, which is composed of a prescaler, time-of-day (TOD) clock, TOD alarm, programmable real-time interrupt, watchdog timer, and minute stopwatch as well as control registers and bus interface hardware.

Chapter 14, “General-Purpose Timers (GPT),”

This chapter describes the two 16-bit timers that can be used as both watchdogs and alarms.

Chapter 15, “General-Purpose I/O (GPIO),” This chapter discusses all GPIO lines found in the i.MX21. Because each pin is individually configurable, a detailed description of the operation is provided.

Chapter 16, “Pulse-Width Modulator (PWM),”

This chapter describes the operation and configuration of the pulse-width modulator. Programming information is also provided.

Part 4 Memory Interfaces

Chapter 17, “SDRAM Memory Controller,”

The operation and programming of the SDRAM controller is described in this chapter. This module provides a glueless interface to 8-bit or 16-bit DRAM supporting Fast Page Mode, and synchronous DRAM.

Chapter 18, “Direct Memory Access Controller (DMAC),”

This chapter describes the operation of the direct memory access controller contained in the i.MX21. The DMA controller provides memory channels and I/O channels to support a wide variety of DMA operations.

Chapter 19, “NAND Flash Memory Controller,”

This chapter describes the NAND Flash Controller that allows standard NAND Flash chips to interface with AMBA™ AHB of the i.MX21.

Chapter 20, “External Interface Module (EIM),”

This chapter describes the external interface module and defines how the module handles the interface to devices external to the i.MX21, including generation of chip-selects for external peripherals and memory.

Chapter 21, “Bus Master Interface (BMI),”

This chapter describes the unique interface requirement for Bus master interface module. This BMI module enables high speed connection between i.MX21 and the alternate bus master devices in the system.

Part 5 InterChip Connectivity

Chapter 22, “I^2C Module,”

This chapter describes the I^2C module of the i.MX21 including I^2C protocol, clock synchronization, and the registers in the I^2C programming mode.

Chapter 23, “Configurable Serial Peripheral Interface (CSPI),”

The programming and operation of the two identical serial peripheral interface modules are described in this chapter.

Chapter 24, “Synchronous Serial Interface (SSI),”

This chapter presents the Synchronous Serial Interface and discusses the architecture, programming model, operating modes, and initialization of the SSI.

Part 6 Peripherals

Chapter 25, “CMOS Sensor Interface (CSI),”

The CSI module is a logic interface that enables the i.MX21 to connect directly to external CMOS image sensors. This chapter describes the CSI module, and discusses the architecture, the programming model, and the software initialization sequence.

Chapter 26, “Liquid Crystal Display Controller (LCDC),”

This chapter describes the operation and programming of the liquid crystal display controller, which provides display data for external LCD drivers or for an LCD panel.

Chapter 27, “Smart Liquid Crystal Display Controller (SLCDC),”

This chapter describes the operation and programming of the Smart Liquid Crystal Display controller module which transfers data from the display memory buffer to the external display device transparently with minimal software intervention.

Chapter 28, “enhanced Multimedia Accelerator (eMMA),”

This chapter describes the operation and configuration of the eMMA module which performs computing extensive video processing functions. It consists of four blocks: video encoding, video decoding, post-processing, and pre-processing.

Chapter 29, “MultimediaCard/Secure Digital Host Controller (MMC/SDHC),”

This chapter describes the Multimedia Card (MMC) host controller which controls flash-based mass storage products. This chapter also describes the secure digital feature of the MMC, its operation and programming information.

Chapter 30, “Digital Audio Mux (AUDMUX),”

This chapter describes the operation and configuration of the Digital Audio Mux (AUDMUX) which provides a programmable interconnect fabric for voice, audio and

synchronous data routing between i.MX21 SSI modules and external SSI, audio and voice codecs.

Part 7 Connectivity and Expansion

Chapter 31, “Universal Asynchronous Receiver/Transmitters (UART) Modules,”
This chapter describes the capabilities and operation of the i.MX21’s UARTs. It also discusses how to configure and program the UART modules.

Chapter 32, “Universal Serial Bus On-The-Go (USB OTG),”
This chapter describes the operation and configuration of the USB full speed OTG module that provides USB On-The-Go (OTG) compliant functionality to i.MX21.

Chapter 33, “PCMCIA/CF Interface,”
This chapter describes the configuration and operation of the i.MX21 PCMCIA host adapter module that provides all control logic for PCMCIA socket interface.

Chapter 34, “Keypad Port (KPP),”
This chapter describes the configuration and operation of the keypad port that supports up to an 8 ¥ 8 keypad.

Chapter 35, “Fast InfraRed Interface (FIRI) Module,”
This chapter describes Fast InfraRed Interface module (FIRI), which is integrated in the i.MX21. It is capable to establish standard speed half duplex links via LED and IR detector. It supports 0.576 Mbit/sec, 1.152 Mbit/sec MIR physical layer protocol and 4Mbit/sec FIR physical layer protocol defined by IrDA, version 1.4. In addition, i.MX21 can support SIR protocol by the UART module.

Chapter 36, “1-Wire Interface (1-Wire®),”
This chapter describes the The 1-Wire® interface that provides the communication line to a 1Kbit Add-Only Memory (DS2502). The 1-Wire module is a peripheral device to the ARM926 core and communicates with it via the IP interface

Document Revision History

The following table provides revision history for this release. This history includes technical content revisions only and not stylistic or grammatical changes.

Table 0-1. i.MX21 Reference Manual Revision History for Rev. 3

Revision Location	Revision
Throughout Book	Several minor technical corrections throughout Book, such as naming signals consistently.
Throughout Book	Incorporated silicon M55B content and values from MC9328MX21RMAD1 *i.MX21 Reference Manual Addendum* into this *i.MX21 Reference Manual*.
Throughout Book	Removed Security Feature, chapters, references.
Several Locations	Noted AC97 fixed mode only supports 48 kHz sampling rate.
Chapter 2, “Signal Descriptions and Pin Assignments.”	• Changes to External DMA and CSPI portions of Signal Description table. • Changes to NVDD3 Multiplexing scheme portion of Signal Multiplexing table.
Old Chapter 6, “ROM Patch Module”	Removed the ARM926EJ-S ROM Patch Module chapter and all references.

Table 0-1. i.MX21 Reference Manual Revision History for Rev. 3

Revision Location	Revision
Chapter 6, “Phase-Locked Loop (PLL), Clock and Reset Controller.”	Changes to recommended settings for frequency stability: 32.768 kHz, 240 target frequency, PD from 1 to 0, and 26 MHz, 240 target frequency, all settings. • Clarification of ipg_clk_xxx as PERCLK. • Added statement regarding PERCLK to Table 6-1 PLL Clock Controller Signal Descriptions for PERCLK1–4, and LDCCLK. • Peripheral Clock Divider Register 0 (PCDR0) Corrected bit 26 number and bit 21 type from r to rw. • Peripheral Clock Divider Register 1 (PCDR1) • Changed reset values of bits 11 and 1 from 1 to 0.
Chapter 8, “System Control.”	Replaced Silicon ID Register and register description table with silicon M55B values.
Chapter 9, “Internal ROM, System Boot Manager.”	• Removed content from the iROM, System Boot Manager and Bootstrap chapters, and renamed from System Boot to Internal ROM, System Boot Manager and removed previous chapters 10 and 11. Reorganized new chapter 10. • Extensive changes to all figures. • Removal of HAB section and all references to HAB
Chapter 17, “SDRAM Memory Controller,” Section 17.2.1 on page -8	• Changed SDRAM Control Register description table to accurately reflect register display from Reserved[3] to SRC [3:0]. • SDRAM Mode Register Description Switched Burst Type (BT) bit A3 settings.
Chapter 19, “NAND Flash Memory Controller.”	• Replaced first paragraph in Functional Overview to reflex limits of the Write Protection Unlock registers. • Removed reference that system boot from and 8-bit device with 2 Kbyte page size is not supported. M55B silicon supports this system boot operation.
Chapter 23, “Configurable Serial Peripheral Interface (CSPI).”	• CSPI Control Register Description Table Changed Bit Count description in reference to the ss rising edge.
Chapter 26, “Liquid Crystal Display Controller (LCDC).”	• Supported Panel Characteristics table Changed CSTN Panel Interface value from 12 to 8. • Corrected Register description name: Was: LCD Graphic Window DMA Control Register To: LCD DMA Control Register
Chapter 27, “Smart Liquid Crystal Display Controller (SLCDC).”	• SLCDC Serial Interface Timing Diagram and Table Changed symbol identification to “T” nomenclature. • SLCDC Parallel Interface Timing Diagram and Table Changed symbol identification to “T” nomenclature.
Chapter 31, “Universal Asynchronous Receiver/Transmitters (UART) Modules.”	• General UART Definitions, Figure 31-1 Changed CTS/RTS directions between the Computer UART and the i.MX21 UART.
Chapter 33, “PCMCIA/CF Interface.”	• Changed features list to include memory space up to 32 kbytes in size. • PCMCIA Base Registers 0–4 (PBR0–PBR4) Changed PBA field from 10–0 to 14–4. • PCMCIA Offset Registers 0–4 (POR0–POR4) Changed POFA field from 10–0 to 14–4. • Replaced BSIZE value table • Replaced BSIZE mask table

Suggested Reading

The following documents are required for a complete description of the i.MX21 and are necessary to design properly with the device. Especially for those not familiar with the ARM926EJ-S processor or previous DragonBall products, the following documents will be helpful when used in conjunction with this manual.

AMBA AHB specifications, (ARM Ltd.)

ARM926EJ-S Platform specifications (also named *ARM926p Platform*)

Hip7a KiloBit Single Port HP SRAM Compiler, MEMCTC (May 8, 2002)

Hip7A SAMI ROM Compiler, MEMCTC (November 16, 2001)

Hip7A KiloBit HD VIA ROM Compiler, MEMCTC (June 28, 2002)

ARM926EJ-S Platform Test Guide (ARM Ltd.)

ARM Architecture Reference Manual (ARM Ltd., order number ARM DDI 0100)

ARM9DT1 Data Sheet Manual (ARM Ltd., order number ARM DDI 0029)

ARM Technical Reference Manual (ARM Ltd., order number ARM DDI 0151C)

MC9328MX1 i.MX Integrated Portable System Processor Reference Manual (order number MC9328MX1RM)

MC9328MXL i.MX Integrated Portable System Processor Reference Manual (order number MC9328MXLRM)

MC9328MX21 Application Processor Data Sheet—266 MHz (order number MC9328MX21)

MC94MX21 Application Processor Data Sheet—333–350 MHz (order number MC94MX21)

The manuals may be found at the ARM Ltd. World Wide Web site at http://www.arm.com and Freescale Semiconductors World Wide Web site at http://www.freescale.com/imx. These documents may be downloaded directly from the World Wide Web site, or printed versions may be ordered. The World Wide Web site may also have useful application notes.

Conventions

This reference manual uses the following conventions:

- $\overline{\text{OVERBAR}}$ is used to indicate a signal that is active when pulled low: for example, $\overline{\text{RESET}}$.
- *Logic level one* is a voltage that corresponds to Boolean true (1) state.
- *Logic level zero* is a voltage that corresponds to Boolean false (0) state.
- To *set* a bit or bits means to establish logic level one.
- To *clear* a bit or bits means to establish logic level zero.
- A *signal* is an electronic construct whose state conveys or changes in state convey information.
- A *pin* is an external physical connection. The same pin can be used to connect a number of signals.
- *Asserted* means that a discrete signal is in active logic state.
  - *Active low* signals change from logic level one to logic level zero.
  - *Active high* signals change from logic level zero to logic level one.

- *Negated* means that an asserted discrete signal changes logic state.
  - *Active low* signals change from logic level zero to logic level one.
  - *Active high* signals change from logic level one to logic level zero.
- LSB means least significant bit or bits, and MSB means most significant bit or bits. References to low and high bytes or words are spelled out.
- Numbers preceded by a percent sign (%) are binary. Numbers preceded by a 0x are hexadecimal.

Definitions, Acronyms, and Abbreviations

The following list defines acronyms and abbreviations used in this document.

ADC	analog-to-digital converter
AFE	analog front end
API	application programming interface
BCD	binary coded decimal
BER	bit error ratio
CGM	clock generation module
CMOS	complimentary metal-oxide semiconductor
CRC	cyclic redundancy check
CSIC	complex instruction set computer
DAC	digital-to-analog converter
DDR RAM	double data rate RAM
DMA	direct memory access
DRAM	dynamic random access memory
DSP	digital signal processor
FEC	forward error correction
FIFO	first in first out
FIRI	fast IR interface
GPIO	general purpose input/output
I/O	Input/Output
ICE	in-circuit emulation
IrDa	infrared data association
JTAG	joint test action group
MAP	mold array process
MAPBGA	mold array process ball grid array
MIPS	million instructions per second
MMC	multimedia card
PLL	phase locked loop
PWM	pulse-width modulator

RTC	real-time clock
SD	secure digital
SDRAM	synchronous dynamic random access memory
SPI	serial peripheral interface
SRAM	static random access memory
TQFP	thin quad flat pack
UART	universal asynchronous receiver/transmitter
USB	universal serial bus
USB OTG	USB On-The-Go
XTAL	crystal
BE / LE	big endian / little endian
BIST	built in self-test
CCM	clock control module, also called “clkctl” module
LV	low voltage
LWB	late-write buffer
MCTL	memory controller
RAM	random access memory
ROM	read only memory
R-AHB bus	reduced advanced high-performance bus (AHB), related to ARM bus architecture
SRAM	static RAM
ARM	Advanced RISC Machines processor architecture
API	Application Programming Interface
Fabrication Path	Path within ROM Bootstrap for fabrication test execution
Flash Path	Path within ROM Bootstrap leading towards executing a Flash application.
GPCR	Global Peripheral Control Registry of the i.MX21.
HW	Hardware
iRAM	Processor-internal RAM
iROM	Processor-internal ROM
NANDFC	NAND Flash Controller
NAND Flash	A Flash ROM Technology
ROM Bootstrap	Internal boot code encompassing main boot flow as well as exception vectors, USB/UART Bootloader blocks.
RAM Path	Path within ROM Bootstrap leading towards downloading and executing a RAM application
SIDR	Silicon ID Register of the i.MX21
Sync Flash	A Flash ROM Technology
TBD	To Be Determined

UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
V-Sync Flash	A Flash ROM Technology
Word	32 bits
TYPE	Identifier that distinguishes a production or engineering device.
UID	Unique ID; a field in the processor and CSF identifying a device or group of devices
WTLS	Wireless Transport layer Security, a part of the Wireless Application Protocol

Part 1
Device Introduction

Chapter 1
Introduction

The DragonBall family of microprocessors has demonstrated leadership in the portable handheld market. Following on the success of the DragonBall MX (Media Extensions) series, the MC9328MX21 (i.MX21) provides a leap in performance with an ARM926EJ-S™ microprocessor core that provides accelerated Java support in addition to highly integrated system functions. DragonBall products specifically address the needs of the smartphone and portable product markets with their intelligent integrated peripherals, advanced processor core, and power management capabilities.

The i.MX21 processor features the advanced and power-efficient ARM926EJ-S core operating at speeds up to 266 MHz (for higher frequency devices, refer to that device's data sheet. See section "Suggested Reading" in the Chapter , "About This Book."). On-chip modules such as an MPEG4 codec, LCD controller, USB OTG, CMOS sensor interface, and an AC97 host controller offer designers a rich suite of peripherals that can enhance any product seeking to provide a rich multimedia experience. For cost sensitive applications, the NAND Flash controller allows the use of low cost Nand Flash devices to be used as primary or secondary non-volatile storage. The on-chip ECC and parity checking circuitry of the Nand Flash controller frees the CPU for other tasks. WLAN, Bluetooth and expansion options are provided through PCMCIA/CF, USB, and MMC/SD host controllers.

The i.MX21 processor is packaged in a 289-pin plastic ball grid array (PBGA).

A summary of the main features of the i.MX21 processor includes:

- Seventh generation of industry-leading DragonBall family for the personal, portable product market
- High level of on-chip integration
- Very low-power system design without compromised performance
- Optimized for multimedia applications
- Optimized for Bluetooth applications with high-speed interfaces to external Bluetooth solutions
- Dedicated graphics accelerator port
- Supports a wide variety of applications including the most popular smartphones, PDAs, and next-generation wireless communicators

1.1 i.MX21 Block Diagram

Figure 1-1 is a simplified functional block diagram of the i.MX21 processor.

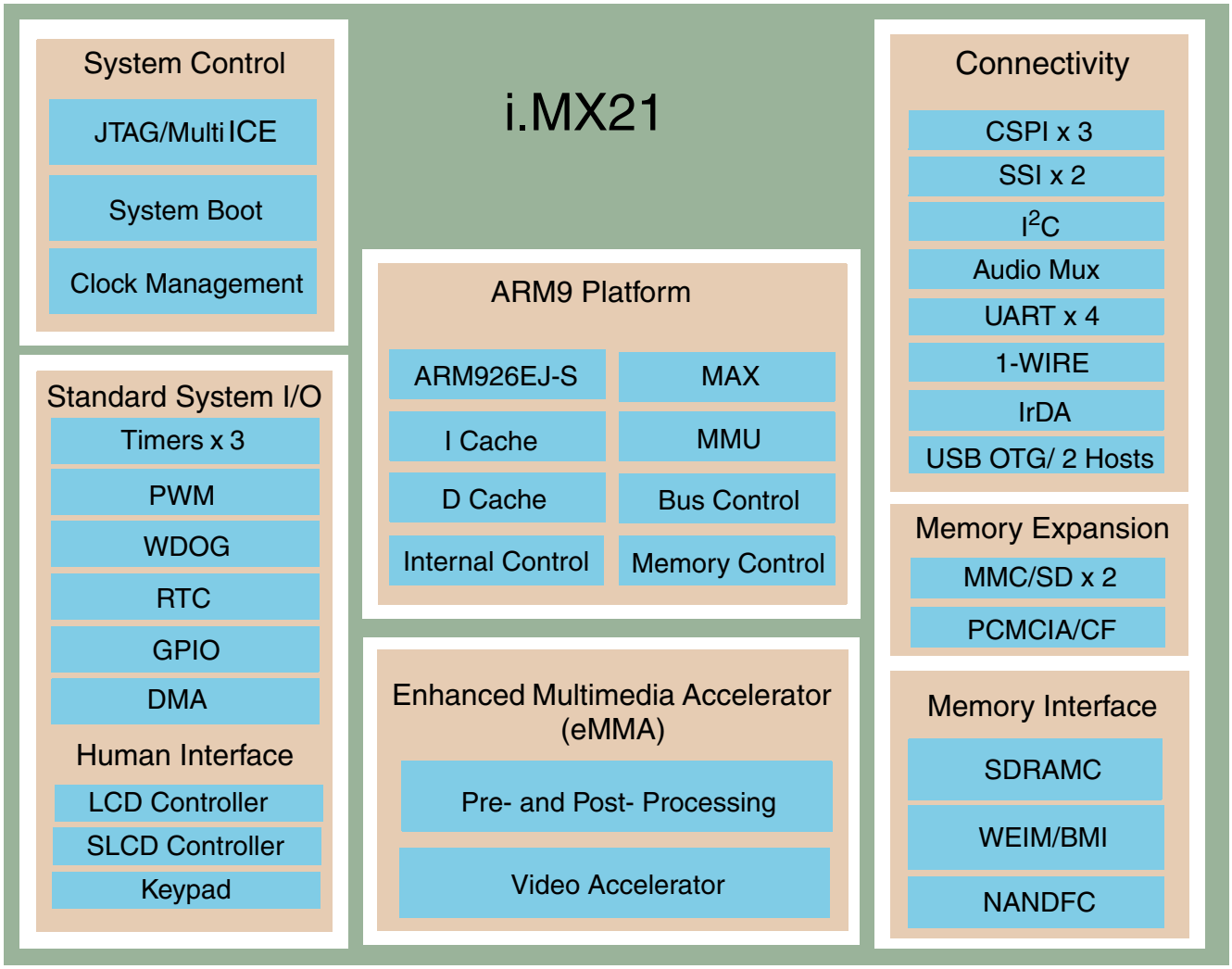


Figure 1-1. MC9328MX21 Functional Block Diagram

1.2 i.MX21 Features

The MC9328MX21 boasts a robust array of features that can support a wide variety of applications. This chapter provides a brief description of the i.MX21 processor's features.

1.2.1 ARM926EJ-S Core Complex

The ARM926EJ-S Core Complex (also known as the ARM926 platform) consists of the ARM926EJ-S processor, a 6 × 4 Multi-Layer AHB crossbar switch, and a primary AHB complex.

- ARM926EJ-S microprocessor core
  - 16K instruction cache and 16K data cache
  - High-performance ARM® 32-bit RISC engine
  - Thumb® 16-bit compressed instruction set for a leading level of code density
  - Efficient execution of Java byte codes

  - EmbeddedICE™ JTAG software debug
  - 100 percent user code binary compatibility with ARM7TDMI®
  - Advanced Microcontroller Bus Architecture (AMBA™) system-on-chip multi-master bus interface
  - Support for mixed loads of real-time and user applications via cache locking facilities
  - Virtual Memory Management Unit (VMMU)
- ARM Interrupt Controller (AITC)
  - The AITC is connected to the primary AHB as a slave device and provides support for up to 64 interrupt sources. It generates normal and fast interrupts to the processor core. The AITC supports a hardware assisted vectoring mode for automatic vectoring to reduce interrupt latency.
- Digital Phase-Locked Loops (DPLLs) and Power Control Module
  - Digital phase-locked loops (DPLLs) and clock controller for all internal clock generation
  - MCUPLL generates system and CPU clocks from a 26MHz crystal
  - USBPLL generates 48 MHz clock for the USB OTG from either a 26 MHz crystal or 32kHz
  - Support for three power modes for different power consumption needs: run, doze, and stop.
- AHB to IP bus interfaces (AIPIs)
  - Provide a communication interface between the high-speed AHB to a lower-speed IP bus for slave peripherals
- The Multi-Layer 6 × 4 AHB Crossbar Switch
  - The crossbar switch allows for concurrent transactions to proceed from any input port (bus master) to any output port (bus slave). That is, it is possible for all four output ports to be active at the same time as a result of four independent input or output requests.
- CPU and System speed
  - ARM926EJ-S core: up to 266 MHz
  - System Clock: up to 133 MHz
  - External memory interface: same clock source as system, up to 133 MHz at 1.8V supply
  - System clock is derived from the CPU clock through an integer divider

1.2.2 System Control and Timers

The i.MX21 processor contains various timers and system control features to optimize the control of both the internal modules and external devices.

1.2.2.1 Watchdog Timer

The Watchdog Timer module (WDOG Timer) provides the following:

- Programmable time out of 0.5 s to 64 s
- Resolution of 0.5 s

1.2.2.2 Real-Time Clock/Sampling Timer

The Real-Time Clock (RTC) module maintains the system clock, provides stopwatch, alarm, and interrupt functions, and supports the following features:

- 32.768 kHz and 32 kHz input operation
- Full clock features: seconds, minutes, hours, days
- Capable of counting up to 512 days
- Minute countdown timer with interrupt
- Programmable daily alarm with interrupt
- Sampling timer with interrupt
- Once-per-second, once-per-minute, once-per-hour, and once-per-day interrupts
- Interrupt generation for digitizer sampling or keyboard debouncing

1.2.2.3 Three General-Purpose 32-Bit Counters/Timers

The General-Purpose Timer (GPT) module contains three identical general-purpose 32-bit timers with programmable prescalers and compare and capture registers with the following features:

- Automatic interrupt generation
- Programmable timer input/output pins
- Input capture capability with programmable trigger edge
- Output compare with programmable mode

1.2.2.4 Pulse-Width Modulator Module

The following features characterize the Pulse-Width Modulator (PWM) module:

- 4 × 16 FIFO to minimize interrupt overhead
- 16-bit resolution
- Sound and melody generation

1.2.2.5 General-Purpose I/O Ports

The GPIO module provides six general purpose I/O ports. Each single GPIO port is a 32-bit port that may be multiplexed with one or more dedicated functions. The GPIO features are:

- Supports level or edge trigger interrupt and is system wake-up capable
- Most I/O signals are multiplexed with dedicated functions for pin efficiency

1.2.2.6 Endianness

The i.MX21 processor system supports only little endian.

1.2.3 Memory Interface

The memory interfaces of the i.MX21 processor consist of the SDRAM controller, the Direct Memory Access controller, the NAND Flash controller and the External Interface module. The individual features of these controllers are provided in this section.

1.2.3.1 SDRAM Controller

The SDRAM controller (SDRAMC) consists of 7 major blocks, including the SDRAM command controller, page and bank address comparators, row/column address multiplexer, data aligner/multiplexer, configuration registers, refresh request counter, and the powerdown timer. The features offered by the SDRAMC are as follows:

- Support for four banks of single data rate 64 Mbit, 128 Mbit, and 256 Mbit SDRAM
  - Two independent chip-selects with up to 64 Mbyte per chip-select
  - Up to four banks active simultaneously for each chip-select
  - JEDEC standard pinout and operation
  - Boot capability from CSD1
- PC133-compliant interface
  - 133 MHz system clock achievable with “-8” option PC133-compliant memories
  - Single and fixed-length (4-word) burst access
  - Access time of 8-1-1-1 at 133 MHz
- Software configurable for differing system requirements
  - 16-bit or 32-bit bus width
  - Configurable row cycle delay (tRC), row precharge delay (tRP), row-to-column delay (tRCD), and column-to-data delay (CAS latency)
- Built-in auto-refresh timer and state machine
- Hardware-supported self-refresh entry and exit: capability to maintain valid data during system reset and low-power modes
- Auto-powerdown (clock suspend) timer

1.2.3.2 Direct Memory Access Controller

The Direct Memory Access Controller (DMAC) provides 16 channels to support linear memory, 2D memory, FIFO and end-of-burst enable FIFO transfers to support a wide variety of DMA operations. Features include:

- Supports 16 channels linear memory, 2D memory and FIFO for both source and destination
- Supports 8-bit, 16-bit, or 32-bit FIFO port size and memory port size data transfer
- DMA burst length is configurable up to maximum of 16 words, 32 half-words, or 64 bytes for each channel
- Bus utilization control for a channel that is not triggered by DMA request
- Interrupts provided to interrupt handler on bulk data transfer complete or transfer error

- DMA burst time-out error to terminate DMA cycle when the burst cannot be completed in a programmed timing period
- Dedicated external DMA request and grant signal
- Support increment, decrement and no increment for source and destination addressing
- Supports DMA chaining

1.2.3.3 NAND Flash Controller

The NAND Flash controller (NFC) interfaces standard NAND Flash parts to the i.MX21 processor and hides the complexities of accessing NAND Flash. The NFC features include:

- Contains hardware bootloader for automatic boot up from NAND Flash devices
- Supports all 8-bit/16-bit NAND Flash devices regardless of density and organization
- Supports 512 byte and 2 Kbyte page sizes
- Internal 2 Kbyte of buffer RAM used as boot RAM during cold startup and as read/write page buffers to relieve CPU intervention
- Automatic ECC detection and selectable correction
- Data protection for ram buffer and NAND Flash pages

1.2.3.4 External Interface Module

The External Interface module (EIM) handles the interface to devices external to the i.MX21 processor, including generation of chip selects for external peripherals and memory, and provides the following features:

- Six Chip Selects (CS0-5) for external devices, each with 16 Mbyte of address space
- CS0 supports boot from ROM, NAND, or NOR Flash of up to 32 Mbyte of address space
- Programmable protection, port size, and wait states for each chip-select
- Internal/external boot ROM selection
- Selectable bus watchdog counter
- Burst support for external AMD™ or Intel® flash with 32-bit data path
- External Data Transfer Acknowledge (DTACK) support for slower devices connected on CS5

1.2.4 Bus Master Interface (BMI)

The BMI module enables high speed connection between the i.MX21 processor and the alternate bus master devices in the system. The BMI provides support for the following functions:

- Supports 8- or 16-bit data bus mode
- Supports external bus master read or write to CPU using memory access timing
- Supports CPU write to external bus slave using memory access timing
- Supports ATI graphic chip burst read/write accesses timing
- High communication speed
- Supports DMA

1.2.5 Inter-Chip Connectivity

This section describes how the modules within the i.MX21 processor interface with each other and provides a high-level overview on how the architecture of the busses are configured and multiplexed.

1.2.5.1 Inter-IC (I^2C) Bus Module

I^2C is a two-wire, bidirectional serial bus that provides a simple, efficient method of data exchange, minimizing the interconnection between devices. This bus is suitable for applications requiring occasional communications over a short distance between many devices. The flexible I^2C allows additional devices to be connected to the bus for expansion and system development. The I^2C features include:

- Multiple-master operation
- Software-programmable for 1 of 64 different serial clock frequencies
- Interrupt-driven, byte-by-byte data transfer
- Arbitration-lost interrupt with automatic mode switching from master to slave
- Calling address identification interrupt
- Start and stop signal generation and detection
- Repeated START signal generation
- Acknowledge bit generation and detection
- Bus-busy detection

1.2.5.2 Three Configurable Serial Peripheral Interfaces for High Speed Data Transfer

The i.MX21 processor has three Configurable Serial Peripheral Interface (CSPI) modules which allow rapid data communication with fewer software interrupts than conventional serial communications. Each CSPI is equipped with data FIFO and is a master/slave configurable serial peripheral interface module, allowing the i.MX21 processor to interface with external SPI master or slave devices.

- Master/slave configurable
- Two chip selects each for master mode operation
- Up to 16-bit programmable data transfer
- 8 × 16 FIFO for both transmit and receive data

1.2.5.3 Two Synchronous Serial Interfaces with Inter-IC Sound (I^2S) and AC97 Host Controller Module (SSI/I2S/AC97)

Features include the following:

- Supports generic SSI interface for timeslot based communication with synchronous voice codecs
- Timeslot mode supports up to 4 channels for communication among devices Bluetooth voice port, voice codecs and baseband audio ports
- Supports Philips standard Inter-IC Sound (I^2S) bus for external digital audio chip interface at 44.1 kHz and 48 kHz

- AC97 Host Controller mode with support for 2 audio channels supporting variable and fixed rate transfers. Fixed mode only supports 48 kHz sampling rate.
- Used together with the Digital Audio Mux (AUDMUX) module to provide flexible audio and voice routing options

1.2.6 Peripheral Support

The i.MX21 processor provides a CMOS sensor interface (CSI) that allows the integration of a extensive variety of popular camera features.

1.2.6.1 CMOS Sensor Interface (CSI)

The CSI features include the following:

- Configurable interface supports wide variety of popular CMOS sensors that output data in YUV, RGB, or Bayer format
- Supports CCIR656 format
- Statistic data generation for auto exposure and auto white balance control which is required for Bayer data
- DMA support for image data transfer
- Private data bus to eMMA Pre-processor module for video and image preprocessing

1.2.7 Display and Video Modules

There are two separate LCD controllers in the i.MX21 processor—the LCDC and SLCDC that support both dumb and smart LCD panels. A dumb LCD panel has no built-in memory and requires an external controller to send display data at a fixed rate. Such panels typically support high refresh rates suitable for graphics, games, and video applications. The LCD controller in the i.MX21 processor is an AHB master and can transfer display data from system memory (SDRAM).

Smart panels have built-in memory and a display controller. An advantage of the built-in memory and controller, is that the refresh function is done by the local LCD controller and only data that is changing must be updated thus offering a reduced transfer rate and lower power operation.

Both LCD controllers in the i.MX21 processor provide glueless connection to external gray-scale or color LCD panels.

The video input port in the i.MX21 processor supports a direct interface to commonly available CMOS sensors. Together with other system resources (DMA and hardware Pre-processor), viewfinder functions can be achieved with extremely low CPU MIPS and low system power consumption.

1.2.7.1 LCD Controller (LCDC)

The LCDC features include the following:

- Software programmable screen size (up to 800 × 600) to support single (non-split) monochrome, color STN panels and color TFT panels

- Support color depth for CSTN panels: 4- or 8-bit mapping from 256 × 18 table, 12-bit true color
- Support color depth for TFT panels: 4- or 8-bit mapping from 256 × 18 table, 16-bit/18-bit/24-bit true color
- Up to 16 grey levels out of 16 palettes
- Capable of directly driving popular LCD drivers from manufacturers including Motorola, Sharp, Hitachi, and Toshiba
- Support for data bus width of 16-bit or 18-bit TFT panels
- Support for data bus width of 8-bit, 4-bit, 2-bit, and 1-bit monochrome LCD panels
- Direct interface to Sharp® 320 × 240 and 240 × 320 HR-TFT panels and other generic panels
- Support for logical operation between color hardware cursor and background
- LCD contrast control using 8-bit PWM
- Support for self-refresh LCD modules
- Hardware panning (soft horizontal scrolling)
- Windowing support for one graphic or text overlay

1.2.7.2 Smart LCD Controller (SLCDC)

The SLCDC transparently and efficiently transfers image data from system memory to an external LCD controller. The SLCDC module contains a DMA controller that transfers image and control data from system memory to the SLCDC FIFO where it is formatted and sent out to the external LCD controller.

The SLCDC can be configured to write image data to an external LCD controller via a 4-line serial, 3-line serial, an 8- or 16- bit parallel interface. The SLCDC has two FIFOs where command and display data are loaded via DMA. The display data is tagged with commands that are used by the SLCDC to communicate display information and data to the Smart LCD panel.

The command tagged data format of the SLCDC provides flexibility and ease of connection to existing and new smart LCD panels.

1.2.8 enhanced Multimedia Accelerator (eMMA)

The i.MX21 processor comes with an enhanced Multimedia Accelerator (eMMA) comprising an ISO/IEC 14496-2 compliant MPEG-4 encoder and decoder, independent Pre-processing and Post-processing stages which provide exceptional image and video quality. The eMMA represents a major breakthrough to solve the problem of high MIPS requirement for video encode and decode operations in mobile and wireless applications. Tight integration and memory pipelining coupled with AHB master mode operation ensures minimal system loading. To further offload the CPU, live video stream data enters the eMMA module directly through an internal private data interface. The eMMA architecture is shown in Figure 1-2.

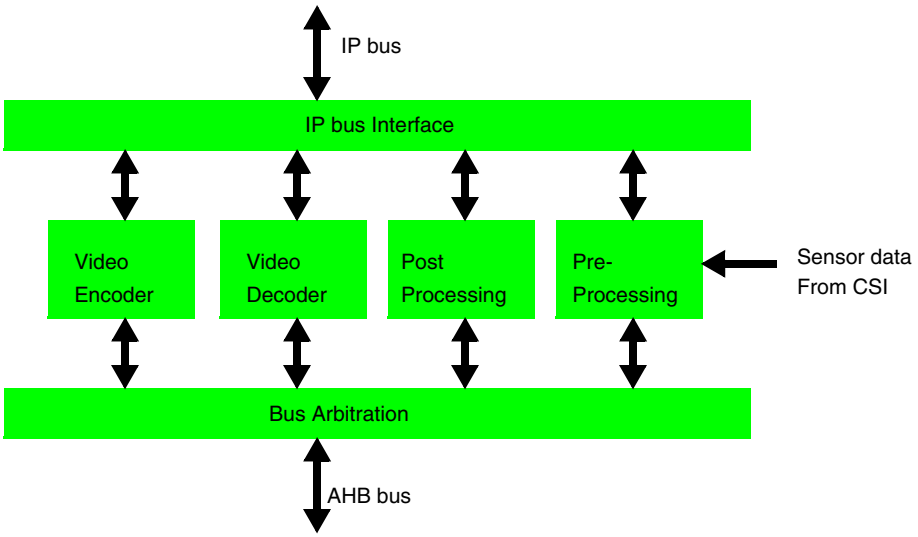


Figure 1-2. eMMA Architecture

1.2.8.1 Video Encoder

eMMA provides the following video encoder capabilities:

- Supports MPEG4 and H.263 (Short Video Header)
- Fully conforms to ISO/IEC 14496-2 Visual Simple Profiles Levels 0 to 3
- Supports real-time encoding images of sizes from 32 × 32 up to CIF or QVGA at 30 fps
- Input data format is YUV 4:2:0 (Planar)
- Supports camera stabilization

1.2.8.2 Video Decoder

eMMA provides the following video decoder capabilities:

- Supports MPEG4 and H.263 (Short Video Header)
- Fully conforms to ISO/IEC 14496-2 Visual Simple Profile Levels 0 to 3
- Supports real-time decoding of image sizes up to CIF or QVGA at 30 fps
- Output data format is YUV 4:2:0 (Planar)

1.2.8.3 Image Pre-processor (PrP)

The image Pre-processor block, shown in Figure 1-3, performs color space conversion and image resizing for viewfinder display and data formatting for video encoder and still image for input to a hardware or software based video encoder or image compressor. The Pre-processor has two media input and output paths and can accept input from system memory or from a private data bus connected to the CMOS Sensor Interface (CSI) module. The Pre-processor can apply frame rate control on the live video stream from the CSI module to adjust for different processing load conditions. The Pre-processor's two output channels are used to output RGB data for display of local camera view and to output image data for compression by the hardware encoder or a software encoder (still image or video encode).

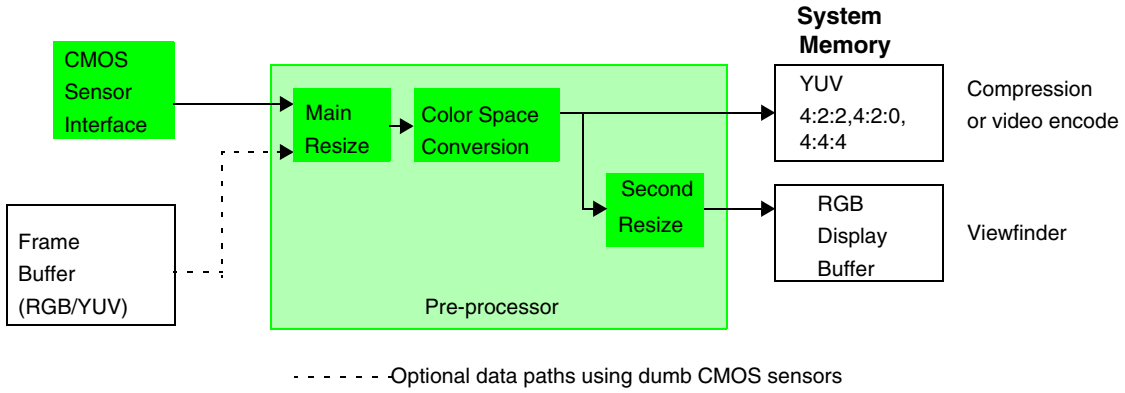


Figure 1-3. Pre-processor Data Flow

Pre-processor features:

- Data input:
  - System memory
  - Private DMA between CMOS Sensor Interface module and pre-processor
- Data input formats:
  - Arbitrarily unpacked RGB input
  - YUV 4:2:2 (Interleaved)
  - YUV 4:2:0 (Planar)
- Input image size: 2044 × 2044
- Image scaling:
  - Main resize ratio: 8:1–1:1 in integral steps, Horizontal 9:8/vertical 6:5 and Horizontal 9:8/Vertical 1:1
  - Secondary resize ratio for viewfinder: 8:1–1:1 in integral steps
- Output data format:
  - RGB565
  - YUV 4:2:2 (Interleaved)
  - YUV 4:2:0 (Planar)
- RGB data and one YUV data format can be generated concurrently

1.2.8.4 Postprocessor (PP)

The Postprocessor, shown in Figure 1-4, performs Deblock, Dering, Image Resize and Color Space Conversion (CSC) functions on the input image data. These functions provide flexibility to meet various RGB formats and YUV formats for display. Besides working in tandem with the decoder sub-block in the eMMA, the Postprocessor can also be used by software decoders (other than MPEG4) to touch up the final output before display. The sub-blocks that perform Deblock, Dering, Resize and CSC operations can be selectively bypassed through software configuration. Figure 1-4 shows the flow for video postprocessing.

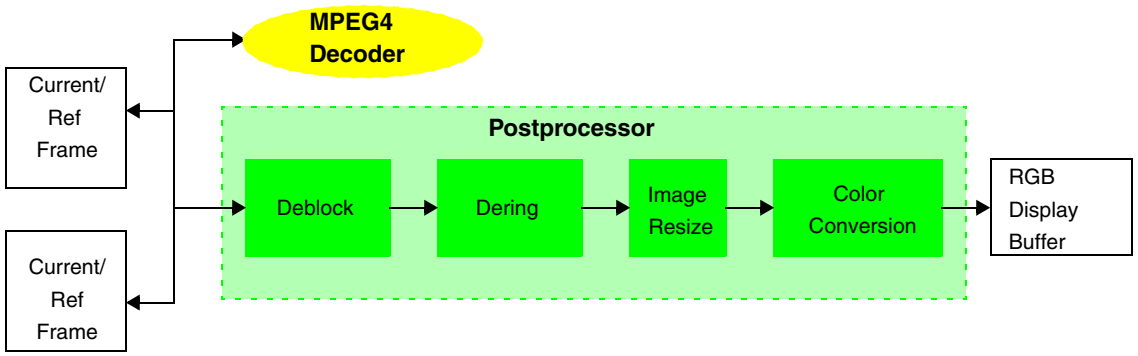


Figure 1-4. Postprocessor

Postprocessor features:

- Input data:
  - From system memory
- Input format:
  - YUV 4:2:0 (Planar)
- Output format:
  - YUV422
  - RGB444
  - RGB565
  - RGB666
  - RGB888 (unpacked)
- Input Size: Maximum size of 2044 × 2044
- Image Resize:
  - Upscaling ratios ranging from 1:1 to 1:4 in fractional steps
  - Downscaling ratios ranging from 1:1 to 2:1 in fractional steps and a fixed 4:1
  - Ratios provide scaling between QCIF, CIF, QVGA (320 × 240) and QVGA (240 × 320)

1.2.8.5 Two Multimedia Card and Secure Digital Host Controller Modules

The Multimedia Card/Secure Digital Host module (MMC/SD) integrates MMC support with SD memory and I/O functions. The features include:

- Fully compatible with the MMC system specification version 2.2
- Fully compatible with the SD Memory Card specification 1.0 and SD I/O specification 1.0 with 1 and 4 channel(s)
- Up to ten MMC cards and one SD supported by standard (maximum data rate with up to ten cards)
- Supports hot swappable operation
- Data rates from 25 Mbps to 100 Mbps
- Dedicated power pin

- Part of the External Memory Interface (EMI) complex comprising the Nand Flash Controller, Wireless External Interface to Memory (WEIM) and SDRAM Controller

1.2.8.6 Digital Audio Mux

The Digital Audio Mux (AUDMUX) provides a programmable interconnect fabric for voice, audio and synchronous data routing between the i.MX21 processor's SSI modules and external SSI, audio and voice codecs. The AUDMUX features include:

- Supports 1 host and 3 peripheral interfaces
- Flexible audio, voice and data routing without host processor intervention
- Built-in support for network mode connection of host and peripheral interfaces
- Separate and simultaneous audio paths from hosts to peripherals
- External 4-wire connection to synchronous devices, audio and voice codecs

1.2.9 Connectivity and Expansion

There are multiple peripheral modules in the i.MX21 processor that provide external connection capability. All peripherals that have FIFOs support DMA transfers to and from the FIFOs. This minimizes CPU intervention and reduces interrupt overhead to the system. The exception to this is the Pulse Width Modulator that includes FIFOs, however, does not support DMA.

1.2.9.1 Four Universal Asynchronous Receiver/Transmitters (UART1, UART2, UART3, and UART4)

The UART modules are capable of standard RS-232 non-return-to-zero (NRZ) encoding format and IrDA-compatible infrared modes. Each UART provides serial communication capability with external devices through an RS-232 cable or through use of external circuitry that converts infrared signals to electrical signals (for reception) or transforms electrical signals to signals that drive an infrared LED (for transmission) to provide low speed IrDA compatibility to the i.MX21 processor. Features include:

- Supports serial data transmit/receive operation: 7 or 8 data bits, 1 or 2 stop bits, programmable parity (even, odd, or none)
- Automatic baud rate detection
- 32-bytes FIFO for transmit and 32 half-words FIFO for receive data
- IrDA Serial Infra-Red (SIR) mode support

1.2.9.2 USB On-The-Go (USB OTG) Controller

The USB controller in the i.MX21 processor implements the USB On-The-Go (USB OTG) supplement. The USB OTG module is complaint to the USB 2.0 and operates at full and low speeds as specified in USB 2.0. The OTG port is capable of connecting to a USB host or client device and uses the Host Negotiation Protocol (HNP) and Session Request Protocol (SRP) to switch between Host and Function roles. One of the dual host ports is dedicated for connection to a smartphone USB client device and the other host port is available for connection to other client devices. The connection to the smartphone forms the interprocessor link as an alternate to a UART-based link.

Built-in switching logic implements a bypass mode in which the internal host port is bypassed to allow an external USB host and the smartphone USB client to be directly connected. This feature enables the external USB host to directly control the smartphone modem for debug or for production programming. Figure 1-5 shows the USB OTG block.

The USB OTG module is a bus master and takes ownership of the bus for DMA. This allows the USB OTG to continue operation while the CPU is in a low power mode.

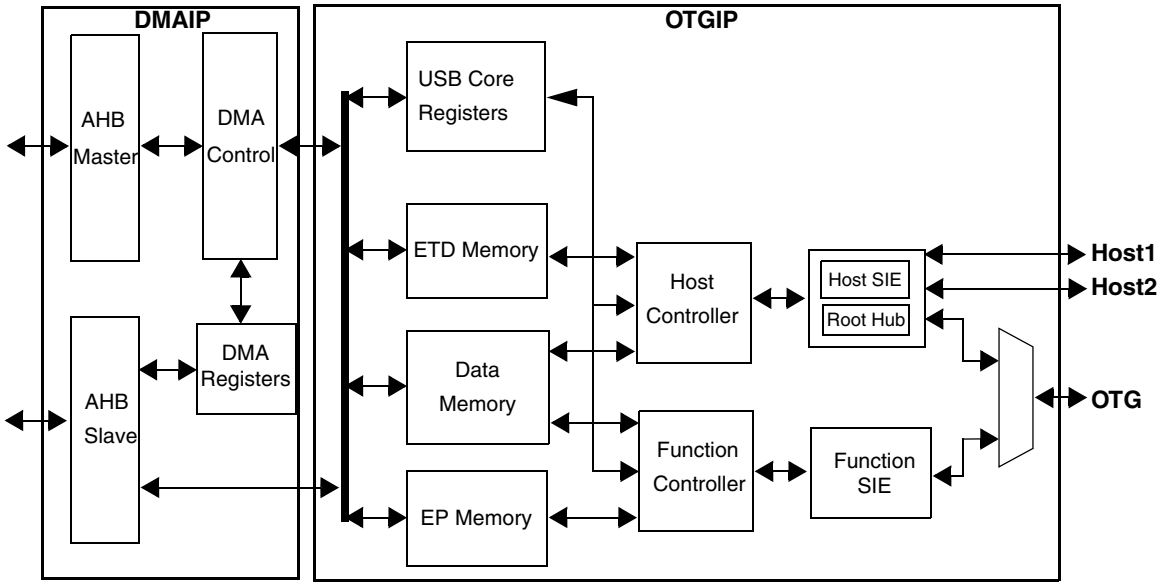


Figure 1-5. USB On-The-Go Controller Block Diagram

- Compliant with the USB 2.0 specification for operation at full speed (12 Mbit/sec) and low speed (1.5 Mbit/sec)
- Fully compliant with the USB On-The-Go specification
- Host Negotiation Protocol (HNP) and Session Request Protocol (SRP) Protocol (SRP) must be implemented in software
- Transaction scheduling and transfer level protocol implemented in hardware including bandwidth management, data toggle and retry
- AMBA AHB 2.0 Bus Master DMA Controller:
  - 32 DMA Channels for Host Controller EndPoint Transfer Descriptors
  - 32 DMA Channels for Function Controller EndPoint Descriptors
- USB function supports 32 physical endpoints:
  - 16 IN endpoints and 16 OUT endpoints
  - Programmable for type (control, interrupt, bulk, isochronous), packet size, and buffering
- Double buffering support for all four types of host and function controller transactions
- Separate descriptor and data memory space
- Direct device to device transfers in one frame
- Power savings mode for Host Controller and suspend mode for Function Controller

- All USB host ports support external transceiver bypass mode

1.2.10 Debug Capability

The i.MX21 processor offers designers and programers with full-debug capabilities through industry-standard JTAG interface and the ability to bootload using either a serial or USB interface.

- UART Bootstrap mode function:
  - Allows system initialization and program or data download to system memory via USB or UART1
  - Accepts execution command to run program stored in system memory
  - Supports memory/register read/write operation of selectable data size of byte, half-word, or word
  - Provides a 16-byte instruction buffer for ARM instruction storage and execution
- USB Bootstrap mode function
  - Supports bootstrapping through USB OTG port
- JTAG port to support generic ARM debug tools

1.2.10.1 PCMCIA/CF Interface

The PCMCIA/CF host controller module provides all the control logic for a PCMCIA socket interface and requires only additional external analog power switching logic and buffering. The controller supports one PCMCIA socket and includes the following features:

- PCMCIA/CF host controller interface compliant with the PCMCIA standard release 2.1 (for I/O Cards) and fully compliant with the Compact Flash Specification V1.4
- Supports one PCMCIA or CF socket
- Supports hot-insertion, card detection and removal
- Mapping to common memory space, attribute memory space and I/O space. Each space is up to 32 kbyte in size.
- Supports 5 programmable memory and IO windows.
- Generates a single interrupt to the CPU
- Programmable card access timing to interface with slower devices
- Supports TrueIDE mode
- Provides special control signals for external buffering to separate high and low speed paths

1.2.10.2 Keypad Port

The Keypad Port is a 16-bit peripheral which can be used either for keypad matrix scanning or as general purpose I/O. Features include:

- Supports up to 8 × 8 external key pad matrix
- Open drain design
- Glitch suppression circuit prevents erroneous key detection

- Multiple keys detection
- Standby key press detection

1.2.10.3 Fast Infra-Red Interface (FIRI)

The Fast Infra-Red Interface (USB device module) in the i.MX21 processor implements both the Medium Infra-Red (MIR) and Fast Infra-Red (FIR) protocols. In MIR mode, the FIRI supports wireless communications at 0.576 Mbps and 1.152 Mbps and uses a framed transmission protocol which follows the High-Level Data Link Controller (HDLC) protocol. In FIR mode, the module operates at 4 Mbps with 4 Pulse Position Modulation (4PPM) defined by IrDA, version 1.4.

In addition to the MIR and FIR modes, the i.MX21 processor supports SIR protocol on all 4 of the UART modules. Only UART1 may be used together with the FIRI as these two modules share their pins for transparent speed and protocol stepping from SIR to MIR or SIR to FIR modes.

Figure 1-6 shows the FIRI sharing pins with the UART module and pin selection is controlled via GPIO configuration.

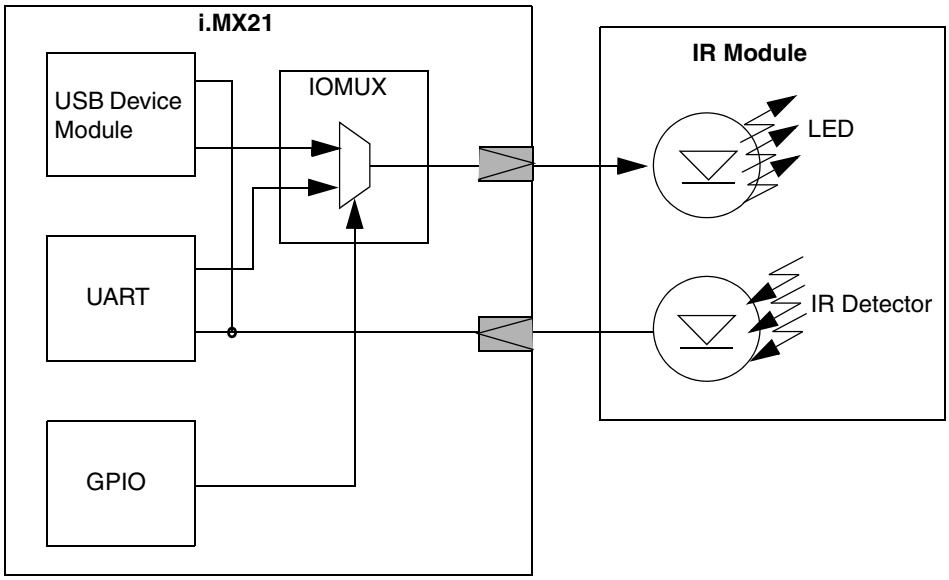


Figure 1-6. Fast Infra-Red Interface

The USB device module can be divided to the following functional parts:

- Packet assembler
- Searcher
- 4PPM modulator/demodulator
- CRC32 encoder and decoder
- DMA capable receive and transmit FIFOs

1.2.10.4 1-Wire® Interface

The 1-Wire module is a peripheral device that communicates with the ARM926EJ-S Core and provides a communication line to a 1 Kbit Add-Only Memory (DS2502). The 1-Wire interface features include:

- Supports 1-Wire interface to a 1Kbit Add-Only Memory (DS2502)
- Implements 1-Wire protocol defined by Dallas Semiconductors

1.2.11 Power Management

The i.MX21 processor's power management features are as follows:

- Support for 3 power modes of operation: RUN, DOZE, and STOP
- Aggressive clock gating within modules to minimize CMOS switching power
- Active well biasing technique to reduce standby mode current consumption
- Voltage/frequency scalable capability (Lib or silicon characterization is needed)

1.2.12 Electronic and Package Information

The i.MX21 processor features the following electronic and package information:

- Operating voltage
  - I/O voltage: 1.7 V to 2.0 V or 2.7 V to 3.3 V
  - Internal logic voltage: 1.45 V to 1.65 V
- Package
  - Type: 0.65 mm and 0.5 mm pitch MAPBGA
  - Dimensions: 14mm × 14mm for 0.65 mm type
  - Pins: 289

Chapter 2
Signal Descriptions and Pin Assignments

This chapter identifies and describes the i.MX21 signals and their pin assignments. It also contains a short section on power-up sequence of the i.MX21.

2.1 Signal Descriptions

Table 2-1 provides complete information on the signal multiplexing of the i.MX21 signals.

The connections of the pins in Table 2-1 depends solely upon the user application, however there are a few factory test signals that are not used in a normal application. Following is a list of these signals and how they are to be terminated for proper operation of the i.MX21 processor:

- CLKMODE[1:0]: To ensure proper operation, leave these signals as no connects.
- OSC26M_TEST: To ensure proper operation, leave this signal as no connect.
- EXT_48M: To ensure proper operation, connect this signal to ground.
- EXT_266M: To ensure proper operation, connect this signal to ground.
- TEST_WB[2:0]: These signals are also multiplexed with GPIO PORT E as well as alternate keypad signals. If not utilizing these signals for GPIO functionality or for their other multiplexed function, then configure as GPIO input with pull up enabled, and leave as a no connect.
- TEST_WB[4:3]: To ensure proper operation, leave these signals as no connects.

Table 2-1. i.MX21 Signal Descriptions

Signal Name	Function/Notes
External Bus/Chip Select (EIM)	
A [25:0]	Address bus signals
D [31:0]	Data bus signals
$\overline{EB0}$	MSB Byte Strobe—Active low external enable byte signal that controls D [31:24], shared with SDRAM DQM0.
$\overline{EB1}$	Byte Strobe—Active low external enable byte signal that controls D [23:16], shared with SDRAM DQM1.
$\overline{EB2}$	Byte Strobe—Active low external enable byte signal that controls D [15:8], shared with SDRAM DQM2 and PCMCIA $\overline{PC\_REG}$.
$\overline{EB3}$	LSB Byte Strobe—Active low external enable byte signal that controls D [7:0], shared with SDRAM DQM3 and PCMCIA $\overline{PC\_IORD}$.
$\overline{OE}$	Memory Output Enable—Active low output enables external data bus, shared with PCMCIA $\overline{PC\_IOWR}$.
$\overline{CS}$ [5:0]	Chip Select—The chip select signals $\overline{CS}$ [3:2] are multiplexed with $\overline{CSD}$ [1:0] and are selected by the Function Multiplexing Control Register (FMCR) in the System Control chapter. By default $\overline{CSD}$ [1:0] is selected. DTACK is multiplexed with $\overline{CS4}$.

Table 2-1. i.MX21 Signal Descriptions (continued)

Signal Name	Function/Notes
$\overline{\text{ECB}}$	Active low input signal sent by flash device to the EIM whenever the flash device must terminate an on-going burst sequence and initiate a new (long first access) burst sequence.
$\overline{\text{LBA}}$	Active low signal sent by flash device causing the external burst device to latch the starting burst address.
BCLK	Clock signal sent to external synchronous memories (such as burst flash) during burst mode.
$\overline{\text{RW}}$	$\overline{\text{RW}}$ signal—Indicates whether external access is a read (high) or write (low) cycle. This signal is also shared with the PCMCIA $\overline{\text{PC\_WE}}$.
DTACK	DTACK signal—External input data acknowledge signal, multiplexed with $\overline{\text{CS4}}$.
Bootstrap	
BOOT [3:0]	System Boot Mode Select—The operational system boot mode upon system reset is determined by the settings of these pins. To hardwire these inputs low, terminate with a 1 KΩ resister to ground. For a logic high, terminate with a 1 KΩ resistor to VDDA. Do not change the state of these inputs after power-up. Boot 3 should always be tied to logic low.
SDRAM Controller	
SDBA [4:0]	SDRAM non-interleave mode bank address signals. These signals are multiplexed with address signals A[20:16].
SDIBA [3:0]	SDRAM interleave addressing mode bank address signals. These signals are multiplexed with address signals A[24:21].
MA [11:0]	SDRAM address signals. MA[9:0] are multiplexed with address signals A[10:1].
DQM [3:0]	SDRAM data qualifier mask multiplexed with $\overline{\text{EB}}$[3:0]. DQM3 corresponds to D[31:24], DQM2 corresponds to D[23:16], DQM1 corresponds to D[15:8], and DQM0 corresponds to D[7:0].
$\overline{\text{CSD0}}$	SDRAM Chip Select signal. This signal is multiplexed with the $\overline{\text{CS2}}$ signal. This signal is selectable by programming the Function Multiplexing Control Register in the System Control chapter.
$\overline{\text{CSD1}}$	SDRAM Chip Select signal. This signal is multiplexed with the $\overline{\text{CS3}}$ signal. This signal is selectable by programming the Function Multiplexing Control Register in the System Control chapter.
$\overline{\text{RAS}}$	SDRAM Row Address Select signal.
$\overline{\text{CAS}}$	SDRAM Column Address Select signal
$\overline{\text{SDWE}}$	SDRAM Write Enable signal
SDCKE0	SDRAM Clock Enable 0
SDCKE1	SDRAM Clock Enable 1
SDCLK	SDRAM Clock
Clocks and Resets	
EXTAL26M	Crystal input (26MHz), or a 16 MHz to 32 MHz oscillator (or square-wave) input when the internal oscillator circuit is shut down. When using an external signal source, feed this input with a square wave signal switching from GND to VDDA.
XTAL26M	Oscillator output to external crystal. When using an external signal source, float this output.
EXTAL32K	32 kHz or 32.768 kHz crystal input. When using an external signal source, feed this input with a square wave signal switching from GND to QVDD5.
XTAL32K	Oscillator output to external crystal. When using an external signal source, float this output.
CLKO	Clock Out signal selected from internal clock signals. Please refer to clock controller for internal clock selection.

Table 2-1. i.MX21 Signal Descriptions (continued)

Signal Name	Function/Notes
EXT_48M	This is a special factory test signal. To ensure proper operation, connect this signal to ground.
EXT_266M	This is a special factory test signal. To ensure proper operation, connect this signal to ground.
$\overline{\text{RESET\_IN}}$	Master Reset—External active low Schmitt trigger input signal. When this signal goes active, all modules (except the reset module, SDRAMC module, and the clock control module) are reset.
$\overline{\text{RESET\_OUT}}$	Reset Out—Internal active low output signal from the Watchdog Timer module and is asserted from the following sources: Power-on reset, External reset ($\overline{\text{RESET\_IN}}$), and Watchdog time-out.
$\overline{\text{POR}}$	Power On Reset—Active low Schmitt trigger input signal. The $\overline{\text{POR}}$ signal is normally generated by an external RC circuit designed to detect a power-up event.
CLKMODE[1:0]	These are special factory test signals. To ensure proper operation, leave these signals as no connects.
OSC26M_TEST	This is a special factory test signal. To ensure proper operation, leave this signal as a no connect.
TEST_WB[2:0]	These are special factory test signals. However, these signals are also multiplexed with GPIO PORT E as well as alternate keypad signals. If not using these signals for GPIO functions or for other multiplexed functions, then configure as GPIO input with pull-up enabled, and leave as a no connect.
TEST_WB[4:3]	These are special factory test signals. To ensure proper operation, leave these signals as no connects.
WKGD	Battery indicator input used to qualify the walk-up process. Also multiplexed with TIN.
JTAG For termination recommendations, see the Table "*JTAG pinouts*" in the *Multi-ICE® User Guide* from ARM® Limited.	
$\overline{\text{TRST}}$	Test Reset Pin—External active low signal used to asynchronously initialize the JTAG controller.
TDO	Serial Output for test instructions and data. Changes on the falling edge of TCK.
TDI	Serial Input for test instructions and data. Sampled on the rising edge of TCK.
TCK	Test Clock to synchronize test logic and control register access through the JTAG port.
TMS	Test Mode Select to sequence the JTAG test controller's state machine. Sampled on the rising edge of TCK.
JTAG_CTRL	JTAG Controller select signal—JTAG_CTRL is sampled during the rising edge of TRST. Must be pulled to logic high for proper JTAG interface to debugger. Pulling JTAG_CRTL low is for internal test purposes only.
RTCK	JTAG Return Clock used to enhance stability of JTAG debug interface devices. This signal is multiplexed with 1-Wire, therefore using 1-Wire renders RTCK unusable and vice versa.
CMOS Sensor Interface	
CSI_D [7:0]	Sensor port data
CSI_MCLK	Sensor port master clock
CSI_VSYNC	Sensor port vertical sync
CSI_HSYNC	Sensor port horizontal sync
CSI_PIXCLK	Sensor port data latch clock
LCD Controller	
LD [17:0]	LCD Data Bus—All LCD signals are driven low after reset and when LCD is off. LD[15:0] signals are multiplexed with SLCDC1_DAT[15:0] from SLCDC1 and BMI_D[15:0]. LD[17] signal is multiplexed with $\overline{\text{BMI\_WRITE}}$ of BMI. LD[16] is multiplexed with BMI_READ_REQ of BMI and $\overline{\text{EXT\_DMAGRANT}}$.
FLM_VSYNC (or simply referred to as VSYNC)	Frame Sync or Vsync—This signal also serves as the clock signal output for gate driver (dedicated signal SPS for Sharp panel HR-TFT). This signal is multiplexed with BMI_RXF_FULL and $\overline{\text{BMI\_WAIT}}$ of the BMI.

Table 2-1. i.MX21 Signal Descriptions (continued)

Signal Name	Function/Notes
LP_HSYNC (or simply referred to as HSYNC)	Line Pulse or HSync
LSCLK	Shift Clock. This signal is multiplexed with the BMI_CLK_CS from BMI.
OE_ACD	Alternate Crystal Direction/Output Enable.
CONTRAST	This signal is used to control the LCD bias voltage as contrast control. This signal is multiplexed with the $\overline{\text{BMI\_READ}}$ from BMI.
SPL_SPR	Sampling start signal for left and right scanning. This signal is multiplexed with the SLCDC1_CLK.
PS	Control signal output for source driver (Sharp panel dedicated signal). This signal is multiplexed with the SLCDC1_CS.
CLS	Start signal output for gate driver. This signal is invert version of PS (Sharp panel dedicated signal). This signal is multiplexed with the SLCDC1_RS.
REV	Signal for common electrode driving signal preparation (Sharp panel dedicated signal). This signal is multiplexed with SLCDC1_D0.
Smart LCD Controller	
SLCDC1_CLK	SLCDC Clock output signal. This signal is multiplexed and available at 2 alternate locations. These are SPL_SPR and SD2_CLK signals of LCDC and SD2, respectively.
SLCDC1_CS	SLCDC Chip Select output signal. This signal is multiplexed and available at 2 alternate signal locations. These are PS and SD2_CMD signals of LCDC and SD2, respectively.
SLCDC1_RS	SLCDC Register Select output signal. This signal is multiplexed and available at 2 alternate signal locations. These are CLS and SD2_D3 signals of LCDC and SD2, respectively.
SLCDC1_D0	SLCDC serial data output signal. This signal is multiplexed and available at 2 alternate signal locations. These are and REV and SD2_D2 signals of LCDC and SD2, respectively. This signal is inactive when a parallel data interface is used.
SLCDC1_DAT[15:0]	SLCDC Data output signals for connection to a parallel SLCD panel interface. These signals are multiplexed with LD[15:0] while an alternate 8-bit SLCD muxing is available on LD[15:8]. Further alternate muxing of these signals are available on some of the USB OTG and USBH1 signals.
SLCDC2_CLK	SLCDC Clock input signal for pass through to SLCD device. This signal is multiplexed with SSI3_CLK signal from SSI3.
SLCDC2_CS	SLCDC Chip Select input signal for pass through to SLCD device. This signal is multiplexed with SSI3_TXD signal from SSI3.
SLCDC2_RS	SLCDC Register Select input signal for pass through to SLCD device. This signal is multiplexed with SSI3_RXD signal from SSI3.
SLCDC2_D0	SLCD Data input signal for pass through to SLCD device. This signal is multiplexed with SSI3_FS signal from SSI3.
Bus Master Interface (BMI)	
BMI_D[15:0]	BMI bidirectional data bus. Bus width is programmable between 8-bit or 16-bit.These signals are multiplexed with LD[15:0] and SLCDC_DAT[15:0].
BMI_CLK_CS	BMI bidirectional clock or chip select signal.This signal is multiplexed with LSCLK of LCDC.
$\overline{\text{BMI\_WRITE}}$	BMI bidirectional signal to indicate read or write access. This is an input signal when the BMI is a slave and an output signal when BMI is the master of the interface. $\overline{\text{BMI\_WRITE}}$ is asserted for write and negated for read.This signal is muxed with LD[17] of LCDC.
$\overline{\text{BMI\_READ}}$	BMI output signal to enable data read from external slave device. This signal is not used and driven high when BMI is slave.This signal is multiplexed with CONTRAST signal of LCDC.

Table 2-1. i.MX21 Signal Descriptions (continued)

Signal Name	Function/Notes
BMI_READ_REQ	BMI Read request output signal to external bus master. This signal is active when the data in the TXFIFO is larger or equal to the data transfer size of a single external BMI access.This signal is muxed with LD[16] of LCDC.
BMI_RXF_FULL	BMI Receive FIFO full active high output signal to reflect if the RxFIFO reaches water mark value.This signal is muxed with VSYNC of the LCDC.
$\overline{\text{BMI\_WAIT}}$	BMI Wait—Active low signal to wait for data ready (read cycle) or accepted (write_cycle). Also multiplexed with VSYNC.
External DMA	
$\overline{\text{EXT\_DMAREQ}}$	External DMA Request input signal. This signal is multiplexed with CSPI1_$\overline{\text{RDY}}$.
$\overline{\text{EXT\_DMAGRANT}}$	External DMA Grant output signal. This signal is multiplexed with LD[16] of LCDC and CSPI1_SS1 of CSPI1.
NAND Flash Controller	
NF_CLE	NAND Flash Command Latch Enable output signal. Multiplexed with PC_POE of PCMCIA.
$\overline{\text{NF\_CE}}$	NAND Flash Chip Enable output signal. This signal is multiplexed with PC_CE1 of PCMCIA.
$\overline{\text{NF\_WP}}$	NAND Flash Write Protect output signal. This signal is multiplexed with PC_CE2 of PCMCIA.
NF_ALE	NAND Flash Address Latch Enable output signal. This signal is multiplexed with $\overline{\text{PC\_OE}}$ of PCMCIA.
$\overline{\text{NF\_RE}}$	NAND Flash Read Enable output signal. This signal is multiplexed with $\overline{\text{PC\_RW}}$ of PCMCIA.
$\overline{\text{NF\_WE}}$	NAND Flash Write Enable output signal. This signal is multiplexed with and PC_BVD2 of PCMCIA.
NF_RB	NAND Flash Ready Busy input signal. This signal is multiplexed with PC_RST of PCMCIA.
NF_IO[15:0]	NAND Flash Data input and output signals. NF_IO[15:7] signals are multiplexed with A[25:21] and A[15:13]. NF_IO[7:0] signals are multiplexed with several PCMCIA signals.
PCMCIA Controller	
PC_A[25:0]	PCMCIA Address signals. These signals are multiplexed with A[25:0].
PC_D[15:0]	PCMCIA Data input and output signals. These signals are multiplexed with D[15:0].
$\overline{\text{PC\_CD1}}$	PCMCIA Card Detect1 input signal. This signal is multiplexed with NFIO[7] signal of NF.
$\overline{\text{PC\_CD2}}$	PCMCIA Card Detect2 input signal. This signal is multiplexed with NFIO[6] signal of NF.
$\overline{\text{PC\_WAIT}}$	PCMCIA Wait input signal to extend current access. This signal is multiplexed with NFIO[5] signal of NF.
PC_READY	PCMCIA Ready input signal indicates card is ready for access. Multiplexed with NFIO[4] signal of NF.
PC_RST	PCMCIA Reset output signal. This signal is multiplexed with NFRB signal of NF.
$\overline{\text{PC\_OE}}$	PCMCIA Memory Read Enable output signal asserted during common or attribute memory read cycles. This signal is multiplexed with NFALE signal of NF.
$\overline{\text{PC\_WE}}$	PCMCIA Memory Write Enable output signal asserted during common or attribute memory cycles. This signal is shared with $\overline{\text{RW}}$ of the EIM.
PC_VS1	PCMCIA Voltage Sense1 input signal. This signal is multiplexed with NFIO[2] signal of NF.
PC_VS2	PCMCIA Voltage Sense2 input signal. This signal is multiplexed with NFIO[1] signal of NF.
PC_BVD1	PCMCIA Battery Voltage Detect1 input signal. This signal is multiplexed with NFIO[0] signal of NF.
PC_BVD2	PCMCIA Battery Voltage Detect2 input signal. This signal is multiplexed with $\overline{\text{NF\_WE}}$ signal of NF.
PC_SPKOUT	PCMCIA Speaker Out output signal. This signal is multiplexed with PWMO signal.
$\overline{\text{PC\_REG}}$	PCMCIA Register Select output signal. This signal is shared with $\overline{\text{EB2}}$ of EIM.

Table 2-1. i.MX21 Signal Descriptions (continued)

Signal Name	Function/Notes
PC_CE1	PCMCIA Card Enable1 output signal. This signal is multiplexed with NFCE signal of NF.
PC_CE2	PCMCIA Card Enable2 output signal. This signal is multiplexed with NFWP signal of NF.
PC_IORD	PCMCIA IO Read output signal. This signal is shared with EB3 of EIM.
PC_IOWR	PCMCIA IO Write output signal. This signal is shared with OE signal of EIM.
PC_WP	PCMCIA Write Protect input signal. This signal is multiplexed with NFIO[3] signal of NF.
PC_POE	PCMCIA Output Enable signal to enable voltage translation buffers and transceivers. This signal is multiplexed with NFCLE signal of NF.
PC_RW	PCMCIA Read Write output signal to control external transceiver direction. Asserted high for read access and negated low for write access. This signal is multiplexed with NFRE signal of NF.
PC_PWRON	PCMCIA input signal to indicate that the card power has been applied and stabilized.
CSPI	
CSPI1_MOSI	Master Out/Slave In signal
CSPI1_MISO	Master In/Slave Out signal
CSPI1_SS[2:0]	Slave Select (Selectable polarity) signal. CSPI1_SS2 is also multiplexed with USBG_RXDAT and CSPI1_SS1 is multiplexed with EXT_DMAGRANT.
CSPI1_SCLK	Serial Clock signal
CSPI1_RDY	Serial Data Ready signal. Also multiplexed with EXT_DMAREQ.
CSPI2_MOSI	Master Out/Slave In signal. This signal is multiplexed with USBH2_TXDP signal of USB OTG.
CSPI2_MISO	Master In/Slave Out signal. This signal is multiplexed with USBH2_TXDM signal of USB OTG.
CSPI2_SS[2:0]	Slave Select (Selectable polarity) signals. These signals are multiplexed with USBH2_FS, USBH2_RXDP and USBH2_RXDM signal of USB OTG
CSPI2_SCLK	Serial Clock signal. This signal is multiplexed with USBH2_OE signal of USB OTG
CSPI3_MOSI	Master Out/Slave In signal. This signal is multiplexed with SD1_CMD.
CSPI3_MISO	Master In/Slave Out signal. This signal is multiplexed with SD1_D0.
CSPI3_SS	Slave Select (Selectable polarity) signal multiplexed with SD1_D3.
CSPI3_SCLK	Serial Clock signal. This signal is multiplexed with SD1_CLK.
General Purpose Timers	
TIN	Timer Input Capture or Timer Input Clock—The signal on this input is applied to all 3 timers simultaneously. This signal is muxed with the Walk-up Guard Mode WKGD signal in the PLL, Clock, and Reset Controller module.
TOUT1 (or simply TOUT)	Timer Output signal from General Purpose Timer1 (GPT1). This signal is multiplexed with SYS_CLK1 and SYS_CLK2 signal of SSI1 and SSI2. The pin name of this signal is simply TOUT.
TOUT2	Timer Output signal from General Purpose Timer1 (GPT2). This signal is multiplexed with PWMO.
TOUT3	Timer Output signal from General Purpose Timer1 (GPT3). This signal is multiplexed with PWMO.
USB On-The-Go	
USB_BYP	USB Bypass input active low signal. This signal can only be used for USB function, not for GPIO.
USB_PWR	USB Power output signal
USB_OC	USB Over current input signal. This signal can only be used for USB function, not for GPIO.

Table 2-1. i.MX21 Signal Descriptions (continued)

Signal Name	Function/Notes
USBG_RXDP	USB OTG Receive Data Plus input signal. This signal is muxed with SLCDC1_DAT15.
USBG_RXDM	USB OTG Receive Data Minus input signal. This signal is muxed with SLCDC1_DAT14.
USBG_TXDP	USB OTG Transmit Data Plus output signal. This signal is muxed with SLCDC1_DAT13.
USBG_TXDM	USB OTG Transmit Data Minus output signal. This signal is muxed with SLCDC1_DAT12.
USBG_RXDAT	USB OTG Transceiver differential data receive signal. Multiplexed with CSPI1_SS2.
$\overline{\text{USBG\_OE}}$	USB OTG Output Enable signal. This signal is muxed with SLCDC1_DAT11.
$\overline{\text{USBG\_ON}}$	USB OTG Transceiver ON output signal. This signal is muxed with SLCDC1_DAT9.
USBG_FS	USB OTG Full Speed output signal. This signal is multiplexed with external transceiver $\overline{\text{USBG\_TXR\_INT}}$ signal of USB OTG. This signal is muxed with SLCDC1_DAT10.
USBH1_RXDP	USB Host1 Receive Data Plus input signal. This signal is multiplexed with UART4_RXD and SLCDC1_DAT6. It also provides an alternative multiplex for $\overline{\text{UART4\_RTS}}$, where this signal is selectable by programming the Function Multiplexing Control Register in the System Control chapter.
USBH1_RXDM	USB Host1 Receive Data Minus input signal. This signal is muxed with SLCDC1_DAT5. It also provides an alternative multiplex for $\overline{\text{UART4\_CTS}}$.
USBH1_TXDP	USB Host1 Transmit Data Plus output signal. This signal is multiplexed with $\overline{\text{UART4\_CTS}}$ and SLCDC1_DAT4. It also provides an alternative multiplex for UART4_RXD, where this signal is selectable by programming the Function Multiplexing Control Register in the System Control chapter.
USBH1_TXDM	USB Host1 Transmit Data Minus output signal. Multiplexed with UART4_TXD and SLCDC1_DAT3.
USBH1_RXDAT	USB Host1 Transceiver differential data receive signal. Multiplexed with USBH1_FS.
$\overline{\text{USBH1\_OE}}$	USB Host1 Output Enable signal. This signal is muxed with SLCDC1_DAT2.
USBH1_FS	USB Host1 Full Speed output signal. Multiplexed with $\overline{\text{UART4\_RTS}}$ and SLCDC1_DAT1 and USBH1_RXDAT.
$\overline{\text{USBH\_ON}}$	USB Host transceiver ON output signal. This signal is muxed with SLCDC1_DAT0.
USBH2_RXDP	USB Host2 Receive Data Plus input signal. This signal is multiplexed with CSPI2_SS[1] of CSPI2.
USBH2_RXDM	USB Host2 Receive Data Minus input signal. This signal is multiplexed with CSPI2_SS[2] of CSPI2.
USBH2_TXDP	USB Host2 Transmit Data Plus output signal. This signal is multiplexed with CSPI2_MOSI of CSPI2.
USBH2_TXDM	USB Host2 Transmit Data Minus output signal. This signal is multiplexed with CSPI2_MISO of CSPI2.
$\overline{\text{USBH2\_OE}}$	USB Host2 Output Enable signal. This signal is multiplexed with CSPI2_SCLK of CSPI2.
USBH2_FS	USB Host2 Full Speed output signal. This signal is multiplexed with CSPI2_SS[0] of CSPI2.
USBG_SCL	USB OTG I^2C Clock input/output signal. This signal is multiplexed with SLCDC1_DAT8.
USBG_SDA	USB OTG I^2C Data input/output signal. This signal is multiplexed with SLCDC1_DAT7.
$\overline{\text{USBG\_TXR\_INT}}$	USB OTG transceiver interrupt input. Multiplexed with USBG_FS.
Secure Digital Interface	
SD1_CMD	SD Command bidirectional signal—If the system designer does not want to make use of the internal pull-up, via the Pull-up enable register, a 4.7k–69k external pull-up resistor must be added. This signal is multiplexed with CSPI3_MOSI.
SD1_CLK	SD Output Clock. This signal is multiplexed with CSPI3_SCLK.
SD1_D[3:0]	SD Data bidirectional signals—If the system designer does not want to make use of the internal pull-up, via the Pull-up enable register, a 50k–69k external pull-up resistor must be added. SD1_D[3] is muxed with CSPI3_SS while SD1_D[0] is muxed with CSPI3_MISO.

Table 2-1. i.MX21 Signal Descriptions (continued)

Signal Name	Function/Notes
SD2_CMD	SD Command bidirectional signal. This signal is multiplexed with SLCDC1_CS signal from SLCDC1.
SD2_CLK	SD Output Clock signal. This signal is multiplexed with SLCDC1_CLK signal from SLCDC1.
SD2_D[3:0]	SD Data bidirectional signals. SD2_D[3:2] are multiplexed with SLCDC1_RS and SLCDC_D0 signals from SLCDC1.
UARTs – IrDA/Auto-Bauding	
UART1_RXD	Receive Data input signal
UART1_TXD	Transmit Data output signal
$\overline{\text{UART1\_RTS}}$	Request to Send input signal
$\overline{\text{UART1\_CTS}}$	Clear to Send output signal
UART2_RXD	Receive Data input signal. This signal is multiplexed with KP_ROW6 signal from KPP.
UART2_TXD	Transmit Data output signal. This signal is multiplexed with KP_COL6 signal from KPP.
$\overline{\text{UART2\_RTS}}$	Request to Send input signal. This signal is multiplexed with KP_ROW7 signal from KPP.
$\overline{\text{UART2\_CTS}}$	Clear to Send output signal. This signal is multiplexed with KP_COL7 signal from KPP.
UART3_RXD	Receive Data input signal. This signal is multiplexed with IR_RXD from FIRI.
UART3_TXD	Transmit Data output signal. This signal is multiplexed with IR_TXD from FIRI.
$\overline{\text{UART3\_RTS}}$	Request to Send input signal
$\overline{\text{UART3\_CTS}}$	Clear to Send output signal
UART4_RXD	Receive Data input signal which is multiplexed with USBH1_RXDP and USBH1_TXDP.
UART4_TXD	Transmit Data output signal which is multiplexed with USBH1_TXDM.
$\overline{\text{UART4\_RTS}}$	Request to Send input signal which is multiplexed with USBH1_FS and USBH1_RXDP.
$\overline{\text{UART4\_CTS}}$	Clear to Send output signal which is multiplexed with USBH1_TXDP and USBH1_RXDM.
Serial Audio Port – SSI (configurable to I^2S protocol and AC97)	
SSI1_CLK	Serial clock signal which is output in master or input in slave
SSI1_TXD	Transmit serial data
SSI1_RXD	Receive serial data
SSI1_FS	Frame Sync signal which is output in master and input in slave
SYS_CLK1	SSI1 master clock. Multiplexed with TOUT.
SSI2_CLK	Serial clock signal which is output in master or input in slave.
SSI2_TXD	Transmit serial data signal
SSI2_RXD	Receive serial data
SSI2_FS	Frame Sync signal which is output in master and input in slave.
SYS_CLK2	SSI2 master clock. Multiplexed with TOUT.
SSI3_CLK	Serial clock signal which is output in master or input in slave. Multiplexed with SLCDC2_CLK
SSI3_TXD	Transmit serial data signal which is multiplexed with SLCDC2_CS
SSI3_RXD	Receive serial data which is multiplexed with SLCDC2_RS
SSI3_FS	Frame Sync signal which is output in master and input in slave. Multiplexed with SLCDC2_D0.
SAP_CLK	Serial clock signal which is output in master or input in slave.

Table 2-1. i.MX21 Signal Descriptions (continued)

Signal Name	Function/Notes
SAP_TXD	Transmit serial data
SAP_RXD	Receive serial data
SAP_FS	Frame Sync signal which is output in master and input in slave.
I²C	
I2C_CLK	I²C Clock
I2C_DATA	I²C Data
1-Wire	
OWIRE	1-Wire input and output signal. This signal is multiplexed with JTAG RTCK.
PWM	
PWMO	PWM Output. This signal is multiplexed with PC_SPKOUT of PCMCIA, as well as TOUT2 and TOUT3 of the General Purpose Timer module.
General Purpose Input/Output	
PF[16]	Dedicated GPIO. When unused, program this signal as an input with the on-chip pull-up resistor enabled.
Keypad	
KP_COL[7:0]	Keypad Column selection signals. KP_COL[7:6] are multiplexed with $\overline{\text{UART2\_CTS}}$ and UART2_TXD respectively. Alternatively, KP_COL6 is also available on the internal factory test signal TEST_WB2. The Function Multiplexing Control Register in the System Control chapter must be used in conjunction with programming the GPIO multiplexing (to select the alternate signal multiplexing) to choose which signal KP_COL6 is available.
KP_ROW[7:0]	Keypad Row selection signals. KP_ROW[7:6] are multiplexed with $\overline{\text{UART2\_RTS}}$ and UART2_RXD signals respectively. Alternatively, KP_ROW7 and KP_ROW6 are available on the internal factory test signals TEST_WB0 and TEST_WB1 respectively. The Function Multiplexing Control Register in the System Control chapter must be used in conjunction with programming the GPIO multiplexing (to select the alternate signal multiplexing) to choose which signals KP_ROW6 and KP_ROW7 are available.
Noisy Supply Pins	
NVDD	Noisy Supply for the I/O pins. There are six (6) I/O voltages, NVDD1 through NVDD6.
NVSS	Noisy Ground for the I/O pins
Supply Pins – Analog Modules	
VDDA	Supply for analog blocks
QVSS (internally connected to AVSS)	Quiet GND for analog blocks (QVSS and AVSS are synonymous)
Internal Power Supplies	
QVDD	Power supply pins for silicon internal circuitry
QVSS	Quiet GND pins for silicon internal circuitry
QVDDX	Power supply pin for the ARM core. Externally connect directly to QVDD

2.2 I/O Power Supply and Signal Multiplexing Scheme

This section describes detailed information about both the power supply for each I/O pin and its function multiplexing scheme. The user can reference information provided in Table 2-2 to configure the power supply scheme for each device in the system (memory and external peripherals). The function multiplexing information also shown in Table 2-2 allows the user to select the function of each pin by configuring the appropriate GPIO registers when those pins are multiplexed to provide different functions. In some cases, the use of the Function Multiplexing Control Register (FMCR) in the System Control chapter may be required to select multiplexed functionality. Table 2-1 on page 2-1 indicates which signals require the use of the FMCR.

Table 2-2. i.MX21 Signal Multiplexing Scheme
(*Note: See end of table for table footnotes and table legend*)

I/O Supply Voltage	BGA Pins	Primary				Alternate				GPIO							Reset (At/After)	Default
		Signal	Dir	PU	OD	Signal	Dir	PU	OD	Mux	PU	AIN	BIN	CIN	AOUT	BOUT		
NVDD1	F3	A25_NFIO15	B														L	A25_NFIO15
NVDD1	F2	D31	B	KP													KP/I[1]	D31
NVDD1	F1	A24_NFIO14	B														L	A24_NFIO14
NVDD1	G4	D30	B	KP													KP/I[1]	D30
NVDD1	G3	A23_NFIO13	B														L	A23_NFIO13
NVDD1	G2	D29	B	KP													KP/I[1]	D29
NVDD1	G1	A22_NFIO12	B														L	A22_NFIO12
NVDD1	H4	D28	B	KP													KP/I[1]	D28
NVDD1	H3	A21_NFIO11	B														L	A21_NFIO11
NVDD1	H2	D27	B	KP													KP/I[1]	D27
NVDD1	H1	A20	O														L	A20
NVDD1	J4	D26	B	KP													KP/I[1]	D26
NVDD1	J1	A19	O														L	A19
NVDD1	J3	D25	B	KP													KP/I[1]	D25
NVDD1	J2	A18	O														L	A18
NVDD1	K4	D24	B	KP													KP/I[1]	D24
NVDD1	K2	A17	O														L	A17
NVDD1	K3	D23	B	KP													KP/I[1]	D23
NVDD1	H7	NVDD1																NVDD1
NVDD1	K1	A16	O														L	A16
NVDD1	K7	NVSS1																NVSS1
NVDD1	L4	D22	B	KP													KP/I[1]	D22
NVDD1	L2	A15_NFIO10	B														L	A15_NFIO10
NVDD1	L3	D21	B	KP													KP/I[1]	D21

Table 2-2. i.MX21 Signal Multiplexing Scheme (continued)
(*Note: See end of table for table footnotes and table legend*)

I/O Supply Voltage	BGA Pins	Primary				Alternate				GPIO							Reset (At/After)	Default
		Signal	Dir	PU	OD	Signal	Dir	PU	OD	Mux	PU	AIN	BIN	CIN	AOUT	BOUT		
NVDD1	L1	A14_NFIO9	B														L	A14_NFIO9
NVDD1	M3	D20	B	KP													KP/I[1]	D20
QVDD	D5	QVDD																QVDD
QVSS	D6	QVSS																QVSS
NVDD1	M2	A13_NFIO8	B														L	A13_NFIO8
NVDD1	M1	D19	B	KP													KP/I[1]	D19
NVDD1	N2	A12	O														L	A12
NVDD1	M4	D18	B	KP													KP/I[1]	D18
NVDD1	N1	A11	O														L	A11
NVDD1	N3	D17	B	KP													KP/I[1]	D17
NVDD1	P2	A10	O														L	A10
NVDD1	N4	D16	B	KP													KP/I[1]	D16
NVDD1	P1	A9	O														L	A9
NVDD1	P3	D15	B	KP													KP/I[1]	D15
NVDD1	R2	A8	O														L	A8
NVDD1	P4	D14	B	KP													KP/I[1]	D14
NVDD1	J7	NVDD1																NVDD1
NVDD1	R1	A7	O														L	A7
NVSS1	L7	NVSS1																NVSS1
NVDD1	R3	D13	B	KP													KP/I[1]	D13
NVDD1	T2	A6	O														L	A6
NVDD1	R4	D12	B	KP													KP/I[1]	D12
NVDD1	T1	A5	O														L	A5
NVDD1	U1	D11	B	KP													KP/I[1]	D11
NVDD1	V1	A4	O														L	A4
NVDD1	V2	$\overline{EB0}$/DQM0	O														H	$\overline{EB0}$/DQM0
NVDD1	T4	D10	B	KP													KP/I[1]	D10
NVDD1	U2	$\overline{EB1}$/DQM1	O														H	$\overline{EB1}$/DQM1
NVDD1	V3	D9	B	KP													KP/I[1]	D9
NVDD1	U3	$\overline{EB2}$/DQM2/$\overline{PC\_REG}$	O														H	$\overline{EB2}$/DQM2/$\overline{PC\_REG}$
NVDD1	W1	A3	O														L	A3
NVDD1	T3	$\overline{EB3}$/DQM3/$\overline{PC\_IORD}$	O														H	$\overline{EB3}$/DQM3/$\overline{PC\_IORD}$
NVDD1	V4	D8	B	KP													KP/I[1]	D8

Table 2-2. i.MX21 Signal Multiplexing Scheme (continued)
(*Note: See end of table for table footnotes and table legend*)

I/O Supply Voltage	BGA Pins	Primary				Alternate				GPIO							Reset (At/After)	Default
		Signal	Dir	PU	OD	Signal	Dir	PU	OD	Mux	PU	AIN	BIN	CIN	AOUT	BOUT		
NVDD1	U4	$\overline{\text{OE/PC\_IOWR}}$	O														H	OE/PC_IOWR
NVDD1	V5	$\overline{\text{CS5}}$	O							PF22	PUEN						Pull-H	PF22
NVDD1	W2	A2	O														L	A2
NVDD1	W3	D7	B	KP													KP/I[1]	D7
NVDD1	U5	$\overline{\text{CS4}}$	O							PF21	PUEN				$\overline{\text{DTACK}}$		Pull-H	PF21
NVSS1	N10	NVSS1																NVSS1
NVDD1	W4	A1	O														L	A1
NVDD1	N12	NVDD1																NVDD1
NVDD1	T5	$\overline{\text{CS3}}$/$\overline{\text{CSD1}}$	O														H	$\overline{\text{CSD1}}$
NVDD1	W5	$\overline{\text{CS2}}$/$\overline{\text{CSD0}}$	O														H	$\overline{\text{CSD0}}$
NVDD1	U6	D6	B	KP													KP/I[1]	D6
NVDD1	W6	A0	O														L	A0
NVDD1	T6	$\overline{\text{CS1}}$	O														H	$\overline{\text{CS1}}$
NVDD1	V6	D5	B	KP													KP/I[1]	D5
NVDD1	T7	BCLK	O														H	BCLK
NVDD1	V7	$\overline{\text{CS0}}$	O														H	$\overline{\text{CS0}}$
NVDD1	U7	$\overline{\text{ECB}}$	I	PU													H	$\overline{\text{ECB}}$
NVDD1	W7	D4	B	KP													KP/I[1]	D4
NVDD1	N7	$\overline{\text{LBA}}$	O														H	$\overline{\text{LBA}}$
NVDD1	V8	$\overline{\text{RW/PC\_WE}}$	O														H	$\overline{\text{RW/PC\_WE}}$
NVDD1	U8	D3	B	KP													KP/I[1]	D3
NVDD1	W8	D2	B	KP													KP/I[1]	D2
NVDD1	T8	MA11	O														L	MA11
NVDD1	V9	D1	B	KP													KP/I[1]	D1
NVDD1	U9	MA10	O														L	MA10
NVDD1	W9	D0	B	KP													KP/I[1]	D0
NVDD1	V10	JTAG_CTRL	I	PU													Pull-H	JTAG_CTRL
QVDD	K12	QVDD																QVDD
QVSS	K13	QVSS																QVSS
NVDD1	N13	NVDD1																NVDD1
NVDD1	W10	SDCLK	O														H[2]	SDCLK
NVSS1	N11	NVSS1																NVSS1
NVDD1	U10	PC_PWRON	I	PD													L	PC_PWRON

Table 2-2. i.MX21 Signal Multiplexing Scheme (continued)
(*Note: See end of table for table footnotes and table legend*)

I/O Supply Voltage	BGA Pins	Primary				Alternate				GPIO							Reset (At/After)	Default
		Signal	Dir	PU	OD	Signal	Dir	PU	OD	Mux	PU	AIN	BIN	CIN	AOUT	BOUT		
NVDD1	T9	$\overline{\text{RAS}}$	O														H	$\overline{\text{RAS}}$
NVDD1	T10	$\overline{\text{CAS}}$	O														H	$\overline{\text{CAS}}$
NVDD1	V11	$\overline{\text{SDWE}}$	O														H	$\overline{\text{SDWE}}$
NVDD1	N9	SDCKE0	O														H	SDCKE0
NVDD1	W11	SDCKE1	O														H	SDCKE1
NVDD1	U11	Reserved	I							PF16	PUEN						Pull-H	PF16
NVDD1	V12	CLKO	O							PF15	PUEN						N/A	CLKO
NVDD1	M11	NFIO7	B							PF14	PUEN				$\overline{\text{PC\_CD1}}$		Pull-H	NFIO7
NVDD1	V13	NFIO6	B							PF13	PUEN				$\overline{\text{PC\_CD2}}$		Pull-H	NFIO6
NVDD1	T11	NFIO5	B							PF12	PUEN				$\overline{\text{PC\_WAIT}}$		Pull-H	NFIO5
NVDD1	U12	NFIO4	B							PF11	PUEN				PC_READY		Pull-H	NFIO4
NVDD1	T12	NFIO3	B							PF10	PUEN				PC_WP		Pull-H	NFIO3
NVDD1	L11	NFIO2	B							PF9	PUEN				PC_VS1		Pull-H	NFIO2
NVDD1	U13	NFIO1	B							PF8	PUEN				PC_VS2		Pull-H	NFIO1
NVDD1	W12	NFIO0	B							PF7	PUEN				PC_BVD1		Pull-H	NFIO0
NVDD1	T13	$\overline{\text{NFWE}}$	O							PF6	PUEN				PC_BVD2		H	$\overline{\text{NFWE}}$
NVDD1	W13	$\overline{\text{NFRE}}$	O							PF5	PUEN	$\overline{\text{PC\_RW}}$					H	$\overline{\text{NFRE}}$
NVDD1	U14	NFALE	O							PF4	PUEN	$\overline{\text{PC\_OE}}$					L	NFALE
NVDD1	U15	NFCLE	O							PF3	PUEN	PC_POE					L	NFCLE
NVDD1	L12	$\overline{\text{NFWP}}$	O							PF2	PUEN	PC_CE2					H	$\overline{\text{NFWP}}$
NVDD1	T15	$\overline{\text{NFCE}}$	O							PF1	PUEN	PC_CE1					H	$\overline{\text{NFCE}}$
NVDD1	M12	NFRB	I							PF0	PUEN	PC_RST					Pull-H	NFRB
NVDD1	M13	EXT_48M	I														N/A[3]	EXT_48M
NVDD1	R17	EXT_266M	I														N/A[3]	EXT_266M
NVDD2	M7	NVDD2																NVDD2
NVDD2	R16	SD1_CLK	O			CSPI3_SCLK	O			PE23	PUEN						Pull-H	PE23
NVDD2	P17	SD1_CMD	B			CSPI3_MOSI	O			PE22	PUEN						Pull-H	PE22
NVDD2	T17	SD1_D3	B			CSPI3_SS	O			PE21	PUEN						Pull-H	PE21
NVDD2	P16	SD1_D2	B							PE20	PUEN						Pull-H	PE20
NVDD2	N18	SD1_D1	B							PE19	PUEN						Pull-H	PE19
NVDD2	N16	SD1_D0	B			CSPI3_MISO	I			PE18	PUEN						Pull-H	PE18
NVSS2	R18	NVSS2																NVSS2
VDDA	V18	VDDA																VDDA

Table 2-2. i.MX21 Signal Multiplexing Scheme (continued)
(*Note: See end of table for table footnotes and table legend*)

I/O Supply Voltage	BGA Pins	Primary				Alternate				GPIO							Reset (At/After)	Default
		Signal	Dir	PU	OD	Signal	Dir	PU	OD	Mux	PU	AIN	BIN	CIN	AOUT	BOUT		
VDDA	W16	EXTAL26M	I														N/A	EXTAL26M
QVSS	V14	QVSS																QVSS
VDDA	W17	XTAL26M	O														N/A	XTAL26M
VDDA	V17	OSC26M_TEST	I														N/A[3]	OSC26M_TEST
VDDA	T14	$\overline{\text{RESET\_IN}}$	I	PU													*See*[5]	$\overline{\text{RESET\_IN}}$
VDDA	V15	$\overline{\text{RESET\_OUT}}$	O							PE17	PUEN						L/H	$\overline{\text{RESET\_OUT}}$
VDDA	U16	$\overline{\text{POR}}$	I														*See*[5]	$\overline{\text{POR}}$
VDDA	U18	BOOT3	I														*See*[6]	BOOT3
VDDA	U17	BOOT2	I														*See*[6]	BOOT2
VDDA	T16	BOOT1	I														*See*[6]	BOOT1
VDDA	V16	BOOT0	I														*See*[6]	BOOT0
VDDA	T18	CLKMODE1	I	PU													N/A[3]	CLKMODE1
VDDA	T19	CLKMODE0	I	PU													N/A[3]	CLKMODE0
QVDD	L9	QVDD																QVDD
QVSS	L10	QVSS																QVSS
QVDD	V19	EXTAL32K	I														N/A	EXTAL32K
QVDD	U19	XTAL32K	I														N/A	XTAL32K
NVDD3	R19	$\overline{\text{TRST}}$	I	PU													Pull-H	$\overline{\text{TRST}}$
NVDD3	P19	TMS	I	PU													Pull-H	TMS
NVDD3	N17	TCK	I	PU													Pull-H	TCK
NVDD3	P18	TDI	I	PU													Pull-H	TDI
NVDD3	K11	TDO	O														H	TDO
NVDD3	N19	RTCK	O			OWIRE	B		OD	PE16	PUEN						L	RTCK
NVDD3	M18	$\overline{\text{UART1\_RTS}}$	I							PE15	PUEN						Pull-H	$\overline{\text{UART1\_RTS}}$
NVDD3	M19	$\overline{\text{UART1\_CTS}}$	O							PE14	PUEN						H	$\overline{\text{UART1\_CTS}}$
NVDD3	K10	UART1_RXD	I							PE13	PUEN						Pull-H	UART1_RXD
NVDD3	L13	UART1_TXD	O							PE12	PUEN						H	UART1_TXD
NVDD3	L17	$\overline{\text{UART3\_RTS}}$	I							PE11	PUEN						Pull-H	PE11
NVDD3	L18	$\overline{\text{UART3\_CTS}}$	O							PE10	PUEN						Pull-H	PE10
NVDD3	M17	UART3_RXD	I							PE9	PUEN				IR_RXD		Pull-H	PE9
NVDD3	L19	UART3_TXD	O							PE8	PUEN	IR_TXD					Pull-H	PE8
NVDD3	M16	UART2_RXD	I			KP_ROW6	B			PE7	PUEN						Pull-H	UART2_RXD
NVDD3	L16	UART2_TXD	O			KP_COL6	B		ODEN	PE6	PUEN						H	UART2_TXD

Table 2-2. i.MX21 Signal Multiplexing Scheme (continued)
(*Note: See end of table for table footnotes and table legend*)

I/O Supply Voltage	BGA Pins	Primary				Alternate				GPIO							Reset (At/After)	Default
		Signal	Dir	PU	OD	Signal	Dir	PU	OD	Mux	PU	AIN	BIN	CIN	AOUT	BOUT		
NVDD3	K17	KP_COL5	B	PU	ODEN												Pull-H	KP_COL5
NVDD3	K18	KP_COL4	B	PU	ODEN												Pull-H	KP_COL4
NVDD3	K16	KP_COL3	B	PU	ODEN												Pull-H	KP_COL3
NVDD3	K19	KP_COL2	B	PU	ODEN												Pull-H	KP_COL2
NVDD3	J17	KP_COL1	B	PU	ODEN												Pull-H	KP_COL1
NVDD3	J18	KP_COL0	B	PU	ODEN												Pull-H	KP_COL0
NVDD3	J13	TEST_WB0	B			KP_ROW7	B			PE2	PUEN						Pull-H[3]	TEST_WB0
NVDD3	J19	TEST_WB1	B			KP_ROW6	B			PE1	PUEN						Pull-H[3]	TEST_WB1
NVDD3	H17	TEST_WB2	B			KP_COL6	B		ODEN	PE0	PUEN						Pull-H[3]	TEST_WB2
VDDA	H18	TEST_WB3	I														Pull-H[3]	TEST_WB3
VDDA	H16	TEST_WB4	I														Pull-H[3]	TEST_WB4
NVDD3	H19	PWMO	O							PE5	PUEN	PC_SPKOUT	TOUT2	TOUT3			Pull-H	PE5
NVDD3	J16	$\overline{\text{UART2\_RTS}}$	I			KP_ROW7	B			PE4	PUEN						Pull-H	$\overline{\text{UART2\_RTS}}$
NVDD3	G18	$\overline{\text{UART2\_CTS}}$	O			KP_COL7	B		ODEN	PE3	PUEN						H	$\overline{\text{UART2\_CTS}}$
NVDD3	J10	KP_ROW5	B	PU													Pull-H	KP_ROW5
NVDD3	G19	KP_ROW4	B	PU													Pull-H	KP_ROW4
NVDD3	G17	KP_ROW3	B	PU													Pull-H	KP_ROW3
QVDD	W14	QVDD																QVDD
QVSS	W15	QVSS																QVSS
NVDD3	J11	KP_ROW2	B	PU													Pull-H	KP_ROW2
NVDD3	G16	KP_ROW1	B	PU													Pull-H	KP_ROW1
NVDD3	F18	KP_ROW0	B	PU													Pull-H	KP_ROW0
NVDD3	J12	CSPI1_MOSI	B							PD31	PUEN						Pull-H	PD31
NVDD3	F17	CSPI1_MISO	B							PD30	PUEN						Pull-H	PD30
NVDD3	H10	CSPI1_SCLK	B							PD29	PUEN						Pull-H	PD29
NVDD3	F19	CSPI1_SS0	B							PD28	PUEN						Pull-H	PD28
NVDD3	F16	CSPI1_SS1	B							PD27	PUEN		$\overline{\text{EXT\_DMA\_GRANT}}$				Pull-H	PD27
NVDD3	E18	CSPI1_SS2	B							PD26	PUEN				USBG_RXDAT		Pull-H	PD26
NVDD3	H11	CSPI1_$\overline{\text{RDY}}$	I							PD25	PUEN				$\overline{\text{EXT\_DMAREQ}}$		Pull-H	PD25
NVDD3	E19	CSPI2_MOSI	B							PD24	PUEN	USBH2_TXDP					Pull-H	PD24
NVDD3	E17	CSPI2_MISO	B							PD23	PUEN	USBH2_TXDM					Pull-H	PD23

Table 2-2. i.MX21 Signal Multiplexing Scheme (continued)
(*Note: See end of table for table footnotes and table legend*)

I/O Supply Voltage	BGA Pins	Primary				Alternate				GPIO							Reset (At/After)	Default
		Signal	Dir	PU	OD	Signal	Dir	PU	OD	Mux	PU	AIN	BIN	CIN	AOUT	BOUT		
NVDD3	D19	CSPI2_SCLK	B							PD22	PUEN	USBH2_OE					Pull-H	PD22
NVDD3	D18	CSPI2_SS0	B							PD21	PUEN	USBH2_FS					Pull-H	PD21
NVSS3	N8	NVSS3																NVSS3
QVDD	W18	QVDD																QVDD
QVSS	W19	QVSS																QVSS
NVDD3	M8	NVDD3																NVDD3
NVDD3	C19	CSPI2_SS1	B							PD20	PUEN				USBH2_RXDP		Pull-H	PD20
NVDD3	B19	CSPI2_SS2	B							PD19	PUEN				USBH2_RXDM		Pull-H	PD19
NVDD3	C18	I2C_CLK	B		OD					PD18	PUEN						Pull-H	PD18
NVDD3	B18	I2C_DATA	B		OD					PD17	PUEN						Pull-H	PD17
NVDD3	C17	SSI3_CLK	B			SLCDC2_CLK	I			PC31	PUEN						Pull-H	PC31
NVDD3	B17	SSI3_TXD	B			SLCDC2_CS	I			PC30	PUEN						Pull-H	PC30
NVDD3	C16	SSI3_RXD	B			SLCDC2_RS	I			PC29	PUEN						Pull-H	PC29
NVDD3	A19	SSI3_FS	B			SLCDC2_D0	I			PC28	PUEN						Pull-H	PC28
NVDD3	D17	SSI2_CLK	B							PC27	PUEN						Pull-H	PC27
NVDD3	A18	SSI2_TXD	B							PC26	PUEN						Pull-H	PC26
NVDD3	A17	SSI2_RXD	B							PC25	PUEN						Pull-H	PC25
NVDD3	B16	SSI2_FS	B							PC24	PUEN						Pull-H	PC24
NVDD3	A16	SSI1_CLK	B							PC23	PUEN						Pull-H	PC23
NVDD3	C15	SSI1_TXD	B							PC22	PUEN						Pull-H	PC22
NVDD3	D16	SSI1_RXD	B							PC21	PUEN						Pull-H	PC21
NVDD3	B15	SSI1_FS	B							PC20	PUEN						Pull-H	PC20
NVDD3	E16	SAP_CLK	B							PC19	PUEN						Pull-H	PC19
NVDD3	A15	SAP_TXD	B							PC18	PUEN						Pull-H	PC18
NVDD3	D15	SAP_RXD	B							PC17	PUEN						Pull-H	PC17
NVDD3	B14	SAP_FS	B							PC16	PUEN						Pull-H	PC16
NVDD3	C14	TIN	I							PC15	PUEN				$\overline{\text{WKGD}}$		Pull-H	PC15
NVDD3	A14	TOUT	O							PC14	PUEN	SYS_CLK	SYS_CLK				Pull-H	PC14
NVDD3	D14	USBG_RXDP	I							PC13	PUEN	SLCDC1_DAT15					Pull-H	PC13
NVDD3	C13	USBG_RXDM	I							PC12	PUEN	SLCDC1_DAT14					Pull-H	PC12
NVDD3	G13	USBG_TXDP	O							PC11	PUEN	SLCDC1_DAT13					Pull-H	PC11

Table 2-2. i.MX21 Signal Multiplexing Scheme (continued)
(*Note: See end of table for table footnotes and table legend*)

I/O Supply Voltage	BGA Pins	Primary				Alternate				GPIO							Reset (At/After)	Default
		Signal	Dir	PU	OD	Signal	Dir	PU	OD	Mux	PU	AIN	BIN	CIN	AOUT	BOUT		
NVDD3	B13	USBG_TXDM	O							PC10	PUEN	SLCDC1_DAT12					Pull-H	PC10
NVDD3	H13	$\overline{\text{USBG\_OE}}$	O							PC9	PUEN	SLCDC1_DAT11					Pull-H	PC9
NVSS3	M9	NVSS3																NVSS3
NVDD3	L8	NVDD3																NVDD3
NVDD3	A13	USBG_FS	O							PC8	PUEN	SLCDC1_DAT10			$\overline{\text{USBG\_TXR\_INT}}$		Pull-H	PC8
NVDD3	D13	$\overline{\text{USBG\_ON}}$	O							PC7	PUEN	SLCDC1_DAT9					Pull-H	PC7
NVDD3	B12	USBG_SCL	B		OD					PC6	PUEN	SLCDC1_DAT8					Pull-H	PC6
NVDD3	G12	USBG_SDA	B		OD					PC5	PUEN	SLCDC1_DAT7					Pull-H	PC5
NVDD3	D12	USBH1_RXDP	I			UART4_RXD	I			PB31	PDEN	SLCDC1_DAT6			$\overline{\text{UART4\_RTS}}$		Pull-L	PB31
NVDD3	C12	USBH1_RXDM	I							PB30	PDEN	SLCDC1_DAT5		$\overline{\text{UART4\_CTS}}$			Pull-L	PB30
NVDD3	H12	USBH1_TXDP	O			$\overline{\text{UART4\_CTS}}$	O			PB29	PDEN	SLCDC1_DAT4			UART4_RXD		Pull-L	PB29
NVDD3	D11	USBH1_TXDM	O			UART4_TXD	O			PB28	PDEN	SLCDC1_DAT3					Pull-L	PB28
NVDD3	A12	$\overline{\text{USBH1\_OE}}$	B							PB27	PUEN	SLCDC1_DAT2					Pull-H	PB27
NVDD3	A11	USBH1_FS	O			$\overline{\text{UART4\_RTS}}$	I			PB26	PDEN	SLCDC1_DAT1		USBH1_RXDAT			Pull-L	PB26
NVDD3	G11	$\overline{\text{USBH\_ON}}$	O							PB25	PUEN	SLCDC1_DAT0					Pull-H	PB25
NVDD3	C11	$\overline{\text{USB\_OC}}$	I							PB24	PUEN						Pull-H	PB24
NVDD3	B11	USB_PWR	O							PB23	PUEN						Pull-L	USB_PWR
NVDD3	G10	$\overline{\text{USB\_BYP}}$	I							PB22	PUEN						Pull-H	$\overline{\text{USB\_BYP}}$
NVDD4	J9	NVDD4																NVDD4
NVDD4	C10	CSI_HSYNC	I							PB21	PUEN						Pull-H	PB21
NVDD4	A10	CSI_VSYNC	I							PB20	PUEN						Pull-H	PB20
NVDD4	D10	CSI_D7	I							PB19	PUEN						Pull-H	PB19
NVDD4	B10	CSI_D6	I							PB18	PUEN						Pull-H	PB18
NVDD4	H9	CSI_D5	I							PB17	PUEN						Pull-H	PB17
NVDD4	A9	CSI_PIXCLK	I							PB16	PUEN						Pull-H	PB16
NVDD4	C9	CSI_MCLK	O							PB15	PUEN						Pull-H	PB15
NVDD4	B9	CSI_D4	I							PB14	PUEN						Pull-H	PB14
NVDD4	G9	CSI_D3	I							PB13	PUEN						Pull-H	PB13
NVDD4	D9	CSI_D2	I							PB12	PUEN						Pull-H	PB12
NVDD4	C8	CSI_D1	I							PB11	PUEN						Pull-H	PB11
NVDD4	A8	CSI_D0	I							PB10	PUEN						Pull-H	PB10
NVSS4	K8	NVSS4																NVSS4

Table 2-2. i.MX21 Signal Multiplexing Scheme (continued)
(*Note: See end of table for table footnotes and table legend*)

I/O Supply Voltage	BGA Pins	Primary				Alternate				GPIO							Reset (At/After)	Default
		Signal	Dir	PU	OD	Signal	Dir	PU	OD	Mux	PU	AIN	BIN	CIN	AOUT	BOUT		
QVDDX	K9	QVDDX																QVDDX
QVSS	M10	QVSS																QVSS
NVDD5	J8	NVDD5																NVDD5
NVDD5	D8	SD2_CLK	O							PB9	PDEN	SLCDC1_CLK					Pull-L	PB9
NVDD5	B8	SD2_CMD	B							PB8	PUEN	SLCDC1_CS					Pull-H	PB8
NVDD5	C7	SD2_D3	B							PB7	PUEN	SLCDC1_RS					Pull-H	PB7
NVDD5	A7	SD2_D2	B							PB6	PUEN	SLCDC1_D0					Pull-H	PB6
NVDD5	D7	SD2_D1	B							PB5	PUEN						Pull-H	PB5
NVDD5	B7	SD2_D0	B							PB4	PUEN						Pull-H	PB4
NVSS5	H8	NVSS5																NVSS5
NVDD6	A6	OE_ACD	O							PA31	PUEN						Pull-H	PA31
NVDD6	B6	CONTRAST	O			$\overline{\text{BMI\_READ}}$	O			PA30	PUEN						Pull-H	PA30
NVDD6	C6	VSYNC	O			BMI_RXF_FULL	O			PA29	PUEN				$\overline{\text{BMI\_WAIT}}$		Pull-H	PA29
NVDD6	A5	HSYNC	O							PA28	PUEN						Pull-H	PA28
NVDD6	E4	SPL_SPR	O							PA27	PDEN	SLCDC1_CLK					Pull-L	PA27
NVDD6	B5	PS	O							PA26	PDEN	SLCDC1_CS					Pull-L	PA26
NVDD6	D4	CLS	O							PA25	PDEN	SLCDC1_RS					Pull-L	PA25
NVDD6	A4	REV	O							PA24	PDEN	SLCDC1_D0					Pull-L	PA24
NVDD6	C5	LD17	O			$\overline{\text{BMI\_WRITE}}$	B			PA23	PUEN						Pull-H	PA23
NVDD6	B4	LD16	O			BMI_READ_REQ	O			PA22	PUEN	$\overline{\text{EXT\_DMAGRANT}}$					Pull-H	PA22
NVDD6	E3	LD15	O			BMI_D15	B			PA21	PUEN	SLCDC1_DAT15		SLCDC1_DAT7			Pull-H	PA21
NVDD6	A3	LD14	O			BMI_D14	B			PA20	PUEN	SLCDC1_DAT14		SLCDC1_DAT6			Pull-H	PA20
NVDD6	D3	LD13	O			BMI_D13	B			PA19	PUEN	SLCDC1_DAT13		SLCDC1_DAT5			Pull-H	PA19
NVDD6	A2	LD12	O			BMI_D12	B			PA18	PUEN	SLCDC1_DAT12		SLCDC1_DAT4			Pull-H	PA18
NVSS6	G8	NVSS6																NVSS6
NVDD6	G7	NVDD6																NVDD6
NVDD6	B3	LD11	O			BMI_D11	B			PA17	PUEN	SLCDC1_DAT11		SLCDC1_DAT3			Pull-H	PA17
NVDD6	C4	LD10	O			BMI_D10	B			PA16	PUEN	SLCDC1_DAT10		SLCDC1_DAT2			Pull-H	PA16

Table 2-2. i.MX21 Signal Multiplexing Scheme (continued)
(*Note: See end of table for table footnotes and table legend*)

I/O Supply Voltage	BGA Pins	Primary				Alternate				GPIO							Reset (At/After)	Default
		Signal	Dir	PU	OD	Signal	Dir	PU	OD	Mux	PU	AIN	BIN	CIN	AOUT	BOUT		
NVDD6	A1	LD9	O			BMI_D9	B			PA15	PUEN	SLCDC1_DAT9		SLCDC1_DAT1			Pull-H	PA15
NVDD6	E1	LD8	O			BMI_D8	B			PA14	PUEN	SLCDC1_DAT8		SLCDC1_DAT0			Pull-H	PA14
NVDD6	B1	LD7	O			BMI_D7	B			PA13	PUEN	SLCDC1_DAT7					Pull-H	PA13
NVDD6	C3	LD6	O			BMI_D6	B			PA12	PUEN	SLCDC1_DAT6					Pull-H	PA12
NVDD6	B2	LD5	O			BMI_D5	B			PA11	PUEN	SLCDC1_DAT5					Pull-H	PA11
NVDD6	E2	LD4	O			BMI_D4	B			PA10	PUEN	SLCDC1_DAT4					Pull-H	PA10
NVDD6	C2	LD3	O			BMI_D3	B			PA9	PUEN	SLCDC1_DAT3					Pull-H	PA9
NVDD6	D1	LD2	O			BMI_D2	B			PA8	PUEN	SLCDC1_DAT2					Pull-H	PA8
NVDD6	C1	LD1	O			BMI_D1	B			PA7	PUEN	SLCDC1_DAT1					Pull-H	PA7
NVDD6	D2	LD0	O			BMI_D0	B			PA6	PUEN	SLCDC1_DAT0					Pull-H	PA6
NVDD6	F4	LSCLK	O			BMI_CLK_CS	B			PA5	PUEN						Pull-H	PA5

KP- Keeper Circuit permanently On when in Primary / Alternate Mode.

PU-Pull Up permanently On when in Primary / Alternate Mode.

PUEN-Pull Up controllable from Module when in Primary / Alternate Mode.

OD-Open Drain permanently On when in Primary / Alternate Mode.

ODEN-Open Drain controllable from Module when in Primary / Alternate Mode.

1. Data Bus resets to a keeper circuit (state retention circuit) and input.
2. At POR, SDCLK is low then toggles for approximately 300ms during chip reset, then idles at logic high.
3. The default for this signal is as an input and is pulled high internally. To enable the GPIO PF16 function, the user must set bit 16 in the GPIO In Use Register Port F. However, if this signal is not used, then it may be left as a no connect.
4. Refer to section 2.4 for recommended termination of these signals.
5. Reset-in and POR need to be connected to reset circuitry.
6. Boot signals should be connected to logic high or low depending on customer applications. Boot 3 should always be tied low.

2.3 Power-Up Sequence

The i.MX21 processor consists of three major sets for power supply voltage named QVDD (core logic supply), VDDA (analog supply), and NVDD (IO supply). The External Voltage Regulators and power-on devices must provide the applications processor with a specific sequence of power and resets to ensure proper operation.

It is important that the applications processor power supplies be powered-up in a certain order to avoid high current situations. The required order is:

1. NVDD (1.8/3.0V) and VDDA (formally AVDD) (3.0V)
2. QVDD (1.5V) and QVDDX (1.5V)

2.4 Package Information

Table 2-3 identifies the pin assignments for the ball grid array (BGA). The connections of these pins depend solely upon the user application, however there are a few factory test signals that are not used in a normal application. Following is a list of these signals and how they are to be terminated for proper operation of the i.MX21 processor:

- CLKMODE[1:0]: To ensure proper operation, leave these signals as no connects.
- OSC26M_TEST: To ensure proper operation, leave this signal as no connect.
- EXT_48M: To ensure proper operation, connect this signal to ground.
- EXT_266M: To ensure proper operation, connect this signal to ground.
- TEST_WB[2:0]: These signals are also multiplexed with GPIO PORT E as well as alternate keypad signals. If not utilizing these signals for GPIO functionality or for their other multiplexed function, then configure as GPIO input with pull up enabled, and leave as a no connect.
- TEST_WB[4:3]: To ensure proper operation, leave these signals as no connects.

Most of the signals shown in Table 2-3 are multiplexed with other signals. For simplicity only the primary signal names are shown. Please refer to Table 2-2 for complete information on the signal multiplexing schemes of these signals.

Table 2-3. i.MX21 Pin Assignment

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
A	LD9	LD12	LD14	REV	HSYNC	OE_ ACD	SD2_D2	CSI_ D0	CSI_ PIXCLK	CSI_ VSYNC	USBH1_ FS	USBH1_ OE	USBG_ FS	TOUT	SAP_ TXDAT	SSI1_ CLK	SSI2_ RXDAT	SSI2_TXDAT	SSI3_ FS
B	LD7	LD5	LD11	LD16	PS	CON TRAST	SD2_D0	SD2_ CMD	CSI_ D4	CSI_D6	USB_ PWR	USBG_ SCL	USBG_ TXDM	SAP_ FS	SSI1_ FS	SSI2_ FS	SSI3_ TXDAT	I2C_DATA	CSPI2_ SS2
C	LD1	LD3	LD6	LD10	LD17	VSYNC	SD2_D3	CSI_ D1	CSI_ MCLK	CSI_ HSYNC	USB_ OC	USBH1_ RXDM	USBG_ RXDM	TIN	SSI1_ TXDAT	SSI3_ RXDAT	SSI3_ CLK	I2C_CLK	CSPI2_ SS1
D	LD2	LD0	LD13	CLS	QVDD	QVSS	SD2_D1	SD2_ CLK	CSI_ D2	CSI_D7	USBH1_ TXDM	USBH1_ RXDP	USBG_ ON	USBG_ RXDP	SAP_ RXDAT	SSI1_ RXDAT	SSI2_ CLK	CSPI2_SS0	CSPI2_ SCLK
E	LD8	LD4	LD15	SPL_ SPR												SAP_ CLK	CSPI2_ MISO	CSPI1_SS2	CSPI2_ MOSI
F	A24_ NFIO14	D31	A25_ NFIO15	LSCLK												CSPI1_ SS1	CSPI1_ MISO	KP_ROW0	CSPI1_ SS0
G	A22_ NFIO12	D29	A23_ NFIO13	D30			NVDD6	NVSS6	CSI_D3	USB_ BYP	USBH_ ON	USBG_ SDA	USBG_ TXDP			KP_ ROW1	KP_ ROW3	UART2_CTS	KP_ ROW4
H	A20	D27	A21_ NFIO11	D28			NVDD1	NVSS5	CSI_D5	CSPI1_ SCLK	CSPI1_ RDY	USBH1_ TXDP	USBG_ OE			TEST_ WB4	TEST_ WB2	TEST_WB3	PWMO
J	A19	A18	D25	D26			NVDD1	NVDD5	NVDD4	KP_ ROW5	KP_ ROW2	CSPI1_ MOSI	TEST_ WB0			UART2_ RTS	KP_COL1	KP_COL0	TEST_ WB1
K	A16	A17	D23	D24			NVSS1	NVSS4	QVDDX	UART1_ RXD	TDO	QVDD	QVSS			KP_ COL3	KP_COL5	KP_COL4	KP_ COL2
L	A14_ NFIO9	A15_ NFIO10	D21	D22			NVSS1	NVDD3	QVDD	QVSS	NFIO2	NFWP	UART1_ TXD			UART2_ TXD	UART3_ RTS	UART3_CTS	UART3_ TXD
M	D19	A13_ NFIO8	D20	D18			NVDD2	NVDD3	NVSS3	QVSS	NFIO7	NFRB	EXT_ 48M			UART2_ RXD	UART3_ RXD	UART1_RTS	UART1_ CTS
N	A11	A12	D17	D16			LBA	NVSS3	SDCKE0	NVSS1	NVSS1	NVDD1	NVDD1			SD1_ D0	TCK	SD1_D1	RTCK
P	A9	A10	D15	D14												SD1_ D2	SD1_ CMD	TDI	TMS
R	A7	A8	D13	D12												SD1_ CLK	EXT_ 266M	NVSS2	TRST
T	A5	A6	EB3	D10	CS3	CS1	BCLK	MA11	RAS	CAS	NFIO5	NFIO3	NFWE	RESET_ IN	NFCE	BOOT1	SD1_D3	CLKMODE1	CLK MODE0
U	D11	EB1	EB2	OE	CS4	D6	ECB	D3	MA10	PC_ PWRON	PF16	NFIO4	NFIO1	NFALE	NFCLE	POR	BOOT2	BOOT3	XTAL32K
V	A4	EB0	D9	D8	CS5	D5	CS0	RW	D1	JTAG_ CTRL	SDWE	CLKO	NFIO6	QVSS	RESET_ OUT	BOOT0	OSC26M_ TEST	VDDA	EXTAL 32K
W	A3	A2	D7	A1	CS2	A0	D4	D2	D0	SDCLK	SDCKE1	NFIO0	NFRE	QVDD	QVSS	EXTAL 26M	XTAL26M	QVDD	QVSS

Chapter 3
Memory Map

This chapter describes the memory maps and the chip configuration registers of the i.MX21 processor.

3.1 Memory Space

Figure 3-1 on page 3-2 shows a detailed block diagram of the interconnection of the various modules in i.MX21. The i.MX21 with a 32-bit address bus is capable of addressing a 4 Gbyte physical address space. This space is divided into sections of 512 Mbyte regions within which various memories and peripherals are mapped.

Table 3-1 shows a simplified breakdown of the eight 512 Mbyte regions decoded within the 4 Gbyte address space.

Table 3-1. 4 Gbyte Memory Map Breakdown

Address	Size	Usage
0x00000000	512 Mbyte	ROM, Primary AHB Slaves, and Peripherals
0x20000000	512 Mbyte	Reserved
0x40000000	512 Mbyte	Reserved
0x60000000	512 Mbyte	Reserved
0x80000000	512 Mbyte	Secondary AHB Slave Port 1
0xA0000000	512 Mbyte	Secondary AHB Slave Port 2
0xC0000000	512 Mbyte	Secondary AHB Slave Port 3
0xE0000000	512 Mbyte	Primary AHB (RAM)

3.1.1 Detailed Memory Map

Figure 3-2 on page 3-3 shows the memory space breakout view for i.MX21. The left-most column shows the eight 512 Mbyte regions. The middle column shows the breakout of primary and secondary AHB slaves and the right-most column shows the breakout of the AIPI1 and AIPI2 address spaces.

Table 3-2 to Table 3-6 show the detailed breakdown of the complete memory map according to the 512 Mbyte regions. Table 3-7 and Table 3-8 show the detailed breakdown of the AIPI1 and AIPI2 modules and the different IP peripherals accessed over the AIPI1 and AIPI2.

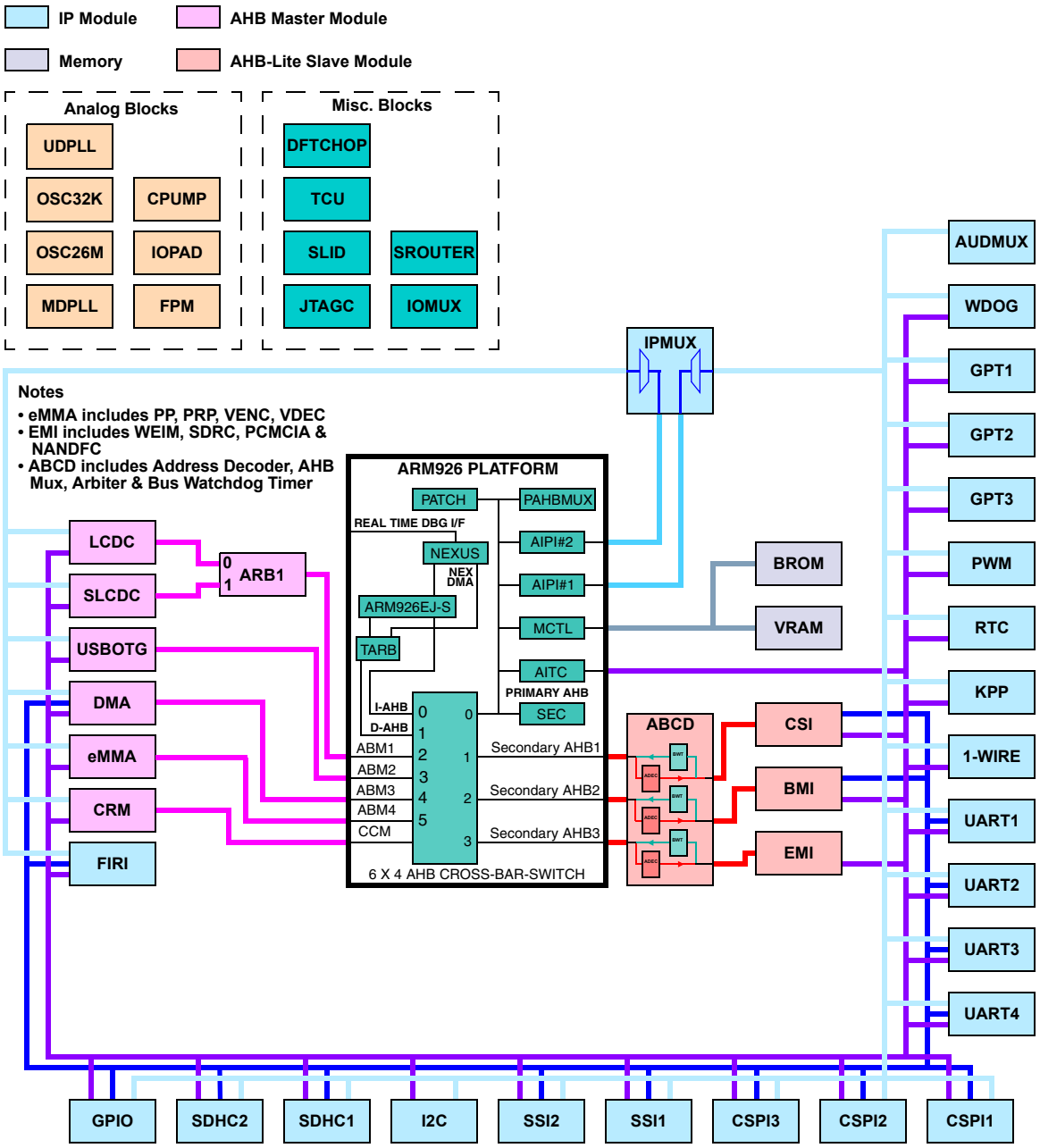


Figure 3-1. Memory Space Breakout View for i.MX21

Accesses to locations defined as Reserved (other than aliased RAM space) results in an AHB error response. Accesses to unimplemented locations within the AITC and ROMPATCH register space will be terminated and write accesses will have no effect and read accesses will return all zeros.

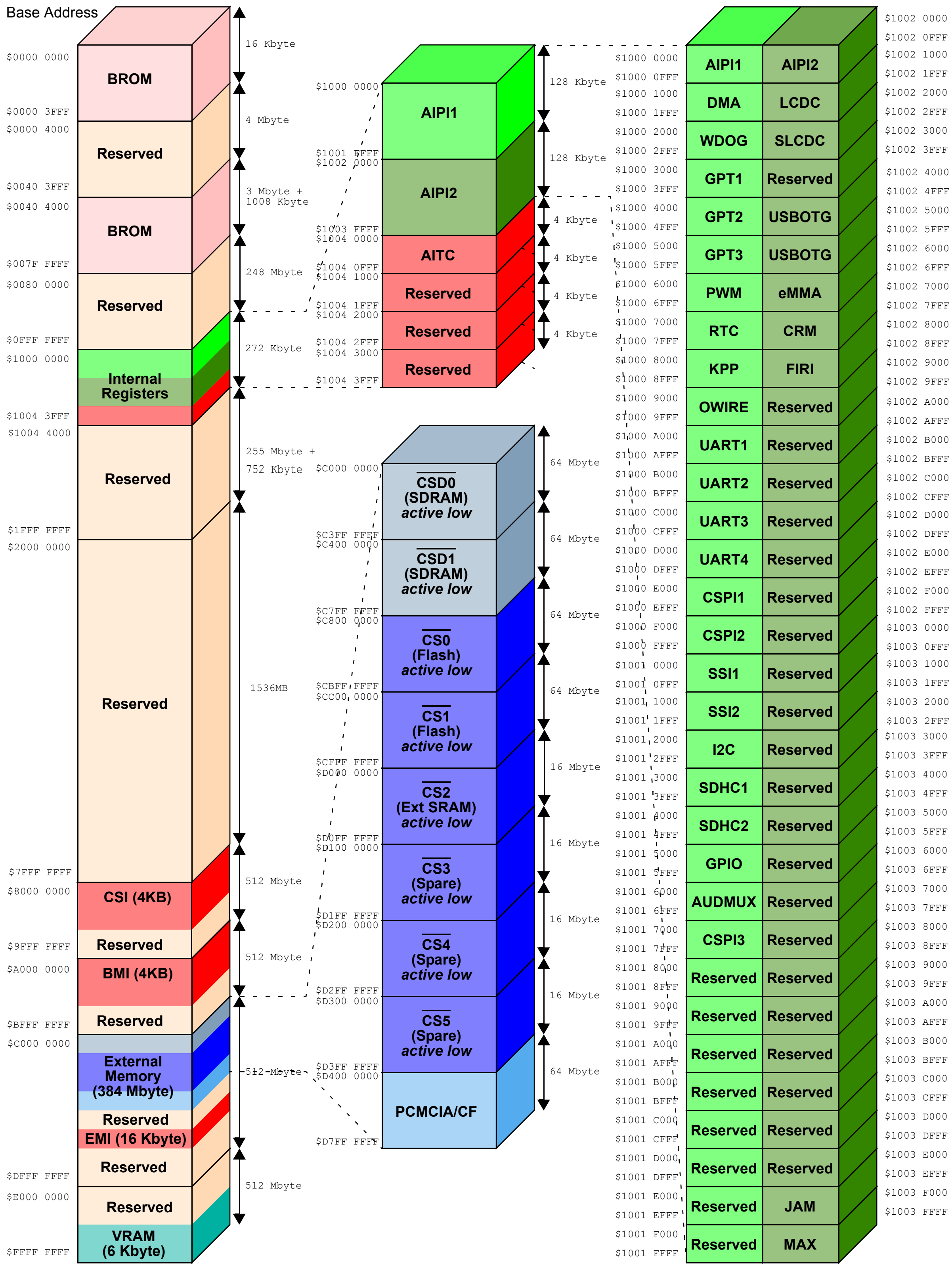


Figure 3-2. i.MX21 Physical Memory Map (4 Gbyte)

Table 3-2. Primary AHB Memory Map (Lower)

Address	Secondary AHB Slave Port 1	Size
0x0000 0000 – 0x0000 3FFF	BROM	16 Kbyte
0x0000 4000 – 0x0040 3FFF	Reserved	4 Mbyte
0x0040 4000 – 0x0040 5FFF	BROM	8 Kbyte
0x0040 6000 – 0x007F FFFF	BROM (Hole)	3MB + 1000 Kbyte
0x0080 0000 – 0x0FFF FFFF	Reserved	248 Mbyte
0x1000 0000 – 0x1001 FFFF	AIPI1	128 Kbyte
0x1002 0000 – 0x1003 FFFF	AIPI2	128 Kbyte
0x1004 0000 – 0x1004 0FFF	AITC	4 Kbyte
0x1004 1000 – 0x1004 1FFF	Reserved	4 Kbyte
0x1004 2000 – 0x1004 2FFF	Reserved	4 Kbyte
0x1004 3000 – 0x1004 3FFF	Reserved	4 Kbyte
0x1004 4000 – 0x7FFF FFFF	Reserved	255 Mbyte + 752 Kbyte

Table 3-2 shows the memory map of the Primary AHB address space in the first 512 Mbyte region. The BROM is split into two sections of 16 Kbyte and 8 Kbyte. The BROM (Hole) indicates that there is no BROM code present in this region. The AIPI1 and AIPI2 address space contains AIPI control registers and the IP slave registers. The AIPI1 and AIPI2 maps are shown in Table 3-7 and Table 3-8, respectively.

Table 3-3. Secondary AHB Port 1 Memory Map

Address	Secondary AHB Port 1	Size
0x8000 0000 – 0x8000 0FFF	CSI	4 Kbyte
0x8000 1000 – 0x9FFF FFFF	Reserved	511 Mbyte + 1020 Kbyte

Table 3-3 and Table 3-4 show the CSI and BMI modules connected to the Secondary AHB Ports 1 and 2 respectively. The CSI and BMI do not have an IP interface and are accessed directly over the AHB.

Table 3-4. Secondary AHB Port 2 Memory Map

Address	Secondary AHB Slave Port 2	Size
0xA000 0000 – 0xA000 0FFF	BMI	4 Kbyte
0xA000 1000 – 0xBFFF FFFF	Reserved	511 Mbyte + 1020 Kbyte

Table 3-5 shows the memory map breakdown for the Secondary AHB Port 3. The SDRAMC, WEIM, PCMCIA and NANDFC module control registers and external memory are addressed via this region. The external memory regions (memory or external peripherals) are accessed via the respective chip selects.

$\overline{\text{CSD1}}$ and $\overline{\text{CSD0}}$ are SDRAMC chip selects and $\overline{\text{CS5}}$ to $\overline{\text{CS0}}$ WEIM Chip Selects. $\overline{\text{CSD1}}$ and $\overline{\text{CS0}}$ chip select spaces are available for external boot.

0xD4000 0000 to 0xD7FF_FFFF is allocated for PCMCIA IO and Memory Space.

Table 3-5. Secondary AHB Port 3 Memory Map

Address	Secondary AHB Port 3	Size
0xC000 0000 – 0xC3FF FFFF	External Memory ($\overline{CSD0}$)	64 Mbyte
0xC400 0000 – 0xC7FF FFFF	External Memory ($\overline{CSD1}$)	64 Mbyte
0xC800 0000 – 0xCBFF FFFF	External Memory ($\overline{CS0}$)	64 Mbyte
0xCC00 0000 – 0xCFFF FFFF	External Memory ($\overline{CS1}$)	64 Mbyte
0xD000 0000 – 0xD0FF FFFF	External Memory ($\overline{CS2}$)	16 Mbyte
0xD100 0000 – 0xD1FF FFFF	External Memory ($\overline{CS3}$)	16 Mbyte
0xD200 0000 – 0xD2FF FFFF	External Memory ($\overline{CS4}$)	16 Mbyte
0xD300 0000 – 0xD3FF FFFF	External Memory ($\overline{CS5}$)	16 Mbyte
0xD400 0000 – 0xD7FF FFFF	PCMCIA/CF IO and Memory Space	64 Mbyte
0xD800 0000 – 0xDEFF FFFF	Reserved	112 Mbyte
0xDF00 0000 – 0xDF00 0FFF	SDRAMC	4 Kbyte
0xDF00 1000 – 0xDF00 1FFF	EIM	4 Kbyte
0xDF00 2000 – 0xDF00 2FFF	PCMCIA	4 Kbyte
0xDF00 3000 – 0xDF00 3FFF	NANDFC	4 Kbyte
0xDF00 4000 – 0xDFFF FFFF	Reserved	15 Mbyte + 1008 Kbyte

Table 3-6 shows the last region of address space that is part of the Primary AHB Memory Map. The Vector-RAM is mapped into this region and i.MX21 uses the high memory (0xFFFF FF00 to 0xFFFF FFFF) to store the interrupt vector table (64 words). This region is aliased on a 128 Kbyte boundary.

Table 3-6. Primary AHB Memory Map (Upper)

Address	Primary AHB	Size
0xE000 0000 – 0xFFFF E7FF	Reserved (aliased)	511 Mbyte + 1018 Kbyte
0xFFFF E800 – 0xFFFF FFFF	VRAM	6 Kbyte

The next two tables, Table 3-7 and Table 3-8 show the detailed break down of the address space controlled by AIPI1 and AIPI2. More details on the AIPI can be found in the AIPI chapter.

Table 3-7. AIPI1 Memory Map

Address	AIPI1 Memory Map	Size
0x1000 0000 – 0x1000 0FFF	AIPI1 (Slot 0)	4 Kbyte
0x1000 1000 – 0x1000 1FFF	DMA	4 Kbyte
0x1000 2000 – 0x1000 2FFF	WDOG	4 Kbyte
0x1000 3000 – 0x1000 3FFF	GPT1	4 Kbyte
0x1000 4000 – 0x1000 4FFF	GPT2	4 Kbyte
0x1000 5000 – 0x1000 5FFF	GPT3	4 Kbyte

Table 3-7. AIPI1 Memory Map (continued)

Address	AIPI1 Memory Map	Size
0x1000 6000 – 0x1000 6FFF	PWM	4 Kbyte
0x1000 7000 – 0x1000 7FFF	RTC	4 Kbyte
0x1000 8000 – 0x1000 8FFF	KPP	4 Kbyte
0x1000 9000 – 0x1000 9FFF	OWIRE	4 Kbyte
0x1000 A000 – 0x1000 AFFF	UART1	4 Kbyte
0x1000 B000 – 0x1000 BFFF	UART2	4 Kbyte
0x1000 C000 – 0x1000 CFFF	UART3	4 Kbyte
0x1000 D000 – 0x1000 DFFF	UART4	4 Kbyte
0x1000 E000 – 0x1000 EFFF	CSPI1	4 Kbyte
0x1000 F000 – 0x1000 FFFF	CSPI2	4 Kbyte
0x1001 0000 – 0x1001 0FFF	SSI1	4 Kbyte
0x1001 1000 – 0x1001 1FFF	SSI2	4 Kbyte
0x1001 2000 – 0x1001 2FFF	I2C	4 Kbyte
0x1001 3000 – 0x1001 3FFF	SDHC1	4 Kbyte
0x1001 4000 – 0x1001 4FFF	SDHC2	4 Kbyte
0x1001 5000 – 0x1001 5FFF	GPIO	4 Kbyte
0x1001 6000 – 0x1001 6FFF	AUDMUX	4 Kbyte
0x1001 7000 – 0x1001 7FFF	CSPI3	4 Kbyte
0x1001 7000 – 0x1001 FFFF	Reserved (Slot 24–31)	8 × 4 Kbyte (32 Kbyte)

Table 3-8. AIPI2 Memory Map

Address	AIPI2 Memory Map	Size
0x1002 0000 – 0x1002 0FFF	AIPI2 (Slot 0)	4 Kbyte
0x1002 1000 – 0x1002 1FFF	LCDC	4 Kbyte
0x1002 2000 – 0x1002 2FFF	SLCDC	4 Kbyte
0x1002 3000 – 0x1002 3FFF	Reserved	4 Kbyte
0x1002 4000 – 0x1002 4FFF	USB OTG	4 Kbyte
0x1002 5000 – 0x1002 5FFF	USB OTG	4 Kbyte
0x1002 6000 – 0x1002 6FFF	eMMA	4 Kbyte
0x1002 7000 – 0x1002 7FFF	CRM	4 Kbyte
0x1002 8000 – 0x1002 8FFF	FIRI	4 Kbyte
0x1002 B000 – 0x1002 DFFF	Reserved (Slots 11–29)	76 Kbyte

Table 3-8. AIPI2 Memory Map (continued)

Address	AIPI2 Memory Map	Size
0x1003 E000 – 0x1003 EFFF	JAM	4 Kbyte
0x1003 F000 – 0x1003 FFFF	MAX	4 Kbyte

3.2 Register Map

The internal registers in the i.MX21 are listed in Table 3-9.

Table 3-9. Register Map

Module Name	Address	Register Name	Description
AIPI1	0x1000 0000	PSR0	Peripheral Size Register0
AIPI1	0x1000 0004	PSR1	Peripheral Size Register1
AIPI1	0x1000 0008	PAR	Peripheral Access Register
DMAC	0x1000 1000	DCR	DMA Control Register
DMAC	0x1000 1004	DISR	DMA Interrupt Status Register
DMAC	0x1000 1008	DIMR	DMA Interrupt Mask Register
DMAC	0x1000 100C	DBTOSR	DMA Burst Time-Out Status Register
DMAC	0x1000 1010	DRTOSR	DMA Request Time-Out Status Register
DMAC	0x1000 1014	DSESR	DMA Transfer Error Status Register
DMAC	0x1000 1018	DBOSR	DMA Buffer Overflow Status Register
DMAC	0x1000 101C	DBTOCR	DMA Burst Time-Out Control Register
DMAC	0x1000 1040	WSRA	W-Size Register A
DMAC	0x1000 1044	XSRA	X-Size Register A
DMAC	0x1000 1048	YSRA	Y-Size Register A
DMAC	0x1000 104C	WSRB	W-Size Register B
DMAC	0x1000 1050	XSRB	X-Size Register B
DMAC	0x1000 1054	YSRB	Y-Size Register B
DMAC	0x1000 1080	SAR0	Channel 0 Source Address Register
DMAC	0x1000 1084	DAR0	Channel 0 Destination Address Register
DMAC	0x1000 1088	CNTR0	Channel 0 Count Register
DMAC	0x1000 108C	CCR0	Channel 0 Control Register
DMAC	0x1000 1090	RSSR0	Channel 0 Request Source Select Register
DMAC	0x1000 1094	BLR0	Channel 0 Burst Length Register
DMAC	0x1000 1098	RTOR0 BUCR0	Channel 0 Request Time-Out Register Channel 0 Bus Utilization Control Register
DMAC	0x1000 109C	CCNR0	Channel 0 Channel Counter Register

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
DMAC	0x1000 10C0	SAR1	Channel 1 Source Address Register
DMAC	0x1000 10C4	DAR1	Channel 1 Destination Address Register
DMAC	0x1000 10C8	CNTR1	Channel 1 Count Register
DMAC	0x1000 10CC	CCR1	Channel 1 Control Register
DMAC	0x1000 10D0	RSSR1	Channel 1 Request Source Select Register
DMAC	0x1000 10D4	BLR1	Channel 1 Burst Length Register
DMAC	0x1000 10D8	RTOR1 BUCR1	Channel 1 Request Time-Out Register Channel 1 Bus Utilization Control Register
DMAC	0x1000 10DC	CCNR1	Channel 1 Channel Counter Register
DMAC	0x1000 1100	SAR2	Channel 2 Source Address Register
DMAC	0x1000 1104	DAR2	Channel 2 Destination Address Register
DMAC	0x1000 1108	CNTR2	Channel 2 Count Register
DMAC	0x1000 110C	CCR2	Channel 2 Control Register
DMAC	0x1000 1110	RSSR2	Channel 2 Request Source Select Register
DMAC	0x1000 1114	BLR2	Channel 2 Burst Length Register
DMAC	0x1000 1118	RTOR2 BUCR2	Channel 2 Request Time-Out Register Channel 2 Bus Utilization Control Register
DMAC	0x1000 111C	CCNR2	Channel 2 Channel Counter Register
DMAC	0x1000 1140	SAR3	Channel 3 Source Address Register
DMAC	0x1000 1144	DAR3	Channel 3 Destination Address Register
DMAC	0x1000 1148	CNTR3	Channel 3 Count Register
DMAC	0x1000 114C	CCR3	Channel 3 Control Register
DMAC	0x1000 1150	RSSR3	Channel 3 Request Source Select Register
DMAC	0x1000 1154	BLR3	Channel 3 Burst Length Register
DMAC	0x1000 1158	RTOR3 BUCR3	Channel 3 Request Time-Out Register Channel 3 Bus Utilization Control Register
DMAC	0x1000 115C	CCNR3	Channel 3 Channel Counter Register
DMAC	0x1000 1180	SAR4	Channel 4 Source Address Register
DMAC	0x1000 1184	DAR4	Channel 4 Destination Address Register
DMAC	0x1000 1188	CNTR4	Channel 4 Count Register
DMAC	0x1000 118C	CCR4	Channel 4 Control Register
DMAC	0x1000 1190	RSSR4	Channel 4 Request Source Select Register
DMAC	0x1000 1194	BLR4	Channel 4 Burst Length Register

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
DMAC	0x1000 1198	RTOR4 BUCR4	Channel 4 Request Time-Out Register Channel 4 Bus Utilization Control Register
DMAC	0x1000 119C	CCNR 4	Channel 4 Channel Counter Register
DMAC	0x1000 11C0	SAR5	Channel 5 Source Address Register
DMAC	0x1000 11C4	DAR5	Channel 5 Destination Address Register
DMAC	0x1000 11C8	CNTR5	Channel 5 Count Register
DMAC	0x1000 11CC	CCR5	Channel 5 Control Register
DMAC	0x1000 11D0	RSSR5	Channel 5 Request Source Select Register
DMAC	0x1000 11D4	BLR5	Channel 5 Burst Length Register
DMAC	0x1000 11D8	RTOR5 BUCR5	Channel 5 Request Time-Out Register Channel 5 Bus Utilization Control Register
DMAC	0x1000 11DC	CCNR5	Channel 5 Channel Counter Register
DMAC	0x1000 1200	SAR6	Channel 6 Source Address Register
DMAC	0x1000 1204	DAR6	Channel 6 Destination Address Register
DMAC	0x1000 1208	CNTR6	Channel 6 Count Register
DMAC	0x1000 120C	CCR6	Channel 6 Control Register
DMAC	0x1000 1210	RSSR6	Channel 6 Request Source Select Register
DMAC	0x1000 1214	BLR6	Channel 6 Burst Length Register
DMAC	0x1000 1218	RTOR6 BUCR6	Channel 6 Request Time-Out Register Channel 6 Bus Utilization Control Register
DMAC	0x1000 121C	CCNR6	Channel 6 Channel Counter Register
DMAC	0x1000 1240	SAR7	Channel 7 Source Address Register
DMAC	0x1000 1244	DAR7	Channel 7 Destination Address Register
DMAC	0x1000 1248	CNTR7	Channel 7 Count Register
DMAC	0x1000 124C	CCR7	Channel 7 Control Register
DMAC	0x1000 1250	RSSR7	Channel 7 Request Source Select Register
DMAC	0x1000 1254	BLR7	Channel 7 Burst Length Register
DMAC	0x1000 1258	RTOR7 BUCR7	Channel 7 Request Time-Out Register Channel 7 Bus Utilization Control Register
DMAC	0x1000 125C	CCNR7	Channel 7 Channel Counter Register
DMAC	0x1000 1280	SAR8	Channel 8 Source Address Register
DMAC	0x1000 1284	DAR8	Channel 8 Destination Address Register
DMAC	0x1000 1288	CNTR8	Channel 8 Count Register
DMAC	0x1000 128C	CCR8	Channel 8 Control Register

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
DMAC	0x1000 1290	RSSR8	Channel 8 Request Source Select Register
DMAC	0x1000 1294	BLR8	Channel 8 Burst Length Register
DMAC	0x1000 1298	RTOR8 BUCR8	Channel 8 Request Time-Out Register Channel 8 Bus Utilization Control Register
DMAC	0x1000 129C	CCNR8	Channel 8 Channel Counter Register
DMAC	0x1000 12C0	SAR9	Channel 9 Source Address Register
DMAC	0x1000 12C4	DAR9	Channel 9 Destination Address Register
DMAC	0x1000 12C8	CNTR9	Channel 9 Count Register
DMAC	0x1000 12CC	CCR9	Channel 9 Control Register
DMAC	0x1000 12D0	RSSR9	Channel 9 Request Source Select Register
DMAC	0x1000 12D4	BLR9	Channel 9 Burst Length Register
DMAC	0x1000 12D8	RTOR9 BUCR9	Channel 9 Request Time-Out Register Channel 9 Bus Utilization Control Register
DMAC	0x1000 12DC	CCNR9	Channel 9 Channel Counter Register
DMAC	0x1000 1300	SAR10	Channel 10 Source Address Register
DMAC	0x1000 1304	DAR10	Channel 10 Destination Address Register
DMAC	0x1000 1308	CNTR10	Channel 10 Count Register
DMAC	0x1000 130C	CCR10	Channel 10 Control Register
DMAC	0x1000 1310	RSSR10	Channel 10 Request Source Select Register
DMAC	0x1000 1314	BLR10	Channel 10 Burst Length Register
DMAC	0x1000 1318	RTOR10 BUCR10	Channel 10 Request Time-Out Register Channel 10 Bus Utilization Control Register
DMAC	0x1000 131C	CCNR10	Channel 10 Channel Counter Register
DMAC	0x1000 1340	SAR11	Channel 11 Source Address Register
DMAC	0x1000 1344	DAR11	Channel 11 Destination Address Register
DMAC	0x1000 1348	CNTR11	Channel 11 Count Register
DMAC	0x1000 134C	CCR11	Channel 11 Control Register
DMAC	0x1000 1350	RSSR11	Channel 11 Request Source Select Register
DMAC	0x1000 1354	BLR11	Channel 11 Burst Length Register
DMAC	0x1000 1358	RTOR11 BUCR11	Channel 11 Request Time-Out Register Channel 11 Bus Utilization Control Register
DMAC	0x1000 135C	CCNR11	Channel 11 Channel Counter Register
DMAC	0x1000 1380	SAR12	Channel 12 Source Address Register
DMAC	0x1000 1384	DAR12	Channel 12 Destination Address Register

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
DMAC	0x1000 1388	CNTR12	Channel 12 Count Register
DMAC	0x1000 138C	CCR12	Channel 12 Control Register
DMAC	0x1000 1390	RSSR12	Channel 12 Request Source Select Register
DMAC	0x1000 1394	BLR12	Channel 12 Burst Length Register
DMAC	0x1000 1398	RTOR12 BUCR12	Channel 12 Request Time-Out Register Channel 12 Bus Utilization Control Register
DMAC	0x1000 139C	CCNR12	Channel 14 Channel Counter Register
DMAC	0x1000 13C0	SAR13	Channel 13 Source Address Register
DMAC	0x1000 13C4	DAR13	Channel 13 Destination Address Register
DMAC	0x1000 13C8	CNTR13	Channel 13 Count Register
DMAC	0x1000 13CC	CCR13	Channel 13 Control Register
DMAC	0x1000 13D0	RSSR13	Channel 13 Request Source Select Register
DMAC	0x1000 13D4	BLR13	Channel 13 Burst Length Register
DMAC	0x1000 13D8	RTOR13 BUCR13	Channel 13 Request Time-Out Register Channel 13 Bus Utilization Control Register
DMAC	0x1000 13DC	CCNR13	Channel 13 Channel Counter Register
DMAC	0x1000 1400	SAR14	Channel 14 Source Address Register
DMAC	0x1000 1404	DAR14	Channel 14 Destination Address Register
DMAC	0x1000 1408	CNTR14	Channel 14 Count Register
DMAC	0x1000 140C	CCR14	Channel 14 Control Register
DMAC	0x1000 1410	RSSR14	Channel 14 Request Source Select Register
DMAC	0x1000 1414	BLR14	Channel 14 Burst Length Register
DMAC	0x1000 1418	RTOR14 BUCR14	Channel 14 Request Time-Out Register Channel 14 Bus Utilization Control Register
DMAC	0x1000 141C	CCNR14	Channel 14 Channel Counter Register
DMAC	0x1000 1440	SAR15	Channel 15 Source Address Register
DMAC	0x1000 1444	DAR15	Channel 15 Destination Address Register
DMAC	0x1000 1448	CNTR15	Channel 15 Count Register
DMAC	0x1000 144C	CCR15	Channel 15 Control Register
DMAC	0x1000 1450	RSSR15	Channel 15 Request Source Select Register
DMAC	0x1000 1454	BLR15	Channel 15 Burst Length Register
DMAC	0x1000 1458	RTOR15 BUCR15	Channel 15 Request Time-Out Register Channel 15 Bus Utilization Control Register
DMAC	0x1000 145C	CCNR15	Channel 15 Channel Counter Register

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
WDOG	0x1000 2000	WCR	Watchdog Control Register
WDOG	0x1000 2002	WSR	Watchdog Service Register
WDOG	0x1000 2004	WRSR	Watchdog Reset Status Register
GPT1	0x1000 3000	TCTL1	GPT Control Register 1
GPT1	0x1000 3004	TPRER1	GPT Prescaler Register 1
GPT1	0x1000 3008	TCMP1	GPT Compare Register 1
GPT1	0x1000 300C	TCR1	GPT Capture Register 1
GPT1	0x1000 3010	TCN1	GPT Counter Register 1
GPT1	0x1000 3014	TSTAT1	GPT Status Register 1
GPT2	0x1000 4000	TCTL2	GPT Control Register 2
GPT2	0x1000 4004	TPRER2	GPT Prescaler Register 2
GPT2	0x1000 4008	TCMP2	GPT Compare Register 2
GPT2	0x1000 400C	TCR2	GPT Capture Register 2
GPT2	0x1000 4010	TCN2	GPT Counter Register 2
GPT2	0x1000 4014	TSTAT2	GPT Status Register 2
GPT3	0x1000 5000	TCTL3	GPT Control Register 3
GPT3	0x1000 5004	TPRER3	GPT Prescaler Register 3
GPT3	0x1000 5008	TCMP3	GPT Compare Register 3
GPT3	0x1000 500C	TCR3	GPT Capture Register 3
GPT3	0x1000 5010	TCN3	GPT Counter Register 3
GPT3	0x1000 5014	TSTAT3	GPT Status Register 3
PWM	0x1000 6000	PWMC	PWM Control Register
PWM	0x1000 6004	PWMS	PWM Sample Register
PWM	0x1000 6008	PWMP	PWM Period Register
PWM	0x1000 600C	PWMCNT	PWM Counter Register
RTC	0x1000 7000	HOURMIN	RTC Hours and Minutes Counter Register
RTC	0x1000 7004	SECONDS	RTC Seconds Counter Register
RTC	0x1000 7008	ALRM_HM	RTC Hours and Minutes Alarm Register
RTC	0x1000 700C	ALRM_SEC	RTC Seconds Alarm Register
RTC	0x1000 7010	RCCTL	RTC Control Register
RTC	0x1000 7014	RTCISR	RTC Interrupt Status Register
RTC	0x1000 7018	RTCIENR	RTC Interrupt Enable Register
RTC	0x1000 701C	STPWCH	Stopwatch Minutes Register

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
RTC	0x1000 7020	DAYR	RTC Days Counter Register
RTC	0x1000 7024	DAYALARM	RTC Day Alarm Register
KPP	0x1000_8000	KPCR	Keypad Control Register
KPP	0x1000_8002	KPSR	Keypad Status Register
KPP	0x1000_8004	KDDR	Keypad Data Direction Register
KPP	0x1000_8006	KPDR	Keypad Data Register
O-Wire	0x1000 9000	CONTROL	1-Wire Control Register
O-Wire	0x1000 9002	TIME_DIVIDER	1-Wire Time Divider Register
O-Wire	0x1000 9004	RESET	1-Wire Reset Register
UART1	0x1000_A000	UXRD_1	UART1 Receiver Register
UART1	0x1000_A040	UTXD_1	UART1 Transmitter Register
UART1	0x1000_A080	UCR1_1	UART1 Control Register
UART1	0x1000_A084	UCR2_1	UART1 Control Register 2
UART1	0x1000_A088	UCR3_1	UART1 Control Register 3
UART1	0x1000_A08C	UCR4_1	UART1 Control Register 4
UART1	0x1000_A090	UFCR_1	UART1 FIFO Control Register
UART1	0x1000_A094	USR1_1	UART1 Status Register 1
UART1	0x1000_A098	USR2_1	UART1 Status Register 2
UART1	0x1000_A09C	UESC_1	UART1 Escape Character Register
UART1	0x1000_A0A0	UTIM_1	UART1 Escape Timer Register
UART1	0x1000_A0A4	UBIR_1	UART1 BRM Incremental Register
UART1	0x1000_A0A8	UBMR_1	UART1 BRM Modulator Register
UART1	0x1000_A0AC	UBRC_1	UART1 Baud Rate Count Register
UART1	0x1000_A0B0	ONEMS_1	UART1 One Millisecond Register
UART1	0x1000_A0B4	UTS_1	UART1 Test Register 1
UART2	0x1000_B000	UXRD_2	UART2 Receiver Register
UART2	0x1000_B040	UTXD_2	UART2 Transmitter Register
UART2	0x1000_B080	UCR1_2	UART2 Control Register
UART2	0x1000_B084	UCR2_2	UART2 Control Register 2
UART2	0x1000_B088	UCR3_2	UART2 Control Register 3
UART2	0x1000_B08C	UCR4_2	UART2 Control Register 4
UART2	0x1000_B090	UFCR_2	UART2 FIFO Control Register
UART2	0x1000_B094	USR1_2	UART2 Status Register 1

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
UART2	0x1000_B098	USR2_2	UART2 Status Register 2
UART2	0x1000_B09C	UESC_2	UART2 Escape Character Register
UART2	0x1000_B0A0	UTIM_2	UART2 Escape Timer Register
UART2	0x1000_B0A4	UBIR_2	UART2 BRM Incremental Register
UART2	0x1000_B0A8	UBMR_2	UART2 BRM Modulator Register
UART2	0x1000_B0AC	UBRC_2	UART2 Baud Rate Count Register
UART2	0x1000_B0B0	ONEMS_2	UART2 One Millisecond Register
UART2	0x1000_B0B4	UTS_2	UART2 Test Register 1
UART3	0x1000_C000	UXRD_3	UART3 Receiver Register
UART3	0x1000_C040	UTXD_3	UART3 Transmitter Register
UART3	0x1000_C080	UCR1_3	UART3 Control Register
UART3	0x1000_C084	UCR2_3	UART3 Control Register 2
UART3	0x1000_C088	UCR3_3	UART3 Control Register 3
UART3	0x1000_C08C	UCR4_3	UART3 Control Register 4
UART3	0x1000_C090	UFCR_3	UART3 FIFO Control Register
UART3	0x1000_C094	USR1_3	UART3 Status Register 1
UART3	0x1000_C098	USR2_3	UART3 Status Register 2
UART3	0x1000_C09C	UESC_3	UART3 Escape Character Register
UART3	0x1000_C0A0	UTIM_3	UART3 Escape Timer Register
UART3	0x1000_C0A4	UBIR_3	UART3 BRM Incremental Register
UART3	0x1000_C0A8	UBMR_3	UART3 BRM Modulator Register
UART3	0x1000_C0AC	UBRC_3	UART3 Baud Rate Count Register
UART3	0x1000_C0B0	ONEMS_3	UART3 One Millisecond Register
UART3	0x1000_C0B4	UTS_3	UART3 Test Register 1
UART4	0x1000_D000	UXRD_4	UART4 Receiver Register
UART4	0x1000_D040	UTXD_4	UART4 Transmitter Register
UART4	0x1000_D080	UCR1_4	UART4 Control Register
UART4	0x1000_D084	UCR2_4	UART4 Control Register 2
UART4	0x1000_D088	UCR3_4	UART4 Control Register 3
UART4	0x1000_D08C	UCR4_4	UART4 Control Register 4
UART4	0x1000_D090	UFCR_4	UART4 FIFO Control Register
UART4	0x1000_D094	USR1_4	UART4 Status Register 1
UART4	0x1000_D098	USR2_4	UART4 Status Register 2

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
UART4	0x1000_D09C	UESC_4	UART4 Escape Character Register
UART4	0x1000_D0A0	UTIM_4	UART4 Escape Timer Register
UART4	0x1000_D0A4	UBIR_4	UART4 BRM Incremental Register
UART4	0x1000_D0A8	UBMR_4	UART4 BRM Modulator Register
UART4	0x1000_D0AC	UBRC_4	UART4 Baud Rate Count Register
UART4	0x1000_D0B0	ONEMS_4	UART4 One Millisecond Register
UART4	0x1000_D0B4	UTS_4	UART4 Test Register 1
CSPI1	0x1000 E000	RXDATA1	Receive Data Register 1
CSPI1	0x1000 E004	TXDATA1	Transmit Data Register 1
CSPI1	0x1000 E008	CONTROL_REG1	CSPI Control Register 1
CSPI1	0x1000 E00C	INT_REG1	Interrupt Control/Status Register 1
CSPI1	0x1000 E010	TEST_REG	CSPI Test Register 1
CSPI1	0x1000 E014	PERIOD1	CSPI Sample Period Control Register 1
CSPI1	0x1000 E018	CSPI_DMA1	CSPI DMA Register 1
CSPI1	0x1000 E01C	CSPI_RESET1	CSPI 1 Soft Reset Register
CSPI2	0x1000 F000	RXDATA2	Receive Data Register 2
CSPI2	0x1000 F004	TXDATA2	Transmit Data Register 2
CSPI2	0x1000 F008	CONTROL_REG2	CSPI Control Register 2
CSPI2	0x1000 F00C	INT_REG2	Interrupt Control/Status Register 2
CSPI2	0x1000 F010	TEST_REG 2	CSPI Test Register 2
CSPI2	0x1000 F014	PERIOD2	CSPI Sample Period Control Register 2
CSPI2	0x1000 F018	CSPI_DMA2	CSPI DMA Register 2
CSPI2	0x1000 F01C	CSPI_RESET2	CSPI 2 Soft Reset Register
SSI 1	0x1001 0000	STX0	SSI Transmit Data Register 0
SSI 1	0x1001 0004	STX1	SSI Transmit Data Register 1
SSI 1	0x1001 0008	SRX0	SSI Receive Data Register 0
SSI 1	0x1001 000C	SRX1	SSI Receive Data Register 1
SSI 1	0x1001 0010	SCR	SSI Control Register
SSI 1	0x1001 0014	SISR	SSI Interrupt Status Register
SSI 1	0x1001 0018	SIER	SSI Interrupt Enable Register
SSI 1	0x1001 001C	STCR	SSI Transmit Configuration Register
SSI 1	0x1001 0020	SRCR	SSI Receive Configuration Register
SSI 1	0x1001 0024	STCCR	SSI Transmit Clock Control Register

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
SSI 1	0x1001 0028	SRCCR	SSI Receive Clock Control Register
SSI 1	0x1001 002C	SFCSR	SSI FIFO Control/Status Register
SSI 1	0x1001 0030	STR	SSI Test Register
SSI 1	0x1001 0034	SOR	SSI Option Register
SSI 1	0x1001 0038	SACNT	SSI AC97 Control Register
SSI 1	0x1001 003C	SACADD	SSI AC97 Command Address Register
SSI 1	0x1001 0040	SACDAT	SSI AC97 Command Data Register
SSI 1	0x1001 0044	SATAG	SSI AC97 Tag Register
SSI 1	0x1001 0048	STMSK	SSI Transmit Time Slot Mask Register
SSI 1	0x1001 004C	SRMSK	SSI Receive Time Slot Mask Register
SSI 2	0x1001 1000	STX0	SSI Transmit Data Register 0
SSI 2	0x1001 1004	STX1	SSI Transmit Data Register 1
SSI 2	0x1001 1008	SRX0	SSI Receive Data Register 0
SSI 2	0x1001 100C	SRX1	SSI Receive Data Register 1
SSI 2	0x1001 1010	SCR	SSI Control Register
SSI 2	0x1001 1014	SISR	SSI Interrupt Status Register
SSI 2	0x1001 1018	SIER	SSI Interrupt Enable Register
SSI 2	0x1001 101C	STCR	SSI Transmit Configuration Register
SSI 2	0x1001 1020	SRCR	SSI Receive Configuration Register
SSI 2	0x1001 1024	STCCR	SSI Transmit Clock Control Register
SSI 2	0x1001 1028	SRCCR	SSI Receive Clock Control Register
SSI 2	0x1001 102C	SFCSR	SSI FIFO Control/Status Register
SSI 2	0x1001 1030	STR	SSI Test Register
SSI 2	0x1001 1034	SOR	SSI Option Register
SSI 2	0x1001 1038	SACNT	SSI AC97 Control Register
SSI 2	0x1001 103C	SACADD	SSI AC97 Command Address Register
SSI 2	0x1001 1040	SACDAT	SSI AC97 Command Data Register
SSI 2	0x1001 1044	SATAG	SSI AC97 Tag Register
SSI 2	0x1001 1048	STMSK	SSI Transmit Time Slot Mask Register
SSI 2	0x1001 104C	SRMSK	SSI Receive Time Slot Mask Register
I^2C	0x1001 2000	IADR	I^2C Address Register
I^2C	0x1001 2004	IFDR	I^2C Frequency Divider Register
I^2C	0x1001 2008	I2CR	I^2C Control Register

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
I^2C	0x1001 200C	I2SR	I^2C Status Register
I^2C	0x1001 2010	I2DR	I^2C Data I/O Register
SDHC1	0x1001 3000	STR_STP_CLK	MMC/SD1 Clock Control Register
SDHC1	0x1001 3004	STATUS (Read Only)	MMC/SD1 Status Register
SDHC1	0x1001 3008	CLK_RATE	MMC/SD1 Clock Rate Register
SDHC1	0x1001 300C	CMD_DAT_CONT	MMC/SD1 Command and Data Control Register
SDHC1	0x1001 3010	RESPONSE_TO	MMC/SD1 Response Time Out Register
SDHC1	0x1001 3014	READ_TO	MMC/SD1 Read Time Out Register
SDHC1	0x1001 3018	BLK_LEN	MMC/SD1 Block Length Register
SDHC1	0x1001 301C	NOB	MMC/SD1 Number of Block Register
SDHC1	0x1001 3020	REV_NO	MMC/SD1 Revision Number Register
SDHC1	0x1001 3024	INT_CNTL	MMC/SD1 Interrupt Control Register
SDHC1	0x1001 3028	CMD	MMC/SD1 Command Number Register
SDHC1	0x1001 302C	ARGH	MMC/SD1 Higher Argument Register
SDHC1	0x1001 3030	ARGL	MMC/SD1 Lower Argument Register
SDHC1	0x1001 3034	RES_FIFO (Read Only)	MMC/SD1 Response FIFO Register
SDHC1	0x1001 3038	BUFFER_ACCESS	MMC/SD1 Buffer Access Register
SDHC2	0x1001 4000	STR_STP_CLK	MMC/SD2 Clock Control Register
SDHC2	0x1001 4004	STATUS (Read Only)	MMC/SD2 Status Register
SDHC2	0x1001 4008	CLK_RATE	MMC/SD2 Clock Rate Register
SDHC2	0x1001 400C	CMD_DAT_CONT	MMC/SD2 Command and Data Control Register
SDHC2	0x1001 4010	RESPONSE_TO	MMC/SD2 Response Time Out Register
SDHC2	0x1001 4014	READ_TO	MMC/SD2 Read Time Out Register
SDHC2	0x1001 4018	BLK_LEN	MMC/SD2 Block Length Register
SDHC2	0x1001 401C	NOB	MMC/SD2 Number of Block Register
SDHC2	0x1001 4020	REV_NO	MMC/SD2 Revision Number Register
SDHC2	0x1001 4024	INT_CNTL	MMC/SD2 Interrupt Control Register
SDHC2	0x1001 4028	CMD	MMC/SD2 Command Number Register
SDHC2	0x1001 402C	ARGH	MMC/SD2 Higher Argument Register
SDHC2	0x1001 4030	ARGL	MMC/SD2 Lower Argument Register
SDHC2	0x1001 4034	RES_FIFO (Read Only)	MMC/SD2 Response FIFO Register
SDHC2	0x1001 4038	BUFFER_ACCESS	MMC/SD2 Buffer Access Register
GPIO	0x1001 5000	PTA_DDIR	Data Direction Register, Port A

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
GPIO	0x1001 5004	PTA_OCR1	Output Configuration Register 1 (OCR1), Port A
GPIO	0x1001 5008	PTA_OCR2	Output Configuration Register 2 (OCR2), Port A
GPIO	0x1001 500C	PTA_ICONFA1	Input Configuration Register A1 (ICONFA1), Port A
GPIO	0x1001 5010	PTA_ICONFA2	Input Configuration Register A1 (ICONFA2), Port A
GPIO	0x1001 5014	PTA_ICONFB1	Input Configuration Register B1 (ICONFB1), Port A
GPIO	0x1001 5018	PTA_ICONFB2	Input Configuration Register B2 (ICONFB2), Port A
GPIO	0x1001 501c	PTA_DR	Data Register, Port A
GPIO	0x1001 5020	PTA_GIUS	GPIO In Use Register, Port A
GPIO	0x1001 5024	PTA_SSR	Sample Status Register, Port A
GPIO	0x1001 5028	PTA_ICR1	Interrupt Configuration Register 1, Port A
GPIO	0x1001 502C	PTA_ICR2	Interrupt Configuration Register 2, Port A
GPIO	0x1001 5030	PTA_IMR	Interrupt Mask Register, Port A
GPIO	0x1001 5034	PTA_ISR	Interrupt Status Register, Port A
GPIO	0x1001 5038	PTA_GPR	General Purpose Register, Port A
GPIO	0x1001 503c	PTA_SWR	Software Reset Register, Port A
GPIO	0x1001 5040	PTA_PUEN	Pull_up Enable Register, Port A
GPIO	0x1001_5100	PTB_DDIR	Data Direction Register, Port B
GPIO	0x1001 5104	PTB_OCR1	Output Configuration Register 1 (OCR1), Port B
GPIO	0x1001 5108	PTB_OCR2	Output Configuration Register 2 (OCR2), Port B
GPIO	0x1001 510c	PTB_ICONFA1	Input Configuration Register A1 (ICONFA1), Port B
GPIO	0x1001 5110	PTB_ICONFA2	Input Configuration Register A1 (ICONFA2), Port B
GPIO	0x1001 5114	PTB_ICONFB1	Input Configuration Register B1 (ICONFB1), Port B
GPIO	0x1001 5118	PTB_ICONFB2	Input Configuration Register B2 (ICONFB2), Port B
GPIO	0x1001 511c	PTB_DR	Data Register, Port B
GPIO	0x1001 5120	PTB_GIUS	GPIO In Use Register, Port B
GPIO	0x1001 5124	PTB_SSR	Sample Status Register, Port B
GPIO	0x1001 5128	PTB_ICR1	Interrupt Configuration Register 1, Port B
GPIO	0x1001 512C	PTB_ICR2	Interrupt Configuration Register 2, Port B
GPIO	0x1001 5130	PTB_IMR	Interrupt Mask Register, Port B
GPIO	0x1001 5134	PTB_ISR	Interrupt Status Register, Port B
GPIO	0x1001 5138	PTB_GPR	General Purpose Register, Port B
GPIO	0x1001 513c	PTB_SWR	Software Reset Register, Port B
GPIO	0x1001 5140	PTB_PUEN	Pull_up Enable Register, Port B

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
GPIO	0x1001_5200	PTC_DDIR	Data Direction Register, Port C
GPIO	0x1001 5204	PTC_OCR1	Output Configuration Register 1 (OCR1), Port C
GPIO	0x1001 5208	PTC_OCR2	Output Configuration Register 2 (OCR2), Port C
GPIO	0x1001 520c	PTC_ICONFA1	Input Configuration Register A1 (ICONFA1), Port C
GPIO	0x1001 5210	PTC_ICONFA2	Input Configuration Register A1 (ICONFA2), Port C
GPIO	0x1001 5214	PTC_ICONFB1	Input Configuration Register B1 (ICONFB1), Port C
GPIO	0x1001 5218	PTC_ICONFB2	Input Configuration Register B2 (ICONFB2), Port C
GPIO	0x1001 521C	PTC_DR	Data Register, Port C
GPIO	0x1001 5220	PTC_GIUS	GPIO In Use Register, Port C
GPIO	0x1001 5224	PTC_SSR	Sample Status Register, Port C
GPIO	0x1001 5228	PTC_ICR1	Interrupt Configuration Register 1, Port C
GPIO	0x1001 522C	PTC_ICR2	Interrupt Configuration Register 2, Port C
GPIO	0x1001 5230	PTC_IMR	Interrupt Mask Register, Port C
GPIO	0x1001 5234	PTC_ISR	Interrupt Status Register, Port C
GPIO	0x1001 5238	PTC_GPR	General Purpose Register, Port C
GPIO	0x1001 523c	PTC_SWR	Software Reset Register, Port C
GPIO	0x1001 5240	PTC_PUEN	Pull_up Enable Register, Port C
GPIO	0x1001 5300	PTD_DDIR	Data Direction Register, Port D
GPIO	0x1001 5304	PTD_OCR1	Output Configuration Register 1 (OCR1), Port D
GPIO	0x1001 5308	PTD_OCR2	Output Configuration Register 2 (OCR2), Port D
GPIO	0x1001 530c	PTD_ICONFA1	Input Configuration Register A1 (ICONFA1), Port D
GPIO	0x1001 5310	PTD_ICONFA2	Input Configuration Register A1 (ICONFA2), Port D
GPIO	0x1001 5314	PTD_ICONFB1	Input Configuration Register B1 (ICONFB1), Port D
GPIO	0x1001 5318	PTD_ICONFB2	Input Configuration Register B2 (ICONFB2), Port D
GPIO	0x1001 531c	PTD_DR	Data Register, Port D
GPIO	0x1001 5320	PTD_GIUS	GPIO In Use Register, Port D
GPIO	0x1001 5324	PTD_SSR	Sample Status Register, Port D
GPIO	0x1001 5328	PTD_ICR1	Interrupt Configuration Register 1, Port D
GPIO	0x1001 532C	PTD_ICR2	Interrupt Configuration Register 2, Port D
GPIO	0x1001 5330	PTD_IMR	Interrupt Mask Register, Port D
GPIO	0x1001 5334	PTD_ISR	Interrupt Status Register, Port D
GPIO	0x1001 5338	PTD_GPR	General Purpose Register, Port D
GPIO	0x1001 533c	PTD_SWR	Software Reset Register, Port D

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
GPIO	0x1001 5340	PTD_PUEN	Pull_up Enable Register, Port D
GPIO	0x1001 5400	PTE_DDIR	Data Direction Register, Port E
GPIO	0x1001 5404	PTE_OCR1	Output Configuration Register 1 (OCR1), Port E
GPIO	0x1001 5408	PTE_OCR2	Output Configuration Register 2 (OCR2), Port E
GPIO	0x1001 540c	PTE_ICONFA1	Input Configuration Register A1 (ICONFA1), Port E
GPIO	0x1001 5410	PTE_ICONFA2	Input Configuration Register A1 (ICONFA2), Port E
GPIO	0x1001 5414	PTE_ICONFB1	Input Configuration Register B1 (ICONFB1), Port E
GPIO	0x1001 5418	PTE_ICONFB2	Input Configuration Register B2 (ICONFB2), Port E
GPIO	0x1001 541c	PTE_DR	Data Register, Port E
GPIO	0x1001 5420	PTE_GIUS	GPIO In Use Register, Port E
GPIO	0x1001 5424	PTE_SSR	Sample Status Register, Port E
GPIO	0x1001 5428	PTE_ICR1	Interrupt Configuration Register 1, Port E
GPIO	0x1001 542C	PTE_ICR2	Interrupt Configuration Register 2, Port E
GPIO	0x1001 5430	PTE_IMR	Interrupt Mask Register, Port E
GPIO	0x1001 5434	PTE_ISR	Interrupt Status Register, Port E
GPIO	0x1001 5438	PTE_GPR	General Purpose Register, Port E
GPIO	0x1001 543c	PTE_SWR	Software Reset Register, Port E
GPIO	0x1001 5440	PTE_PUEN	Pull_up Enable Register, Port E
GPIO	0x1001 5500	PTF_DDIR	Data Direction Register, Port F
GPIO	0x1001 5504	PTF_OCR1	Output Configuration Register 1 (OCR1), Port F
GPIO	0x1001 5508	PTF_OCR2	Output Configuration Register 2 (OCR2), Port F
GPIO	0x1001 550C	PTF_ICONFA1	Input Configuration Register A1 (ICONFA1), Port F
GPIO	0x1001 5510	PTF_ICONFA2	Input Configuration Register A1 (ICONFA2), Port F
GPIO	0x1001 5514	PTF_ICONFB1	Input Configuration Register B1 (ICONFB1), Port F
GPIO	0x1001 5518	PTF_ICONFB2	Input Configuration Register B2 (ICONFB2), Port F
GPIO	0x1001 551c	PTF_DR	Data Register, Port F
GPIO	0x1001 5520	PTF_GIUS	GPIO In Use Register, Port F
GPIO	0x1001 5524	PTF_SSR	Sample Status Register, Port F
GPIO	0x1001 5528	PTF_ICR1	Interrupt Configuration Register 1, Port F
GPIO	0x1001 552C	PTF_ICR2	Interrupt Configuration Register 2, Port F
GPIO	0x1001 5530	PTF_IMR	Interrupt Mask Register, Port F
GPIO	0x1001 5534	PTF_ISR	Interrupt Status Register, Port F
GPIO	0x1001 5538	PTF_GPR	General Purpose Register, Port F

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
GPIO	0x1001 553c	PTF_SWR	Software Reset Register, Port F
GPIO	0x1001 5540	PTF_PUEN	Pull_up Enable Register, Port F
GPIO	0x1001 5600	PMASK	GPIO Port Interrupt Mask
AUDMUX	0x1001 6000	HPCR1	Host Port Configuration Register 1
AUDMUX	0x1001 6004	HPCR2	Host Port Configuration Register 2
AUDMUX	0x1001 6008	HPCR3	Host Port Configuration Register 3
AUDMUX	0x1001 6010	PPCR1	Peripheral Port Configuration Register 1
AUDMUX	0x1001 6014	PPCR2	Peripheral Port Configuration Register 2
AUDMUX	0x1001 601C	PPCR3	Peripheral Port Configuration Register 3
CSPI3	0x1001 7000	RXDATA3	Receive Data Register 3
CSPI3	0x1001 7004	TXDATA3	Transmit Data Register 3
CSPI3	0x1001 7008	CONTROL_REG3	CSPI Control Register 3
CSPI3	0x1001 700C	INT_REG3	Interrupt Control/Status Register 3
CSPI3	0x1001 7010	TEST_REG3	CSPI Test Register 3
CSPI3	0x1001 7014	PERIOD3	CSPI Sample Period Control Register 3
CSPI3	0x1001 7018	CSPI_DMA3	CSPI DMA Register 3
CSPI3	0x1001 701C	CSPI_RESET3	CSPI Soft Reset Register 3
AIPI2	0x1002 0000	PSR0	Peripheral Size Register0
AIPI2	0x1002 0004	PSR1	Peripheral Size Register1
AIPI2	0x1002 0008	PAR	Peripheral Access Register
LCDC	0x1002 1000	LSSAR	LCD Screen Start Address Register
LCDC	0x1002 1004	LSR	LCD Size Register
LCDC	0x1002 1008	LVPWR	LCD Virtual Page Width Register
LCDC	0x1002 100C	LCPR	LCD Cursor Position Register
LCDC	0x1002 1010	LCWHBR	LCD Cursor Width Height and Blink Register
LCDC	0x1002 1014	LCCMR	LCD Color Cursor Mapping Register
LCDC	0x1002 1018	LPCR	LCD Panel Configuration Register
LCDC	0x1002 101C	LHCR	LCD Horizontal Configuration Register
LCDC	0x1002 1020	LVCR	LCD Vertical Configuration Register
LCDC	0x1002 1024	LPOR	LCD Panning Offset Register
LCDC	0x1002 1028	LSCR	LCD Sharp Configuration Register
LCDC	0x1002 102C	LPCCR	LCD PWM Contrast Control Register
LCDC	0x1002 1030	LDCR	LCD DMA Control Register

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
LCDC	0x1002 1034	LRMCR	LCD Refresh Mode Control Register
LCDC	0x1002 1038	LICR	LCD Interrupt Configuration Register
LCDC	0x1002 103C	LIER	LCD Interrupt Enable Register
LCDC	0x1002 1040	LISR	LCD Interrupt Status Register
LCDC	0x1002 1050	LGWSAR	LCD Graphic Window Start Address Register
LCDC	0x1002 1054	LGWSR	LCD Graphic Window Size Register
LCDC	0x1002 1058	LGWVPWR	LCD Graphic Window Virtual Page Width Register
LCDC	0x1002 105C	LGWPOR	LCD Graphic Window Panning Offset Register
LCDC	0x1002 1060	LGWPR	LCD Graphic Window Position Registration Register
LCDC	0x1002 1064	LGWCR	LCD Graphic Window Control Register
LCDC	0x1002 1068	LGWDCR	LCD Graphic Window DMA Control Register
SLCDC	0x1002 2000	DATA_BASE_ADDR	SLCD Data Base Address Register
SLCDC	0x1002 2004	DATA_BUFF_SIZE	SLCD Data Buffer Size Register
SLCDC	0x1002 2008	CMD_BASE_ADDR	SLCD Command Buffer Base Address Register
SLCDC	0x1002 200C	CMD_BUFF_SIZE	SLCD Command Buffer Size Register
SLCDC	0x1002 2010	CMD_STR_SIZE	SLCD Command String Size Register
SLCDC	0x1002 2014	FIFO_CONFIG	SLCD FIFO Configuration Register
SLCDC	0x1002 2018	LCD_CONFIG	SLCD Configuration Register
SLCDC	0x1002 201C	LCD_XFER_CONFIG	SLCD Transfer Configuration Register
SLCDC	0x1002 2020	DMA_CTRL_STAT	SLCD DMA Control/Status Register
SLCDC	0x1002 2024	LCD_CLK_CONFIG	SLCD Clock Configuration Register
SLCDC	0x1002 2028	LCD_WRITE_DATA	SLCD Write Data Register
USBOTG	0x1002 4000	USB_HDW_MODE	USB OTG Hardware Mode Register
USBOTG	0x1002 4004	USB_INT_STATUS	USB OTG Module Interrupt Status Register
USBOTG	0x1002 4008	USB_INT_EN	USB OTG Module Interrupt Status Enable Register
USBOTG	0x1002 400C	USB_CLK_CTRL	USB OTG Clock Control Register
USBOTG	0x1002 4010	USB_RST_CTRL	USB OTG Reset Control Register
USBOTG	0x1002 4014	USB_FRAME_INTV	USB OTG Frame Interval Register
USBOTG	0x1002 4018	USB_FRAME_REMAIN	USB OTG Frame Remaining Register
USBOTG	0x1002 401C	USB_HNP_CSR	USB OTG HNP Control Status Register
USBOTG	0x1002 402C	USB_HNP_ISR	USB OTG HNP Interrupt Status Register
USBOTG	0x1002 4030	USB_HNP_IEN	USB OTG HNP Interrupt Enable Register
USBOTG	0x1002 4040	USB_FUNC_CMD_STAT	USB OTG Function Command Status Register

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
USBOTG	0x1002 4044	USB_DEV_ADD	USB OTG Device Address Register
USBOTG	0x1002 4048	USB_SYS_INT_STAT	USB OTG System Interrupt Status Register
USBOTG	0x1002 404C	USB_SYS_INT_EN	USB OTG System Interrupt Enable Register
USBOTG	0x1002 4050	USB_XBUFF_INT_STAT	USB OTG X Buffer Interrupt Status Register
USBOTG	0x1002 4054	USB_YBUFF_INT_STAT	USB OTG Y Buffer Interrupt Status Register
USBOTG	0x1002 4058	USB_XYINT_INT_EN	USB OTG XY Interrupt Enable Register
USBOTG	0x1002 405C	USB_XFILL_STAT	USB OTG X Filled Status Register
USBOTG	0x1002 4060	USB_YFILL_STAT	USB OTG Y Filled Status Register
USBOTG	0x1002 4064	USB_ENDPT_EN	USB OTG Endpoint Enable Register
USBOTG	0x1002 4068	USB_ENDPT_RDY	USB OTG Endpoint Ready Register
USBOTG	0x1002 406C	USB_IMM_INT	USB OTG Immediate Interrupt Register
USBOTG	0x1002 4070	USB_ENDPT_DONE_STAT	USB OTG Endpoint Done Status Register
USBOTG	0x1002 4074	USB_ENDPT_DONE_EN	USB OTG Endpoint Done Enable Register
USBOTG	0x1002 4078	USB_ENDPT_TOGGBITS	USB OTG EndpointToggle Bits Register
USBOTG	0x1002 407C	USB_FRAME_NO	USB OTG Frame Number Register
USBOTG	0x1002 4080	USB_HOST_CTRL	USB OTG Host Control Register
USBOTG	0x1002 4088	USB_SYS_ISR	USB OTG System Interrupt Status Register
USBOTG	0x1002 408C	USB_SYS_IEN	USB OTG System Interrupt Enable Register
USBOTG	0x1002 4098	USB_XBUF_ISR	USB OTG X Buffer Interrupt Status Register
USBOTG	0x1002 409C	USB_YBUF_ISR	USB OTG Y Buffer Interrupt Status Register
USBOTG	0x1002 40A0	USB_XYINT_EN	USB OTG XY Interrupt Enables Register
USBOTG	0x1002 40A8	USB_XFILL_STAT	USB OTG X Filled Status Register
USBOTG	0x1002 40AC	USB_YFILL_STAT	USB OTG Y Filled Status Register
USBOTG	0x1002 40C0	USB_ETD_EN	USB OTG ETD Enable Register
USBOTG	0x1002 40C8	NOT DEFINED	–
USB OTG	0x1002 40CC	USB_IM_INT	USB OTG Immediate Interrupt Register
USB OTG	0x1002 40D0	USB_ETD_DONE	USB OTG ETD Done Status Register
USBOTG	0x1002 40D4	USB_ETD_DONE_EN	USB OTG ETD Done Enable Register
USBOTG	0x1002 40E0	USB_FR_NUM	USB OTG Frame Number Register
USBOTG	0x1002 40E4	USB_LSTR	USB OTG Low Speed Threshold Register
USBOTG	0x1002 40E8	USB_RHD_A	USB OTG Root Hub DescriptorA Register
USBOTG	0x1002 40EC	USB_RHD_B	USB OTG Root Hub DescriptorB Register
USBOTG	0x1002 40F0	USB_RH_STAT	USB OTG Root Hub Status Register

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
USBOTG	0x1002 40F4	USB_PORT_STAT1	USB OTG Port Status 1 Register
USBOTG	0x1002 40F8	USB_PORT_STAT2	USB OTG Port Status 2 Register
USBOTG	0x1002 40FC	USB_PORT_STAT3	USB OTG Port Status 3 Register
USBOTG	0x1002 4600	USB_CTRL	USB Control Register
USBOTG	0x1002 4800	USB_DMA_REV	DMA Revision Register
USBOTG	0x1002 4804	USB_DMA_INT_STAT	DMA Interrupt Status Register
USBOTG	0x1002 4808	USB_DMA_INT_EN	DMA Interrupt Enable Register
USBOTG	0x1002 480C	USB_ETDDMA_ER_STAT	ETD DMA Error Status Register
USBOTG	0x1002 4810	USB_EPDMA_ER_STAT	EP DMA Error Status Register
USBOTG	0x1002 4814 through 0x1002 481C	Reserved	Reserved
USBOTG	0x1002 4820	USB_ETD_DMA_EN	ETD DMA Enable Register
USBOTG	0x1002 4824	USB_EP_DMA_EN	EP DMA Enable Register
USBOTG	0x1002 4828	USB_ETDMA_XTRG_REQ	ETD DMA Enable XTrigger Request Register
USBOTG	0x1002 482C	USB_EPMA_XTRG_REQ	EP DMA Enable XTrigger Request Register
USBOTG	0x1002 4830	USB_ETDMA_XYTRG_REQ	ETD DMA Enable XYTrigger Request Register
USBOTG	0x1002 4834	USB_EPMA_XYTRG_REQ	EP DMA Enable XYTrigger Request Register
USBOTG	0x1002 4838	Reserved	Reserved
USBOTG	0x1002 483C	Reserved	Reserved
USBOTG	0x1002 4840	USB_MISC_CNTL	Misc Control Register
USBOTG	0x1002 4900 through 0x1002 497C	USB_ETD0_SYSMEM_ADR through USB_ETD31_SYSMEM_ADR	ETD 0: ETD 0 Sys Memory Start Address Register through ETD 31: ETD 31 Sys Memory Start Address Register
USBOTG	0x1002 4980	USB_EP0_OUT_MEM_ADR	EP 0 OUT: EP 0 Sys Memory Start Address Register
USBOTG	0x1002 4984	USB_EP0_IN_MEM_ADR	EP 0 IN: EP 0 Sys Memory Start Address Register
USBOTG	0x1002 4988	USB_EP1_OUT_MEM_ADR	EP 1 OUT: EP 1 Sys Memory Start Address Register
USBOTG	0x1002 498C	USB_EP1_IN_MEM_ADR	EP 1 IN: EP 1 Sys Memory Start Address Register
USBOTG	0x1002 4990	USB_EP2_OUT_MEM_ADR	EP 2 OUT: EP 2 Sys Memory Start Address Register
USBOTG	0x1002 4994	USB_EP2_IN_MEM_ADR	EP 2 IN: EP 2 Sys Memory Start Address Register
USBOTG	0x1002 4998	USB_EP3_OUT_MEM_ADR	EP 3 OUT: EP 3 Sys Memory Start Address Register
USBOTG	0x1002 499C	USB_EP3_IN_MEM_ADR	EP 3 IN: EP 3 Sys Memory Start Address Register
USBOTG	0x1002 49A0	USB_EP4_OUT_MEM_ADR	EP 4OUT: EP 4 Sys Memory Start Address Register
USBOTG	0x1002 49A4	USB_EP4_IN_MEM_ADR	EP 4 IN: EP 4 Sys Memory Start Address Register

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
USBOTG	0x1002 49A8	USB_EP5_OUT_MEM_ADR	EP 5 OUT: EP 5 Sys Memory Start Address Register
USBOTG	0x1002 49AC	USB_EP5_IN_MEM_ADR	EP 5 IN: EP 5 Sys Memory Start Address Register
USBOTG	0x1002 49B0	USB_EP6_OUT_MEM_ADR	EP 6 OUT: EP 6 Sys Memory Start Address Register
USBOTG	0x1002 49B4	USB_EP6_IN_MEM_ADR	EP 6 IN: EP 6 Sys Memory Start Address Register
USBOTG	0x1002 49B8	USB_EP7_OUT_MEM_ADR	EP 7 OUT: EP 7 Sys Memory Start Address Register
USBOTG	0x1002 49BC	USB_EP7_IN_MEM_ADR	EP 7 IN: EP 7 Sys Memory Start Address Register
USBOTG	0x1002 49C0	USB_EP8_OUT_MEM_ADR	EP 8 OUT: EP 8 Sys Memory Start Address Register
USBOTG	0x1002 49C4	USB_EP8_IN_MEM_ADR	EP 8 IN: EP 8 Sys Memory Start Address Register
USBOTG	0x1002 49C8	USB_EP9_OUT_MEM_ADR	EP 9 OUT: EP 9 Sys Memory Start Address Register
USBOTG	0x1002 49CC	USB_EP9_IN_MEM_ADR	EP 9 IN: EP 9 Sys Memory Start Address Register
USBOTG	0x1002 49D0	USB_EP10_OUT_MEM_ADR	EP 10 OUT: EP 10 Sys Memory Start Address Register
USBOTG	0x1002 49D4	USB_EP10_IN_MEM_ADR	EP 10 IN: EP 10 Sys Memory Start Address Register
USBOTG	0x1002 49D8	USB_EP11_OUT_MEM_ADR	EP 11 OUT: EP 11 Sys Memory Start Address Register
USBOTG	0x1002 49DC	USB_EP11_IN_MEM_ADR	EP 11 IN: EP 11 Sys Memory Start Address Register
USBOTG	0x1002 49E0	USB_EP12_OUT_MEM_ADR	EP 12OUT: EP 12 Sys Memory Start Address Register
USBOTG	0x1002 49E4	USB_EP12_IN_MEM_ADR	EP 12 IN: EP 12 Sys Memory Start Address Register
USBOTG	0x1002 49E8	USB_EP13_OUT_MEM_ADR	EP 13 OUT: EP 13 Sys Memory Start Address Register
USBOTG	0x1002 49EC	USB_EP13_IN_MEM_ADR	EP 13 IN: EP 13 Sys Memory Start Address Register
USBOTG	0x1002 49F0	USB_EP14_OUT_MEM_ADR	EP 14 OUT: EP 14 Sys Memory Start Address Register
USBOTG	0x1002 49F4	USB_EP14_IN_MEM_ADR	EP 14 IN: EP 14 Sys Memory Start Address Register
USBOTG	0x1002 49F8	USB_EP15_OUT_MEM_ADR	EP 15 OUT: EP 15 Sys Memory Start Address Register
USBOTG	0x1002 49FC	USB_EP15_IN_MEM_ADR	EP 15 IN: EP 15 Sys Memory Start Address Register
USBOTG	0x1002 4A00 through 0x1002 4A7C	USB_ETD0 DMABUF XFER through USB_ETD31 DMABUF XFER	ETD 0: ETD0 DMA Buffer Transfer Pointer Register through ETD 31: ETD31 DMA Buffer Transfer Pointer Register
USBOTG	0x1002 4A80	USB_EP0_OUT_BUF_INDX	EP 0 OUT: EP 0 Current buffer index register
USBOTG	0x1002 4A84	USB_EP0_IN_BUF_INDX	EP 0 IN: EP 0 Current buffer index register
USBOTG	0x1002 4A88	USB_EP1_OUT_BUF_INDX	EP 1 OUT: EP 1 Current buffer index register
USBOTG	0x1002 4A8C	USB_EP1_IN_BUF_INDX	EP 1 IN: EP 1 Current buffer index register
USBOTG	0x1002 4A90	USB_EP2_OUT_BUF_INDX	EP 2 OUT: EP 2 Current buffer index register

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
USBOTG	0x1002 4A94	USB_EP2_IN_BUF_INDX	EP 2 IN: EP 2 Current buffer index register
USBOTG	0x1002 4A98	USB_EP3_OUT_BUF_INDX	EP 3 OUT: EP 3 Current buffer index register
USBOTG	0x1002 4A9C	USB_EP3_IN_BUF_INDX	EP 3 IN: EP 3 Current buffer index register
USBOTG	0x1002 4AA0	USB_EP4_OUT_BUF_INDX	EP 4OUT: EP 4 Current buffer index register
USBOTG	0x1002 4AA4	USB_EP4_IN_BUF_INDX	EP 4 IN: EP 4 Current buffer index register
USBOTG	0x1002 4AA8	USB_EP5_OUT_BUF_INDX	EP 5 OUT: EP 5 Current buffer index register
USBOTG	0x1002 4AAC	USB_EP5_IN_BUF_INDX	EP 5 IN: EP 5 Current buffer index register
USBOTG	0x1002 4AB0	USB_EP6_OUT_BUF_INDX	EP 6 OUT: EP 6 Current buffer index register
USBOTG	0x1002 4AB4	USB_EP6_IN_BUF_INDX	EP 6 IN: EP 6 Current buffer index register
USBOTG	0x1002 4AB8	USB_EP7_OUT_BUF_INDX	EP 7 OUT: EP 7 Current buffer index register
USBOTG	0x1002 4ABC	USB_EP7_IN_BUF_INDX	EP 7 IN: EP 7 Current buffer index register
USBOTG	0x1002 4AC0	USB_EP8_OUT_BUF_INDX	EP 8 OUT: EP 8 Current buffer index register
USBOTG	0x1002 4AC4	USB_EP8_IN_BUF_INDX	EP 8 IN: EP 8 Current buffer index register
USBOTG	0x1002 4AC8	USB_EP9_OUT_BUF_INDX	EP 9 OUT: EP 9 Current buffer index register
USBOTG	0x1002 4ACC	USB_EP9_IN_BUF_INDX	EP 9 IN: EP 9 Current buffer index register
USBOTG	0x1002 4AD0	USB_EP10_OUT_BUF_INDX	EP 10 OUT: EP 10 Current buffer index register
USBOTG	0x1002 4AD4	USB_EP10_IN_BUF_INDX	EP 10 IN: EP 10 Current buffer index register
USBOTG	0x1002 4AD8	USB_EP11_OUT_BUF_INDX	EP 11 OUT: EP 11 Current buffer index register
USBOTG	0x1002 4ADC	USB_EP11_IN_BUF_INDX	EP 11 IN: EP 11 Current buffer index register
USBOTG	0x1002 4AE0	USB_EP12_OUT_BUF_INDX	EP 12OUT: EP 12 Current buffer index register
USBOTG	0x1002 4AE4	USB_EP12_IN_BUF_INDX	EP 12 IN: EP 12 Current buffer index register
USBOTG	0x1002 4AE8	USB_EP13_OUT_BUF_INDX	EP 13 OUT: EP 13 Current buffer index register
USBOTG	0x1002 4AEC	USB_EP13_IN_BUF_INDX	EP 13 IN: EP 13 Current buffer index register
USBOTG	0x1002 4AF0	USB_EP14_OUT_BUF_INDX	EP 14 OUT: EP 14 Current buffer index register
USBOTG	0x1002 4AF4	USB_EP14_IN_BUF_INDX	EP 14 IN: EP 14 Current buffer index register
USBOTG	0x1002 4AF8	USB_EP15_OUT_BUF_INDX	EP 15 OUT: EP 15 Current buffer index register
USBOTG	0x1002 4AFC	USB_EP15_IN_BUF_INDX	EP 15 IN: EP 15 Current buffer index register
USBOTG	0x1002 4B00 through 0x1002 4FFF	Reserved	Reserved
USBOTG	0x1002 5000 through 0x1002 5FFF	Data Memory	USB OTG Data Memory
eMMA	0x1002 6000	PP_CNTL	PP Control Register

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
eMMA	0x1002 6004	PP_INTRCNTL	PP Interrupt Control Register
eMMA	0x1002 6008	PP_INTRSTATUS	PP interrupt Status Register
eMMA	0x1002 600C	PP_SOURCE_Y_PTR	PP Source “Y” FramE Data Pointer Register
eMMA	0x1002 6010	PP_SOURCE_CB_PTR	PP Source “CB” Frame Data Pointer Register
eMMA	0x1002 6014	PP_SOURCE_CR_PTR	PP Source “CR” Frame Data Pointer Register
eMMA	0x1002 6018	PP_DEST_RGB_PTR	PP Destination “RGB” Frame Start Address Register
eMMA	0x1002 601C	PP_QUANTIZER_PTR	PP Quantizer Start Address Register
eMMA	0x1002 6020	PP_PROCESS_FRAME_PARA	PP Process Frame Parameter, Width And Height Register
eMMA	0x1002 6024	PP_SOURCE_FRAME_WIDTH	PP Source Frame Width Register
eMMA	0x1002 6028	PP_DEST_DISPLAY_WIDTH	PP Destination Display Width Register
eMMA	0x1002 602C	PP_DEST_IMAGE_SIZE	PP Destination Image Size Register
eMMA	0x1002 6030	PP_DEST_FRAME_FMT_CNTL	PP Destination Frame Format Control Register
eMMA	0x1002 6034	PP Resize Table index Reg	PP Resize Table Index Register
eMMA	0x1002 6038	PP_CSC_COEFF_012	PP CSC Coefficient 0, 1, and 2 Register
eMMA	0x1002 603C	PP_CSC_COEFF_34	PP CSC Coefficient 3 and 4 Register
eMMA	0x1002 6100 through 0x1002 619C	PP_RESIZE_COEF_TBL	PP Resize Coefficient Table Register
eMMA	0x1002 6400	PrP_CNTL	PrP Control Register
eMMA	0x1002 6404	PrP_INTRCNTL	PrP Interrupt Control Register
eMMA	0x1002 6408	PrP_INTRSTATUS	PrP interrupt Status Register
eMMA	0x1002 640C	PrP_SOURCE_Y_PTR	PrP Source “Y” Frame Start Address Register
eMMA	0x1002 6410	PrP_SOURCE_CB_PTR	PrP Source “CB” Frame Start Address Register
eMMA	0x1002 6414	PrP_SOURCE_CR_PTR	PrP Source “CR” Frame Start Address Register
eMMA	0x1002 6418	PrP_DEST_RGB1_PTR	PrP Destination “RGB” Frame-1 Start Address Register
eMMA	0x1002 641C	PrP_DEST_RGB2_PTR	PrP Destination “RGB” Frame-2 Start Address Register
eMMA	0x1002 6420	PrP_DEST_Y_PTR	PrP Destination “Y” Frame Start Address Register
eMMA	0x1002 6424	PrP_DEST_CB_PTR	PrP Destination “CB” Frame Start Address Register
eMMA	0x1002 6428	PrP_DEST_CR_PTR	PrP Destination “CR” Frame Start Address Register
eMMA	0x1002 642C	PrP_SOURCE_FRAME_SIZE	PrP Source Frame Size Register
eMMA	0x1002 6430	PrP_CH1_LINE_STRIDE	PrP Channel-1 Line stride Register
eMMA	0x1002 6434	PrP_SRC_PIXEL_FORMAT_CNTL	PrP Source Pixel Format Control Register

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
eMMA	0x1002 6438	PrP_CH1_PIXEL_FORMAT_CNTL	PrP CH1 Pixel Format Control Register
eMMA	0x1002 643C	PrP_CH1_OUT_IMAGE_SIZE	PrP CH1 Output Image Size Register
eMMA	0x1002 6440	PrP_CH2_OUT_IMAGE_SIZE	PrP CH2 Output Image Size Register
eMMA	0x1002 6444	PrP_SOURCE_LINE_STRIDE	PrP Source Line Stride Register
eMMA	0x1002 6448	PrP_CSC_COEFF_012	PrP CSC Coefficients 0, 1, and 2 Register
eMMA	0x1002 644C	PrP_CSC_COEFF_345	PrP CSC Coefficients 3, 4, and 5 Register
eMMA	0x1002 6450	PrP_CSC_COEFF_678	PrP CSC Coefficients 6, 7, and 8 Register
eMMA	0x1002 6454	PrP_CH1_HRESIZE_COEFF1	PrP CH1 Horizontal Resize Coefficients Register
eMMA	0x1002 6458	PrP_CH1_HRESIZE_COEFF2	PrP CH1 Horizontal Resize Coefficients Register
eMMA	0x1002 645C	PrP_CH1_HRESIZE_VALID	PrP CH1 Horizontal Resize Valid Register
eMMA	0x1002 6460	PrP_CH1_VRESIZE_COEFF1	PrP CH1 Vertical Resize Coefficients Register
eMMA	0x1002 6464	PrP_CH1_VRESIZE_COEFF2	PrP CH1 Vertical Resize Coefficients Register
eMMA	0x1002 6468	PrP_CH1_VRESIZE_VALID	PrP CH1 Vertical Resize Valid Register
eMMA	0x1002 646C	PrP_CH2_HRESIZE_COEFF1	PrP CH2 Horizontal Resize Coefficients Register
eMMA	0x1002 6470	PrP_CH2_HRESIZE_COEFF2	PrP CH2 Horizontal Resize Coefficients Register
eMMA	0x1002 6474	PrP_CH2_HRESIZE_VALID	PrP CH2 Horizontal Resize Valid Register
eMMA	0x1002 6478	PrP_CH2_VRESIZE_COEFF1	PrP CH2 Vertical Resize Coefficients Register
eMMA	0x1002 647C	PrP_CH2_VRESIZE_COEFF2	PrP CH2 Vertical Resize Coefficients Register
eMMA	0x1002 6480	PrP_CH2_VRESIZE_VALID	PrP CH2 Vertical Resize Valid Register
PLLCLK	0x1002 7000	CSCR	Clock Source Control Register
PLLCLK	0x1002 7004	MPCTL0	MPLL Control Register 0
PLLCLK	0x1002 7008	MPCTL1	MPLL Control Register 1
PLLCLK	0x1002 700C	SPCTL0	SPLL Control Register 0
PLLCLK	0x1002 7010	SPCTL1	SPLL Control Register 1
PLLCLK	0x1002 7014	OSC26MCTL	Oscillator 26M Register
PLLCLK	0x1002 7018	PCDR0	Peripheral Clock Divider Register 0
PLLCLK	0x1002 701C	PCDR1	Peripheral Clock Divider Register 1
PLLCLK	0x1002 7020	PCCR0	Peripheral Clock Control Register 0

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
PLLCLK	0x1002 7024	PCCR1	Peripheral Clock Control Register 1
PLLCLK	0x1002 7028	CCSR	Clock Control Status Register
PLLCLK	0x1002 702C	PMCTL	PMOS Control Register
PLLCLK	0x1002 7030	PMCOUNT	PMOS Counter Register
PLLCLK	0x1002 7034	WKGDCTL	Wakeup Guard Mode Control Register
SYSCTRL	0x1002 7804	SIDR	Silicon ID Register
SYSCTRL	0x1002 7814	FMCR	Function Multiplexing Control Register
SYSCTRL	0x1002 7818	GPCR	Global Peripheral Control Register
SYSCTRL	0x1002 781C	WBCR	Well Bias Control Register
SYSCTRL	0x1002 7820	DSCR1	Driving Strength Control Register 1
SYSCTRL	0x1002 7824	DSCR2	Driving Strength Control Register 2
SYSCTRL	0x1002 7828	DSCR3	Driving Strength Control Register 3
SYSCTRL	0x1002 782C	DSCR4	Driving Strength Control Register 4
SYSCTRL	0x1002 7830	DSCR5	Driving Strength Control Register 5
SYSCTRL	0x1002 7834	DSCR6	Driving Strength Control Register 6
SYSCTRL	0x1002 7838	DSCR7	Driving Strength Control Register 7
SYSCTRL	0x1002 783C	DSCR8	Driving Strength Control Register 8
SYSCTRL	0x1002 7840	DSCR9	Driving Strength Control Register 9
SYSCTRL	0x1002 7844	DSCR10	Driving Strength Control Register 10
SYSCTRL	0x1002 7848	DSCR11	Driving Strength Control Register 11
SYSCTRL	0x1002 784C	DSCR12	Driving Strength Control Register 12
SYSCTRL	0x1002 7850	PCSR	Priority Control and Select Register
FIRI	0x1002 8000	FIRITCR	FIRI Transmit Control Register
FIRI	0x1002 8004	FIRITCTR	FIRI Transmit Count Register
FIRI	0x1002 8008	FIRIRCR	FIRI Receive Control Register
FIRI	0x1002 800C	FIRITSR	FIRI Transmit Status Register
FIRI	0x1002 8010	FIRIRSR	FIRI Receive Status Register
FIRI	0x1002 8014	FIRIXMITFIFO	Transmitter FIFO
FIRI	0x1002 8018	FIRIRCVRFIFO	Receiver FIFO
FIRI	0x1002 801C	FIRICR	FIRI Control Register
MAX	0x1003 F000	MPR0	Master Priority Register for Slave Port 0
MAX	0x1003 F100	MPR1	Master Priority Register for Slave Port 1
MAX	0x1003 F200	MPR2	Master Priority Register for Slave Port 2

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
MAX	0x1003 F300	MPR3	Master Priority Register for Slave Port 3
MAX	0x1003 F004	AMPR0	Alternate Master Priority Register for Slave Port 0
MAX	0x1003 F104	AMPR1	Alternate Master Priority Register for Slave Port 1
MAX	0x1003 F204	AMPR2	Alternate Master Priority Register for Slave Port 2
MAX	0x1003 F304	AMPR3	Alternate Master Priority Register for Slave Port 3
MAX	0x1003 F010	SGPCR0	General Purpose Control Register for Slave Port 0
MAX	0x1003 F110	SGPCR1	General Purpose Control Register for Slave Port 1
MAX	0x1003 F210	SGPCR2	General Purpose Control Register for Slave Port 2
MAX	0x1003 F310	SGPCR3	General Purpose Control Register for Slave Port 3
MAX	0x1003 F014	ASGPCR0	Alternate SGPCR for Slave Port 0
MAX	0x1003 F114	ASGPCR1	Alternate SGPCR for Slave Port 1
MAX	0x1003 F214	ASGPCR2	Alternate SGPCR for Slave Port 2
MAX	0x1003 F314	ASGPCR3	Alternate SGPCR for Slave Port 3
MAX	0x1003 F800	MGPCR0	General Purpose Control Register for Master Port 0
MAX	0x1003 F900	MGPCR1	General Purpose Control Register for Master Port 1
MAX	0x1003 FA00	MGPCR2	General Purpose Control Register for Master Port 2
MAX	0x1003 FB00	MGPCR3	General Purpose Control Register for Master Port 3
MAX	0x1003 FC00	MGPCR4	General Purpose Control Register for Master Port 4
MAX	0x1003 FD00	MGPCR5	General Purpose Control Register for Master Port 5
AITC	0x1004 0000	INTCNTL	Interrupt Control Register
AITC	0x1004 0004	NIMASK	Normal Interrupt Mask Register
AITC	0x1004 0008	INTENNUM	Interrupt Enable Number Register
AITC	0x1004 000C	INTDISNUM	Interrupt Disable Number Register
AITC	0x1004 0010	INTENABLEH	Interrupt Enable Register High
AITC	0x1004 0014	INTENABLEL	Interrupt Enable Register Low
AITC	0x1004 0018	INTTYPEH	Interrupt Type Register High
AITC	0x1004 001C	INTTYPEL	Interrupt Type Register Low
AITC	0x1004 0020	NIPRIORITY7	Normal Interrupt Priority Level Register 7
AITC	0x1004 0024	NIPRIORITY6	Normal Interrupt Priority Level Register 6
AITC	0x1004 0028	NIPRIORITY5	Normal Interrupt Priority Level Register 5
AITC	0x1004 002C	NIPRIORITY4	Normal Interrupt Priority Level Register 4
AITC	0x1004 0030	NIPRIORITY3	Normal Interrupt Priority Level Register 3
AITC	0x1004 0034	NIPRIORITY2	Normal Interrupt Priority Level Register 2

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
AITC	0x1004 0038	NIPRIORITY1	Normal Interrupt Priority Level Register 1
AITC	0x1004 003C	NIPRIORITY0	Normal Interrupt Priority Level Register 0
AITC	0x1004 0040	NIVECSR	Normal Interrupt Vector and Status Register
AITC	0x1004 0044	FIVECSR	Fast Interrupt Vector and Status Register
AITC	0x1004 0048	INTSRCH	Interrupt Source Register High
AITC	0x1004 004C	INTSRCL	Interrupt Source Register Low
AITC	0x1004 0050	INTFRCH	Interrupt Force Register High
AITC	0x1004 0054	INTFRCL	Interrupt Force Register Low
AITC	0x1004 0058	NIPNDH	Normal Interrupt Pending Register High
AITC	0x1004 005C	NIPNDL	Normal Interrupt Pending Register Low
AITC	0x1004 0060	FIPNDH	Fast Interrupt Pending Register High
AITC	0x1004 0064	FIPNDL	Fast Interrupt Pending Register Low
CSI	0x8000 0000	CSICR1	CSI Control Register 1
CSI	0x8000 0004	CSICR2	CSI Control Register 2
CSI	0x8000 0008	CSISR	CSI Status Register
CSI	0x8000 000C	CSISTATR	CSI Statistic FIFO Register
CSI	0x8000 0010	CSIRXR	CSI RxFIFO Register
CSI	0x8000 0014	CSIRXCNT	CSI RX Count Register
CSI	0x8000 0018	CSIDEBUG	CSI Debug Register
CSI	0x8000 001C	CSICR3	CSI Control Register 3
BMI	0xA000 0000	BMICTLR1	BMI Control Register 1
BMI	0xA000 0004	BMICTLR2	BMI Control Register 2
BMI	0xA000 0008	BMISTR	BMI Status Register
BMI	0xA000 000C	BMIRXD	BMI RxFIFO
BMI	0xA000_0010	BMITXD	BMI TxFIFO
SDRAMC	0xDF00 0000	SDCTL0	SDRAM 0 Control Register
SDRAMC	0xDF00 0004	SDCTL1	SDRAM 1 Control Register
SDRAMC	0xDF00 0014	MISC	SDRAM Miscellaneous Register
SDRAMC	0xDF00 0018	SDRST	SDRAM Reset Register
WEIM	0xDF00 1000	CS0U	Chip Select 0 Upper Control Register
WEIM	0xDF00 1004	CS0L	Chip Select 0 Lower Control Register
WEIM	0xDF00 1008	CS1U	Chip Select 1 Upper Control Register
WEIM	0xDF00 100C	CS1L	Chip Select 1 Lower Control Register

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
WEIM	0xDF00 1010	CS2U	Chip Select 2 Upper Control Register
WEIM	0xDF00 1014	CS2L	Chip Select 2 Lower Control Register
WEIM	0xDF00 1018	CS3U	Chip Select 3 Upper Control Register
WEIM	0xDF00 101C	CS3L	Chip Select 3 Lower Control Register
WEIM	0xDF00 1020	CS4U	Chip Select 4 Upper Control Register
WEIM	0xDF00 1024	CS4L	Chip Select 4 Lower Control Register
WEIM	0xDF00 1028	CS5U	Chip Select 5 Upper Control Register
WEIM	0xDF00 102C	CS5L	Chip Select 5 Lower Control Register
WEIM	0xDF00 1030	EIM	EIM Configuration Register
PCMCIA	0xDF00 2000	PIPR	PCMCIA input pins register
PCMCIA	0xDF00 2004	PSCR	PCMCIA Status Changed Register
PCMCIA	0xDF00 2008	PER	PCMCIA Enable Register
PCMCIA	0xDF00 200C	PBR0	PCMCIA Base Register 0
PCMCIA	0xDF00 2010	PBR1	PCMCIA Base Register 1
PCMCIA	0xDF00 2014	PBR2	PCMCIA Base Register 2
PCMCIA	0xDF00 2018	PBR3	PCMCIA Base Register 3
PCMCIA	0xDF00 201C	PBR4	PCMCIA Base Register 4
PCMCIA	0xDF00 2028	POR0	PCMCIA Option Register 0
PCMCIA	0xDF00 202C	POR1	PCMCIA Option Register 1
PCMCIA	0xDF00 2030	POR2	PCMCIA Option Register 2
PCMCIA	0xDF00 2034	POR3	PCMCIA Option Register 3
PCMCIA	0xDF00 2038	POR4	PCMCIA Option Register 4
PCMCIA	0xDF00 2044	POFR0	PCMCIA Offset Register 0
PCMCIA	0xDF00 2048	POFR1	PCMCIA Offset Register 1
PCMCIA	0xDF00 204C	POFR2	PCMCIA Offset Register 2
PCMCIA	0xDF00 2050	POFR3	PCMCIA Offset Register 3
PCMCIA	0xDF00 2054	POFR4	PCMCIA Offset Register 4
PCMCIA	0xDF00 2060	PGCR	PCMCIA General Control Register
PCMCIA	0xDF00 2064	PGSR	PCMCIA General Status Register
NANDFC	0xDF00 3E00	NFC_BUFSIZE	Internal SRAM Size
NANDFC	0xDF00 3E02	BLOCK_ADD_LOCK	NAND Flash Block Address for Lock Check
NANDFC	0xDF00 3E04	RAM_BUFFER_ADDRESS	Buffer Number for Page Data Transfer To/From Flash Memory

Table 3-9. Register Map (continued)

Module Name	Address	Register Name	Description
NANDFC	0xDF00 3E06	NAND_FLASH_ADD	NAND Flash Address
NANDFC	0xDF00 3E08	NAND_FLASH_CMD	NAND Flash Command
NANDFC	0xDF00 3E0A	NFC_CONFIGURATION	NFC Internal Buffer Lock Control
NANDFC	0xDF00 3E0C	ECC_STATUS_RESULT	Controller Status/Result of Flash Operation
NANDFC	0xDF00 3E0E	ECC_RSLT_MAIN_AREA	ECC Error Position of Main Area Data Error
NANDFC	0xDF00 3E10	ECC_RSLT_SPARE_AREA	ECC Error Position of Spare Area Data Error
NANDFC	0xDF00 3E12	NF_WR_PROT	Nand Flash Write Protection
NANDFC	0xDF00 3E14	UNLOCK_START_BLK_ADD	Start Address for Write Protection Unlock

Part 2
Core Technology

Chapter 4
ARM9 Platform

4.1 Introduction

The ARM926EJ-S™ is a member of the ARM9 family of general-purpose microprocessors. The ARM926EJ-S processor is targeted at multi-tasking applications where full memory management, high performance, low die size, and low-power are essential for mobile computing.

The ARM926EJ-S processor supports 16-bit and 32-bit ARM Thumb® instruction sets. These instruction sets allows the processor to be utilized in applications that need to balance high performance execution and high code density. The ARM926EJ-S processor includes features for efficient execution of Java byte codes, providing Java performance similar to Just-In-Time (JIT) compilation, but without the associated code overhead.

4.2 Features

The ARM926EJ-S processor supports the ARM debug architecture and includes logic to assist in both hardware and software debug through the JTAG/Multi-ICE port on the i.MX21. The ARM926EJ-S processor has a Harvard cached architecture and provides a complete high-performance processor subsystem, that includes the following:

- ARM9EJ-S integer core
- Memory Management Unit (MMU)
- Separate instruction and data AMBA AHB interfaces
- 16 Kbyte instruction and data caches

The ARM926EJ-S processor implements ARM architecture version 5TEJ (ARMv5TEJ).

Refer to the ARM926EJ-S Technical Reference Manual (Document Number: ARM DDI0198B) for more information.

Chapter 5
ARM926EJ-S Interrupt Controller (AITC)

The ARM926EJ-S™ Interrupt Controller (AITC) is a 32-bit peripheral which collects interrupt requests from up to 64 sources and provides an interface to the ARM926EJ-S core. The AITC includes software controlled priority levels for normal interrupts. The AITC block diagram is shown in Figure 5-1 on page 5-2.

The AITC performs the following functions:

- Supports up to 64 interrupt sources
- Supports fast and normal interrupts
- Selects normal or fast interrupt request for any interrupt source
- Indicates pending interrupt sources via a register for normal and fast interrupts
- Indicates highest priority interrupt number via register (can be used as a table index)
- Independently enable or disable any interrupt source
- Provides a mechanism for software to schedule an interrupt
- Supports up to 16 software controlled priority levels for normal interrupts and priority masking

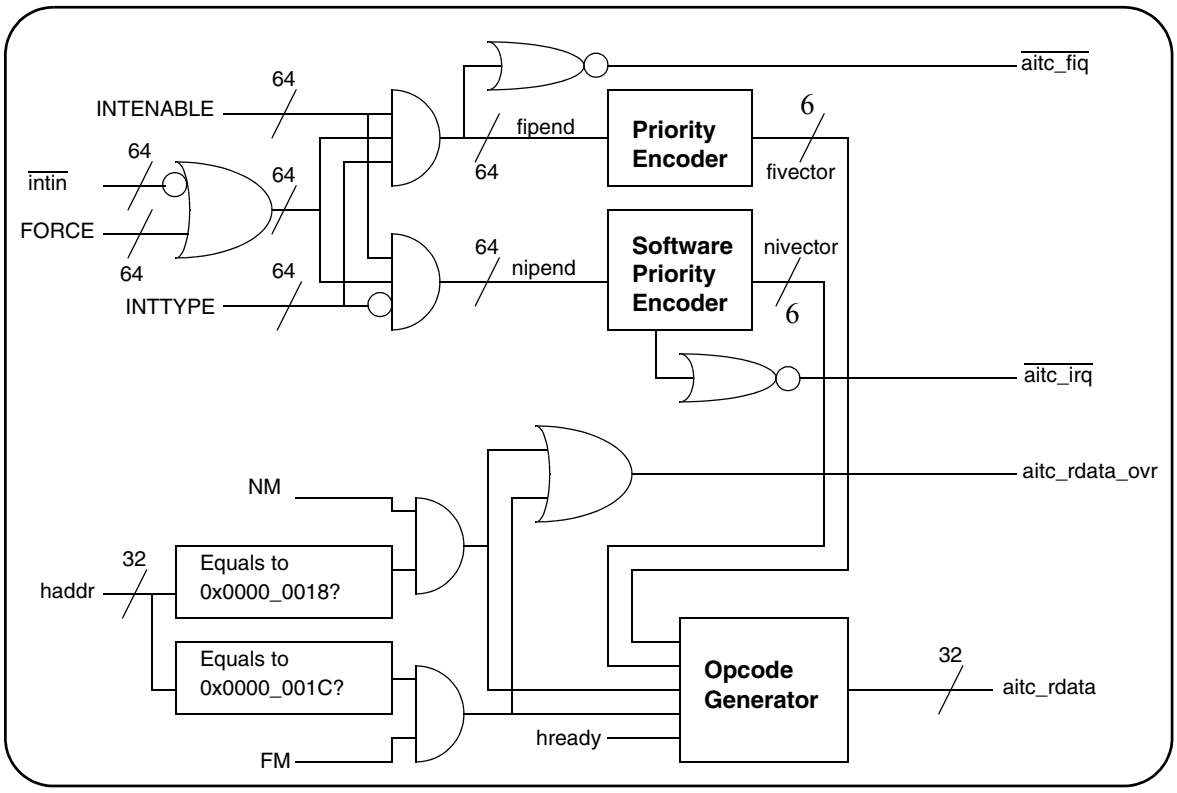


Figure 5-1. AITC Block Diagram

The interrupt controller consists of a set of control registers and associated logic to perform interrupt masking, and priority support of normal interrupts.

The interrupt source registers (INTSRCH / INTSRCL) are a pair of 32-bit status registers with a single interrupt source associated with each of the 64 bits. An interrupt line or set of interrupt lines are routed from each interrupt source to the INTSRCH or INTSRCL register. This allows up to 64 distinct interrupt sources in an implementation. Interrupt requests may be forced to be asserted by way of the interrupt force registers (INTFRCH / INTFRCL). Each bit in this register is logically “OR-ed” with the corresponding hardware request line prior to feeding the INTSRCH or INTSRCL register inputs.

There is a corresponding set of interrupt enable registers (INTENABLEH / INTENABLEL), also 32-bits wide which allow individual bit masking of the INTSRCH / INTSRCL registers. There is also a corresponding set of interrupt type register (INTTYPEH / INTTYPEL) which selects whether an interrupt source will generate a normal or fast interrupt to the ARM926EJ-S core.

There is a corresponding set of normal interrupt pending registers (NIPNDH / NIPNDL) which indicate pending normal interrupt requests. These registers are equivalent to the logical AND of the interrupt source registers (INTSRCH / INTSRCL), the interrupt enable registers (INTENABLEH / INTENABLEL), and the NOT of the interrupt type registers (INTTYPEH / INTTYPEL). (Refer to Figure 5-1 on page 5-2) The NIPNDH / NIPNDL register bits are bit-wise “NOR-ed” together to form the nIRQ signal routed to the ARM926EJ-S core. This core input signal is maskable by the normal interrupt disable bit (I bit) in the processor status register (CPSR). The normal interrupt vector register (NIVECSR) indicates the vector index of highest priority pending normal interrupt.

There is a corresponding set of fast interrupt pending registers (FIPNDH / FIPNDL) which indicate pending fast interrupt requests. These registers are equivalent to the logical AND of the interrupt source registers (INTSRCH / INTSRCL), the interrupt enable registers (INTENABLEH / INTENABLEL), and the interrupt type registers (INTTYPEH / INTTYPEL). (Refer to Figure 5-1 on page 5-2) The FIPNDH / FIPNDL register bits are bit-wise “NOR-ed” together to form the nFIQ signal routed to the ARM926EJ-S core. This core input signal is maskable by the fast interrupt disable bit (F bit) in the CPSR. The fast interrupt vector register (FIVECSR) indicates the vector index of highest priority pending fast interrupt.

All interrupt controller registers are readable and writable in privileged mode only. Writes attempted to read-only registers will be ignored. These registers can be only modified using 32-bit writes.

The INTFRCH / INTFRCL registers are provided for software generation of interrupts. By enabling interrupts for these bit positions, software can force an interrupt request. This register can also be used to debug hardware interrupt service routines by providing an alternate method of interrupt assertion.

The interrupt requests are prioritized in the following sequence:

1. Fast interrupt requests, in order of highest number
2. Normal interrupt requests, in order of highest priority level, then highest source number with the same priority

The AITC provides 16 software controlled priority levels for normal interrupts. Every interrupt can be placed in any priority level. The AITC also provides a normal interrupt priority level mask (NIMASK) which disables any interrupt with a priority level lower than or equal to the mask. If a level 0 normal interrupt and a level 1 normal interrupt are asserted at the same time, the level 1 normal interrupt will be selected assuming that NIMASK has not disabled level 1 normal interrupts. If two level 1 normal interrupts are asserted at the same time, the level 1 normal interrupt with the highest source number will be selected, also assuming that NIMASK has not disabled level 1 normal interrupts.

5.1 Programming Model

The AITC module has 26 registers. All of these registers are single cycle access as the AITC sits on the native bus of the ARM926EJ-S core. This section provides detailed descriptions of the various AITC registers and are summarized in Table 5-1.

Table 5-1. AITC Register Summary

Name		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
INTCNTL (0x1004 0000)	R	0	0	0	0	0	0	ABFLAG	ABFEN	0	NIDIS	FIDIS	NIAD	FIAD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																																
NIMASK (0x1004 0004)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NIMASK				
	W																																
INTENNUM (0x1004 0008)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																											ENNUM				
INTDISNUM (0x1004 000C)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																											DISNUM				
INTENABLEH (0x1004 0010)	R	INTENABLE[63:32]																															
	W																																

Table 5-1. AITC Register Summary (continued)

Name		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
INTENABLEL (0x1004 0014)	R	INTENABLE[31:0]																															
	W																																
INTTYPEH (0x1004 0018)	R	INTTYPE[63:32]																															
	W																																
INTTYPEL (0x1004 001C)	R	INTTYPE[31:0]																															
	W																																
NIPRIORITY7 (0x1004 0020)	R	NIPR63				NIPR62				NIPR61				NIPR60				NIPR59				NIPR58				NIPR57				NIPR56			
	W																																
NIPRIORITY6 (0x1004 0024)	R	NIPR55				NIPR54				NIPR53				NIPR52				NIPR51				NIPR50				NIPR49				NIPR48			
	W																																
NIPRIORITY5 (0x1004 0028)	R	NIPR47				NIPR46				NIPR45				NIPR44				NIPR43				NIPR42				NIPR41				NIPR40			
	W																																
NIPRIORITY4 (0x1004 002C)	R	NIPR39				NIPR38				NIPR37				NIPR36				NIPR35				NIPR34				NIPR33				NIPR32			
	W																																
NIPRIORITY3 (0x1004 0030)	R	NIPR31				NIPR30				NIPR29				NIPR28				NIPR27				NIPR26				NIPR25				NIPR24			
	W																																
NIPRIORITY2 (0x1004 0034)	R	NIPR23				NIPR22				NIPR21				NIPR20				NIPR19				NIPR18				NIPR17				NIPR16			
	W																																
NIPRIORITY1 (0x1004 0038)	R	NIPR15				NIPR14				NIPR13				NIPR12				NIPR11				NIPR10				NIPR9				NIPR8			
	W																																
NIPRIORITY0 (0x1004 003C)	R	NIPR7				NIPR6				NIPR5				NIPR4				NIPR3				NIPR2				NIPR1				NIPR0			
	W																																
NIVECSR (0x1004 0040)	R	NIVECTOR																NIPRILVL															
	W																																
FIVECSR (0x1004 0044)	R	FIVECTOR																															
	W																																
INTSRCH (0x1004 0048)	R	INTIN[63:32]																															
	W																																
INTSRCL (0x1004 004C)	R	INTIN[31:0]																															
	W																																
INTFRCH (0x1004 0050)	R	FORCE[63:32]																															
	W																																
INTFRCL (0x1004 0054)	R	FORCE[31:0]																															
	W																																
NIPNDH (0x1004 0058)	R	NIPEND[63:32]																															
	W																																
NIPNDL (0x1004 005C)	R	NIPEND[31:0]																															
	W																																
FIPNDH (0x1004 0060)	R	FIPEND[63:32]																															
	W																																
FIPNDL (0x1004 0064)	R	FIPEND[31:0]																															
	W																																

5.1.1 Interrupt Control Register (INTCNTL)

The interrupt control register (INTCNTL) controls the interrupts in the AITC. Both normal interrupts and fast interrupts can be enabled to jump directly to the interrupt service routine. For fast interrupts, it may be faster to begin to fast interrupt routine at 0x0000_001C instead of jumping to a service routine.

The vector table can be sourced in high memory, 0xFFFF_FF00 to 0xFFFF_FFFF, or in low memory. If the vector table is located in low memory (MD=1), a register has been provided to control where the vector table is located.

This register is located on the ARM926EJ-S native bus, accessible in 1 cycle, and can only be accessed to in privileged mode. This register can be only modified using 32-bit writes.

INTCNTL — Interrupt Control Register — 0x10040000

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							ABFLAG	ABFEN		NIDIS	FIDIS	NIAD	FIAD			MD
TYPE	rw	rw	rw	rw	rw	rw	r / w1c	rw	rw	rw	rw	rw	rw	r	r	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					POINTER											
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5-2. Interrupt Control Register Description

Name	Description	Settings
Reserved Bits 31–26	Reserved—These bits are reserved and should read 0.	N/A
ABFLAG Bit 25	**Core Arbitration Prioritization Risen Flag**—This status bit indicates that the AITC is asserting the aitc_rise_arb signal to rise the priority level of the ARM core in the bus arbitration logic. This signal is directly tied to the aitc_rise_arb output signal. When the ABFEN bit is cleared, this bit will indicate the current value of the aitc_rise_arb signal to the ARM core. This bit will be set if the nFIQ is asserted and the FIAD bit is set or if the nIRQ is asserted and the NIAD bit is set. When the ABFEN bit is set, this bit will be keep the “1” until written to with a “1”. This bit will remain set on either of the above conditions normally set the ABFLAG bit.	0 = AITC is not affecting the bus arbitration priority levels 1 = AITC is rising ARM core priority level in bus arbitration logic.
ABFEN Bit 24	**ABFLAG Sticky Enable**—This bit controls whether the ABFLAG bit is *sticky*. This allows the arbitration logic to keep the ARM core at the higher priority level until the ABFLAG bit is written.	0 = ABFLAG bit is normal 1 = ABFLAG bit is sticky and requires a write of 1 to clear the bit.
Reserved Bit 23	Reserved—This bit is reserved and should read 0.	N/A
NIDIS Bit 22	**Normal Interrupt Disable**—This bit, when set, disables the generation of the normal interrupt signal. This bit is similar to the I bit of the ARM926EJ-S core.	0 = Does not affect the normal interrupt generation 1 = Disable all normal interrupts

Table 5-2. Interrupt Control Register Description (continued)

Name	Description	Settings
FIDIS Bit 21	**Fast Interrupt Disable**—This bit, when set, disables the generation of the fast interrupt signal. This bit is similar to the F bit of the ARM926EJ-S core.	0 = Does not affect the fast interrupt generation 1 = Disable all fast interrupts
NIAD Bit 20	**Normal Interrupt Arbiter Rise ARM Level**—This bit, when asserted, increases the bus arbitration priority of the ARM core when the normal interrupt signal (nIRQ) is asserted. If an alternate master has ownership of the bus when a normal interrupt occurs, the bus will be given back to the processor core *after* the DMA device has completed its accesses. The NIAD bit does not affect alternate master accesses that are in progress. To prevent an alternate master from accessing the bus during an interrupt service routine, the interrupt flag must not be cleared until the end of the service routine. Another option is to use the ABFEN and ABFLAG bits.	0 = Disregard the normal interrupt flag when evaluating bus requests 1 = Normal interrupt flag increases the bus arbitration priority of the ARM core to decrease the latency of the interrupt service routine.
FIAD Bit 19	**Fast Interrupt Arbiter Rise ARM Level**—This bit functions the same as the NIAD bit except for the fast interrupts (nFIQ).	0 = Disregard the fast interrupt flag when evaluating bus requests 1 = Fast interrupt flag increases the bus arbitration priority of the ARM core to decrease the latency of the interrupt service routine.
Reserved Bits 18–17	Reserved—These bits are reserved and should read 0.	N/A
MD Bit 16	**Interrupt Vector Table Mode**—Indicates whether the interrupt vector is located in high memory or low memory.	0 = Interrupt vector table located in high memory from 0xFFFF_FF00 to 0xFFFF_FFFF 1 = Interrupt vector table located in low memory from POINTER to POINTER+0xFF
Reserved Bits 15–12	Reserved—These bits are reserved and should read 0.	N/A
POINTER Bits 11–2	**Interrupt Vector Table Pointer**—Indicates start of the vector table when in low memory (MD=1).	The value stored in the 10 bits, times 4, must be set greater than or equal to 0x0000_0024 and less than or equal to 0x0000_0F00. Only word-aligned tables are allowed, and two zeros are added in the LSBs when this value is used by the AITC. The value stored here is left shifted by two bits, so the actual table vector can be directly written into the appropriate bits.
Reserved Bits 1–0	Reserved—These bits are reserved and should read 0.	N/A

5.1.2 Normal Interrupt Mask Register (NIMASK)

The normal interrupt mask register (NIMASK) controls the normal interrupt mask level. All normal interrupts with a priority level lower than or equal to the NIMASK will be disabled. The priority level of normal interrupts are determined by the normal interrupt priority level registers (NIPRIORITY7, NIPRIORITY6, NIPRIORITY5, NIPRIORITY4, NIPRIORITY3, NIPRIORITY2, NIPRIORITY1, and NIPRIORITY0). The reset state of this register will not disable any normal interrupts.

Writing all 1's, or -1, to the NIMASK will set the normal interrupt mask to -1 which will not disable any normal interrupt priority levels.

This hardware mechanism can be used to create reentrant normal interrupt routines by disabling lower priority normal interrupts. Refer to Section 5.2.7 for more details on the use of the NIMASK register.

This register is located on the ARM926EJ-S native bus, accessible in 1 cycle, and can only be accessed to in privileged mode. This register can be only modified using 32-bit writes.

NIMASK — Normal Interrupt Mask Register — 0x10040004

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												NIMASK				
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Table 5-3. Normal Interrupt Mask Register Description

Name	Description	Settings
Reserved Bits 31–5	Reserved—These bits are reserved and should read 0.	N/A
NIMASK Bits 4–0	**Normal Interrupt Mask**—Controls normal interrupt mask level. All normal interrupts of priority level lower than or equal to the NIMASK will be disabled.	0 = Disable priority level 0 normal interrupts 1 = Disable priority level 1 and lower normal interrupts ... 0xE (14) = Disable priority level 14 and lower normal interrupts 0xF (15) = Disable all normal interrupts 0x10–0x1F = Do not disable any normal interrupts

5.1.3 Interrupt Enable Number Register (INTENNUM)

The interrupt enable number register (INTENNUM) provides a hardware accelerated enabling of interrupts. Any write to this register will enable one interrupt source. If the 6 LSBs are equal 000000, then interrupt source 0 is enabled. If the 6 LSBs equal 000001, then interrupt source 1 is enabled. And so forth. This register is decoded into a one hot mask which will be logically OR-ed with the INTENABLEH / INTENABLEL register.

This hardware mechanism alleviates the need for an atomic read/modify/write sequence to enable an interrupt source. To enable interrupts 10 and 20, the software need only preform two writes to the AITC: first write 10 to INTENNUM register, then write 20 to INTENNUM register (the order of the writes is irrelevant).

This register is located on the ARM926EJ-S native bus, accessible in 1 cycle, and can only be accessed to in privileged mode. This register can be only modified using 32-bit writes. This register will always read back all 0s.

INTENNUM — Interrupt Enable Number Register — 0x10040008

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											ENNUM					
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	slfclr	slfclr	slfclr	slfclr	slfclr	slfclr
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5-4. Interrupt Enable Number Register Description

Name	Description	Settings
Reserved Bits 31–6	Reserved—These bits are reserved and should read 0.	N/A
ENNUM Bits 5–0	**Interrupt Enable Number**—Writing to this register will enable the interrupt source associated with this value.	0 = Enable interrupt source 0 1 = Enable interrupt source 1 ... 63 = Enable interrupt source 63

5.1.4 Interrupt Disable Number Register (INTDISNUM)

The interrupt disable number register (INTDISNUM) provides a hardware accelerated disabling of interrupts. Any write to this register will disable one interrupt source. If the 6 LSBs are equal 000000, then interrupt source 0 is disabled. If the 6 LSBs equal 000001, then interrupt source 1 is disabled. And so forth. This register is decoded into a one hot mask which will be inverted and logically AND-ed with the INTENABLEH / INTENABLEL register.

This hardware mechanism alleviates the need for an atomic read/modify/write sequence to disable an interrupt source. To disable interrupts 10 and 20, the software need only preform two writes to the AITC: first write 10 to INTDISNUM register, then write 20 to INTDISNUM register (the order of the writes is irrelevant).

This register is located on the ARM926EJ-S native bus, accessible in 1 cycle, and can only be accessed to in privileged mode. This register can be only modified using 32-bit writes. This register will always read back all 0s.

INTDISNUM — Interrupt Disable Number Register — 0x1004000C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											DISNUM					
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	slfclr	slfclr	slfclr	slfclr	slfclr	slfclr
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5-5. Interrupt Disable Number Register Description

Name	Description	Settings
Reserved Bits 31–6	Reserved—These bits are reserved and should read 0.	N/A
DISNUM Bits 5–0	**Interrupt Disable Number**—Writing to this register will disable the interrupt source associated with this value.	0 = Disable interrupt source 0 1 = Disable interrupt source 1 ... 63 = Disable interrupt source 63

5.1.5 Interrupt Enable Register High (INTENABLEH) and Low (INTENABLEL)

The interrupt enable register high (INTENABLEH) and the interrupt enable register low (INTENABLEL) are used to enable pending interrupt requests to the core. Each bit in this register corresponds to an interrupt source available in the system. The reset state of these registers are all interrupts masked.

This register can be updated by various methods: writing directly to the INTENABLEH / INTENABLEL registers, setting bits with the INTENNUM register, or clearing bits with the INTDISNUM register.

These registers are located on the ARM926EJ-S native bus, accessible in 1 cycle, and can only be accessed to in privileged mode. These registers can be only modified using 32-bit writes.

INTENABLEH — Interrupt Enable Register High — 0x10040010

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTENABLE[63:48]															
TYPE	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTENABLE[47:32]															
TYPE	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

INTENABLEL — Interrupt Enable Register Low — 0x10040014

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTENABLE[31:16]															
TYPE	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTENABLE[15:0]															
TYPE	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm	rwm
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5-6. Interrupt Enable Register High and Low Register Description

Name	Description	Settings
INTENABLE Bits 31–0	**Interrupt Enable**—This bit enables the corresponding interrupt source to request a normal interrupt or a fast interrupt. A reset operation clears this bit. If an enable bit is set and the corresponding interrupt source is asserted, the interrupt controller will assert a normal or a fast interrupt request depending on associated INTTYPEH/INTTYPEL setting.	0 = Interrupt disabled 1 = Interrupt enabled and will generate a normal or fast interrupt upon assertion

5.1.6 Interrupt Type Register High (INTTYPEH) and Low (INTTYPEL)

The interrupt type register high (INTTYPEH) and the interrupt type register low (INTTYPEL) are used to select whether a pending interrupt source, when enabled with the INTENABLEH / INTENABLEL, will create a normal interrupt or a fast interrupt to the core. Each bit in this register corresponds to an interrupt source available in the system. The reset state of these registers will cause all enabled interrupt sources to generate a normal interrupt.

These registers are located on the ARM926EJ-S native bus, accessible in 1 cycle, and can only be accessed to in privileged mode. These registers can be only modified using 32-bit writes.

INTTYPEH — Interrupt Type Register High — 0x10040018

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTTYPE[63:48]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTTYPE[47:32]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

INTTYPEL — Interrupt Type Register Low — 0x1004001C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTTYPE[31:16]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTTYPE[15:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5-7. Interrupt Type Register High and Low Register Description

Name	Description	Settings
INTTYPE Bits 31–0	**Interrupt Type**—This bit controls whether the corresponding interrupt source will request a normal interrupt or a fast interrupt. If a INTTYPE bit is set and the corresponding interrupt source is asserted, the interrupt controller will assert a fast interrupt request.	0 = Interrupt source will generate a normal interrupt (nIRQ) 1 = Interrupt source will generate a fast interrupt (nFIQ)

5.1.7 Normal Interrupt Priority Level Registers (NIPRIORITY*n*)

The normal interrupt priority level registers (NIPRIORITY7, NIPRIORITY6, NIPRIORITY5, NIPRIORITY4, NIPRIORITY3, NIPRIORITY2, NIPRIORITY1, and NIPRIORITY0) provide a software controllable prioritization of normal interrupts. Normal interrupts with a higher priority level will preempt normal interrupts with a lower priority. The reset state of these registers forces all normal interrupts to the lowest priority level.

If a level 0 normal interrupt and a level 1 normal interrupt are asserted at the same time, the level 1 normal interrupt will be selected assuming that NIMASK has not disabled level 1 normal interrupts. If two level 1 normal interrupts are asserted at the same time, the level 1 normal interrupt with the highest source number will be selected, also assuming that NIMASK has not disabled level 1 normal interrupts.

These registers are located on the ARM926EJ-S native bus, accessible in 1 cycle, and can only be accessed to in privileged mode. These registers can be only modified using 32-bit writes.

NIPRIORITY7 Normal Interrupt Priority Level Register 7 0x10040020

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NIPR63				NIPR62				NIPR61				NIPR60			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NIPR59				NIPR58				NIPR57				NIPR56			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5-8. Normal Interrupt Priority Level Register 7 Description

Name	Description	Settings
NIPR63 Bits 31–28 **NIPR62** Bits 27–24 **NIPR61** Bits 23–20 **NIPR60** Bits 19–16 **NIPR59** Bits 15–12 **NIPR58** Bits 11–8 **NIPR57** Bits 7–4 **NIPR56** Bits 3–0	**Normal Interrupt Priority Level**—Selects the software controlled priority level for the associated normal interrupt source. These registers do not affect the prioritization of fast interrupt priorities.	0 = Lowest priority normal interrupt ... 15 = Highest priority normal interrupt

NIPRIORITY6 Normal Interrupt Priority Level Register 6 0x10040024

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NIPR55				NIPR54				NIPR53				NIPR52			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NIPR51				NIPR50				NIPR49				NIPR48			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5-9. Normal Interrupt Priority Level Register 6 Description

Name	Description	Settings
NIPR55 Bits 31–28 **NIPR54** Bits 27–24 **NIPR53** Bits 23–20 **NIPR52** Bits 19–16 **NIPR51** Bits 15–12 **NIPR50** Bits 11–8 **NIPR49** Bits 7–4 **NIPR48** Bits 3–0	**Normal Interrupt Priority Level**—Selects the software controlled priority level for the associated normal interrupt source. These registers do not affect the prioritization of fast interrupt priorities.	0 = Lowest priority normal interrupt ... 15 = Highest priority normal interrupt

NIPRIORITY5 Normal Interrupt Priority Level Register 5 0x10040028

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NIPR47				NIPR46				NIPR45				NIPR44			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NIPR43				NIPR42				NIPR41				NIPR40			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5-10. Normal Interrupt Priority Level Register 5 Description

Name	Description	Settings
NIPR47 Bits 31–28 **NIPR46** Bits 27–24 **NIPR45** Bits 23–20 **NIPR44** Bits 19–16 **NIPR43** Bits 15–12 **NIPR42** Bits 11–8 **NIPR41** Bits 7–4 **NIPR40** Bits 3–0	**Normal Interrupt Priority Level**—Selects the software controlled priority level for the associated normal interrupt source. These registers do not affect the prioritization of fast interrupt priorities.	0 = Lowest priority normal interrupt ... 15 = Highest priority normal interrupt

NIPRIORITY4 Normal Interrupt Priority Level Register 4 0x1004002C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NIPR39				NIPR38				NIPR37				NIPR36			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NIPR35				NIPR34				NIPR33				NIPR32			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5-11. Normal Interrupt Priority Level Register 4 Description

Name	Description	Settings
NIPR39 Bits 31–28 **NIPR38** Bits 27–24 **NIPR37** Bits 23–20 **NIPR36** Bits 19–16 **NIPR35** Bits 15–12 **NIPR34** Bits 11–8 **NIPR33** Bits 7–4 **NIPR32** Bits 3–0	**Normal Interrupt Priority Level**—Selects the software controlled priority level for the associated normal interrupt source. These registers do not affect the prioritization of fast interrupt priorities.	0 = Lowest priority normal interrupt ... 15 = Highest priority normal interrupt

NIPRIORITY3 Normal Interrupt Priority Level Register 3 0x10040030

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NIPR31				NIPR30				NIPR29				NIPR28			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NIPR27				NIPR26				NIPR25				NIPR24			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5-12. Normal Interrupt Priority Level Register 3 Description

Name	Description	Settings
NIPR31 Bits 31–28 **NIPR30** Bits 27–24 **NIPR29** Bits 23–20 **NIPR28** Bits 19–16 **NIPR27** Bits 15–12 **NIPR26** Bits 11–8 **NIPR25** Bits 7–4 **NIPR24** Bits 3–0	**Normal Interrupt Priority Level**—Selects the software controlled priority level for the associated normal interrupt source. These registers do not affect the prioritization of fast interrupt priorities.	0 = Lowest priority normal interrupt ... 15 = Highest priority normal interrupt

NIPRIORITY2 Normal Interrupt Priority Level Register 2 0x10040034

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NIPR23				NIPR22				NIPR21				NIPR20			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NIPR19				NIPR18				NIPR17				NIPR16			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5-13. Normal Interrupt Priority Level Register 2 Description

Name	Description	Settings
NIPR23 Bits 31–28 **NIPR22** Bits 27–24 **NIPR21** Bits 23–20 **NIPR20** Bits 19–16 **NIPR19** Bits 15–12 **NIPR18** Bits 11–8 **NIPR17** Bits 7–4 **NIPR16** Bits 3–0	**Normal Interrupt Priority Level**—Selects the software controlled priority level for the associated normal interrupt source. These registers do not affect the prioritization of fast interrupt priorities.	0 = Lowest priority normal interrupt ... 15 = Highest priority normal interrupt

NIPRIORITY1 Normal Interrupt Priority Level Register 1 0x10040038

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NIPR15				NIPR14				NIPR13				NIPR12			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NIPR11				NIPR10				NIPR9				NIPR8			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5-14. Normal Interrupt Priority Level Register 1 Description

Name	Description	Settings
NIPR15 Bits 31–28 **NIPR14** Bits 27–24 **NIPR13** Bits 23–20 **NIPR12** Bits 19–16 **NIPR11** Bits 15–12 **NIPR10** Bits 11–8 **NIPR9** Bits 7–4 **NIPR8** Bits 3–0	**Normal Interrupt Priority Level**—Selects the software controlled priority level for the associated normal interrupt source. These registers do not affect the prioritization of fast interrupt priorities.	0 = Lowest priority normal interrupt ... 15 = Highest priority normal interrupt

NIPRIORITY0 Normal Interrupt Priority Level Register 0 0x1004003C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NIPR7				NIPR6				NIPR5				NIPR4			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NIPR3				NIPR2				NIPR1				NIPR0			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5-15. Normal Interrupt Priority Level Register 0 Description

Name	Description	Settings
NIPR7 Bits 31–28 **NIPR6** Bits 27–24 **NIPR5** Bits 23–20 **NIPR4** Bits 19–16 **NIPR3** Bits 15–12 **NIPR2** Bits 11–8 **NIPR1** Bits 7–4 **NIPR0** Bits 3–0	**Normal Interrupt Priority Level**—Selects the software controlled priority level for the associated normal interrupt source. These registers do not affect the prioritization of fast interrupt priorities.	0 = Lowest priority normal interrupt ... 15 = Highest priority normal interrupt

5.1.8 Normal Interrupt Vector and Status Register (NIVECSR)

The normal interrupt vector and status register (NIVECSR) provides the priority of the highest pending normal interrupt and provides the vector index of the interrupt's service routine. This hardware mechanism replaces the previous necessity for core support of the FF1 command. This number can be directly used as an index into a vector table to select the highest pending normal interrupt source.

This read-only register is located on the ARM926EJ-S native bus, accessible in 1 cycle, and can only be accessed to in privileged mode.

NIVECSR — Normal Interrupt Vector and Status Register — 0x10040040

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NIVECTOR															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NIPRILVL															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 5-16. Normal Interrupt Vector and Status Register Description

Name	Description	Settings
NIVECTOR Bits 31–16	**Normal Interrupt Vector**—Indicates vector index for the highest pending normal interrupt.	-1 = No normal interrupt request pending 0 = Interrupt 0 highest priority pending normal interrupt 1 = Interrupt 1 highest priority pending normal interrupt ... 63 = Interrupt 63 highest priority pending normal interrupt 64+ (not -1) = unused, will not occur
NIPRILVL Bits 15–0	**Normal Interrupt Priority Level**—Indicates the priority level of the highest priority normal interrupt. This number can be written to the NIMASK to disable the current priority normal interrupts to build a reentrant normal interrupt system.	-1 = No normal interrupt request pending 0 = Highest priority normal interrupt is level 0 1 = Highest priority normal interrupt is level 1 ... 15 = Highest priority normal interrupt is level 15 16+ (not -1) = unused, will not occur

5.1.9 Fast Interrupt Vector and Status Register (FIVECSR)

The fast interrupt vector and status register (FIVECSR) provides the vector index for the highest priority active fast interrupt's service routine (the higher the source number of the fast interrupt, the higher the priority level). This hardware mechanism replaces the previous necessity for core support of the FF1 command. This number can be directly used as an index into a vector table to select the highest pending fast interrupt source.

This read-only register is located on the ARM926EJ-S native bus, accessible in 1 cycle, and can only be accessed to in privileged mode.

FIVECSR — Fast Interrupt Vector and Status Register — 0x10040044

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FIVECTOR[31:16]															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FIVECTOR[15:0]															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 5-17. Fast Interrupt Vector and Status Register Description

Name	Description	Settings
FIVECTOR Bits 31–0	**Fast Interrupt Vector**—Indicates vector index for the highest pending fast interrupt.	-1 = No fast interrupt request pending 0 = Interrupt 0 highest pending fast interrupt 1 = Interrupt 1 highest pending fast interrupt ... 63 = Interrupt 63 highest pending fast interrupt 64+ (not -1) = unused, will not occur

5.1.10 Interrupt Source Register High (INTSRCH) and Low (INTSRCL)

The interrupt source register high (INTSRCH) and the interrupt source register low (INTSRCL) are each 32 bits wide. INTSRCH and INTSRCL reflect the status of all interrupt request inputs into the interrupt controller. Unused bit positions always read zero (no request pending). The state of this register out of reset is determined by the peripheral circuits generating the requests; normally, the requests would be inactive.

These read-only registers are located on the ARM926EJ-S native bus, accessible in 1 cycle, and can only be accessed in privileged mode.

INTSRCH — Interrupt Source Register High — 0x10040048

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTIN[63:48]															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTIN[47:32]															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: The state of this register out of reset is determined by the peripheral circuits generating the requests; normally, the requests would be inactive. This read-only register must be accessed with 32-bit reads only.

INTSRCL — Interrupt Source Register Low — 0x1004004C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTIN[31:16]															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTIN[15:0]															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: The state of this register out of reset is determined by the peripheral circuits generating the requests; normally, the requests would be inactive. This read-only register must be accessed with 32 bit reads only.

Table 5-18. Interrupt Source Register High and Low Description

Name	Description	Settings
INTIN Bits 31–0	**Interrupt Source**—Indicates the state of the corresponding hardware interrupt source.	0 = Interrupt source negated 1 = Interrupt source asserted

5.1.10.1 Interrupt Assignments High

Table 5-19 and Table 5-20 show the interrupt assignments of INTSRCH and INTSRCL.

Table 5-19. Interrupt Source High (INTSRCH) Assignment

Name	Interrupt Source Module	Notes
Reserved Bit 31	Reserved	–
Reserved Bit 30	Reserved	–
INT_LCDC Bit 29	LCD Controller (LCDC)	–
INT_SLCDC Bit 28	Smart LCD Controller (SLCDC)	–
Reserved Bit 27	Reserved	–
INT_USBCTRL Bit 26	–	USB OTG Control Interrupt
INT_USBMNP Bit 25	USBOTG	USBOTG HNP Interrupt
INT_USBFUNC Bit 24	USBOTG	USBOTG Function Interrupt
INT_USBHOST Bit 23	USBOTG	USBOTG Host Interrupt
INT_USBDMA Bit 22	USBOTG	USBOTG DMA Interrupt
INT_USBWKUP Bit 21	USBOTG	USBOTG Wakeup Interrupt
INT_EMMAPP Bit 20	eMMA	eMMA Post Processor Interrupt
INT_EMMAPRP Bit 19	eMMA	eMMA Pre Processor Interrupt
INT_EMMADEC Bit 18	eMMA	eMMA Decoder Interrupt
INT_EMMAENC Bit 17	eMMA	eMMA Encoder Interrupt
Reserved Bit 16	Reserved	Reserved for OWIRE
INT_DMACH15 Bit 15	DMA Channel 15	DMA Channel Interrupts
INT_DMACH14 Bit 14	DMA Channel 14	–
INT_DMACH13 Bit 13	DMA Channel 13	–
INT_DMACH12 Bit 12	DMA Channel 12	–
INT_DMACH11 Bit 11	DMA Channel 11	–

Table 5-19. Interrupt Source High (INTSRCH) Assignment (continued)

Name	Interrupt Source Module	Notes
INT_DMACH10 Bit 10	DMA Channel 10	–
INT_DMACH9 Bit 9	DMA Channel 9	–
INT_DMACH8 Bit 8	DMA Channel 8	–
INT_DMACH7 Bit 7	DMA Channel 7	–
INT_DMACH6 Bit 6	DMA Channel 6	–
INT_DMACH5 Bit 5	DMA Channel 5	–
INT_DMACH4 Bit 4	DMA Channel 4	–
INT_DMACH3 Bit 3	DMA Channel 3	–
INT_DMACH2 Bit 2	DMA Channel 2	–
INT_DMACH1 Bit 1	DMA Channel 1	–
INT_DMACH0 Bit 0	DMA Channel 0	–

5.1.10.2 Interrupt Assignments Low

Table 5-20. Interrupt Source Low (INTSRCL) Assignment

Name	Interrupt Source Module	Notes
INT_CSI Bit 31	CMOS Sensor Interface (CSI)	–
INT_BMI Bit 30	Bus Master Interface (BMI)	–
INT_NFC Bit 29	Nand Flash Controller (NFC)	–
INT_PCMCIA Bit 28	PCMCIA/CF Host Controller (PCMCIA)	–
INT_WDOG Bit 27	Watchdog (WDOG)	–
INT_GPT1 Bit 26	General Purpose Timer (GPT1)	–
INT_GPT2 Bit 25	General Purpose Timer (GPT2)	–
INT_GPT3 Bit 24	General Purpose Timer (GPT3)	–
INT_PWM Bit 23	Pulse Width Modulator (PWM)	–

Table 5-20. Interrupt Source Low (INTSRCL) Assignment (continued)

Name	Interrupt Source Module	Notes
INT_RTC Bit 22	Real-Time Clock (RTC)	–
INT_KPP Bit 21	Key Pad Port (KPP)	–
INT_UART1 Bit 20	UART1	–
INT_UART2 Bit 19	UART2	–
INT_UART3 Bit 18	UART3	–
INT_UART4 Bit 17	UART4	–
INT_CSPI1 Bit 16	Configurable SPI (CSPI1)	–
INT_CSPI2 Bit 15	Configurable SPI (CSPI2)	–
INT_SSI1 Bit 14	Synchronous Serial Interface 1(SSI1)	–
INT_SS2 Bit 13	Synchronous Serial Interface (SSI2)	–
INT_I2C Bit 12	I^2C Bus Controller (I^2C)	–
INT_SDHC1 Bit 11	Secure Digital Host Controller (SDHC1)	–
INT_SDHC2 Bit 10	Secure Digital Host Controller (SDHC2)	–
INT_FIRI Bit 9	Fast Infra Red Interface (FIRI)	–
INT_GPIO Bit 8	General Purpose Input/Output (GPIO)	–
Reserved Bit 7	Reserved	–
INT_CSPI3 Bit 6	Configurable SPI (CSPI3)	–
Reserved Bits 5–0	Unused	–

5.1.11 Interrupt Force Register High (INTFRCH) and Low (INTFRCL)

The interrupt force register high (INTFRCH) and the interrupt force register low (INTFRCL) are each 32 bits wide. The interrupt force registers allow for software generation of interrupts for each of the possible interrupt sources for functional or debug purposes. The system level design may reserve one or more sources for software purposes to allow software to self-schedule interrupts by forcing one or more of these “sources” in the appropriate interrupt force register(s).

These registers are located on the ARM926EJ-S native bus, accessible in 1 cycle, and can only be accessed to in privileged mode. These registers can be only modified using 32-bit writes.

INTFRCH — Interrupt Force Register High — 0x10040050

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FORCE[63:48]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FORCE[47:32]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

INTFRCL — Interrupt Force Register Low — 0x10040054

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FORCE[31:16]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FORCE[15:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5-21. Interrupt Force Register High and Low Description

Name	Description	Settings
FORCE Bits 31–0	**Interrupt Source Force Request**—Used to force a request for the corresponding interrupt source.	0 = Standard interrupt operation 1 = Interrupt forced asserted

5.1.12 Normal Interrupt Pending Register High (NIPNDH) and Low (NIPNDL)

The normal interrupt pending register high (NIPNDH) and the normal interrupt pending register low (NIPNDL) are 32-bit wide registers used to monitor the outputs of the enable and masking operations. These registers are actually only a set of buffers; therefore, the reset state of these registers are determined by the normal interrupt enable registers, the interrupt mask register, and the interrupt source registers. The value reflected in these registers is unaffected by the value of the NIMASK register.

These read-only registers are located on the ARM926EJ-S native bus, accessible in 1 cycle, and can only be accessed to in privileged mode.

NIPNDH — Normal Interrupt Pending Register High — 0x10040058

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NIPEND[63:48]															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NIPEND[47:32]															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NIPNDL — Normal Interrupt Pending Register Low — 0x1004005C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NIPEND[31:16]															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NIPEND[15:0]															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5-22. Normal Interrupt Pending Register High and Low Description

Name	Description	Settings
NIPEND Bits 31–0	**Normal Interrupt Pending Bit**—If a normal interrupt enable bit is set and the corresponding interrupt source is asserted, the interrupt controller will assert a normal interrupt request. The normal interrupt pending bits reflect the interrupt input lines which are asserted and are currently enabled to generate a normal interrupt.	0 = No normal interrupt request 1 = Normal interrupt request pending

5.1.13 Fast Interrupt Pending Register High (FIPNDH) and Low (FIPNDL)

The fast interrupt pending register high (FIPNDH) and the fast interrupt pending register low (FIPNDL) are 32-bit wide registers used to monitor the outputs of the enable and masking operations. These registers are actually only a set of buffers; therefore, the reset state of these registers are determined by the fast interrupt enable registers, the interrupt mask register, and the interrupt source registers.

These read-only registers are located on the ARM926EJ-S native bus, accessible in 1 cycle, and can only be accessed to in privileged mode.

FIPNDH — Fast Interrupt Pending Register High — 0x10040060

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FIPEND[63:48]															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FIPEND[47:32]															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FIPNDL — Fast Interrupt Pending Register Low — 0x10040064

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FIPEND[31:16]															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FIPEND[15:0]															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5-23. Fast Interrupt Pending Register High and Low Description

Name	Description	Settings
FIPEND Bits 31–0	**Fast Interrupt Pending Bit**—If a fast interrupt enable bit is set and the corresponding interrupt source is asserted, the interrupt controller will assert a fast interrupt request. The fast interrupt pending bits reflect the interrupt input lines which are asserted and are currently enabled to generate a fast interrupt.	0 = No fast interrupt request 1 = Fast interrupt request pending

5.2 ARM926EJ-S Interrupt Controller Operation

5.2.1 ARM926EJ-S Prioritization of Exception Sources

The ARM926EJ-S core imposes the following priority among the various exceptions:

- Reset (highest priority)
- Data Abort
- Fast Interrupt
- Normal Interrupt
- Prefetch Abort
- Undefined Instruction and SWI (lowest priority)

5.2.2 AITC Prioritization of Interrupt Sources

The AITC module prioritizes the various interrupt sources by source number where higher source numbers have higher priority. Fast interrupt always have higher priority over normal interrupts.

The interrupt requests are prioritized in the following sequence:

1. Fast interrupt requests, in order of highest source number
2. Normal interrupt requests, in order of highest priority level, then in order of highest source number with the same priority level

5.2.3 Assigning and Enabling Interrupt Sources

The interrupt controller provides for flexible assignment of any interrupt source to either of the two core interrupt request inputs. This is done by setting the appropriate bits in the INTENABLEH / INTENABLEL registers and the INTTYPEH / INTTYPEL registers. Usually, interrupt assignment is done once during system initialization and does not affect interrupt latency.

Interrupt assignment is the first of three steps required to enable an interrupt source, and this is done at chip integration. The second step is to program the source to generate interrupt requests. The final step is to enable the interrupt inputs in the core by clearing the normal interrupt disable (I) and/or the fast interrupt disable (F) bits in the program status register (CPSR).

5.2.4 Enabling Interrupt Sources

There are two methods of enabling or disabling interrupts in the AITC. The first method is directly reading the INTENABLEH / INTENABLEL registers, logically OR or BIT CLEAR these registers with a generated masks, then writing back to the INTENABLEH / INTENABLEL registers.

The second method is performing an atomic write to the source number to INTENNUM register. The AITC will decode this 6 bit register and enable one of the 64 interrupt sources. The AITC will automatically generate a “one hot” enable mask and logically OR this mask to the correct INTENABLEH or INTENABLEL register. To disable interrupts is exactly the same except the source number is written to the INTDISNUM register.

5.2.5 Controlling Bus Arbitration With AITC

The AITC has some logic to rise the priority level of the ARM core when either a fast or normal interrupt occurs. When a fast interrupt occurs and the FIAD bit is set, the AITC will assert the aitc_rise_arb signal. When a normal interrupt occurs and the NIAD bit is set, the AITC will assert the aitc_rise_arb signal. This signal will rise the priority level of the ARM core, so that it has priority of all other alternate masters. This signal will not stop the current alternate master transfer instead will prevent/limit future bus arbitration away from the ARM core.

The AITC includes some logic in the ABFLAG and ABFEN bits. The ABFLAG bit indicates the AITC is currently asserted the aitc_rise_arb signal to the bus arbitration logic. The ABFEN bit changes the ABFLAG from a read-only status bit to a “sticky” write-1-to-clear status bit.

5.2.6 Typical Interrupt Entry Sequences

The following is a typical pipeline sequence for the ARM926EJ-S core when a normal interrupt occurs. Assuming single cycle memories, it takes approximately 6 clocks from the acknowledgment of the normal interrupt within the ARM926EJ-S until the first opcode of the interrupt routine is fetched.

Table 5-24. Typical Hardware Accelerated Normal Interrupt Entry Sequence

ADDR	TIME										
	-2	-1	0	1	2	3	4	5	6	7	8
	nIRQ assert		nIRQ ack								
Last ADDR before nIRQ	Fetch	Dec	Exec	Link	Adjust						
+4 / +2		Fetch	Dec								
+8 / +4			Fetch								
0x0000_0018				Fetch	Dec	Exec	Data	Wrbk			
+4					Fetch	Dec					
+8						Fetch					
Vector Table							Vector				
n/a											
nIRQ Routine									Fetch	Dec	Exec
+4										Fetch	Dec
+8											Fetch

The following is a typical pipeline sequence for the ARM926EJ-S core when a fast interrupt occurs, assuming that the FIQ service routine begins at 0x0000_001C and single cycle memories.

Table 5-25. Typical Fast Interrupt Entry Sequence

ADDR	TIME					
	-2	-1	0	1	2	3
	nFIQ assert		nFIQ ack			
Last ADDR before nFIQ	Fetch	Dec	Exec	Link	Adjust	
+4 / +2		Fetch	Dec			
+8 / +4			Fetch			
0x0000_001c				Fetch	Dec	Exec
+4					Fetch	Dec
+8						Fetch

5.2.7 Writing Reentrant Normal Interrupt Routines

The AITC can be used to create a reentrant normal interrupt system. This enables preempting of lower priority level interrupts by higher priority level interrupts. This requires a small amount of software support and overhead.

1. Push the link register (LR_irq) on to the stack (SP_irq)
2. Push the saved status register (SPSR_irq) on to the stack
3. Read the current value of NIMASK and push this value on to the stack
4. Read current priority level via NIVECSR
5. Interrupts of the equal or lesser priority than the current priority level must be masked via the NIMASK register by writing value from NIVECSR
6. Clear the I bit in the ARM926EJ-S core via a MSR / MRS command sequence (now a higher priority normal interrupt can preempt a lower priority one)
   Also change the operating mode of the core to System Mode from IRQ mode
7. Push System Mode link register (LR) on to the stack (SP_user)
8. The traditional interrupt service routine is now included
9. Pop System Mode link register (LR) from the stack (SP_user)
10. Set I bit in the ARM926EJ-S core via a MSR / MRS command sequence (thus disabling all normal interrupts)
    Also change the operating mode of the core to IRQ Mode from System mode
11. Pop the original value of normal interrupt mask and write to the NIMASK register
12. The saved status register must be popped from the stack (SP_irq)
13. The link register must be popped from the stack into the PC
14. Return from nIRQ

NOTE

Steps 1, 2, 13, and 14 are automatically done by most C compilers and are included for completeness.

Chapter 6
Phase-Locked Loop (PLL), Clock and Reset Controller

There are two clock controller modules in the i.MX21. The ARM9 platform clock controller and the PLL Clock Controller module which produces the clock signals used and distributed by the ARM9 platform clock controller. The PLL Clock Controller generates clock signals used throughout the i.MX21 chip and also for external peripherals. The PLL Clock Controller also serves as the interface between the ARM9 platform and the peripherals on the i.MX21.

The primary function of the ARM9 platform clock controller is to take the clock signals from the PLL Clock Controller distribute them to various peripherals on the ARM9 platform. The clock control module contains the logic to turn clocks on or off and determine when the ARM9's clock can be turned off. This module also synchronizes the JTAG interface to the CLK domain. The ARM9 platform clock controller is not a user programmable or accessible module where the PLL Clock Controller is, and is described in this chapter.

6.1 Clock Controller Architecture Block Diagram

There are two DPLLs in the PLL Clock Controller —the MCU/System PLL (MPLL) and the Serial Peripheral PLL (SPLL) use digital and mixed analog/digital circuits to provide clock frequencies for wireless communication and other applications. The MPLL primarily generates the CLK signal to the ARM9 and HCLK (also called System clock) for the system bus and for most of the on-chip peripherals including the clock for LCDC pixel clock, NAND Flash Controller clock and FIRI MIR clock. The SPLL produces the primary clock to the clock dividers for USB OTG, SSI1,SSI2 and FIRI FIR clock.

Both DPLLs (MPLL and SPLL) accept either the output of the FPM or the OSC26M as a source from which to generate the required frequencies for ARM9 platform and/or peripherals using a fractional frequency multiplication method, detailed information about the calculation of the DPLL settings is shown in Section 6.1.4 on page -5.

To produce the wide range of on-chip clock frequencies required by the i.MX21, the core clock generator uses a two-stage phase locked loop. The first stage is a Frequency Pre-Multiplier PLL (FPM) which multiplies the input frequency by a factor of 512. If the input crystal frequency is 32.768 kHz, the premultiplier multiplies it by a factor of 512 to 16.78 MHz (16.384 MHz. for a 32 kHz crystal). The output of the FPM is one of the clock sources for the MPLL and SPLL. Power management of i.MX21 is accomplished by controlling the clock output of the MPLL and SPLL units.

The distribution of clocks in the i.MX21 is shown in the general block diagram of the entire module (Figure 6-1). The signals are described in Table 6-1 on page 6-3. There are two external clock sources to the PLL Clock Controller:

- 32 kHz external crystal
- 26 MHz external source/crystal

Settings in the Clock Source Control Register (CSCR) are used to independently configure the external clock sources applied to the MPLL and SPLL.

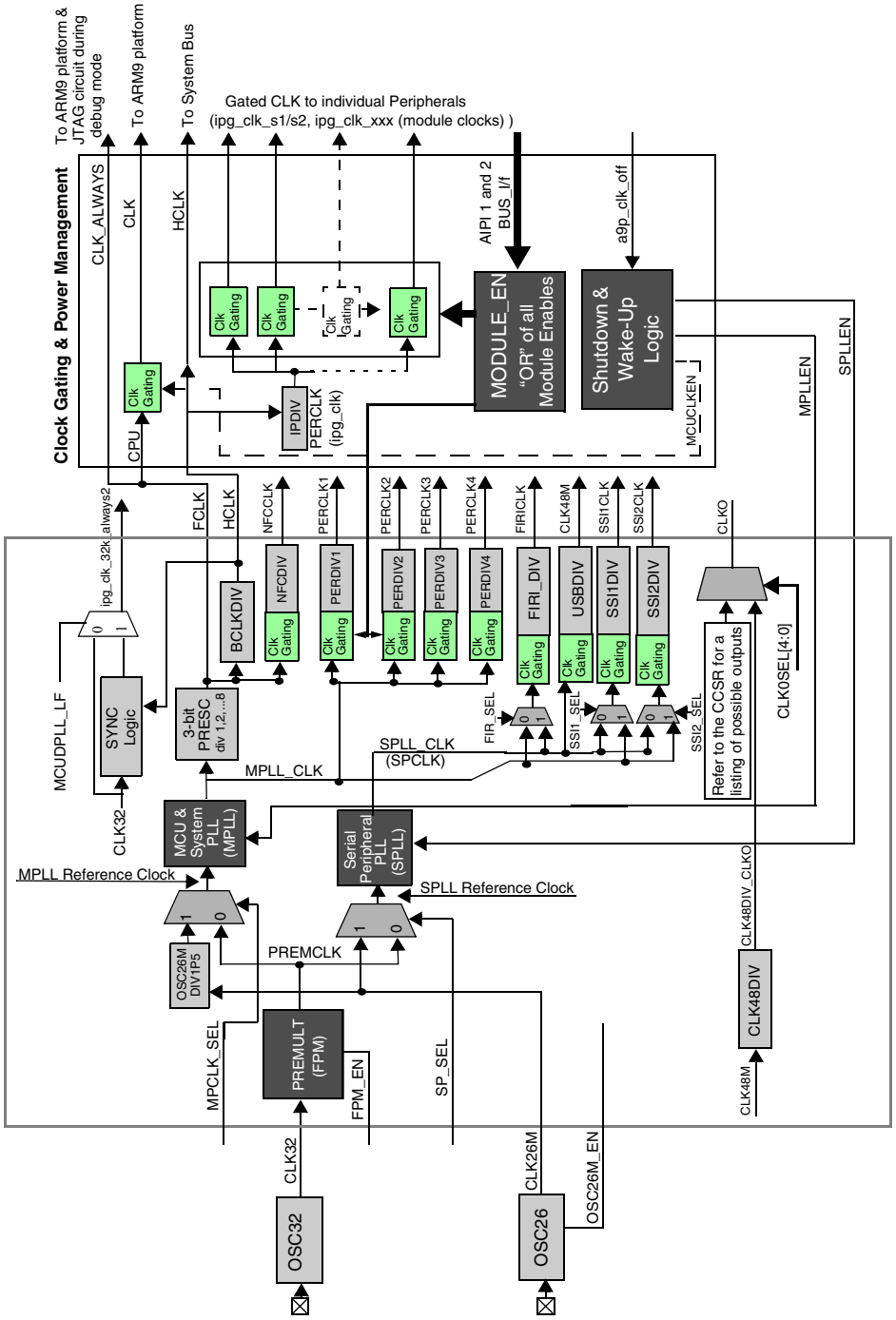


Figure 6-1. i.MX21 Clock Distribution Block Diagram

Table 6-1. PLL Clock Controller Descriptions

Signal Names	Description
CLK_ALWAYS	Fast clock used by the logic inside the ARM9 platform and the JTAG circuit to detect the debug mode and exit Wait-For-Interrupt (WFI) mode. It is required to be always on even when CLK is stopped during debug mode.
CLK	Fast clock used only by ARM9 platform for internal operations such as executing instructions from the cache. It can be gated during doze and stop mode when all the criteria to enter a low power are met.
HCLK	System clock—This signal appears as the BCLK input to the CPU and the HCLK to the system. This is a continuous clock (when the system is not in sleep mode) It can be gated during doze and stop mode when all the criteria to enter a low power are met
HCLKEN	Used to signify the rising edge of CLK that corresponds to the rising edge of HCLK. Used by the ARM9 platform only.
PERCLK	Peripheral Clock to all modules, also known as ipg_clk.
SPCLK	Serial peripheral (SPLL) clock
CLK48M	Divided to 48 MHz clock for the USB OTG Module
SSI1CLK	Divided clock output for the SSI1 module
SSI2CLK	Divided clock output for the SSI2 module
FIRICLK	Divided clock output for the FIRI module
NFCCLK	Divided clock output for the Nand Flash Controller module. In normal operation, the maximum frequency of the NFCCLK will be equal or less than PERCLK (ipg_clk).
PERCLK1	Divided clock output for the peripheral set 1 (UART, Timer, PWM). In normal operation, the maximum frequency of the PERCLK1 will be equal or less than PERCLK (ipg_clk).
PERCLK2	Divided clock output for the peripheral set 2 (SDHC, CSPI). In normal operation, the maximum frequency of the PERCLK2 will be equal or less than PERCLK (ipg_clk).
PERCLK3	Divided clock output for LCDC. In normal operation, the maximum frequency of the PERCLK3 will be equal or less than PERCLK (ipg_clk).
PERCLK4	Divided clock output for the CSI.
CLKO	Selected internal clock output to the CLKO pin.
A9P_CLK_OFF	Control signal from ARM9 platform clock controller
CLK48DIV_CLKO	Clock output from the 3-bit CLK48DIV, used to divide the CLK48M and output on the CLKO pin.
ipg_clk_s1, ipg_clk_s2	Internal clock signals used for AIPI1 module accesses (ipg_clk_s1) or AIPI2 module accesses (ipg_clk_s2). These clocks can be programmed to run continuously or toggle whenever there is an AIPI1 or AIPI2 module access, as determined by the CLOCK_GATING_EN bit in the GPCR in the System Control Chapter.

6.1.1 OSC32K – 32/32.768 kHz Reference Oscillator (Analog)

The i.MX21 can use either a 32 kHz or a 32.768 kHz crystal as the external low frequency source. Throughout this chapter, the low frequency crystal is referred to as the 32 kHz crystal regardless of which frequency of crystal is used. The signal from the external 32 kHz crystal is the source of the CLK32 signal that is sent to the real time clock (RTC). The output of the 32 kHz crystal is also input to the FPM (Frequency PreMultiplier) to produce the 16.384 MHz signal that is input to the DPLLs (it is 16.78 MHz

if a 32.768 kHz crystal is used). The output of the MPLL is sent to the prescaler (PRESC) module to produce the clock (CLK) signal for the ARM core as well as the HCLK for the system bus and module operation.

See Section 6.1.5 on page -5 for more detailed information on phase and frequency jitter specifications using this configuration.

6.1.2 OSC26M – 26 MHz Reference Oscillator (Analog)

The OSC26M oscillator provides a stable frequency reference for the MPLL and SPLL. This oscillator is designed to work in conjunction with a external 26 MHz crystal with the integrated biasing resistor and loading capacitors (19 pF) to generate the 26 MHz reference clock (CLK26M).

6.1.2.1 OSC26M Start-Up Considerations

Startup of the 26 MHz oscillator introduces a system level issue that must be addressed by the system programmer. On power up or after a system reset the 6-bit AGC control field in the Oscillator 26M Control Register (OSC26MCTL) is automatically reset to logic ‘1’ (0x2F) to prevent a condition in which the oscillator may not start. Setting the AGC field to its maximum value applies the maximum supply to the OSC amp, resulting in the highest possible gain. After the oscillator has started up and a valid clock signal is generated, the oscillator peak level must be trimmed using the microprocessor to incrementally trim down the supply level for the oscillator by decrementing the AGC field. Outputs from two comparators generate a 2-bit value in the OSC26M_PEAK field of the OSC26MCTL register to indicate if the current oscillation peak level is too high, too low, or in range.

For a detailed description of the OSC26M startup programming algorithm please refer to Section 6.3.6 on page -19.

6.1.3 High Frequency Clock Source and Distribution

Two DPLLs on the i.MX21 (MPLL and SPLL) are used to generate two separate clock frequencies from either the Frequency Pre-multiplier (FPM) or an external high frequency source (CLK26M). The clock source for each DPLL is individually selected using bits in the Clock Source Control Register (CSCR).

The MCU/System PLL (MPLL) is configured by the MPCTL registers (MPCTL0, MPCTL1) to produce system clock signals that are divided down to output the FCLK (for example 266 MHz) and the HCLK (for example 133 MHz) clock signals. MPLL serves as the clock source for the PERCLK4, PERDIV3, PERDIV2 and PERDIV1. FCLK serves as the clock source for the NFCDIV divider. These dividers produce the clock signals for the following:

- NAND Flash Controller (NFC)
- Peripheral set 1 (PERCLK1): UART, Timer, and PWM
- Peripheral set 2 (PERCLK2): SDHC and CSPI
- LCDC Pixel Clock (PERCLK3)
- CSI (PERCLK4)

The Serial Peripheral PLL (SPLL) is configured by the SPCTL registers (SPCTL0, SPCTL1) and produces the input signals for USBDIV, SSI1DIV, SSI2DIV and FIRIDIV dividers that generate the clock signals for serial peripherals that require special clock frequencies:

- CLK48M—for the USB OTG
- SSI1CLK—Clock signal for SSI1
- SSI2CLK—Clock signal for SSI2
- FIRICLK—Clock signal for FIRI

The FIRIDIV divider divides down the input clock frequency as determined by the setting of the FIRI_DIV setting in the Peripheral Clock Divider Register (PCDR) to provide required frequencies for FIR/MIR modes of operation.

The clock source for the SSI1DIV, SSI2DIv and the FIRIDIV dividers can be the MPLL or SPLL. Source selection is controlled by the respective bits in the Clock Source Control register (CSCR).

6.1.3.1 FPM – Frequency Premultiplier

The Frequency Premultiplier is a digital phase locked loop which accepts the low frequency clock source of 32 kHz or 32.768 kHz and multiplies the frequency by a factor of 512 for input to the MPLL and SPLL.

6.1.4 Output Frequency Calculations

Both DPLLs produce a high frequency clock that exhibits both a low frequency jitter and a low phase jitter. The DPLL output clock frequency (f_{dpll}) is determined by the Equation 6-1:

$$f_{dpll} = 2f_{ref} \bullet \frac{MFI + MFN / (MFD+1)}{PD+1} \qquad \textit{Eqn. 6-1}$$

where:

- f_{ref} is the reference frequency (512 × 32 kHz, 512 × 32.768 kHz, or 26 MHz)
- MFI is an integer part of a multiplication factor (MF)
- MFN is the numerator and MFD is the denominator of the MF
- PD is the predivider factor

6.1.5 DPLL Phase and Frequency Jitter

Spectral purity of the DPLL output clock is characterized by both phase and frequency jitter. Phase jitter is a measure of clock phase fluctuations relative to an ideal clock phase. The output clock also can be skewed relative to the reference clock. Frequency jitter is a measure of clock period fluctuations relative to an ideal clock period. Frequency jitter is calculated as a difference of phase jitter values for adjacent clocks.

DPLL jitter requirements vary according to system configuration. For many stand-alone processors and asynchronous multiprocessor applications, only the frequency jitter value is important (slow phase jitter and clock skew do not affect system performance). In these systems, it is not necessary to adjust the output

clock phase with an input clock phase. The clock generation mode in which slow phase fluctuations are permissible is called Frequency Only Lock (FOL) mode.

Phase error is sometimes important for synchronous applications and sampling analog-to-digital (A/D) and digital-to-analog (D/A) precision converters. The DPLL mode providing minimum phase jitter and skew elimination is Frequency and Phase Lock (FPL) mode. The modes to the MPLL and SPLL in i.MX21 are programmable by setting the corresponding bits in the MPCTL0 and SPCTL0 registers.

6.2 Power Management

The PLL clock controller module is designed with clock control at various stages of clock supply to achieve optimum power savings. The operation of the PLL and clock controller at different stages of power management is described in the following sections.

6.2.1 PLL Operation at Power-Up

The crystal oscillator begins oscillating within several hundred milliseconds of initial power-up. While system reset remains asserted the PLL begins the lockup sequence and locks 1 ms after the crystal oscillator becomes stable. Both DPLLs are enabled on power-up. The system reset is held asserted by the PLL Clock Controller for 300 ms + 14 cycles of the 32 kHz as shown in Figure 6-3 on page 6-32.

6.2.2 PLL Operation at Wake-Up

When the device is awakened from stop mode by a wake-up event, the DPLL locks within 350 µs. The crystal oscillator is always on after initial power-up, so crystal startup time is not a factor. The PLL output clock starts operating as soon as it achieves lock.

6.2.3 i.MX21 Low-Power Modes

i.MX21 provides two power saving modes—Doze mode and Sleep mode:

- In Doze mode, the ARM9 executes a wait for interrupt (WFI) instruction. System clocks are still active.
- In Sleep mode, the ARM9 executes a wait for interrupt (WFI) instruction. The output of the MPLL and SPLL are shut down and only the 32 kHz clock is running.

These modes are controlled by the clock control logic and a sequence of CPU instructions. Most of the peripheral modules can enable or disable the incoming clock signal through clock gating circuitry from the peripheral bus. Each module has a module enable bit which, when disabled, disables the operational clock to the module.

6.2.3.1 Doze Mode

Doze mode is defined as when the ARM9 executes a wait for interrupt instruction. after which the buffered clock supply to the MCU is turned off.

The sequence of operation to set the system to Doze mode:

1. Enable desired interrupts for wake-up from Doze mode
2. Disable watchdog timer interrupt
3. Execute wait-for-interrupt instruction

The ARM9 executes a wait for interrupt instruction if all required conditions are met (no irq, fiq or debug requests pending), the ARM9 PLATFORM generates an A9P_CLK_OFF signal to the PLL Clock Controller Module. The CLK signal to the MCU is immediately turned off when the A9P_CLK_OFF signal goes active. CLK_ALWAYS and system bus (HCLK) remains running. HCLK is required by the Cross Bar Switch within the ARM9 platform for continuous operation of peripheral modules. When an unmasked interrupt event occurs, the CLK signal to the ARM9 is re-enabled.

6.2.3.2 Sleep Mode

Sleep Mode is defined as when all the DPLLs clock outputs are disabled. A sequence of operations and criteria must be satisfied before the system turns off the MPLL and SPLL. The Sleep Mode sequence is initiated when the MPEN bit in the CSCR register is cleared disabling the MPLL. This action also automatically turns off the SPLL.

The sequence of operation to perform to put the system into SLEEP mode:

1. Disable AHB peripherals from bus accesses
2. Enable desired interrupts to be used for system wake-up
3. Disable watchdog timer interrupt
4. Set the required value to the SD_CNT (CSCR register) for shutdown countdown
5. Disable the MPLL by clearing the MPEN bit (CSCR register)
6. Execute wait-for-interrupt instruction

The example of programming setup to enter Sleep mode is as follows:

Example 6-1. Programming Setup for Sleep Mode

```
MRS     r0, CPSR                                ; Enable interrupts
AND     r1, r0, #(ENABLE_IRQ+ENABLE_FIQ+MODE_BITS)
MSR     CPSR_c, r1
LDR     r3, =WDG_BASEADDR                       ; Disable WDG Timer
LDRH    r4, [r3, #0x0]
ORR     r4, r4, #0x00000001
STRH    r4, [r3, #0x0]
LDR     r1, =CRM_BASEADDR                       ; Set SDCNT to '01'
LDR     r2, [r1, #0x0]
ORR     r2, r2, #0x0100_0000
STR     r2, [r1, #0x0]
BIC     r2, r2, #0x00000001                     ; Disable MPEN
STR     r2, [r1, #0x0]
LDR     r1, 0x00000000
MCR     p15, 0, r1, c7, c0, 4                   ; WFI
```

The MPLL and SPLL are turned off when the countdown value in SD_CNT is satisfied. For the MPLL, there are a number of conditions that must be satisfied before the Clock Controller module turns off the

DPLL. The conditions to be satisfied before the PLL Clock Controller actually turns off the MPLL are as follows:

1. Clock Controller Module has successfully mastered the system bus
2. The A9P_CLK_OFF signal from the ARM9 PLATFORM is active
3. SDRAM controller has successfully placed the external SDRAM into self-refresh mode
4. After the above conditions are satisfied the countdown based on the value in the SD_CNT field will be initiated
5. SD_CNT countdown completes

When the conditions listed above are satisfied the MPLL and the SPLL will be turned off. The Frequency Premultiplier (FPM) is also disabled in the Sleep mode. The FPM_EN bit (CSCR register) must not be cleared if the FPM is providing the clock source to the DPLL.

When an unmasked interrupt event occurs, the FPM and then the MPLL are re-enabled and the MPLL enable bit (MPEN) automatically restored to its enable setting. The SPLL is restored to its original state based on the setting of the SPEN bit before Sleep mode. If the SPPLL was not enabled before entering sleep mode the SPPLL will not be enabled.

The total start-up time from Sleep mode is the sum of the FPM lock time and the DPLL lock time.

In Sleep Mode, the i.MX21 retains all RAM data and register configuration values. Data to output terminals is also maintained and thus will continue to sink/source static current.

NOTE

System software must ensure that if there are any clocks being sourced by the i.MX21 processor to external peripherals (for example, CSI_MCLK), then the corresponding PLL must not be turned off. In such cases, the i.MX21 processor must remain in doze mode.

6.2.4 SDRAM Power Modes

When the SDRAM controller (SDRAMC) is enabled, the external SDRAM operates in distributed-refresh mode or in self-refresh mode (as shown in Table 6-2). The SDRAM wake-up latency is approximately 20 system clock cycles (HCLK). The SDRAMC can wake up from self-refresh mode when it is in a SDRAM cycle.

In Doze and Run mode, the Power Down timers within the SDRAMC can be enabled to cause the SDRAM to enter Power Down Mode on detecting no activity. The SDRAMC still controls the refresh and it will take the SDRAM out of the Power Down mode to perform refresh when needed and then put it back into the Power Down Mode. In Power Down mode the clock to the SDRAM is gated off and the CKE pin goes low. In addition since the SDRAM will be in self refresh just when the system get into sleep mode, no bus cycle can access the SDRAM to cause it to exit the self refresh mode. Exit from self refresh mode will happen when the chip will exit the sleep mode and re-enable the MPEN.

Table 6-2. SDRAM/SyncFlash Operation During Power Modes

SDRAM	Run	Doze	Stop
SDRAM	Distributed-refresh, Note 1	Distributed-refresh, [1]	Self-refresh

[1] The Power Down Timers can be enabled and once the timer expires, it will enter low power mode.

6.2.5 Power Management in the PLL Clock Controller

The i.MX21 has a very efficient clock control scheme that enables clocking control of the modules and devices at various stages. Power management in i.MX21 is achieved by controlling the duty cycles of the clock system efficiently. The clocking control scheme is shown in Table 6-3.

Table 6-3. Power Management in the Clock Controller

Device/Signal	Shut-Down Conditions	Wake-Up Conditions
MPLL	When 0 is written to the MPEN bit and the PLL shut-down count times out (for details see the SD_CNT settings in Table 6-5).	When $\overline{\text{IRQ}}$ or $\overline{\text{FIQ}}$ is asserted
SPLL	When 0 is written to the SPEN bit.	When the SPEN bit is set to 1.
FPM	When 0 is written to the FPMEN bit.	When the FPMEN bit is set to 1.
CLK32	Continuously running.	Continuously running

Most modules in the i.MX21 have a module enable bit assigned which must be enabled before the module is active. Enabling the module enables the clock source for the module to be provided for its main operations. The clock input to the dividers from the SPLL is also controlled separately in the same manner.

6.3 Programming Model

The PLL Clock Controller module includes six user-accessible 32-bit registers. Table 6-4 summarizes these registers and their addresses.

Table 6-4. PLL Clock Controller Module Register Summary

Description	Name	Address
Clock Source Control Register	CSCR	0x10027000
MPLL Control Register 0	MPCTL0	0x10027004
MPLL Control Register 1	MPCTL1	0x10027008
SPLL Control Register 0	SPCTL0	0x1002700C
SPLL Control Register 1	SPCTL1	0x10027010
Oscillator 26M Register	OSC26MCTL	0x10027014
Peripheral Clock Divider Register 0	PCDR0	0x10027018
Peripheral Clock Divider Register 1	PCDR1	0x1002701C
Peripheral Clock Control Register 0	PCCR0	0x10027020
Peripheral Clock Control Register 1	PCCR1	0x10027024
Clock Control Status Register	CCSR	0x10027028
PMOS Switch Control Register	PMCTL	0x1002702C
PMOS Switch Counter Register	PMCOUNT	0x10027030
Wakeup Guard Mode Control Register	WKGDCTL	0x10027034

6.3.1 Clock Source Control Register (CSCR)

The Clock Source Control Register controls the various clock sources to the internal modules of the i.MX21.

CSCR — Clock Source Control Register — Addr 0x10027000

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PRESC			USB_DIV			SD_CNT			SPLL_ RESTART	MPLL_ RESTART	SSi2_SEL	SSI1_SEL	FIR_SEL	SP _SEL	MCU_ SEL
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	r	rw	rw	rw	rw	rw	rw	rw
RESET	0	1	1	1	0	1	1	1	0	0	0	1	1	0	0	0
	0x7718															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			BCLKDIV				IPDIV					OSC26M_ DIV1P5	OSC26M_ DIS	FPM_EN	SPEN	MPEN
TYPE	r	r	rw	rw	rw	rw	rw	r	r	r	r	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1
	0x0607															

Table 6-5. Clock Source Control Register Description

Name	Description	Settings
PRESC Bit 31–29	**Prescaler**—Defines the MPU PLL clock 3-bit prescaler.	0 = Prescaler divides by 1 1 = Prescaler divides by 2 2 = Prescaler divides by 3 3 = Prescaler divides by 4 4 = Prescaler divides by 5 5 = Prescaler divides by 6 6 = Prescaler divides by 7 7 = Prescaler divides by 8
USB_DIV Bits 28–26	**USB Clock Divider**—Contains the 3-bit integer divider value for the generation of the CLK48M.	000 = SPLL_CLK divided by 1 001 = SPLL_CLK divided by 2 ... 111 = SPLL_CLK divided by 8
SD_CNT Bits 25–24	**Shut-Down Control**—Contains the value that determines the duration of the DPLL clock output before it goes off after a 0 is written to the MPEN or SPEN bit. **Note:** The power controller requests the bus before SPLL shutdown. Any unmasked interrupt event enables the MPLL.	00 = DPLL shuts down after next rising edge of CLK32 is detected and the current bus cycle is completed. A minimum of 16 HCLK cycles is occurs after writing 0 to MPEN bit. 01 = DPLL shuts down after the second rising edge of CLK32 is detected and the current bus cycle is completed. 10 = DPLL shuts down after the third rising edge of CLK32 is detected and the current bus cycle is completed. 11 = DPLL shuts down after forth rising edge of CLK32 is detected and the current bus cycle is completed.
Reserved Bit 23	Reserved—This bit is reserved and should read 0.	

Table 6-5. Clock Source Control Register Description (continued)

Name	Description	Settings
SPLL_RESTART Bit 22	**SPLL Restart**—Restarts the SPLL at new assigned frequency. SPLL_RESTART self-clears after 1 (min) or 2 (max) cycles of CLK32.	0 = No Effect 1 = Restarts SPLL at new frequency
MPLL_RESTART Bit 21	**MPLL Restart**—Restarts the MPLL at a new assigned frequency. MPLL_RESTART self-clears after 1 (min) or 2 (max) cycles of CLK32.	0 = No Effect 1 = Restarts MPLL at new frequency
SSI2_SEL Bit 20	**SSI2 Baud Source Select**—Selects the clock source to the SSI2 fractional divider (SSI2_DIV).	0 = Source clock to SSI2 fractional divider from SPLL 1 = Source clock to SSI2 fractional divider from MPLL
SSI1_SEL Bit 19	**SSI1 Baud Source Select**—Selects the clock source to the SSI1 fractional divider (SSI1_DIV).	0 = Source clock to SSI1 fractional divider from SPLL 1 = Source clock to SSI1 fractional divider from MPLL
FIR_SEL Bit 18	**FIR and MIR Select**—Selects the clock source to the FIR module divider FIRI_DIV during FIR or MIR mode.	0 = Source clock to FIRI baud clock divider from MPLL 1 = Source clock to FIRI baud clock divider from SPLL
SP_SEL Bit 17	**SPLL Select**—Selects the clock source of the SPLL input. When set, the external high frequency clock input is selected.	0 = Clock source is the internal premultiplier 1 = Clock source is the external high frequency clock
MCU_SEL Bit 16	**MPLL Select**—Selects the clock source of the MPLL input. When set, the external high frequency clock input is selected.	0 = Clock source is the internal premultiplier 1 = Clock source is the external high frequency clock
Reserved Bits 15–14	Reserved—These bits are reserved and should read 0.	
BCLKDIV Bits 13–10	**System Bus Clock Divider**—Contains the 4-bit integer divider value for the generation of the HCLK.	0000 = BCLK divided by 1 0001 = BCLK divided by 2 ... 1111 = BCLK divided by 16
IPDIV Bits 9	**Peripheral Clock Divider**—Contains the 1-bit integer divider value for the generation of the peripheral clock ipg_clk (PERCLK) for the system peripherals. User should not set this bit to 0 in normal operation	0 = only for test purpose 1 = HCLK divided by 2
Reserved Bits 8–5	Reserved—These bits are reserved and should read 0.	
OSC26M_DIV1P5 Bit 4	**Oscillator 26M Divide**—Divides osc26m output by 1 or 1.5.	0 = osc26m output divide by 1 (default) 1 = osc26m output divide by 1.5
OSC26M_DIS Bit 3	**Oscillator Disable**—Disables the internal (chip inside) 26 MHz oscillator circuit when this bit is set to 1.	0 = Enable the internal 26 MHz oscillator circuit 1 = Disable the internal 26 MHz oscillator circuit
FPM_EN Bit 2	**Frequency Premultiplier Enable**—Enables the FPM when set. When clear, the FPM is disabled. The bit is set automatically on system reset. When the software writes a 0 to this bit, the FPM is shut down immediately. This bit must remain at 1 prior and during sleep mode if the FPM is providing the source to the DPLL.	0 = Disable the frequency premultiplier circuit 1 = Enable the frequency premultiplier circuit

Table 6-5. Clock Source Control Register Description (continued)

Name	Description	Settings
SPEN Bit 1	**Serial Peripheral PLL Enable**— Enables/disables the SPLL. When software writes 0 to SPEN, the SPLL shuts down after timeout determined by SD_CNT. SPEN sets automatically when SPLLEN asserts, and on system reset.	0 = Serial Peripheral PLL disabled 1 = Serial Peripheral PLL enabled
MPEN Bit 0	**MPLL Enable**—Enables/disables the MPLL. When software writes 0 to MPEN, the MPLL shuts down after SDCNT timeout. MPEN sets automatically when MPLLEN asserts, and on system reset.	0 = MCU and Serial PLL disabled 1 = MCU and Serial PLL enabled

NOTE

When the PRESC and BCLKDIV are to be modified at the same time, it must be performed in 2 programming steps; first step is to program the BCLKDIV followed by PRESC.

6.3.2 MPLL Control Register 0 (MPCTL0)

The MCU & System PLL Control Register 0 (MPCTL0) is a 32-bit register that controls the operation of the MPLL. The MPCTL0 control bits are described in the following sections. The following is the recommended procedure for changing the MPLL settings:

1. Program the desired values of PD, MFD, MFI, and MFN into the MPCTL0.
2. Set the MPLL_RESTART bit in the CSCR (it will self-clear).
3. New MPLL settings will take effect.
4. The new PLL clock output is valid upon the assertion of the DPLL lock flag.

MPCTL0 — MCU & System PLL Control Register 0 — Addr 0x10027004

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CPLM		PD				MFD									
TYPE	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	0x0007															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			MFI				MFN									
TYPE	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1
	0x1C07															

Table 6-6. MPLL Register 0 Description

Name	Description	Settings
CPLM Bit 31	**Phase Lock Mode**—The DPLL operates in the Frequency Only Lock mode (FOL) when the CPLM bit is cleared, and in Frequency and Phase Lock mode (FPL) when the bit is set. The FPL mode can be used for both an integer and fractional multiplication factor, but phase skew elimination is accomplished only for integer MF.	0 = FOL 1 = FPL
Reserved Bit 30	Reserved—This bit is reserved and should read 0.	
PD Bits 29–26	**Predivider Factor**—Defines the predivider factor (PD) applied to the MPLL input frequency. PD is an integer between 0 and 15 (inclusive). PD is chosen to ensure that the resulting output frequency remains within the specified range. When a new value is written into PD bits, the MPLL loses its lock; after a time delay, the MPLL re-locks. The output of the MPLL is determined by Equation 6-1.	0000 = 0 0001 = 1 … 1111 = 15
MFD Bits 25–16	**Multiplication Factor (Denominator Part)**—Defines the denominator part of the BRM value for the MF. When a new value is written into the MFD bits, the MPLL loses its lock; after a time delay, the PLL re-locks.	0x000 = Reserved 0x001 = 1 … 0x3FF = 1023
Reserved Bits 15–14	Reserved—These bits are reserved and should read 0.	
MFI Bits 13–10	**Multiplication Factor (Integer)**—Defines the integer part of the BRM value for the MF. The MFI is encoded so that MFI < 5 results in MFI = 5. When a new value is written into the MFI bits, the PLL loses its lock: after a time delay, the PLL re-locks. The VCO oscillates at a frequency determined by Equation 6-1. Where PD is the division factor of the predivider, MFI is the integer part of the total MF, MFN is the numerator of the fractional part of the MF, and MFD is its denominator part. The MF is chosen to ensure that the resulting VCO output frequency remains within the specified range.	0000 = 0 0101 = 5 0110 = 6 ... 1111 = 15
MFN Bits 9–0	**Multiplication Factor (Numerator)**—Defines the numerator of the BRM value for the MF. When a new value is written into the MFN bits, the MPLL loses its lock; after a time delay, the PLL re-locks.	0x000 = 0 0x001 = 1 ... 0x3FE = 1022 0x3FF = Reserved

Recommended settings for the MPLL and SPLL, which produces the least amount of signal jitter, are shown in Table 6-7.

Table 6-7. Recommended Settings for Frequency Stability

Ref Frequency	Target Frequency	MFI	MFN	MFD	PD	MPCTL0/SPCTL0 Setting	Actual Calculated Frequency
32.768kHz	288MHz	8	365	626	0	0x0272216D	287.9687378
	266MHz	7	115	123	0	0x007B1C73	266.0000537
	258MHz	7	368	532	0	0x02141D70	258.0480615
	240MHz	7	137	897	0	0x03811C89	240.00013
	240MHz	14	191	625	0	0x067138BF	239.9999509
32.000kHz	288MHz	8	101	127	0	0x007F2065	288
	266MHz	8	2	16	0	0x00102002	265.9990588
	258MHz	7	7	7	0	0x00071C07	258.048
	240MHz	7	83	255	0	0x00FF1c53	240
26MHz	288MHz	5	7	12	0	0x000C1407	288
	266MHz	5	3	25	0	0x00191403	266
	240MHz	4	16	25	0	0x00191010	240
26/1.5MHz	288MHz	7	35	51	0	0x00331C23	266

$$f_{dpll} = 2f_{ref} \bullet \frac{MFI + MFN / (MFD+1)}{PD+1}$$

6.3.2.1 MPLL Control Register 1

The MCU & System PLL Control Register 1 (MPCTL1) is a 32-bit register that directs the operation of the on-chip MCU PLL. The MPCTL1 control bits are described in Table 6-8.

MPCTL1 — MCU & System PLL Control Register 1 — Addr 0x10027008

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LF									BRMO						
TYPE	r	r	r	r	r	r	r	r	r	rw	r	r	r	r	r	r
RESET	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x8000															

Table 6-8. MCU & System PLL Control Register 1 Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
LF Bit 15	**Lock Flag**—Indicates whether the MPLL is locked. When set, the MPLL clock output is valid. When cleared, the MPLL clock output remains at logic high.	0 = MPLL is not locked 1 = MPLL is locked
Reserved Bits 14–7	Reserved—These bits are reserved and should read 0.	
BRMO Bit 6	**BRM Order**—Controls the BRM order which affect the jitter performance of the MPLL. The first order BRM is used if a MF fractional part is both more than 1/10 and less than 9/10. In other cases, the second order BRM is used. The BRMO bit is cleared by a hardware reset. A delay of reference cycles is required between two write accesses to BRMO.	0 = BRM contains first order 1 = BRM contains second order
Reserved Bits 5–0	Reserved—These bits are reserved and should read 0.	

6.3.3 Programming the Serial Peripheral PLL (SPLL)

One of the clock frequencies that the SPLL generates is for the USB OTG module (CLK48M). Its frequency is set to 48 MHz using the SPLL control registers assuming a default input clock frequency 16.384 MHz. This input clock frequency assumes a 32 kHz crystal input. The predivider/multiplier output depends on the input clock frequency. Recommended settings are provided for the Serial Peripheral PLL as shown in Table 6-9.

Table 6-9. Serial PLL Multiplier Factor

PLL Input Frequency	Premultiplier	PD	MFD	MFI	MFN	PLL Output Frequency	USBDIV[2:0]	USB OTG Clock Frequency
32 kHz	16.384 MHz	0	255	7	83	240 MHz	100	48 MHz
32 kHz	16.384 MHz	0	127	8	101	288 MHz	101	48 MHz
32.768 kHz	16.778 MHz	0	897	7	137	240.00013 MHz	100	48.000026 MHz
32.768 kHz	16.778 MHz	0	944	8	551	288 MHz	101	48 MHz
26 MHz	–	0	12	5	7	288 MHz	101	48 MHz

6.3.4 SPLL Control Register 0 (SPCTL0)

The Serial Peripheral PLL Control Register 0 (SPCTL0) is a 32-bit register that controls the operation of the SPLL. The SPCTL0 control bits are described in the following sections. The following is a procedure for changing the Serial Peripheral PLL settings:

1. Program the desired values of PD, MFD, MFI, and MFN into the SPCTL0.
2. Set the SPLL_RESTART bit in the CSCR (it will self-clear).
3. New PLL settings will take effect.
4. The new PLL clock output is valid upon the assertion of the DPLL lock flag.

SPCTL0 — Serial Peripheral PLL Control Register 0 — Addr 0x1002700C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CPLM		PD				MFD									
TYPE	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
	0x807F															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			MFI				MFN									
TYPE	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	1
	0x2065															

Table 6-10. SPLL Control Register 0 Description

Name	Description	Settings
CPLM Bit 31	**Phase Lock Mode**—The DPLL operates in the Frequency Only Lock mode (FOL) when the CPLM bit is cleared, and in Frequency and Phase Lock mode (FPL) when the bit is set. The FPL mode can be used for both an integer and fractional multiplication factor, but phase skew elimination is accomplished only for integer MF.	0 = FOL 1 = FPL
Reserved Bit 30	Reserved—These bit is reserved and should read 0.	
PD Bits 29–26	**Predivider Factor**—Defines the predivider factor (PD) that is applied to the PLL input frequency. PD is an integer between 0 and 15 (inclusive). The SPLL oscillates at a frequency determined by Equation 6-1. The PD is chosen to ensure that the resulting VCO output frequency remains within the specified range. When a new value is written into the PD bits, the SPLL loses its lock: after a time delay, the SPLL re-locks.	0000 = 0 0001 = 1 … 1111 = 15
MFD Bits 25–16	**Multiplication Factor (Denominator Part)**—Defines the denominator part of the BRM value for the MF. When a new value is written into the MFD9–MFD0 bits, the PLL loses its lock: after a time delay, the PLL re-locks.	0x000 = Reserved 0x001 = 1 … 0x3FF = 1023
Reserved Bits 15–14	Reserved—These bits are reserved and should read 0.	

Table 6-10. SPLL Control Register 0 Description (continued)

Name	Description	Settings
MFI Bits 13–10	**Multiplication Factor (Integer Part)**—Defines the integer part of the BRM value for the MF. The MFI is decoded so that MFI < 5 results in MFI = 5. The SPLL oscillates at a frequency determined by Equation 6-1. Where PD is the division factor of the predivider, MFI is the integer part of the total MF, MFN is the numerator of the fractional part of the MF, and MFD is the denominator part of the MF. The MF is chosen to ensure that the resulting VCO output frequency remains within the specified range. When a new value is written into the MFI bits, the SPLL loses its lock; after a time delay, the SPLL re-locks.	0000–0101 = 5 0110 = 6 ... 1111 = 15
MFN Bits 9–0	**Multiplication Factor (Numerator Part)**—Defines the numerator part of the BRM value for the MF. When a new value is written into the MFN bits, the PLL loses its lock; after a time delay, the PLL re-locks.	0x000 = 0 0x001 = 1 ... 0x3FE = 1022 0x3FF = Reserved

6.3.5 SPLL Control Register 1

The Serial PLL control register 1 (SPCTL1) is a 32-bit read/write register that directs the operation of the SPLL. The SPCTL1 control bits are described in this section.

SPCTL1 — Serial Peripheral PLL Control Register 1 — Addr 0x10027010

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LF									BRMO						
TYPE	r	r	r	r	r	r	r	r	r	rw	r	r	r	r	r	r
RESET	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x8000															

Table 6-11. Serial Peripheral PLL Control Register 1 Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
LF Bit 15	**Lock Flag**—Indicates whether the SPLL is locked. When set, the S PLL clock output is valid. When cleared, the SPLL clock output remains at logic high.	0 = SPLL is not locked 1 = SPLL is locked
Reserved Bits 14–7	Reserved—These bits are reserved and should read 0.	
BRMO Bit 6	**BRM Order Bit**—Controls the BRM order which affects the jitter performance of the DPLL. The first order BRM is used if a MF fractional part is both more than 1/10 and less than 9/10. In other cases, the second order BRM is used. The BRMO bit is cleared by a hardware reset.	0 = BRM has first order 1 = BRM has second order
Reserved Bits 5–0	Reserved—These bits are reserved and should read 0.	

6.3.6 Oscillator 26M Register

This register is use to program the 26 MHz oscillator test modes as well as the gain control. Trimming of the oscillator is necessary only on initial power up; the trim may be stored in Flash for future reference.

OSC26MCTL — Oscillator 26M Control Register — Addr 0x10027014

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																OSC26M_PEAK
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			AGC													
TYPE	r	r	rw	rw	rw	rw	rw	rw	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
	0x3F00															

Table 6-12. Oscillator 26M Control Register Description

Name	Description	Settings
Reserved Bits 31–18	Reserved—These bits are reserved and should read 0.	
OSC26M_PEAK Bits 17–16	**OSC26M_PEAK**—These bits are the status values from the oscillator indicates the current amplitude.	00 = Amplitude in desired operating range. 01 = Amplitude too low; trim higher. 10 = Amplitude too high; trim lower. 11 Invalid state.
AGC Bits 13–8	**Automatic Gain Control**—These bits set the magnitude of the crystal oscillations based on the OSC26M_PEAK status. The optimum setting of these bits is determined using the algorithm in Section 6.3.6.1 on page -19.	
Reserved Bits 7–0	Reserved—These bits are reserved and should read 0.	

6.3.6.1 Adjusting the 26 MHz Oscillator Trim

To ensure a proper startup of the 26 MHz oscillator on power-up or system reset use the following steps to determine the optimum trim of the oscillator AGC.

26 MHz Oscillator Trim Programming Algorithm

1. At power up or system reset, OSC26M_AGC[5:0] bits in the OSC26MCTL register are reset to logic 1 (done in hardware, no software interaction required).
2. Read the peak amplitude value in bits OSC26M_PEAK[1:0] in the OSC26MCTL register.

3. If the amplitude is not in the desired range, adjust by decrementing the OSC26M_AGC[5:0] by 1 count.
4. Wait at least 30.5 us (1 cycle of 32 kHz clock) for system to update OSC26M_PEAK bits.
5. Repeat steps 2 to 4 until trimmed in desired range.
6. Decrement 4 additional counts to provide a margin of error for temperature drift.
7. Store trim value in an external memory—that is, Flash, for future use.

It is suggested that the proceeding algorithm be run to determine the optimum AGC setting. Once this is done on power-up or system reset the software must read the trim value from the external memory and write it to the OSC26M_AGC[5:0].

6.3.7 Peripheral Clock Divider Register 0 (PCDR0)

The Peripheral Clock Divider Register 0 (PCDR0) contains the divider values for the peripheral clock dividers in the PLL Clock Controller. Peripherals in i.MX21 requires special clock frequency which is divided down from the MPLL and the SPLL clock output. Each of these peripheral modules receive their clock input from the respective clock divider. These modules will still have the clock gating scheme as with other modules for power saving advantages. Table 6-15 lists the clock sources associated with the i.MX21 peripherals given in the PCDR0.

PCDR0 — Peripheral Clock Divider Register 0 — Addr 0x10027018

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SSI2DIV										SSI1DIV					
TYPE	rw	rw	rw	rw	rw	rw	r	r	r	r	rw	rw	rw	rw	rw	rw
RESET	0	1	1	0	0	1	0	0	0	0	0	1	1	0	0	1
	0x6419															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NFCDIV								CLKO_48MDIV			FIRI_DIV				
TYPE	rw	rw	rw	rw	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1
	0x3007															

Table 6-13. Peripheral Clock Divider Register 0 Description

Name	Description	Settings
SSI2DIV Bits 31–26	**SSI2 Baud Clock Divider—**Contains the 6-bit fractional divider that produces the clock for the SSI2CLK clock signal for the peripherals.	000000, 000001 = Divide by 62 Others = clkin / (SSI2DIV[high:1] + 0.5 * SSI2DIV[0])
Reserved Bits 25–22	These bits are reserved and should read 0.	

Table 6-13. Peripheral Clock Divider Register 0 Description (continued)

Name	Description	Settings
SSI1DIV Bits 21–16	**SSI1 Baud Clock Divider—**Contains the 6-bit fractional divider that produces the clock for the SSI1CLK clock signal for the peripherals.	000000, 000001 = Divide by 62 Others = clkin / (SSI1DIV[high:1] + 0.5 * SSI1DIV[0])
NFCDIV Bits 15–12	**Nand Flash Controller Clock Divider—**Contains the 4-bit divider that produces the clock for the NFCCLK clock signal for the Nand Flash Controller.	0000 = Divide by 1 0001 = Divide by 2 … 1111 = Divide by 16
Reserved Bits 11–8	These bits are reserved and should read 0.	
CLKO_48MDIV Bits 7–5	**Clock Out 48M Clock Divider 1**—Contains the 3-bit divider that divides the CLK48M to output at CLKO pin, where the output of the divider is labeled CLK48DIV_CLKO.	000 = Divide by 1 001 = Divide by 2 … 111 = Divide by 8
FIRI_DIV Bits 4–0	**FIRI Divider**—Contains the 5-bit integer divider that produces the FIRICLK clock signal for the peripherals during MIR/FIR mode.	00000 = Divide by 1 00001 = Divide by 2 … 11111 = Divide by 32

6.3.8 Peripheral Clock Divider Register 1 (PCDR1)

The Peripheral Clock Divider Register 1 (PCDR1) contains the divider values for the peripheral clock dividers in the PLL Clock Controller. Peripherals in i.MX21 requires special clock frequency which is divided down from the MPLL and the SPLL clock output. Each of these peripheral modules receive their clock input from the respective clock divider. These modules will still have the clock gating scheme as with other modules for power saving advantages. Table 6-15 lists the clock sources associated with the i.MX21 peripherals given in the PCDR1.

PCDR1 — Peripheral Clock Divider Register 1 — Addr 0x1002701C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			PERDIV4								PERDIV3					
TYPE	r	r	rw	rw	rw	rw	rw	rw	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	1
	0x0307															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			PERDIV2								PERDIV1					
TYPE	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1
	0x070F															

Table 6-14. Peripheral Clock Divider Register 1 Description

Name	Description	Settings
Reserved Bits 31–30	These are reserved bits and should read 0.	
PERDIV4 Bits 29-24	**Peripheral Clock Divider 4**—Contains the 6-bit integer divider that produces the PERCLK4 clock signal for the CSI MCLK Clock.	000000 = Divide by 1 000001 = Divide by 2 … 111111 = Divide by 64
Reserved Bits 23–22	These are reserved bits and should read 0.	
PERDIV3 Bits 21–16	**Peripheral Clock Divider 3**—Contains the 6-bit integer divider that produces the PERCLK3 clock signal for the LCDC Pixel Clock.	000000 = Divide by 1 000001 = Divide by 2 … 111111 = Divide by 64
Reserved Bits 15–14	These are reserved bits and should read 0.	
PERDIV2 Bits 13–8	**Peripheral Clock Divider 2**—Contains the 6-bit integer divider that produces the PERCLK2 clock signal for the peripheral 2set (CSPI and SDHC).	000000 = Divide by 1 000001 = Divide by 2 … 111111 = Divide by 64
Reserved Bits 7–6	These are reserved bits and should read 0.	
PERDIV1 Bits 5–0	**Peripheral Clock Divider 1**—Contains the 6-bit integer divider that produces the PERCLK1 clock signal for the peripheral 1 set (UART, GPT, PWM).	000000 = Divide by 1 000001 = Divide by 2 … 111111 = Divide by 64

Table 6-15. Clock Sources for i.MX21 Peripherals

Clock Source	Peripherals
PERCLK4	CSI
PERCLK3	LCDC
PERCLK2	SDHC, CSPI
PERCLK1	UART, GPT, PWM

6.3.9 Peripheral Clock Control Register 0 (PCCR0)

The Peripheral Clock Control Register 0 (PCCR0) provides additional power saving capabilities by controlling the clocks in the i.MX21 modules. It also controls the clock source for Bootstrap mode. The PCCR0 allows for gating of HCLK to modules or peripherals that access the AHB bus and perform AHB bus transfers and also allows for gating of the ipg clk (PERCLK) to specific peripherals.

PCCR0 Peripheral Clock Control Register 0 Addr 0x10027020

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	HCLK_CSI_EN	HCLK_DMA_EN		HCLK_BROM_EN	HCLK_EMMA_EN	HCLK_LCDC_EN	HCLK_SLCDC_EN	HCLK_USBOTG_EN	HCLK_BMI_EN	PERCLK4_EN	SLCDC_EN	FIRI_BUAD_EN	NFC_EN	PERCLK3_EN	SSI1_BAUD_EN	SSI2_BAUD_EN
TYPE	rw	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0
	0x3108															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMMA_EN	USBOTG_EN	DMA_EN	I2C_EN	GPIO_EN	SDHC2_EN	SDHC1_EN	FIRI_EN	SSI2_EN	SSI1_EN	CSPI2_EN	CSPI1_EN	UART4_EN	UART3_EN	UART2_EN	UART1_EN
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1
	0x4003															

Table 6-16. Peripheral Clock Control Register 0 Description

Name	Description	Settings
HCLK_CSI_EN Bit 31	**CMOS Sensor Interface Clock Enable**—Enables/Disables HCLK clock input to the CSI module.	0 = CSI HCLK clock input is disabled (default). 1 = CSI HCLK clock input is enabled.
HCLK_DMA_EN Bit 30	**DMA Clock Enable**—Enables/Disables HCLK clock input to the DMA module.	0 = DMA HCLK clock input is disabled. (default) 1 = DMA HCLK clock input is enabled.
Reserved Bit 29	Reserved	
HCLK_BROM_EN Bit 28	**BROM Clock Enable**—Enables/Disables HCLK clock input to the BROM module.	0 = BROM HCLK clock input is disabled. 1 = BROM HCLK clock input is enabled (default).
HCLK_EMMA_EN Bit 27	**EMMA Clock Enable**—Enables/Disables HCLK clock input to the EMMA module.	0 = EMMA HCLK clock input is disabled (default). 1 = EMMA HCLK clock input is enabled.
HCLK_LCDC_EN Bit 26	**LCDC Clock Enable**—Enables/Disables HCLK clock input to the LCDC module.	0 = LCDC HCLK clock input is disabled (default). 1 = LCDC HCLK clock input is enabled.

Table 6-16. Peripheral Clock Control Register 0 Description (continued)

Name	Description	Settings
HCLK_SLCDC_EN Bit 25	**SLCDC Clock Enable**—Enables/Disables HCLK clock input to the SLCDC module.	0 = SLCDC HCLK clock input is disabled (default). 1 = SLCDC HCLK clock input is enabled.
HCLK_USBOTG_EN Bit 24	**USB OTG Clock Enable**—Enables/Disables HCLK clock input to the USB OTG module.	0 = USB OTG HCLK clock input is disabled. 1 = USB OTG HCLK clock input is enabled (default).
HCLK_BMI_EN Bit 23	**BMI Clock Enable**—Enables/Disables HCLK clock input to the BMI module.	0 = BMI HCLK clock input is disabled (default). 1 = BMI HCLK clock input is enabled.
PERCLK4_EN Bit 22	**PERCLK4 Clock Enable**—Enables/Disables peripheral clock input to CSI module.	0 = CSI peripheral clock input is disabled (default). 1 = CSI peripheral clock input is enabled.
SLCDC_EN Bit 21	**SLCDC Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the SLCDC module.	0 = SLCDC ipg clock input is disabled (default). 1 = SLCDC ipg clock input is enabled.
FIRI_BAUD_EN Bit 20	**FIRI Baud Clock Enable**—Enables/Disables baud rate clock input to the FIRI module.	0 = FIRI baud clock input is disabled (default). 1 = FIRI baud clock input is enabled.
NFC_EN Bit 19	**Nand Flash Controller Clock Enable**—Enables/Disables clock input to the NFC module.	0 = NFC clock input is disabled. 1 = NFC clock input is enabled (default).
PERCLK3_EN Bit 18	**PERCLK3 Clock Enable**—Enables/Disables pixel clock input to the LCDC module.	0 = LCDC pixel clock input is disabled (default). 1 = LCDC pixel clock input is enabled.
SSI1_BAUD_EN Bit 17	**SSI 1 Baud Clock Enable**—Enables/Disables baud rate clock input to the SSI1 module.	0 = SSI1 Baud clock input is disabled (default). 1 = SSI1 Baud clock input is enabled.
SSI2_BAUD_EN Bit 16	**SSI 2 Baud Clock Enable**—Enables/Disables baud rate clock input to the SSI2 module.	0 = SSI2 Baud clock input is disabled (default). 1 = SSI2 Baud clock input is enabled.
EMMA_EN Bit 15	**EMMA Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the EMMA module.	0 = EMMA ipg clock input is disabled (default). 1 = EMMA ipg clock input is enabled.
USBOTG_EN Bit 14	**USB OTG Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the USB OTG module.	0 = USB OTG ipg clock input is disabled. 1 = USB OTG ipg clock input is enabled (default).
DMA_EN Bit 13	**DMA Clock Enable**—Enables/Disables ipg clock input to the DMA module.	0 = DMA IPG clock input is disabled. (default). 1 = DMA ipg clock input is enabled.
I2C_EN Bit 12	**I^2C Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the I^2C module.	0 = I^2C ipg clock input is disabled (default). 1 = I^2C ipg clock input is enabled.
GPIO_EN Bit 11	**GPIO Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the GPIO module.	0 = GPIO ipg clock input is disabled (default). 1 = GPIO ipg clock input is enabled.
SDHC2_EN Bit 10	**SDHC2 Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the SDHC2 module.	0 = SDHC2 ipg clock input is disabled (default). 1 = SDHC2 ipg clock input is enabled.
SDHC1_EN Bit 9	**SDHC1 Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the SDHC1 module.	0 = SDHC1 ipg clock input is disabled (default). 1 = SDHC1 ipg clock input is enabled.

Table 6-16. Peripheral Clock Control Register 0 Description (continued)

Name	Description	Settings
FIRI_EN Bit 8	**FIRI Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the FIRI module.	0 = FIRI ipg clock input is disabled (default). 1 = FIRI ipg clock input is enabled.
SSI2_EN Bit 7	**SSI2 Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the SSI2 module.	0 = SSI2 ipg clock input is disabled. (default). 1 = SSI2 ipg clock input is enabled
SSI1_EN Bit 6	**SSI1 Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the SSI1 module.	0 = SSI1 ipg clock input is disabled. (default) 1 = SSI1 ipg clock input is enabled.
CSPI2_EN Bit 5	**CSPI2 Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the CSPI2 module.	0 = CSPI2 ipg clock input is disabled (default). 1 = CSPI2 ipg clock input is enabled.
CSPI1_EN Bit 4	**CSPI1Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the CSPI1 module.	0 = CSPI1 ipg clock input is disabled (default). 1 = CSPI1 ipg clock input is enabled.
UART4_EN Bit 3	**UART4 Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the UART4 module.	0 = UART4 ipg clock input is disabled (default). 1 = UART4 ipg clock input is enabled.
UART3_EN Bit 2	**UART3 Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the UART3 module.	0 = UART3 ipg clock input is disabled (default). 1 = UART3 ipg clock input is enabled.
UART2_EN Bit 1	**UART2 Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the UART2 module.	0 = UART2 ipg clock input is disabled. 1 = UART2 ipg clock input is enabled (default).
UART1_EN Bit 0	**UART1 Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the UART1 module.	0 = UART1 ipg clock input is disabled. 1 = UART1 ipg clock input is enabled (default).

NOTE

PERCLK1 and PERCLK2 are gated only when all of the corresponding modules driven from these clocks are disabled at the module itself and in the PCCR0 and PCCR1 registers.

6.3.10 Peripheral Clock Control Register 1

The Peripheral Clock Control Register 1 (PCCR1) provides additional power saving capabilities by controlling the clocks in the i.MX21 modules. It also controls the clock source for Bootstrap mode. The PCCR1 allows for gating of the ipg clk (PERCLK) to specific peripherals.

PCCR1 — Peripheral Clock Control Register 1 — Addr 0x10027024

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OWIRE_EN	KPP_EN	RTC_EN	PWM_EN	GPT3_EN	GPT2_EN	GPT1_EN	WDT_EN	CSPI3_EN							
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 6-17. Peripheral Clock Control Register 1 Description

Name	Description	Settings
OWIRE_EN Bit 31	**OWIRE Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the 1-Wire module.	0 = 1-Wire ipg clock input is disabled (default). 1 = 1-Wire ipg clock input is enabled.
KPP_EN Bit 30	**KPP Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the KPP module.	0 = KPP ipg clock input is disabled (default). 1 = KPP ipg clock input is enabled.
RTC_EN Bit 29	**RTC Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the RTC module.	0 = RTC ipg clock input is disabled (default). 1 = RTC ipg clock input is enabled.
PWM_EN Bit 28	**PWM Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the PWM module.	0 = PWM ipg clock input is disabled (default). 1 = PWM ipg clock input is enabled.
GPT3_EN Bit 27	**GPT3 Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the GPT3 module.	0 = GPT3 ipg clock input is disabled (default). 1 = GPT3 ipg clock input is enabled.
GPT2_EN Bit 26	**GPT2 Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the GPT2 module.	0 = GPT2 ipg clock input is disabled (default). 1 = GPT2 ipg clock input is enabled.
GPT1_EN Bit 25	**GPT1 Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the GPT1 module.	0 = GPT1 ipg clock input is disabled (default). 1 = GPT1 ipg clock input is enabled.
WDT_EN Bit 24	**WDT Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the WDT module.	0 = WDT ipg clock input is disabled (default). 1 = WDT ipg clock input is enabled.

Table 6-17. Peripheral Clock Control Register 1 Description (continued)

Name	Description	Settings
CSPI3_EN Bit 23	**CSPI3 Clock Enable**—Enables/Disables peripheral clock ipg clock (PERCLK) input to the CSPI3 module.	0 = CSPI3 ipg clock input is disabled (default). 1 = CSPI3 ipg clock input is enabled.
Reserved Bit 22–0	Reserved—These bits are reserved and should read 0.	

6.3.11 Clock Control Status Register

The Clock Control Status Register (CCSR) provides information on the configuration of the Analog and Digital block. The clocks within the chip can also be monitored by the CLKO_SEL programming.

CCSR — Clock Control Status Register — Addr 0x10027028

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	32K_SR											CLKO_SEL				
TYPE	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
	0x0300															

Table 6-18. Clock Control Status Register Description

Name	Description	Settings
Reserved Bit 31–16	Reserved—These bit are reserved and should read 0.	
32K_SR Bit 15	**32K Status Register**—The 32K_SR contains the status information of the 32KHz clock. The bit value is cleared to zero during the assertion of the $\overline{\text{HARD\_ASYNC\_RESET}}$ signal. The sampled 32KHz clock phase is continuously registered into the bit upon the deassertion of the $\overline{\text{HARD\_ASYNC\_RESET}}$ signal.	0 = CLK32 in low phase 1 = CLK32 in high phase
Reserved Bit 14–5	Reserved—These bits are reserved and should read 0.	

Table 6-18. Clock Control Status Register Description (continued)

Name	Description	Settings
CLKO_SEL Bits 4–0	**CLKO Select**—Selects the clock signal source that is output on the CLKO pin.	00000 = CLK32 00001 = PREMCLK 00010 = CLK26M 00011 = MPLL Reference CLK 00100 = SPLL Reference CLK 00101 = MPLL CLK 00110 = SPLL CLK 00111 = FCLK 01000 = HCLK 01001 = PERCLK (IPG_CLK) 01010 = PERCLK1 01011 = PERCLK2 01100 = PERCLK3 01101 = PERCLK4 01110 = SSI 1 Baud 01111 = SSI 2 Baud 10000 = NFC Baud 10001 = FIRI Baud 10010 = CLK48M Always 10011 = CLK32K Always 10100 = CLK48M 10101 = CLK48DIV_CLKO

6.3.12 Wakeup Guard Mode Control Register

The Wakeup Guard Mode Control Register (WKGDCTL) provides the configuration of the wakeup guard mode. This is a write once only bit in order to be compatible with the watchdog behavior. After enable/disable, it will not be modifiable. When enabled, the battery detector external to the chip provides a glitch free signal through the TIN pin. Battery must be intact for the chip to wakeup from sleep.

WKGDCTL — Wakeup Guard Mode Control Register — Addr 0x10027034

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																WKGD_EN
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 6-19. Wakeup Guard Mode Control Register

Name	Description	Settings
Reserved Bit 31–1	Reserved—These bit are reserved and should read 0.	
WKDG_EN Bits 0	**Wakeup Guard Mode Enable**— Enables /disables the wakeup guard logic. Write- once-only bit and can only be cleared through system reset. Once enabled, the battery indicator through TIN will be used to qualify the wakeup process. When battery is intact—that is, TIN=1, wakeup from sleep proceed as per normal. When WKGD_EN=1 and battery is removed, the 32 kHz to the watchdog module is gated off. Clock resumed when battery is back in place.	0 = Wakeup guard mode is disabled. 1 = Wakeup guard mode is enabled.

6.4 Functional Description of the Reset Module

The reset module controls or distributes all of the system reset signals used by the i.MX21. A simplified block diagram of the reset module is shown in Figure 6-2. The reset module generates two distinct events—a global reset and an ARM9 platform reset.

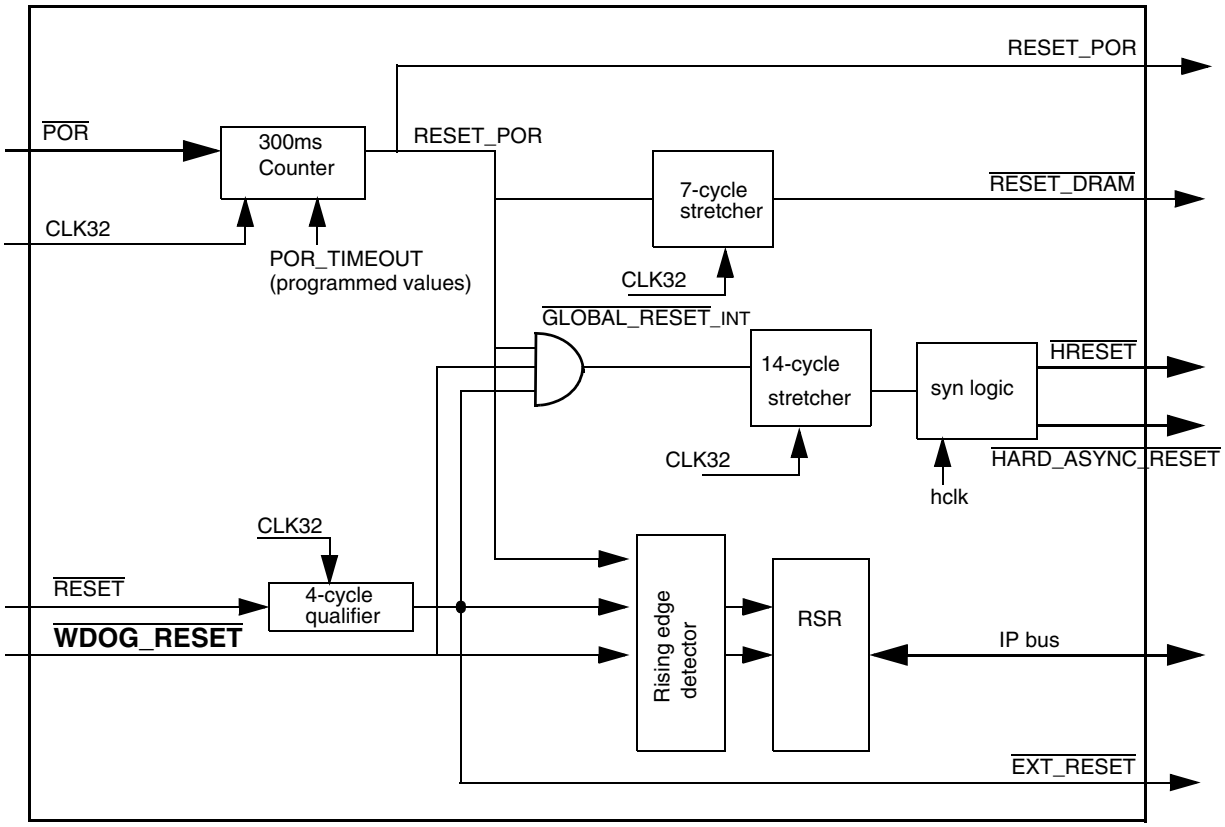


Figure 6-2. Reset Module Clock Diagram

6.4.1 Global Reset

A global reset is produced by the simultaneous assertion of the following resets:

- RESET_DRAM
- HRESET
- HARD_ASYNC_RESET
- RESET_POR

There is one source capable of generating a global reset: A low condition on the POR pin when the 32 kHz crystal oscillator is running.

The HRESET and HARD_ASYNC_RESET are armed simultaneously; they remain in that state for 14 CLK32 cycles.

RESET_DRAM is deasserted seven CLK32 cycles before HRESET and HARD_ASYNC_RESET are deasserted. The SDRAM executes the necessary self refresh operations during this time.

The timing diagram in Figure 6-2 shows the relationship of the reset signal timings. See Table 6-20 for reset module signal and pin definitions.

The following signal conditions are not capable of generating a global reset, however their assertion will reset the ARM9 platform:

- An external qualified low condition on the $\overline{\text{RESET\_IN}}$ pin
- A low condition on $\overline{\text{WDOG\_RESET}}$

Furthermore, these reset conditions will not reset the SDRAMC, Real Time Clock, WatchDog module, or allow a change in boot mode—that is, changes made to BOOT[3:0] during these resets conditions will not take affect. Only the global reset is capable of this.

The source of the last hardware reset can be determined in the watchdog status register.

NOTE

Due to the asynchronous nature of the $\overline{\text{RESET\_IN}}$ signal, the time period required to qualify the signal may vary, and the $\overline{\text{HRESET}}$ timing relative to the rising edge of the $\overline{\text{RESET\_IN}}$ is also affected. A $\overline{\text{RESET\_IN}}$ signal shorter than three CLK32 cycles will not be qualified, a $\overline{\text{RESET\_IN}}$ signal equal to or longer than four CLK32 cycles will always be qualified, and any period length that is more than three and less than four CLK32 cycles is undefined.

$\overline{\text{POR}}$ is the reset signal for all the reset module flip-flops. For this reason, an external reset signal is qualified if it lasts more than four CLK32 cycles when $\overline{\text{POR}}$ is deasserted.

During power on the user must ensure that $\overline{\text{POR}}$ stay asserted (low) long enough for the 32kHz crystal to stabilize.

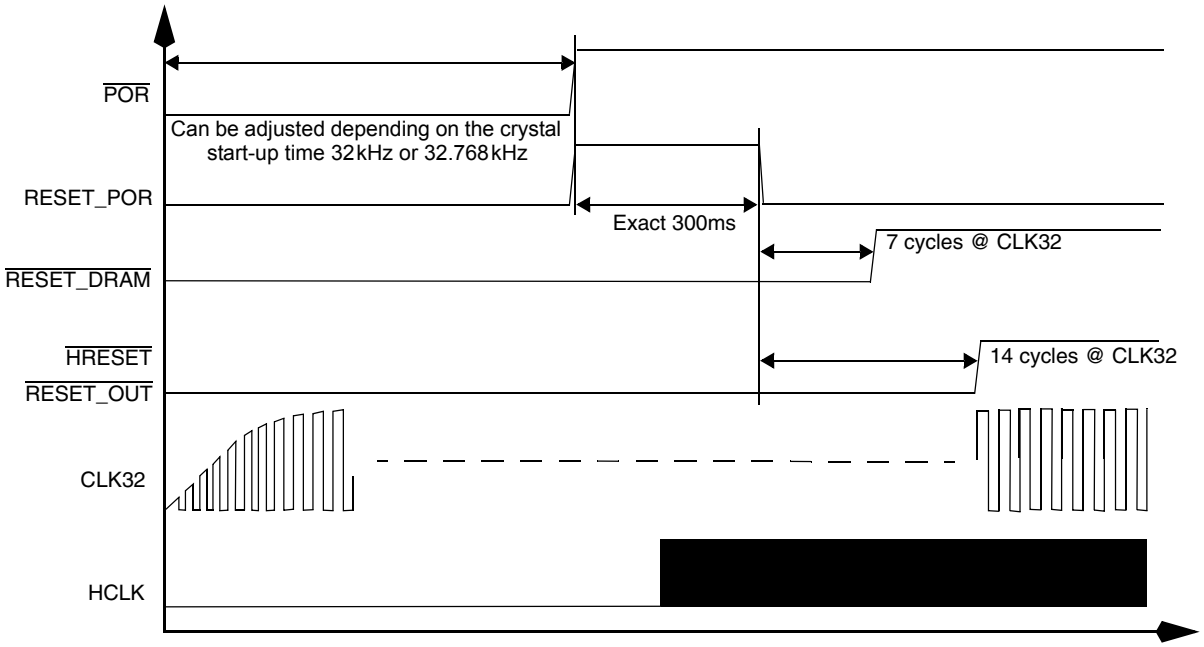


Figure 6-3. DRAM and Internal Reset Timing Diagram

6.4.2 ARM9 Platform Reset

Any qualified global reset signal resets the ARM9 platform and all related peripherals to their default state. After the internal reset is deasserted, the ARM9 processor begins fetching code from the internal bootstrap ROM or CS0 space. The memory location of the fetch depends on the configuration of the BOOT pins and the value of the TEST pin on the rising edge of the HRESET.

Table 6-20. Reset Module Pin and Signal Descriptions

Signal Name	Direction	Signal Description
CLK32	IN	**32 kHz Clock**—A 32 kHz clock signal derived from the 32K crystal oscillator circuit in the PLL Clock Controller.
POR	IN	**Power-On Reset**—An internal active Schmitt trigger signal from the POR pin. The POR signal is normally generated by an external RC circuit designed to detect a power-up event.
RESET_IN	IN	**Reset**—An external active low Schmitt trigger signal from the RESET_IN pin. When this signal goes active, all modules (except the SDRAMC, Real Time Clock, WatchDog, and the BOOT[3:0] signals) are reset.
WDOG_RESET	IN	**Watchdog Timer Reset**—An active low signal generated by the watchdog timer when a time-out period has expired. Resets the same modules as RESET_IN.
HARD_ASYN_RESET	OUT	**Hard Asynchronous Reset**—An active low signal that resets all peripheral modules except the watchdog module's status register. The rising edge of this signal is synchronous with IPG_CLK.
HRESET	OUT	**Hard Reset**—An active low signal that resets the ARM9 platform. This signal is deasserted during the low phase of HCLK. This signal also appears on the RESET_OUT pin of the i.MX21.
RESET_DRAM	OUT	**DRAM Reset**—An active low signal that resets the SDRAM controller.

Chapter 7
AHB-Lite IP Interface (AIPI) Module

This chapter provides an overview of the AHB-Lite to IP bus interface (AIPI) module. The AIPI acts as an interface between the advanced ARM High-performance Bus "Lite" (AHB-Lite) and lower bandwidth peripherals conforming to the *Freescale IP Bus Specification*. There are two AIPI modules in i.MX21.

The following list summarizes the key features of the AIPI:

- All peripheral read transactions require a minimum of 2 system clocks (R-AHB side) and all write transactions require a minimum of 3 system clocks (R-AHB side).
- The AIPI supports 8-bit, 16-bit and 32-bit IP bus peripherals. (Byte, half word and word reads and write are supported to each.)
- The AIPI supports multi-cycle accesses (16-bit operations to 8-bit peripherals and 32-bit operations to 16-bit and 8-bit peripherals).
- The AIPI supports 31 external IP bus peripherals each with a 4 Kbyte memory map (a slot).

Table 7-1. AHB-Lite To IP Bus V2.0 Interface Operation (Little Endian)

Transfer Size	haddr		IP Bus Size	ips_addr		Active Bus Section (R-AHB to IP Bus)			
	[1]	[0]		[1]	[0]	R-AHB[31:24]	R-AHB[23:16]	R-AHB[15:8]	R-AHB[7:0]
Byte	0	0	8 bit	0	0	–	–	–	ips_data[7:0]
	0	1		0	1	–	–	ips_data[7:0]	–
	1	0		1	0	–	ips_data[7:0]	–	–
	1	1		1	1	ips_data[7:0]	–	–	–
	0	0	16 bit	0	X	–	–	–	ips_data[7:0]
	0	1				–	–	ips_data[15:8]	–
	1	0		1	X	-	ips_data[7:0]	–	–
	1	1				ips_data[15:8]	–	–	–
	0	0	32 bit	X	X	–	–	–	ips_data[7:0]
	0	1				–	–	ips_data[15:8]	–
	1	0		X	X	–	ips_data[23:16]	–	–
	1	1				ips_data[31:24]	–	–	–

Table 7-1. AHB-Lite To IP Bus V2.0 Interface Operation (Little Endian) (continued)

Transfer Size	haddr		IP Bus Size	ips_addr		Active Bus Section (R-AHB to IP Bus)			
	[1]	[0]		[1]	[0]	R-AHB[31:24]	R-AHB[23:16]	R-AHB[15:8]	R-AHB[7:0]
Half Word	0	NA	8 bit	0	0	–	–	–	ips_data[7:0]
					1	–	–	ips_data[7:0]	–
	1			1	0	–	ips_data[7:0]	–	–
					1	ips_data[7:0]	–	–	–
	0		16 bit	0	X	–	–	ips_data[15:8]	ips_data[7:0]
	1			1	X	ips_data[15:8]	ips_data[7:0]	–	–
	0		32 bit	X	X	–	–	ips_data[15:8]	ips_data[7:0]
	1			X	X	ips_data[31:24]	ips_data[23:16]	–	–
Word	NA	NA	8 bit	0	0	–	–	–	ips_data[7:0]
					1	–	–	ips_data[7:0]	–
				1	0	–	ips_data[7:0]	–	–
					1	ips_data[7:0]	–	–	–
			16 bit	0	X	–	–	ips_data[15:8]	ips_data[7:0]
				1	X	ips_data[15:8]	ips_data[7:0]	–	–
			32 bit	X	X	ips_data[31:24]	ips_data[23:16]	ips_data[15:8]	ips_data[7:0]

7.1 Programming Model

There are three registers that reside inside the AIPI. These registers correspond to the first slot (4 Kbyte memory region) of each of the 2 AIPIs in i.MX21 at 0x1000_0000 and 0x1002_0000, respectively. All three registers are 32-bit registers and can only be accessed in supervisor mode. Additionally, these registers can only be read from or written to by a 32-bit access.

Two system clocks are required for read accesses and three system clocks are required for write accesses to the AIPI registers.

CAUTION

Writing to reserved register locations within the 4 Kbyte memory map of the AIPI register space (other than the three AIPI registers) will result in unknown behavior and an abort exception.

Access to Reserved or Unoccupied locations in the AIPI space will result in an abort exception.

7.1.1 Peripheral Size Registers[1:0]

These registers are used to tell the AIPI what size of IP bus peripheral is in each IP bus peripheral location. Peripheral locations that are not occupied should have their corresponding bits in the peripheral size registers (PSRs) programmed to 1 in each register.

The least significant bit in the PSRs is a read-only bit as it governs the AIPI registers themselves. They are set and cleared appropriately to indicate the registers are 32-bit.

PSR0 — Peripheral Size Register 0 — Addr 0x1000_0000 0x1002_0000

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

PSR1 — Peripheral Size Register 1 — Addr 0x1000_0004 0x1002_0004

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The PSRs work together to indicate the size of the IP bus peripheral occupying the corresponding location, or to indicate there is no IP bus peripheral occupying the corresponding location. Table 7-2 shows how to program the PSR registers based on the size or availability of an IP bus peripheral.

Table 7-2. PSR 1–0 Data Bus Size Encoding

PSR 1–0	IP Bus Peripheral SIZE [x]
00	8-bit
01	16-bit
10	32-bit
11	Unoccupied

7.1.2 Peripheral Access Register

The peripheral access register (PAR) tells the AIPI whether the IP bus peripheral corresponding to the bit location in this register may be accessed in user mode. If the peripheral may be accessed in supervisor mode only and a user mode access is attempted, an abort is generated and no IP bus activity occurs.

The least significant bit in the PAR is a read-only bit as it governs the AIPI registers themselves. It is set to indicate supervisor access only.

PAR — Peripheral Access Register[1] — Addr 0x1000_0008, 0x1002_0008

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

[1] A “1” indicates the corresponding peripheral is a supervisor access only peripheral. A “0” indicates the decision is left up to the peripheral (the AIPI allows user accesses).

7.2 AIPI1 and AIPI2 Peripheral Widths and PSR Setting

Table 7-5 shows the data bus widths of the peripherals on the AIPI1 and the AIPI2 interfaces. System software should make use of information in the column, “Data Bus Width” to configure the PSR registers, accordingly.

Table 7-3 and Table 7-4 show PSR settings for AIPI1 and AIPI2.

Table 7-5 shows the peripheral sizes for the occupied, reserved and unoccupied locations. All reserved locations must be programmed as 32-bit locations.

Table 7-3. AIPI1 PSR Setting

PSR	Setting
PSR[1]	0xFFFB_FCFB
PSR[0]	0x0004_0304

Table 7-4. AIPI2 PSR Setting

PSR	Setting
PSR[1]	0xFFFF_FFFF
PSR[0]	0x3FFC_0000

Table 7-5. i.MX21 AIPI Peripheral Access Sizes and IP Access Types

Location	Peripheral	PSR[1]	PSR[0]	Data Bus Width	16-bit		8-bit	
					Read	Write	Read	Write
AIPI1								
0	AIPI1 Control	1	0	32-bit	–	–	–	–
1	DMA	1	0	32-bit	Y	Y	Y	Y
2	WDOG	0	1	16-bit	–		Y	Y
3	GPT1	1	0	32-bit	N	N	N	N
4	GPT2	1	0	32-bit	N	N	N	N
5	GPT3	1	0	32-bit	N	N	N	N
6	PWM	1	0	32-bit	N	N	N	N
7	RTC	1	0	32-bit	Y	Y	Y	Y
8	KPP	0	1	16-bit	–		N	N
9	1-Wire	0	1	16-bit	–		N	N
10	UART1	1	0	32-bit	N	Y	N	Y
11	UART2	1	0	32-bit	N	Y	N	Y
12	UART3	1	0	32-bit	N	Y	N	Y
13	UART4	1	0	32-bit	N	Y	N	Y
14	CSPI1	1	0	32-bit	N	N	N	N
15	CSPI2	1	0	32-bit	N	N	N	N
16	SSI1	1	0	32-bit	N	N	N	N
17	SSI2	1	0	32-bit	N	N	N	N
18	I2C	0	1	16-bit	–		Y	Y
19	SDHC1	1	0	32-bit	N	N	N	N
20	SDHC2	1	0	32-bit	N	N	N	N
21	GPIO	1	0	32-bit	N	N	N	N
22	AUDMUX	1	0	32-bit	N	N	N	N
23	CSPI3	1	0	32-bit	N	N	N	N
23-31	Reserved	1	0	32-bit	N	N	N	N
AIPI2								
0	AIPI2	1	0	32-bit	–			
1	LCDC	1	0	32-bit	N	N	N	N
2	SLCDC	1	0	32-bit	N	N	N	N
3	Reserved	1	0	32-bit	N	N	N	N

Table 7-5. i.MX21 AIPI Peripheral Access Sizes and IP Access Types (continued)

Location	Peripheral	PSR[1]	PSR[0]	Data Bus Width	16-bit		8-bit	
					Read	Write	Read	Write
4	USB OTG	1	0	32-bit	N	N	Y	Y
5	USB OTG	1	0	32-bit	N	N	N	N
6	EMMA	1	0	32-bit	N	N	N	N
7	CRM	1	0	32-bit	Y	Y	Y	Y
8	FIRI	1	0	32-bit	Y	Y	Y	Y
9	Reserved	–	–	–	–	–	–	–
10-17	Reserved	1	0	32-bit	N	N	N	N
18-29	Unoccupied	1	1	–	N	N	N	N

7.3 Interface Timing

This section describes AIPI interface timing characteristics.

7.3.1 Read Cycles

Two clock read accesses are possible with the AIPI when the requested access size is equal to or smaller than the size of the targeted IP bus peripheral. If the requested access size is larger than that of the targeted IP bus peripheral (for example, a 32-bit access to a 16 bit peripheral) then a minimum of three clocks are required to complete the access.

7.3.2 Write Cycles

Three clock write accesses are possible with the AIPI when the requested access size is equal to or smaller than the size of the targeted IP bus peripheral. If the requested access size is larger than that of the targeted IP bus peripheral (for example, a 32-bit access to a 16 bit peripheral) then a minimum of four clocks are required to complete the access.

7.3.3 Aborted Cycles

The AIPI follows a standard procedure when a cycle is aborted and the abort is initiated by the AIPI itself or the targeted IP bus peripheral. The AIPI either fails to initiate or immediately terminates any IP bus activity that is ongoing.

There are several conditions that can cause the AIPI to abort the current operation and report an error. The first is the case in which the targeted IP bus peripheral asserts its internal error signal. In this case the AIPI immediately terminates access to the targeted IP bus peripheral. Whether the current IP bus access is a multi-cycle access or a single cycle access has no bearing on the behavior of the AIPI. The AIPI responds identically in both cases.

The second case that can cause an error response to the AHB-Lite is when a user-mode access is attempted to an IP bus peripheral whose corresponding PAR bit indicates it is a supervisor-only peripheral. In this case the AIPI does not initiate any IP bus activity but instead responds immediately by following the abort procedure described above.

The third case that can cause an error response to the AHB-Lite is when an access is attempted to a location at which the PSRs indicate there is no IP bus peripheral. In this case the AIPI does not initiate any IP bus activity but instead responds immediately by following the abort procedure described above.

Part 3
System Control

Chapter 8
System Control

This chapter describes the system control module of the i.MX21 microprocessor. The system control module enables system software to control, customize, or read the status of the following functions:

- Chip ID
- Multiplexing of I/O signals
- I/O Driving Strength
- Well Bias Control
- System boot mode selection

8.1 Programming Model

The system control module includes one 128-bit Silicon ID and fifteen user-accessible 32-bit registers. Table 8-1 summarizes these registers and their addresses.

Table 8-1. System Control Module Register Summary

Description	Name	Address
Silicon ID Register	SIDR	0x10027804
Function Multiplexing Control Register	FMCR	0x10027814
Global Peripheral Control Register	GPCR	0x10027818
Well Bias Control Register	WBCR	0x1002781C
Driving Strength Control Register 1	DSCR1	0x10027820
Driving Strength Control Register 2	DSCR2	0x10027824
Driving Strength Control Register 3	DSCR3	0x10027828
Driving Strength Control Register 4	DSCR4	0x1002782C
Driving Strength Control Register 5	DSCR5	0x10027830
Driving Strength Control Register 6	DSCR6	0x10027834
Driving Strength Control Register 7	DSCR7	0x10027838
Driving Strength Control Register 8	DSCR8	0x1002783C
Driving Strength Control Register 9	DSCR9	0x10027840
Driving Strength Control Register 10	DSCR10	0x10027844
Driving Strength Control Register 11	DSCR11	0x10027848
Driving Strength Control Register 12	DSCR12	0x1002784C
Priority Control and Select Register	PCSR	0x10027850

8.1.1 Silicon ID Register

This register consists of laser programmable Silicon ID and Unique Identifier. The Unique Identifier is 48 laser programmable bits where the Silicon ID and Unique Identifier are accessible by the IP bus interface to the MCU. The 96-bit register can only be accessed in 32-bit access mode. The register bit[31:0] is located on 0x10027804. The settings for the bits in the register are listed in Table 8-2 (As laser fuse: Fuse-burn means logic 0, Fuse-non-burn means logic 1).

SIDR — Silicon ID Register — 0x10027804

BIT	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
	CID															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	1
	0x101D															

BIT	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
	CID															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	1
	0x101D															

BIT	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000_0000															

BIT	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	SUID															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000_0000															

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SUID															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SUID															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 8-2. Silicon ID Register Description

Name	Description	Settings
CID Bits 95–64	**Chip ID**—Contains the chip identification number of the i.MX21 (mask set number identification). **Note:** Refer to the MC9328MX21RM Addendum if your silicon ID is the older, *n*L45X silicon version, for specific PCMCIA register and signal differences.	0M55B = 0x101D101D 1M55B = 0x201D101D M55B = 0x201D101D
Reserved Bits 63–48	Reserved—These bits are reserved and should read 0.	
SUID Bits 47–0	**Silicon Unique ID**—Contains the chip unique identification number of the i.MX21, the ID number is laser programmed during package.	Blown[x]=1 Intact[x]= 0

8.1.2 Function Multiplexing Control Register

The Function Multiplexing Control Register (FMCR) controls the multiplexing of the signal lines shared by the SLCDC module, UART module, and Keypad module as well as the SDRAM chip select lines. The FMCR also allows control or indicates the boot status of the NAND Flash page size and data port size. Finally, the FMCR contains a bit setting for the Well Bias control to enable optimal power savings during STOP mode when Well Bias Mode is enabled (refer to Section 8.1.4 on page -7 for more information on the Well Bias feature). See Table 8-3 for detailed description of bit settings.

FMCR — Function Multiplexing Control Register — Addr 0x10027814

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							UART4_RXD_CTL	UART4_RTS_CTL						KP_COL6_CTL	KP_ROW7_CTL	KP_ROW6_CTL
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0xFFFF															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				CRM_SPA_SEL	Reserved						NF_FMS	NF_16BIT_SEL		SLCDC_SEL	SDCS1_SEL	SDCS0_SEL
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1
	0xFFCB															

Table 8-3. Function Multiplexing Control Register Description

Name	Description	Settings
Reserved Bits 31–26	Reserved—These bits are reserved and should read 1.	
UART4_RXD_CTL Bit 25	**UART4 RXD Control**—When set, the alternate signal of USBH1_RXDP (PB31) is input to RXD of UART4. When 0, the USBH1_TXDP (PB29) GPIO's AOUT is input to RXD of UART4. With either setting, the user must also ensure that the proper GPIO registers have been programmed to select the desired multiplexing.	0 = The USBH1_TXDP (PB29) GPIO's AOUT is input to RXD of UART4. 1 = The alternate signal of USBH1_RXDP (PB31) is input to RXD of UART4.
UART4_RTS_CTL Bit 24	**UART4 RTS Control**—When set, the alternate signal of USBH1_FS (PB26) is input to RTS of UART4. When 0, the USBH1_RXDP (PB31) GPIO's AOUT is input to RTS of UART4. With either setting, the user must also ensure that the proper GPIO registers have been programmed to select the desired multiplexing.	0 = The USBH1_RXDP (PB31) GPIO's AOUT is input to RTS of UART4. 1 = The alternate signal of USBH1_FS (PB26) is input to RTS of UART4.
Reserved Bits 23–19	Reserved—These bits are reserved and should read 1.	
KP_COL6_CTL Bit 18	**Keypad Column 6 Control**—When set, the alternate signal of UART2_TXD (PE6) is input to column 6 of keypad. When 0, the alternate signal of TEST_WB2 (PE0) is input to column 6 of keypad. With either setting, the user must also ensure that the proper GPIO registers have been programmed to select the desired multiplexing.	0 = The alternate signal of TEST_WB2 (PE0) is input to column 6 of keypad. 1 = The alternate signal of UART2_TXD (PE6) is input to column 6 of keypad.
KP_ROW7_CTL Bit 17	**Keypad Row 7 Control**—When set, the alternate signal of UART2_RTS (PE4) is input to row 7of keypad. When 0, the alternate signal of TEST_WB0 (PE2) is input to row 7 of keypad. With either setting, the user must also ensure that the proper GPIO registers have been programmed to select the desired multiplexing.	0 = The alternate signal of TEST_WB0 (PE2) is input to row 7 of keypad. 1 =The alternate signal of UART2_RTS (PE4) is input to row 7of keypad.
KP_ROW6_CTL Bit 16	**Keypad Row 6 Control**—When set, the alternate signal of UART2_RXD (PE7) is input to row 6 of keypad. When 0, the alternate signal of TEST_WB1 (PE1) is input to row 6 of keypad. With either setting, the user must also ensure that the proper GPIO registers have been programmed to select the desired multiplexing.	0 = The alternate signal of TEST_WB1 (PE1) is input to row 6 of keypad. 1 = The alternate signal of UART2_RXD (PE7) is input to row 6 of keypad.
Reserved Bits 15–13	Reserved—These bits are reserved and should read 1.	
CRM_SPA_SEL Bit 12	**CRM Set Point Adjust Select**—When enabling Well Bias, the user should set this bit to 1 for optimal power savings during STOP mode. A setting of 0 is reserved when Well Bias is enabled.	0 = Reserved when Well Bias is enabled. 1 = Selects Well Bias optimal power savings when Well Bias is enabled. This bit has no effect when Well Bias is disabled.
Reserved Bit 11	Reserved—This bit is reserved and should read 1.	
Reserved Bits 10–6	Reserved—These bits are reserved and should read 1.	

Table 8-3. Function Multiplexing Control Register Description (continued)

Name	Description	Settings
NF_FMS Bit 5	**Flash Memory Select**—When Boot[3:0] = 0010 or 0011, the NF_FMS will be set, otherwise it will be 0. After Bootup, this bit is user programmable.	0 = NAND Flash with 512B page size (64Mb/128Mb/256Mb/512Mbyte/ 1Gbyte DDP) 1 = NAND Flash with 2 Kbyte page size (1Gbyte/2Gbyte DDP/2Gbyte) **Note:** DDP means Double Density Package.
NF_16BIT_SEL Bit 4	**Nand Flash 16-bit Select**—Selects 16-bit NF. Setting this bit by SW will force to select NAND Flash 16-bit mode and the NAND Flash upper data presents to the pins. Clearing this bit by SW will force the NF to 8-bit mode and the A[25:21]signals become the function pins. The muxing is done in the EMI module, not I/O mux module. During system boot up, if the BOOT[3:0] input pins are configured to select 16-bit mode, this NF_16BIT_SEL bit is set.	0 = NAND Flash 8-bit 1 = NAND Flash 16-bit
Reserved Bit 3	Reserved—This bit is reserved and should read 0.	
SLCDC_SEL Bit 2	**SLCDC Select**—Select whether a BaseBand chip (BB) or the i.MX21 Application Processor drives the SLCDC display port in serial mode.	0 = On Chip SLCDC drives the SLCDC port. 1 = BB can write directly to the SLCDC port.
SDCS1_SEL Bit 1	**SDRAM Chip Select**—Selects the function of the $\overline{\text{CS3/CSD1}}$ pin.	0 = $\overline{\text{CS3}}$ selected 1 = $\overline{\text{CSD1}}$ selected
SDCS0_SEL Bit 0	**SDRAM Chip Select**—Selects the function of the $\overline{\text{CS2/CSD0}}$ pin.	0 = $\overline{\text{CS2}}$ selected 1 = $\overline{\text{CSD0}}$ selected

8.1.3 Global Peripheral Control Register (GPCR)

The Global Peripheral Control Register (GPCR) displays the current boot mode of the i.MX21. The clock gating to the IP modules can be controlled by this register. Descriptions of the register settings appear in Table 8-4.

GPCR — Global Peripheral Control Register — Addr 0x10027818

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													BOOT			
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													CLOCK_GATING_EN			
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	rw	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
	0x000C															

Table 8-4. Global Peripheral Control Register Description

Name	Description	Settings
Reserved Bits 31–20	Reserved—These bits are reserved and should read 0.	
BOOT Bits 19–16	**Boot Mode**—These are 4-bit system boot mode for the i.MX21.	0000 = Bootstrap from UART/USB 0001 = Bootstrap from UART/USB 0010 = 8-bit NAND Flash (2 Kbyte per page) 0011 = 16-bit Nand Flash (2 Kbyte per page) 0100 = 16-bit Nand Flash (512Bytes per page) 0101 = 16-bit CS0 0110 = 32-bit CS0 0111 = 8 bit Nand Flash (512Bytes per page) 1xxx = Reserved for Test Modes
Reserved Bit 15–4	Reserved—These bits are reserved and should read 0.	
CLOCK_GATING_EN Bit 3	**Clock Gating Enable**—When set to 1, the peripheral register access clocks ipg_clk_s1 and ipg_clk_s2 will be gated by the aipi1_ips_module_en and aipi2_ips_module_en. For example, when there is a register read or write access to the peripherals of AIPI1, the ipg_clk_s1 clock will be running, otherwise if no access is taking place the clock will shut off and when there is a register read or write access to the peripherals of AIPI2, the ipg_clk_s2 clock will be running, otherwise if no access is taking place the clock shuts off. When this bit is cleared to 0 then ipg_clk_s1 and ipg_clk_s2 will become a continuous clock, regardless of peripheral accesses. It is recommended for maximum power savings to ensure this bit is set to 1.	
Reserved Bit 2–0	Reserved—These bit are reserved and should read 0.	

8.1.4 Well Bias System

The i.MX21 processor employs an innovative system feature to help reduce leakage current of the ARM926 core logic during STOP mode called Well Biasing. The Well Bias System reduces the leakage current of the QVDDx sub-system (mainly the ARM core logic) during STOP (low-power) mode by increasing the threshold voltage of the QVDDx sub-system transistors. The following section describes how to enable and take advantage of this power saving feature.

8.1.5 Well Bias Control Register (WBCR)

The Well Bias Control Register (WBCR) allows the user to enable the Well Biasing System. At default the Well Biasing System is disabled. To enable the Well Biasing System and take advantage of this power saving feature, the following bit settings must occur in the WBCR: the CRM_WBS bits must set to 01, the CRM_WBFA bit must be set to 1, and the CRM_WBM bits must be set to 001. In addition, the CRM_SPA_SEL bit in the FMCR must also be set to 1.

WBCR — Well Bias Control Register — Addr 0x1002781C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved			Reserved					CRM_WBS		Reserved		CRM_WBFA	CRM_WBM		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 8-5. Well Bias Control Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
Reserved Bits 15–13	Reserved—These bits are reserved and should read 0.	
Reserved Bits 12–8	Reserved—These bits are reserved and should read 0.	

Table 8-5. Well Bias Control Register Description (continued)

Name	Description	Settings
CRM_WBS Bits 7–6	**Well Bias Switching Mode**—For proper operation in Well Bias mode, these bits should be set to 01.	00 = Reserved 01 = Suggested mode when Well Bias enabled 10 = Reserved 11 = Reserved These bits have no effect when Well Bias is disabled.
Reserved Bits 5–4	Reserved—These bits are reserved and should read 0.	
CRM_WBFA Bit 3	**Well Bias Frequency Adjust—**For optimal power savings, the user should set this bit to 1 when Well Bias is enabled.	0 = Reserved 1 = Suggested setting for optimal power savings when Well Bias is enabled This bit has no effect when Well Bias is disabled.
CRM_WBM Bits 2–0	**CRM_WBM**—Enables or disables Well Bias System during STOP mode. To enable Well Bias during STOP mode, these bits must be set to 001. To disable Well Bias, these bits must be set to 000. All other bit settings are reserved.	000 = Well Bias not applied 001 = Well Bias @ STOP 010–111 = Reserved

8.1.6 Driving Strength Control Register 1

The Driving Strength Control Register 1 (DSCR1) controls the driving force parameters of the all the slow I/O signals in the i.MX21. Descriptions of the register settings appear in Table 8-6.

DSCR1 — Driving Strength Control Register 1 — Addr 0x10027820

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								DS_SLOW8			DS_SLOW7			DS_SLOW6		
TYPE	r	r	r	r	r	r	r	rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	0x0040															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		DS_SLOW5			DS_SLOW4			DS_SLOW3			DS_SLOW2			DS_SLOW1		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 8-6. Driving Strength Control Register 1 Description

Name	Description	Settings
Reserved Bits 31–25	Reserved—These bits are reserved and should read 0.	
DS_SLOW8 Bits 24–22	**Driving Strength Slow I/O**—Controls the driving strength of the LCDC and BMI I/O signals.	000 = 2 mA 001 = 4 mA 011 = 8 mA 111 = 12 mA
DS_SLOW7 Bits 21–19	**Driving Strength Slow I/O**—Controls the driving strength of the SDI2 I/O signals.	000 = 2 mA 001 = 4 mA 011 = 8 mA 111 = 12 mA
DS_SLOW6 Bits 18–16	**Driving Strength Slow I/O**—Controls the driving strength of the CSI I/O signals.	000 = 2 mA 001 = 4 mA 011 = 8 mA 111 = 12 mA
Reserved Bits 15	Reserved—This bit is reserved and should read 0.	
DS_SLOW5 Bits 14–12	**Driving Strength Slow I/O**—Controls the driving strength of the OSBOTG and UART4 I/O signals.	000 = 2 mA 001 = 4 mA 011 = 8 mA 111 = 12 mA
DS_SLOW4 Bits 11–9	**Driving Strength Slow I/O**—Controls the driving strength of the SSI1, SSI2, SAP, GPT and SSI3 I/O signals.	000 = 2 mA 001 = 4 mA 011 = 8 mA 111 = 12 mA
DS_SLOW3 Bits 8–6	**Driving Strength Slow I/O**—Controls the driving strength of the I^2C, CSPI1 and CSPI2 I/O signals.	000 = 2 mA 001 = 4 mA 011 = 8 mA 111 = 12 mA
DS_SLOW2 Bits 5–3	**Driving Strength Slow I/O**—Controls the driving strength of the RESET_OUT, KP, JTAG, UART1, UART2, UART3 and PWM I/O signals.	000 = 2 mA 001 = 4 mA 011 = 8 mA 111 = 12 mA
DS_SLOW1 Bits 2–0	**Driving Strength Slow I/O**—Controls the driving strength of the SD1 I/O signals.	000 = 2 mA 001 = 4 mA 011 = 8 mA 111 = 12 mA

8.1.7 Driving Strength Control Register 2

The Driving Strength Control Register 2 (DSCR2) controls the driving force parameters of the fast I/O signals in the i.MX21. Descriptions of the register settings appear in Table 8-7.

DSCR2 — Driving Strength Control Register 2 — Addr 0x10027824

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		DS_D22			DS_D23			DS_D24			DS_D25			DS_D26		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		DS_D27			DS_D28			DS_D29			DS_D30			DS_D31		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 8-7. Driving Strength Control Register 2 Description

Name	Description	Settings
Reserved Bit 31	Reserved—This bit is reserved and should read 0.	
DS_D22 Bits 30–28	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D22.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D23 Bits 27–25	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D23.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D24 Bits 24–22	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D24.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D25 Bits 21–19	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D25.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D26 Bits 18–16	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D26.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
Reserved Bit 15	Reserved—This bit is reserved and should read 0.	

Table 8-7. Driving Strength Control Register 2 Description (continued)

Name	Description	Settings
DS_D27 Bits 14–12	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D27.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D28 Bits 11–9	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D28.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D29 Bits 8–6	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D29.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D30 Bits 5–3	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D30.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D31 Bits 2–0	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D31.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA

8.1.8 Driving Strength Control Register 3

The Driving Strength Control Register 3 (DSCR3) controls the driving force parameters of the fast I/O signals in the i.MX21. Descriptions of the register settings appear in Table 8-8.

DSCR3 — Driving Strength Control Register 3 — Addr 0x10027828

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		DS_D12			DS_D13			DS_D14			DS_D15			DS_D16		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		DS_D17			DS_D18			DS_D19			DS_D20			DS_D21		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 8-8. Driving Strength Control Register 3 Description

Name	Description	Settings
Reserved Bit 31	Reserved—This bit is reserved and should read 0.	
DS_D12 Bits 30–28	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D12.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D13 Bits 27–25	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D13.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D14 Bits 24–22	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D14.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D15 Bits 21–19	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D15.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D16 Bits 18–16	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D16.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
Reserved Bit 15	Reserved—This bit is reserved and should read 0.	
DS_D17 Bits 14–12	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D17.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D18 Bits 11–9	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D18.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D19 Bits 8–6	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D19.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D20 Bits 5–3	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D20.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D21 Bits 2–0	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D21.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA

8.1.9 Driving Strength Control Register 4

The Driving Strength Control Register 4 (DSCR4) controls the driving force parameters of the fast I/O signals in the i.MX21. Descriptions of the register settings appear in Table 8-9.

DSCR4 — Driving Strength Control Register 4 — Addr 0x1002782C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		DS_D2			DS_D3			DS_D4			DS_D5			DS_D6		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		DS_D7			DS_D8			DS_D9			DS_D10			DS_D11		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 8-9. Driving Strength Control Register 4 Description

Name	Description	Settings
Reserved Bits 31	Reserved—This bit is reserved and should read 0.	
DS_D2 Bits 30–28	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D2.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D3 Bits 27–25	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D3.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D4 Bits 24–22	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D4.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D5 Bits 21–19	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D5.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D6 Bits 18–16	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D6.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
Reserved Bits 15	Reserved—This bit is reserved and should read 0.	

Table 8-9. Driving Strength Control Register 4 Description (continued)

Name	Description	Settings
DS_D7 Bits 14–12	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D7.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D8 Bits 11–9	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D8.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D9 Bits 8–6	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D9.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D10 Bits 5–3	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D10.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D11 Bits 2–0	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D11.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA

8.1.10 Driving Strength Control Register 5

The Driving Strength Control Register 5 (DSCR5) controls the driving force parameters of the fast I/O signals in the i.MX21. Descriptions of the register settings appear in Table 8-10.

DSCR5 — Driving Strength Control Register 5 — Addr 0x10027830

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		DS_A18			DS_A19			DS_A20			DS_A21_NFIO11			DS_A22_NFIO12		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		DS_A23_NFIO13			DS_A24_NFIO14			DS_A25_NFIO15			DS_D0			DS_D1		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 8-10. Driving Strength Control Register 5 Description

Name	Description	Settings
Reserved Bit 31	Reserved—This bit is reserved and should read 0.	
DS_A18 Bits 30–28	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A18.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A19 Bits 27–25	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A19.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A20 Bits 24–22	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A20.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A21_NFIO11 Bits 21–19	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A21_NFIO11.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A22_NFIO12 Bits 18–16	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A22_NFIO12.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
Reserved Bit 15	Reserved—This bit is reserved and should read 0.	
DS_A23_NFIO13 Bits 14–12	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A23_NFIO13.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A24_NFIO14 Bits 11–9	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A24_NFIO14.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A25_NFIO15 Bits 8–6	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A25_NFIO15.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D0 Bits 5–3	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D0.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_D1 Bits 2–0	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin D1.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA

8.1.11 Driving Strength Control Register 6

The Driving Strength Control Register 6 (DSCR6) controls the driving force parameters of the fast I/O signals in the i.MX21. Descriptions of the register settings appear in Table 8-11.

DSCR6 — Driving Strength Control Register 6 — Addr 0x10027834

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		DS_A8			DS_A9			DS_A10			DS_A11			DS_A12		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		DS_A13_NFIO8			DS_A14_NFIO9			DS_A15_NFIO10			DS_A16			DS_A17		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 8-11. Driving Strength Control Register 6 Description

Name	Description	Settings
Reserved Bit 31	Reserved—This bit is reserved and should read 0.	
DS_A8 Bits 30–28	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A8.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A9 Bits 27–25	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A9.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A10 Bits 24–22	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A10.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A11 Bits 21–19	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A11.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A12 Bits 18–16	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A12.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
Reserved Bit 15	Reserved—This bit is reserved and should read 0.	

Table 8-11. Driving Strength Control Register 6 Description (continued)

Name	Description	Settings
DS_A13_NFIO8 Bits 14–12	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A13_NFIO8.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A14_NFIO9 Bits 11–9	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A14_NFIO9.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A15_NFIO10 Bits 8–6	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A15_NFIO10.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A16 Bits 5–3	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A16.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A17 Bits 2–0	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A17.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA

8.1.12 Driving Strength Control Register 7

The Driving Strength Control Register 7 (DSCR7) controls the driving force parameters of the fast I/O signals in the i.MX21. Descriptions of the register settings appear in Table 8-12.

DSCR7 — Driving Strength Control Register 7 — Addr 0x10027838

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		DS_BCLK			DS_OE_B			DS_A0			DS_A1			DS_A2		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		DS_A3			DS_A4			DS_A5			DS_A6			DS_A7		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 8-12. Driving Strength Control Register 7 Description

Name	Description	Settings
Reserved Bits 31	Reserved—This bit is reserved and should read 0.	
DS_BCLK Bits 30–28	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin BCLK (EIM Burst Clock).	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_OE_B Bits 27–25	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin OE_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A0 Bits 24–22	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A0.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A1 Bits 21–19	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A1.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A2 Bits 18–16	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A2.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
Reserved Bit 15	Reserved—This bit is reserved and should read 0.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A3 Bits 14–12	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A3.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A4 Bits 11–9	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A4.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A5 Bits 8–6	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A5.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A6 Bits 5–3	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A6.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_A7 Bits 2–0	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin A7.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA

8.1.13 Driving Strength Control Register 8

The Driving Strength Control Register 8 (DSCR8) controls the driving force parameters of the fast I/O signals in the i.MX21 Driving Strength Control Register. Descriptions of the register settings appear in Table 8-13.

DSCR8 — Driving Strength Control Register 8 — Addr 0x1002783C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		DS_CS0_B			DS_CS1_B			DS_CS2_B			DS_CS3_B			DS_CS4_B		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		DS_CS5_B			DS_EB3_B			DS_EB2_B			DS_EB1_B			DS_EB0_B		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 8-13. Driving Strength Control Register 8 Description

Name	Description	Settings
Reserved Bit 31	Reserved—This bit is reserved and should read 0.	
DS_CS0_B Bits 30–28	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin CS0_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_CS1_B Bits 27–25	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin CS1_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_CS2_B Bits 24–22	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin CS2_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_CS3_B Bits 21–19	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin CS3_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_CS4_B Bits 18–16	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin CS4_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA

Table 8-13. Driving Strength Control Register 8 Description (continued)

Name	Description	Settings
Reserved Bit 15	Reserved—This bit is reserved and should read 0.	
DS_CS5_B Bits 14–12	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin CS5_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_EB3_B Bits 11–9	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin EB3_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_EB2_B Bits 8–6	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin EB2_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_EB1_B Bits 5–3	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin EB1_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_EB0_B Bits 2–0	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin EB0_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA

8.1.14 Driving Strength Control Register 9

The Driving Strength Control Register 9 (DSCR9) controls the driving force parameters of the fast I/O signals in the i.MX21 Driving Strength Control Register. Descriptions of the register settings appear in Table 8-14.

DSCR9 — Driving Strength Control Register 9 — Addr 0x10027840

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		DS_CAS_B			DS_RAS_B			DS_SDCLK								
TYPE	r	rw			rw			rw			r				r	
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		DS_MA10			DS_MA11			DS_RW_B			DS_LBA_B					
TYPE	r	rw			rw			rw			rw				r	
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 8-14. Driving Strength Control Register 9 Description

Name	Description	Settings
Reserved Bit 31	Reserved—This bit is reserved and should read 0.	
DS_CAS_B Bits 30–28	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin CAS_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_RAS_B Bits 27–25	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin RAS_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_SDCLK Bits 24–22	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin SDCLK.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
Reserved Bits 21–15	Reserved—These bits are reserved and should read 0.	
DS_MA10 Bits 14–12	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin MA10.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_MA11 Bits 11–9	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin MA11.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_RW_B Bits 8–6	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin RW_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_LBA_B Bits 5–3	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin LBA_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
Reserved Bits 2–0	Reserved—These bits are reserved and should read 0.	

8.1.15 Driving Strength Control Register 10

The Driving Strength Control Register 10 (DSCR10) controls the driving force parameters of the fast I/O signals in the i.MX21 Driving Strength Control Register. Descriptions of the register settings appear in Table 8-15.

DSCR10 — Driving Strength Control Register 10 — Addr 0x10027844

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		DS_NFIO3			DS_NFIO4			DS_NFIO5			DS_NFIO6			DS_NFIO7		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		DS_CLKO			DS_PF16			DS_SDCKE1			DS_SDCKE0			DS_SDWE_B		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 8-15. Driving Strength Control Register 10 Description

Name	Description	Settings
Reserved Bit 31	Reserved—This bit is reserved and should read 0.	
DS_NFIO3 Bits 30–28	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin NFIO3.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_NFIO4 Bits 27–25	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin NFIO4.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_NFIO5 Bits 24–22	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin NFIO5.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_NFIO6 Bits 21–19	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin NFIO6.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_NFIO7 Bits 18–16	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin NFIO7.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
Reserved Bit 15	Reserved—This bit is reserved and should read 0.	

Table 8-15. Driving Strength Control Register 10 Description (continued)

Name	Description	Settings
DS_CLKO Bits 14–12	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin CLKO.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_PF16 Bits 11–9	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin PF16.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_SDCKE1 Bits 8–6	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin SDCKE1.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_SDCKE0 Bits 5–3	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin SDCKE0.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_SDWE_B Bits 2–0	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin SDWE_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA

8.1.16 Driving Strength Control Register 11

The Driving Strength Control Register 11 (DSCR11) controls the driving force parameters of the fast I/O signals in the i.MX21 Driving Strength Control Register. Descriptions of the register settings appear in Table 8-6.

DSCR11 — Driving Strength Control Register 11 — Addr 0x10027848

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		DS_NFRB			DS_NFCE_B			DS_NFWP_B			DS_NFCLE			DS_NFALE		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		DS_NFRE_B			DS_NFWE_B			DS_NFIO0			DS_NFIO1			DS_NFIO2		
TYPE	r	rw			rw			rw			rw			rw		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 8-16.

Name	Description	Settings
Reserved Bit 31	Reserved—This bit is reserved and should read 0.	
DS_NFRB Bits 30–28	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin NFRB.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_NFCE_B Bits 27–25	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin NFCE_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_NFWP_B Bits 24–22	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin NFWP_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_NFCLE Bits 21–19	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin NFCLE.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_NFALE Bits 18–16	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin NFALE.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
Reserved Bit 15	Reserved—This bit is reserved and should read 0.	
DS_NFRE_B Bits 14–12	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin NFRE_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_NFWE_B Bits 11–9	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin NFWE_B.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_NFIO0 Bits 8–6	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin NFIO0.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_NFIO1 Bits 5–3	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin NFIO1.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_NFIO2 Bits 2–0	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin NFIO2.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA

8.1.17 Driving Strength Control Register 12

The Driving Strength Control Register 12 (DSCR12) controls the driving force parameters of the fast I/O signals in the i.MX21 Driving Strength Control Register. Descriptions of the register settings appear in Table 8-17.

DSCR12 — Driving Strength Control Register 12 — Addr 0x1002784C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r		r			r			rw			r			r	
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								DS_LSCLK			DS_CSI_MCLK			DS_CSI_PIXCLK		
TYPE	r		r			r			rw			rw			rw	
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 8-17. Driving Strength Control Register 12 Description

Name	Description	Settings
Reserved Bits 31–9	Reserved—These bits are reserved and should read 0.	
DS_LSCLK Bits 8–6	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin LSCLK.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_CSI_MCLK Bits 5–3	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin CSI_MCLK.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA
DS_CSI_PIXCLK Bits 2–0	**Driving Strength Fast I/O**—Controls the driving strength of fast I/O signals on pin CSI_PIXCLK.	000 = 3.5 mA 001 = 4.5 mA 011 = 5.5 mA 111 = 6.5 mA

8.1.18 Priority Control and Select Register

The Priority Control and Select Register (PCSR) consist of the slave alternate context priority select to the ARM9 platform. Descriptions of the register settings appear in Table 8-18.

PCSR — Priority Control and Select Register Description — Addr 0x10027850

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													s3_ ampr_sel	s2_ ampr_sel	s1_ ampr_sel	s0_ ampr_sel
TYPE							r						rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE					r						rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
	0x0003															

Table 8-18. Global Peripheral Control Register Description

Name	Description	Settings
Reserved Bits 31–20	Reserved—These bits are reserved and should read 0.	
s3_ampr_sel **s2_ampr_sel** **s1_ampr_sel** **s0_ampr_sel** Bits 19–16	**Slave Alternate Context Priority Select**—Inputs to the ARM9 platform to select the priority determination and control source for the appropriate slave port. (Note s0 is the primary AHB and does not come out of the ARM9 platform.	0 = Priority determination and control is made by regular registers. 1 = Priority determination and control is made by alternate registers set in the Crossbar switch.
Reserved Bits 15–0	Reserved—These bits are reserved and should read 0.	

8.2 System Boot Mode Selection

The operational system boot mode of the i.MX21 upon POR reset is determined by the configuration of the four external input pins, BOOT[3:0]. The settings of these pins control where the system is boot from and the memory port size.

NOTE

The BOOT pins must not change once the i.MX21 is out of POR reset. For proper operation, BOOT[3] must always be tied to VSS.

Table 8-19. System Boot Mode Selection

Inputs BOOT[3:0]	Output Signals Active Device	Boot Address
0000	iROM (Bootstrap USB/UART)	0x00000030
0001	iROM (Bootstrap USB/UART)	0x00000030
0010	iROM (8-bit 2 Kbyte NAND Flash)	0xDF003000
0011	iROM (16-bit 2 Kbyte NAND Flash)	0xDF003000
0100	iROM (16-bit 512 byte NAND Flash)	0xDF003000
0101	iROM (16-bit CS0 at D[15:0] (NOR Flash))	0xC8000000
0110	iROM (32-bit CS0 at D[31:0] (NOR Flash))	0xC8000000
0111	iROM (8-bit 512 byte NAND Flash)	0xDF003000

Chapter 9
Internal ROM, System Boot Manager

This chapter describes the system boot up sequence of the i.MX21 processor. The system bootup is designed according to the configuration of the external BOOT pins, and hence sub-divided into Boot-external, Boot-internal, In-Factory test, Bootstrap mode and Normal flash bootup modes.

The i.MX21 processor comes with a 24 Kbyte internal ROM (iROM) which contains a System Boot Manager to process the serial bootstrap operations according to all possible scenarios from power-up. The iROM is controlled by various pins and laser fuse configurations, for details please refer to Chapter 8, “System Control.”

The system Boot Manager allows multiple-layers of boot-up, for example, NAND Flash boot-up. Different from NOR Flash, NAND Flash requires setting up the file system and caching system to access inner memory content. Before the file system is setup, only the first page of content is accessible (2–16 Kbytes).

To interface with an external terminal such as a PC, a bootstrap program with a proprietary protocol is designed to interact with the external computer through the use of a serial channel UART, or USB. The bootstrap program must be entered when the corresponding boot pins are selected (please refer to Chapter 8, “System Control,” for details of the pin selection). When programmed in this manner, the system will automatically detect the existence of a serial channel during power-up.

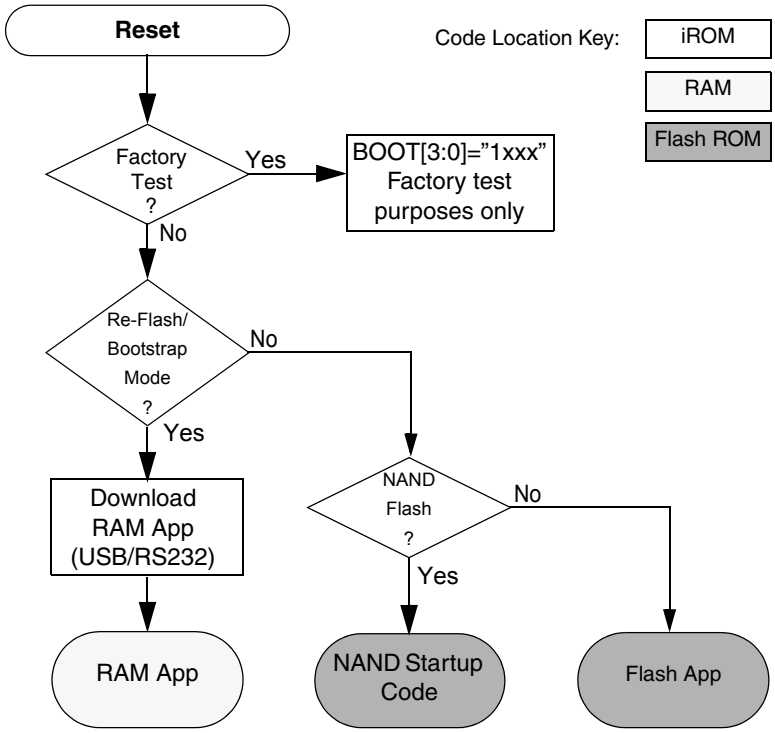


Figure 9-1. iROM Boot Flow Diagram (Simplified)

9.1 Bootstrap Mode Operation

The bootstrap program is a program that resides in the internal ROM of i.MX21. It is activated when the BOOT[3:0] selection pins are configured as Bootstrap Mode. The program handles the commands from either the USB or UART1 to establish a channel to interface with the i.MX21 processor's hardware and an external machine such as a PC. The program downloads a binary image to memory so as to execute in run time or perform Flash update.

9.1.1 UART/USB Configuration

The configuration for RS 232 is using baud rate 115200, 8 Data bits, No Parity, 1 Stop bits and No Flow Control. The Configuration for USB is for Control Endpoint 0 with Max Packet Size equal 8 byte. Bulk IN at Endpoint 2 with Max Packet Size equal 64 bytes, Bulk OUT at Endpoint 1 with Max Packet Size equal 64 bytes.

9.1.2 Enter Bootstrap Mode Configuration

The i.MX21 processor enters bootstrap mode under the following conditions:

1. BOOT[3:0] is selected for bootstrap mode

9.1.3 Bootstrap Flow

The overall flow of the bootstrap program is shown in Figure 9-2.

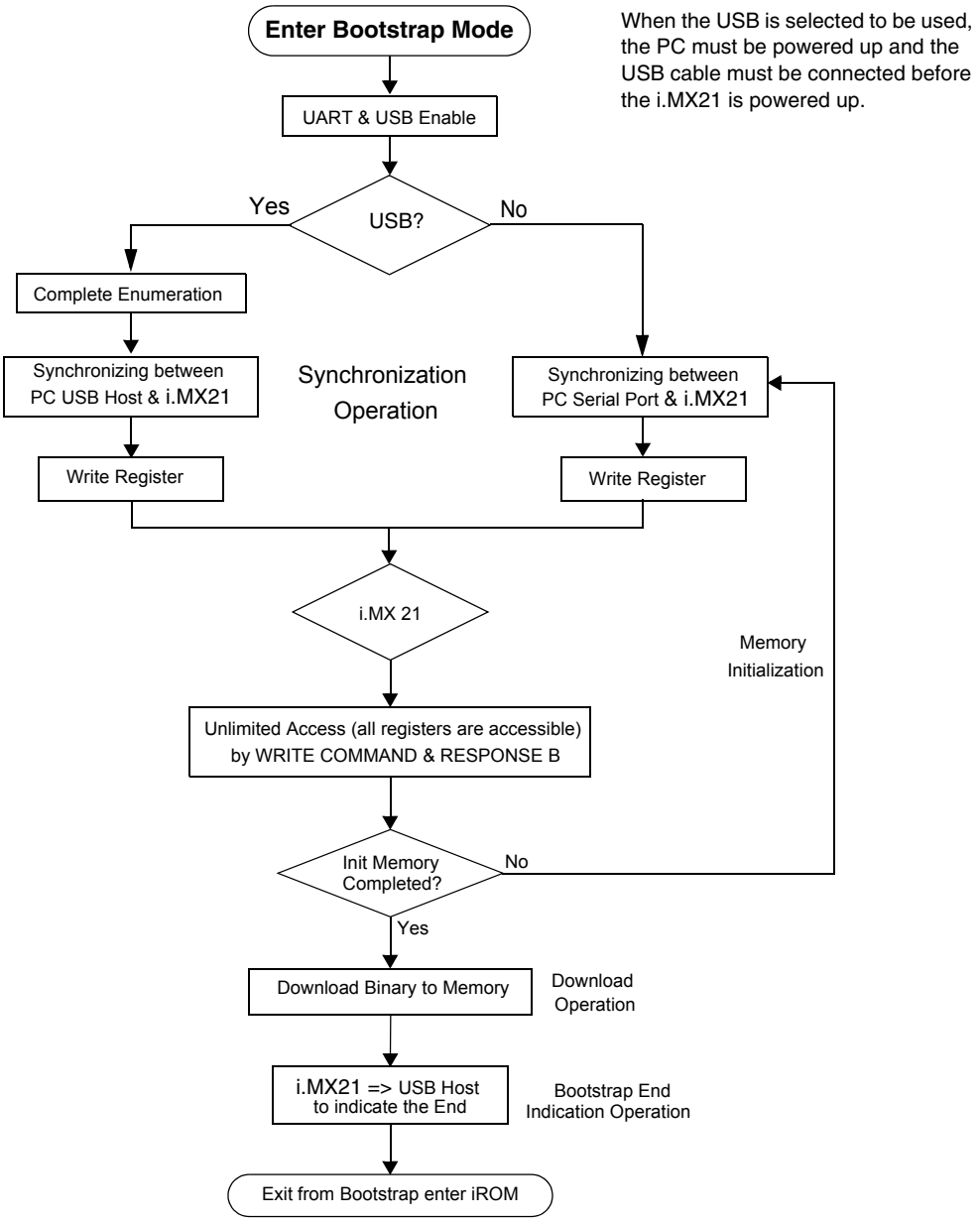


Figure 9-2. Flow Diagram for Bootstrap Mode

9.1.3.1 Bootstrap Protocol and Definition

In this section, bootstrap protocol and the command, response definition is defined. For the i.MX21 processor's boot-up sequence, please refer to System Boot.

9.1.3.1.1 Synchronization Operation

When bootstrap is first entered, the status of the iROM can be obtained by issuing the command shown in Figure 9-3.

Figure 9-3. iROM Status Command

The SYNC COMMAND is composed of 16 bytes using the format shown in Table 9-1. The RESPONSE is 4 bytes long and indicates that there is communication with the processor.

Table 9-1. Synch Command Definition

Header (2 bytes)	Address (4 bytes)	Format (1 byte)	Bytecount (4 bytes)	Data (4 bytes)	End (1 byte)
0505	00000000	00	00000000	00000000	00

9.1.3.1.2 Write Register Operation

To write to a register through bootstrap, requires a specific protocol. After the command is sent from the PC to the i.MX21 processor, two responses are returned from i.MX21. The second response, RESPONSE C, is used to indicate whether the write operation is successful.

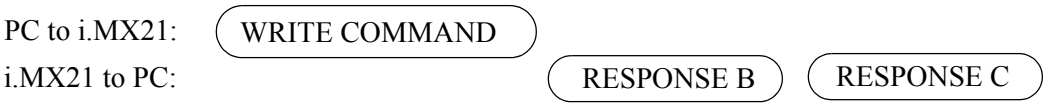


Figure 9-4. Write Register Command

WRITE COMMAND is 16 bytes long using the format shown in Table 9-2.

Table 9-2. Write Register Command Definition

Header (2 bytes)	Address (4 bytes)	Format (1 byte)	Bytecount (4 bytes)	Data (4 bytes)	End (1 byte)
0202	Address to be written	Format to be written (08: byte access 10: halfword access 20: word access)	00	Data to be written to the register	00

RESPONSE B indicates the type of silicon. It is composed of 8 bytes using the format shown in Table 9-3.

Table 9-3. Response B Definition

	Byte 0	Byte 1	Byte 2	Byte 3
Development part	56	78	78	56

RESPONSE C indicates the success of a write operation as shown in Table 9-4.

Table 9-4. Response C Definition

Byte 0	Byte 1	Byte 2	Byte 3
12	8A	8A	12

9.1.3.1.3 Download Operation

To download a binary file to the memory, after the memory is initialized. The following command can be issued:

Figure 9-5. Download Command

The DOWNLOAD COMMAND is 16 bytes long using the formats shown in Table 9-5.

Table 9-5. Download Command Definition

	Header (2 bytes)	Address (4 Bytes)	Format (1 byte)	Bytecount (4 bytes)	Data (4 bytes)	End (1 byte)
Image file	0404	Start address where the binary data is to be downloaded	00	Number of byte to be written in Hex (max 0x1F0000)	Start address in memory where data is to be written	00
Image file	0404	Start address where the binary data is to be downloaded	00	Number of byte to be written in Hex	Start address in memory where data is to be written	AA

RESPONSE B is 8 bytes long using the format shown in Table 9-6.

Table 9-6. Response B Silicon Type Definition

	BYTE 0	BYTE 1	BYTE 2	BYTE 3
Development part	56	78	78	56

After the RESPONSE B is received by the i.MX21 processor, the attached PC can start to download the binary data to i.MX21 until all the BYTECOUNT is downloaded. Each time the Image File is downloaded through HEADER (0404), the maximum data to be download is 0x1F0000. Thus, if the Image File size is greater then 0x1F0000, it will send the command repeatedly with END (0x00). After all the data is downloaded, PC must send a DOWNLOAD command with END (AA) to the target execution address.

9.1.3.1.4 Bootstrap End Indication Operation

After all the bootstrap operations are completed, the i.MX21 processor will send RESPONSE D to PC after the Application Pointer was sent to indicate bootstrap was completed. After RESPONSE D is complete, the i.MX21 processor executes the image.

Figure 9-6. Bootstrap End Indication Operation Diagram

RESPONSE D is indicates the success of the write operation as shown in Table 9-7.

Table 9-7. Bootstrap End Indication Operation Diagram

BYTE 0	BYTE 1	BYTE 2	BYTE 3
88	88	88	88

Chapter 10
Multi-layer AHB Crossbar Switch (MAX)

This section provides an overview of the MAX (Multi-Layer AHB Crossbar Switch). The purpose of the MAX is to concurrently support simultaneous connections between 6 master ports and 4 slave ports.

The features of MAX include the following:

- Supports concurrent transfers between 6 master ports and 4 slave ports. The MAX has the ability to gain control of all the slave ports and prevent any masters from making accesses to the slave ports.
- The MAX can put each slave port into a low-power park mode so that a slave port will not dissipate any power transitioning address, control, or data signals when not being actively accessed by a master port.
- Each slave port can also support multiple master priority schemes.
- The MAX allows for concurrent transactions to occur from any master port to any slave port. It is possible for all master ports and slave ports to be in use at the same time as a result of independent master requests. If a slave port is simultaneously requested by more than one master port, arbitration logic selects the higher priority master and grants it ownership of the slave port. All other masters requesting that slave port are stalled until the higher priority master completes its transactions.

10.1 Limitations

The MAX routes bus transactions initiated on the master ports to the appropriate slave ports. There is no provision included to route transactions initiated on the slave ports to other slave ports or to master ports. Simply put, the slave ports do not support the bus request/bus grant protocol, the MAX assumes it is the sole master of each slave port.

10.2 General Operation

When a master makes an access to the MAX the access will be immediately taken by the MAX. If the targeted slave port of the access is available then the access will be immediately presented on the slave port. It is possible to make single clock (zero wait state) accesses through the MAX. If the targeted slave port of the access is busy or parked on a different master port the requesting master will simply see wait states inserted until the targeted slave port can service the master's request. The latency in servicing the request will depend on each master's priority level and the responding peripheral's access time.

Since the MAX appears to be just another slave to the master device, the master device will have no knowledge of whether or not it actually owns the slave port it is targeting. While the master does not have control of the slave port it is targeting it will simply be wait stated.

A master will be given control of the targeted slave port only after a previous access to a different slave port has completed, regardless of its priority on the newly targeted slave port. This prevents deadlock from occurring when a master has an outstanding request to one slave port that has a long response time, has a pending access to a different slave port, and a lower priority master is also making a request to the same slave port as the pending access of the higher priority master.

Once the master has control of the slave port it is targeting the master will remain in control of that slave port until it gives up the slave port by running an IDLE cycle or by leaving that slave port for its next access. The master could also lose control of the slave port if another higher priority master makes a request to the slave port; however, if the master is running a locked or fixed length burst transfer it will retain control of the slave port until that transfer is completed. Based on the AULB bit in the MGPCR (Master General Purpose Control Register) the master will either retain control of the slave port when doing undefined length incrementing burst transfers or will lose the bus to a higher priority master.

The MAX will terminate all master IDLE transfers (as opposed to allowing the termination to come from one of the slave busses). Additionally, when no master is requesting access to a slave port the MAX will drive IDLE transfers onto the slave bus. When a slave bus is being IDLEd by the MAX it can park the slave port on the master port indicated by the PARK bits in the SGPCR (Slave General Purpose Control Register) or ASGPCR (Alternate SGPCR). This can be done in an attempt to save the initial clock of arbitration delay that would otherwise be seen if the master had to arbitrate to gain control of the slave port. The slave port can also be put into low power park mode in an attempt to save power.

10.3 Programming Model

This section provides the register programming information for the MAX registers.

There are four registers that reside in each slave port of the MAX and one register that resides in each master port of the MAX. These registers are IP bus compliant registers. Read and write transfers both require two IP bus clock cycles. The registers can only be read from and written to in supervisor mode. Additionally, these registers can only be read from or written to by 32-bit accesses.

CAUTION

The MAX registers are fully decoded and an error response is returned if an unimplemented location is accessed within the MAX.

The slave registers also feature a bit, which when written with a 1, will prevent the registers from being written to again. The registers will still be readable, but future write attempts will have no effect on the registers and will be terminated with an error response.

10.3.1 Master Priority Register

The Master Priority Register (MPR) sets the priority of each master port on a per slave port basis and resides in each slave port.

The Master Priority Register can only be accessed in supervisor mode with 32-bit accesses. Once the RO (Read Only) bit has been set in the slave General Purpose Control Register the Master Priority Register can only be read from, attempts to write to it will have no effect on the MPR and result in an error response.

Additionally, no two available master ports may be programmed with the same priority level. Attempts to program two or more available masters with the same priority level will result in an error response and the MPR will not be updated.

MPR0	Master Priority Register for Slave Port 0	Addr
MPR1	Master Priority Register for Slave Port 1	0x1003_F000
MPR2	Master Priority Register for Slave Port 2	0x1003_F100
MPR3	Master Priority Register for Slave Port 3	0x1003_F200
		0x1003_F300

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME										MSTR_5				MSTR_4		
TYPE	r	rw	rw	rw	r	rw	rw	rw	r	rw	rw	rw	r	rw	rw	rw
RESET	0	1	1	1	0	1	1	0	0	1	0	1	0	1	0	0
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME		MSTR_3				MSTR_2				MSTR_1				MSTR_0		
TYPE	r	rw	rw	rw	r	rw	rw	rw	r	rw	rw	rw	r	rw	rw	rw
RESET	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0

Table 10-1. Master Priority Register Description

Name	Description	Settings
Reserved Bit 31–23	**Reserved**—These bits are reserved for future expansion. These are read as zero and should be written with zero for upward compatibility.	
MSTR_5 Bits 22–20	**Master 5 Priority**—These bits set the arbitration priority for master port 5 on the associated slave port. These bits are initialized by hardware reset. The reset value is 101.	000 = This master has the highest priority when accessing the slave port. 111 = This master has the lowest priority when accessing the slave port.
Reserved Bit 19	**Reserved**—This bit is reserved for future expansion. It is read as zero and should be written with zero for upward compatibility.	
MSTR_4 Bits 18–16	**Master 4 Priority**—These bits set the arbitration priority for master port 4 on the associated slave port. These bits are initialized by hardware reset. The reset value is 100.	000 = This master has the highest priority when accessing the slave port. 111 = This master has the lowest priority when accessing the slave port.
Reserved Bit 15	**Reserved**—This bit is reserved for future expansion. It is read as zero and should be written with zero for upward compatibility.	
MSTR_3 Bits 14–12	**Master 3 Priority**—These bits set the arbitration priority for master port 3 on the associated slave port. These bits are initialized by hardware reset. The reset value is 011.	000 = This master has the highest priority when accessing the slave port. 111 = This master has the lowest priority when accessing the slave port.
Reserved Bit 11	**Reserved**—This bit is reserved for future expansion. It is read as zero and should be written with zero for upward compatibility.	
MSTR_2 Bits 10–8	**Master 2 Priority**—These bits set the arbitration priority for master port 2 on the associated slave port. These bits are initialized by hardware reset. The reset value is 010.	000 = This master has the highest priority when accessing the slave port. 111 = This master has the lowest priority when accessing the slave port.

Table 10-1. Master Priority Register Description (continued)

Name	Description	Settings
Reserved Bit 7	**Reserved**—This bit is reserved for future expansion. It is read as zero and should be written with zero for upward compatibility.	
MSTR_1 Bits 6–4	**Master 1 Priority**—These bits set the arbitration priority for master port 1 on the associated slave port. These bits are initialized by hardware reset. The reset value is 001.	000 = This master has the highest priority when accessing the slave port. 111 = This master has the lowest priority when accessing the slave port.
Reserved Bit 3	**Reserved**—This bit is reserved for future expansion. It is read as zero and should be written with zero for upward compatibility.	
MSTR_0 Bits 2–0	**Master 0 Priority**—These bits set the arbitration priority for master port 0 on the associated slave port. These bits are initialized by hardware reset. The reset value is 000.	000 = This master has the highest priority when accessing the slave port. 111 = This master has the lowest priority when accessing the slave port.

10.3.2 Alternate Master Priority Register

The Alternate Master Priority Register (AMPR) sets the alternate priority of each master port on a per slave port basis.

The AMPR has identical function as the MPR. The purpose of the AMPR is to allow the user to set up an alternate set of priorities in the event they want to do some sort of context switching. A hardware input to the MAX controls (on a slave port by slave port basis) whether or not the slave port uses the MPR or the AMPR.

AMPR0	Alternate Master Priority Register for Slave Port 0	0x1003_F004
AMPR1	Alternate Master Priority Register for Slave Port 1	0x1003_F104
AMPR2	Alternate Master Priority Register for Slave Port 2	0x1003_F204
AMPR3	Alternate Master Priority Register for Slave Port 3	0x1003_F304

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME										MSTR_5				MSTR_4		
TYPE	r	rw	rw	rw	r	rw	rw	rw	r	rw	rw	rw	r	rw	rw	rw
RESET	0	1	1	1	0	1	1	0	0	1	0	1	0	1	0	0
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME		MSTR_3				MSTR_2				MSTR_1				MSTR_0		
TYPE	r	rw	rw	rw	r	rw	rw	rw	r	rw	rw	rw	r	rw	rw	rw
RESET	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0

Please reference Table 10-1 for descriptions of bit fields in the AMPR as they are identical.

The Alternate Master Priority Register can only be accessed in supervisor mode with 32-bit accesses. Once the RO (Read Only) bit has been set in the General Purpose Control Register the Alternate Master Priority Register can only be read from, attempts to write to it will have no effect on the AMPR and result in an error response.

Additionally, no two available master ports may be programmed with the same priority level. Attempts to program two or more available masters with the same priority level will result in an error response and the AMPR will not be updated.

10.3.3 Slave General Purpose Control Register

The Slave General Purpose Control Register (SGPCR) controls several features of each slave port.

The Read Only (RO) bit will prevent any registers associated with this slave port from being written to once set. This bit may be written with 0 as many times as the user desires, but once it is written to a 1 only a reset condition will allow it to be written again.

The Halt Low Priority (HLP) bit will set the priority of the `max_halt_request` input to the lowest possible priority for initial arbitration of the slave ports. By default it is the highest priority. Please note, setting this bit will not effect the `max_halt_request` from attaining highest priority once it has control of the slave ports.

The PCTL bits determine how the slave port will park when no master is actively making a request. The available options are to park on the master defined by the PARK bits, park on the last master to use the slave port, or go into a low-power park mode which will force all the outputs of the slave port to inactive states when no master is requesting an access. The low power park feature can result in an overall power savings if a the slave port is not saturated; however, it will force an extra clock of latency whenever any master tries to access it when it is not in use because it will not be parked on any master.

The PARK bits determine which master the slave will park on when no master is making an active request and the `max_halt_request` input is negated. Please use caution to only select master ports that are actually present in the design. If the user programs the PARK bits to a master not present in the current design implementation undefined behavior will result.

The SGPCR can only be accessed in supervisor mode with 32-bit accesses. After the RO (Read Only) bit has been set in the SGPCR the SGPCR can only be read, attempts to write to it will have no effect on the SGPCR and result in an error response.

SGPCR0	Slave General Purpose Control Register for Slave Port 0	0x1003_F010
SGPCR1	Slave General Purpose Control Register for Slave Port 1	0x1003_F110
SGPCR2	Slave General Purpose Control Register for Slave Port 2	0x1003_F210
SGPCR3	Slave General Purpose Control Register for Slave Port 3	0x1003_F310

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	RO	HLP														
TYPE	rw[1]	rw	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME							ARB				PCTL			PARK		
TYPE	r	r	r	r	r	r	rw	rw	r	r	rw	rw	r	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

[1]Once this bit is written to a 1 only hardware reset will return it to a 0.

Table 10-2. Slave General Purpose Control Register Description

Name	Description	Setting
RO Bit 31	**Read Only**—This bit is used to force all of a slave port's registers to be read only. Once written to 1 it can only be cleared by hardware reset. This bit is initialized by hardware reset. The reset value is 0	0 = All this slave port's registers can be written. 1 = All this slave port's registers are read only and cannot be written (attempted writes have no effect and result in an error response).
HLP Bit 30	**Halt Low Priority**—This bit is used to set the initial arbitration priority of the max_halt_request input. This bit is initialized by hardware reset. The reset value is 0	0 = The max_halt_request input has the highest priority for arbitration on this slave port. 1 = The max_halt_request input has the lowest initial priority for arbitration on this slave port.
Reserved Bits 29–10	**Reserved**—These bits are reserved for future expansion. They are read as zero and should be written with zero for upward compatibility.	
ARB Bits 9–8	**Arbitration Mode**—These bits are used to select the arbitration policy for the slave port. These bits are initialized by hardware reset. The reset value is 00	00 Fixed Priority. 01 Round Robin (rotating) Priority. 10 Reserved 11 Reserved
Reserved Bits 7–6	**Reserved**—These bits are reserved for future expansion. They are read as zero and should be written with zero for upward compatibility.	
PCTL Bits 5–4	**Parking Control**—These bits determine the parking control used by this slave port.These bits are initialized by hardware reset. The reset value is 00.	00 = When no master is making a request the arbiter will park the slave port on the master port defined by the PARK bit field. 01 = When no master is making a request the arbiter will park the slave port on the last master to be in control of the slave port. 10 = When no master is making a request the arbiter will park the slave port on no master and will drive all outputs to a constant safe state. 11 = Reserved
Reserved Bit 3	**Reserved**—This bit is reserved for future expansion. It is read as zero and should be written with zero for upward compatibility.	
PARK Bits 2–0	**PARK**—These bits are used to determine which master port this slave port parks on when no masters are actively making requests and the PCTL bits are set to 00. These bits are initialized by hardware reset. The reset value is 000	000 = Park on Master Port 0 001 = Park on Master Port 1 010 = Park on Master Port 2 011 = Park on Master Port 3 100 = Park on Master Port 4 101 = Park on Master Port 5 11x = Reserved

10.3.4 Alternate Slave General Purpose Control Register

The Alternate Slave General Purpose Control Register (ASGPCR) has identical function as the SGPCR with the notable exception that it lacks the RO (Read Only) bit contained in the SGPCR. The purpose of the ASGPCR is the same as the AMPR, to allow the user to set up an alternate set of general control fields in the event they want to do some sort of context switching. A hardware input to the MAX controls (on a slave port by slave port basis) whether or not the slave port uses the SGPCR or ASGPCR.

NOTE

The ASGPCR does not contain a RO (Read Only) bit. The RO bit in the SGPCR has control over the ASGPCR's ability to be written.

ASGPCR0	Alternate SGPCR for Slave Port 0	0x1003_F014
ASGPCR1	Alternate SGPCR for Slave Port 1	0x1003_F114
ASGPCR2	Alternate SGPCR for Slave Port 2	0x1003_F214
ASGPCR3	Alternate SGPCR for Slave Port 3	0x1003_F314

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME		HLP														
TYPE	r	rw	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME							ARB				PCTL			PARK		
TYPE	r	r	r	r	r	r	rw	rw	r	r	rw	rw	r	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Please reference Table 10-3 for descriptions of bit fields in the ASGPCR as they are identical except for the RO bit.

The ASGPCR can only be accessed in supervisor mode with 32-bit accesses. Once the RO (Read Only) bit has been set in the SGPCR the ASGPCR can only be read from, attempts to write to it will have no effect on the ASGPCR and result in an error response.

10.3.5 Master General Purpose Control Register

The Master General Purpose Control Register (MGPCR) presently controls only whether the master's undefined length burst accesses will be allowed to complete uninterrupted or whether they can be broken by requests from higher priority masters.

The AULB (Arbitrate on Undefined Length Bursts) bit determines whether the MAX will arbitrate away the slave port the master owns when the master is performing undefined length burst accesses. Arbitration occurs transparently and masters who lose control only see that their target slave is not ready to accept new cycles.

The MGPCR can only be accessed in supervisor mode with 32-bit accesses.

MGPCR0	Master General Purpose Control Register for Master Port 0	0x1003_F800
MGPCR1	Master General Purpose Control Register for Master Port 1	0x1003_F900
MGPCR2	Master General Purpose Control Register for Master Port 2	0x1003_FA00
MGPCR3	Master General Purpose Control Register for Master Port 3	0x1003_FB00
MGPCR4	Master General Purpose Control Register for Master Port 4	0x1003_FC00
MGPCR5	Master General Purpose Control Register for Master Port 5	0x1003_FD00

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME																
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME														AULB		
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 10-3. Master General Purpose Control Register Description

Name	Description	Setting
Reserved Bits 31–1	**Reserved**—These bits are reserved for future expansion. They are read as zero and should be written with zero for upward compatibility.	
AULB Bit 0	**Arbitrate on Undefined Length Bursts**—These bits are used to select the arbitration policy during undefined length bursts by this master. These bits are initialized by hardware reset. The reset value is 000.	000=No arbitration will be allowed during an undefined length burst. 001=Arbitration will be allowed at any time during an undefined length burst. 010=Arbitration will be allowed after four beats of an undefined length burst. 011=Arbitration will be allowed after eight beats of an undefined length burst. 100=Arbitration will be allowed after 16 beats of an undefined length burst. 101=Reserved 110=Reserved 111=Reserved

10.4 Function

This section describes the functionality of the MAX in greater detail.

10.4.1 Arbitration

The MAX supports two types of arbitration schemes, a simple fixed priority comparison algorithm and a round robin priority—that is, the ID is defined by the master port number and not the AHB hmaster signal. The desired algorithm is selected by the ARB bits in the SGPCR0 to SGPCR3 registers. The default setting of the ARB is to use a simple fixed priority comparison algorithm. In this simple fixed priority comparison algorithm, each master is assigned a unique priority level in the MPR and gains control over the slave port. In round robin (rotating) priority, all masters essentially have the same priority, and upon completion of

one master's access, the next master will initiate access, and upon completion, will move onto the next master, and so on, until all masters complete one access. Control then rotates back around to the first master to start the process again.

10.4.2 Arbitration During Undefined Length Bursts

Arbitration points during an undefined length burst are defined by the current master's MGPCR AULB field setting. When a defined length is imposed on the burst via the AULB bits the undefined length burst will be treated as a single or series of single back to back fixed length burst accesses.

Example: A master runs an undefined length burst and the AULB bits in the MGPCR indicate arbitration will occur after the fourth beat of the burst. The master runs two sequential beats and then starts what will be an 12 beat undefined length burst access to a new address within the same slave port region as the previous access. The MAX will not allow an arbitration point until the fourth overall access (second beat of the second burst). At that point all remaining accesses will be open for arbitration until the master loses control of the slave port.

Assume the master loses control of the slave port after the fifth beat of the second burst. Once the master regains control of the slave port no arbitration point will be available until after the master has run four more beats of its burst. After the fourth beat of the (now continued) burst (ninth beat of the second burst from the master's perspective) is taken all beats of the burst will once again be open for arbitration until the master loses control of the slave port. Assume the master again loses control of the slave port on the fifth beat of the third (now continued) burst (10th beat of the second burst from the master's perspective). Once the master regains control of the slave port it will be allowed to complete its final two beats of its burst without facing arbitration. Note that fixed length burst accesses will not be affected by the AULB bits. All fixed length burst accesses will lock out arbitration until the last beat of the fixed length burst.

10.4.3 Fixed Priority Operation

When operating in fixed-priority mode, each master is assigned a unique priority level in the MPR (Master Priority Register) and AMPR (Alternate Master Priority Register). If two masters both request access to a slave port the master with the highest priority in the selected priority register will gain control over the slave port. Any time a master makes a request to a slave port the slave port checks to see if the new requesting master's priority level is higher than that of the master that currently has control over the slave port (unless the slave port is in a parked state). The slave port does an arbitration check at every clock edge to ensure that the proper master (if any) has control of the slave port. If the new requesting master's priority level is higher than that of the master that currently has control of the slave port the new requesting master will be granted control over the slave port at the next clock edge.

The exception to this rule is if the master that currently has control over the slave port is running a fixed length burst transfer or a locked transfer. In this case the new requesting master will have to wait until the end of the burst transfer or locked transfer before it will be granted control of the slave port. If the master is running an undefined length burst transfer the new requesting master must wait until an arbitration point for the undefined length burst transfer before it will be granted control of the slave port. Arbitration points for an undefined length burst are defined in the MGPCR for each master.

If the new requesting master's priority level is lower than that of the master that currently has control of the slave port the new requesting master will be forced to wait until the master that currently has control of the slave port either runs an IDLE cycle or runs a non IDLE cycle to a location other than the current slave port.

10.4.4 Round-Robin Priority Operation

When operating in round-robin mode, each master is assigned a relative priority based on the master number. This relative priority is compared to the ID of the last master to perform a transfer on the slave bus. The highest priority requesting master will become owner of the slave bus as the next transfer boundary (accounting for locked and fixed-length burst transfers). Priority is based on how far ahead the ID of the requesting master is to the ID of the last master (ID is defined by master port number, not the **hmaster** field).

Once granted access to a slave port, a master may perform as many transfers as desired to that port until another master makes a request to the same slave port. The next master in line will be granted access to the slave port on the next clock cycle if the current master has no pending access request.

As an example of arbitration in round-robin mode, assume the MAX is implemented with master ports 0, 1, 2, 3, 4 and 5. If the last master of the slave port was master 1, and master 0, 4 and 5 make simultaneous requests, (master ports 2 and 3 make no requests), they will be serviced in the order 4, 5 and then 0.

Parking may still be used in a round-robin mode, but will not affect the round-robin pointer unless the parked master actually performs a transfer. Hand-off will occur to the next master in line after one cycle of arbitration. If the slave port is put into low power park mode the round-robin pointer will be reset to point at master port 0, giving it the highest priority.

10.4.5 Priority Assignment

Each master port needs to be assigned a unique 3 bit priority level. If an attempt is made to program multiple master ports with the same priority level within a register (MPR or AMPR) the MAX will respond with an error and the registers will not be updated.

10.4.5.1 Context Switching

The MAX has a hardware input per slave port (`sX_ampr_sel`) which is used to select which registers the master priority levels and general purpose control bits will be taken from. When `sX_ampr_sel` is 0 the MPR and SGPCR will be selected, when `sX_ampr_sel` is 1 the AMPR and the ASGPCR will be selected. This hardware input is useful for context switching so the user does not have to rewrite the MPR or SGPCR if a particular slave port would temporarily benefit from modifying the master priority levels or functions affected by the bits in the SGPCR.

Chapter 11
JTAG Controller

The JTAG Controller module supports Debug access to arm926 core, i.MX21 BIST test (excluding arm926 platform BIST, BootROM and VectorRAM bist) and tristate enable of the I/O pads. The overall strategy is to achieve good test and debug features without much increase on pin count and reduce the complexity on I/O muxing. The TAP ports—TCK, TDI, TMS, $\overline{\text{TRST}}$ and TDO—are not muxed with scan chains and are not used for any scan control.

The JTAG Controller is compatible with IEEE1149.1 Standard Test Access Port and Boundary Scan Architecture, but without the logic and instructions needed for Boundary scan.

11.1 Features

The Test and debug features of JTAG provide the following capabilities:

- Provide debug access to the ARM926 core and execute its specific JTAG instructions independently
- BIST test of i.MX21 RAMs excluding ARM926 platform BIST, BootROM, and VectorRAM BIST (reserved for internal factory testing)
- Controls tristate enable of I/O pads (reserved for internal factory testing)

11.2 Implementation

The JTAG Controller consists of the JTAG Controller state machine, Instruction Register (IR), Bypass Register, Instruction decode and various user specific data registers collectively reside inside the ExtraDebug register.

TDO output from the JTAG Controller is the muxed output based on whether i.MX21 JTAG controller or ARM926 platform JTAG mode is active. It changes on falling edge of TCK.

The TDO output enable is selected based on whether i.MX21 JTAG controller or ARM926 platform JTAG mode is active.

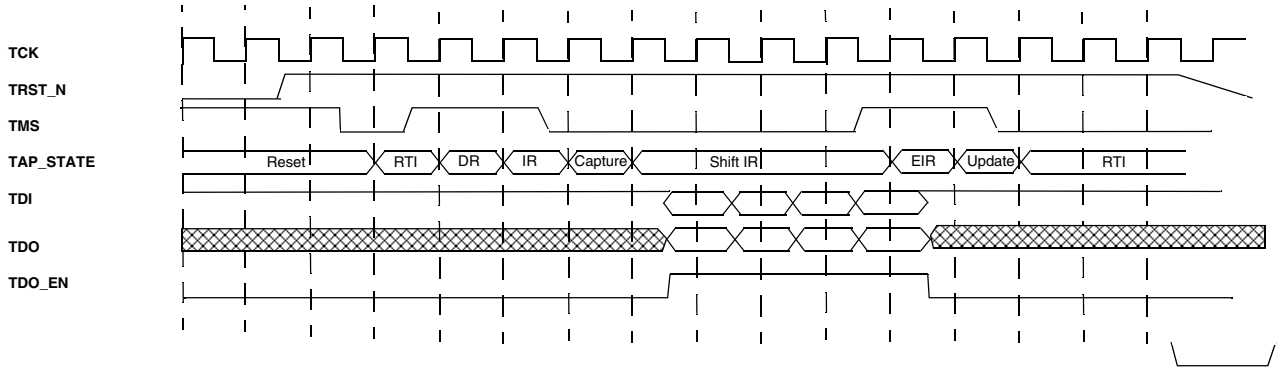


Figure 11-1. JTAG Signals Timing Diagram

- **Test Mode Select** (TMS)—Input from external pin by default connects to the ARM926 platform after gating with the laser fuse output. At the rising edge of $\overline{\text{TRST}}$, the ipp_jtag_control input controls whether the TMS pin should be connected to ARM926 platform or to i.MX21 JTAG controller. When ipp_jtag_control input is HIGH (by default), the TMS pin will be connected to ARM926 platform after gating with the laser fuse output. The TMS input of i.MX21 JTAG controller will be held HIGH in this case. When ipp_jtag_control input is LOW, the TMS pin will be connected to i.MX21 JTAG controller. The TMS input of ARM926 platform will be held HIGH here.
- **Test Reset** ($\overline{\text{TRST}}$)—Input from external pin will be connected to both ARM926 platform and i.MX21 JTAG controller.
- **The Test Data Input** (TDI)—Input from external pin will be connected to both ARM926 platform and i.MX21 JTAG controller.
- **Test Clock** (TCK)—Input from external pin will be connected to both ARM926 platform and i.MX21 JTAG controller.

11.3 JTAG Controller Pin List

The signals associated with the JTAG controller are listed in Table 11-1.

Table 11-1. JTAG Controller Pin List

Pin Name	Direction	Description
ipp_tdo	Output	Test Data Output TDO is asserted during rising edge of TCK
ipp_tck	Input	Test Clock Test Clock input is used to synchronize the Test Logic. This includes an internal pull-up resistor.
ipp_tdi	Input	Test Data Input TDI is captured during rising edge of TCK. TDI includes an internal pull-up resistor.
TMS	Input	Test Mode Select TMS is captured during rising edge of TCK. TMS includes an internal pull-up resistor. TMS input for the arm926 platform and JTAG controller is gated by the system logic.
$\overline{\text{TRST}}$	Input	Test Reset $\overline{\text{TRST}}$ includes an internal pull-up resistor

Table 11-1. JTAG Controller Pin List (continued)

Pin Name	Direction	Description
JTAG_CTRL	Input	Control whether ARM926 or JTAG has access of JTAG ports 1=ARM mode (required for debugger support) 0=JTAG mode (internal factory test mode only) This pin is captured during rising edge of $\overline{\text{TRST}}$
jtagc_tristate_en	Output	Tristate enable to all output and bidirectional pads
RTCK	Output	JTAG optional return clock for devices that support the return clock signal to help enhance stable communication. This signal is multiplexed with the One Wire signal.

11.4 JTAG Overview

The following sections provide and overview of the JTAG controller operation and the different operational modes:

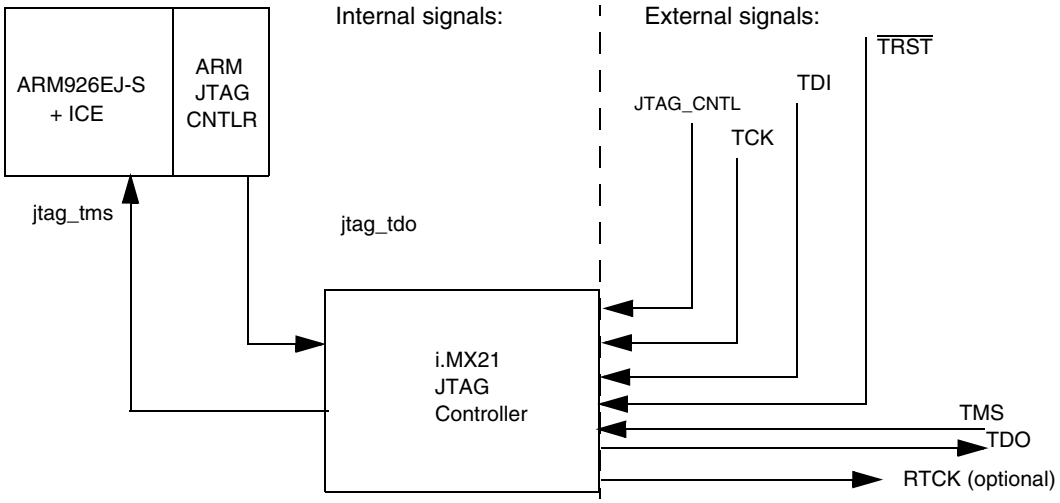


Figure 11-2. i.MX21 JTAG Block Diagram

11.5 JTAG Modes

Two JTAG modes are created based on the I/O pin ipp_jtag_control. These modes are used to maintain compatibility to ARM MCU Multi-ICE™ products as well as maintain IEEE JTAG standards.

11.5.1 ARM926 Platform JTAG Mode

This mode will connect the processed TMS input to the ARM926 platform. $\overline{\text{TRST}}$ must be asserted to leave this mode.

11.5.2 i.MX21 JTAG Controller MODE

This mode connects the processed TMS input to the i.MX21 JTAG controller. This provides a dedicated user-accessible test access port that uses the same communication style as the IEEE1149.1 Standard. $\overline{\text{TRST}}$ or $\overline{\text{POR}}$ must be asserted to leave this mode.

In this mode, i.MX21 JTAG controller supports the following capabilities:

- Query identification information (manufacturer, part number and version) of i.MX21 (IDCODE)
- Supports BIST test of all RAMs in i.MX21
- Tristate I/O pads for iddq test (ENABLE_EXTRADEBUG)
- BYPASS instruction

11.6 JTAG Instruction Register

The JTAG instruction register is 3 bits wide and the field definitions are shown in Table 11-2.

Table 11-2. JTAG Instruction Register

Bit2	Bit1	Bit0	Instruction
0	0	0	Reserved
0	0	1	Reserved
0	1	0	IDCODE
0	1	1	ENABLE_ExtraDebug
1	0	0	Reserved
1	0	1	ACCESS_GENERIC_MBIST
1	1	0	Reserved
1	1	1	BYPASS

The instruction register is reset to 3’b010 which is equivalent to the IDCODE instruction.

During the capture-IR state, the parallel inputs to the instruction register are loaded with the code 01 in the least significant bits as required by the IEEE standard, the most significant bits are loaded with the values 0, leading to a capture value of 3’b001.

11.6.1 IDCODE Instruction

Selects the ID register and the system logic controls the I/O pins. This instruction is a public instruction to allow the manufacturer, part number and the version of the IC to be available through TAP. The following table shows the ID register configuration.

Table 11-3. ID Configuration Register

31 28	27 22	21 17	16 12	11 1	0
Version Information	Customer Part Number		Manufacturer Identity		1
	Design Center Number	Device Number			
0000	01110	0111010001		00000001110	1

11.6.2 ENABLE_Extra Debug Instruction

The Extradebug register consists of 44 bits maximum comprising a 40-bits register (maximum), a 3-bit address field and one read/write bit. The register data field does not need to be filled in during register read. The particular Extradebug register connected between TDI and TDO is selected by the ExtraDebug Controller based on the currently decoded address during Update_DR state. All communication with the ExtraDebug controller is done through the Select-DR-Scan path of the i.MX21 JTAG Controller.

11.6.3 ACCESS_GENERIC_MBIST Instruction

This instruction is reserved for manufacturing test. IT allows access to i.MX21 Generic BIST Engines through the TDI/TDO during Shift_DR state.

11.6.4 BYPASS Instruction

Selects the single bit Bypass register and the system logic controls the I/O pins. This creates a shift register path from TDI to the bypass register and finally to TDO.

When the bypass register is selected by the current instruction, the shift-register stage is set to a logic zero on the rising edge of TCK in the capture-DR controller state. The first bit to be shifted out after selecting the bypass register will always be a logic zero.

11.7 Extra Debug Register

Accessed through the Select-DR-Scan path, the ExtraDebug shift register consists of 44 bits (maximum) comprising a 40 bit data field (max length, see extradebug register description), a 3 bit address field and read/write bit. The write actually takes place when the JTAG TAP controller enters the Update-DR state. On a read, the data field is ignored (the user should shift only 4 times to enter Read=1 and the address), the read will take place on the next path through DR at the Capture-DR state, the data will be shifted-out during the Shift-DR state.

On the second path for the read access (where the value of the register is shifted out), the command converter shifts in 0 so the machine will decode Write to Core Status Register which has not effect on the circuit.

11.8 TMS Sequences

11.8.1 ID Check TMS Sequence

The Table 11-4 shows the TMS sequence to check the ID Code value, starting from any point in the state machine.

Table 11-4. TMS Sequence to Check ID Code

Step	TCK	TMS	State	Comment
0	x5	1	Test Logic Reset	SEQUENCE IS: IDCODE READ
1	x1	0	Run-Test/Idle	–

Table 11-4. TMS Sequence to Check ID Code (continued)

Step	TCK	TMS	State	Comment
2	x1	1	Select DR	–
3	x1	1	Select IR	IR Path: Loading ‘Idcode’ instruction
4	x1	0	Capture IR	–
5	x1	0	Shift IR	Shift ‘Idcode’ instruction= 3'b010 through TDI
6	x2	0	Shift	–
7	x1	1	Exit1	–
8	x1	1	Update	Select ‘Idcode’ register
9	x1	0	Run-Test/Idle	–
10	x1	1	Select DR	DR Path: Reading Idcode register
11	x1	0	Capture DR	Capture Idcode value
12	x1	0	Shift DR	Shift out Idcode on 32bits
13	x31	0	Shift	–
14	x1	1	Exit1	–
15	x1	1	Update	–
16	x1	0	Run-Test/Idle	–

11.8.2 Write to ExtraDebug Register TMS Sequence

The following table shows the TMS sequence to Write to any of the ExtraDebug registers, starting from any point in the state machine.

Table 11-5. TMS Sequence to Write ExtraDebug Register

Step	TCK	TMS	State	Comment
0	x5	1	Test Logic Reset	SEQUENCE IS: WRITE ExtraDebug Register
1	x1	0	Run-Test/Idle	–
2	x1	1	Select DR	–
3	x1	1	Select IR	IR Path: Select ExtraDebug register.
4	x1	0	Capture IR	–
5	x1	0	Shift IR	Shift ‘Enable ExtraDebug’ instruction = 3'b011 through TDI
6	x2	0	Shift	–
7	x1	1	Exit1	–
8	x1	1	Update	Select ExtraDebug register
9	x1	0	Run-Test/Idle	–
10	x1	1	Select DR	DR Path: Select Extradebug register to write data

Table 11-5. TMS Sequence to Write ExtraDebug Register (continued)

Step	TCK	TMS	State	Comment
11	x1	0	Capture DR	–
12	x1	0	Shift DR	Shift In Writ bit(1'b0) + Register Address + Data
13	x43	0	Shift	–
14	x1	1	Exit1	–
15	x1	1	Update	Write to the ExtraDebug register
16	x1	0	Run-Test/Idle	–

11.8.3 TMS Sequence to Read ExtraDebug Register

The following table shows the TMS sequence to READ any of the ExtraDebug registers, starting from any any point in the state machine.

Table 11-6. TMS Sequence to Read ExtraDebug Register

Step	TCK	TMS	State	Comment
0	x5	1	Test Logic Reset	SEQUENCE IS: READ ExtraDebug Register
1	x1	0	Run-Test/Idle	–
2	x1	1	Select DR	–
3	x1	1	Select IR	IR Path: Select ExtraDebug register.
4	x1	0	Capture IR	–
5	x1	0	Shift IR	Shift 'Enable ExtraDebug' instruction = 3'b011 through TDI
6	x2	0	Shift	–
7	x1	1	Exit1	–
8	x1	1	Update	Select ExtraDebug register
9	x1	0	Run-Test/Idle	–
10	x1	1	Select DR	DR Path: Select Extradebug register to Read
11	x1	0	Capture DR	–
12	x1	0	Shift DR	Shift In Read bit(1'b1) + Register Address
13	x3	0	Shift	–
14	x1	1	Exit1	–
15	x1	1	Update	Decode the 4 bits shifted in
16	x1	0	Run-Test/Idle	–
17	x1	1	Select DR	2nd DR Path: ExtraDebug Read access
18	x1	0	Capture DR	Read the ExtraDebug register
19	x1	0	Shift DR	Shift out the captured value

Table 11-6. TMS Sequence to Read ExtraDebug Register (continued)

Step	TCK	TMS	State	Comment
20	x39	0	Shift	–
21	x1	1	Exit1	–
22	x1	1	Update	–
23	x1	0	Run-Test/Idle	–

11.9 i.MX21 JTAG Restrictions

$\overline{\text{TRST}}$ must be externally asserted to force the selection of ARM926 platform tap or i.MX21 JTAG controller. During $\overline{\text{POR}}$ assertion, arm926 platform tap is selected.

If TMS either remains unconnected or connected to VDD, then the TAP Controller cannot leave the Test-Logic-Reset state regardless of TCK.

Chapter 12
Watchdog Timer Module (WDOG)

12.1 Overview

The Watchdog Timer module (WDOG Timer) protects against system failures by providing a method of escaping from unexpected events or programming errors. Once activated, the timer must be serviced by software on a periodic basis. If servicing does not take place, the timer times out. Upon a time-out, the WDOG Timer module either asserts the $\overline{\text{WDOG}}$ interrupt signal or a system reset signal $\overline{\text{WDOG\_RESET}}$ depending on software configuration. The WDOG Timer interrupt module generates a system reset ($\overline{\text{WDOG\_RESET}}$ signal) under any of the following conditions:

- Clearing the SRS bit in the Watchdog Control Register (WCR)
- A Power On Reset has occurred ($\overline{\text{POR}}$)
- An External Reset has occurred ($\overline{\text{EXT\_RST}}$)

The $\overline{\text{WDOG}}$ signal is asserted by software writing to the WCR, or upon a watchdog time-out. The module can be activated/deactivated any number of times.

12.1.1 Timing Specifications

The Watchdog Timer provides time-out periods from 0.5 seconds up to 128 seconds with a time resolution of 0.5 seconds. It uses the 32 kHz clock as an input to pre-scalers. The pre-scalers divide the clock by a fixed value of 16384 (div4 and div4K) to achieve the resolution of 0.5 seconds and a frequency of 2 Hz (refer to Figure 12-1 on page 12-3). The output of the pre-scaler circuitry is connected to the input of a 8-bit counter to obtain a range of 0.5 to 128 seconds. The user can determine the time-out period by writing to the Watchdog Time-out field (WT[7:0]) in the Watchdog Control Register (WCR).

12.1.2 Watchdog During Reset

When a hard reset ($\overline{\text{HARD\_ASYNC\_RESET}}$) is asserted all registers except WRSR are reset to their reset values and the counter is placed in the idle state until the watchdog is enabled. The Watchdog Reset Status Register (WRSR) contains the source of the reset event and is not reset by hard reset.

12.1.3 Watchdog After Reset

The following sub-sections define the state of the Watchdog Timer after reset.

12.1.3.1 Initial Load

The Watchdog Time-out Field (WT) of the Watchdog Control Register (WCR), must have some value written to it prior to enabling the Watchdog timer by setting the Watchdog Enable (WDE) bit in the WCR. A time-out value is loaded into the counter after the service sequence is written to the Watchdog Service Register (WSR) or after the Watchdog has been enabled. The service sequence is described in Section 12.1.3.3. When the Watchdog is enabled inside debug mode by setting the WDE bit and clearing the WDBG bit, the service sequence must be run immediately after it, to reload the counter with contents of WT field of WCR register.

12.1.3.2 Countdown

The counter begins to count down from its initial programmed value after the Watchdog timer is enabled. If any system errors have occurred which prevents the software from servicing the Watchdog Service Register (WSR), the timer will time-out when the counter reaches zero. If the WSR is serviced prior to the counter reaching zero, the Watchdog will reload its counter to the time-out value indicated by bits WT[7:0] of the WCR and re-start the countdown. A reset will reset the counter and place it in the idle state at any time during the countdown.

12.1.3.3 Reload

The proper service sequence is to write a 0x5555 followed by a 0xAAAA to the WSR. In order to reload the counter, the writes must take place within the time-out value defined by bits WT[7:0] of the WCR. Any number of instructions can be executed between the two writes. This service sequence is also used to activate the counter during the initial load. See Section 12.1.3.1, “Initial Load,” on page -2.

If the WSR is not loaded with a 0x5555 prior to a write of 0xAAAA to the WSR, the counter will not be reloaded. If any value other than 0xAAAA is written to the WSR after 0x5555, the counter will not be reloaded.

12.1.3.4 Time-Out

If the counter reaches zero, the Watchdog will output either a system reset or the $\overline{\text{WDOG}}$ interrupt signal depending on the state of the WRE bit in the WCR. A 0 written to the WRE bit will configure the Watchdog to generate a system reset. A 1 will configure the Watchdog to generate the $\overline{\text{WDOG}}$ interrupt signal.

12.1.4 Low-Power and DEBUG Modes

The Watchdog is affected by the low-power and DEBUG modes. A diagram of low-power control is shown in Figure 12-1.

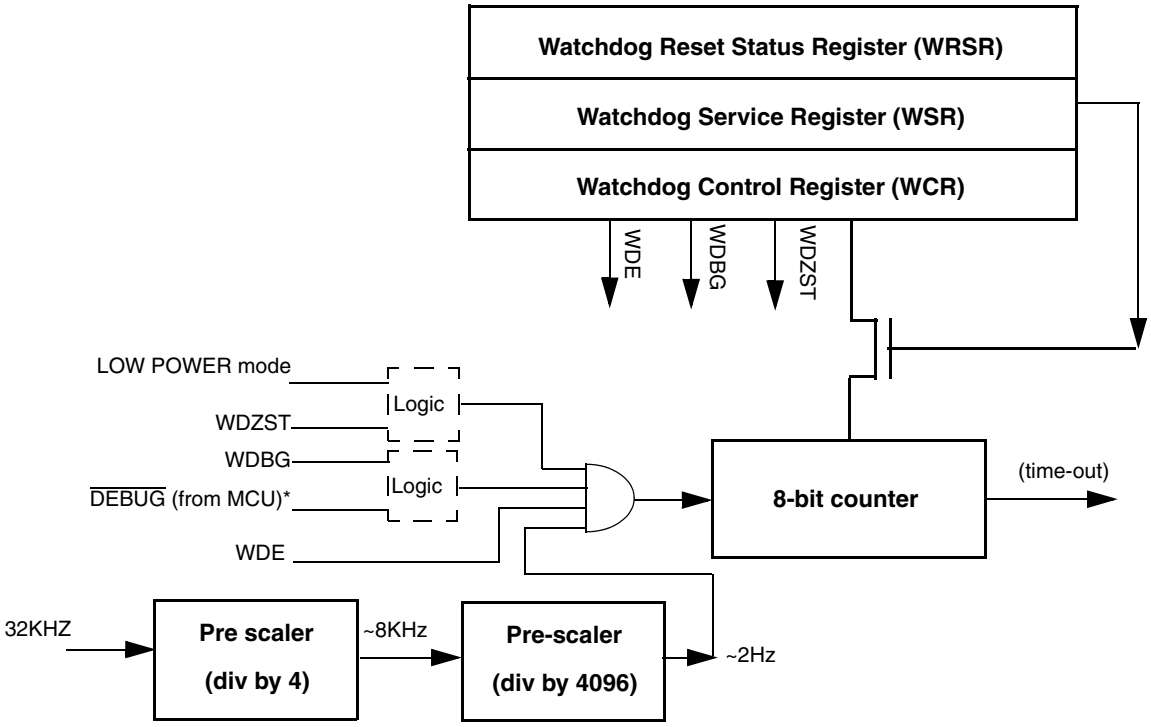
12.1.4.1 Low-Power Modes

The Watchdog Timer can be configured for continual operation or it can be suspended during low-power modes. If the Watchdog Low Power Enable (WDZST) bit in the WCR is set to ‘0”, the Watchdog continues to operate during low-power mode. If the Low Power Enable (WDZST) bit is set to 1, then the Watchdog

operation is suspended. Upon exiting low-power mode, the Watchdog operation returns to what it was prior to entering the mode.

12.1.4.2 DEBUG Mode

The Watchdog Timer can be configured for continual operation or be suspended. If the Watchdog DEBUG Enable (WDBG) bit is set to 1 in the Watchdog Control Register (WCR), then the Watchdog Timer module operation is suspended. At this point, the counter is stopped, but register read and write accesses continue to function normally. Also, while in DEBUG mode, the WDE bit can be enabled-disabled directly.



DEBUG is an active low signal from the MCU indicating DEBUG mode.

Figure 12-1. Low-Power Control

12.2 Watchdog Reset Control

12.2.1 Reset Sources

The watchdog-generated reset signal $\overline{\text{WDOG\_RESET}}$ is asserted through software writes to the Software Reset Signal ($\overline{\text{SRS}}$) bit of the WCR. The $\overline{\text{WDOG\_RESET}}$ can also be generated as a result of a WDOG time-out, a power-on-reset ($\overline{\text{POR}}$), or an external reset ($\overline{\text{EXT\_RST}}$).

$\overline{\text{WDOG\_RESET}}$ is active for 0.5 seconds for time-outs but is deasserted early if a hard reset is detected. In case of software reset, it gets asserted 3 clocks after the resetting of the SRS bit and remains asserted for 3 PER_CLOCK cycles. If a hard reset occurs before this time, it deasserts before 3 clock periods have elapsed.

For power on reset and external reset, it is asserted as long as they remain active.

The watchdog-generated reset signal WDOG_RESET is an output to the PLL Clock Controller and Reset Module which generates a HARD_ASYNC_RESET on assertion of WDOG_RESET.

12.2.2 WDOG Operation

WDOG can be asserted through software writes to the WDOG Assertion (WDA) bit of the WCR. It can also be generated as a result of a WDOG time-out.

If asserted by a software write to WDA bit, WDOG remains asserted as long as WDA bit is 0. If it is generated by a counter time-out, it is asserted for 0.5 sec. Either a hard reset (HARD_ASYNC_RESET) or Power On Reset (POR) can deassert WDOG before the 0.5 sec.

12.3 Programming Model

The Watchdog Timer has three registers in its programming model: Watchdog Control Register (WCR), Watchdog Service Register (WSR), and Watchdog Reset Status Register (WRSR). This section provides detailed descriptions of the WDOG registers.

12.3.1 Watchdog Control Register

The WCR is a 16-bit read/write (byte writable) register. It controls the Watchdog operation. All bits except for bits[5:4] are cleared during reset. Bits[5:4] are set to 1 during reset.

WCR — Watchdog Control Register — Addr 0x10002000

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WT										WDA	SRS	WRE	WDE	WDBG	WDZST
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	r	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Table 12-1. Watchdog Control Register Description

Name	Description	Settings
WT Bits 15–8	**Watchdog Time-Out Field**—This 8-bit field contains the time-out value and is loaded into the Watchdog counter after the service routine has been performed. After reset, WT must be written before enabling the Watchdog.	Set to desired time-out value.
Reserved Bits 7–6	Reserved—These bits are reserved and should always read as zero and must be written with zeros for future compatibility.	
WDA Bit 5	**Watchdog Assertion**—Controls the software assertion of the WDOG signal.	0 = Assert WDOG output 1 = No effect on system
SRS Bit 4	**Software Reset Signal**—Controls the software assertion of the WDOG-generated reset signal. This bit is automatically set to 1 after it has been asserted to 0. **Note:** This bit does not generate software reset to the module.	0 = Assert system reset signal 1 = No effect on system

Table 12-1. Watchdog Control Register Description (continued)

Name	Description	Settings
WRE Bit 3	**$\overline{\text{WDOG}}$ or $\overline{\text{WDOG\_RESET}}$ Enable**—Determines if the watchdog will generate the reset signal or $\overline{\text{WDOG}}$ upon a watchdog time-out. *Write once-only.*	0 = Generate a reset signal 1 = Generate $\overline{\text{WDOG}}$ interrupt
WDE Bit 2	**Watchdog Enable**—Enables or disables the WDOG module. Software can only write 1 in this bit. It is not possible to reset this bit by a software write, once the bit is set	0 = Disable Watchdog 1 = Enable Watchdog
WDBG Bit 1	**Watchdog DEBUG Enable**—Determines the operation of the WDOG module during DEBUG mode. *Write once-only.*	0 = Continue timer operation 1 = Suspend watchdog timer
WDZST Bit 0	**Watchdog Low Power**—Determines the operation of the WDOG module during Low power modes. *Write once-only.*	0 = Continue timer operation 1 = Suspend watchdog timer

NOTE

WDE bit is not used for clock gating in i.MX21.

12.3.2 Watchdog Service Register

When enabled, the Watchdog requires that a service sequence be written to the Watchdog Service Register (WSR) as described in Table 12-2.

WSR — Watchdog Service Register — Addr 0x10002002

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WSR															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 12-2. Watchdog Service Register Description

Name	Description	Settings
WSR Bits 15–0	**Watchdog Service Register**—This 16-bit field contains the watchdog service sequence. **Note:** Both writes must occur in the order listed prior to the time-out, but any number of instructions can be executed between the two writes.	The service sequence must be performed as follows: a. Write $5555 to the Watchdog Service Register (WSR). b. Write $AAAA to the Watchdog Service Register (WSR).

12.3.3 Watchdog Reset Status Register

The WRSR is a read-only register which records the source of the output reset assertion. It is not cleared by reset. It records the source of the output reset assertion. Therefore, one and only one bit in the WRSR will always be asserted high. And every time the register will represent the source of last reset.

RESET can be generated by the following sources as listed in priority from highest to lowest: Power-on reset, External reset, Watchdog Time-out, and Software reset.

WRSR — Watchdog Reset Status Register — Addr 0x10002004

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												PWR	EXT		TOUT	SFTW
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?

Table 12-3. Watchdog Reset Status Register Description

Name	Description	Settings
Reserved Bits 15–5	Reserved—These bits are read as 0 and should be written as 0 for future compatibility.	
PWR Bit 4	**Power-On Reset**—Indicates whether the reset was a result of a power-on reset.	0 = Reset is not a result of a power-on reset 1 = Reset is a result of a power-on reset
EXT Bit 3	**External Reset**—Indicates whether the reset was a result of an external reset.	0 = Reset is not a result of an external reset 1 = Reset is a result of an external reset
Reserved Bits 2	Reserved—These bits are read as 0 and should be written as 0 for future compatibility.	
TOUT Bit 1	**Time-out**—Indicates whether the reset was a result of WDOG time-out.	0 = Reset is not the result of WDOG time-out 1 = Reset is the result of a WDOG time-out
SFTW Bit 0	**Software Reset**—Indicates whether the reset was a result of a software reset.	0 = Reset is not a result of a software reset 1 = Reset is a result of a software reset

Note: Do not write to this register. Attempting to write to the WRSR register will generate a bus error.

Chapter 13
Real-Time Clock (RTC)

This section discusses how to operate and program the real-time clock (RTC) module that maintains the system clock, provides stopwatch, alarm, and interrupt functions, and supports the following features:

- Full clock—days, hours, minutes, seconds
- Minute countdown timer with interrupt
- Programmable daily alarm with interrupt
- Sampling timer with interrupt
- Once-per-day, once-per-hour, once-per-minute, and once-per-second interrupts
- Operation at 32.768 kHz or 32 kHz, or 38.4 kHz (determined by reference clock crystal)

As shown in the RTC block diagram (Figure 13-1), the real-time clock module consists of the following blocks:

- Prescaler
- Time-of-day (TOD) clock counter
- Alarm
- Sampling timer
- Minute stopwatch
- Associated control and bus interface hardware

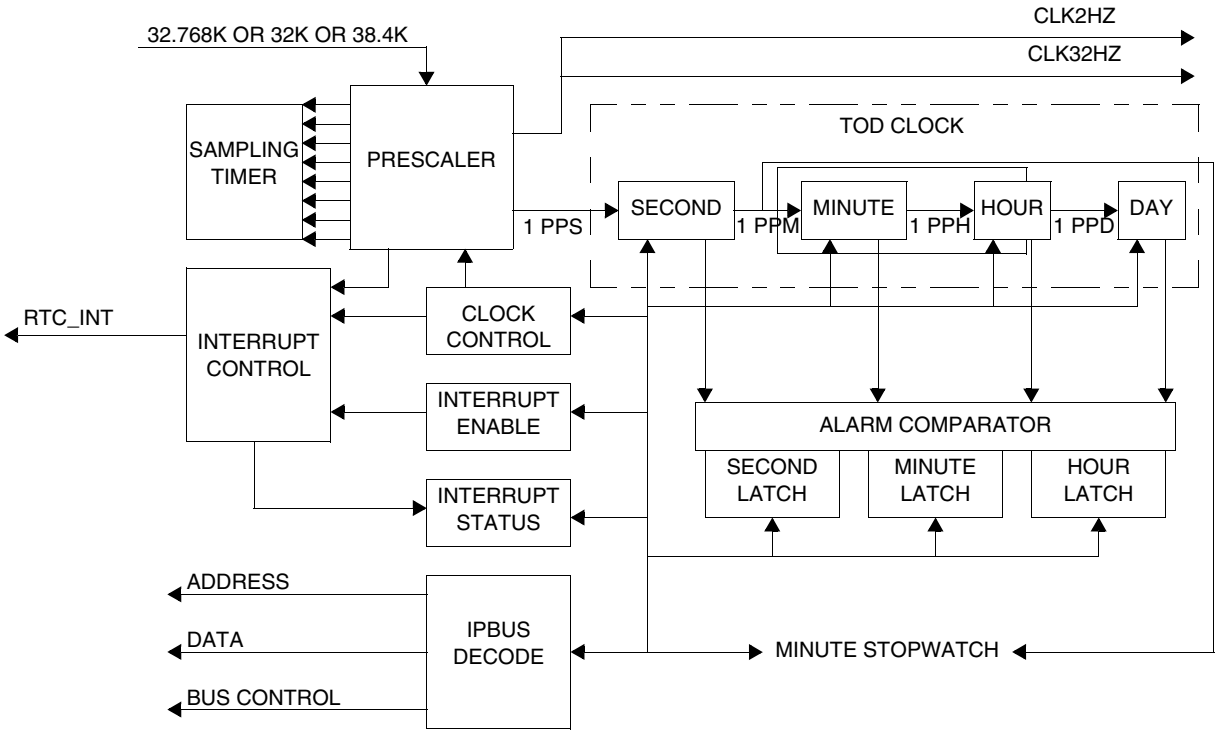


Figure 13-1. Real-Time Clock Block Diagram

13.1 Operation

The prescaler converts the incoming crystal reference clock to a 1 Hz signal which is used to increment the seconds, minutes, hours, and days TOD counters. The RTC also generates 32 Hz and 2Hz clock signal. The 32Hz clock is used by the LCDC to divide using the BD value to blink the cursor on/off and it is also is also reflected as an interrupt to AITC when SAM3 interrupt is enabled. The 2 Hz clock output can be used for flashing a LED or blinking the dots on the time display on the smartphone on/off without intervention by the ARM9 platform. The alarm functions, when enabled, generate RTC interrupts when the TOD settings reach programmed values. The sampling timer generates fixed-frequency interrupts, and the minute stopwatch allows for efficient interrupts on minute boundaries.

13.2 Prescaler and Counter

The prescaler divides the reference clock down to 1 Hz. The reference frequencies of 32.768 kHz, 38.4 kHz and 32 kHz are supported. The counter portion of the RTC module consists of four groups of counters that are physically located in three registers:

- The 6-bit seconds counter is located in the SECONDS register
- The 6-bit minutes counter and the 5-bit hours counter are located in the HOURMIN register
- The 16-bit day counter is located in the DAYR register

These counters cover a 24-hour clock over 65536 days. All three registers can be read or written at any time.

Interrupts signal when each of the four counters increments, and can be used to indicate when a counter rolls over. For example, each tick of the seconds counter causes the 1Hz interrupt flag to be set. When the seconds counter rolls from 59 to 00, the minute counter increments and the MIN interrupt flag is set. The same is true for the minute counter with the HR signal, and the hour counter with the DAY signal.

13.2.1 Alarm

There are three alarm registers that mirror the three counter registers. An alarm is set by accessing the real-time clock alarm registers (ALRM_HM, ALRM_SEC, and DAYALARM) and loading the exact time that the alarm should generate an interrupt. When the TOD clock value and the alarm value coincide, if the ALM bit in the real-time clock interrupt enable register (RTCIENR) is set, an interrupt occurs.

NOTE

If the alarm is not disabled, it will reoccur every 65536 days. If a single alarm is desired, the alarm function must be disabled using the Alarm bit (ALM) in the RTC Interrupt Enable Register (RTCIENR).

13.2.2 Sampling Timer

The sampling timer is provided to support application software. The sampling timer generates a periodic interrupt with the frequency specified by the SAMx bits of the RTCIENR register. This timer can be used for digitizer sampling, keyboard debouncing, or communication polling. The sampling timer operates only if the real-time clock is enabled. The following table lists the interrupt frequencies of the sampling timer for the possible reference clocks.

Multiple SAMx bits may be set in the RTC Interrupt Enable Register (RTCIENR). The corresponding bits in the RTC Interrupt Status Register (RTCISR) will be set at the noted frequencies.

Table 13-1. Sampling Timer Frequencies

Sampling Frequency	32.768 kHz Reference Clock	32 kHz Reference Clock	38.4 kHz Reference Clock
SAM7	512 Hz	500 Hz	600 Hz
SAM6	256 Hz	250 Hz	300 Hz
SAM5	128 Hz	125 Hz	150 Hz
SAM4	64 Hz	62.5 Hz	75 Hz
SAM3	32 Hz	31.25 Hz	37.5 Hz
SAM2	16 Hz	15.625 Hz	18.75 Hz
SAM1	8 Hz	7.8125 Hz	9.375 Hz
SAM0	4 Hz	3.90625 Hz	4.6875 Hz

13.2.3 Minute Stopwatch

The minute stopwatch performs a countdown with a one minute resolution. It can be used to generate an interrupt on a minute boundary. For example, to turn off the LCD controller after five minutes of inactivity,

program a value of 0x04 into the Stopwatch Count (CNT) field of the Stopwatch Minutes (STPWCH) register (see Table 13-12 for a complete list of settings for the STPWCH register). At each minute, the value in the stopwatch is decremented. When the stopwatch value reaches -1, the interrupt occurs. The value of the register does not change until it is reprogrammed. Note that the actual delay includes the seconds from setting the stopwatch to the next minute tick.

13.3 Programming Model

The RTC module includes ten 32-bit registers. Table 13-2 summarizes these registers and their addresses.

Table 13-2. RTC Module Register Summary

Description	Name	Address
RTC Days Counter Register	DAYR	0x10007020
RTC Hours and Minutes Counter Register	HOURMIN	0x10007000
RTC Seconds Counter Register	SECONDS	0x10007004
RTC Day Alarm Register	DAYALARM	0x10007024
RTC Hours and Minutes Alarm Register	ALRM_HM	0x10007008
RTC Seconds Alarm Register	ALRM_SEC	0x1000700C
RTC Control Register	RCCTL	0x10007010
RTC Interrupt Status Register	RTCISR	0x10007014
RTC Interrupt Enable Register	RTCIENR	0x10007018
Stopwatch Minutes Register	STPWCH	0x1000701C

13.3.1 RTC Days Counter Register

The real-time clock days counter register (DAYR) is used to program the day for the TOD clock. When the HOUR field of the HOURMIN register rolls over from 23 to 00, the day counter increments. It can be read or written at any time. After a write, the time changes to the new value. This register cannot be reset since the real-time clock is always enabled at reset.

NOTE

This day counter only supports halfword and word write operations. That means that all 16 bits must be set simultaneously.

DAYR RTC Days Counter Register **Addr 0x10007020**

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DAYS															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
	0x????															

Table 13-3. RTC Days Counter Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
DAYS Bits 15–0	**Day Setting**—Indicates the current day count.	DAYS can be set to any value between 0 and 65535.

13.3.2 RTC Hours and Minutes Counter Register

The real-time clock hours and minutes counter register (HOURMIN) is used to program the hours and minutes for the TOD clock. It can be read or written at any time. After a write, the time changes to the new value. This register cannot be reset since the real-time clock is always enabled at reset.

HOURMIN RTC Hours and Minutes Counter Register **Addr 0x10007000**

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				HOURS							MINUTES					
TYPE	r	r	r	rw	rw	rw	rw	rw	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	?	?	?	?	?	0	0	?	?	?	?	?	?
	0x????															

Table 13-4. RTC Hours and Minutes Counter Register Description

Name	Description	Settings
Reserved Bits 31–13	Reserved—These bits are reserved and should read 0.	
HOURS Bits 12–8	**Hour Setting**—Indicates the current hour.	HOURS can be set to any value between 0 and 23.
Reserved Bits 7–6	Reserved—These bits are reserved and should read 0.	
MINUTES Bits 5–0	**Minute Setting**—Indicates the current minute.	MINUTES can be set to any value between 0 and 59.

13.3.3 RTC Seconds Counter Register

The real-time clock seconds register (SECONDS) is used to program the seconds for the TOD clock. It can be read or written at any time. After a write, the time changes to the new value. This register cannot be reset since the real-time clock is always enabled at reset.

SECONDS RTC Seconds Counter Register **Addr 0x10007004**

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											SECONDS					
TYPE	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?
	0X00??															

Table 13-5. RTC Seconds Counter Register Description

Name	Description	Settings
Reserved Bits 31–6	Reserved—These bits are reserved and should read 0.	
SECONDS Bits 5–0	**Seconds Setting**—Indicates the current second.	SECONDS can be set to any value between 0 and 59.

13.3.4 RTC Day Alarm Register

The real-time clock day alarm (DAYALARM) register is used to configure the day for the alarm. The alarm settings can be read or written at any time.

DAYALARM RTC Day Alarm Register **Addr 0x10007024**

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DAYSAL															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 13-6. RTC Day Alarm Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
DAYSAL Bits 15–0	**Day Setting of the Alarm**—Indicates the current day setting of the alarm.	DAYSAL can be set to any value between 0 and 65535.

13.3.5 RTC Hours and Minutes Alarm Register

The real-time clock hours and minutes alarm (ALRM_HM) register is used to configure the hours and minutes setting for the alarm. The alarm settings can be read or written at any time.

ALRM_HM RTC Hours and Minutes Alarm Register **Addr 0x10007008**

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				HOURS							MINUTES					
TYPE	r	r	r	rw	rw	rw	rw	rw	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 13-7. RTC Hours and Minutes Alarm Register Description

Name	Description	Settings
Reserved Bits 31–13	Reserved—These bits are reserved and should read 0.	
HOURS Bits 12–8	**Hour Setting of the Alarm**—Indicates the current hour setting of the alarm.	HOURS can be set to any value between 0 and 23.
Reserved Bits 7–6	Reserved—These bits are reserved and should read 0.	
MINUTES Bits 5–0	**Minute Setting of the Alarm**—Indicates the current minute setting of the alarm.	MINUTES can be set to any value between 0 and 59.

13.3.6 RTC Seconds Alarm Register

The real-time clock seconds alarm (ALRM_SEC) register is used to configure the seconds setting for the alarm. The alarm settings can be read or written at any time.

ALRM_SEC RTC Seconds Alarm Register **Addr 0x10007000C**

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											SECONDS					
TYPE	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 13-8. RTC Seconds Alarm Register Description

Name	Description	Settings
Reserved Bits 31–6	Reserved—These bits are reserved and should read 0.	
SECONDS Bits 5–0	**Seconds Setting of the Alarm**—Indicates the current seconds setting of the alarm.	SECONDS can be set to any value between 0 and 59.

13.3.7 RTC Control Register

The real-time clock control (RTCCTL) register is used to enable the real-time clock module and specify the reference frequency information for the prescaler.

RCCTL RTC Control Register **Addr 0x10007010**

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									EN	XTL						SWR
TYPE	r	r	r	r	r	r	r	r	rw	rw	rw	r	r	r	r	rw
RESET	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	0x0080															

Table 13-9. RTC Control Register Description

Name	Description	Settings
Reserved Bits 31–8	Reserved—These bits are reserved and should read 0.	
EN Bit 7	**Enable**—Enables/Disables the real-time clock. The software reset bit (SWR) has no effect on this bit.	0 = Disable the real-time clock 1 = Enable the real-time clock
XTL Bits 6–5	**Crystal Selection**—Selects the proper input crystal frequency. It is important to set these bits correctly or the real-time clock will be inaccurate.	00 = 32.768 kHZ 01 = 32 kHz 10 = 38.4 kHz 11 = 32.768 kHz
Reserved Bits 4–1	Reserved—These bits are reserved and should read 0.	
SWR Bit 0	**Software Reset**—Resets the module to its default state. However, a software reset will have no effect on the clock enable (EN) bit.	0 = No effect. 1 = Reset the module to its default state.

13.3.8 RTC Interrupt Status Register

The real-time clock interrupt status register (RTCISR) indicates the status of the various real-time clock interrupts. When an event of the types included in this register occurs, then the bit will be set in this register, regardless of whether the interrupt is enabled. These bits are cleared by writing a value of 1, which also clears the interrupt. Interrupts may occur while the system clock is idle or in sleep mode. For more information about the frequency of the sampling timer interrupts (SAM7–SAM0), refer to Table 13-1.

RTCISR RTC Interrupt Status Register **Addr 0x10007014**

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAM7	SAM6	SAM5	SAM4	SAM3	SAM2	SAM1	SAM0	2HZ		HR	1HZ	DAY	ALM	MIN	SW
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 13-10. RTC Interrupt Status Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
SAM7 Bit 15	**Sampling Timer Interrupt Flag at SAM7 Frequency**—Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 512, 600, or 500 Hz. The actual rate of the interrupt depends on the input clock value. See Table 13-1 on page 13-3.	0 = No SAM7 interrupt occurred. 1 = A SAM7 interrupt occurred.
SAM6 Bit 14	**Sampling Timer Interrupt Flag at SAM6 Frequency**—Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 256, 250, or 300 Hz. The actual rate of the interrupt depends on the input clock value. See Table 13-1 on page 13-3.	0 = No SAM6 interrupt occurred. 1 = A SAM6 interrupt occurred.
SAM5 Bit 13	**Sampling Timer Interrupt Flag at SAM5 Frequency**—Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 128, 125, or 150 Hz. The actual rate of the interrupt depends on the input clock value. See Table 13-1 on page 13-3.	0 = No SAM5 interrupt occurred. 1 = A SAM5 interrupt occurred.
SAM4 Bit 12	**Sampling Timer Interrupt Flag at SAM4 Frequency**—Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 64, 62.5, or 75 Hz. The actual rate of the interrupt depends on the input clock value. See Table 13-1 on page 13-3.	0 = No SAM4 interrupt occurred. 1 = A SAM4 interrupt occurred.
SAM3 Bit 11	**Sampling Timer Interrupt Flag at SAM3 Frequency**—Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 32, 31.25, or 37.5 Hz. The actual rate of the interrupt depends on the input clock value. See Table 13-1 on page 13-3.	0 = No SAM3 interrupt occurred. 1 = A SAM3 interrupt occurred.
SAM2 Bit 10	**Sampling Timer Interrupt Flag at SAM2 Frequency**—Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 16, 15.625, or 18.75 Hz. The actual rate of the interrupt depends on the input clock value. See Table 13-1 on page 13-3.	0 = No SAM2 interrupt occurred. 1 = A SAM2 interrupt occurred.

Table 13-10. RTC Interrupt Status Register Description (continued)

Name	Description	Settings
SAM1 Bit 9	**Sampling Timer Interrupt Flag at SAM1 Frequency**—Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 8, 7.8125, or 9.375 Hz. The actual rate of the interrupt depends on the input clock value. See Table 13-1 on page 13-3.	0 = No SAM1 interrupt occurred. 1 = A SAM1 interrupt occurred.
SAM0 Bit 8	**Sampling Timer Interrupt Flag at SAM0 Frequency**—Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 4, 3.90625, or 4.6875 Hz. The actual rate of the interrupt depends on the input clock value. See Table 13-1 on page 13-3.	0 = No SAM0 interrupt occurred. 1 = A SAM0 interrupt occurred.
2HZ Bit 7	**2 Hz Flag**—Indicates that an interrupt has occurred. If enabled, this bit is set at every 2 Hz frequency.	0 = No 2 Hz interrupt occurred. 1 = A 2 Hz interrupt has occurred.
Reserved Bit 6	Reserved—This bit is reserved and should read 0.	
HR Bit 5	**Hour Flag**—Indicates that the hour counter has incremented. If enabled, this bit is set on every increment of the hour counter in the time-of-day clock.	0 = No 1-hour interrupt occurred. 1 = A 1-hour interrupt has occurred
1HZ Bit 4	**1 Hz Flag**—Indicates that the second counter has incremented. If enabled, this bit is set on every increment of the second counter of the time-of-day clock.	0 = No 1 Hz interrupt occurred. 1 = A 1 Hz interrupt has occurred.
DAY Bit 3	**Day Flag**—Indicates that the day counter has incremented. If enabled, this bit is set on every increment of the day counter of the time-of-day clock.	0 = No 24-hour rollover interrupt occurred. 1 = A 24-hour rollover interrupt has occurred.
ALM Bit 2	**Alarm Flag**—Indicates that the real-time clock matches the value in the alarm registers. Note that the alarm will reoccur every 512 days. For a single alarm, clear the interrupt enable for this bit in the interrupt service routine.	0 = No alarm interrupt occurred. 1 = An alarm interrupt has occurred.
MIN Bit 1	**Minute Flag**—Indicates that the minute counter has incremented. If enabled, this bit is set on every increment of the minute counter in the time-of-day clock.	0 = No 1-minute interrupt occurred. 1 = A 1-minute interrupt has occurred.
SW Bit 0	**Stopwatch Flag**—Indicates that the stopwatch countdown timed out.	0 = The stopwatch did not time out. 1 = The stopwatch timed out.

13.3.9 RTC Interrupt Enable Register

The real-time clock interrupt enable register (RTCIENR) is used to enable/disable the various real-time clock interrupts. When an event of the types included in this register occurs, the corresponding bit gets set in the RTC Interrupt Status Register (RTCISR). An interrupt will occur if the corresponding bit is enabled in this register. For more information about the frequency of the sampling timer interrupts (SAM7–SAM0), refer to Table 13-1 on page 13-3.

RTCIENR RTC Interrupt Enable Register **Addr 0x10007018**

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAM7	SAM6	SAM5	SAM4	SAM3	SAM2	SAM1	SAM0	2HZ		HR	1HZ	DAY	ALM	MIN	SW
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 13-11. RTC Interrupt Enable Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
SAM7 Bit 15	**Sampling Timer Interrupt Flag at SAM7 Interrupt Enable**—Enables/Disables the real-time sampling timer interrupt 7. The frequency of this interrupt is shown in Table 13-1 on page 13-3.	0 = The SAM7 interrupt is disabled. 1 = The SAM7 interrupt is enabled.
SAM6 Bit 14	**Sampling Timer Interrupt Flag at SAM6 Interrupt Enable**—Enables/Disables the real-time sampling timer interrupt 6. The frequency of this interrupt is shown in Table 13-1 on page 13-3.	0 = The SAM6 interrupt is disabled. 1 = The SAM6 interrupt is enabled.
SAM5 Bit 13	**Sampling Timer Interrupt Flag at SAM5 Interrupt Enable**—Enables/Disables the real-time sampling timer interrupt 5. The frequency of this interrupt is shown in Table 13-1 on page 13-3.	0 = The SAM5 interrupt is disabled. 1 = The SAM5 interrupt is enabled.
SAM4 Bit 12	**Sampling Timer Interrupt Flag at SAM4 Interrupt Enable**—Enables/Disables the real-time sampling timer interrupt 4. The frequency of this interrupt is shown in Table 13-1 on page 13-3.	0 = The SAM4 interrupt is disabled. 1 = The SAM4 interrupt is enabled.
SAM3 Bit 11	**Sampling Timer Interrupt Flag at SAM3 Interrupt Enable**—Enables/Disables the real-time sampling timer interrupt 3. The frequency of this interrupt is shown in Table 13-1 on page 13-3.	0 = The SAM3 interrupt is disabled. 1 = The SAM3 interrupt is enabled.
SAM2 Bit 10	**Sampling Timer Interrupt Flag at SAM2 Interrupt Enable**—Enables/Disables the real-time sampling timer interrupt 2. The frequency of this interrupt is shown in Table 13-1 on page 13-3.	0 = The SAM2 interrupt is disabled. 1 = The SAM2 interrupt is enabled.
SAM1 Bit 9	**Sampling Timer Interrupt Flag at SAM1 Interrupt Enable**—Enables/Disables the real-time sampling timer interrupt 1. The frequency of this interrupt is shown in Table 13-1 on page 13-3.	0 = The SAM1 interrupt is disabled. 1 = The SAM1 interrupt is enabled.
SAM0 Bit 8	**Sampling Timer Interrupt Flag at SAM0 Interrupt Enable**—Enables/Disables the real-time sampling timer interrupt 0. The frequency of this interrupt is shown in Table 13-1 on page 13-3.	0 = The SAM0 interrupt is disabled. 1 = The SAM0 interrupt is enabled.
2HZ Bit 7	**2 Hz Interrupt Enable**—Enables/Disables an interrupt at a 2 Hz rate.	0 = The 2 Hz interrupt is disabled. 1 = The 2 Hz interrupt is enabled.
Reserved Bit 6	Reserved—This bit is reserved and should read 0.	

Table 13-11. RTC Interrupt Enable Register Description (continued)

Name	Description	Settings
HR Bit 5	**Hour Interrupt Enable**—Enables/Disables an interrupt whenever the hour counter of the real-time clock increments.	0 = The 1-hour interrupt id disabled. 1 = The 1-hour interrupt is enabled.
1HZ Bit 4	**1 Hz Interrupt Enable**—Enables/Disables an interrupt whenever the second counter of the real-time clock increments.	0 = The 1 Hz interrupt is disabled. 1 = The 1 Hz interrupt is enabled.
DAY Bit 3	**Day Interrupt Enable**—Enables/Disables an interrupt whenever the hours counter rolls over from 23 to 0 (midnight rollover).	0 = The 24-hour interrupt is disabled. 1 = The 24-hour interrupt is enabled.
ALM Bit 2	**Alarm Interrupt Enable**—Enables/Disables the alarm interrupt.	0 = The alarm interrupt is disabled. 1 = The alarm interrupt is enabled.
MIN Bit 1	**Minute Interrupt Enable**—Enables/Disables an interrupt whenever the minute counter of the real-time clock increments.	0 = The 1-minute interrupt is disabled. 1 = The 1-minute interrupt is enabled.
SW Bit 0	**Stopwatch Interrupt Enable**—Enables/Disables the stopwatch interrupt. **Note:** The stopwatch counts down and remains at decimal -1 until it is reprogrammed. If this bit is enabled with -1 (decimal) in the STPWCH register, an interrupt will be posted on the next minute tick.	0 = Stopwatch interrupt is disabled. 1 = Stopwatch interrupt is enabled.

13.3.10 Stopwatch Minutes Register

The stopwatch minutes (STPWCH) register contains the current stopwatch countdown value. When the minute counter of the TOD clock increments, the value in this register decrements.

STPWCH Stopwatch Minutes Register **Addr 0x1000701C**

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											CNT					
TYPE	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
	0x003F															

Table 13-12. Stopwatch Minutes Register Description

Name	Description	Settings
Reserved Bits 31–6	Reserved—These bits are reserved and should read 0.	
CNT Bits 5–0	**Stopwatch Count**—Contains the stopwatch countdown value. **Note:** The stopwatch counter is decremented by the minute (MIN) tick output from the real-time clock, so the average tolerance of the count is 0.5 minutes. For better accuracy, enable the stopwatch by polling the MIN bit of the RTCISR register or by polling the minute interrupt service routine.	These bits can be set to any value between 0 and 62. Once the countdown has completed, the value will not change until a nonzero value (1–62) is written.

Chapter 14
General-Purpose Timers (GPT)

The GPT module of the i.MX21 contains three identical general-purpose 32-bit timers (GPT) with programmable prescalers and compare and capture registers. Each timer's counter value can be captured using an external event and can be configured to trigger a capture event on either the leading or trailing edges of an input pulse. Each GPT can also generate an interrupt when the timer reaches a programmed value. Each GPT has an11-bit prescaler providing a programmable clock frequency derived from the IPG_CLK_PERCLK1 (or simply PERCLK1 from the PLL, Clock, and Reset Controller chapter). Figure 14-1 illustrates the general-purpose timer block diagram for one of the timers.

The General-Purpose Timers have the following features:

- Programmable sources for the clock input, including external clock
- Input capture capability with programmable trigger edge
- Output compare with programmable mode
- Free-run and restart modes
- Software reset function

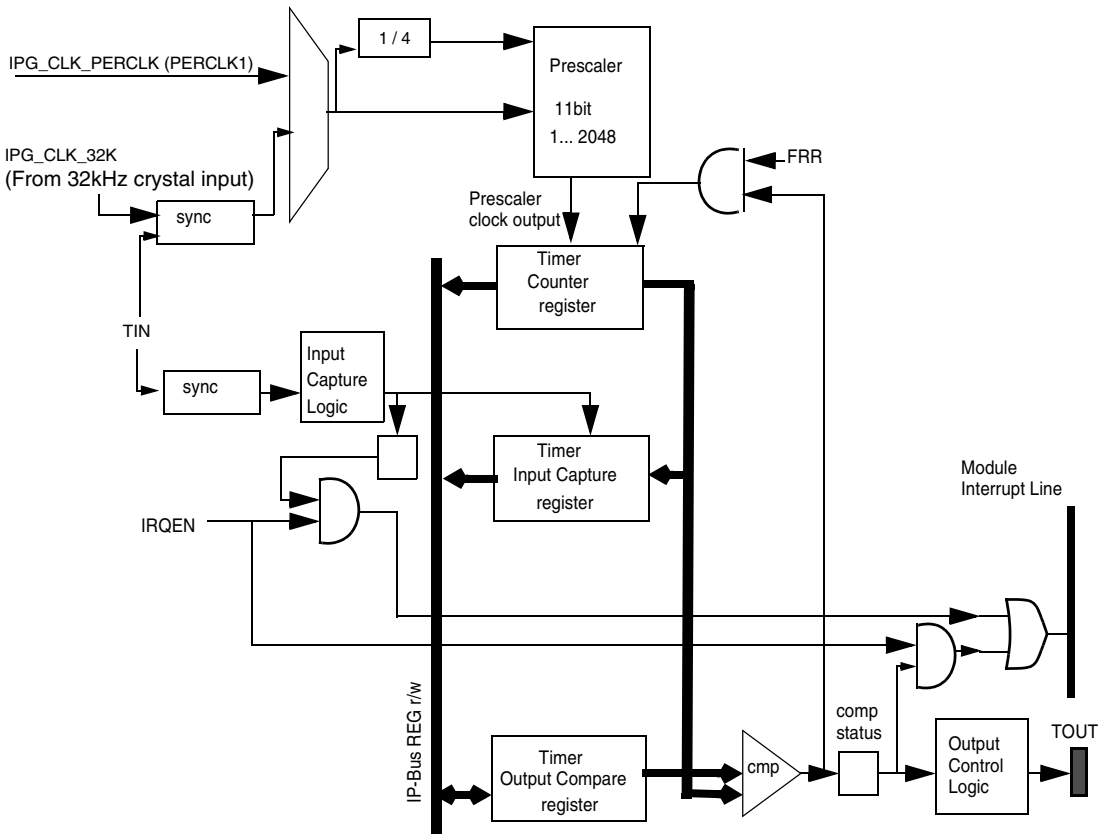


Figure 14-1. General-Purpose Timer Block Diagram

14.1 Operation

After a hardware reset the Counter, Control, Prescaler, Status and Capture registers of the GPT are reset. The Compare value is set to 0xFFFFFFFF. The output pin (TOUT) is also reset. The GPT is enabled when the TEN bit in the GPT Control Register (TCTL) is set and the counter starts runnning. It is recommended that all the registers be set to appropiate values first before enabling the GPT. When the TEN bit is cleared the counter value freezes or clears depending on CC bit in GPT Control Register (TCTL). All other register values are retained.

There is a Software Reset bit (SWR) in the TCTL register. When this bit is set the GPT generates a reset signal. The software reset results in all the registers in the GPT being reset except for the TEN bit which is not cleared by software reset if set. The software reset can be asserted with the TEN bit off, however, the IPG_CLK clock (module clock) signals to GPT must be on for the software reset to be functional.

14.1.1 Clocks

The clock that feeds the prescaler can be selected from the

- IPG_CLK_PERCLK1 (divided by 1 or by 4). The frequency of IPG_CLK_PERCLK1 is constant and is independent of changes to the IPG_CLK. The module internally synchronizes PERCLK1 to the IPG_CLK. This clock is synchronized inside the module to IPG_CLK. Hence the frequency of this clock has to be at least 1/4 that of IPG_CLK if the prescaler is programmed to divide by 1.

The clock input source is determined by the CLKSOURCE field of the GPT control register. The CLKSOURCE value should only be changed when the GPT is disabled. The GPT prescaler register (TPRER) selects the divide ratio of the input clock that drives the main counter (TCN). The prescaler can divide the input clock by a value between 1 and 2048.

14.1.2 Operation During Low-Power Mode

In low-power mode when the IPG_CLK (module clock used for register accesses) and IPG_CLK_PERCLK clocks are switched off, the GPT halts and the counter (TEN) freezes at its current value. Since all clocks to the counter are synchronized with IPG_CLK the counter stops counting as and when the IPG_CLK is turned off. Thus when IPG_CLK is switched off this clock will be off and even though 32KHz, IPG_CLK_PERCLK1 (PERCLK1) or IPP_GPT_TIN (Timer input from TIN pin) clock is available at the module input, the counter freezes and all GPT operations are halted.

14.1.3 Capture Event

Each of the three GP Timers have a 32-bit capture register that takes a snapshot of the counter when a defined transition of TIN is detected by the capture edge detector. The type of transition that triggers this capture is selected by the CAP field of the GPT Control Register (TCTL). Each transition must be valid for at least two IPG_CLK periods to ensure a capture event is triggered.

When a capture event occurs, the corresponding status bit is set in the GPT status register (TSTAT) and an interrupt is posted if the capture function is enabled and if the CAPTEN bit of the GPT control register is set. If another capture event occurs the new count value will still be captured in the capture register even if the interrupt is not serviced and capture status bit is set.

14.1.4 Compare Event

Each GPT has a 32-bit compare register. When the value in this register matches with the value of the counter register a compare event occurs. On a compare event the appropriate GPT output pin (TOUT) is toggled or an active low pulse (for one count period) is generated on it according to the setting of Output Mode (OM) bit in the GPT control register. When in toggle mode the toggling takes place at the end of the count period in which the match has occurred.

- The Timer output of GPT1, TOUT1, is available at TOUT pin.
- The Timer output of GPT2, TOUT2, is available at the Bin output at the PWM pin.
- The Timer output of GPT3, TOUT3, is available at the Cin output at the PWM pin.

NOTE

TOUT2 and TOUT3 are muxed via the IOMUX, see Chapter 47, GPIO,

and made available at the PWM pin as alternate functions.

The corresponding status bit is set in the status register and an interrupt is posted if the COMPEN bit of the GPT control register is set. The GPT output pin continues to produce an output on a compare event even if the interrupt is not serviced and compare status bit is set.

14.1.5 Modes of Operation

The GPT can be configured for free-run or restart modes by programming the Free-run / Restart bit (FRR) of the GPT control register.

- **Restart mode**: In restart mode, when a compare event occurs, the counter resets to 0x00000000. Subsequently it resumes counting up.
- **Freerun mode**: In free-run mode, when a compare event occurs, it has no effect on the counter value. The counter continues counting until 0xFFFFFFFF is reached and then it is reset to 0x00000000 and resumes counting.

14.2 Programming Model

The General-Purpose Timer modules each have six user-accessible 32-bit registers. They are shown with offsets from their respective base addresses in the detailed register descriptions.

Table 14-1 summarizes the general-purpose timer registers and their addresses. Table 14-2 provides the detailed register summary.

Table 14-1. GP Timer Register Summary

Description	Name	Address
GPT Control Register 1	TCTL1	0x10003000
GPT Control Register 2	TCTL2	0x10004000
GPT Control Register 3	TCTL3	0x10005000
GPT Prescaler Register 1	TPRER1	0x10003004
GPT Prescaler Register 2	TPRER2	0x10004004
GPT Prescaler Register 3	TPRER3	0x10005004
GPT Compare Register 1	TCMP1	0x10003008
GPT Compare Register 2	TCMP2	0x10004008
GPT Compare Register 3	TCMP3	0x10005008
GPT Capture Register 1	TCR1	0x1000300C
GPT Capture Register 2	TCR2	0x1000400C
GPT Capture Register 3	TCR3	0x1000500C
GPT Counter Register 1	TCN1	0x10003010
GPT Counter Register 2	TCN2	0x10004010
GPT Counter Register 3	TCN3	0x10005010
GPT Status Register 1	TSTAT1	0x10003014
GPT Status Register 2	TSTAT2	0x10004014
GPT Status Register 3	TSTAT3	0x10005014

Name		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
TCTL1 (0x10003000) TCTL2 (0x10004000) TCTL3 (0x10005000)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	0	0	0	0	0	CC	OM	FRR	CAP		CAPT EN	COMP EN	CLK SOURCE			TEN
	W	SWR															
TPRER1 (0x10003004) TPRER2 (0x10004004) TPRER3 (0x10005004)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	0	0	0	0	0	PRESCALER										
	W																
TCMP1 (0x10003008) TCMP2 (0x10004008) TCMP3 (0x10005008)	R	COMPARE VALUE															
	W																
	R	COMPARE VALUE															
	W																
TCR1 (0x1000300c) TCR2 (0x1000400c) TCR3 (0x1000500c)	R	CAPTURE VALUE															
	W																
	R	CAPTURE VALUE															
	W																
TCN1 (0x10003010) TCN2 (0x10004010) TCN3 (0x10005010)	R	COUNTER VALUE															
	W																
	R	COUNTER VALUE															
	W																
TSTAT1 (0x10003014) TSTAT2 (0x10004014) TSTAT3 (0x10005014)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CAPT	COMP
	W															*W1C*	*W1C*

14.2.1 GPT Control Register

The GPT control (TCTL) register controls the overall operation of the timer.

TCTL1 GPT Control Register 1 0x10003000
TCTL2 GPT Control Register 2 0x10004000
TCTL3 GPT Control Register 3 0x10005000

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SWR					CC	OM	FRR	CAP		CAPT EN	COMP EN	CLK SOURCE			TEN
TYPE	slfclr	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 14-2. GPT Control Registers 1, 2, and 3 Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should not be used.	
SWR Bit 15	**SOFTWARE RESET**—GPT is reset when this bit is set to 1. This bit is set when the module is in reset state and is cleared when the reset procedure is over. The reset signal is asserted 2 clock cycles (IPG_CLK) after this bit is set and the reset remains asserted for 3 clock cycles. The whole reset procedure is complete 5 clock cycles after this bit is asserted This software reset does not reset the TEN bit.	0 = GPT is not reset 1 = GPT is reset
Reserved Bits 14–11	Reserved—These bits are reserved and should not be used.	
CC Bit 10	**Counter Clear**—This bit determines whether the counter is to be cleared when TEN=0 (timer disabled).	0= counter is halted at the current count when TEN = 0. 1 = counter will be reset when TEN=0.
OM Bit 9	**Output Mode**—This bit controls the output mode of the timer after compare event occurs.	0 = Active-low pulse for one clock period. 1 = Toggle output.
FRR Bit 8	**Free-Run / Restart**—This bit controls how the timer operates after a compare event occurs. In free-run mode, the timer continues counting till 0xffffffff. In restart mode, the counter resets to 0x00000000 and resumes counting.	0 = Restart mode 1 = Free-run mode

Table 14-2. GPT Control Registers 1, 2, and 3 Description

Name	Description	Settings
CAP Bits 7–6	**Capture Edge**—This field controls the operation of the capture function. The value in the counter is loaded into the Capture register on the detection of an event on the TIN pin. The event which will trigger this capture is determined by this field.	00 = Capture function disabled 01 = Capture on rising edge and generate interrupt 10 = Capture on falling edge and generate interrupt 11 = Capture on rising or falling edge and generate interrupt
CAPT EN Bit 5	**Capture Interrupt Enable**—This bit enables the capture interrupt.	0 = Capture interrupt disabled. 1 = Capture interrupt enabled.
COMP EN Bit 4	**Compare Interrupt Enable**—This bit enables the compares interrupt.	0 = Compare interrupt disabled. 1 = Compare interrupt enabled.
CLKSOURCE Bits 3–1	**Clock Source**—This field controls the source of the clock to the prescaler. The stop-count freezes the timer at its current value. **Note:** This field value should only be changed when the GPT is disabled.	000 = Stop count (clock disabled). 001 = PERCLK1 to prescaler. 010 = PERCLK1 divided by 4 to prescaler. 011 = TIN to prescaler. 1xx = 32kHz clock to prescaler.
TEN Bit 0	**Timer Enable**—TEN bit enables the general-purpose timer. The bit can be cleared either by writing 0 or by a hardware reset. The bit cannot be cleared by asserting the software reset.	0 = Timer is disabled 1 = Timer is enabled.

14.2.2 GPT Prescaler Register

The GPT prescaler register (TPRER) controls the divide value of the prescaler.

TPRER1	GPT Prescaler Register 1	0x10003004
TPRER2	GPT Prescaler Register 2	0x10004004
TPRER3	GPT Prescaler Register 3	0x10005004

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						PRESCALER										
TYPE	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 14-3. GPT Prescaler Register 1, 2, and 3 Description

Name	Description	Settings
Reserved Bits 31–11	Reserved—These bits are reserved and should read 0.	
PRESCALER Bits 10–0	**Counter Clock Prescaler**—This field determines the division value of the prescaler between 1 and 2048. 0x00 divides by 1 and 0x7FF divides by 2048.	0x00 = Divide by 1 ... 0x7ff = Divide by 2048

14.2.3 GPT Compare Register

The GPT compare (TCMP) register contains the value that is compared with the free-running counter. A compare event is generated when the counter matches the value in this register. This register reset to 0xFFFFFFFF.

TCMP1 GPT Compare Register 1 0x10003008
TCMP2 GPT Compare Register 2 0x10004008
TCMP3 GPT Compare Register 3 0x10005008

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	COMPARE VALUE															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COMPARE VALUE															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 14-4. GPT Compare Registers 1, 2, and 3 Description

Name	Description
COMPARE Bits 31–0	**Compare Value**—A compare event occurs when the counter value matches the value in this field.

14.2.4 GPT Capture Register

This register is read only, and resets to 0x00000000. The GPT capture register (TCR) stores the counter value when a capture event occurs.

TCR1	GPT Capture Register 1	0x1000300C
TCR2	GPT Capture Register 2	0x1000400C
TCR3	GPT Capture Register 3	0x1000500C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CAPTURE VALUE															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CAPTURE VALUE															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 14-5. GPT Capture Registers 1, 2, and 3 Description

Name	Description
CAPTURE Bits 31–0	**Capture Value**—This field stores the counter value at the time of a capture event.

14.2.5 GPT Counter Register

The read-only GPT counter (TCN) register can be read at anytime without disturbing the current count.

TCN1	GPT Counter Register 1	0x10003010
TCN2	GPT Counter Register 2	0x10004010
TCN3	GPT Counter Register 3	0x10005010

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	COUNTER VALUE															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COUNTER VALUE															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 14-6. GPT Counter Registers 1, 2, and 3 Description

Name	Description
COUNT Bits 31–0	**Count Value**—This field contains the current count value. Whenever there is an update of compare register the counter is reset to zero and the count starts afresh.

14.2.6 GPT Status Register

The GPT status (TSTAT) register (write 1 to clear) indicates the GPT's status. When a capture event occurs, the CAPT bit is set. When a compare event occurs, the COMP bit is set. These bits can only be cleared by writing 1 to clear the interrupt status.

TSTAT1	GPT Status Register 1	0x10003014
TSTAT2	GPT Status Register 2	0x10004014
TSTAT3	GPT Status Register 3	0x10005014

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															CAPT	COMP
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	W1C	W1C
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 14-7. GPT Status Registers 1, 2, and 3 Description

Name	Description	Settings
Reserved Bits 31–2	Reserved—These bits are reserved and should read 0.	
CAPT Bit 1	**Capture Event**—This bit indicates that a capture event has occurred.	0 = No capture event 1 = A capture event has occurred
COMP Bit 0	**Compare Event**—This bit indicates that a compare event has occurred.	0 = No compare event 1 = A compare event has occurred

Chapter 15
General-Purpose I/O (GPIO)

The GPIO module in i.MX21 provides six general purpose I/O (GPIO) ports (PA, PB, PC, PD, PE and PF). Each single GPIO port is a 32-bit port that may be multiplexed with one or more dedicated functions.

This chapter contains the description of the top level I/O multiplexing strategy in the i.MX21 which consists of two parts:

- Software controllable multiplexing done in the GPIO module
- Hardware multiplexing by the IOMUX module

The I/O multiplexing strategy is designed to configure the inputs and outputs of the i.MX21 chip in different modes. It allows a user to use the same I/O pad for alternative purposes of the chip. The design of I/O multiplexer is targeted to be as flexible as possible. Please refer to chapter 2, “Signal Descriptions and Pin Assignments” for detailed I/O multiplexing information.

Figure 15-1 shows the block diagram of the GPIO and IOMUX modules partition at the top level of i.MX21. Make note that the signals A_IN, B_IN, C_IN, A_OUT and B_OUT are internal signals and do not represent individual port signals. Figure 15-2 on page 15-2 shows a block diagram of an individual port of the GPIO module.

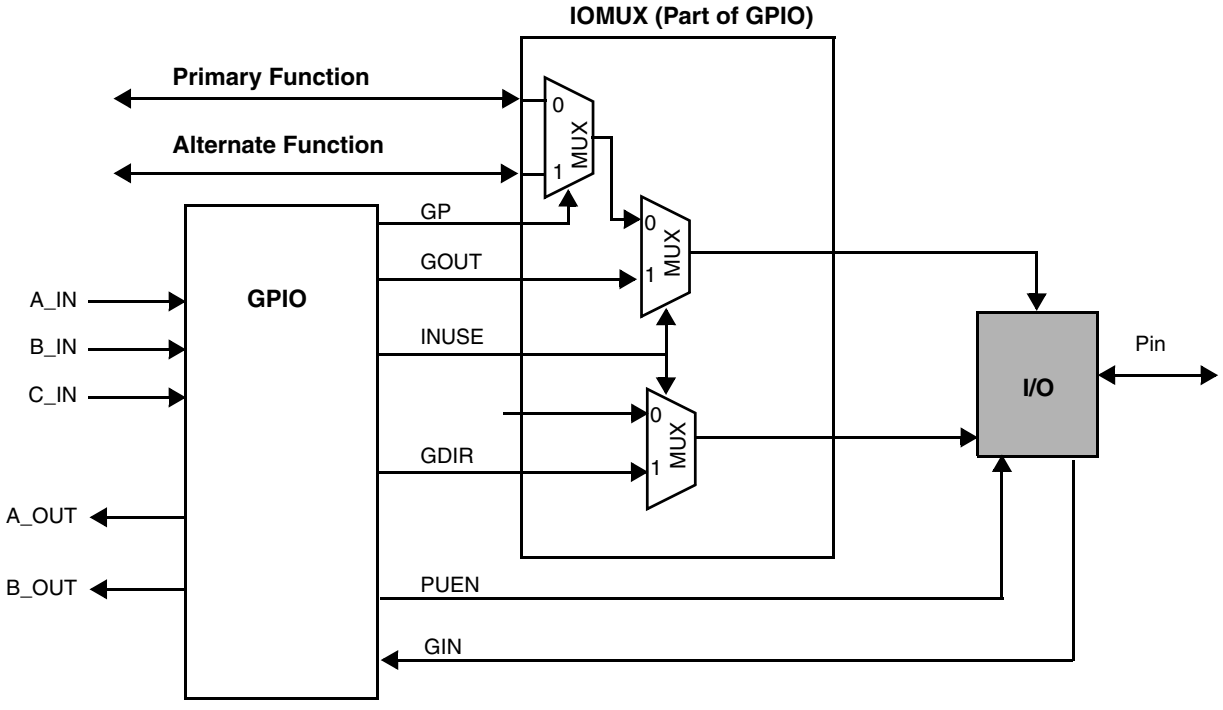


Figure 15-1. Functional Block Diagram of GPIO

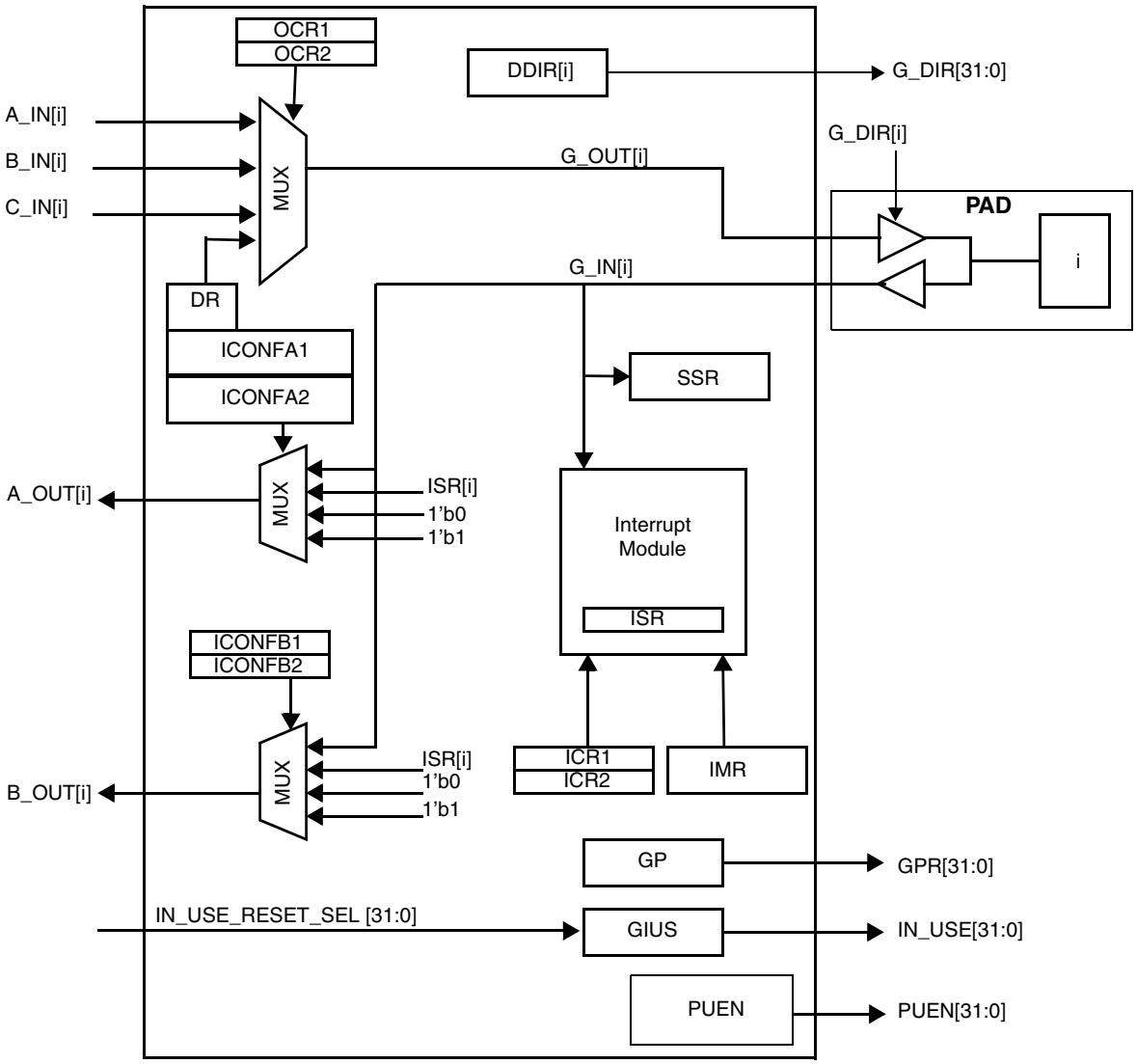


Figure 15-2. GPIO Block Diagram for an Individual Port

15.1 Overview

The GPIO module provides General Purpose I/O capability to the device. Each I/O port can be programed as either a general purpose input or general purpose output. Besides GPIO functionality, pins can be changed from their default dedicated functions to alternate functions. Input and output signals of peripherals are connected to the IOMUX module at the dedicated (primary) or alternate inputs. In the output direction, one out of three alternate sources (originating from peripherals) can be selected. From the input direction, one out of two alternate destinations (input to a peripheral) can be selected.

15.2 GPIO Features

The following list contains the GPIO features.

- Six 32-bit ports. Each with direction-configurable pins.

- Software control for input/output pin configuration through 32-bit direction register.
- Software control for multiplexing one out of four different sources for every output. Three of them are functional pins from internal modules while the fourth is from the data register of the module
- Software control for routing of every input to two different destinations.
- Input data can be sampled to the data register.
- Inputs can be internally tied to a logic 1 or 0 to ensure any transitions attempted to be processed are ignored.
- One 32-bit general purpose register is dedicated to each GPIO port. These registers may be used for software control of IOMUX block of the GPIO.
- Every input is configurable as an interrupt and each interrupt can be defined as either:
  - rising-edge triggered
  - falling-edge triggered
  - level sensitive
- The interrupts can be masked using a 32-bit mask register.
- Two levels of interrupt masking are provided. Interrupts can be individually masked at the bit level or at the port level.
- Software reset function: when the SWR bit (SWR register, 0 bit) is written as a 1, the entire GPIO module is reset immediately, and this reset signal is asserted for 3 system cycles. After this, the reset signal will be released automatically.

15.3 External Signals Description

Please refer to chapter 2, “Signal Descriptions and Pin Assignments” for details of the I/O multiplexing scheme and external connection to the GPIO module.

15.4 Interrupts

Every external input passes through the interrupt module in the GPIO module. Inside this module, the interrupts may be defined as rising-edge triggered, or falling-edge triggered. Each interrupt can be masked and also be designated as a high-level interrupt, or a low-level sensitive interrupt. The interrupt status register bits corresponding to the interrupts waiting for service are stored as a value of 1. The interrupt status register is Write 1 to Clear (w1c). The user is responsible for clearing the interrupt status register bit after it has been serviced.

15.5 Programming Model

GPIO has six ports and each port has 17 registers. In total, the GPIO has 102 registers. The registers, other than the Sample Status Register (SSR) and the Interrupt Status Register (ISR), have both read and write capability. The Sample Status Register is a read only register, while the Interrupt Status Register is a w1c register; the register can be read, but writing a 1 to any register bit clears the bit. Writing a value of 0 to the bit has no effect.

While there are six GPIO ports, each capable of representing 32 GPIO configurable pins as inputs or outputs, not all bits are mapped to a pin and hence these bits do not have any effect and are marked as reserved. These reserved bits are indicated in the Section 15.5.9, “GPIO IN USE Register (GIUS).”

In Table 15-1shows the absolute address of each memory mapped register.

Table 15-1. GPIO Register Summary

Description	Name	Address
Data Direction Register (DDIR)	PTA_DDIR	0x1001 5000
–	PTB_DDIR	0x1001 5100
–	PTC_DDIR	0x1001 5200
–	PTD_DDIR	0x1001 5300
–	PTE_DDIR	0x1001 5400
–	PTF_DDIR	0x1001 5500
Output Configuration Register 1 (OCR1)	PTA_OCR1	0x1001 5004
–	PTB_OCR1	0x1001 5104
–	PTC_OCR1	0x1001 5204
–	PTD_OCR1	0x1001 5304
–	PTE_OCR1	0x1001 5404
–	PTF_OCR1	0x1001 5504
Output Configuration Register 2 (OCR2)	PTA_OCR2	0x1001 5008
–	PTB_OCR2	0x1001 5108
–	PTC_OCR2	0x1001 5208
–	PTD_OCR2	0x1001 5308
–	PTE_OCR2	0x1001 5408
–	PTF_OCR2	0x1001 5508
Input Configuration Register A1 (ICONFA1)	PTA_ICONFA1	0x1001 500c
–	PTB_ICONFA1	0x1001 510c
–	PTC_ICONFA1	0x1001 520c
–	PTD_ICONFA1	0x1001 530c
–	PTE_ICONFA1	0x1001 540c
–	PTF_ICONFA1	0x1001 550c
Input Configuration Register A2 (ICONFA2)	PTA_ICONFA2	0x1001 5010
–	PTB_ICONFA2	0x1001 5110
–	PTC_ICONFA2	0x1001 5210
–	PTD_ICONFA2	0x1001 5310
–	PTE_ICONFA2	0x1001 5410
–	PTF_ICONFA2	0x1001 5510

Table 15-1. GPIO Register Summary (continued)

Description	Name	Address
Input Configuration Register B1 (ICONFB1)	PTA_ICONFB1	0x1001 5014
–	PTB_ICONFB1	0x1001 5114
–	PTC_ICONFB1	0x1001 5214
–	PTD_ICONFB1	0x1001 5314
–	PTE_ICONFB1	0x1001 5414
–	PTF_ICONFB1	0x1001 5514
Input Configuration Register B2 (ICONFB2)	PTA_ICONFB2	0x1001 5018
–	PTB_ICONFB2	0x1001 5118
–	PTC_ICONFB2	0x1001 5218
–	PTD_ICONFB2	0x1001 5318
–	PTE_ICONFB2	0x1001 5418
–	PTF_ICONFB2	0x1001 5518
Data Register (DR)	PTA_DR	0x1001 501c
–	PTB_DR	0x1001 511c
–	PTC_DR	0x1001 521c
–	PTD_DR	0x1001 531c
–	PTE_DR	0x1001 541c
–	PTF_DR	0x1001 551c
GPIO In Use Register (GIUS)	PTA_GIUS	0x1001 5020
–	PTB_GIUS	0x1001 5120
–	PTC_GIUS	0x1001 5220
–	PTD_GIUS	0x1001 5320
–	PTE_GIUS	0x1001 5420
–	PTF_GIUS	0x1001 5520
Sample Status Register (SSR)	PTA_SSR	0x1001 5024
–	PTB_SSR	0x1001 5124
–	PTC_SSR	0x1001 5224
–	PTD_SSR	0x1001 5324
–	PTE_SSR	0x1001 5424
–	PTF_SSR	0x1001 5524
Interrupt Configuration Register 1 (ICR1)	PTA_ICR1	0x1001 5028
–	PTB_ICR1	0x1001 5128
–	PTC_ICR1	0x1001 5228
–	PTD_ICR1	0x1001 5328
–	PTE_ICR1	0x1001 5428
–	PTF_ICR1	0x1001 5528
Interrupt Configuration Register 2 (ICR2)	PTA_ICR2	0x1001 502c
–	PTB_ICR2	0x1001 512c
–	PTC_ICR2	0x1001 522c
–	PTD_ICR2	0x1001 532c
–	PTE_ICR2	0x1001 542c
–	PTF_ICR2	0x1001 552c

Table 15-1. GPIO Register Summary (continued)

Description	Name	Address
Interrupt Mask Register (IMR)	PTA_IMR	0x1001 5030
–	PTB_IMR	0x1001 5130
–	PTC_IMR	0x1001 5230
–	PTD_IMR	0x1001 5330
–	PTE_IMR	0x1001 5430
–	PTF_IMR	0x1001 5530
Interrupt Status Register (ISR)	PTA_ISR	0x1001 5034
–	PTB_ISR	0x1001 5134
–	PTC_ISR	0x1001 5234
–	PTD_ISR	0x1001 5334
–	PTE_ISR	0x1001 54 34
–	PTF_ISR	0x1001 5534
General Purpose Register (GPR)	PTA_GPR	0x1001 5038
–	PTB_GPR	0x1001 5138
–	PTC_GPR	0x1001 5238
–	PTD_GPR	0x1001 5338
–	PTE_GPR	0x1001 5438
–	PTF_GPR	0x1001 5538
Software Reset Register (SWR)	PTA_SWR	0x1001 503c
–	PTB_SWR	0x1001 513c
–	PTC_SWR	0x1001 523c
–	PTD_SWR	0x1001 533c
–	PTE_SWR	0x1001 543c
–	PTF_SWR	0x1001 553c
Pull_Up Enable Register (PUEN)	PTA_PUEN	0x1001 5040
–	PTB_PUEN	0x1001 5140
–	PTC_PUEN	0x1001 5240
–	PTD_PUEN	0x1001 5340
–	PTE_PUEN	0x1001 5440
–	PTF_PUEN	0x1001 5540
GPIO Port Interrupt Mask (PMASK)	PMASK	0x1001 5600

15.5.1 Data Direction Register

The data direction registers determine whether each pin of the port is an input or an output pin.

Data Direction Registers	Addr
PTA_DDIR	0x10015000
PTB_DDIR	0x10015100
PTC_DDIR	0x10015200
PTD_DDIR	0x10015300
PTE_DDIR	0x10015400
PTF_DDIR	0x10015500

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DDIR															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DDIR															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 15-2. Data Direction Register Description

Name	Description	Settings
DDIR Bits 31–0	**Data Direction**—This is a read/write register which defines the current direction of the 32 pins of a port in the GPIO module.	0 = Pin number i is input 1 = Pin number i is output

15.5.2 Output Configuration Register 1 (OCR1)

Each port consists of 32-pins. Because the output configuration for each pin is described using a two-bit combination the output configuration of the pins is controlled by two identical 32-bit registers (OCR1 and OCR2). The Output Configuration register 1 (OCR1) configures the output signal for lower 16 pins (0–15) of the associated port.

	Output Configuration Register 1	Addr
PTA_OCR1		0x10015004
PTB_OCR1		0x10015104
PTC_OCR1		0x10015204
PTD_OCR1		0x10015304
PTE_OCR1		0x10015404
PTF_OCR1		0x10015504

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OCR1															
	pin 15		pin 14		pin 13		pin 12		pin 11		pin 10		pin 9		pin 8	
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCR1															
	pin 7		pin 6		pin 5		pin 4		pin 3		pin 2		pin 1		pin 0	
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 15-3. Output Configuration Register 1 Description

Name	Description	Settings		
OCR1 Bits 31–0	**Output Configuration Register 1**—This is a read/write register that selects how each pin (0–15) is used as an output by the GPIO.	OCR1 [2i +1]	OCR1 [2i]	Output Selected
		0	0	Input A_IN[i]
		0	1	Input B_IN[i]
		1	0	Input C_IN[i]
		1	1	Data Register [i]

15.5.3 Output Configuration Register 2 (OCR2)

The Output Configuration register 2 (OCR2) specifies the output signal for upper 16 pins (16-31) of the associated port. The output configuration for each pin is described with a two-bit combination.

	Output Configuration Register 2	Addr
PTA_OCR2		0x10015008
PTB_OCR2		0x10015108
PTC_OCR2		0x10015208
PTD_OCR2		0x10015308
PTE_OCR2		0x10015408
PTF_OCR2		0x10015508

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OCR2															
	pin 31		pin 30		pin 29		pin 28		pin 27		pin 26		pin 25		pin 24	
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCR2															
	pin 23		pin 22		pin 21		pin 20		pin 19		pin 18		pin 17		pin 16	
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 15-4. Output Configuration Register 2 Description

Name	Description	Settings		
OCR2 Bits 31–0	**Output Configuration Register 2**—This is a read/write register that selects how each pin (16–31) is used as an output by the GPIO.	OCR2 [2i - 32 +1]	OCR2 [2i - 32]	Output Selected
		0	0	Input A_IN[i]
		0	1	Input B_IN[i]
		1	0	Input C_IN[i]
		1	1	Data Register [i]

15.5.4 Input Configuration Register A1 (ICONFA1)

The input configuration registers (ICONFA1) specify the signal or value driven to the A_OUT signals that is connected to internal modules of the i.MX21. Each port pin is defined by two bits in the input configuration registers. Please refer to Chapter 2, “Signal Descriptions and Pin Assignments,” for detailed I/O pin information.

	Input Configuration Register A1	Addr
PTA_ICONFA1		**0x1001500C**
PTB_ICONFA1		**0x1001510C**
PTC_ICONFA1		**0x1001520C**
PTD_ICONFA1		**0x1001530C**
PTE_ICONFA1		**0x1001540C**
PTF_ICONFA1		**0x1001550C**

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICONFA1															
	pin 15		pin 14		pin 13		pin 12		pin 11		pin 10		pin 9		pin 8	
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICONFA1															
	pin 7		pin 6		pin 5		pin 4		pin 3		pin 2		pin 1		pin 0	
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 15-5. Input Configuration Register A1 Description

Name	Description	Settings		
ICONFA1 Bits 31–0	**Input Configuration**—Corresponds to port pins 0–15 and defines which one of the four options is driven to A_OUT [i]. Each port pin [i] (i = 0–15) requires two ICONFA1 bits to determine the input value.	ICONFA1 [2i +1]	ICONFA1 [2i]	Output selected
		0	0	GPIO_In [i]
		0	1	Interrupt Status register [i]
		1	0	0
		1	1	1

15.5.5 Input Configuration Register A2 (ICONFA2)

The input configuration registers (ICONFA2) specify the signal or value driven to the A_OUT signals connected to internal modules. There are two bits in the input configuration registers for each port pin. Please refer to Chapter 2, “Signal Descriptions and Pin Assignments,” for detail I/O pin information.

	Input Configuration Register A2	Addr
PTA_ICONFA2		0x10015010
PTB_ICONFA2		0x10015110
PTC_ICONFA2		0x10015210
PTD_ICONFA2		0x10015310
PTE_ICONFA2		0x10015410
PTF_ICONFA2		0x10015510

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICONFA2															
	pin 31		pin 30		pin 29		pin 28		pin 27		pin 26		pin 25		pin 24	
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICONFA2															
	pin 23		pin 22		pin 21		pin 20		pin 19		pin 18		pin 17		pin 16	
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 15-6. Input Configuration Register A2 Description

Name	Description	Settings		
ICONFA2 Bits 31–0	**Input Configuration**—Corresponds to pins 16–31 of the port and defines which one of the four options is driven to a_OUT [i]. Each port pin [i] (i = 16 through 31) requires two ICONFA2 bits to determine the input value.	ICONFA2 [2i +1]	ICONFA2 [2i]	Output selected
		0	0	GPIO_In [i]
		0	1	Interrupt Status register [i]
		1	0	0
		1	1	1

15.5.6 Input Configuration Register B1 (ICONFB1)

The input configuration registers ICONFB1 specify the signal or value driven to the B_OUT signals connected to internal modules. There are two bits in the input configuration registers for each port pin. Please refer to Chapter 2, “Signal Descriptions and Pin Assignments,” for detail I/O pin information.

	Input Configuration Register B1	Addr
PTA_ICONFB1		0x10015014
PTB_ICONFB1		0x10015114
PTC_ICONFB1		0x10015214
PTD_ICONFB1		0x10015314
PTE_ICONFB1		0x10015414
PTF_ICONFB1		0x10015514

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICONFB1															
	pin 15		pin 14		pin 13		pin 12		pin 11		pin 10		pin 9		pin 8	
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICONFB1															
	pin 7		pin 6		pin 5		pin 4		pin 3		pin 2		pin 1		pin 0	
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 15-7. Input Configuration Register B1 Description

Name	Description	Settings		
ICONFB1 Bits 31–0	**Input Configuration**—Corresponds to pins 0–15 of the port and defines which one of the four options is driven to b_OUT [i]. Each port pin [i] (i = 0 through 15) requires two ICONFB1 bits to determine the input value.	ICONFB1 [2i +1]	ICONFB1 [2i]	Output selected
		0	0	GPIO_In[i]
		0	1	Interrupt Status register [i]
		1	0	0
		1	1	1

15.5.7 Input Configuration Register B2 (ICONFB2)

The input configuration registers ICONFB2 specify the signal or value driven to the B_OUT signals connected to internal modules. There are two bits in the input configuration registers for each port pin. Please refer to Chapter 2, “Signal Descriptions and Pin Assignments,” for detail I/O pin information.

	Input Configuration Register B2	Addr
PTA_ICONFB2		0x10015018
PTB_ICONFB2		0x10015118
PTC_ICONFB2		0x10015218
PTD_ICONFB2		0x10015318
PTE_ICONFB2		0x10015418
PTF_ICONFB2		0x10015518

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICONFB2															
	pin 31		pin 30		pin 29		pin 28		pin 27		pin 26		pin 25		pin 24	
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICONFB2															
	pin 23		pin 22		pin 21		pin 20		pin 19		pin 18		pin 17		pin 16	
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 15-8. Input Configuration Register B2 Description

Name	Description	Settings		
ICONFB2 Bits 31–0	**Input Configuration**—Corresponds to pins 16–31 of the port and defines which one of the four options is driven to b_OUT [i]. Each port pin [i] (i = 16 through 31) requires two ICONFB2 bits to determine the input value.	ICONFB2 [2i - 32 +1]	ICONFB2 [2i - 32]	Output selected
		0	0	GPIO_In[i]
		0	1	Interrupt Status register [i]
		1	0	0
		1	1	1

15.5.8 Data Register (DR)

The Data Register holds data for output from an associated port when a pin is configured as an output and the Data Register is chosen using Output Configuration Register 1 and Output Configuration Register 2.

	Data Register	ADDR
PTA_DR		0x1001501c
PTB_DR		0x1001511c
PTC_DR		0x1001521c
PTD_DR		0x1001531c
PTE_DR		0x1001541c
PTF_DR		0x1001551c

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DR															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DR															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 15-9. Data Register Description

Name	Description	Settings
DR Bits 31–0	**Data Register**—Contains the GPIO output values when the Output Configuration Registers select the Data Register as the output for the pin (selection 11).	0 = Drives the output signal low 1 = Drives the output signal high

15.5.9 GPIO IN USE Register (GIUS)

The GPIO In Use Registers control a multiplexer in the IOMUX module. The settings in these registers choose whether a pin is utilized for a peripheral function or for its GPIO function. If the register is set to a zero for a corresponding pin, then this register is used in conjunction with the GPR register to control the peripheral functionality. Reset values for individual registers are shown in the following sections

	GPIO In Use Register	Addr
PTA_GIUS		0x10015020
PTB_GIUS		0x10015120
PTC_GIUS		0x10015220
PTD_GIUS		0x10015320
PTE_GIUS		0x10015420
PTF_GIUS		0x10015520

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GIUS															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Reset Value—INUSE_RESET_SEL[31:16]

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GIUS															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Reset Value—INUSE_RESET_SEL[15:0]

Table 15-10. GPIO In Use Register Description

Name	Description	Settings
GIUS Bits 31–0	**GPIO In Use**—Informs the IOMUX module whether the port pin is utilized for its GPIO function. When the pin is utilized for its GPIO function, the multiplexed functions are not available. The reset value of this register is determined by the input value of the signal INUSE_RESET_SEL [31:0].	0 = Pin utilized for multiplexed function 1 = Pin utilized for GPIO function

The following sections describe the GPIO In Use (GIUS) reset values for the various ports. Additionally, the registers also indicate the reserved bits (unimplemented GPIO bits) of the GPIO ports.

15.5.9.1 GPIO IN USE Register A (PTA_GIUS)

The reset value of the PTA_GIUS register is (0xFFFF_FFE0).

PTA_GIUS — GPIO In Use Register A — Addr 0x10015020

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GIUS															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GIUS															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	Reserved				
Reset	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0

15.5.9.2 GPIO IN USE Register B (PTB_GIUS)

The reset value of the PTB_GIUS register is (0xFF3F_FFF0).

PTB_GIUS — GPIO In Use Register B — Addr 0x10015120

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GIUS															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GIUS															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	Reserved			
Reset	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

15.5.9.3 GPIO IN USE Register C (PTC_GIUS)

The reset value of the PTC_GIUS register is (0xFFFF_FFE0).

PTC_GIUS — GPIO In Use Register C — Addr 0x10015220

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GIUS															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GIUS															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	Reserved				
Reset	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0

15.5.9.4 GPIO IN USE Register D (PTD_GIUS)

The reset value of the PTD_GIUS register is (0xFFFE_0000).

PTD_GIUS — GPIO In Use Register D — Addr 0x10015320

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GIUS															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	Reserved
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GIUS															
Type	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.9.5 GPIO IN USE Register E (PTE_GIUS)

The reset value of the PTE_GIUS register is (0x00FC_0F20).

PTE_GIUS — GPIO In Use Register E — Addr 0x10015420

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GIUS															
Type	Reserved								rw	rw	rw	rw	rw	rw	rw	rw
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GIUS															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	0	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0

15.5.9.6 GPIO IN USE Register F (PTF_GIUS)

The reset value of the PTF_GIUS register is (0x0060_0000).

PTF_GIUS — GPIO In Use Register F — Addr 0x10015520

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GIUS															
Type	Reserved									rw	rw	Reserved				rw
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GIUS															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.10 Sample Status Register (SSR)

The read-only Sample Status Registers contain the value of the GPIO pins for each associated port. The register is updated on every clock tick. The contents are used as a status indicator when the pins are configured as inputs.

	Sample Status Register	Addr
PTA_SSR		0x10015024
PTB_SSR		0x10015124
PTC_SSR		0x10015224
PTD_SSR		0x10015324
PTE_SSR		0x10015424
PTF_SSR		0x10015524

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SSR															
Type	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SSR															
Type	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 15-11. Sample Status Register Description

Name	Description	Settings
SSR Bits 31–0	**Sample Status**—Contains the value of the GPIO pin [i]. It is sampled on every clock.	0 = Pin value is low 1 = Pin value is high

15.5.11 Interrupt Configuration Register 1 (ICR 1)

This register specifies the external interrupt configuration for each of the lower 16 interrupts of a port. There are two bits in the register for each port pin.

	Interrupt Configuration Register 1	Addr
PTA_ICR1		0x10015028
PTB_ICR1		0x10015128
PTC_ICR1		0x10015228
PTD_ICR1		0x10015328
PTE_ICR1		0x10015428
PTF_ICR1		0x10015528

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICR1															
	pin 15		pin 14		pin 13		pin 12		pin 11		pin 10		pin 9		pin 8	
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICR1															
	pin 7		pin 6		pin 5		pin 4		pin 3		pin 2		pin 1		pin 0	
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 15-12. Interrupt Configuration Register 1 Description

Name	Description	Settings		
ICR1 Bits 31–0	**Interrupt Configuration**—Corresponds to interrupts 0–15 of the port and defines which one of the four options is the sensitivity of the interrupt. Each interrupt [i] (i= 0 through 15) requires two ICR1 bits to determine the sensitivity.	ICR1 [2i +1]	ICR1 [2i]	Sensitivity selected
		0	0	Rising edge sensitive
		0	1	Falling edge sensitive
		1	0	High level sensitive
		1	1	Low level sensitive

15.5.11.1 Interrupt Configuration Register 2 (ICR 2)

This register specify the external interrupt configuration for each of the upper 16 interrupts of the port. There are two bits in the register for each port pin.

	Interrupt Configuration Register 2	Addr
PTA_ICR2		0x1001502c
PTB_ICR2		0x1001512c
PTC_ICR2		0x1001522c
PTD_ICR2		0x1001532c
PTE_ICR2		0x1001542c
PTF_ICR2		0x1001552c

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICR2															
Pin Number	pin 31		pin 30		pin 29		pin 28		pin 27		pin 26		pin 25		pin 24	
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICR2															
Pin Number	pin 23		pin 22		pin 21		pin 20		pin 19		pin 18		pin 17		pin 16	
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 15-13. Interrupt Configuration Register 2 Description

Name	Description	Settings		
ICR2 Bits 31–0	**Interrupt Configuration**—Corresponds to interrupts 16–31 of the port and defines which one of the four options is the sensitivity of the interrupt. Each interrupt [i] (i= 16 through 31) requires two ICR2 bits to determine the sensitivity.	ICR2 [2i - 32 +1]	ICR2 [2i -32]	Sensitivity selected
		0	0	Rising edge sensitive
		0	1	Falling edge sensitive
		1	0	High level sensitive
		1	1	Low level sensitive

15.5.12 Interrupt Mask Register (IMR)

The Interrupt Mask Registers (IMR) determine if an interrupt will be asserted when an interrupt event occurs and when the pin and corresponding bit is configured in an interrupt mode. An interrupt is asserted when corresponding bits in the IMR and ISR are set.

	Interrupt Mask Register	Addr
PTA_IMR		0x10015030
PTB_IMR		0x10015130
PTC_IMR		0x10015230
PTD_IMR		0x10015330
PTE_IMR		0x10015430
PTF_IMR		0x10015530

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IMR															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IMR															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 15-14. Interrupt Mask Register Description

Name	Description	Settings
IMR Bits 31–0	**Interrupt Mask**—Masks the interrupts for this module.	0 = Interrupt is masked 1 = Interrupt is not masked

15.5.13 Interrupt Status Register (ISR)

The Interrupt Status Registers (ISR) indicate if an interrupt has occurred. When an interrupt event occurs, the bit in this register is set. The condition necessary to set the bit is determined by the Interrupt Configuration Registers (ICR) and the inputs satisfying the interrupt condition.

	Interrupt Status Register	Addr
PTA_ISR		0x10015034
PTB_ISR		0x10015134
PTC_ISR		0x10015234
PTD_ISR		0x10015334
PTE_ISR		0x10015434
PTF_ISR		0x10015534

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ISR															
Type	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ISR															
Type	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 15-15. Interrupt Status Register Description

Name	Description	Settings
ISR Bits 31–0	**Interrupt Status**—Indicates whether the interrupt [i] has occurred for in the GPIO module. The bits of this register are write 1 to clear. The w1c bits will be cleared when a value of 1 is written to the associated bit.	0 = Interrupt has not occurred 1 = Interrupt has occurred

15.5.14 General Purpose Register (GPR)

The General Purpose Registers (GPR) control a multiplexer in the IOMUX module. When the corresponding bit in the associated GIUS register is set to zero, the settings in these registers determine whether a pin is utilized for its primary peripheral function or for its alternate peripheral function. When the corresponding bit in the GIUS is set, the settings of this register have no effect.

PTA_GPR	General Purpose Register	Addr
PTB_GPR		0x10015038
PTC_GPR		0x10015138
PTD_GPR		0x10015238
PTE_GPR		0x10015338
PTF_GPR		0x10015438
		0x10015538

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GPR															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPR															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 15-16. General Purpose Register Description

Name	Description	Settings
GPR Bits 31–0	**General Purpose Register**—Selects between the primary and alternate functions of the pin. When the associated bit in the GIUS register is set, this bit has no meaning. **Note:** Ensure that this bit is cleared when there is no alternate function for a particular pin.	0 = Select primary pin function 1 = Select alternate pin function

15.5.15 Software Reset Register (SWR)

The Software Reset Register (SWR) controls the reset of the individual ports in the GPIO module. When the SWR bit of the Software Reset Register is set, the GPIO circuitry for the individual port resets immediately.

The total time of the software reset sequence will take six clock cycles. The reset will be asserted from the third cycle and remains asserted for three clocks.

PTA_SWR	Software Reset Register	Addr
PTB_SWR		0x1001503c
PTC_SWR		0x1001513c
PTD_SWR		0x1001523c
PTE_SWR		0x1001533c
PTF_SWR		0x1001543c
		0x1001553c

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Type	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SWR
Type	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	slfclr
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 15-17. Software Reset Register

Name	Description	Settings
Reserved bits 31–1	Reserved—These bits are reserved and should read 0	
SWR Bit 0	**Software Reset**—Controls software reset of the port. The reset signal is active for 3 system clock cycles and then it is released automatically.	0 = No effect 1 = GPIO circuitry for Port X reset

15.5.16 Pull-Up Enable Register (PUEN)

The Pull-Up Enable (PUEN) Registers enable or disable a 69 kOhm pull-up resistor on the associated pin. The pull-up can be applied to any GPIO pin regardless of whether it is configured as primary, alternate or GPIO function. The pin is tri-stated when the pull-up is disabled and the pin is not driven.

NOTE

Bits 27–24 on Port A (PTA_PUEN) enables or disables a 69 kOhm pull-down resistor on the associated pin.

Bits 31–28, 26, and 9 on Port B (PTB_PUEN) enables or disables a 69 kOhm pull-down resistor on the associated pin.

Pull-Up Enable Register	Addr
PTA_PUEN	0x10015040
PTB_PUEN	0x10015140
PTC_PUEN	0x10015240
PTD_PUEN	0x10015340
PTE_PUEN	0x10015440
PTF_PUEN	0x10015540

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PUEN															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PUEN															
Type	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 15-18. Pull-Up Enable Register Description

Name	Description	Settings
PUEN [i] Bits 31–0	**Pull-Up Enable**—Determines whether the corresponding pad is pulled up to a logic-high or tri-stated. When the pin is configured as an input, clearing this bit causes the signal to be tri-stated when not driven by an external source. When the pin is configured as an output, clearing this bit causes the signal to be tri-stated when it is not enabled.	0 = Pin [i] is tri-stated when not driven internally or externally 1 = Pin [i] is pulled high1 when not driven internally or externally

15.5.17 Port Interrupt Mask Register (PMASK)

The GPIO has six ports, each with interrupt generation capability. The PMASK register provides interrupt masking capability at the port level while the Interrupt Mask Register provides control over individual interrupts. If a bit is zero, then all interrupts for that port are masked. A software reset on a port (SWR is set) will clear the corresponding mask bit of the port in this register.

PMASK — Port Interrupt Mask Register — Addr 0x10015600

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Type	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											PTF	PTE	PTD	PTC	PTB	PTA
Type	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Table 15-19. Port Interrupt Mask Register Description

Name	Description	Settings
Reserved Bits 31–6	Reserved—These bits are reserved and should read 0.	
PTF Bit 5	**Port F**—The bit helps in masking the Port F interrupt. The bit clears during software reset of Port F.	0 = Interrupt is masked 1 = Interrupt is not masked
PTE Bit 4	**Port E**—The bit helps in masking the Port E interrupt. The bit clears during software reset of Port E.	0 = Interrupt is masked 1 = Interrupt is not masked
PTD Bit 3	**Port D**—The bit helps in masking the Port D interrupt. The bit clears during software reset of Port D.	0 = Interrupt is masked 1 = Interrupt is not masked
PTC Bit 2	**Port C**—The bit helps in masking the Port C interrupt. The bit clears during software reset of Port C.	0 = Interrupt is masked 1 = Interrupt is not masked
PTB Bit 1	**Port B**—The bit helps in masking the Port B interrupt. The bit clears during software reset of Port B.	0 = Interrupt is masked 1 = Interrupt is not masked
PTA Bit 0	**Port A**—The bit helps in masking the Port A interrupt. The bit clears during software reset of Port A.	0 = Interrupt is masked 1 = Interrupt is not masked

Chapter 16
Pulse-Width Modulator (PWM)

The pulse-width modulator (PWM) of the i.MX21 is a 16-bit PWM module which is optimized to generate sound from stored sample audio images and it can also generate tones. It uses 16-bit resolution and a 4 × 16 data FIFO to generate sound.

The following features characterize the PWM:

- 16-bit resolution
- 4 × 16 FIFO to minimize interrupt overhead
- 2 stage input clock divider (2, 4, 8, 16-divider and 7-bit prescaler)
- Sound and melody generation
- Software reset function

16.1 Operation

The output of the PWM is a toggling signal whose frequency and duty cycle can be modulated by suitable programming of its registers. It has a 16-bit up counter which counts from 0x0000 until the Counter value equals the [Value in Period register] + 1. When this match occurs the Counter is reset to 0x0000.

At the beginning of a count period cycle, the PWMO pin is set to one and the counter begins counting up from 0x0000. The sample value in the Sample FIFO is compared on each count or prescaler clock. When the sample and count values match, the PWMO signal is cleared to zero. The counter continues counting until the Period match occurs and subsequently another period cycle begins.

When the PWM is enabled the counter starts running and generates output with the reset values in the period and sample registers. It is recommended that the programming of these registers be done before pwm is enabled.

A hardware reset results in all the PWM count and sample registers begin cleared and the FIFO being flushed. The control register shows that FIFO is empty and FIFO can be written into, and PWM is disabled. A software reset has the same result and can be given even when the PWM enable bit is off. The clocks from CRM to PWM should be on for the software reset to be functional.

16.1.1 Clocks

The clock input source is determined by the CLKSRC field of the PWM control register. The CLKSRC value should only be changed when the PWM is disabled.

The PWM divider divides the selected clock by 2,4,8,16 according to the CLKSEL bits in the control register. Furthermore the clock can be divided by 1–128 by a prescaler by appropriately setting the PRESCALE field in the control register.

The PWM clock gating is done inside the CRM module and the appropriate bit in PCCR register of CRM should be set for PWM to receive the clocks.

16.1.2 FIFO

Digital sample values can be loaded into the pulse-width modulator as 16-bit words in the same endian format as the processor is using. The endianess can be changed by suitably using the BCTR and HCTR bits of the control register. A 4-word(16-bit) FIFO minimizes interrupt overhead. A maskable interrupt is generated when there are 1 or 0 words in the FIFO, in which case, the software can write three 16-bit samples into the FIFO.

A write in the sample register results in the value being stored into the FIFO if it is not full. A write when FIFO is full is ignored and FIFO contents remain unchanged. The FIFO can only be written to when the PWM is disabled. Attempting to read data written to the FIFO when the PWM is enabled may result in invalid data.

A read on the sample register yields the current FIFO value being or will be used by PWM for generation on the output signal. Therefore a write and subsequent read on the sample register may result in different values being obtained.

16.1.3 Low-Power Mode

In Low Power mode when the clocks are switched off the PWM ceases to count. The counter freezes at its current value and all PWM operations are ceased.

16.2 Programming Model

The PWM module includes 4 user-accessible 32-bit registers. The Control Register which contains the control and status bits for configuring and monitoring the PWM.The Sample Register which is the entry point for the 4 × 16 FIFO and can be read to get the sample value begin used. It determines the duty cycle of the output signal. The Period Register which determines the period of the output signal. The Counter Register which can be read to get the present count value.

The Table 16-1 summarizes these registers and their addresses.

Table 16-1. PWM Register Summary

Name		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
PWMC (0x10006000)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	HCTR	BCTR	0
	W																SWR
	R	CLK SRC	PRESCALAR							IRQ	IRQ EN	FIFOAV	EN	REPEAT		CLKSEL	
	W									RTC							
PWMS (0x10006004)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	SAMPLE															
	W																
PWMP (0x10006008)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	PERIOD															
	W																
PWMCNT (0x1000600c)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	COUNT															
	W																

16.2.1 PWM Control Register

The pulse-width modulator control register (PWMC) controls how the overall pulse-width modulator operates. You can also find out the status of the FIFO with this register.

PWMC — PWM Control Register — 0x10006000

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
														HCTR	BCTR	SWR
TYPE	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	rw	rw	r0
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CLK SRC	PRESCALAR							IRQ	IRQ EN	FIFOAV	EN	REPEAT		CLKSEL	
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rtc	rw	r	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0

Table 16-2. PWM Control Register Description

Name	Description	Settings
Reserved Bits 31–19	Reserved—These bits are reserved and should read zero.	
HCTR Bit 18	**Half-Word Data Swap Control**—This bit determines which half word data from the 32-bit IP-Bus interface is written into the lower 16 bits of the sample register.	0 = Half word swapping does not take place 1 = Half words from write data bus are swapped
BCTR Bit 17	**Byte Data Swap Control**—This bit determines the byte ordering of the 16-bit data when it goes into the FIFO from the sample register.	0 = Byte ordering remains the same 1 = Byte ordering is reversed
SWR Bit 16	**Software Reset**—PWM is Reset when this bit is set to 1. This bit is set when the module is in reset state and is cleared when the reset procedure is over. The reset signal is asserted 2 clock cycles after this bit is set and the reset remains asserted for 3 clock cycles. The whole reset procedure is complete 5 clock cycles after this bit is asserted.	0 = PWM is out of reset 1 = PWM is undergoing reset
CLKSRC Bit 15	**Select Clock Source**—This bit is used to select the clock for the PWM counter. This field value should only be changed when the PWM is disabled.	0 = perclk (fixed freq. functional clock) 1 = 32 KHz clock (Low freq clock)
PRESCALAR Bits 14–8	**Counter Clock Prescaler Value**—This bit field determines the value by which the clock will be divided before it goes to the counter.	0000000 = [clock freq] / 1 0000001 = [clock freq] / 2 ... 1111111 = [clock freq] / 128
IRQ Bit 7	**Fifo Empty Interrupt Status**—This bit indicates that the FIFO has one or no word remaining. It is a signal to fill the FIFO by writing no more than three 16-bit words into the Sample register.This bit automatically clears itself after this register is read, thus eliminating an extra write cycle in the interrupt service routine.	0 = FIFO is not empty 1 = FIFO has one or no sample bytes
IRQEN Bit 6	**Fifo Empty Interrupt Enable**—This bit controls the pulse-width modulator interrupt. While this bit is low, the interrupt is disabled.	0 = FIFO interrupt disabled 1 = FIFO interrupt enabled
FIFOAV Bit 5	**FIFO Available**—This read-only bit indicates that the FIFO is available for at least one word of sample data. Data words can be loaded into the FIFO as long as this bit is set. If the FIFO is loaded while this bit is cleared, the write will be ignored.	0 = FIFO cannot be written into 1 = FIFO can be written into
EN Bit 4	**PWM Enable**—This bit enables the pulse-width modulator. If this bit is not enabled the following events occur: • The clock prescaler is reset and frozen. • The counter is reset and frozen. When the pulse-width modulator is enabled, it begins a new period, and the following events occur: • The output pin is set to start a new period. • The prescaler and counter are released and counting begins.	0 = The pulse-width modulator is disabled 1 = The pulse-width modulator is enabled

Table 16-2. PWM Control Register Description (continued)

Name	Description	Settings
REPEAT Bits 3–2	**Sample Repeat**—This bit field determines the no of times each sample from the FIFO is to be used.	00 = Use each sample once 01 = Use each sample twice 10 = Use each sample four times 11 = Use each sample eight times
CLKSEL Bits 1–0	**Divide Value**—This bit filed selects the divide value of the divider chain which divides the clock going to the counter.	00 = Divide by 2 01 = Divide by 4 10 = Divide by 8 11 = Divide by 16

16.2.2 PWM Sample Register

The pulse-width modulator sample (PWMS) register is the input to the FIFO. When you write successive audio sample values to this register, they are automatically loaded into the FIFO in big-endian format. 16-bit words are loaded into the FIFO. The FIFO can only be written into when PWM is disabled. The read may result in invalid data. The pulse-width modulator will run at the last set duty-cycle setting even if all the values of the FIFO has been utilized until the FIFO is reloaded or the pulse-width modulator is disabled. When a new value is written the duty cycle changes after the present period is over.

A value of zero in the sample register will result in ipp_pwm_pwmo being always low and hence no output waveform. If the value in this register is higher than the PERIOD + 1, the output will never be reset, which results in a 100% duty-cycle.

PWMS — PWM Sample Register — 0x10006004

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAMPLE															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 16-3. PWM Sample Register Description

Name	Description
Reserved Bits 31–16	Reserved—These bits are reserved and should read zero.
SAMPLE Bits 15–0	**Sample Value**—This is the input to the 4 × 16 FIFO. The value in this register denotes the value of the sample being currently used.

16.2.3 PWM Period Register

The read/write pulse-width modulator period register (PWMP) controls the pulse-width modulator period. After the counter value matches PERIOD + 1, the counter is reset to start another period. Therefore, PWMO (Hz) = PCLK(Hz) / (period +2). A value of Zero in the period register a period of two clock cycles for the output signal. Writing 0xFFFF to this register will achieve the same result as writing 0xFFFE.

Writing into the period register results in the counter being loaded with zero and the start of a new count period.

PWMP — PWM Period Register — 0x10006008

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PERIOD															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Table 16-4. PWM Period Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
PERIOD Bits 15–0	**Period Value**—This register determines the Period of the count cycle. The counter counts upto [Period Value] +1 and is then reset to 0x0000.	0x0000 = Count upto 0x0001 ... 0xfffe = Count upto 0xffff 0xffff = Count upto 0xffff

16.2.4 PWM Counter Register

The read-only pulse-width modulator counter register (PWMCNT) contains the current count value and can be read at any time without disturbing the counter.

PWMCNT — PWM Counter Register — 0x1000600C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COUNT															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 16-5. PWM Counter Register Description

Name	Description
Reserved Bits 31–16	Reserved—These bits are reserved and should read zero.
COUNT Bits 15–0	**Counter Value**—This is the counter register value and denotes the current count state the Counter register is in.

Part 4
Memory Interfaces

Chapter 17
SDRAM Memory Controller

This chapter describes the SDRAM Controller of the i.MX21 and how to configure and program it to work with a wide variety of SDRAM devices. The block diagram of the SDRAM Controller is shown in Figure 17-1.

The SDRAM Controller includes these distinctive features:

- Supports 4 banks of 64, 128, 256, or 512 Mbit synchronous DRAMs
- Includes 2 independent chip selects
  - Up to 64 Mbytes per chip select
  - Up to four banks simultaneously active per chip select
  - JEDEC standard pinout/operation
- Supports SDRAM-interface burst memory
- PC133 compliant interface
  - 133 MHz system clock achievable with “-75” option PC133 compliant memories
  - single and fixed-length (8-word) word access
  - Typical access time of 8-1-1-1-1-1-1-1 at 133 MHz
- Software configurable bus width, row and column sizes and delays for differing system requirements
- Built in auto-refresh timer and state machine
- Hardware supported self-refresh entry and exit which keeps data valid during system reset and low power modes
- Auto power-down (precharge and active power-down) timer

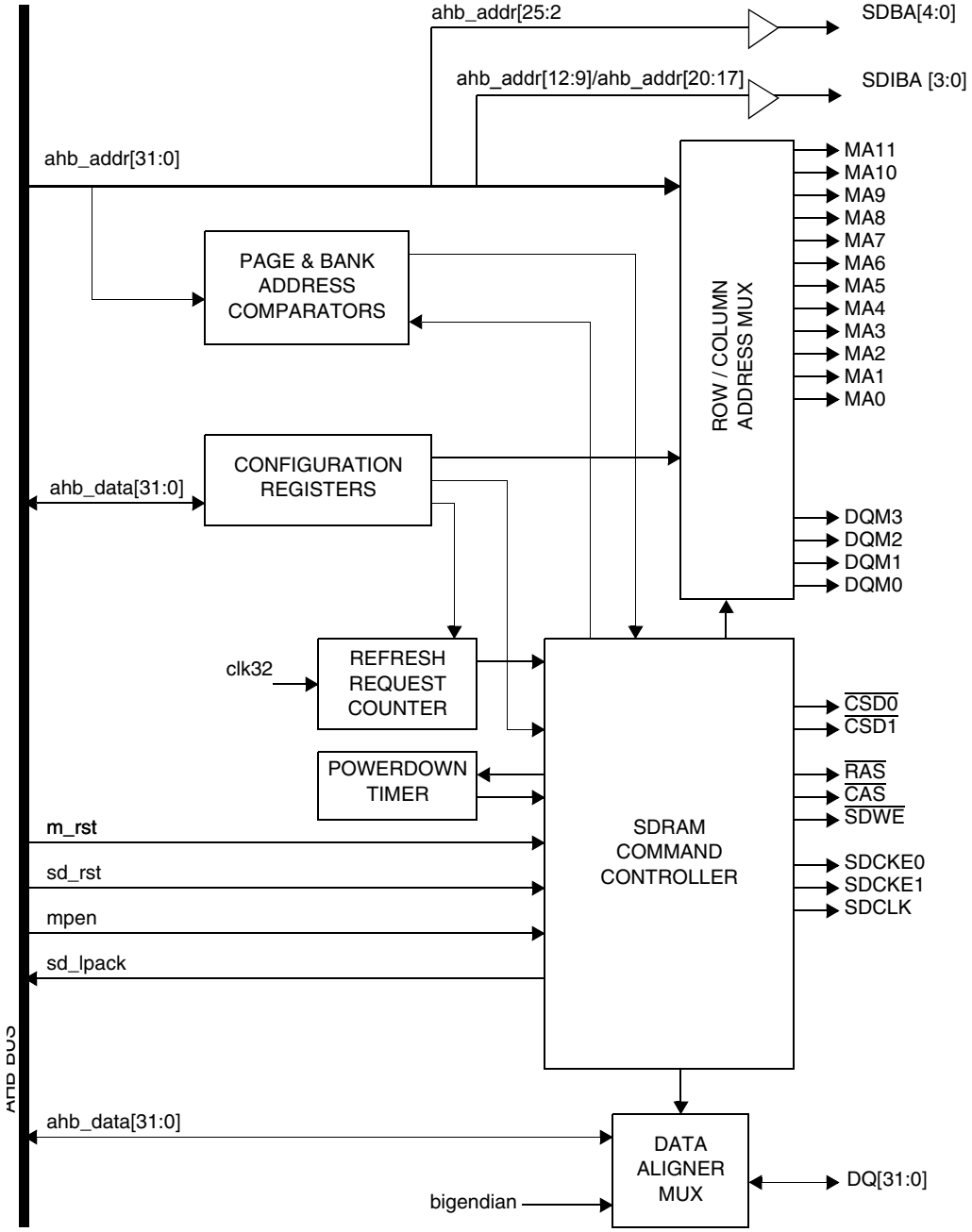


Figure 17-1. SDRAM Controller Block Diagram

17.1 Functional Overview

The SDRAM Controller consists of 7 major blocks, including the SDRAM command controller, page and bank address comparators, row/column address multiplexer, data aligner/multiplexer, configuration registers, refresh request counter, and the powerdown timer.

17.1.1 SDRAM Command Controller

The command controller handles the majority of the actions within the SDRAM controller including sequencing accesses to the memories, initializing the DRAM, keeping track of active banks within each memory region, scheduling refresh operations, transitioning into and out of low power modes, and controlling the address and data multiplexers.

17.1.2 Page and Bank Address Comparators

There are a total of 8 address comparators. Each chip select has a unique comparator for each of its four banks. The comparators are used to determine if a requested access falls within the address range of a currently active DRAM page.

17.1.3 Row/Column Address Multiplexer

All synchronous DRAMs incorporate a multiplexed address bus, although the address folding points vary according to memory density, data I/O size, and processor data bus width. The address folding point is described as the point where the column address bits end and the row (or bank) address begin. The SDRAM Controller takes these variables into account and provides the proper alignment of the multiplexed address through the row/column address multiplexer, non-multiplexed address pins, and the connections between the controller and the memory devices.

17.1.4 Data Aligner/Multiplexer

The data alignment block is responsible for aligning the data between the internal AHB bus and the external memory device(s) including little endian byte swapping. It is also used to concatenate data when a 32-bit AHB access has been broken down into 2 external 16-bit accesses.

17.1.5 Configuration Registers

Configuration registers determine the operating mode of the SDRAM Controller by selecting memory device density and bus width, the number of memory devices, CAS latency, row to column delay, and the burst length. Enable bits are provided for refresh and the auto-powerdown timer. Control bits provide a mechanism for software-initiated SDRAM initialization, SDRAM mode register settings, and all bank precharge and auto-refresh cycles.

17.1.6 Refresh Request Counter

SDRAM memories require a periodic refresh to retain data. The refresh request counter generates requests to the SDRAM Command Controller to perform these refresh cycles. Requests are scheduled according to a 32 KHz clock input. 1, 2 or 4 refresh cycles are generated per clock.

17.1.7 Powerdown Timer

The powerdown timer detects periods of inactivity to the SDRAM and disables the clock when the inactive period surpasses the selected timeout. Data is retained during the powerdown state. Subsequent requests

to the SDRAM incur only a minimal added start-up delay (beyond the normal access time). The powerdown timer may be programmed to expire anytime the controller is not actively reading/writing the memory, after 64 or 128 clocks of inactivity, or may be disabled entirely.

17.1.8 DMA Operation with the SDRAM Controller

The DMA controller has the capability to perform burst transfers of byte and half word data types while the SDRAM controller support is restricted to burst transfers of word (32-bit) data types. Therefore, when using the DMA in conjunction with the SDRAM controller, ensure that all burst transfers to/from the SDRAM controller are of word data types. This is configured in the DMA Channel Control Register. When choosing SDRAM memory as the source or destination address, set the SDRAMC as a 32-bit port. Refer to Chapter 18, “Direct Memory Access Controller (DMAC),” for more details.

17.1.9 External Interface

This section discusses input and output signals between the SDRAM Controller and the external memory devices. Other than the chip select outputs ($\overline{\text{CSD0}}$ / $\overline{\text{CSD1}}$) and clock enables (SDCKE0 / SDCKE1), all signals are shared between the two chip select regions. The interface signals are summarized in Table 17-1 and detailed in Section 17.1.10 through Section 17.1.20. Interconnect and timing diagrams are included as part of the detailed discussion on controller operation in Section 17.3, “Operating Modes.”

All external interface signals are referenced to the SDRAM clock, SDCLK.

Table 17-1. SDRAM Interface Pin Characteristics

SDRAMC Signal Name	i.MX21 Pin Name	Function	Direction	Reset State
SDCLK	SDCLK	Clock to SDRAM	Output	Enabled
SDCKE0	SDCLKE0	Clock enable to SDRAM 0	Output	High
SDCKE1	SDCLKE1	Clock enable to SDRAM 1	Output	High
CSD0	CS2	Chip select to SDRAM array 0	Output	High
CSD1	CS3	Chip select to SDRAM array 1	Output	High
MA [9:0]	A [10:1]	Multiplexed address bus	Output	Low
MA[11:10]	MA[11:10]	Multiplexed address bus signals 11 and 10	Output	Low
SDBA [4:0] / arm_addr [25:21]	A [20:16]	Non-multiplexed address bus	Output	Low
SDIBA [3:0] / arm_addr [12:9] / arm_addr[20:17]	A [24:21]	Non-multiplexed address bus	Output	Low
DQM3	DQM3	Data qualifier mask byte 3 (D [31:24])	Output	Low
DQM2	DQM2	Data qualifier mask byte 2 (D [23:16])	Output	Low
DQM1	DQM1	Data qualifier mask byte 1 (D [15:8])	Output	Low
DQM0	DQM0	Data qualifier mask byte 0 (D [7:0])	Output	Low
DQ [31:0]	D [31:0]	Data I/O bus	I/O	Unknown

Table 17-1. SDRAM Interface Pin Characteristics (continued)

SDRAMC Signal Name	i.MX21 Pin Name	Function	Direction	Reset State
SDWE	SDWE	Write enable	Output	High
RAS	RAS	Row address strobe	Output	High
CAS	CAS	Column address strobe	Output	High

17.1.10 SDCLK—SDRAM Clock

The SDCLK output provides the timing reference for the memory devices. All other SDRAM interface signals are referenced to this clock. SDCLK is synchronous to the system clock, but is gated off during low power operating modes when both SDCKE0 and SDCKE1 are negated.

17.1.11 SDCKE0, SDCKE1—SDRAM Clock Enables

The SDCKE0 and SDCKE1 pins are clock enable outputs to the SDRAM memory devices. SDCKE0 corresponds to SDRAM array 0 and SDCKE1 to SDRAM array 1. When these pins are asserted high, the memory's clock input is active, which means that a stable clock is being supplied. The low assertion deactivates the memory's clock input. A low assertion of SDCKEx initiates Power-Down, Self Refresh, and Suspend modes to the SDRAM.

17.1.12 $\overline{\text{CSD0}}$, $\overline{\text{CSD1}}$—SDRAM Chip Select

$\overline{\text{CSD0}}$ and $\overline{\text{CSD1}}$ are used to select SDRAM Array 0 and SDRAM Array 1, respectively. When a valid command is present on the other control signals, the chip select signals are used to indicate which device the command is directed towards. these signals are multiplexed with chip select signals $\overline{\text{CS2}}$ and $\overline{\text{CS3}}$ respectively.

17.1.13 DQ [31:0]—Data Bus

The 32 data lines are used to transfer data between the SDRAM Controller and memory. Data bit 31 is the most significant bit and bit 0 is the least significant. The memory alignment can be set to the upper 16 bits [31:16], lower 16 bits [15:0] or 32 bits by programming the DSIZ field in the SDCTLs registers. See Section 17.2.1, "SDRAM Control Registers." The SDRAM Controller data bus DQ[31:0] are multiplexed on the external data bus D[31:0].

17.1.14 MA [11:0]—Multiplexed Address Bus

The multiplexed address bus specifies the SDRAM page and the location within the page targeted by the current access. The multiplexed address pins are used in conjunction with some of the non-multiplexed ARM926EJ-S processor address signals to comprise the complete SDRAM address. Connections between the SDRAM Controller and memory vary depending on the SDRAM device density. See Section 17.4.1, "Address Multiplexing," and specifically Table 17-12 through Table 17-19 for details on supported SDRAM configurations. The SDRAM Controller multiplexed address bus MA [9:0] are multiplexed on

the external address bus A [10:1] while MA[11:10] are not multiplexed with any address signals (they are simply brought out as MA[11:10]).

17.1.15 SDBA [4:0], SDIBA [3:0]—Non-Multiplexed Address Bus

The non-multiplexed address pins specify the SDRAM bank to which the current command is targeted. The SDRAM Controller address signals SDBA [4:0] refer to non-interleaved bank addressing while SDIBA [3:0] refer to the interleaved bank addressing. SDBA[4:0] sample the address signals ahb_addr[25:12] and is multiplexed out on the external address bus signals A [20:16]. SDIBA [3:0] sample the address signals ahb_addr [12:9] or ahb_addr [20:17] and these signals are multiplexed out on the external address bus A [24:21]. In some density/width configurations, these pins also supply the most significant bits of the row address. Table 17-10 and Table 17-11 document which address pins are used for any given configuration.

17.1.16 DQM3, DQM2, DQM1, DQM0—Data Qualifier Mask

During read cycles, the DQMx pins controls the SDRAM data output buffers. DQMx asserted high disables the output buffers leaving them in a high-impedance state. DQMx asserted low allows the data buffers to drive normally.

During write cycles, DQMx controls which bytes are written in the SDRAM. DQMx asserted low enables a write to the corresponding byte, while DQMx asserted high leaves the byte unchanged.

DQM3 corresponds to the most significant byte and DQM0 to the least significant. Sixteen bit memories require only two DQM connections. Memories aligned to the upper data bus (D [31:16]) connect to DQM3 and DQM2, while memories aligned to the lower data bus (D [15:0]) connect to DQM1 and DQM0. Memory alignment is selected in the SDCTLx Registers.

17.1.17 $\overline{\text{SDWE}}$—Write Enable

Write enable is part of the three bit command field ($\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ make up the other two bits) used by the SDRAM. Generally, $\overline{\text{SDWE}}$ will be asserted low if a command transfers data to the memory. A detailed summary of the supported SDRAM commands is provided in Table 17-8.

17.1.18 $\overline{\text{RAS}}$—Row Address Strobe

Row address strobe is also part of the SDRAM command field. It is generally used to indicate an operation affecting an entire bank or row. When $\overline{\text{RAS}}$ is asserted (low), a new SDRAM row address must be latched. Table 17-8 provides details on SDRAM command encoding.

17.1.19 $\overline{\text{CAS}}$—Column Address Strobe

The column address strobe is the third signal comprised in the command field. It generally signifies a column oriented command. When $\overline{\text{CAS}}$ is asserted (low), the column address has changed. Table 17-8 provides details on SDRAM command encoding.

17.1.20 Pin Configuration for SDRAMC

Table 17-2 lists the pins used for the SDRAMC module. These pins are multiplexed with other functions on the device, and must be configured for SDRAM operation.

Table 17-2. Pin Configuration

Pin	Setting	Set-Up Procedure
SDCLK	Not Multiplexed	
SDCKE0	Not Multiplexed	
SDCKE1	Not Multiplexed	
CSD0	Alternate Function of $\overline{CS2}$/$\overline{CSD0}$ pin	Set bit 0 (SDCS0_SEL) in the Function Multiplexing Control Register of the System Control Module.
CSD1	Alternate Function of $\overline{CS3}$/$\overline{CSD1}$ pin	Set bit 1 (SDCS1_SEL) in the Function Multiplexing Control Register of the System Control Module.
MA[11:10]	Not Multiplexed	
MA [9:0]	Multiplexed with A [10:1]	Internal Signal from SDRAMC Asserted for SDRAM Accesses
SDBA [4:0]	Multiplexed with A [20:16]	Internal Signal from SDRAMC Asserted for SDRAM Accesses
SDIBA [3:0]	Multiplexed with A [24:21]	Internal Signal from SDRAMC Asserted for SDRAM Accesses
DQM3	Not Multiplexed	
DQM2	Not Multiplexed	
DQM1	Not Multiplexed	
DQM0	Not Multiplexed	
SDWE	Not Multiplexed	
RAS	Not Multiplexed	
CAS	Not Multiplexed	

17.2 Programming Model

The SDRAM module includes four 32-bit registers. Table 17-3 summarizes these registers and their addresses. Registers are accessible in supervisor mode only. Attempts to access the registers in user mode will result in a transfer error (TEA) being returned on the ARM926EJ-S processor's local bus.

Table 17-3. SDRAM Module Register Summary

Description	Name	Address
SDRAM 0 Control Register	SDCTL0	0xDF000000
SDRAM 1 Control Register	SDCTL1	0xDF000004
SDRAM Reset Register	SDRST	0xDF000018
Miscellaneous Register	MISCELLANEOUS	0xDF000014

The SDRAM arrays are mapped according to Table 17-4. A 64 Mbyte space is allocated to each chip select. Memories smaller than the allocated region are redundantly mapped throughout the remainder of the region. Attempted accesses to a disabled chip select region (SDE bit of the SDCTLx = 0) and User accesses to a protected (SP bit of the SDCTL = 1) region will result in a transfer error.

Table 17-4. SDRAM Array Memory Map

Address	Use	Access
0xC000 0000–0xC3FF FFFF	SDRAM 0 Memory Array	R/W
0xC400 0000–0xC7FF FFFF	SDRAM 1 Memory Array	R/W

17.2.1 SDRAM Control Registers

There are two SDRAM Control Registers, one for each of the two memory regions. SDCTL0 defines the operating characteristics for the SDRAM 0 region (selected by $\overline{CSD0}$), while SDCTL1 does the same for the SDRAM 1 region (selected by $\overline{CSD1}$). Bit field assignments within the registers are identical.

														Addr			
SDCTL0							SDRAM 0 Control Register							0xDF000000			
SDCTL1							SDRAM 1 Control Register							0xDF000004			
BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	SDE	SMODE			SP		ROW				COL		IAM		DSIZ		
TYPE	rw	rw	rw	rw	rw	r	rw	rw	r	r	rw	rw	rw	rw**	rw	rw	
RESET	0	0	0	0	0	0	0	1	0	0	0	0	0	0**	0	0	
	0x0100*																
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	SREFR		PWDT		CI		SCL			SRP	SRCD		SRC				
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	r	rw	rw	rw	rw	rw	rw	rw	
RESET	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	
	0x0300																

* For SDCTL1, the reset is 0x0106.

** For SDCTL1, the reset value is 1. Bit 18, need to be cleared by software for proper operation.

Table 17-5. SDRAM 0 Control Register and SDRAM 1 Control Register Description

Name	Description	Settings
SDE Bit 31	**SDRAM Controller Enable**—Enables/Disables the SDRAM controller. The module is disabled on reset. Disabling the module shuts off all clocks within the module with the exception of register accesses.	0 = Disabled 1 = Enabled
SMODE Bits 30–28	**SDRAM Controller Operating Mode**—Determines the operating mode of the SDRAM controller. The controller is capable of operating in seven different modes. These modes are primarily used for SDRAM initialization. Any access to the SDRAM memory space, while in one of the alternate modes, will result in the corresponding special cycle being run. Moving from Normal to any other mode does not close (precharge) any banks that may be open (activated). Under most circumstances, software should run a precharge-all cycle when transitioning out of Normal Read/Write mode. Operating mode details are provided in Section 17.5, “SDRAM Operation,” . Reset initializes the operating mode to “Normal Read/Write”.	000 = Normal Read/Write 001 = Precharge Command 010 = Auto-Refresh Command 011 = Set Mode Register Command 100 = Manual Self Refresh Command 101 = Reserved 110 = Reserved 111 = Reserved
SP Bit 27	**Supervisor Protect**—Restricts user accesses within the chip select region.	0 = User mode accesses are allowed to this chip select region. 1 = User mode accesses are prohibited. An attempted access to this chip select region while in user mode will result in a TEA being returned back to the CPU. The chip select will not be asserted.
Reserved Bit 26	Reserved—This bit is reserved and should read 0.	
ROW Bits 25–24	**Row Address Width**—Specifies the number of row addresses used by the memory array. This number does not include the bank, column, or data qualifier addresses. Parameters affected by the programming of this field include the page-hit address comparators and the bank address bit locations (non-interleaved mode only).	00 = 11 01 = 12 10 = 13 11 = Reserved
Reserved Bits 23–22	Reserved—These bits are reserved and should read 0.	
COL Bits 21–20	**Column Address Width**—Specifies the number of column addresses in the memory array and will determine the break point in the address multiplexer. Column width is the number of multiplexed column addresses and does not include bank and row addresses, or addresses used to generate the DQM signals.	00 = 8 01 = 9 10 = reserved 11 = reserved

Table 17-5. SDRAM 0 Control Register and SDRAM 1 Control Register Description (continued)

<table>
<tr><th>Name</th><th>Description</th><th>Settings</th></tr>
<tr><td>IAM
Bit 19</td><td>Interleaved Address Mode—Controls bank address alignment. Bank addresses fall between the row and column addresses, resulting in an interleaved memory map with the banks alternately striped through the memory region. They are more significant than row and column addresses and result in a linear addressing of the banks through the memory map.

Bank address bit placement has a significant effect on how well the SDRAM page buffers are utilized, with a corresponding impact on system performance.

The SDRAM Controller supports two bank address alignments which will satisfy most system requirements.

See Figure 17-2 for memory bank interleaving options.

Note: Memory address linearity is of little concern to the user when the memory is comprised of RAM devices. It is recommended that the user choose linear addressing (IAM = 0) for block oriented devices.</td><td>0 = Linear Address Map. Addresses flow through each page in the first bank, into the second bank, and so on. Linear Addressing is best suited to applications with large continuous blocks of linear accessed data such as an LCD display buffer.
1 = Interleaved Address Map. Addresses flow through one page in the first bank, to one page in the second, to the third, etc. The banks alternate at each SDRAM page boundary. Interleaved Mapping is better suited for ARM9 code space. The interleaving of the banks eliminates the need to continually open and close pages when loops and LRW constants cross page boundaries, resulting in higher system throughput.</td></tr>
<tr><td>Reserved
Bit 18</td><td colspan="2">Reserved—For SDCTL0 these bit is reserved and should read 0.
For SDCTL1 these bit is reserved but read as 1. For proper operation this bit need to be cleared (by writing “0”) by software if SDCTL1 is used.</td></tr>
<tr><td>DSIZ
Bits 17–16</td><td>SDRAM Memory Data Width—Defines the width of the SDRAM memory and its alignment on the external data bus. 16-bit ports may be aligned to either the high or low half-word to equalize capacitive loading on the bus. Data qualifier mask control outputs must be matched to the selected data bus alignment. Memories aligned to D [31:16] use DQM3 and DQM2. Memories aligned to D [15:0] use DQM1 and DQM0.</td><td>00 = 16-bit memory aligned to D [31:16]
01 = 16-bit memory aligned to D [15:0]
1x = 32-bit memory</td></tr>
<tr><td>SREFR
Bits 15–14</td><td>SDRAM Refresh Rate—Enables/Disables SDRAM refresh cycles and controls the refresh rate. Refresh cycles are referenced to a 32 KHz clock. At each rising edge, 1, 2, or 4 rows will be refreshed. Multiple refresh cycles are separated by the row cycle delay specified in the SRC control field.
Note: If SREF!=2’b00 the SDCTL will issue AUTO_REFRESH commands (each 32k clk, to the respective CS) regardless of SDE bits.</td><td>
<table>
<tr><th>SREFR [1:0]</th><th>Rows Each Refresh Clock</th><th>~Row/ 64ms @ 32 KHz</th><th>Row Rate @ 32 KHz</th></tr>
<tr><td>00</td><td colspan="3">Refresh Disabled</td></tr>
<tr><td>01</td><td>1</td><td>2048</td><td>31.25 µS</td></tr>
<tr><td>10</td><td>2</td><td>4096</td><td>15.62 µS</td></tr>
<tr><td>11</td><td>4</td><td>8192</td><td>7.81 µS</td></tr>
</table>
</td></tr>
<tr><td>PWDT
Bits 13–12</td><td>Power-Down Timeout—Determines if and when the SDRAM will be placed in a power-down condition. The power-down timeout can be triggered on either the absence of an active bank (PWDT = 00, precharge power-down) or based on a clock count from the last access (PWDT = 10 or 11, active power-down). Count-based timeouts do not force the SDRAM into an idle condition (for example, any active banks remain open). Section 17.4.4, “Power-Down Low-Power Mode,” provides a comprehensive description of this operating mode.</td><td>00 = Disabled
01 = Anytime all banks are inactive (precharge power-down)
10 = 64 clocks after completion of last access (active power-down)
11 = 128 clocks after completion of last access (active power-down)</td></tr>
</table>

Table 17-5. SDRAM 0 Control Register and SDRAM 1 Control Register Description (continued)

Name	Description	Settings
CI Bits 11–10	**Cache Inhibit**—Defines the size of memory, starting from the lowest location on the chip select space, to be cache inhibit. This function is not necessary and should be disabled in a system having MMU which can mark cache inhibit space itself.	00 = Disabled 01 = 1Mbytes 10 = 2Mbytes 11 = 4Mbytes
SCL Bits 9–8	**SDRAM CAS Latency**—Determines the latency between a read command and the availability of data on the bus. This field does not affect the second and subsequent data words in a burst, and has no effect on write cycles.	00 = Reserved 01 = 1 clock 10 = 2 clocks 11 = 3 clocks **Note:** See Figure 17-3 on page 17-13 is the CAS Latency Timing diagram.
Reserved Bit 7	Reserved—This bit is reserved and should read 0.	
SRP Bit 6	**SDRAM Row Precharge Delay**—Determines the number of idle clocks inserted between a precharge command and the next row activate command to the same bank.	0 = 3 clocks inserted 1 = 2 clocks inserted
SRCD Bits 5–4	**SDRAM Row to Column Delay**—Determines the number of clocks inserted between a row activate command and a subsequent read or write command to the same bank. See Figure 17-5.	00 = 4 clocks inserted 01 = 1 clock inserted 10 = 2 clocks inserted 11 = 3 clocks inserted
SRC Bits 3–0	**SDRAM Row Cycle Delay**—Determines the minimum delay (in number of clocks) between a refresh and any subsequent refresh or read/write access. This delay corresponds to the minimum row cycle time captured in the tRC/tRFC memory timing spec. An example timing diagram for SRC = 3 can be found in Figure 17-6. **Note:** The SRC control field is not used to enforce t_{RC} timing for row activate to row activate within the same bank as this is implicitly guaranteed by the sum of $t_{RCD} + t_{CL} + t_{RP}$.	0000 = 10 clocks 0001 = 1 clock 0010 = 2 clocks 0011 = 3 clocks 0100 = 4 clocks 0101 = 5 clocks 0110 = 6 clocks 0111 = 7 clocks 1000 = 8 clocks 1001 = 9 clocks

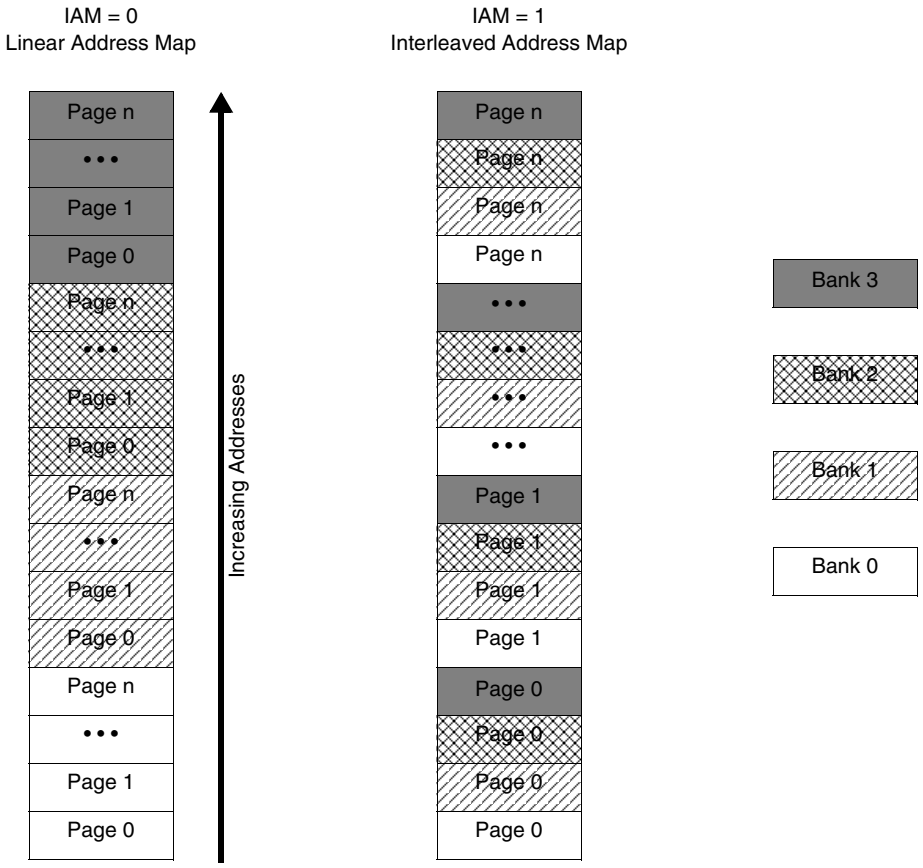


Figure 17-2. Memory Bank Interleaving Options

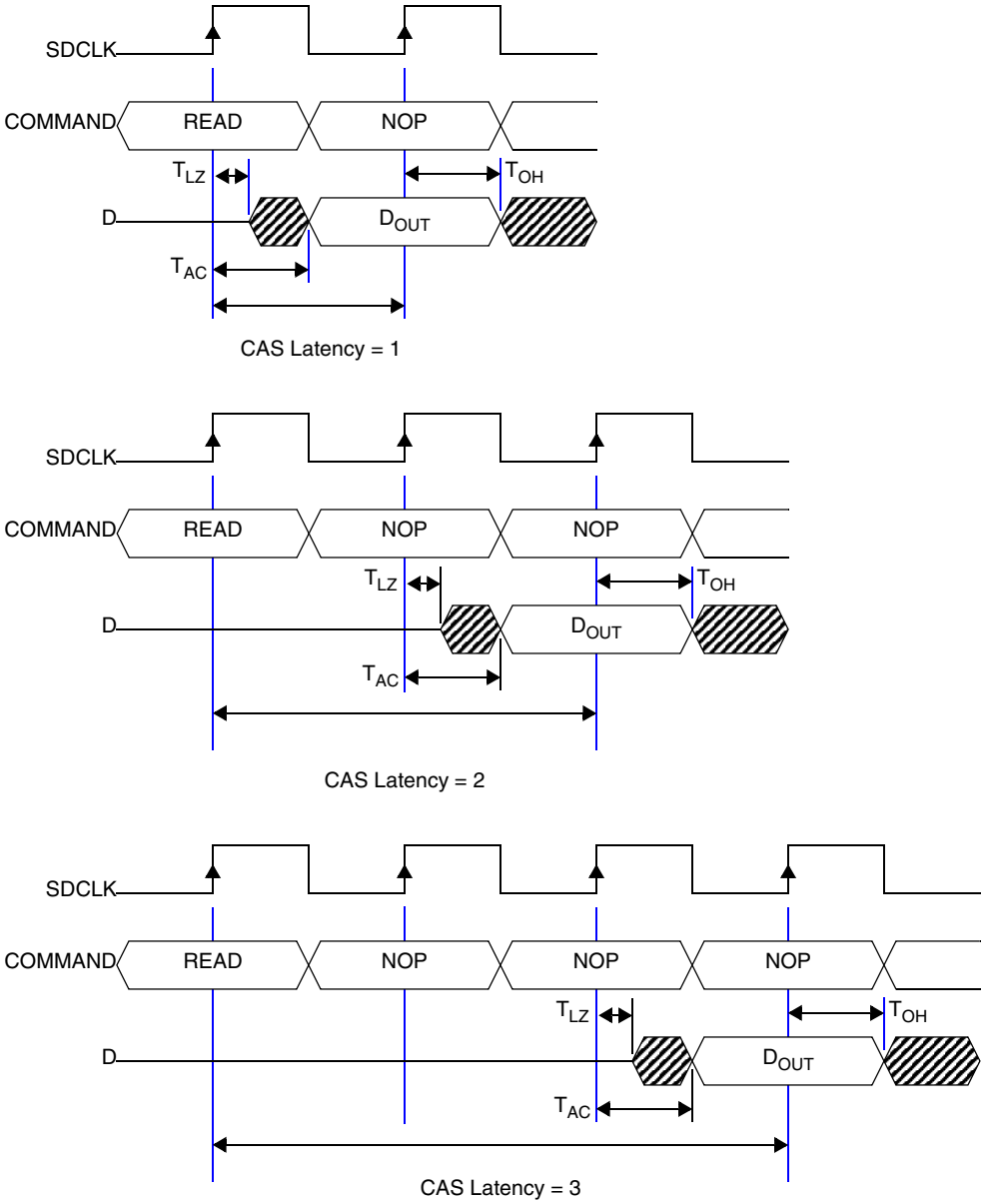


Figure 17-3. CAS Latency Timing

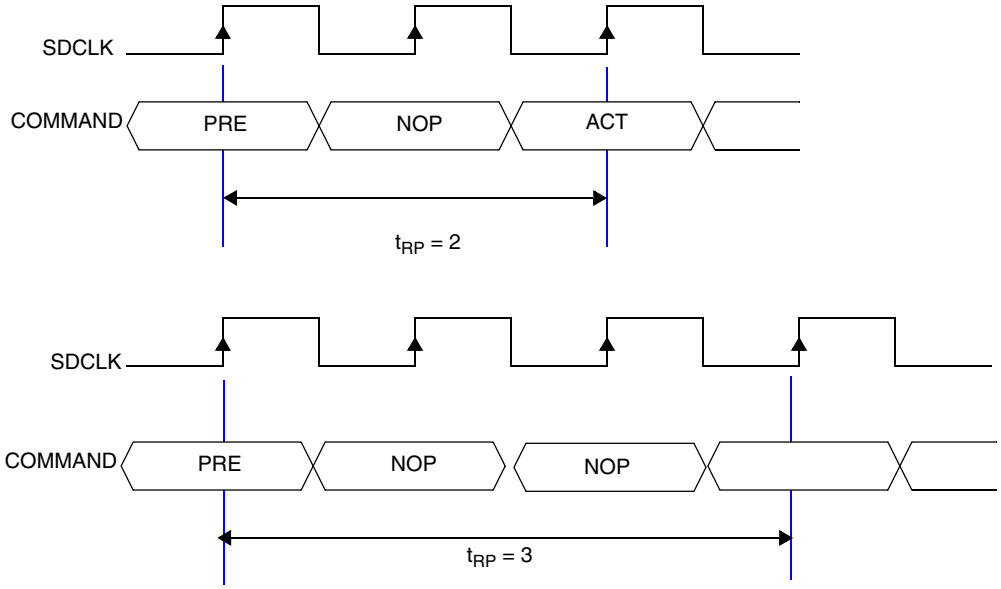


Figure 17-4. Precharge Delay Timing

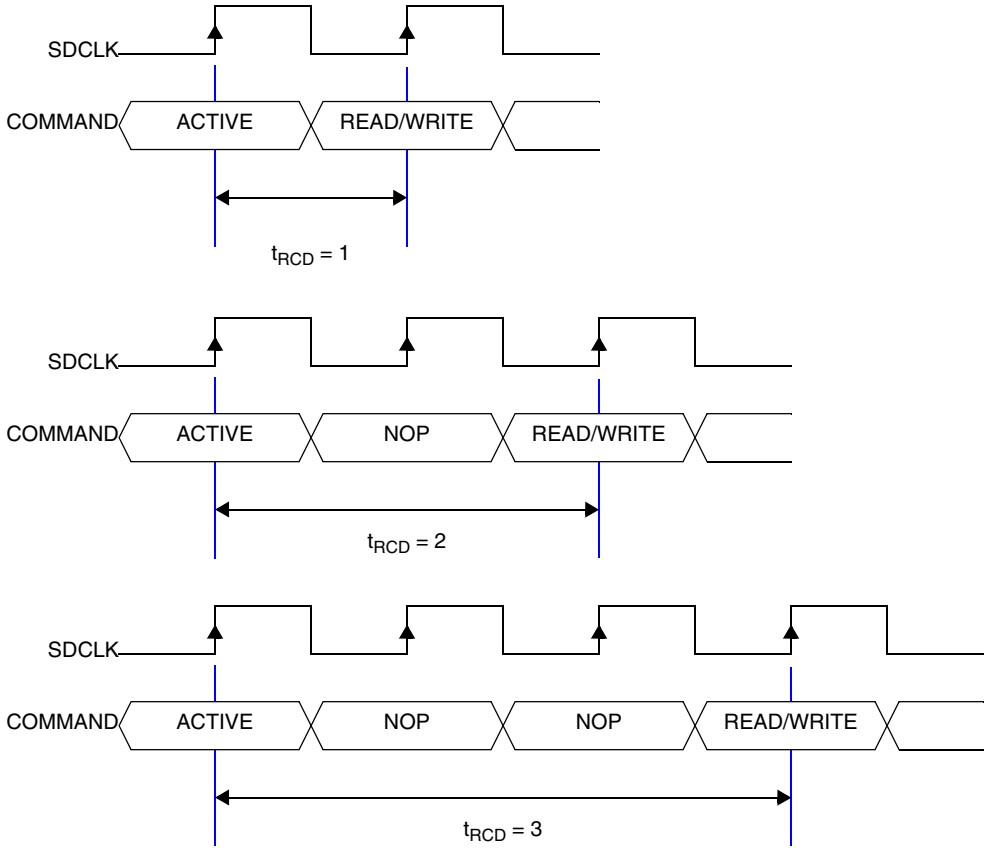


Figure 17-5. Row to Column Delay Timing

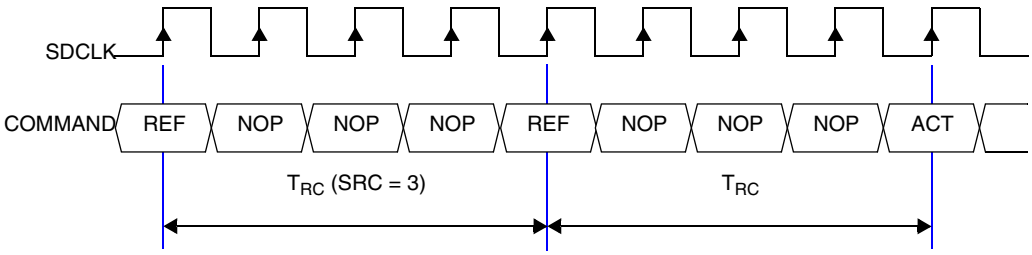


Figure 17-6. Row Cycle Timing

17.2.2 SDRAM Reset Register

The write-only SDRAM Reset Register controls the reset pulse timing.

SDRST — SDRAM Reset Register — Addr 0xDF000018

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RST															
TYPE	w	w	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 17-6. SDRAM Reset Register Description

Name	Description	Settings
RST Bits 31–30	**Software Initiated Local Module Reset Bits**—Generates local module reset to SDRAM controller	00 = No effect to the SDRAMC 01 = One HCLK Cycle Reset Pulse 10 = One HCLK Cycle Reset Pulse 11 = Two HCLK Cycle Reset Pulse
Reserved Bits 29–0	Reserved—These bits are reserved and should read 0.	

17.2.3 Miscellaneous Register

MISCELLANEOUS — Miscellaneous Register — Addr 0xDF000014

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OMA															
TYPE	rw	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																RMA0
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 17-7. Miscellaneous Register Description

Name	Description	Settings
OMA Bit 31	**Multiplexed Address Override**—Enables/Disables the MA0 pinto be a meaningful bit. The multiplexed address, original MA0, is always zero at read access in 16-bit port memory configuration. Working with bit 0 in tandem, this overrides the original MA0 generated by the internal address multiplexer during access to 16-bit device.	0 = Address from internal address multiplexed is routed out to MA0 1 = Force RMA0, bit 0, out to MA0 pin
Reserved Bits 30–1	Reserved—These bits are reserved and should read 0.	
RMA0 Bit 0	**MA0 Replacement**—Contains Value of MA0 when OMA is set.	

17.3 Operating Modes

Each of the SDRAM Controller operating modes are described in this section, including details on basic operation, relationship to SDRAM operating modes, and any special precautions to observe. State and timing diagrams are included where appropriate.

17.3.1 SDRAM Command Encodings

Table 17-8 summarizes the command encodings used by this controller. This command list is a subset of the commands defined by the JEDEC standard. The encodings are based from the view of the SDRAM memory to the controller and therefore use the SDRAM memory signal names.

Table 17-8. SDRAM Command Encoding

Function	Symbol	CKEn-1	CKEn	CS	RAS	CAS	WE	A11	A10	BA[1:0]	A[9:0]
Deselect	DSEL	H	X	H	X	X	X	X	X	X	X
No Operation	NOP	H	X	L	H	H	H	X	X	X	X
Read	READ	H	X	L	H	L	H	V	L	V	V
Write	WRIT	H	X	L	H	L	L	V	L	V	V
Bank Activate	ACT	H	X	L	L	H	H	V	V	V	V
Burst Terminate	TBST	H	X	L	H	H	L	X	X	V	X
Precharge Select Bank	PRE	H	X	L	L	H	L	X	L	V	X
Precharge All Banks	PALL	H	X	L	L	H	L	X	H	X	X
Auto-Refresh	CBR	H	X	L	L	L	H	X	X	X	X
Self Refresh Entry	SLFRSH	H	L	L	L	L	H	X	X	X	X
Self Refresh Exit	SLFRSHX	L	H	H	X	X	X	X	X	X	X
Power-Down Entry	PWRDN	H	L	X	X	X	X	X	X	X	X
Power-Down Exit	PWRDNX	L	H	H	X	X	X	X	X	X	X
Mode Register Set	MRS	H	X	L	L	L	L	L	L	V	V

Assertion of the $\overline{\text{sd\_rst}}$ signal initializes the controller into the idle state. While disabled, the controller remains in the idle state with the internal clocks stopped.

Read/write cycles, refresh and low-power mode requests, and Power-Down time-outs will all trigger transitions out of the idle state. State transitions due to a read or write request depend on the operating mode. Other transitions require the corresponding function to be enabled in the SDCTL register. Some state transitions have been removed from the figure to minimize complexity and allow an easier understanding of the basic controller operation.

The following subsections document the operation of each of the operating modes.

17.3.2 Normal Read/Write Mode

The Normal Read/Write mode (SMODE = 000) is used for general read and write access to the SDRAM, Both single and burst accesses are supported, although burst requests are limited to a length of 8 words (8 half-words to 16-bit memories).

Read or write requests to the SDRAM Controller initiate a check to see whether the page is already open. This check consists of comparing the request address against the last row accessed within the corresponding bank. If the rows are different, that means that a precharge has occurred since the last access, or there has never been an access to the bank. In that case, the access must follow the “off-page” sequence. If the requested row and last row match, the shorter “on-page” access is used.

An off-page sequence must first activate the requested row, an operation which is analogous to a conventional DRAM RAS cycle. An activate cycle is the first operation depicted in Figure 17-7 on page

17-19. During the activate cycle, the appropriate chip select is driven low, the row addresses are placed on the multiplexed address pins, the non-multiplexed addresses are driven to their respective values, write enable is driven high, $\overline{CAS}$ is driven high, and $\overline{RAS}$ is driven low. These latter three pins form the SDRAM command word. The data bus is unused during the activate command.

Once the selected row has been activated, the read operation begins after the row to column delay (t_{RCD}) has been met. This delay is either 2 or 3 clocks, as determined by the SRCD field of the appropriate Control Register. During the read cycle, the chip select is once again asserted, the column addresses are driven onto the multiplexed address bus, the non-multiplexed addresses remain driven to the value presented during the activate cycle, the write enable is driven high, $\overline{RAS}$ is driven high, and $\overline{CAS}$ is driven low. After the CAS latency has expired, data is transferred across the data bus. CAS latency is programmable via the SCL field of the Control Register. As data is being returned across the AHB, transfer acknowledge is asserted back to the CPU indicating that the CPU should latch data. While data is still on the bus, the SDRAM Controller must begin monitoring transfer request since the CPU is free to issue the next bus request on the same edge that data is being latched.

Data transfers can be either single operand or a burst of up to 8 operands. Burst requests are designated as such by the $\overline{p\_burst}$ attribute. This AHB signal is asserted low for all but the last operand of a burst transfer. Non-burst transfers do not assert the signal.

SDRAM memories assume that all transfers are burst transfers unless terminated early. Burst transfers can be terminated by a variety of mechanisms: another read or write cycle, a precharge operation, or through a burst terminate command. Burst terminate commands are the general mechanism used by the SDRAM Controller for early burst termination, The burst terminate command is subject to the CAS latency and must be pipelined similar to the Read command, as shown in Figure 17-7 and Figure 17-8 on page 17-20.

SDRAM write cycles are different than read cycles in one important aspect. Whereas read data was delayed by the CAS latency, write data has no delay and is supplied at the same time as the Write command. Figure 17-13 illustrates an off-page write cycle followed by an on-page write cycle. Note that the write data is driven during the same clock that the Write command is issued. A Burst Terminate command cancels the burst operation, but again without the CAS latency. When the SDRAM Controller issues a burst write sequence, it also supplies the corresponding column address and write command with each data word. Therefore when programming the SDRAM memory mode register the write burst length must be set to single location writes.

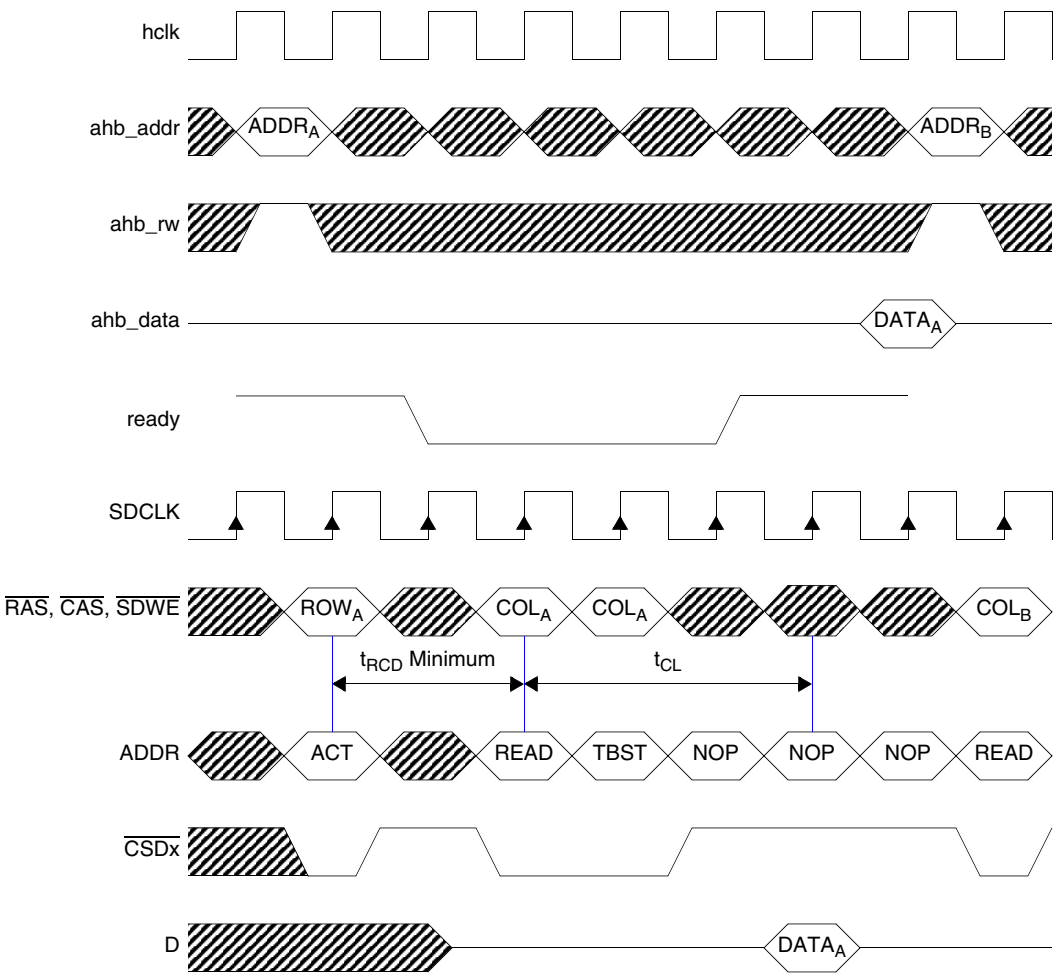


Figure 17-7. Off-Page Single Read Timing Diagram (32-Bit Memory)

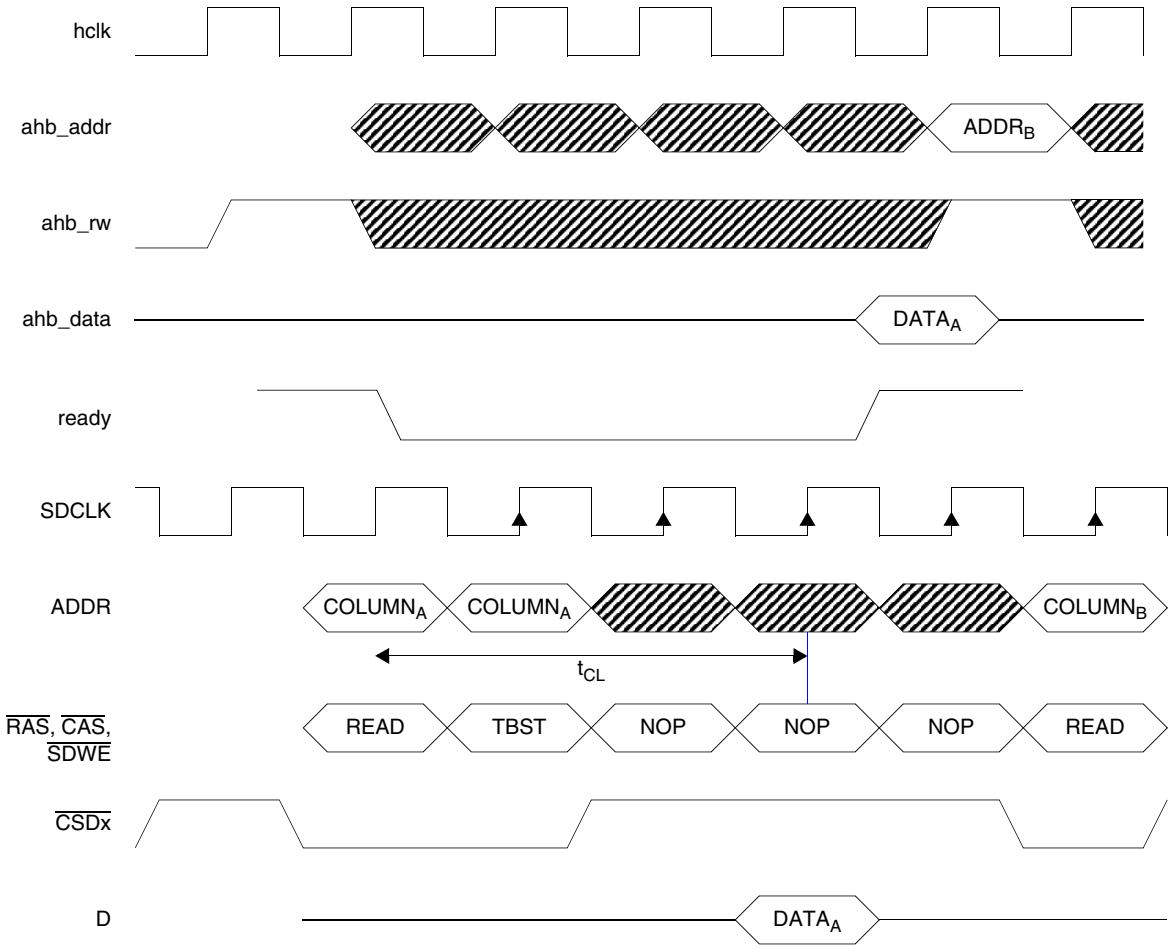


Figure 17-8. On-Page Single Read Timing Diagram (32-Bit Memory)

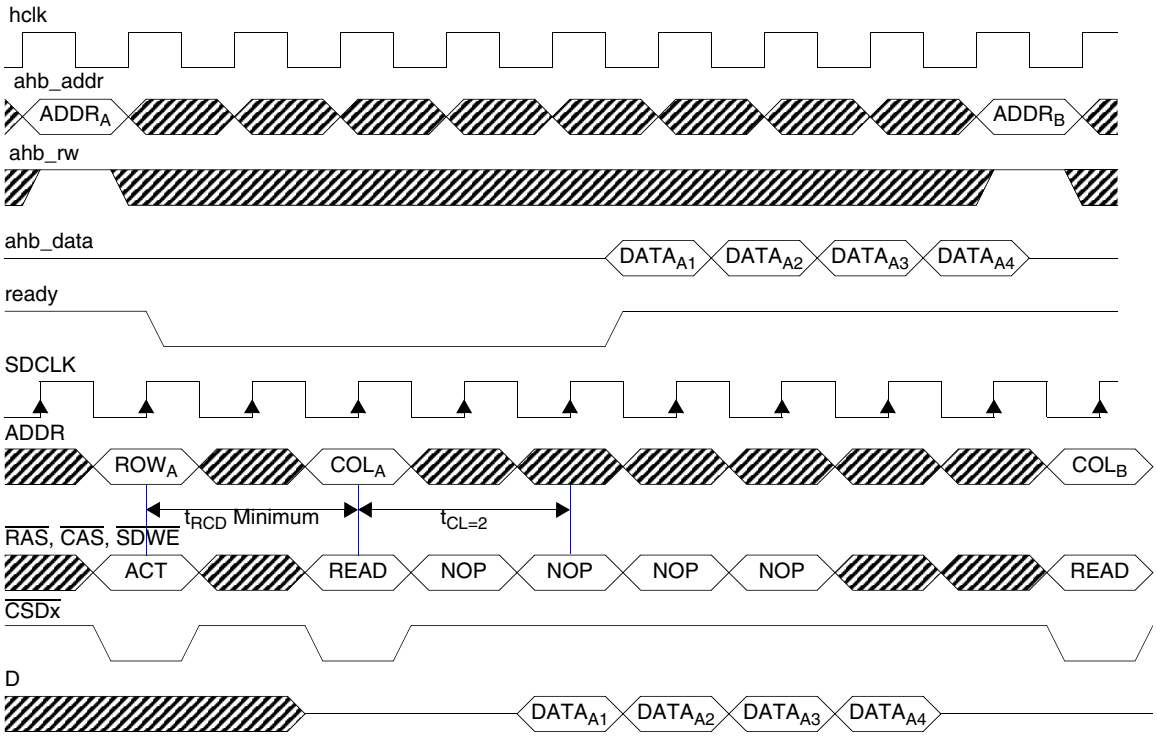


Figure 17-9. Off-Page Burst Read Timing Diagram (32-Bit Memory)

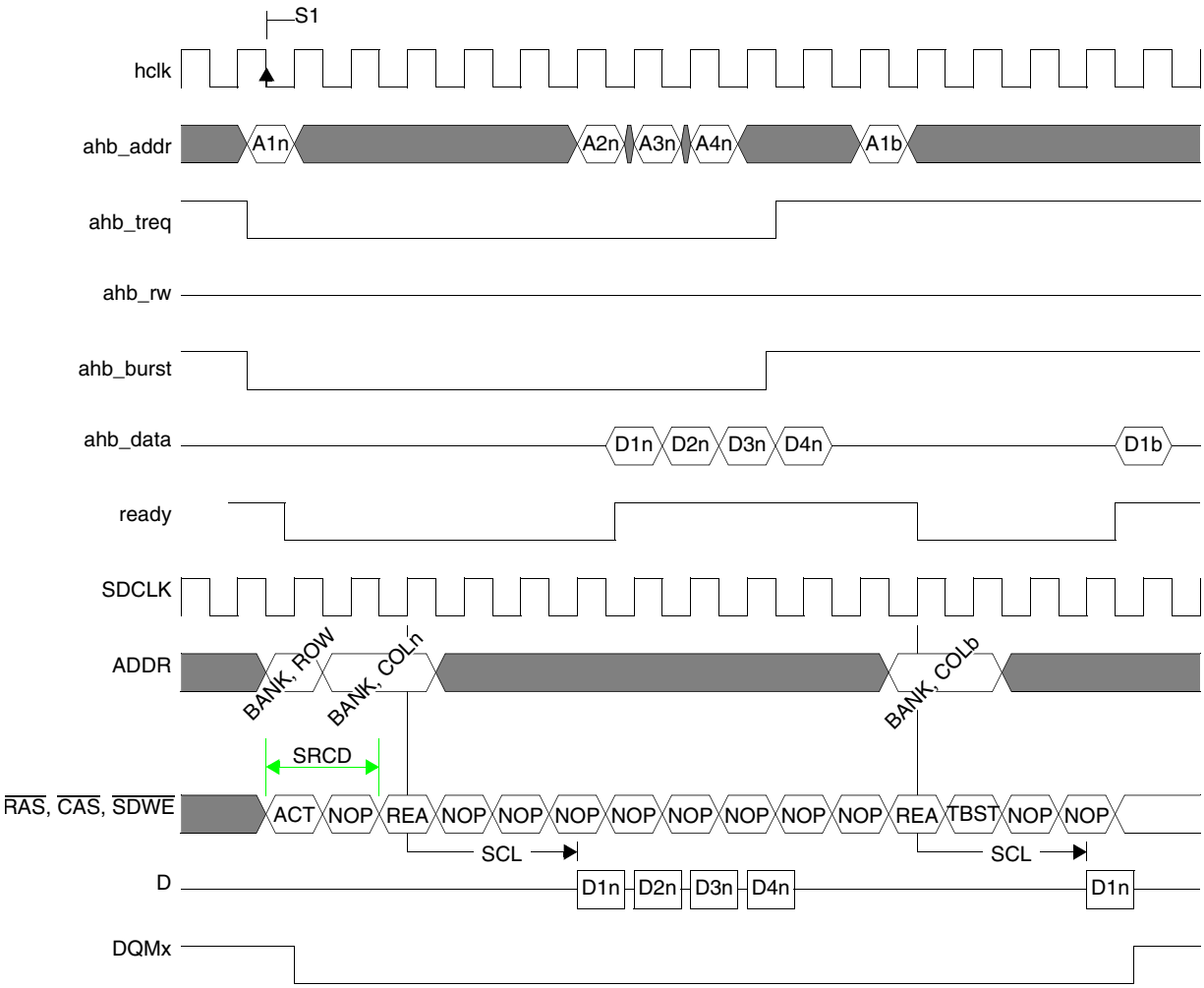


Figure 17-10. On-Page Burst Read Timing Diagram (32-Bit Memory)

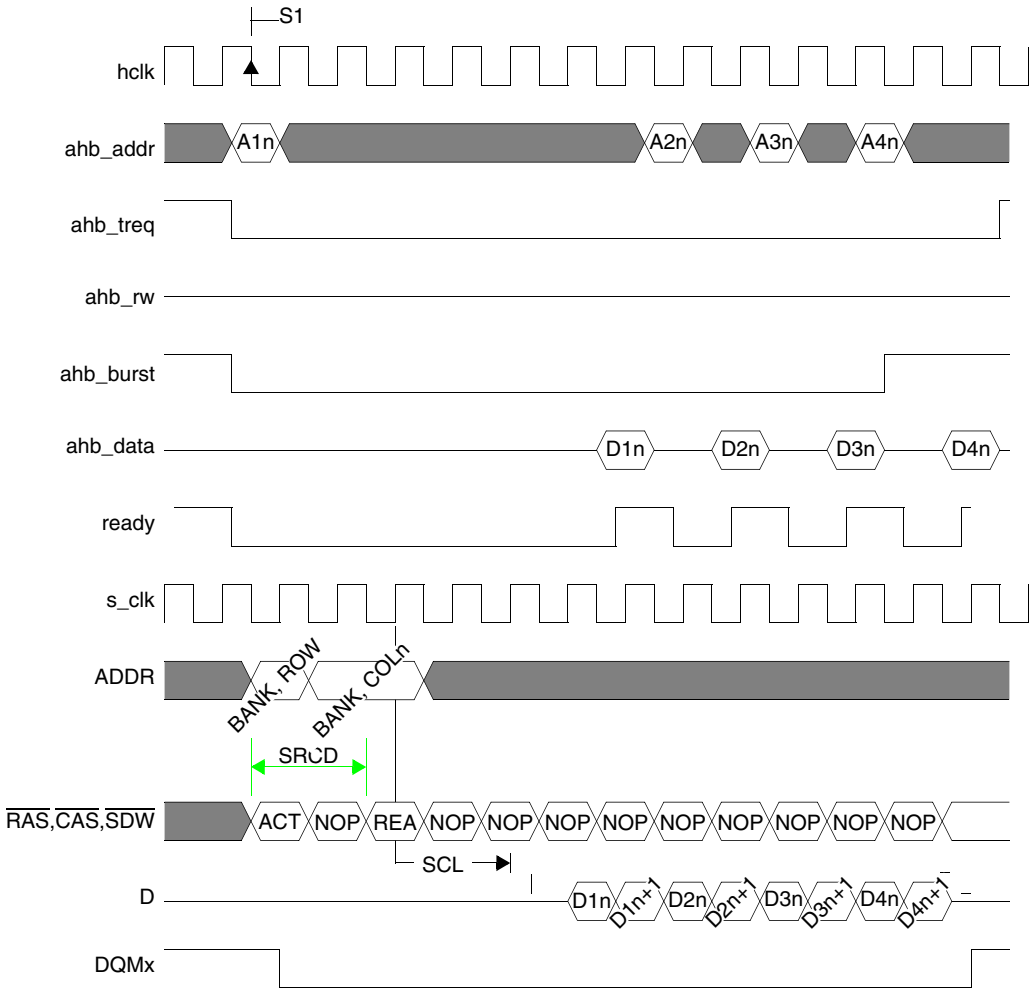


Figure 17-11. Off-Page Burst Read Timing Diagram (16-bit Memory)

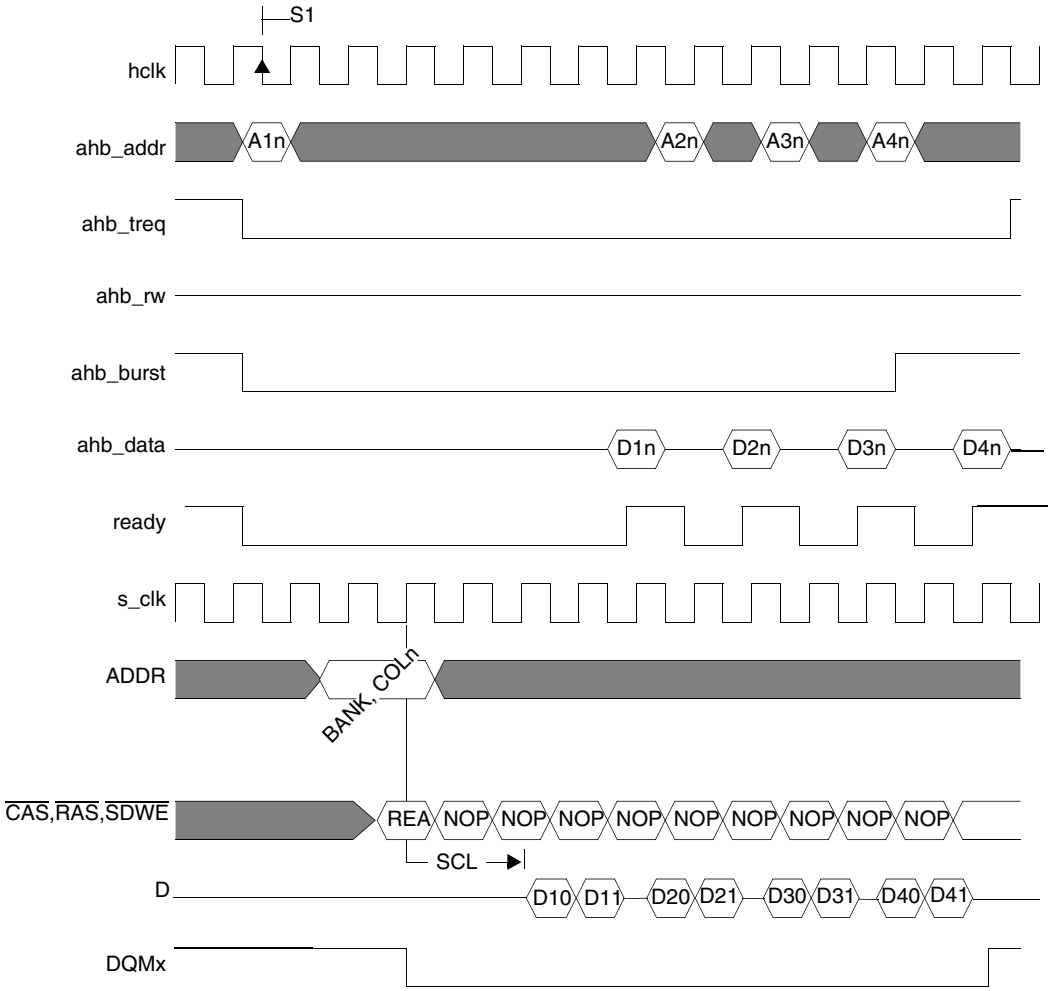


Figure 17-12. On-Page Burst Read Timing Diagram (16-Bit Memory)

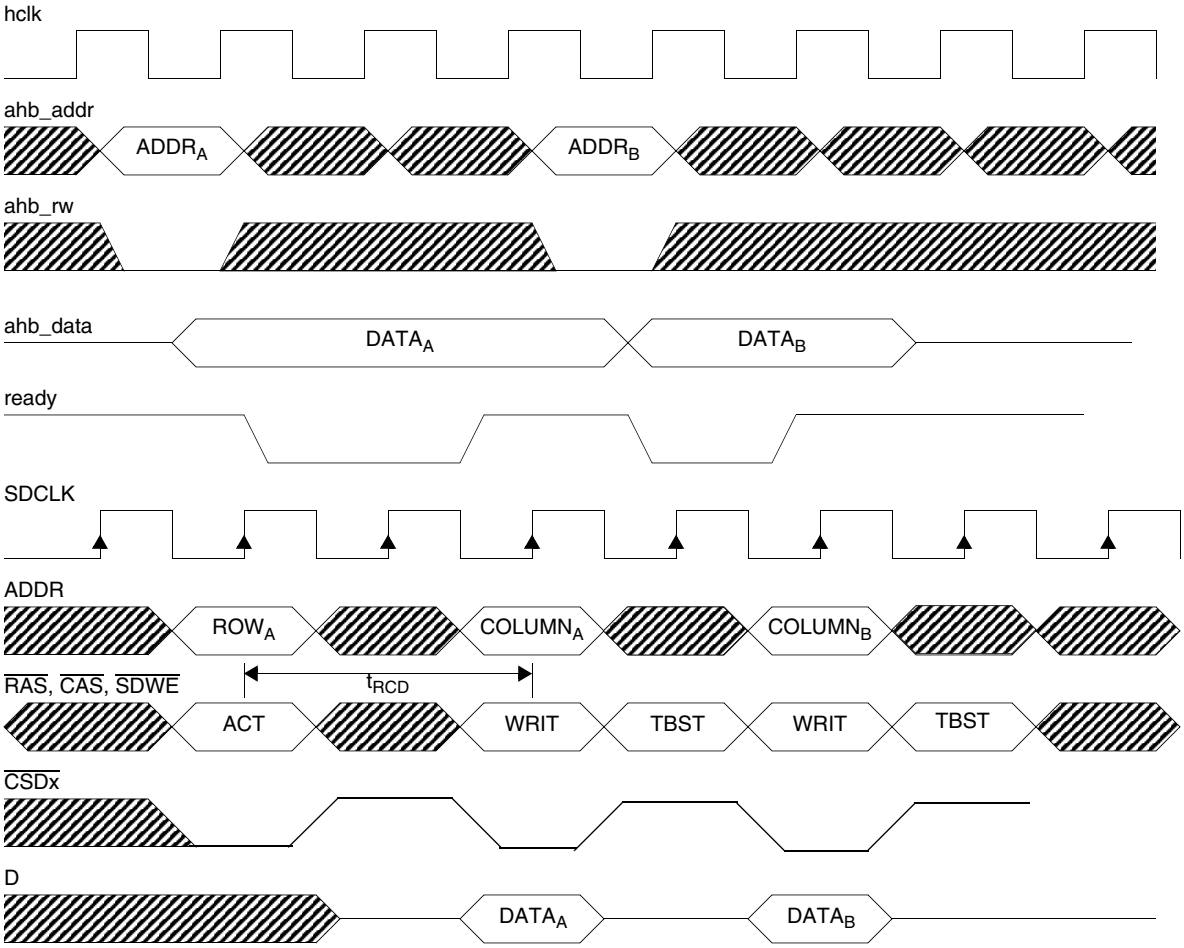


Figure 17-13. Off-Page Write Followed by On-Page Write Timing Diagram

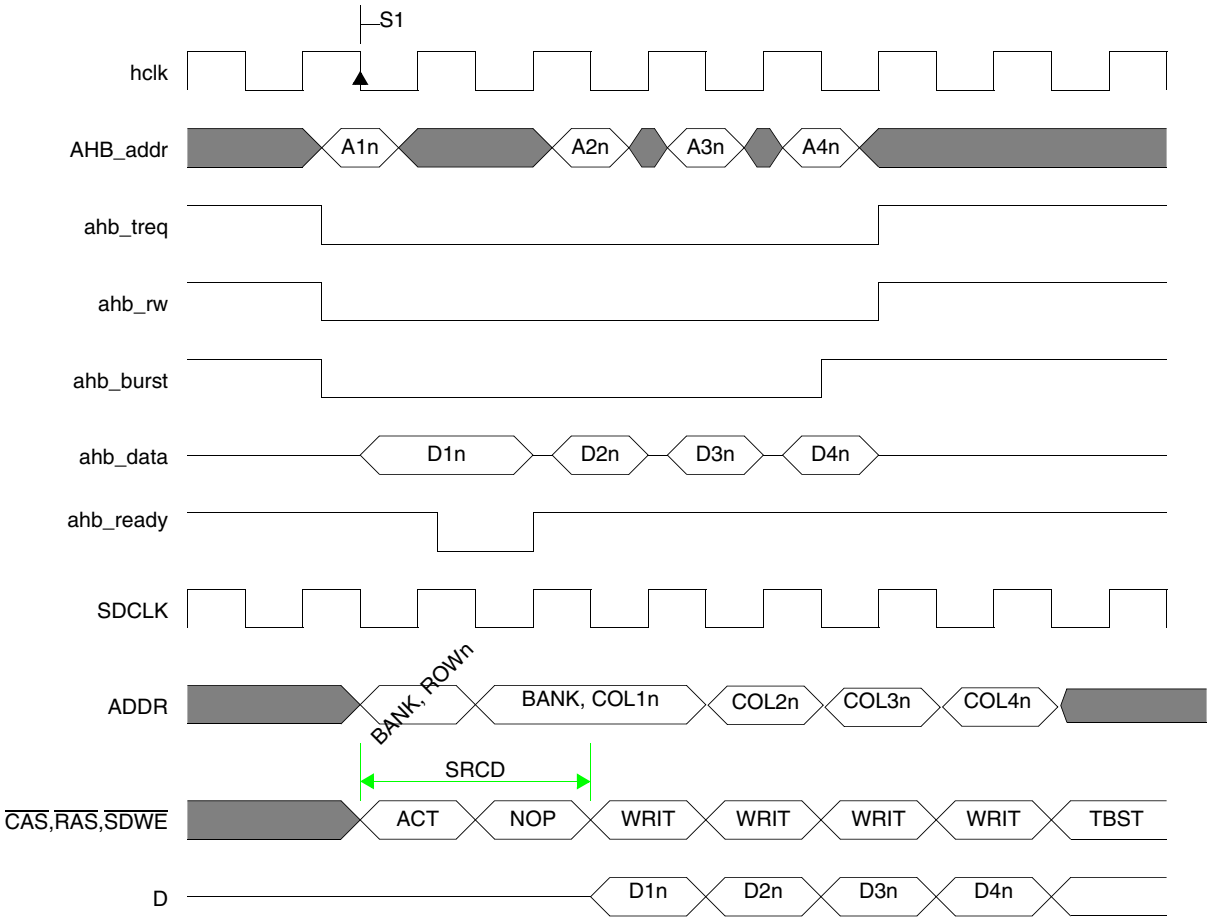


Figure 17-14. Off-Page Burst Write Timing Diagram

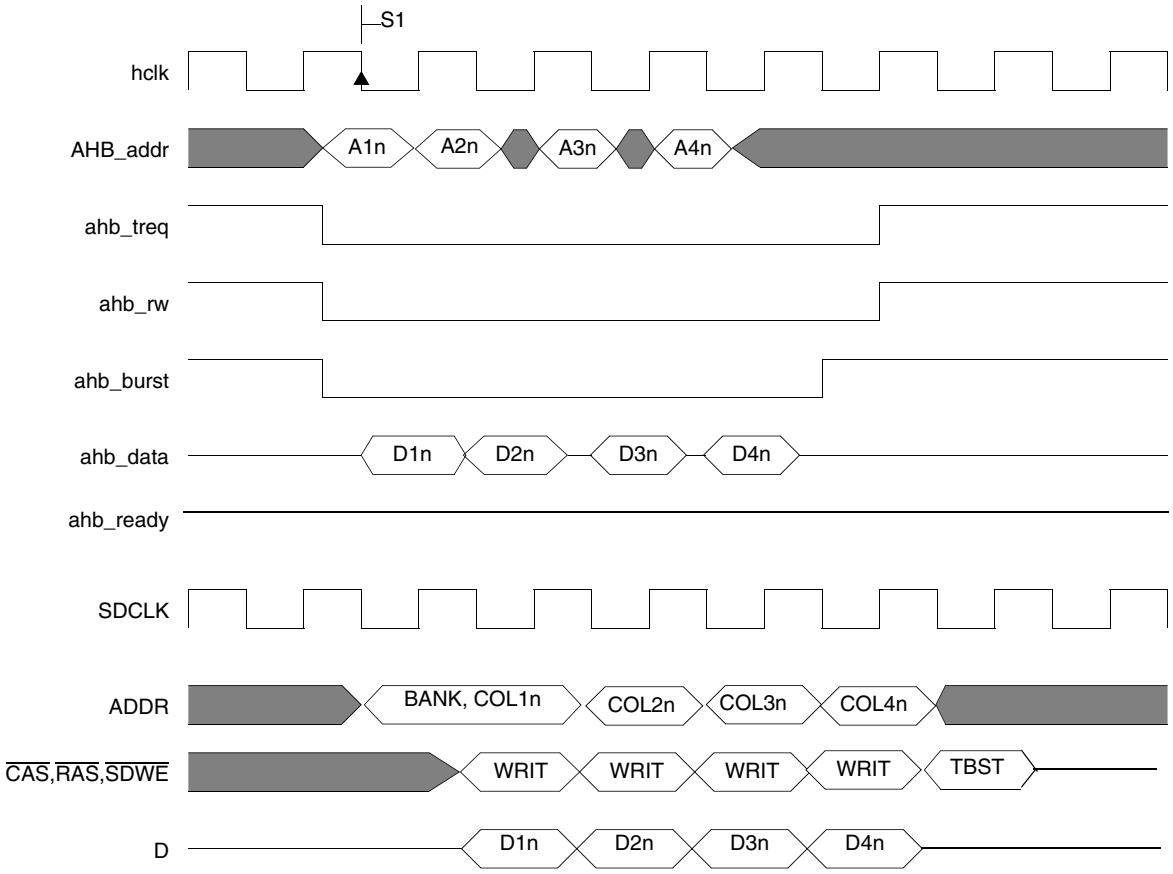


Figure 17-15. On-Page Burst Write Timing Diagram

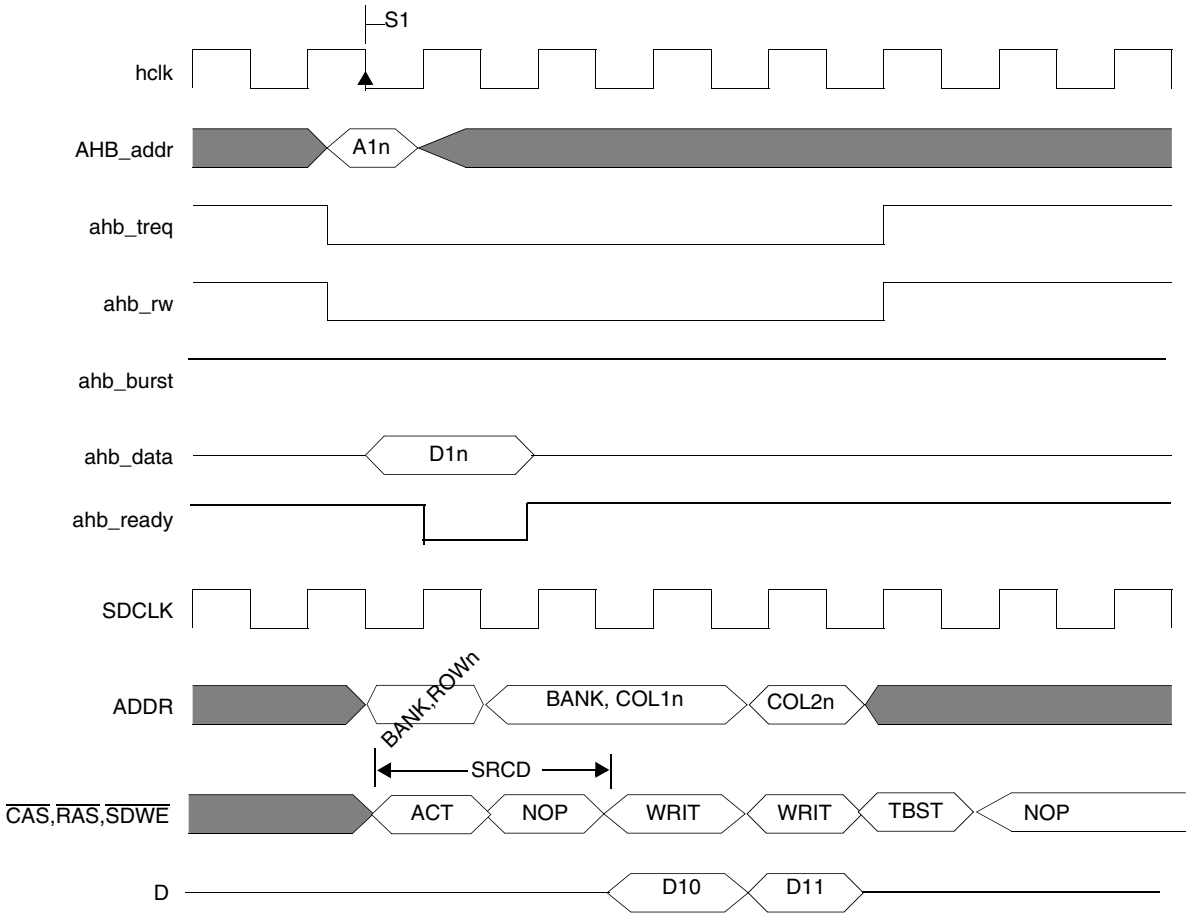


Figure 17-16. Off-Page Write Timing Diagram (16-Bit Memory)

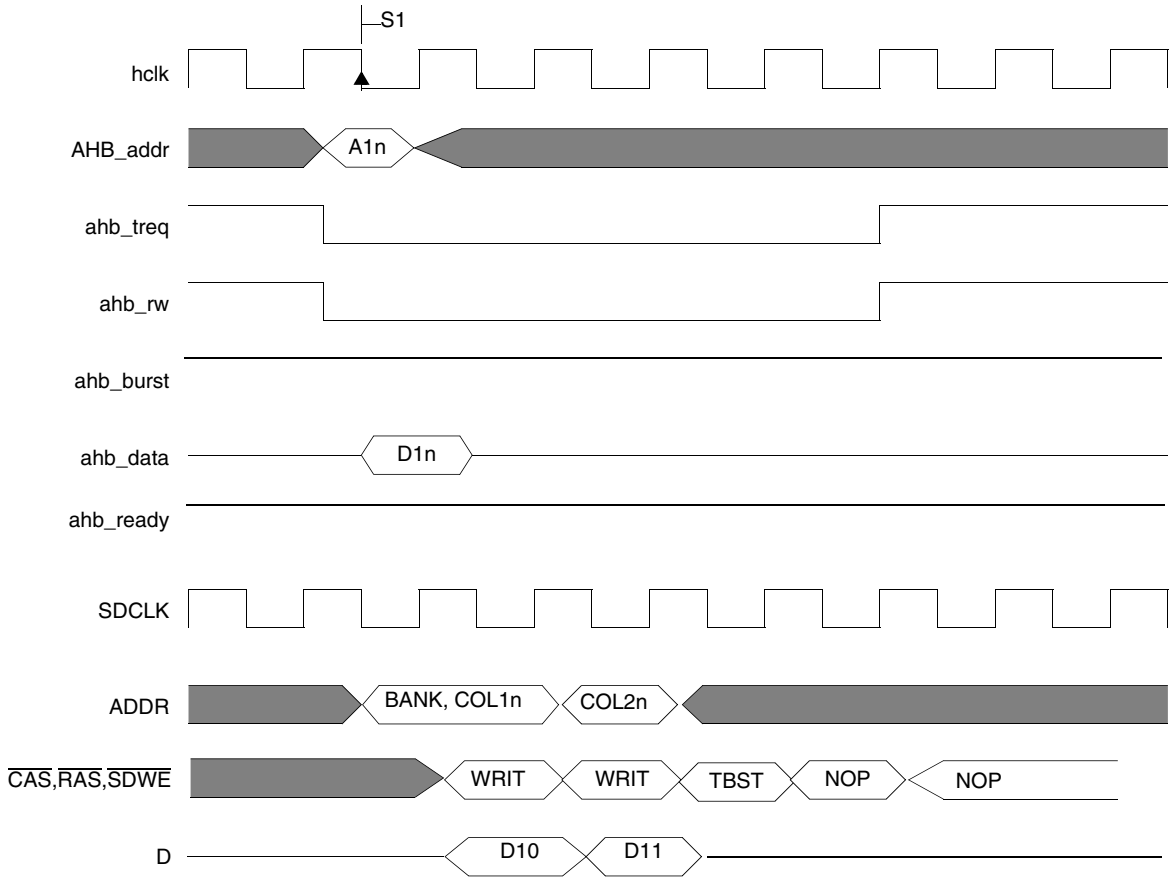


Figure 17-17. On-Page Write Timing Diagram (16-Bit Memory)

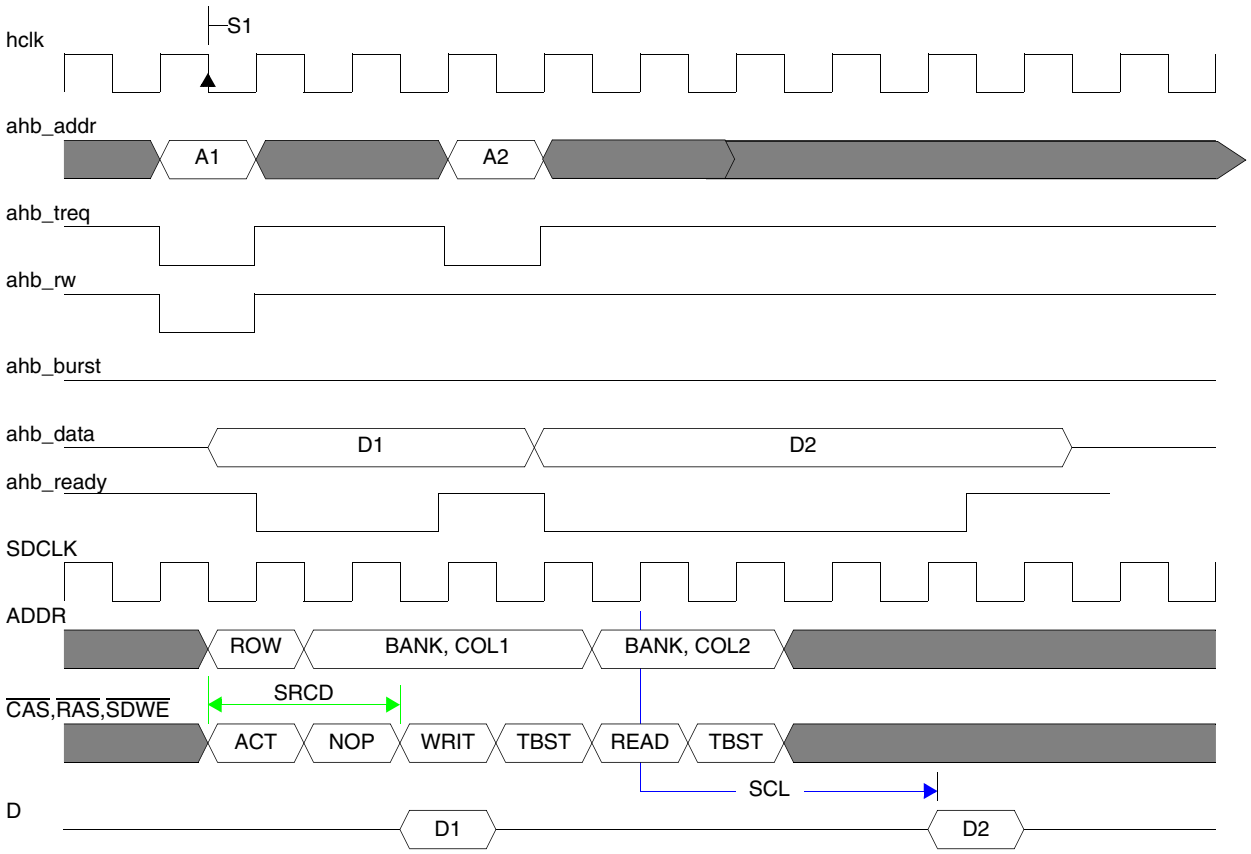


Figure 17-18. Single Write Followed by On-Page Read Timing Diagram

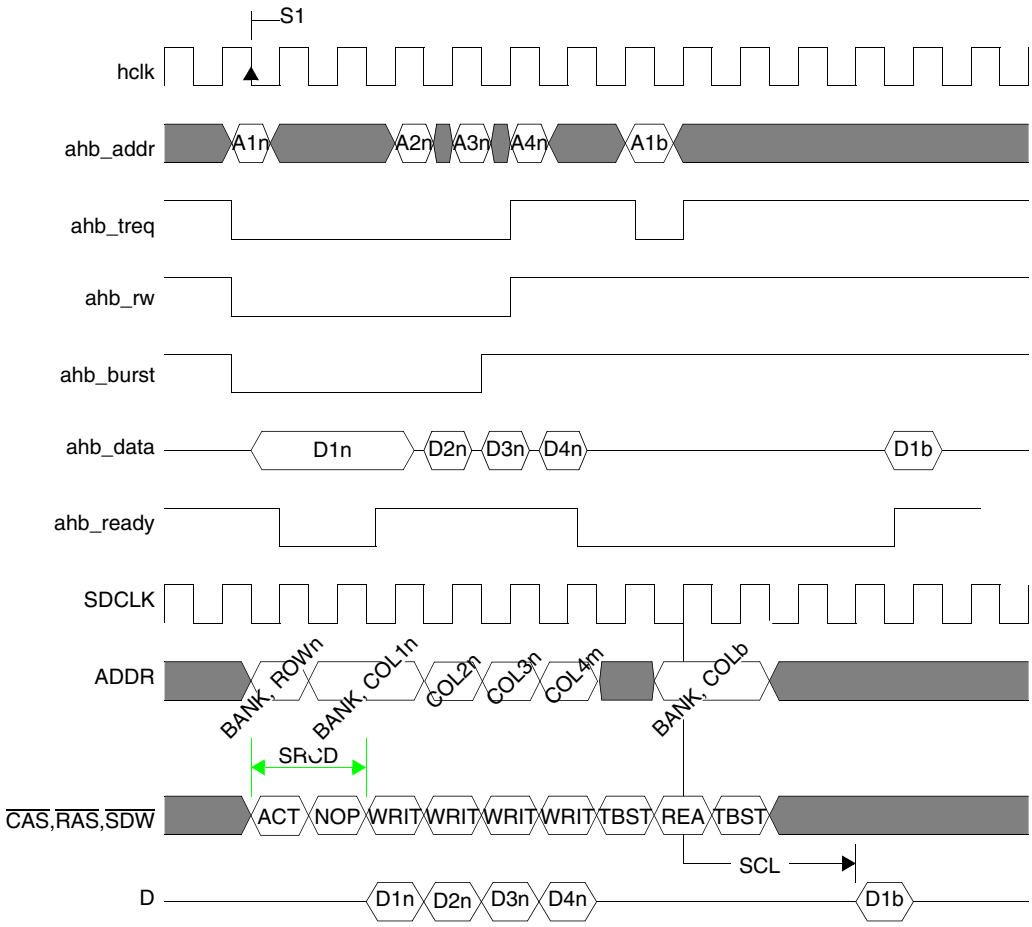


Figure 17-19. Burst Write Followed by On-Page Read Timing Diagram

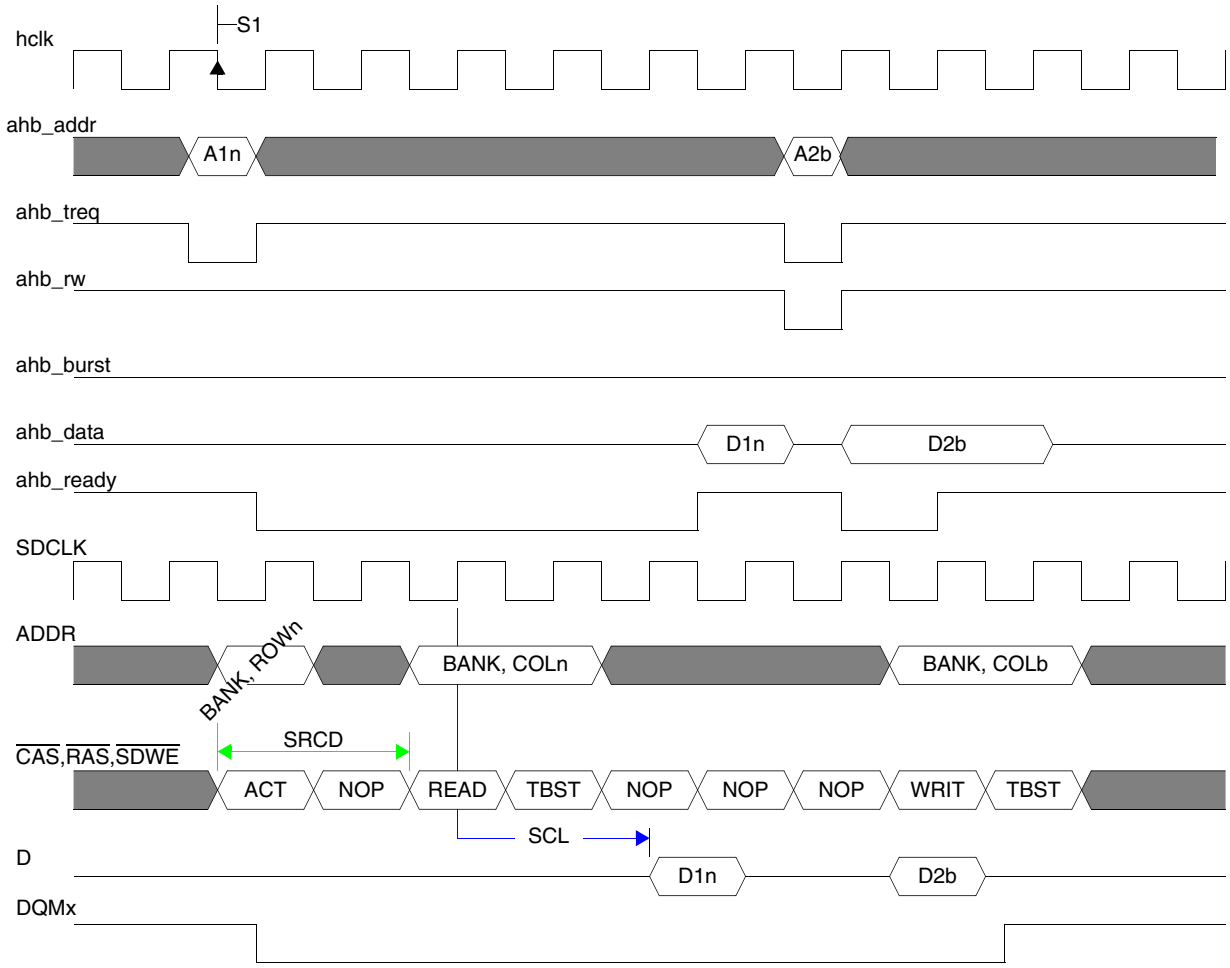


Figure 17-20. Single Read Followed by On-Page Write Timing Diagram

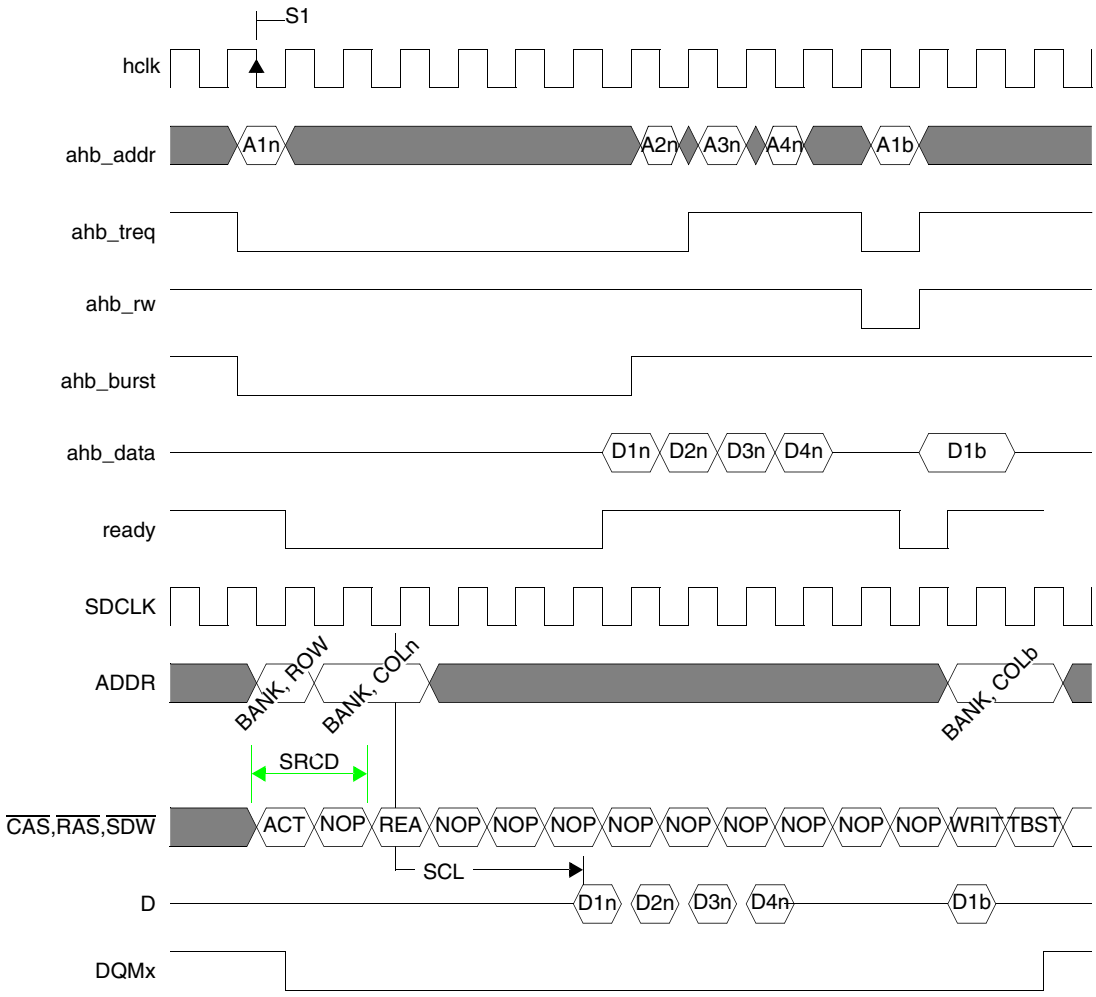


Figure 17-21. Burst Read Followed by On-Page Write Timing Diagram

17.3.3 Precharge Command Mode

The Precharge Command Mode (SMODE = 001) is used during SDRAM device initialization, and to manually deactivate an/all active bank/s. While in this mode, an access (either read or write) to the SDRAM address space will generate a precharge command cycle. SDRAM address bit A10 determines whether a single bank, or all banks, are precharged by the command. (See Figure 17-22). Accessing an address with the SDRAM address A10 low will precharge only the bank selected by the bank address. Conversely, accesses with A10 high will precharge all banks regardless of the bank address. Note that A10 is the SDRAM pin, not the ARM926EJ-S processor's address. Translation of the SDRAM A10 to the corresponding ARM926EJ-S processor's address is dependent on the memory configuration. The precharge command access is two clocks in length on the AHB, and one cycle to the SDRAM.

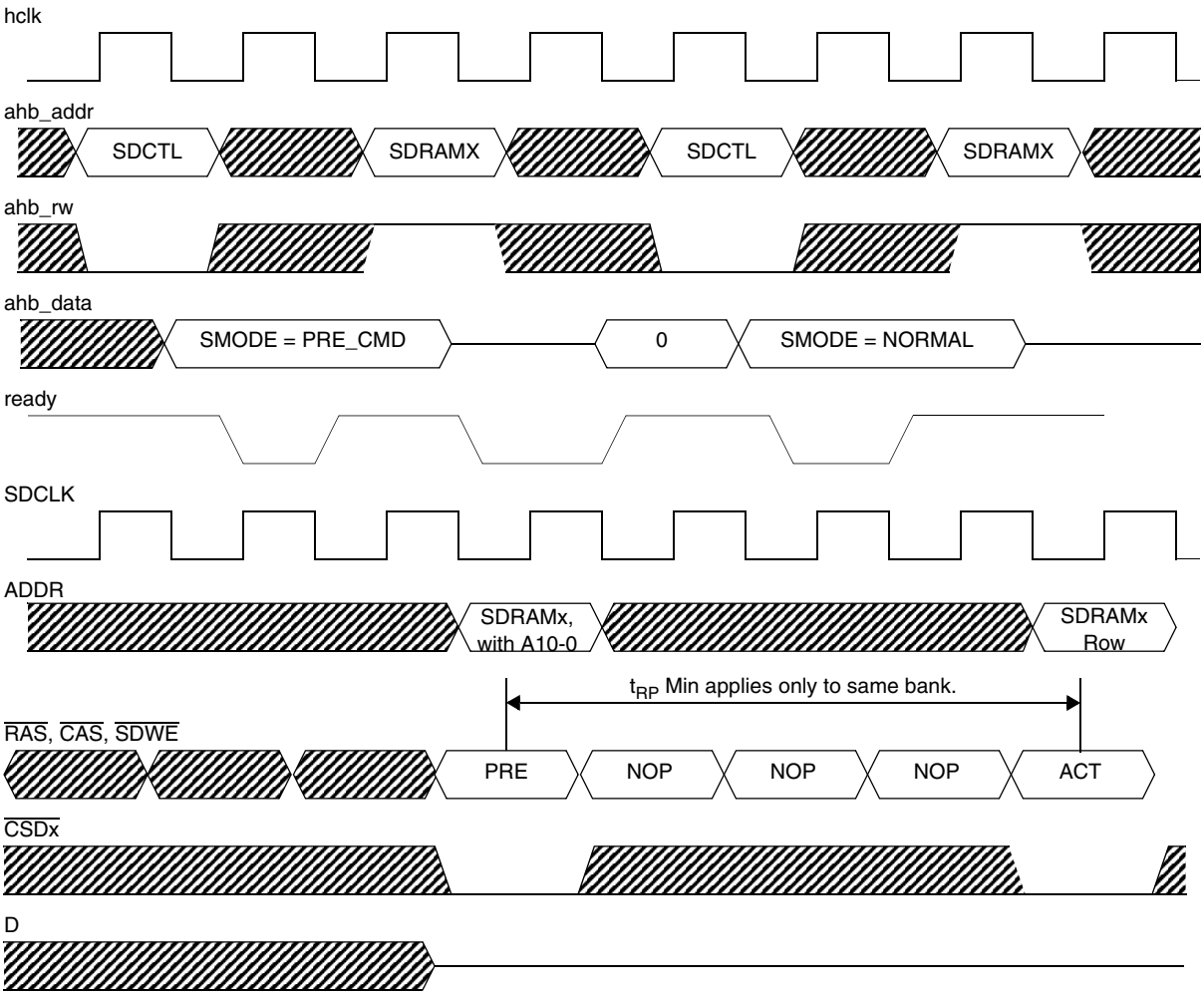


Figure 17-22. Precharge Bank Timing Diagram

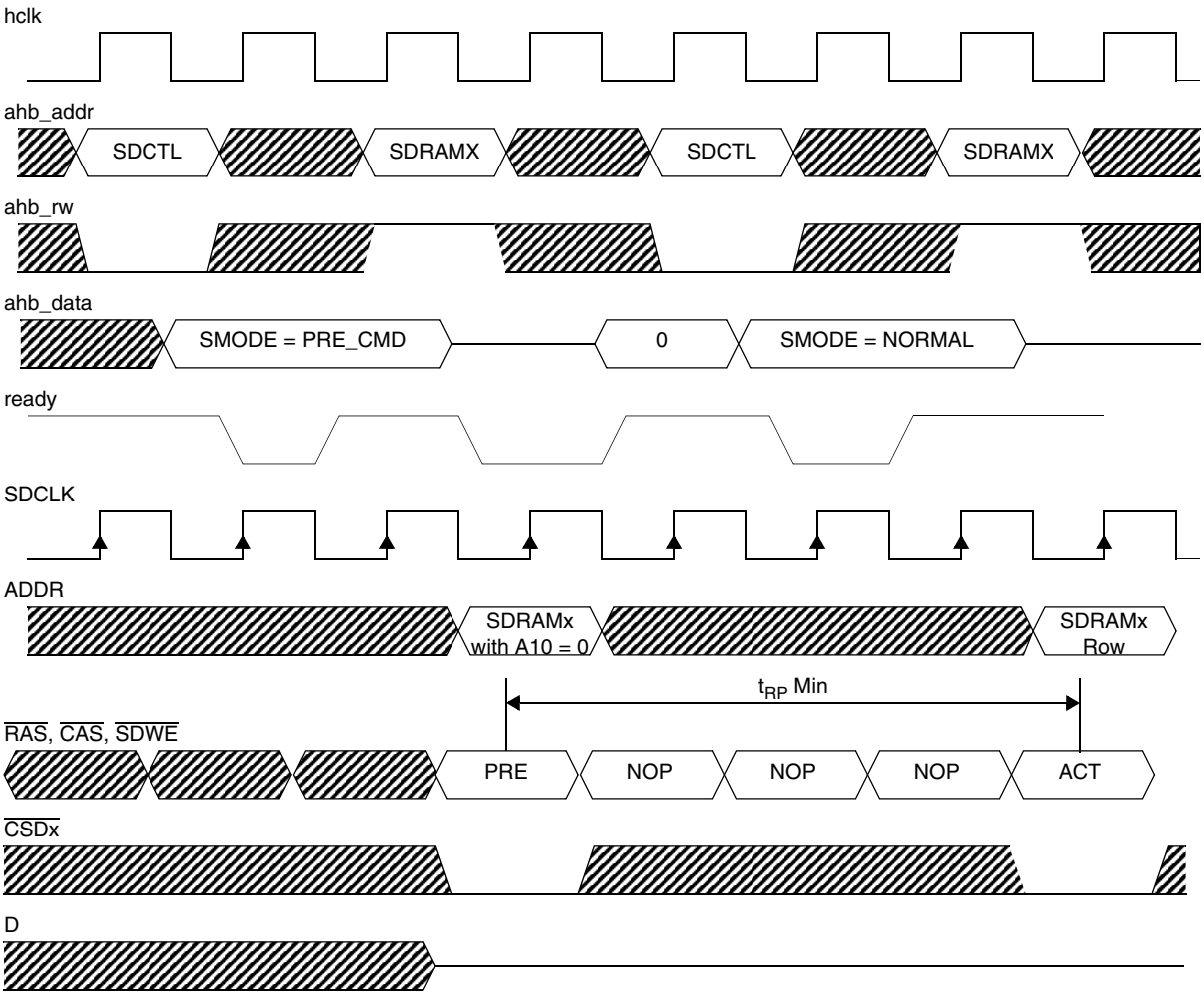


Figure 17-23. Precharge All Timing Diagram

17.3.4 Auto-Refresh Mode

The Auto-Refresh Mode (SMODE = 010) is used to manually request SDRAM refresh cycles. It is normally used only during device initialization, since the SDRAM Controller will automatically generate refresh cycles when properly configured. The auto-refresh command refreshes all banks in the device, so the address supplied during the refresh command only needs to specify the correct SDRAM device. The lower address lines are ignored. Either a read or write cycle may be used. If a write is used, the data will be ignored and the external data bus will not be driven. The cycle will be 2 clocks on the AHB and a single clock to the SDRAM device.

The SDRAM Controller guarantees that the SDRAM is in the idle state before the auto-refresh command is given. If one or more rows are active, a precharge-all command will be issued prior to the auto-refresh command. The precharge-all command adds one additional clock to the access time.

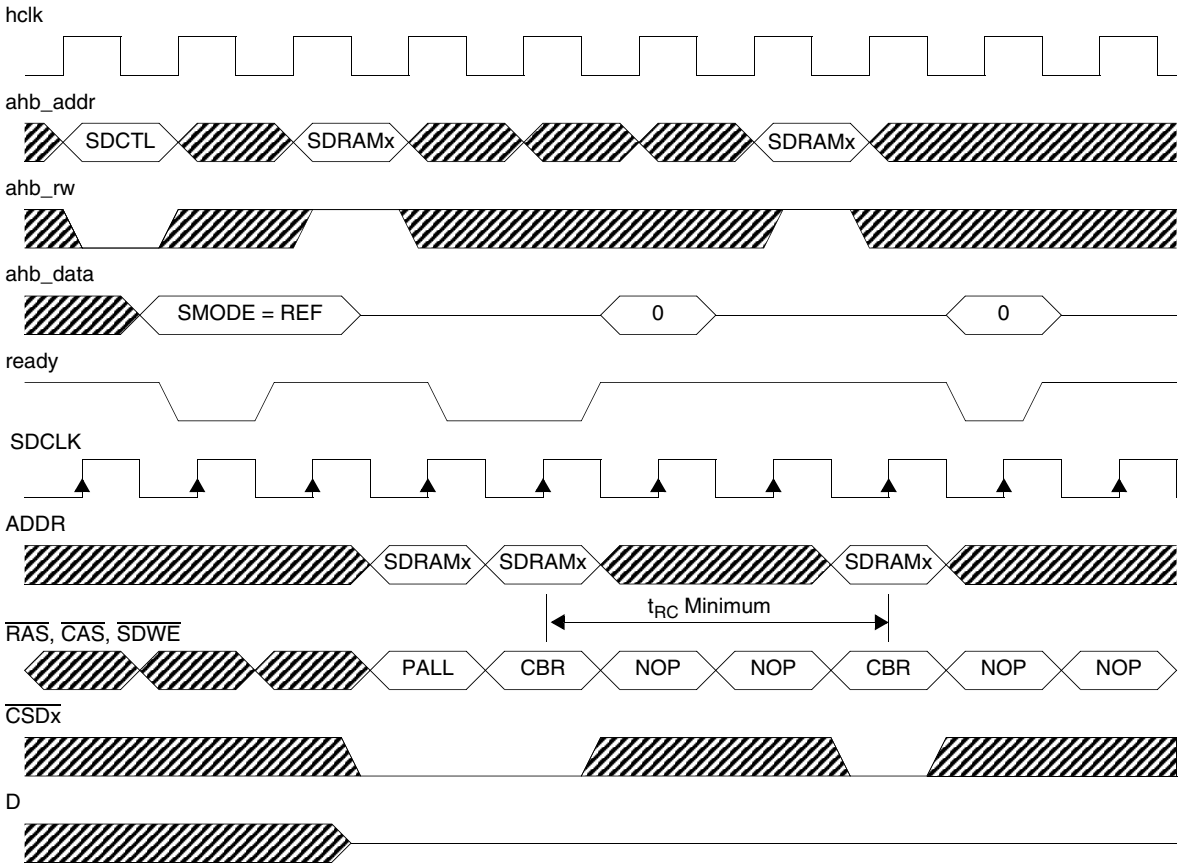


Figure 17-24. Software Initiated Auto-Refresh Timing Diagram

NOTE

SDRAM devices require a minimum delay of t_{RC} between refresh cycles. The SDRAM Controller incorporates a timer to guarantee this timing is met. The timer is user configurable through the SRC field in the SDCTLx register.

17.3.5 Set Mode Register Mode

The Set Mode Register mode (SMODE = 011) is used to program the SDRAM mode register. This mode differs from normal SDRAM write cycles because the data to be written is transferred across the address bus. Reads of the mode register are not allowed.

Either a read or write cycle may be used for this transfer. In the case of a write, the AHB data will be ignored and the external data bus will not be driven. The row and bank address signals are used to transfer the data. The cycle will be 2 clocks on the AHB and a single clock to the SDRAM device.

Figure 17-25 and Figure 17-26 illustrate the bus sequence for a mode register set operation. Mode register set commands must be issued while the SDRAM is idle.

The SDRAM Controller does not guarantee that the SDRAMs have returned to the idle state before issuing the mode register set command. Therefore, software must generate a precharge all sequence before issuing

the mode register set command if there is any possibility that one or more banks could be active. Also keep in mind that the row cycle time (t_{RC}) must be met before the mode register set command is issued.

Section 17.5.4, “Mode Register Programming,” provides a detailed example of the mode register data value calculation and mapping to the ARM926EJ-S processor’s address.

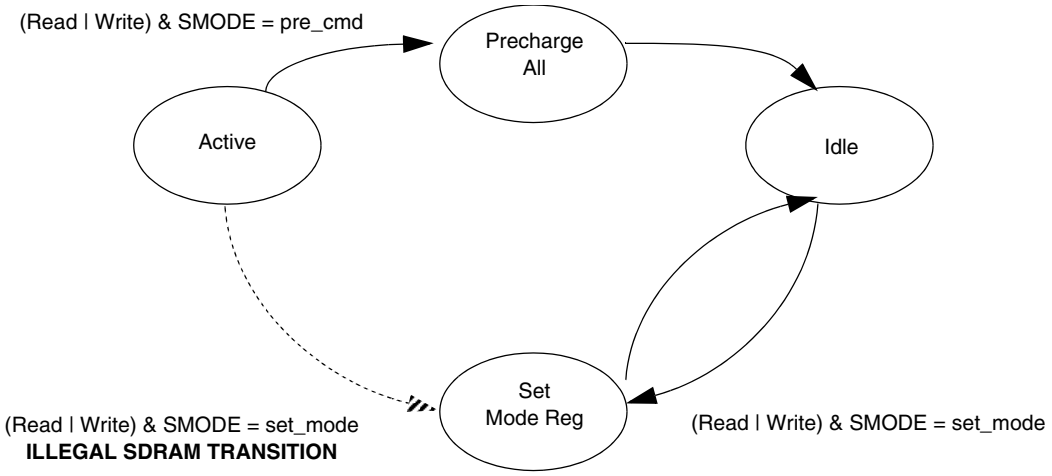


Figure 17-25. Set Mode Register State Diagram

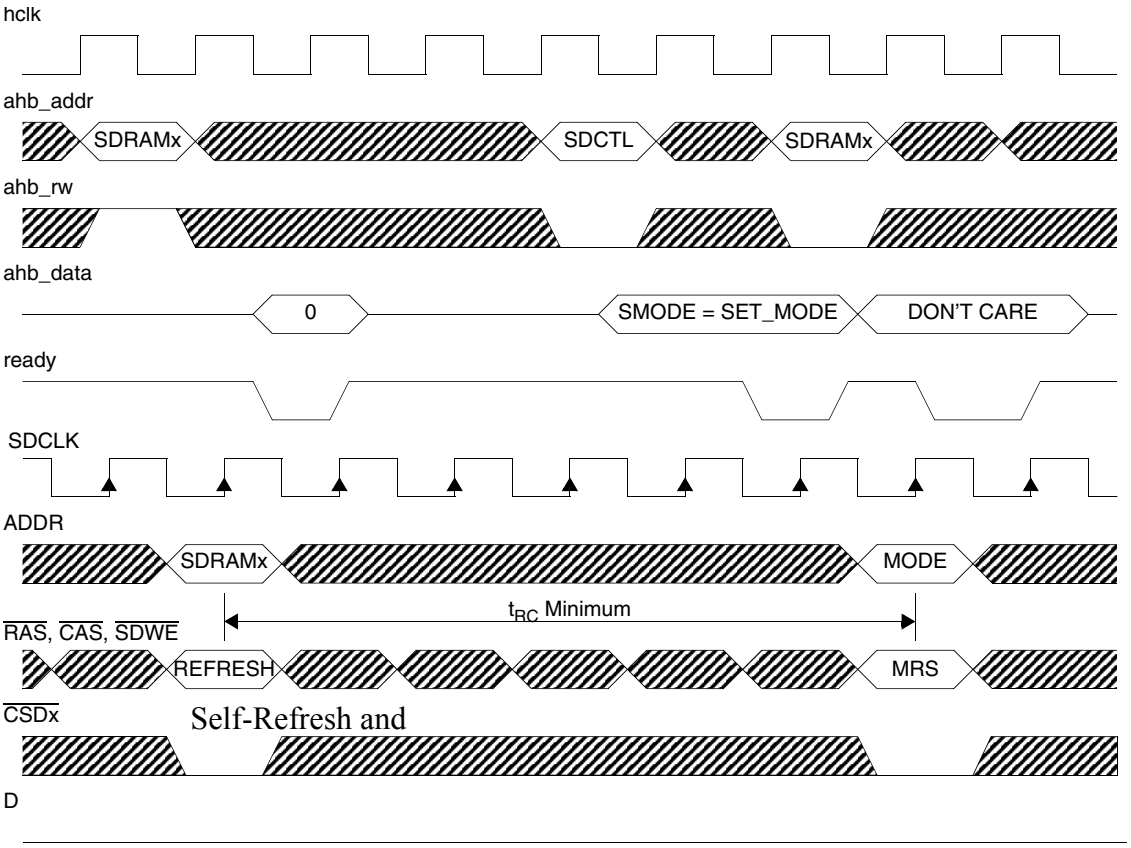


Figure 17-26. Set Mode Register Timing Diagram

17.3.6 Manual Self Refresh Mode

This mode allows the software to control a Self refresh mode entry of the external SDRAM if refresh has been enabled. When this mode is selected (SMODE=100) and refresh is enabled the controller will finish any active accesses and send a self refresh command to the external memory. No access is allowed while SMODE bits are set in the manual self refresh mode. If refresh has not been enabled, the SDRAM Controller places the memories in a lower power consumption mode known as Powerdown.

When a different mode is selected on the SMODE bits, the controller will take the external SDRAM out of self refresh mode and will begin issuing auto refresh cycles (if refresh has been enabled).

After exiting manual self refresh mode, the software should wait a tRC period of time before issuing "normal" accesses to the SDRAM.

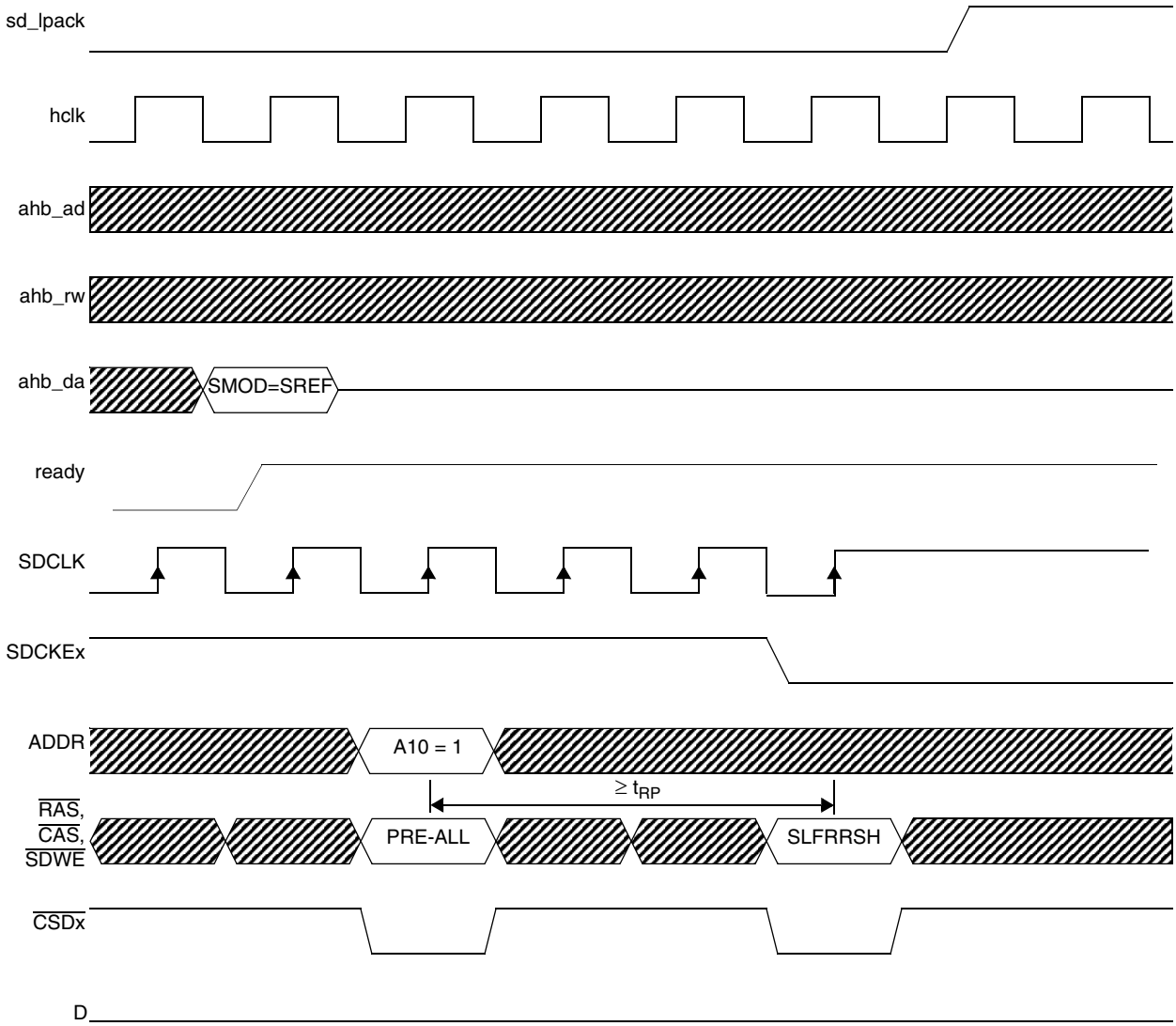


Figure 17-27. Software Initiated Self-Refresh Entry Timing Diagram

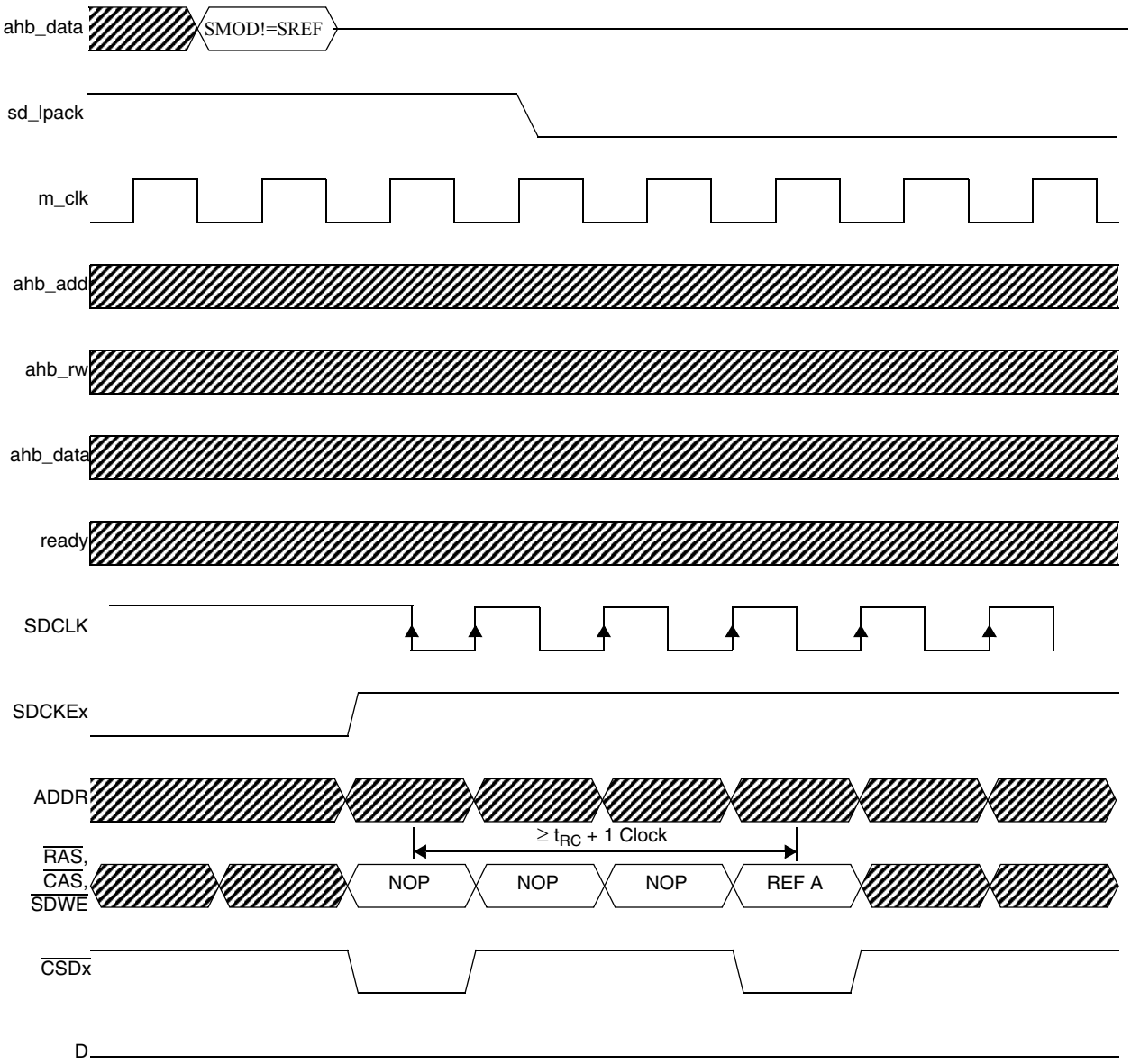


Figure 17-28. Software Initiated Self-Refresh Exit Timing Diagram

17.4 General Operation

The general operation of the SDRAM Controller includes address multiplexing, refresh and self-refresh are discussed. The SDRAM Controller is designed to support a broad range of broad range of JEDEC standard SDRAM configurations including devices of 64-, 128-, and 256-Mbit densities. Given the physical size constraints of the target applications, the design was optimized for memory device data widths of 16 and 32 bits. Table 17-9 summarizes the devices targeted by the design; however, the controller is capable of interfacing with devices of other widths and densities. However, only devices with 4 banks are supported. 133 MHz system bus operation is possible with PC133 compliant Single Data Rate memory devices.

Table 17-9. JEDEC Standard Single Data Rate SDRAMs

	SDRAM Configuration					
Size	**64 Mbit**		**128 Mbit**		**256 Mbit**	
Bus Width	16	32	16	32	16	32
Depth	4 Mword	2 Mword	8 Mword	4 Mword	16 Mword	8 Mword
Refresh Rows	4096 (15.62 µS)	4096 (15.62 µS)	4096 (15.62 µS)	4096 (15.62 µS)	8192 (7.81 µS)	4096 (15.62 µS)
# Banks	4	4	4	4	4	4
Bank Address	2	2	2	2	2	2
Row Address	12	11	12	12	13	12
Column Address	8	8	9	8	9	9
Data Qualifiers	2	4	2	4	2	4

17.4.1 Address Multiplexing

The JEDEC standard SDRAMs around which the controller was optimized use an asymmetrical array architecture with more row than column address lines. The SDRAM Controller multiplexes only those pins which change between the row and column addresses. The remaining (most significant) row addresses and the bank addresses are not multiplexed.

17.4.1.1 Multiplexed Address Bus

The SDRAM Controller multiplexed address bus is aligned to the column addresses so that the ARM address line A1 always appears on pin MA0. With this alignment, the “folding point” in the multiplexor is driven solely by the number of column address bits, although interleave mode causes a two bit shift to account for the bank addresses. Column bus widths of 8 to 11 bits are supported in non-interleave mode, although only 8 and 9 bit widths are allowed in interleave mode. Table 17-10 summarizes the multiplex options supported by the controller. Column addresses through A10 are driven regardless of the multiplexor configuration, although some of the lines will be unused for the smaller page sizes.

Memory width does not affect the multiplexer; however, it does affect how the memories are connected to the SDRAM Controller pins. The width of the multiplexed bus is one bit larger than in previous generations of the SDRAMC so that 16- and 32-bit memory systems can be supported with a minimal impact on the multiplex hardware. 16-bit memories utilize column address bits MA [n:0] for a 2^{n+1} byte/page memory, whereas 32-bit memory systems are shifted left by one bit and use MA [n+1:1]. This is demonstrated in the last two rows of Table 17-10 by the grayed out boxes. Note that the AP signal is duplicated in two bit positions to permit this signal to always appear on memory pin A10. Compare Figure 17-27 and Figure 17-28 as an example of the connection differences between 16- and 32-bit memory systems.

Table 17-10. Address Multiplexing by Column Width

Column Bits		Memory Width	SDRAM Controller Pin											
IAM = 0	IAM = 1		MA 11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
ROW														
8	–	16, 32	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9
9	–	16, 32	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
10	8	16, 32	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
11	9	16, 32	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12
			COLUMN											
ALL		16	AP	AP	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
		32	AP	AP	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1

Indicates address lines not required for this memory width.

17.4.1.2 Non-Multiplexed Address Bus

The most significant row address bits are not sampled when the column addresses are being driven, and therefore do not need to be multiplexed. The SDRAM Controller implementation takes advantage of this fact and uses the existing non-multiplexed address bus to provide these signals. Addresses A21 through A25 are needed across the supported configurations. The specific connections which will need to be made are dependent on the memory device type, density and bank interleave mode. These configuration-dependent connections are handled through the design of the external hardware. Examples are provided in Section 17.5, “SDRAM Operation,” .

17.4.1.3 Bank Addresses

Bank address connections are summarized in Table 17-11 while Figure 17-29 on page 17-43 illustrates the logical structure of how the internal ARM address signals are multiplexed out to the SDRAM memory bank address signals. Linear bank addressing (IAM = 0) utilizes the most-significant addresses to specify the active bank, where the actual bits used are dependent on the density of the memory system. In interleaved bank mode (IAM = 1), memory density has no affect on the selection of bank addresses. Instead, the bank addresses are totally dependent on the page size of the memory system, as determined by the number of column address bits and the memory data width. Page size and density for a number of potential configurations are documented in Table 17-11 through Table 17-19. For undocumented configurations, the following equations can be used to calculate page size and density.

$$\text{Page Size (bytes)} = 2^{\#\text{Column Address Bits}} \times (\text{Memory Width in Bits} / 8) \qquad \textit{Eqn. 17-1}$$

$$\text{Density (bytes)} = 2^{(\text{\# Column Address Bits} + \text{\# Row Address Bits})} \times (\text{Memory Width in Bits} / 2) \qquad \textit{Eqn. 17-2}$$

Table 17-11. Bank Address Bit Assignment

IAM	Density (Bytes)	Page Size (Bytes)	BA1	BA0
0	8M	X	ARM_A22	ARM_A21
	16M		ARM_A23	ARM_A22
	32M		ARM_A24	ARM_A23
	64M		ARM_A25	ARM_A24
1	X	512	ARM_A10	ARM_A9
		1024	ARM_A11	ARM_A10
		2048	ARM_A12	ARM_A11

17.4.2 Refresh

SDRAM Controller hardware satisfies all SDRAM refresh requirements after an initial configuration by the user software. 0, 1, 2, or 4 refresh cycles are scheduled at 31.25 µS (nominal 32 KHz clock) intervals, providing 0, 2048, 4096, or 8192 refresh cycles every 64 ms. The refresh rate is programmed through the SREFR field in the SDCTLx registers. Each array can have a different rate, allowing a mix of SDRAM devices (different density). Refresh is disabled by hardware reset.

A refresh request is made pending at each rising edge on the 32 KHz clock. In response to this request, the hardware gains control of the SDRAM as soon as any in-process bus cycle completes. Once it has gained control of the memory, commands are issued to precharge all banks. Following a row precharge delay (t_{RP}), an auto-refresh command is issued. At t_{RC} intervals, additional auto-refresh cycles are issued until the specified number of cycles have been run. Figure 17-29 illustrates a 2 refresh sequence.

Burst transfers in progress when the refresh request arrives are allowed to complete prior to the refresh operation. SDRAM bus accesses queued after the refresh request are held off until the refresh completes. In Figure 17-30, an access is queued just as the refresh begins. This cycle is delayed until the precharge and single refresh (SREFR = 01) cycles are run. Bus cycles targeted to other memory or peripheral devices are allowed to progress normally while the refresh is in progress. None of the pins shared between the SDRAM and other devices are required for the refresh operation.

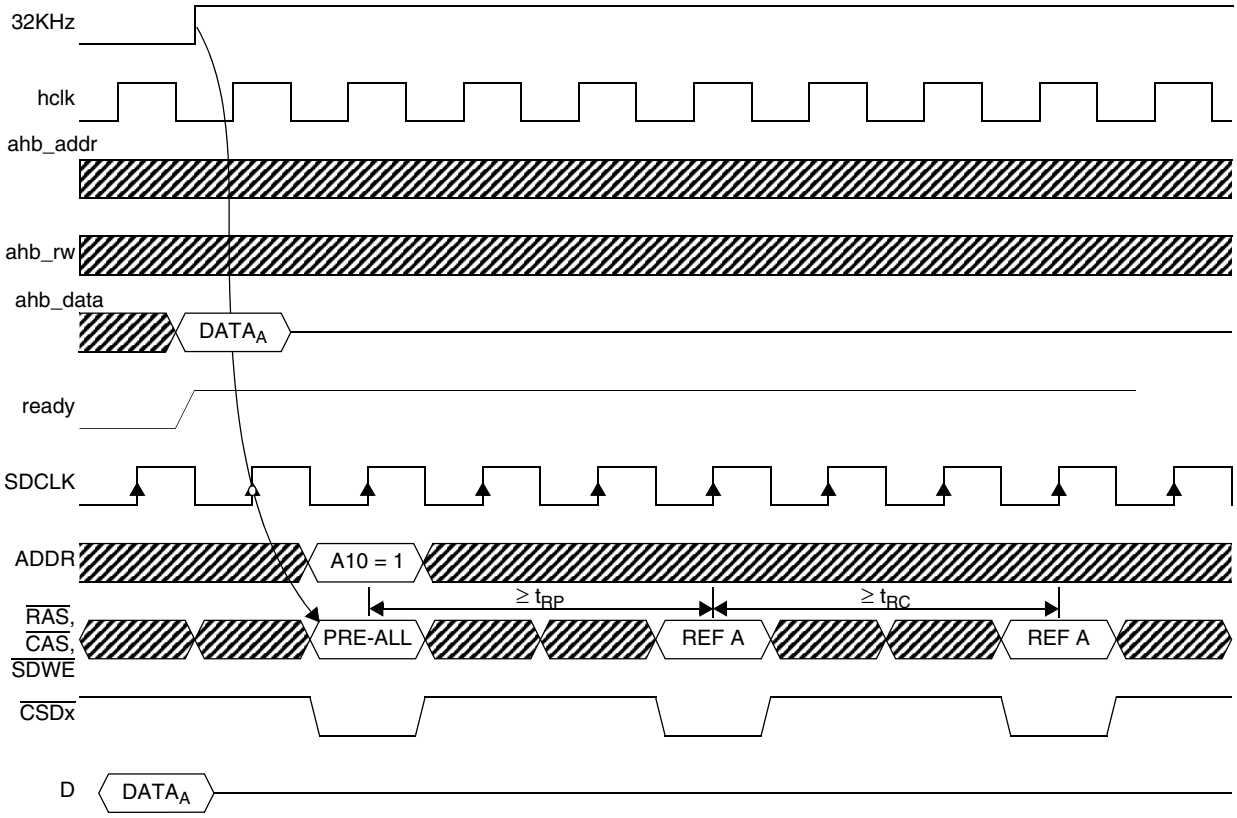


Figure 17-29. Hardware Refresh Timing Diagram

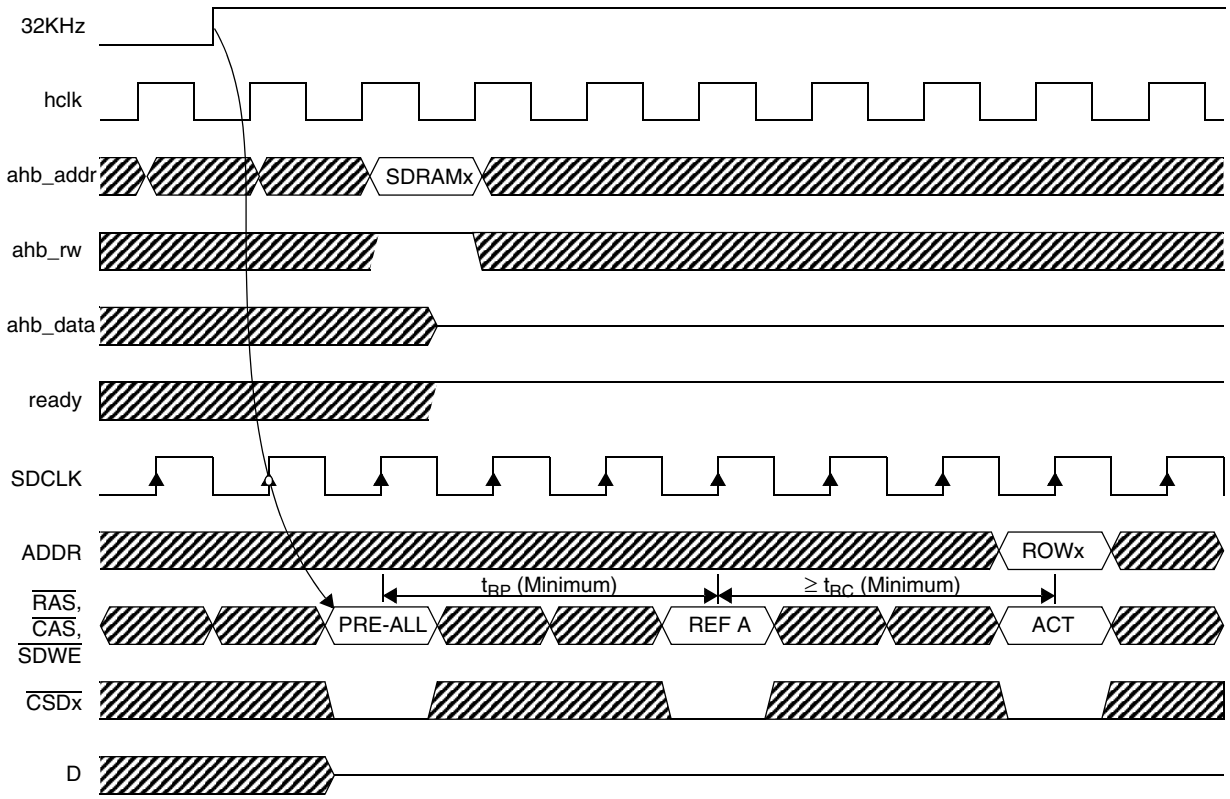


Figure 17-30. Hardware Refresh with Pending Bus Cycle Timing Diagram

17.4.3 Self-Refresh

SDRAM data must be retained during system reset and power conservation modes if refresh has been enabled. The SDRAM Controller detects these conditions and places the memory in self-refresh. If refresh has not been enabled, the SDRAM Controller places the memories in a lower power consumption mode known as Powerdown. This operation is described in Section 17.4.3.3, “Powerdown Operation During Reset and Low-Power Modes.” Self refresh mode can also be initiated manually by setting the SMODE bits to ‘100’. Refer to Section 17.3.6 for more details on manual self refresh.

17.4.3.1 Self-Refresh During Reset

The assertion of system reset triggers the SDRAM Controller to place the memory in self-refresh provided refresh had been previously enabled. Refresh during system reset is disabled by an SDRAM Controller reset. It remains disabled until the refresh rate is programmed to a non-zero value. Once enabled, self-refresh is invoked anytime system reset is asserted without a corresponding SDRAM reset.

17.4.3.2 Self-Refresh During Low-Power Mode

Low-power mode (STOP) also forces the SDRAM into self-refresh mode if refresh is enabled. Any time “mpen” is sampled low, the SDRAM Controller assumes that the bus masters are entering a low-power condition and it begins the self-refresh sequence once any in-progress bus access has completed. A Precharge All command is issued to close any open memory pages, the Self-Refresh command is issued,

and the clock enable is brought low. Once the memories are safely in their low-power state, the SDRAMC tells the system clock controller to enter sleep mode.

17.4.3.3 Powerdown Operation During Reset and Low-Power Modes

The powerdown mode is used instead of self-refresh whenever system reset or any of the low power modes occur and refresh has not been enabled. This memory operating mode does not remove power, as the name might imply. It simply lowers power consumption by disabling the clock input buffer and halting all internal activity. Since powerdown can only be entered if all banks are idle, a Precharge All command must be issued to the memories prior to stopping the clock. Figure 17-33 illustrates the powerdown sequence following assertion of system reset.

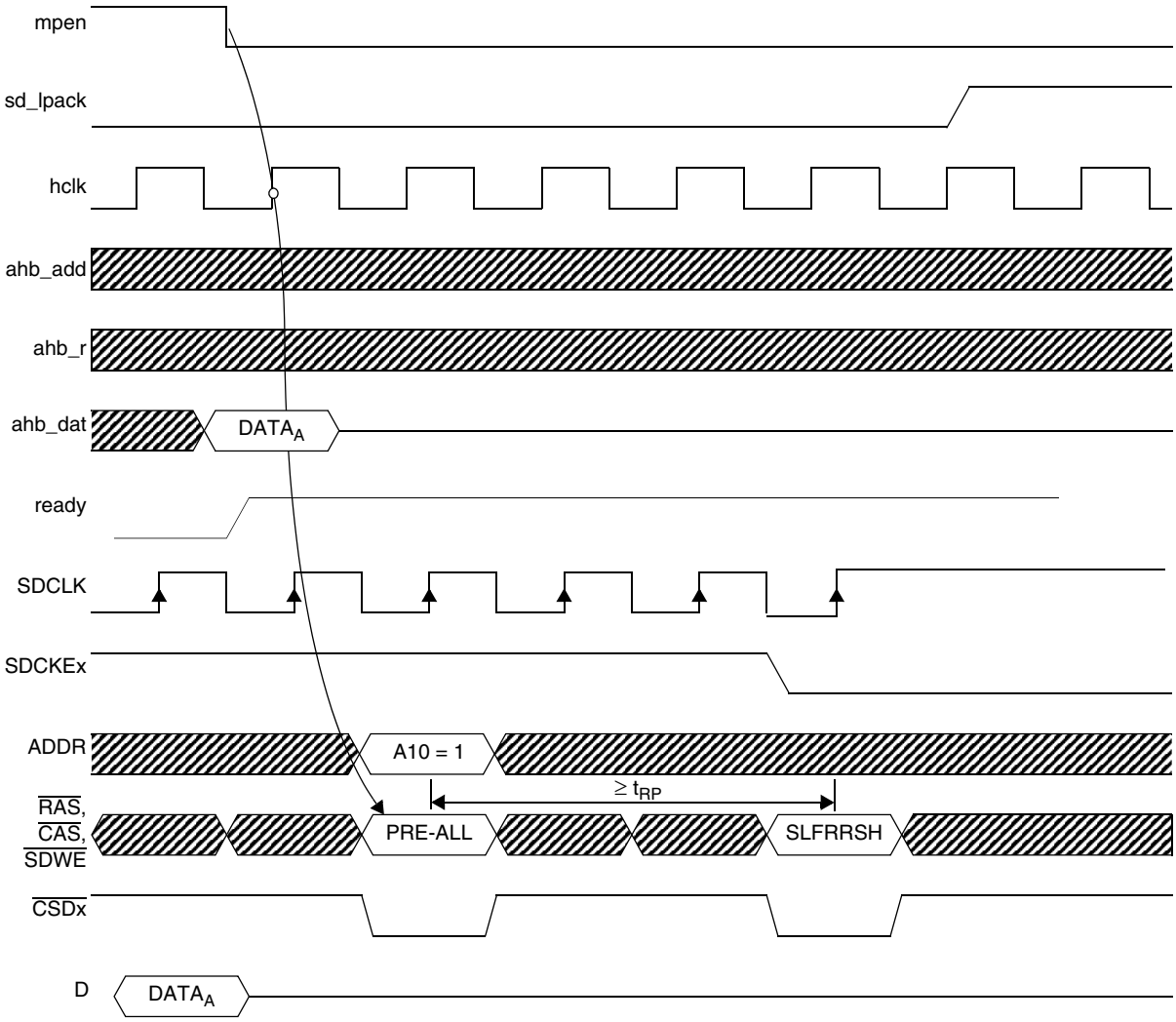


Figure 17-31. Self-Refresh Entry Due to Low-Power Mode Timing Diagram

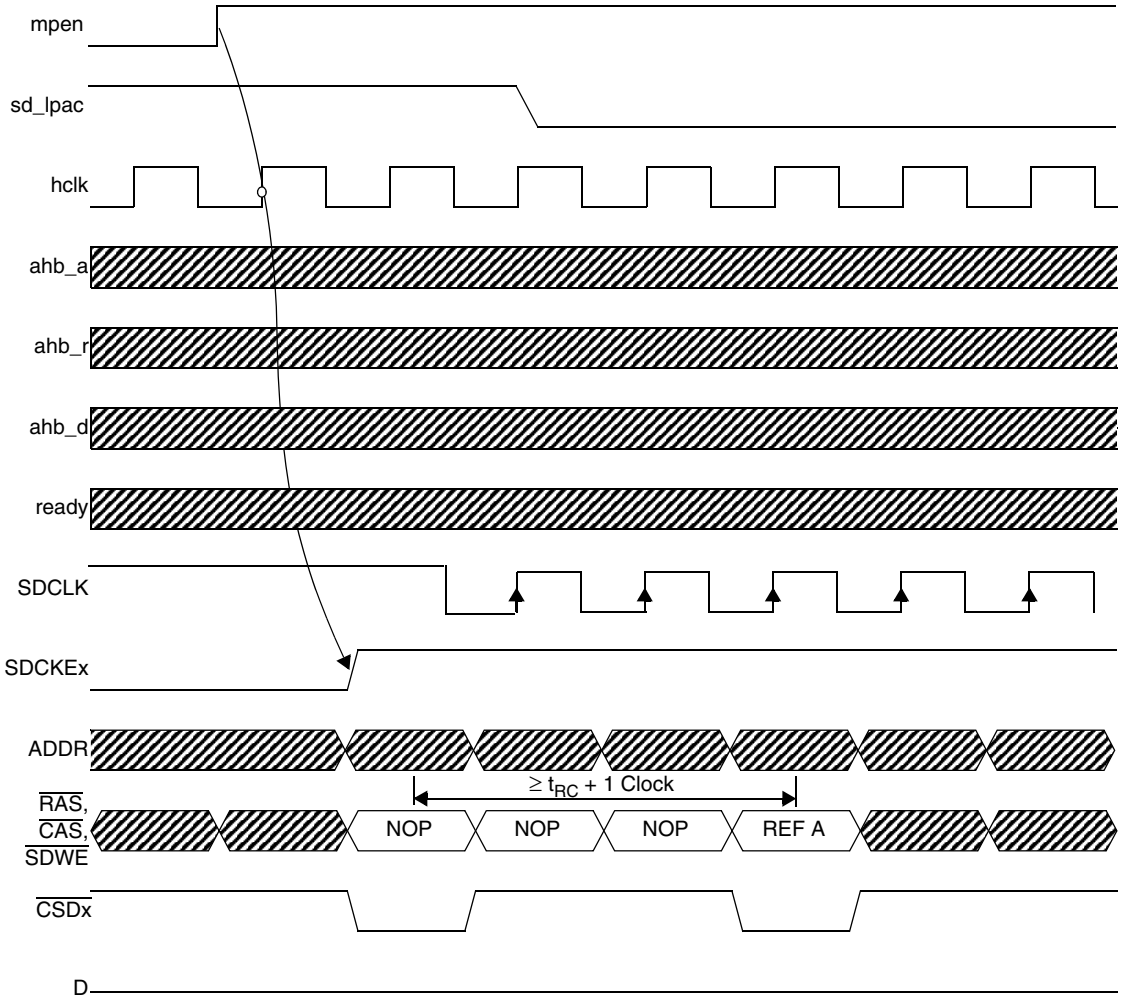


Figure 17-32. Low-Power Mode Self-Refresh Exit Timing Diagram

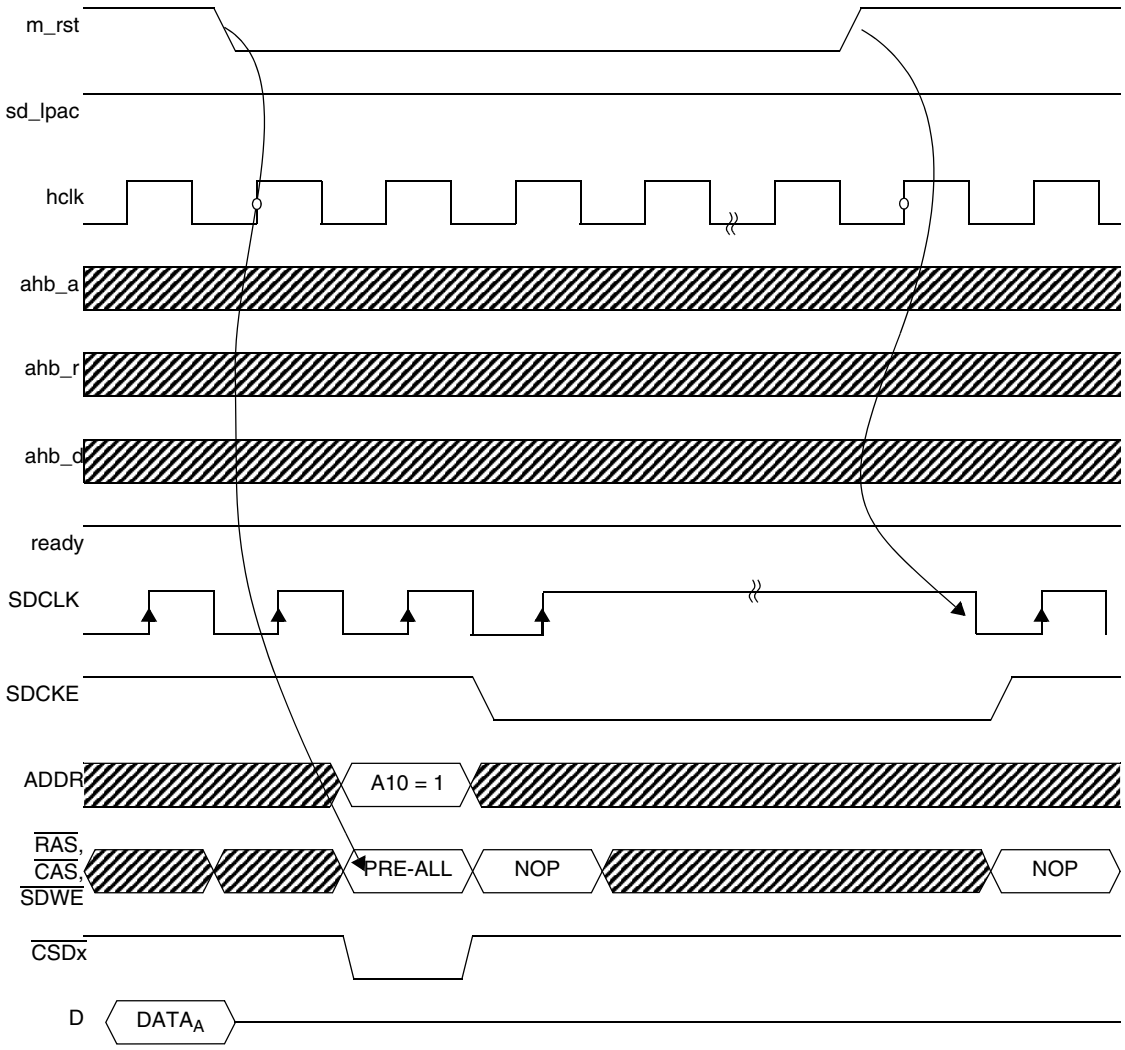


Figure 17-33. Power-Down Mode Resulting From Reset With Refresh Disabled

17.4.4 Power-Down Low-Power Mode

SDRAM memories incorporate a command sequence to disable the clock input buffer and suspend internal activity, lowering power consumption as much as an order of magnitude. The SDRAM Controller implements a Power-Down timeout mechanism to take advantage of this feature. Several software selectable time-outs are provided to accommodate for varying system conditions, with the timeout condition being specified in the Power-Down Timeout (PWDT) field in the SDCTLx register. The feature is disabled out of reset.

SDRAM Power-Down actually consists of two sub-modes: Active Power-Down and Precharge Power-Down. The distinguishing factor between them is whether banks remain active while the clock is stopped. Active Power-Down allows banks to remain activated, while Precharge Power-Down/Deep Power-Down Mode does not.

17.4.4.1 Precharge Power-Down

Programming PWDT [1:0] = 01 causes the SDRAM Controller to place the memories in powerdown mode anytime the controller detects that no banks are active. This mode is useful in applications where a memory array is accessed infrequently and the chances of another access to the same page are minimal.

Reading or writing to memory activates a page within the addressed bank. Reset, software generated precharge, and hardware initiated refresh are three ways to close an active bank. The periodically occurring refresh will be the normal means that invokes the powerdown mode. At each refresh interval, all banks will be closed by a precharge-all command, followed by the refresh operation. The controller will then issue the powerdown command to the memories. A few cycle delay is incurred with the first read or write cycle in order to restart the clocks, but only on the first cycle. After that, the clocks will continue to run until the next refresh operation or until any active banks are manually precharged.

Page misses on read and write cycles cause the addressed bank to be closed (precharged) and a new page opened within the bank. This operation does not cause the clocks to stop, nor does manually precharging only a single bank within the memory. All banks within the memory must be inactive before the powerdown mode is invoked.

17.4.4.2 Active Power-Down

The second Power-Down mode is selected whenever PWDT [1:0] = 1x. In this mode the clocks are stopped after a selectable delay from the last access to the array. Active banks are not closed prior to disabling the SDRAM clock. Either 64 (PWDT [1:0] = 10) or 128 (PWDT [1:0] = 11) cycle delays are possible. SDRAM clocks are counted from the end of the last read or write access. Subsequent read or write accesses, and self-refresh modes reset the counter. Auto-refresh cycles do not affect the counter; however, if the counter expires during a refresh operation the clock will be disabled immediately following the refresh.

17.4.4.3 Refresh During Precharge Power-Down/Active Power-Down

Refresh requests queued while the clock is suspended will restart the clock, run the appropriate number of refresh cycles, and then disable the clock again.

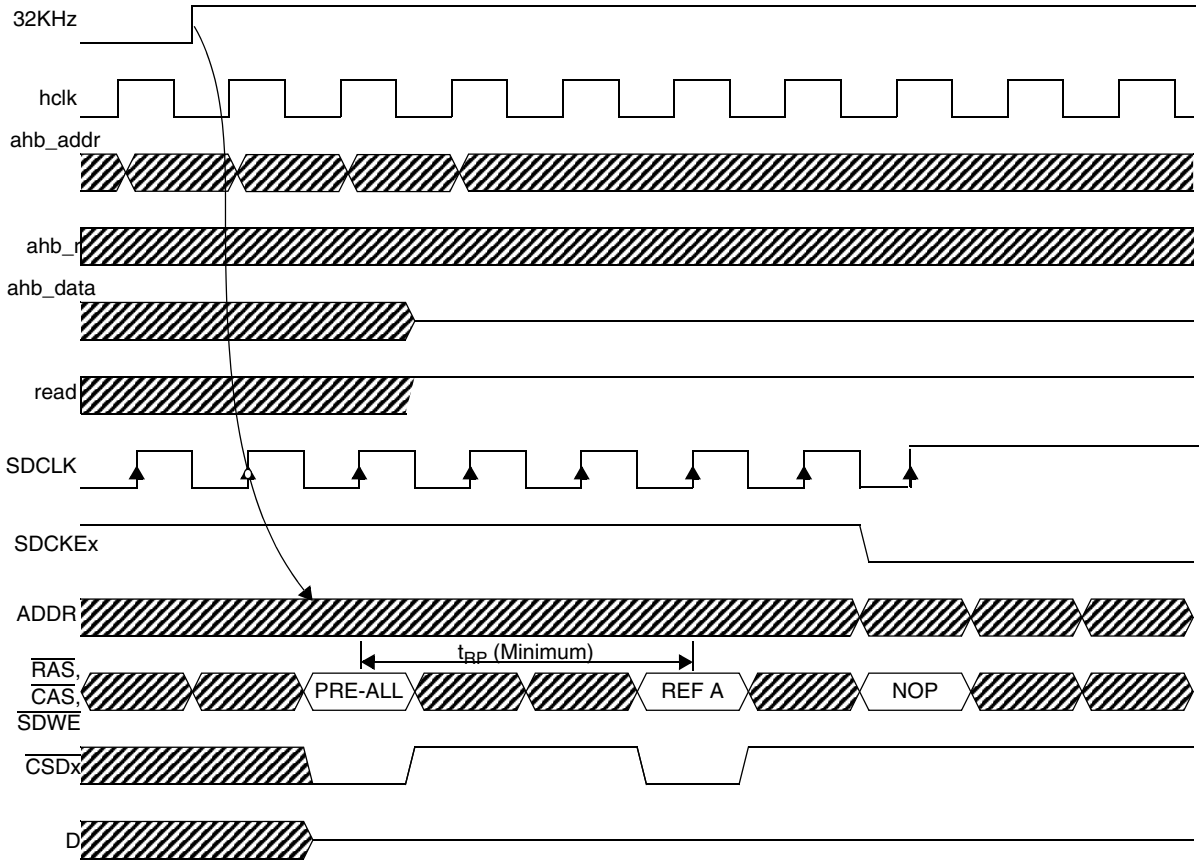


Figure 17-34. Precharge Power-Down Mode Entry Timing Diagram

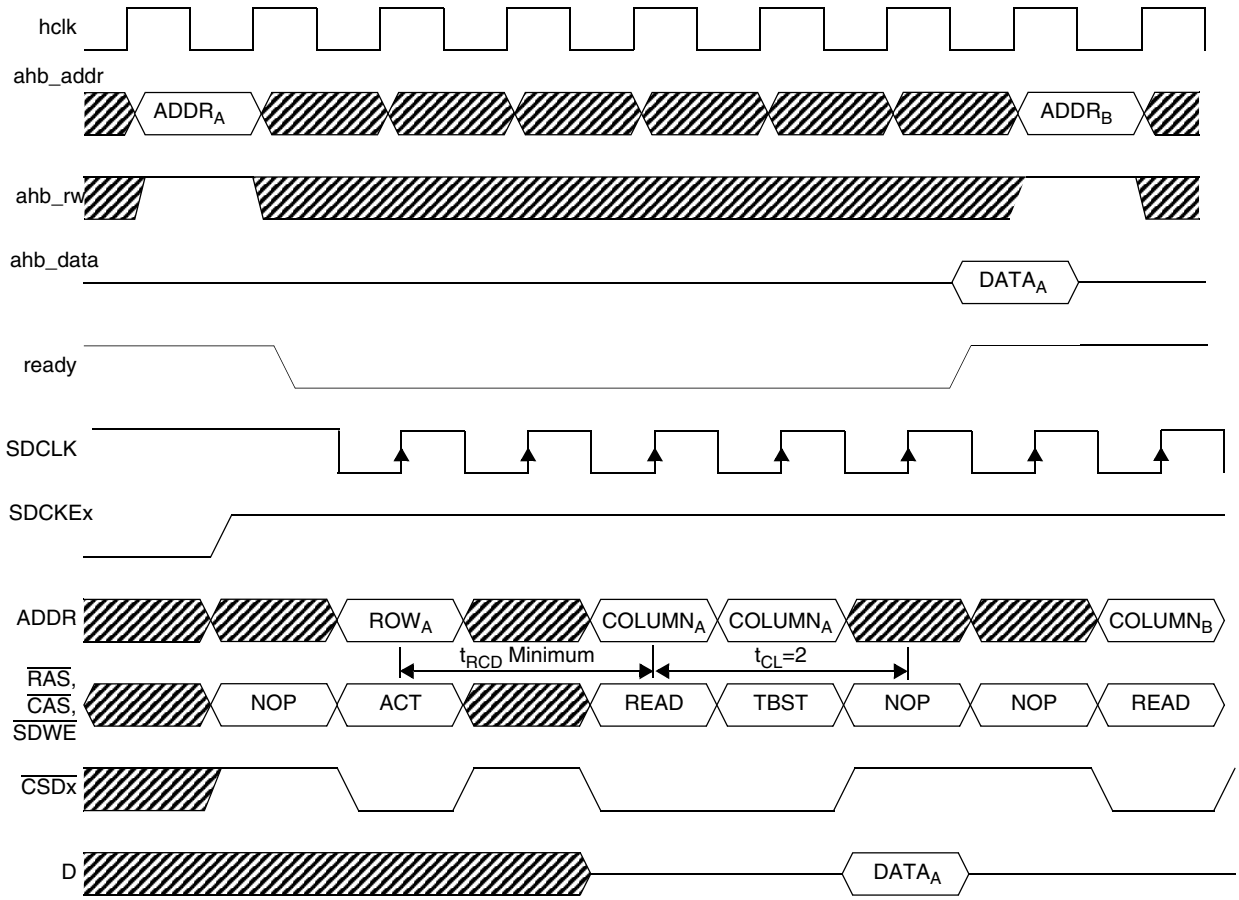


Figure 17-35. Power-Down Exit Timing Diagram

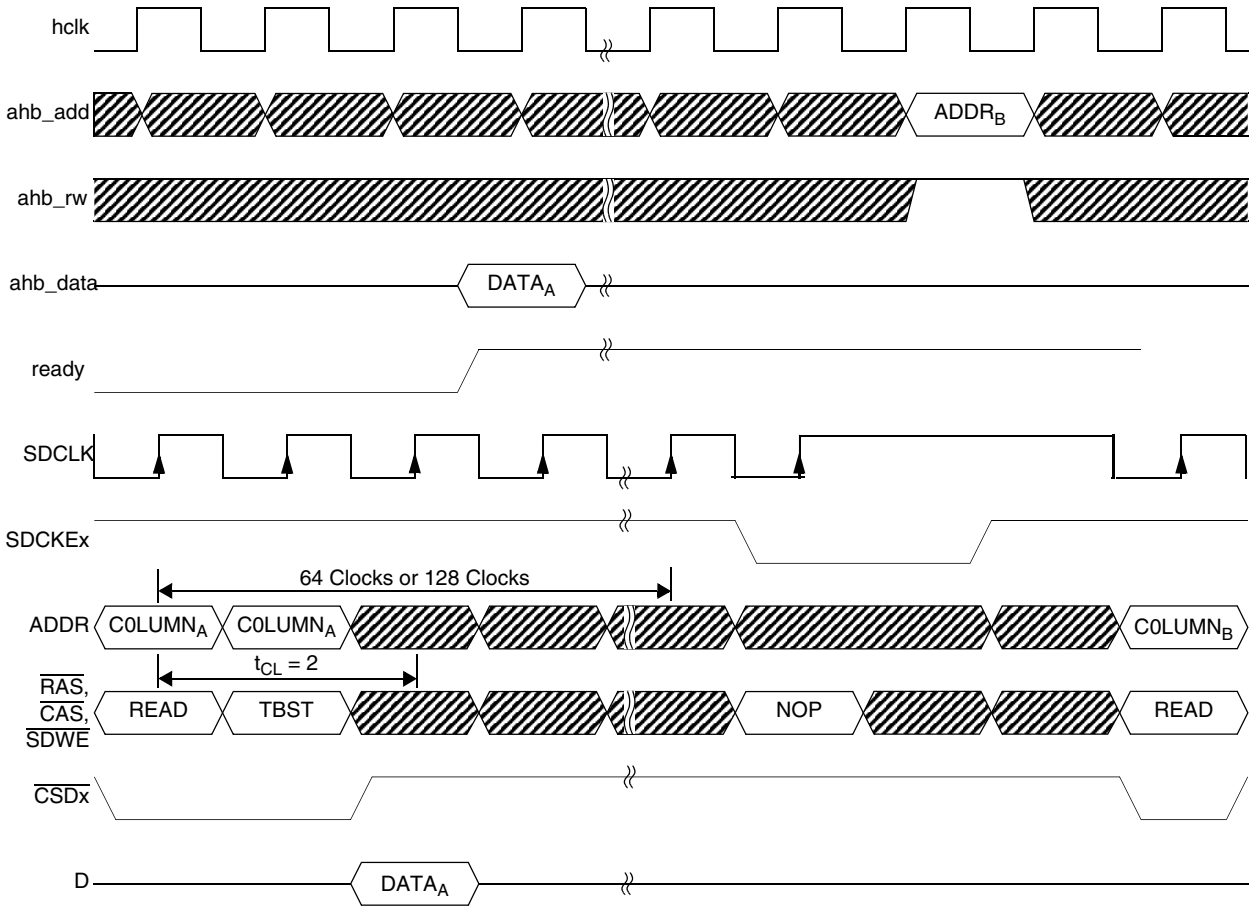


Figure 17-36. Active Power-Down Timing Diagram (64 Clock Example)

17.5 SDRAM Operation

The following subsections provide details on selecting compatible SDRAM memories and configuring the controller to work with the memory system.

17.5.1 SDRAM Selection

Table 17-9 lists the memories around which the controller is optimized, but many other memory types are also suitable for use. Important characteristics to consider when choosing a memory module are:

- The page comparators expect 4 bank memories. 2 bank devices are not supported. Their use will result in the memory and controller losing synchronization when crossing page boundaries.
- Page (column) addressing must match one of the supported sizes.
- Memory density can be larger or smaller than those directly supported, although some memory may be inaccessible or redundantly mapped. In linear mode (IAM bit of the Control Register is 0), the bank addresses are the most significant addresses and connecting a memory smaller than the selected density will result in one or more banks being inaccessible.

- The controller is designed for memories meeting PC133 timing specifications at 133 MHz system operation. Use of non-compliant memories or other system clock rates require a thorough analysis of all timing parameters.

17.5.2 Configuring Controller for SDRAM Memory Array

Configuration register programming options and controller-memory physical connections provide flexibility to accommodate different memory types and system configurations. Options are broadly grouped into 3 categories:

- Physical Characteristics: row and column address bus widths, data bus width, and interleave mode
- Timing Parametrics: CAS latency, row precharge and cycle delays, and refresh rate
- Functional Features: Power-Down timer and supervisor/user protection

Table 17-12 through Table 17-19 are provided to assist the designer with the selection of the correct physical parameters for a number of preferred memory configurations. Timing parametrics are addressed in the following subsections.

17.5.2.1 CAS Latency

CAS latency is determined by the operating frequency and access time of the memories. For a 133 MHz system clock frequency and PC133 compliant memories, the CAS latency will generally be programmed to 3 clocks, although the memory specifications should always be consulted to confirm this value. CAS latency must be programmed in two places: the chip select Control Register and the SDRAM Mode Register. See Table 17-5 for a description of the control register encoding and Section 17.5.4, “Mode Register Programming,” for the details on programming the SDRAM mode register.

17.5.2.2 Row Precharge Delay

Row precharge delay is defined as the delay between a precharge command and the next row activate command to the same bank. The SDRAM memory specification provides the minimum precharge delay as t_{RP}, usually in terms of ns. Therefore the user will have to calculate the number clocks needed for the row precharge delay and program this value into the SRP bit in SDRAM Control register described in Table 17-5. For example, a t_{RP} specification of 15 ns with a 133 MHz clock (7.52 ns period) results in 1.995 clocks, rounded to 2.

17.5.2.3 Row-to-Column Delay

The row-to-column delay is defined as the delay between a row activate command and a subsequent read or write command. The SDRAM memory specification provides the minimum row-to-column (or ACTIVE to READ or WRITE) delay as t_{RCD}, usually in terms of ns. Given the system bus speed, the user must calculate the number of clocks needed after an activate command is given before the subsequent read or write command and program this value into the SRCD bit field in the SDRAM Control register, as described in Table 17-5. For example, a t_{RCD} specification of 15 ns with a 133 MHz clock (7.52 ns period) results in 1.995 clocks, rounded to 2.

17.5.2.4 Row Cycle Delay

Row cycle delay is defined as the delay between a refresh and any subsequent refresh or read/write access. The SDRAM memory data sheet usually denotes this timing parameter as the Auto Refresh period t_{RFC}, and is specified in terms of ns. Given the system bus speed, the user must calculate the number of clocks after a refresh command is given before the next refresh or read/write command is issued. This value must then be programmed into the SRC bit field in the SDRAM Control register. For example, a t_{RFC} specification of 66 ns with a 133 MHz clock results in a minimum of 8.78 clocks, rounded to 9. Therefore, SRC must be set for 9 clocks.

17.5.2.5 Refresh Rate

The refresh rate is the rate by which the SDRAM controller is required to refresh each row in the SDRAM memory. The SDRAM memory specification provides the total number of rows per bank, and the corresponding refresh rate must be programmed into the SREFR field in the SDRAM control register (See Table 17-5). For example, if the SDRAM memory contains 8192 rows, or requires 8192 AUTO REFRESH cycles every 64 ms, then the refresh rate can be calculated as 64 ms / 8192 rows = 7.81 µs per row. Therefore the user programs the SREFR field with 11. Refer to Section 17.4.2, “Refresh,” on page -42 for more information.

17.5.2.6 Memory Configuration Examples

Nine different SDRAM configurations are demonstrated. These examples are 64 Mbit, 128 Mbit, and 256 Mbit SDRAM memories in single x16, dual x16, and single x32 configurations. All examples use bank address interleaving (Control Register bit IAM = 1).

All single-configuration 16-bit examples are shown connected to the lower half of the data bus. Alternatively, the memory can be connected to the upper half of the data bus by swapping the data connections to D [31:16] and the data qualifier mask connections to DQM3 and DQM2. In this case, it will be necessary to program the DSIZ field in the Control Register to a value of 0 (configurations shown require a value of 1).

For a more comprehensive look at the memory configuration schemes, refer to the address muxing tables in section 17.5.2.7 "Address Muxing Tables".

NOTE

The following examples assume that the number of rows and columns for each given SDRAM density follow the JEDEC standard. The SDRAM Controller can interface to SDRAMS which do not follow the JEDEC standards for row and column sizes, however the user must ensure that the SDRAM Control Register bits ROW and COL are programmed to the appropriate number of rows and columns given in the SDRAM data sheet. These examples do not cover non-JEDEC standard SDRAMS.

Table 17-12. Single 8Mx16 Control Register Values

Control Field	Value
Density	16 Mbyte
Page Size	1024
ROW	12
COL	9
DSIZ	16 (D [15:0])
IAM	Interleaved

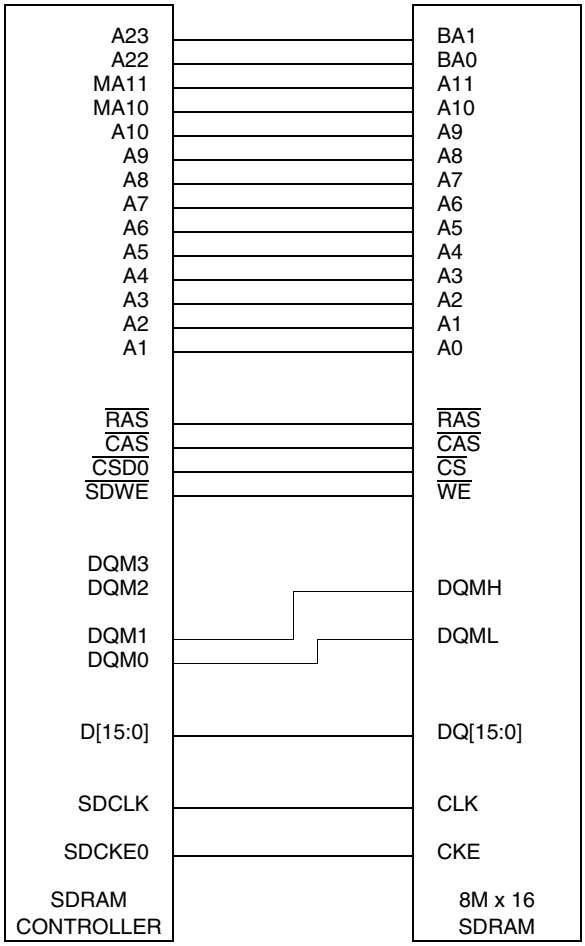


Figure 17-37. Single 128 Mbit (8M x 16) Connection Diagram (IAM = 1)

Table 17-13. Single 16Mx16 Control Register Values

Control Field	Value
Density	32 Mbyte
Page Size	1024
ROW	13
COL	9
DSIZ	16 (D [15:0])
IAM	Interleaved

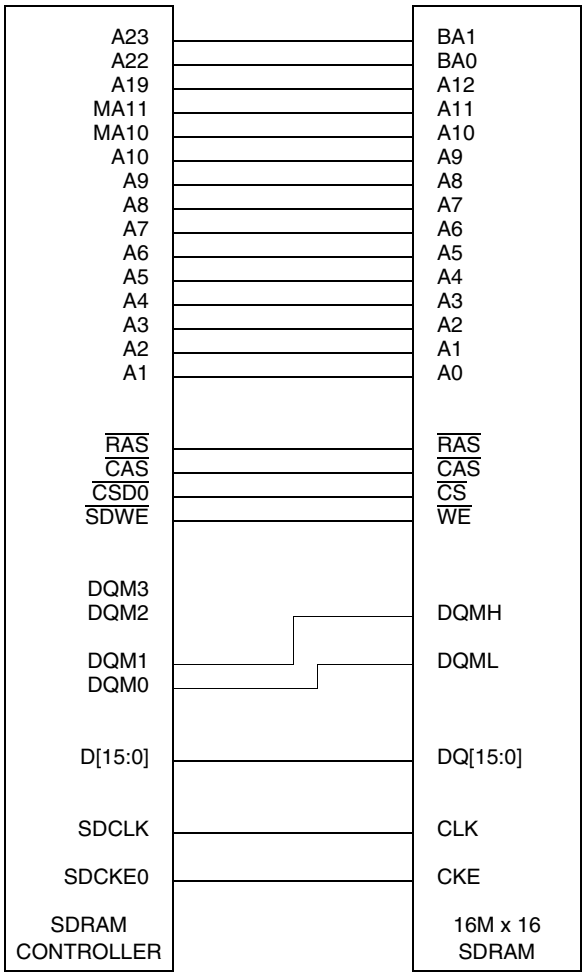


Figure 17-38. Single 256 Mbit (16M x 16) Connection Diagram (IAM = 1)

Table 17-14. Dual 4Mx16 Control Register Values

Control Field	Value
Density	16 Mbyte
Page Size	1024
ROW	12
COL	8
DSIZ	32 (D [31:0])
IAM	Interleaved

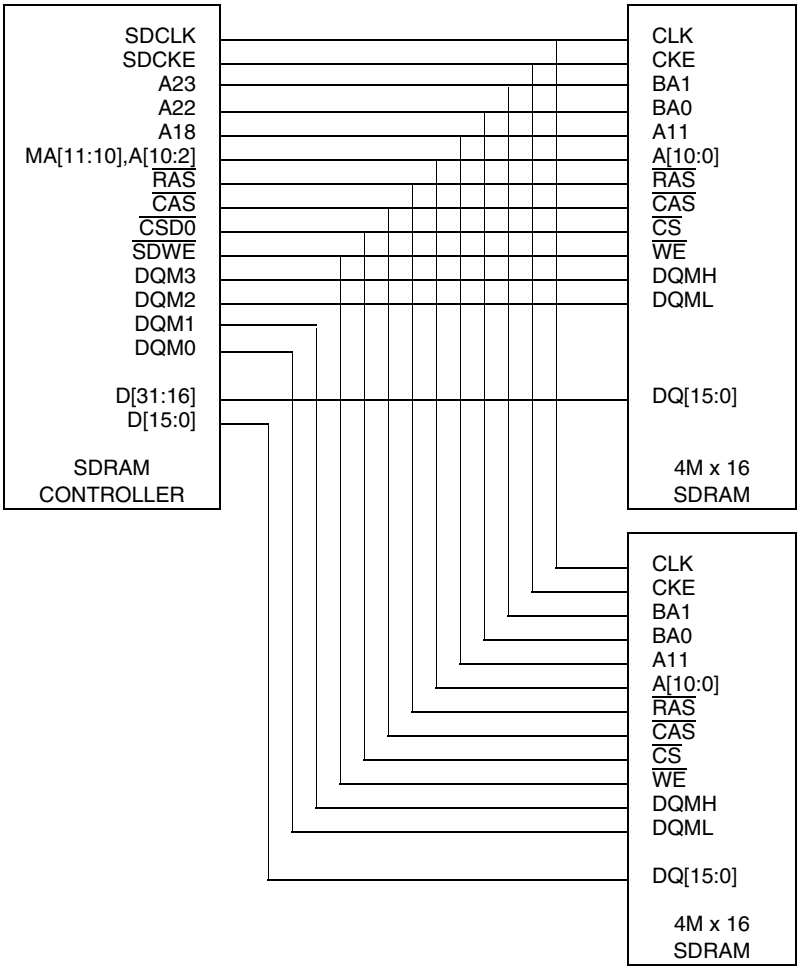


Figure 17-39. Dual 64 Mbit (4M x 16 x 2) Connection Diagram (IAM = 1)

Table 17-15. Dual 8Mx16 Control Register Values

Control Field	Value
Density	32 Mbyte
Page Size	2048
ROW	12
COL	9
DSIZ	32 (D [31:0])
IAM	Interleaved

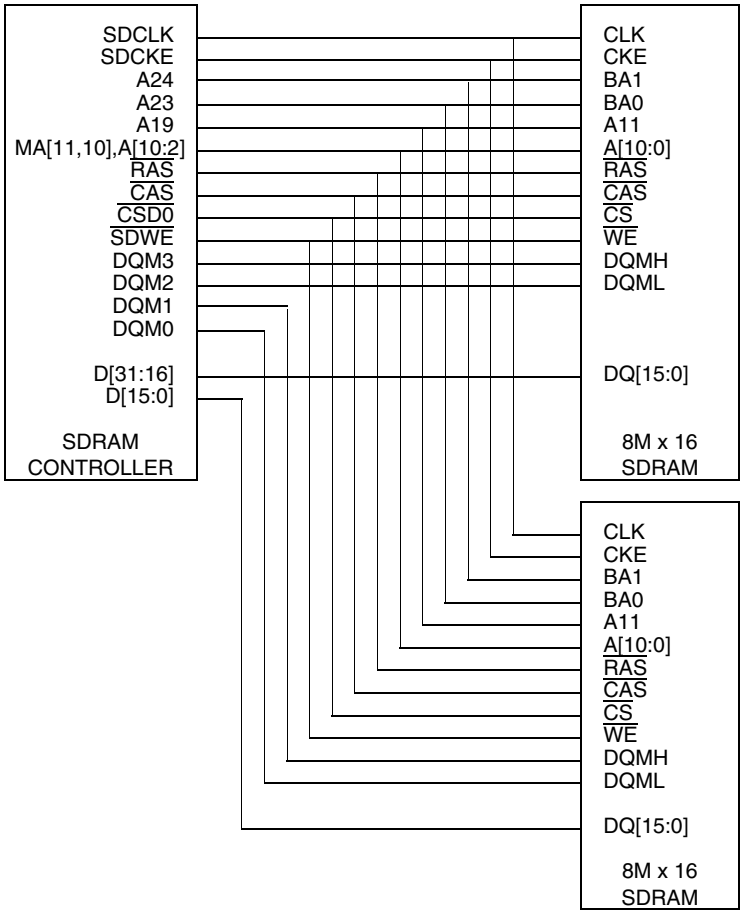


Figure 17-40. Dual 128 Mbit (8M x 16 x 2) Connection Diagram (IAM = 1)

Table 17-16. Dual 16Mx16 Control Register Values

Control Field	Value
Density	64 Mbyte
Page Size	2048
ROW	13
COL	9
DSIZ	32 (D [31:0])
IAM	Interleaved

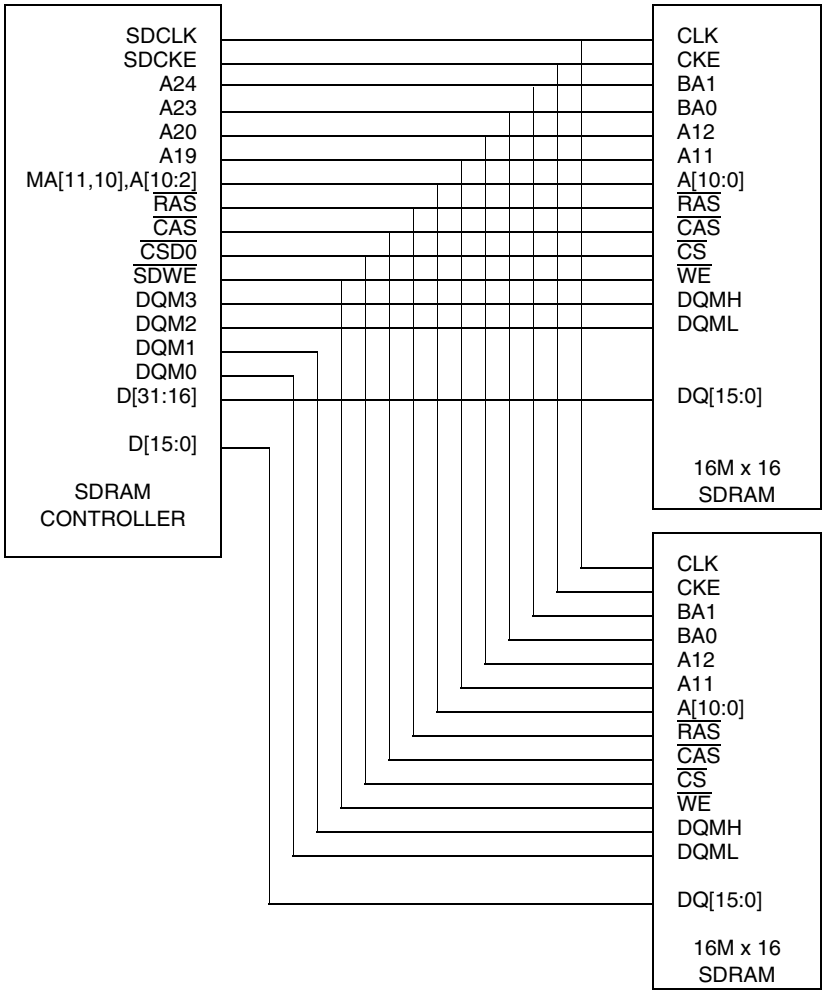


Figure 17-41. Dual 256 Mbit (16M x 16 x 2) Connection Diagram (IAM = 1)

Table 17-17. Single 2Mx32 Control Register Values

Control Field	Value
Density	8 Mbyte
Page Size	1024
ROW	11
COL	8
DSIZ	32 (D [31:0])
IAM	Interleaved

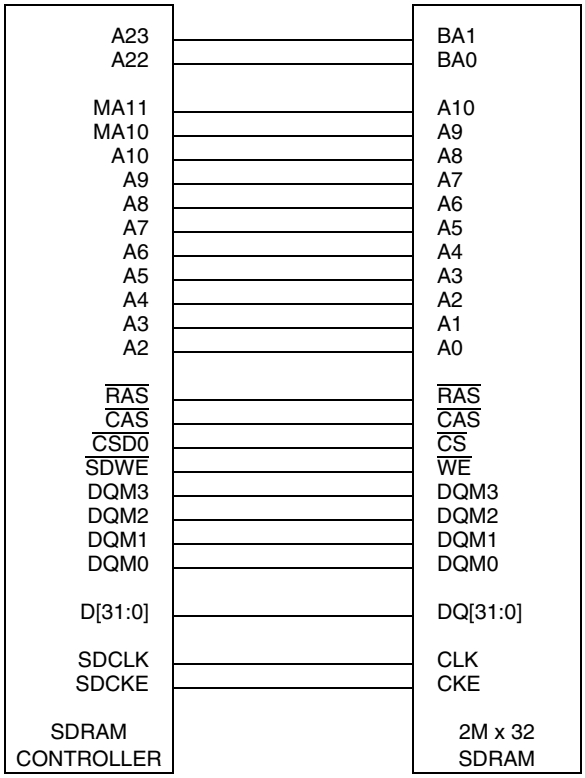


Figure 17-42. Single 64 Mbit (2M x 32) Connection Diagram (IAM = 1)

Table 17-18. Single 4Mx32 Control Register Values

Control Field	Value
Density	16 Mbyte
Page Size	1024
ROW	12
COL	8
DSIZ	32 (D [31:0])
IAM	Interleaved

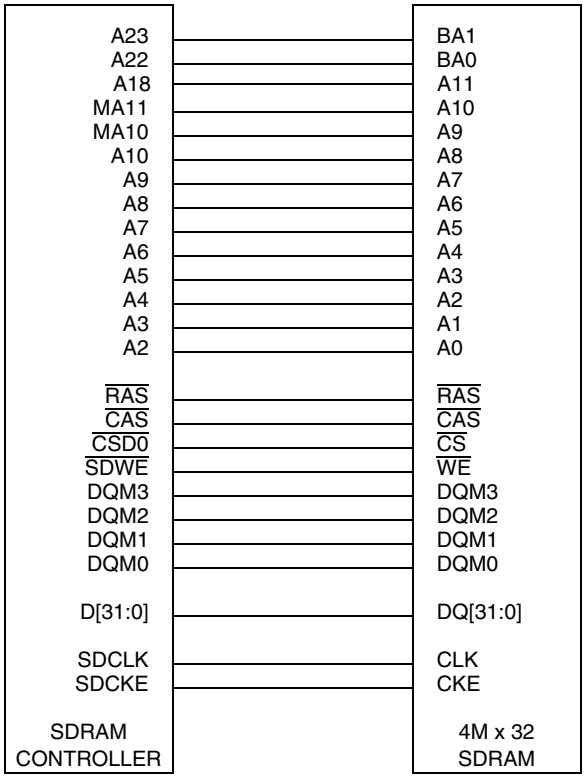


Figure 17-43. Single 128 M Mbit (4M x 32) Connection Diagram (IAM = 1)

NOTE

JEDEC has not issued a standard pinout and array configuration for the 128 Mbit density memories in a x32 package option. This connection diagram is based on the PC133 Standard.

Table 17-19. Single 8Mx32 Control Register Values

Control Field	Value
Density	32 Mbyte
Page Size	1024
ROW	12
COL	9
DSIZ	32 (D [31:0])
IAM	Interleaved

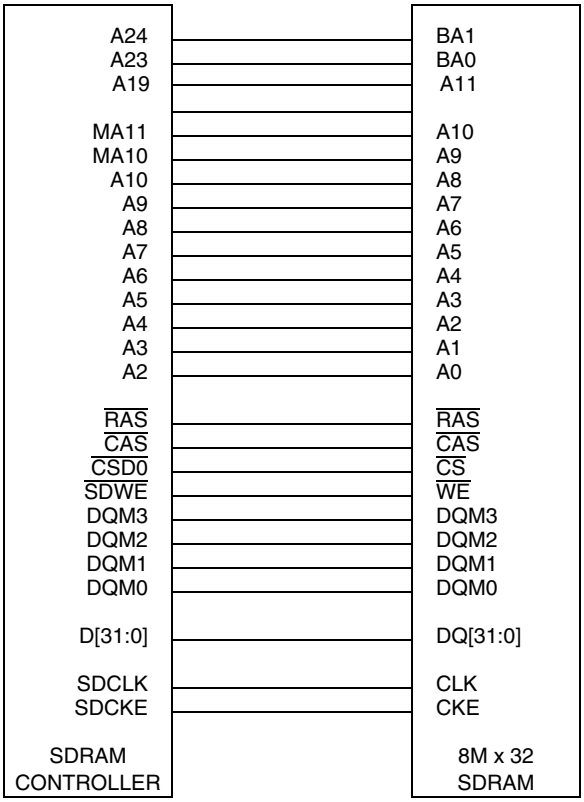


Figure 17-44. Single 256 Mbit (8M x 32) Connection Diagram (IAM = 1)

NOTE

JEDEC has not issued a standard pinout and array configuration for the 256 Mbit density memories in a ×32 package option. This connection diagram is based on the PC133 Standard.

17.5.2.7 Address Muxing Tables

The following tables detail the address multiplexing scheme from the ARM AHB through the SDRAM controller and out to the external address bus, and illustrates how the SDRAM memory address bus is interfaced to the external address bus for common memory configurations.

Table 17-20. 8Mx16x1 IAM=0 SDRAM Address Translation Table

i.MX21 AHB Address Bus (denoted as ARM_Axx)	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9	ARM_A8	ARM_A7	ARM_A6	ARM_A5	ARM_A4	ARM_A3	ARM_A2	ARM_A1
Bank, Row, Column	BA1	BA0	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	C8	C7	C6	C5	C4	C3	C2	C1	C0
SDRAM Controller Muxing Scheme	ARM_A23	ARM_A22	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0

SDRAM Controller row/column Address Muxing	ARM_A23	ARM_A22	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0

Rotate Columns bits around row/column folding point

i.MX21 External Address Signals	A18	A17	MA11	MA10	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
SDRAM Address Signals	BA1	BA0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

i.MX21 to SDRAM interface

Table 17-21. 8Mx16x1 IAM=1 SDRAM Address Translation Table

i.MX21 AHB Address Bus (denoted as ARM_Axx)	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9	ARM_A8	ARM_A7	ARM_A6	ARM_A5	ARM_A4	ARM_A3	ARM_A2	ARM_A1
Bank, Row, Column	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	BA1	BA0	C8	C7	C6	C5	C4	C3	C2	C1	C0
SDRAM Controller Muxing Scheme	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0	ARM_A11	ARM_A10	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0

SDRAM Controller row/column Address Muxing	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0	ARM_A11	ARM_A10

Rotate Columns bits around row/column folding point

i.MX21 External Address Signals	MA11	MA10	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A23	A22
SDRAM Address Signals	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	BA1	BA0

i.MX21 to SDRAM interface

Table 17-22. 16Mx16x1 IAM=0 SDRAM Address Translation Table

i.MX21 AHB Address Bus (denoted as ARM_Axx)	ARM_A24	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9	ARM_A8	ARM_A7	ARM_A6	ARM_A5	ARM_A4	ARM_A3	ARM_A2	ARM_A1
Bank, Row, Column	BA1	BA0	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	C8	C7	C6	C5	C4	C3	C2	C1	C0
SDRAM Controller Muxing Scheme	ARM_A24	ARM_A23	ARM_A22	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0

SDRAM Controller row/column Address Muxing	ARM_A24	ARM_A23	ARM_A22	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0

Rotate Columns bits around row/column folding point

i.MX21 External Address Signals	A19	A18	A17	MA11	MA10	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
SDRAM Address Signals	BA1	BA0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

i.MX21 to SDRAM interface

Table 17-23. 16Mx16x1 IAM=1 SDRAM Address Translation Table

i.MX21 AHB Address Bus (denoted as ARM_Axx)	ARM_A24	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9	ARM_A8	ARM_A7	ARM_A6	ARM_A5	ARM_A4	ARM_A3	ARM_A2	ARM_A1
Bank, Row, Column	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	BA1	BA0	C8	C7	C6	C5	C4	C3	C2	C1	C0
SDRAM Controller Muxing Scheme	ARM_A24	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0	ARM_A11	ARM_A10	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0

SDRAM Controller row/column Address Muxing	ARM_A24	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0	ARM_A11	ARM_A10

Rotate Columns bits around row/column folding point

i.MX21 External Address Signals	A19	MA11	MA10	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A23	A22
SDRAM Address Signals	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	BA1	BA0

i.MX21 to SDRAM interface

Table 17-24. 4Mx16x2 IAM=0 SDRAM Address Translation Table

i.MX21 AHB Address Bus (denoted as ARM_Axx)	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9	ARM_A8	ARM_A7	ARM_A6	ARM_A5	ARM_A4	ARM_A3	ARM_A2
Bank, Row, Column	BA1	BA0	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	C7	C6	C5	C4	C3	C2	C1	C0
SDRAM Controller Muxing Scheme	ARM_A23	ARM_A22	ARM_A21	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

SDRAM Controller row/column Address Muxing	ARM_A23	ARM_A22	ARM_A21	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

Rotate Columns bits around row/column folding point

i.MX21 External Address Signals	A18	A17	A16	MA11	MA10	A10	A9	A8	A7	A6	A5	A4	A3	A2
SDRAM Address Signals	BA1	BA0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

i.MX21 to SDRAM interface

Table 17-25. 4Mx16x2 IAM=1 SDRAM Address Translation Table

i.MX21 AHB Address Bus (denoted as ARM_Axx)	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9	ARM_A8	ARM_A7	ARM_A6	ARM_A5	ARM_A4	ARM_A3	ARM_A2
Bank, Row, Column	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	BA1	BA0	C7	C6	C5	C4	C3	C2	C1	C0
SDRAM Controller Muxing Scheme	ARM_A23	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	ARM_A11	ARM_A10	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

SDRAM Controller row/column Address Muxing	ARM_A23	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	ARM_A11	ARM_A10

Rotate Columns bits around row/column folding point

i.MX21 External Address Signals	A18	MA11	MA10	A10	A9	A8	A7	A6	A5	A4	A3	A2	A23	A22
SDRAM Address Signals	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	BA1	BA0

i.MX21 to SDRAM interface

Table 17-26. 8Mx16x2 IAM=0 SDRAM Address Translation Table

i.MX21 AHB Address Bus (denoted as ARM_Axx)	ARM_A24	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9	ARM_A8	ARM_A7	ARM_A6	ARM_A5	ARM_A4	ARM_A3	ARM_A2
Bank, Row, Column	BA1	BA0	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	C8	C7	C6	C5	C4	C3	C2	C1	C0
SDRAM Controller Muxing Scheme	ARM_A23	ARM_A22	ARM_A21	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

SDRAM Controller row/column Address Muxing	ARM_A24	ARM_A23	ARM_A22	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

Rotate Columns bits around row/column folding point

i.MX21 External Address Signals	A19	A18	A17	MA11	MA10	A10	A9	A8	A7	A6	A5	A4	A3	A2
SDRAM Address Signals	BA1	BA0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

i.MX21 to SDRAM interface

Table 17-27. 8Mx16x2 IAM=1 SDRAM Address Translation Table

i.MX21 AHB Address Bus (denoted as ARM_Axx)	ARM_A24	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9	ARM_A8	ARM_A7	ARM_A6	ARM_A5	ARM_A4	ARM_A3	ARM_A2
Bank, Row, Column	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	BA1	BA0	C8	C7	C6	C5	C4	C3	C2	C1	C0
SDRAM Controller Muxing Scheme	ARM_A24	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	ARM_A12	ARM_A11	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

SDRAM Controller row/column Address Muxing	ARM_A24	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	ARM_A12	ARM_A11

Rotate Columns bits around row/column folding point

i.MX21 External Address Signals	A19	MA11	MA10	A10	A9	A8	A7	A6	A5	A4	A3	A2	A24	A23
SDRAM Address Signals	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	BA1	BA0

i.MX21 to SDRAM interface

Table 17-28. 16Mx16x2 IAM=0 SDRAM Address Translation Table

i.MX21 AHB Address Bus (denoted as ARM_Axx)	ARM_A25	ARM_A24	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9	ARM_A8	ARM_A7	ARM_A6	ARM_A5	ARM_A4	ARM_A3	ARM_A2
Bank, Row, Column	BA1	BA0	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	C8	C7	C6	C5	C4	C3	C2	C1	C0
SDRAM Controller Muxing Scheme	ARM_A25	ARM_A24	ARM_A23	ARM_A22	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

SDRAM Controller row/column Address Muxing	ARM_A25	ARM_A24	ARM_A23	ARM_A22	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

Rotate Columns bits around row/column folding point

i.MX21 External Address Signals	A20	A19	A18	A17	MA11	MA10	A10	A9	A8	A7	A6	A5	A4	A3	A2
SDRAM Address Signals	BA1	BA0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

i.MX21 to SDRAM interface

Table 17-29. 16Mx16x2 IAM=1 SDRAM Address Translation Table

i.MX21 AHB Address Bus (denoted as ARM_Axx)	ARM_A25	ARM_A24	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9	ARM_A8	ARM_A7	ARM_A6	ARM_A5	ARM_A4	ARM_A3	ARM_A2
Bank, Row, Column	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	BA1	BA0	C8	C7	C6	C5	C4	C3	C2	C1	C0
SDRAM Controller Muxing Scheme	ARM_A25	ARM_A24	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	ARM_A12	ARM_A11	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

SDRAM Controller row/column Address Muxing	ARM_A25	ARM_A24	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	ARM_A12	ARM_A11

Rotate Columns bits around row/column folding point

i.MX21 External Address Signals	A20	A19	MA11	MA10	A10	A9	A8	A7	A6	A5	A4	A3	A2	A24	A23
SDRAM Address Signals	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	BA1	BA0

i.MX21 to SDRAM interface

Table 17-30. 2Mx32 IAM=0 SDRAM Address Translation Table

i.MX21 AHB Address Bus (denoted as ARM_Axx)	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9	ARM_A8	ARM_A7	ARM_A6	ARM_A5	ARM_A4	ARM_A3	ARM_A2
Bank, Row, Column	BA1	BA0	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	C7	C6	C5	C4	C3	C2	C1	C0
SDRAM Controller Muxing Scheme	ARM_A22	ARM_A21	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

SDRAM Controller row/column Address Muxing	ARM_A22	ARM_A21	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

Rotate Columns bits around row/column folding point

i.MX21 External Address Signals	A17	A16	MA11	MA10	A10	A9	A8	A7	A6	A5	A4	A3	A2
SDRAM Address Signals	BA1	BA0	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

i.MX21 to SDRAM interface

Table 17-31. 2Mx32 IAM=1 SDRAM Address Translation Table

i.MX21 AHB Address Bus (denoted as ARM_Axx)	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9	ARM_A8	ARM_A7	ARM_A6	ARM_A5	ARM_A4	ARM_A3	ARM_A2
Bank, Row, Column	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	BA1	BA0	C7	C6	C5	C4	C3	C2	C1	C0
SDRAM Controller Muxing Scheme	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	ARM_A11	ARM_A10	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

SDRAM Controller row/column Address Muxing	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	ARM_A11	ARM_A10

Rotate Columns bits around row/column folding point

i.MX21 External Address Signals	MA11	MA10	A10	A9	A8	A7	A6	A5	A4	A3	A2	A23	A22
SDRAM Address Signals	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	BA1	BA0

i.MX21 to SDRAM interface

Table 17-32. 4Mx32 IAM=0 SDRAM Address Translation Table

i.MX21 AHB Address Bus (denoted as ARM_Axx)	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9	ARM_A8	ARM_A7	ARM_A6	ARM_A5	ARM_A4	ARM_A3	ARM_A2
Bank, Row, Column	BA1	BA0	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	C7	C6	C5	C4	C3	C2	C1	C0
SDRAM Controller Muxing Scheme	ARM_A23	ARM_A22	ARM_A21	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

SDRAM Controller row/column Address Muxing	ARM_A23	ARM_A22	ARM_A21	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

Rotate Columns bits around row/column folding point

i.MX21 External Address Signals	A18	A17	A16	MA11	MA10	A10	A9	A8	A7	A6	A5	A4	A3	A2
SDRAM Address Signals	BA1	BA0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

i.MX21 to SDRAM interface

Table 17-33. 4Mx32 IAM=1 SDRAM Address Translation Table

i.MX21 AHB Address Bus (denoted as ARM_Axx)	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9	ARM_A8	ARM_A7	ARM_A6	ARM_A5	ARM_A4	ARM_A3	ARM_A2
Bank, Row, Column	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	BA1	BA0	C7	C6	C5	C4	C3	C2	C1	C0
SDRAM Controller Muxing Scheme	ARM_A23	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	ARM_A11	ARM_A10	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

SDRAM Controller row/column Address Muxing	ARM_A23	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	ARM_A11	ARM_A10

Rotate Columns bits around row/column folding point

i.MX21 External Address Signals	A18	MA11	MA10	A10	A9	A8	A7	A6	A5	A4	A3	A2	A23	A22
SDRAM Address Signals	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	BA1	BA0

i.MX21 to SDRAM interface

Table 17-34. 8Mx32 IAM=0 SDRAM Address Translation Table

i.MX21 AHB Address Bus (denoted as ARM_Axx)	ARM_A24	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9	ARM_A8	ARM_A7	ARM_A6	ARM_A5	ARM_A4	ARM_A3	ARM_A2
Bank, Row, Column	BA1	BA0	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	C8	C7	C6	C5	C4	C3	C2	C1	C0
SDRAM Controller Muxing Scheme	ARM_A24	ARM_A23	ARM_A22	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

SDRAM Controller row/column Address Muxing	ARM_A24	ARM_A23	ARM_A22	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

Rotate Columns bits around row/column folding point

i.MX21 External Address Signals	A19	A18	A17	MA11	MA10	A10	A9	A8	A7	A6	A5	A4	A3	A2
SDRAM Address Signals	BA1	BA0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

i.MX21 to SDRAM interface

Table 17-35. 8Mx32 IAM=1 SDRAM Address Translation Table

i.MX21 AHB Address Bus (denoted as ARM_Axx)	ARM_A24	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9	ARM_A8	ARM_A7	ARM_A6	ARM_A5	ARM_A4	ARM_A3	ARM_A2
Bank, Row, Column	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	BA1	BA0	C8	C7	C6	C5	C4	C3	C2	C1	C0
SDRAM Controller Muxing Scheme	ARM_A24	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	ARM_A12	ARM_A11	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

SDRAM Controller row/column Address Muxing	ARM_A24	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	ARM_A12	ARM_A11

Rotate Columns bits around row/column folding point

i.MX21 External Address Signals	A19	MA11	MA10	A10	A9	A8	A7	A6	A5	A4	A3	A2	A24	A23
SDRAM Address Signals	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	BA1	BA0

i.MX21 to SDRAM interface

Table 17-36. 16Mx32 IAM=0 SDRAM Address Translation Table

i.MX21 AHB Address Bus (denoted as ARM_Axx)	ARM_A25	ARM_A24	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9	ARM_A8	ARM_A7	ARM_A6	ARM_A5	ARM_A4	ARM_A3	ARM_A2
Bank, Row, Column	BA1	BA0	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	C8	C7	C6	C5	C4	C3	C2	C1	C0
SDRAM Controller Muxing Scheme	ARM_A25	ARM_A24	ARM_A23	ARM_A22	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

SDRAM Controller row/column Address Muxing	ARM_A25	ARM_A24	ARM_A23	ARM_A22	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

Rotate Columns bits around row/column folding point

i.MX21 External Address Signals	A20	A19	A18	A17	MA11	MA10	A10	A9	A8	A7	A6	A5	A4	A3	A2
SDRAM Address Signals	BA1	BA0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

i.MX21 to SDRAM interface

Table 17-37. 16Mx32 IAM=1 SDRAM Address Translation Table

i.MX21 AHB Address Bus (denoted as ARM_Axx)	ARM_A25	ARM_A24	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9	ARM_A8	ARM_A7	ARM_A6	ARM_A5	ARM_A4	ARM_A3	ARM_A2
Bank, Row, Column	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	BA1	BA0	C8	C7	C6	C5	C4	C3	C2	C1	C0
SDRAM Controller Muxing Scheme	ARM_A25	ARM_A24	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	ARM_A12	ARM_A11	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1

SDRAM Controller row/column Address Muxing	ARM_A25	ARM_A24	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	ARM_A12	ARM_A11

Rotate Columns bits around row/column folding point

i.MX21 External Address Signals	A20	A19	MA11	MA10	A10	A9	A8	A7	A6	A5	A4	A3	A2	A24	A23
SDRAM Address Signals	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	BA1	BA0

i.MX21 to SDRAM interface

17.5.3 SDRAM Reset Initialization

SDRAM initialization must follow a defined sequence following the power-on condition. The steps are as follows:

1. Apply power and start clock. Attempt to maintain CKE high, DQM high and NOP conditions at the command inputs.
2. Maintain stable power, clock and NOP conditions for a minimum of 200 µs.
3. Issue precharge commands for all banks either with precharge all or precharge individual bank commands.
4. After all banks are in the idle state for a minimum time of t_{RP}, issue 8 or more auto-refresh commands.
5. Issue a mode register set command to initialize the mode register.
6. SDRAM is now ready for normal operation.

The SDRAM Controller accomplishes steps 1 and 2 in hardware, but relies on software assistance to complete the remaining actions. The 200 µs stabilization period is guaranteed by the use of 2 reset signals whose negations are separated by this amount. An SDRAM reset signal ($\overline{\text{SD\_RST}}$) is asserted coincident

with the system reset used by the rest of the chip, but it negates 200 µs prior to the negation of system reset. The SDRAM Controller leaves the SDRAM arrays in a NOP condition following the negation of the DRAM reset.

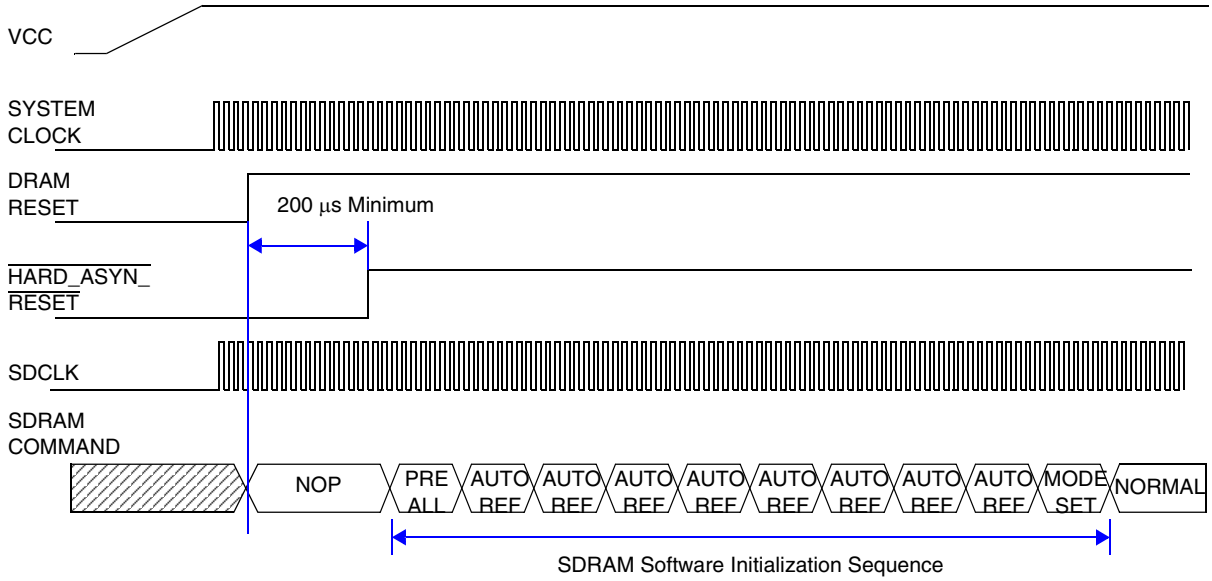


Figure 17-45. SDRAM Power-On Initialization Sequence

Following negation of system reset, initialization software must complete steps 3 through 5 using the special operating modes enabled by the SMODE field in the SDRAM Control Register. To precharge the SDRAM array, the SDRAM Controller operating mode is set to “precharge command” and an access is made to the SDRAM address range with address bit A10 = 1. Instead of running a normal read or write cycle, the controller issues a precharge all command to the addressed array. The operating mode is then switched to “auto-refresh” and 8 accesses made to the SDRAM address space. Each of the accesses will result in a refresh command to the addressed array. A “mode register set” command is required to complete the initialization sequence. The value written is system dependent. Consult Section 17.5.4, “Mode Register Programming,” for details. Finally, the controller is placed back in the normal mode of operation so that subsequent accesses to the address space will result in normal read and write cycles to the SDRAM array.

Although the initialization sequence described in the previous paragraphs is only required at power-on, it may be repeated at any time the programmer deems necessary.

Example 17-1. init_sdram

```
init_sdram:
         ldr      r2,CSD_REGS                // base address of registers
         ldr      r3,PRE_ALL_CMD
         st       r3,(0,r2)                  // put array 0 in precharge command mode
         st       r3,(4,r2)                  // put array 1 in precharge command mode
         ldr      r4,SDRAM_ARRAY_0           // get address of first array
         ldr      r5,SDRAM_ARRAY_1           // get address of second array
         ld       r3,(0,r4)                  // precharge array 0
         ld       r3,(0,r5)                  // precharge array 1
         ldr      r3,AUTO_REF_CMD
         st       r3,(0,r2)                  // put array 0 in auto-refresh mode
         st       r3,(4,r2)                  // put array 1 in auto-refresh mode
         movi     r6,7                       // load loop counter
L1       ld       r3,(0,r4)                  // run auto-refresh cycle to array 0
         ld       r3,(0,r5)                  // run auto-refresh cycle to array 1
         decgt    r6
         bt       L1                         // 8 refresh cycles complete?
         ldr      r3,SET_MODE_REG_CMD
         st       r3,(0,r2)                  // setup CSD0 for mode register write
         st       r3,(4,r2)                  // setup CSD1 for mode register write
         ldr      r3,MODE_REG_VAL0           // array 0 mode register value
         ld       r3,(0,r3)                  // New mode register value on address bus
         ldr      r3,MODE_REG_VAL1           // array 1 mode register value
         ld       r3,(0,r3)                  // Write CSD1 mode register
         ldr      r3,NORMAL_MODE
         st       r3,(0,r2)                  // setup CSD0 for normal operation
         st       r3,(4,r2)                  // setup CSD1 for normal operation

CSD_REGS          .long    0xDF000000
SDRAM_ARRAY_0:    .long    0xC0000000
SDRAM ARRAY_1:    .long    0xC4000000
PRE_ALL_CMD       .long    0xXXXXXXXX
AUTO_REF_CMD      .long    0xXXXXXXXX
SET_MODE_REG_CMD.long      0xXXXXXXXX
MODE_REG_VAL0     .long    0xXXXXXXXX
MODE_REG_VAL1     .long    0xXXXXXXXX
NORMAL_MODE       .long    0xXXXXXXXX
```

17.5.4 Mode Register Programming

The mode register is used to set the SDRAM operating characteristics including CAS latency, burst length, burst mode, and write data length. The settings depend on system characteristics including the operating frequency, memory device type, burst buffer/cache line length, and bus width. Operating characteristics vary by device type, so the data sheet must be consulted to determine the actual value to be written. In order to demonstrate the procedure, the following system characteristics will be used:

- 2 JEDEC Standard 64 Mbit (4M x 16) SDR SDRAMs configured as a x32 memory in bank interleaved mode (IAM = 1)
- 133 MHz System Clock Frequency
- Sequential burst, burst length of 8
- Single word writes—that is, no bursting on writes

Table 17-38 illustrates the Mode Register bit assignments for the 64 Mbit SDRAM.

64 Mbit SDRAM Mode Register

	BA0	BA1	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
TYPE	0	0	0	0	WM	0	0	LTMODE			BT	BL		

Table 17-38. SDRAM Mode Register Description

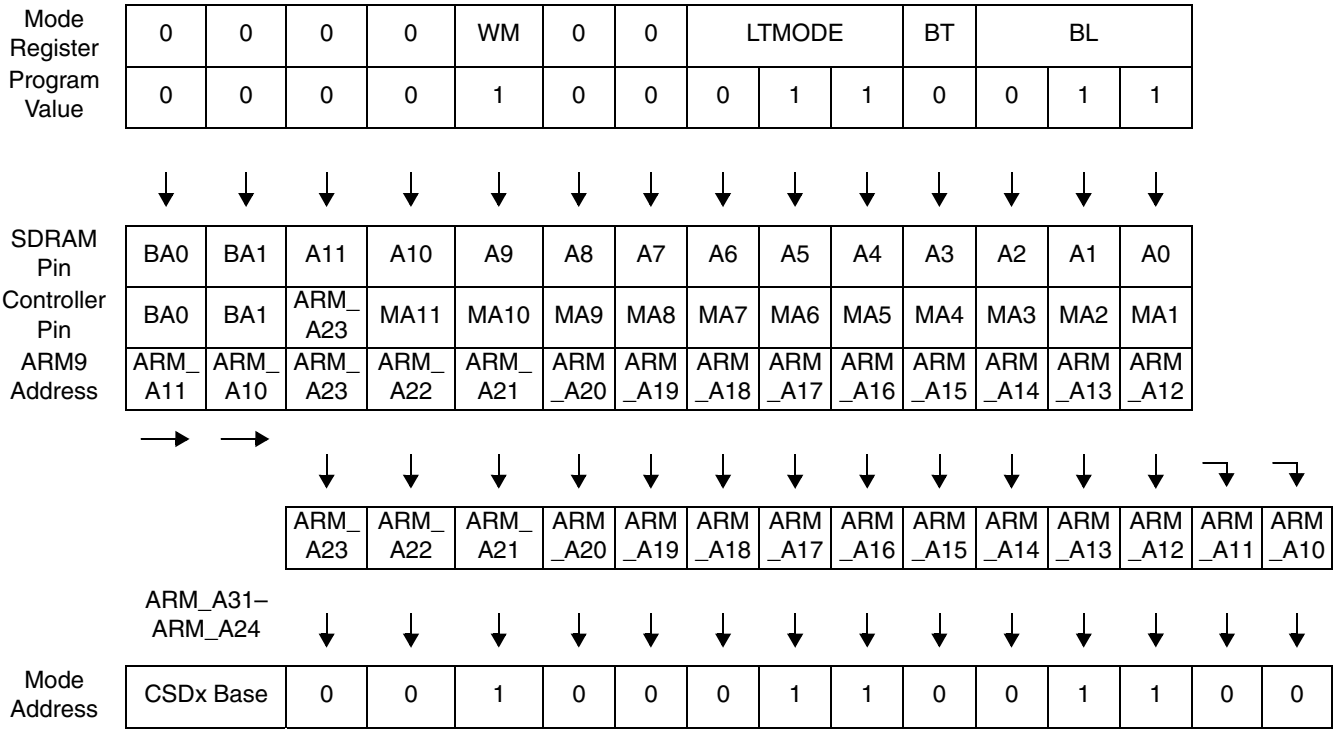
Name	Description	Settings
WM Bit A9	**Write Mode**—Selects between burst writes and single location writes	0 = Burst Writes 1 = Single Word Writes
LTMODE Bits A6–A4	**Latency Mode**—Sets latency between column address and data	000 = Reserved 001 = 1 clock 010 = 2 clocks 011 = 3 clocks 1xx = Reserved
BT Bit A3	**Burst Type**—Selects burst type	0 = Sequential 1 = Interleave
BL Bits A2–A0	**Burst Length**—A burst length of four matches the M340 cache line length when the SDRAM is 32 bits wide. A 16 bit wide SDRAM requires a burst length of eight because the four 32-bit line fill cycles will be decomposed into eight 16-bit accesses.	000 = 1 001 = 2 010 = 4 011 = 8 111 = Full Page 10x = Reserved 1x0 = Reserved

For this example:

- Sequential burst (BT = 0)
- Burst length of 8 (BL = 011)
- Single word writes (WM = 1)
- 3 Clock Latency (LTMODE = 011)

Once the mode register value has been determined, it must be converted to an address. The mode register is written via the address bus and the memory data sheet will specify the SDRAM address bits on which to place the data. One final transformation is necessary to align the address to the multiplexed outputs of the SDRAM Controller. Memory density and bus width determine the alignment of the SDRAM to the controller pins and must be taken into account during the calculation. Table 17-39 provides an example calculation using the same system characteristics used in the previous example.

Table 17-39. Example Address Calculation for Mode Set



The following two tables illustrate the mode register bits relative to the internal AHB bus for common SDRAM memory configurations.

Table 17-40. i.MX21 SDRAM Mode Register Bits Relative to i.MX21 Internal Address Bus for x16 Configurations

		i.MX21 Internal Address Bus (Denoted as ARM_Axx)																						
Memory Configuration	**IAM**	**ARM_A31**	**ARM_A30**	**ARM_A29**	**ARM_A28**	**ARM_A27**	**ARM_A26**	**ARM_A25**	**ARM_A24**	**ARM_A23**	**ARM_A22**	**ARM_A21**	**ARM_A20**	**ARM_A19**	**ARM_A18**	**ARM_A17**	**ARM_A16**	**ARM_A15**	**ARM_A14**	**ARM_A13**	**ARM_A12**	**ARM_A11**	**ARM_A10**	**ARM_A9**
8Mx16x1	0	x	x	0	0	0	x	0	0	BA1[M13]	BA0[M12]	A11[M11]	A10[M10]	A9[M9]	A8[M8]	A7[M7]	A6[M6]	A5[M5]	A4[M4]	A3[M3]	A2[M2]	A1[M1]	A0[M0]	0
16Mbyte	1	x	x	0	0	0	x	0	0	A11[M11]	A10[M10]	A9[M9]	A8[M8]	A7[M7]	A6[M6]	A5[M5]	A4[M4]	A3[M3]	A2[M2]	A1[M1]	A0[M0]	BA1[M13]	BA0[M12]	0
16Mx16x1	0	x	x	0	0	0	x	0	BA1[M14]	BA0[M13]	A12[M12]	A11[M11]	A10[M10]	A9[M9]	A8[M8]	A7[M7]	A6[M6]	A5[M5]	A4[M4]	A3[M3]	A2[M2]	A1[M1]	A0[M0]	0
		i.MX21 CSDx Base						**SDRAM Memory Address and Mode Register Bits**																

Table 17-40. i.MX21 SDRAM Mode Register Bits Relative to i.MX21 Internal Address Bus for x16 Configurations (continued)

		i.MX21 Internal Address Bus (Denoted as ARM_Axx)																						
Memory Configuration	IAM	ARM_A31	ARM_A30	ARM_A29	ARM_A28	ARM_A27	ARM_A26	ARM_A25	ARM_A24	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9
32Mbyte	1	x	x	0	0	0	x	0	A12[M12]	A11[M11]	A10[M10]	A9[M9]	A8[M8]	A7[M7]	A6[M6]	A5[M5]	A4[M4]	A3[M3]	A2[M2]	A1[M1]	A0[M0]	BA1[M14]	BA0[M13]	0
		i.MX21 CSDx Base						SDRAM Memory Address and Mode Register Bits																

Table 17-41. SDRAM Mode Register Bits Relative to i.MX21 Internal Address Bus for x32 Configurations

		i.MX21 Internal Address Bus (denoted as ARM_Axx)																						
Memory Configuration	IAM	ARM_A31	ARM_A30	ARM_A29	ARM_A28	ARM_A27	ARM_A26	ARM_A25	ARM_A24	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9
4Mx16x2	0	x	x	0	0	0	x	0	0	BA1[M13]	BA0[M12]	A11[M11]	A10[M10]	A9[M9]	A8[M8]	A7[M7]	A6[M6]	A5[M5]	A4[M4]	A3[M3]	A2[M2]	A1[M1]	A0[M0]	0
16Mbyte	1	x	x	0	0	0	x	0	0	A11[M11]	A10[M10]	A9[M9]	A8[M8]	A7[M7]	A6[M6]	A5[M5]	A4[M4]	A3[M3]	A2[M2]	A1[M1]	A0[M0]	BA1[M13]	BA0[M12]	0
8Mx16x2	0	x	x	0	0	0	x	0	BA1[M13]	BA0[M12]	A11[M11]	A10[M10]	A9[M9]	A8[M8]	A7[M7]	A6[M6]	A5[M5]	A4[M4]	A3[M3]	A2[M2]	A1[M1]	A0[M0]	0	0
32Mbyte	1	x	x	0	0	0	x	0	A11[M11]	A10[M10]	A9[M9]	A8[M8]	A7[M7]	A6[M6]	A5[M5]	A4[M4]	A3[M3]	A2[M2]	A1[M1]	A0[M0]	BA1[M13]	BA0[M12]	0	0
16Mx16x2	0	x	x	0	0	0	x	BA1[M14]	BA0[M13]	A12[M12]	A11[M11]	A10[M10]	A9[M9]	A8[M8]	A7[M7]	A6[M6]	A5[M5]	A4[M4]	A3[M3]	A2[M2]	A1[M1]	A0[M0]	0	0
		i.MX21 CSDx Base						SDRAM Memory Address and Mode Register Bits																

Table 17-41. SDRAM Mode Register Bits Relative to i.MX21 Internal Address Bus for x32 Configurations (continued)

Memory Configuration	IAM	i.MX21 Internal Address Bus (denoted as ARM_Axx)																								
		ARM_A31	ARM_A30	ARM_A29	ARM_A28	ARM_A27	ARM_A26	ARM_A25	ARM_A24	ARM_A23	ARM_A22	ARM_A21	ARM_A20	ARM_A19	ARM_A18	ARM_A17	ARM_A16	ARM_A15	ARM_A14	ARM_A13	ARM_A12	ARM_A11	ARM_A10	ARM_A9
64Mbyte	1	x	x	0	0	0	x	A12[M12]	A11[M11]	A10[M10]	A9[M9]	A8[M8]	A7[M7]	A6[M6]	A5[M5]	A4[M4]	A3[M3]	A2[M2]	A1[M1]	A0[M0]	BA1[M14]	BA0[M13]	0	0
2Mx32x1	0	x	x	0	0	0	x	0	0	0	BA1[M12]	BA0[M11]	A10[M10]	A9[M9]	A8[M8]	A7[M7]	A6[M6]	A5[M5]	A4[M4]	A3[M3]	A2[M2]	A1[M1]	A0[M0]	0
8Mbyte	1	x	x	0	0	0	x	0	0	0	A10[M10]	A9[M9]	A8[M8]	A7[M7]	A6[M6]	A5[M5]	A4[M4]	A3[M3]	A2[M2]	A1[M1]	A0[M0]	BA1[M12]	BA0[M11]	0
4Mx32x1	0	x	x	0	0	0	x	0	0	BA1[M13]	BA0[M12]	A11[M11]	A10[M10]	A9[M9]	A8[M8]	A7[M7]	A6[M6]	A5[M5]	A4[M4]	A3[M3]	A2[M2]	A1[M1]	A0[M0]	0
16Mbyte	1	x	x	0	0	0	x	0	0	A11[M11]	A10[M10]	A9[M9]	A8[M8]	A7[M7]	A6[M6]	A5[M5]	A4[M4]	A3[M3]	A2[M2]	A1[M1]	A0[M0]	BA1[M13]	BA0[M12]	0
8Mx32x1	0	x	x	0	0	0	x	0	BA1[M13]	BA0[M12]	A11[M11]	A10[M10]	A9[M9]	A8[M8]	A7[M7]	A6[M6]	A5[M5]	A4[M4]	A3[M3]	A2[M2]	A1[M1]	A0[M0]	0	0
32Mbyte	1	x	x	0	0	0	x	0	A11[M11]	A10[M10]	A9[M9]	A8[M8]	A7[M7]	A6[M6]	A5[M5]	A4[M4]	A3[M3]	A2[M2]	A1[M1]	A0[M0]	BA1[M13]	BA0[M12]	0	0
16Mx32x1	0	x	x	0	0	0	x	BA1[M14]	BA0[M13]	A12[M12]	A11[M11]	A10[M10]	A9[M9]	A8[M8]	A7[M7]	A6[M6]	A5[M5]	A4[M4]	A3[M3]	A2[M2]	A1[M1]	A0[M0]	0	0
64Mbyte	1	x	x	0	0	0	x	A12[M12]	A11[M11]	A10[M10]	A9[M9]	A8[M8]	A7[M7]	A6[M6]	A5[M5]	A4[M4]	A3[M3]	A2[M2]	A1[M1]	A0[M0]	BA1[M14]	BA0[M13]	0	0
		i.MX21 CSDx Base						SDRAM Memory Address and Mode Register Bits																

17.5.5 SDRAM Memory Refresh

SDRAM memory specifications generally specify an interval during which all rows in the device must be refreshed. The memory refresh requirements are outlined in Table 17-42. The SDRAM Controller refresh rate (SREFR field of the Control Register) is programmable to meet these varying requirements. Refresh must be enabled prior to storing data in the memory.

Table 17-42. SDRAM Memory Refresh

Device Size	Array Size	Refresh Interval	Refresh Rate	Requirement	SREFR Value
64 MBit	4096 rows	64 ms	1 row every 15.6 µseconds	2 rows refreshed during each 32 KHz clock	10
128 MBit	4096 rows	64 ms	1 row ever 15.6 µseconds	2 rows refreshed during each 32 KHz clock	10
256 MBit	8192 rows	64 ms	1 row every 7.8 µseconds	4 rows refreshed during each 32 KHz clock	11

NOTE

The memory datasheet should always be consulted to determine the correct refresh interval and array architecture (number of rows). Refresh clock rates other the nominal value require recalculation of the value to be programmed into REFR.

Chapter 18
Direct Memory Access Controller (DMAC)

The Direct Memory Access Controller (DMAC) of the i.MX21 provides 16 channels supporting linear memory, 2D memory, FIFO transfers to provide support for a wide variety of DMA operations.

The i.MX21 DMAC features are as follows:

- Sixteen channels support linear memory, 2D Memory, FIFO for both source and destination.
- DMA chaining for variable length buffer exchanges and high allowable interrupt latency requirement.
- Increment, decrement, and no-change support for source and destination addresses.
- Each channel is configurable to response to any of the 32 DMA request signals.
- Supports 8, 16, or 32-bit FIFO and memory port size data transfers.
- DMA burst length configurable up to a maximum of 16 words, 32 half-words, or 64 bytes for each channel.
- Bus utilization control for the channel that is not trigger by a DMA request.
- Burst time-out errors terminate the DMA cycle when the burst cannot be completed within a programmed time count.
- Buffer overflow error terminates the DMA cycle when the internal buffer receives more than 64 bytes of data.
- Transfer error terminates the DMA cycle when a transfer error is detected during a DMA burst.
- DMA request time-out errors are generated for channels that are triggered by DMA requests to interrupt the CPU when a DMA burst does not start on that channel after a programmed time count.
- Interrupts provided to the interrupt controller (and subsequently to the core) on bulk data transfer complete or transfer error.
- Each peripheral supporting DMA transfer generates a $\overline{\text{DMA\_REQ}}$ signal to the DMA controller, assuming that each FIFO has a unique system address and generates a dedicated $\overline{\text{dma\_req}}$ signal to the DMA controller. For example, a USB device with 8 end-points has 8 DMA request signals to the DMA if they all support DMA transfer.
- The DMA controller provides an acknowledge signal to the peripheral after a DMA burst is complete. This signal is sometimes used by the peripheral to clear status bits.
- Repeat data transfer function supports automatic USB host–USB device bulk/iso data stream transfer.
- Dedicated external DMA request and grant signal.

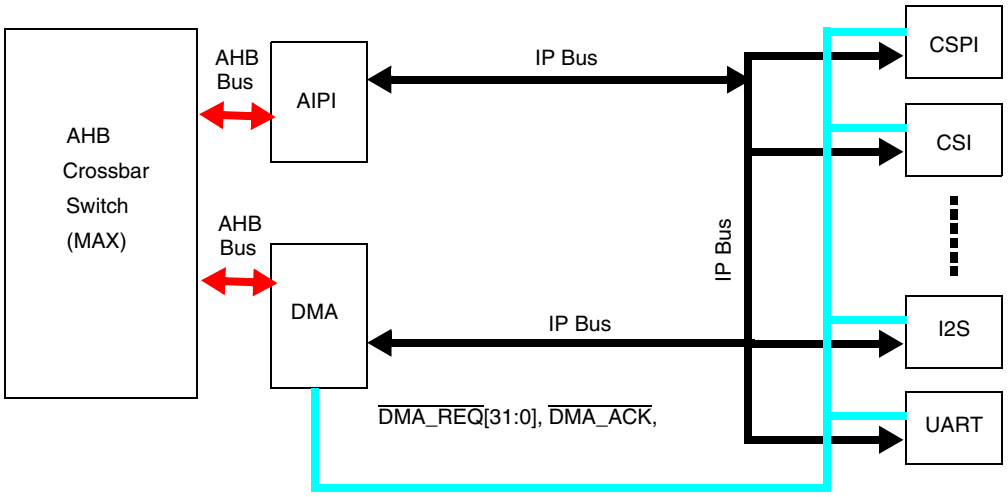


Figure 18-1. Block Diagram of the DMAC in i.MX21

The DMA has a FIFO for storing data read from the AHB Bus. This FIFO is 32 bits wide and 16 deep to store up to 64 bytes. This FIFO is common for all channels and is used for the active channel.

18.1 DMA Request and Acknowledge

Initiation of a DMA cycle can be done through software control (setting CEN = 1 and REN =0 in the channel control register) or by assertion of a DMA request (setting CEN = 1 and REN =1 in the channel control register). A DMA cycle consists of a number of DMA bursts depending on the burst length and the count register settings. Table 18-1 contains the DMA request map.

18.1.1 DMA Request

The DMA request is an active low signal asserted by the peripheral. The sampling of the signal is done when REN and CEN bits in Channel Control register are set and there is no other ongoing DMA transfer on the AHB bus. There is no configurable priority associated with any request. However, the 16 channels have a fixed priority; channel 15 has the highest priority and channel 0 has the lowest priority. The priority of any request depends on the channel number to which the request is mapped (through Request Source Select register settings). The DMAC does not store the DMA request inputs, it processes on the highest priority channel request out of the asserted channel requests (when no other transfer is taking place). A peripheral must keep the request asserted until it is serviced by the DMAC. There are 32 input DMA request signals available. One DMA request will initiate one DMA burst. Once a DMA burst has started, the DMA request can be de-asserted by the peripheral. The peripheral should de-assert the DMA request based on data read from or written into the peripheral. If the request is not de-asserted till the end of the DMA burst, it can initiate another DMA burst.

18.1.2 External DMA Request and Grant

After assertion of External DMA Request the DMA burst will start when the corresponding DMA channel becomes the current highest priority channel. The External DMA Request should be kept asserted until it is serviced by the DMAC. One External request will initiate at least one DMA burst.

The output External Grant signal from the DMAC is an active-low signal. This signal will be asserted during the time when a DMA burst is ongoing for an External DMA Request, when the following conditions are true:

- The DMA channel for which the DMA burst is ongoing has request source as external DMA Request (as per RSSR settings).
- REN and CEN bit of this channel are set
- External DMA Request is asserted

Once the grant is asserted the External DMA Request will not be sampled until completion of the DMA burst. The priority of the external request will become low, for the next consecutive burst, if another DMA request signal is asserted.

NOTE

Refer to Chapter 2, “Signal Descriptions and Pin Assignments,” for signal multiplexing of the EXT_DMAREQ and EXT_DMAGRANT signals.

The waveforms are shown for the worst case—that is, smallest burst (1 byte read/write). Minimum and maximum timings for the External request and External grant signal are present in the data sheet.

Figure 18-2 shows the minimum time for which the External Grant signal remains asserted if External DMA request is de-asserted immediately after sensing grant signal active.

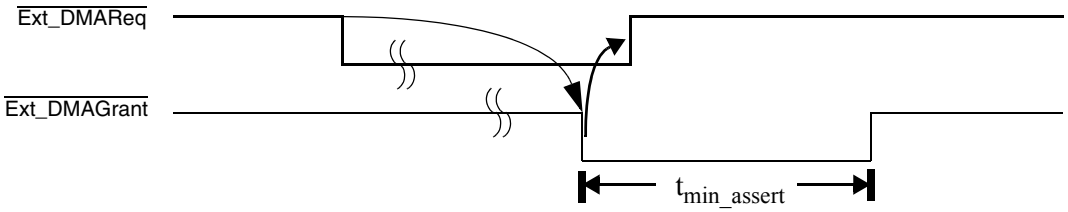
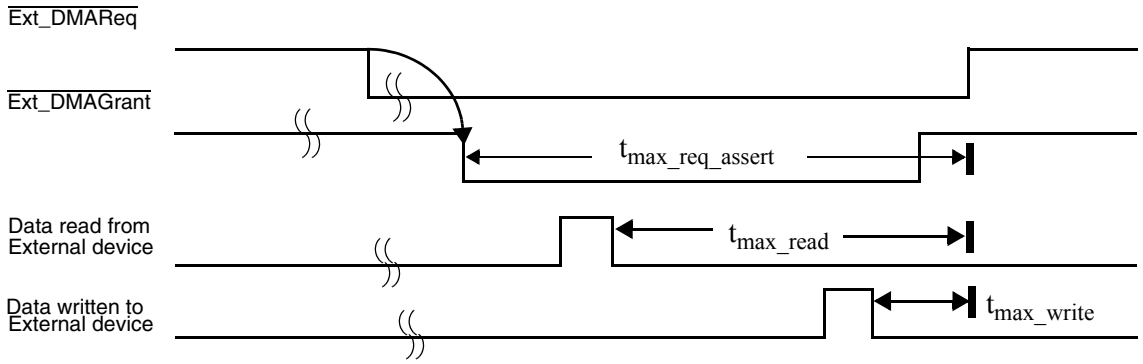


Figure 18-2. Assertion of DMA External Grant Signal

Figure 18-3 shows the safe max. time for which External DMA request can be kept asserted, after sensing grant signal active such that a new burst is not initiated.



NOTE: Assuming in worst case the data is read/written from/to External device as per the above waveform.

Figure 18-3. Safe Maximum Timings for External Request De-Assertion

18.2 DMA Request Mapping

Table 18-1 shows the connection of the requests from various modules in the i.MX21 to the DMA Request input of the DMA controller.

Table 18-1. DMA Request Mapping

DMA Request Number	Module Assigned	Channel Name
$\overline{\text{DMA\_REQ[31]}}$	CSI	CSI_RX_FIFO
$\overline{\text{DMA\_REQ[30]}}$	CSI	CSI_STAT_FIFO
$\overline{\text{DMA\_REQ[29]}}$	BMI	BMI_RX_FIFO
$\overline{\text{DMA\_REQ[28]}}$	BMI	BMI_TX_FIFO
$\overline{\text{DMA\_REQ[27]}}$	UART1	UART1_TX_FIFO
$\overline{\text{DMA\_REQ[26]}}$	UART1	UART1_RX_FIFO
$\overline{\text{DMA\_REQ[25]}}$	UART2	UART2_TX_FIFO
$\overline{\text{DMA\_REQ[24]}}$	UART2	UART2_RX_FIFO
$\overline{\text{DMA\_REQ[23]}}$	UART3	UART3_TX_FIFO
$\overline{\text{DMA\_REQ[22]}}$	UART3	UART3_RX_FIFO
$\overline{\text{DMA\_REQ[21]}}$	UART4	UART4_TX_FIFO
$\overline{\text{DMA\_REQ[20]}}$	UART4	UART4_RX_FIFO
$\overline{\text{DMA\_REQ[19]}}$	CSPI1	CSPI1_TX_FIFO
$\overline{\text{DMA\_REQ[18]}}$	CSPI1	CSPI1_RX_FIFO
$\overline{\text{DMA\_REQ[17]}}$	CSPI2	CSPI2_TX_FIFO
$\overline{\text{DMA\_REQ[16]}}$	CSPI2	CSPI2_RX_FIFO
$\overline{\text{DMA\_REQ[15]}}$	SSI1	SSI1_TX1_FIFO
$\overline{\text{DMA\_REQ[14]}}$	SSI1	SSI1_RX1_FIFO
$\overline{\text{DMA\_REQ[13]}}$	SSI1	SSI1_TX0_FIFO
$\overline{\text{DMA\_REQ[12]}}$	SSI1	SSI1_RX0_FIFO
$\overline{\text{DMA\_REQ[11]}}$	SSI2	SSI2_TX1_FIFO
$\overline{\text{DMA\_REQ[10]}}$	SSI2	SSI2_RX1_FIFO
$\overline{\text{DMA\_REQ[9]}}$	SSI2	SSI2_TX0_FIFO
$\overline{\text{DMA\_REQ[8]}}$	SSI2	SSI2_RX0_FIFO
$\overline{\text{DMA\_REQ[7]}}$	SDHC1	SDHC1
$\overline{\text{DMA\_REQ[6]}}$	SDHC2	SDHC2
$\overline{\text{DMA\_REQ[5]}}$	FIRI	FIRI_TX_FIFO
$\overline{\text{DMA\_REQ[4]}}$	FIRI	FIRI_RX_FIFO
$\overline{\text{DMA\_REQ[3]}}$	External DMA request	–
$\overline{\text{DMA\_REQ[2]}}$	CSPI3	CSPI3_TX_FIFO
$\overline{\text{DMA\_REQ[1]}}$	CSPI3	CSPI3_RX_FIFO
$\overline{\text{DMA\_REQ[0]}}$	Reserved	–

18.3 Programming Model

The DMA Controller module registers are 32-bit registers. The registers are divided into four groups according to the register functions as follows.

- General registers for all functional blocks (see Section 18.3.1 on page -10)
- 2D memory registers to control the display width and the x and y of the window (see Section 18.3.2 on page -17)
- Channel registers to control and configure channels 0–15 (see Section 18.3.3 on page -21)
- Test Registers

The base address of DMA Controller for i.MX21 is 0x10001000. Table 18-2 summarizes the registers and offset addresses.

Table 18-2. DMAC Register Summary

Name		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
DCR (0x000)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	DAM	0	DEN
	W															DRST	
DISR (0x004)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH 6	CH5	CH4	CH3	CH2	CH1	CH0
	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
DIMR (0x008)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH 6	CH5	CH4	CH3	CH2	CH1	CH0
	W																
DBTOSR (0x00C)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH 6	CH5	CH4	CH3	CH2	CH1	CH0
	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
DRTOSR (0x010)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH 6	CH5	CH4	CH3	CH2	CH1	CH0
	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
DSESR (0x014)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH 6	CH5	CH4	CH3	CH2	CH1	CH0
	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
DBOSR (0x018)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH 6	CH5	CH4	CH3	CH2	CH1	CH0
	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C

Table 18-2. DMAC Register Summary (continued)

Name		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
DBTOCR (0x01C)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	EN	CNT [14: 0]														
	W																
WSRA (0x040)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	WS [15: 0]															
	W																
XSRA (0x044)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	XS [15: 0]															
	W																
YSRA (0x048)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	YS [15: 0]															
	W																
WSRB (0x04C)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	WS [15: 0]															
	W																
XSRB (0x050)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	XS [15: 0]															
	W																
YSRB (0x054)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	YS [15: 0]															
	W																
SAR0 (0x080) SAR1 (0x0C0) SAR2 (0x100) SAR3 (0x140) SAR4 (0x180) SAR5 (0x1C0) SAR6 (0x200) SAR7 (0x240) SAR8 (0x280) SAR9 (0x2C0) SAR10 (0x300) SAR11 (0x340) SAR12 (0x380) SAR13 (0x3C0) SAR14 (0x400) SAR15 (0x440)	R	SA [31: 16]															
	W																
	R	SA [15: 0]															
	W																

Table 18-2. DMAC Register Summary (continued)

Name		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
DAR0 (0x084) DAR1 (0x0C4) DAR2 (0x104) DAR3 (0x144) DAR4 (0x184) DAR5 (0x1C4) DAR6 (0x204) DAR7 (0x244) DAR8 (0x284) DAR9 (0x2C4) DAR10 (0x304) DAR11 (0x344) DAR12 (0x384) DAR13 (0x3C4) DAR14 (0x404) DAR15 (0x444)	R	DA [31: 16]															
	W																
	R	DA [15: 0]															
	W																
CNTR0 (0x088) CNTR1 (0x0C8) CNTR2 (0x108) CNTR3 (0x148) CNTR4 (0x188) CNTR5(0x1C8) CNTR6 (0x208) CNTR7 (0x248) CNTR8 (0x288) CNTR9(0x2C8) CNTR10 (0x308) CNTR11 (0x348) CNTR12 (0x388) CNTR13 (0x3C8) CNTR14 (0x408) CNTR15 (0x448)	R	0	0	0	0	0	0	0	0	CNT [23: 16]							
	W																
	R	CNT [15: 0]															
	W																
CCR0 (0x08C) CCR1 (0x0CC) CCR2 (0x10C) CCR3 (0x14C) CCR4 (0x18C) CCR5(0x1CC) CCR6(0x20C) CCR7(0x24C) CCR8(0x28C) CCR9 (0x2CC) CCR10 (0x30C) CCR11 (0x34C) CCR12 (0x38C) CCR13 (0x3CC) CCR14 (0x40C) CCR15 (0x44C)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	0	ACRPT	DMOD		SMOD		MDIR	MSEL	DSIZ		SSIZ		REN	RPT	0	CEN
	W															FRC	

Table 18-2. DMAC Register Summary (continued)

Name		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
RSSR0 (0x090) RSSR1 (0x0D0) RSSR2 (0x110) RSSR3 (0x150) RSSR4 (0x190) RSSR5 (0x1D0) RSSR6 (0x210) RSSR7 (0x250) RSSR8 (0x290) RSSR9 (0x2D0) RSSR10 (0x310) RSSR11 (0x350) RSSR12 (0x390) RSSR13 (0x3D0) RSSR14 (0x410) RSSR15 (0x450)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	0	0	0	0	0	0	0	0	0	0	0	RSS [4: 0]				
	W																
BLR0 (0x094) BLR1 (0x0D4) BLR2 (0x114) BLR3 (0x154) BLR4 (0x194) BLR5 (0x1D4) BLR6 (0x214) BLR7 (0x254) BLR8 (0x294) BLR9 (0x2D4) BLR10 (0x314) BLR11 (0x354) BLR12 (0x394) BLR13 (0x3D4) BLR14 (0x414) BLR15 (0x454)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	0	0	0	0	0	0	0	0	0	0	BL [5: 0]					
	W																
RTOR0 (0x098) RTOR1 (0x0D8) RTOR2 (0x118) RTOR3 (0x158) RTOR4 (0x198) RTOR5 (0x1D8) RTOR6 (0x218) RTOR7 (0x258) RTOR8 (0x298) RTOR9 (0x2D8) RTOR10 (0x318) RTOR11 (0x358) RTOR12 (0x398) RTOR13 (0x3D8) RTOR14 (0x418) RTOR15 (0x458)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	EN	CLK	PSC	CNT [12: 0]												
	W																

Table 18-2. DMAC Register Summary (continued)

Name		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
BUCR0 (0x098) BUCR1 (0x0D8) BUCR2 (0x118) BUCR3 (0x158) BUCR4 (0x198) BUCR5 (0x1D8) BUCR6 (0x218) BUCR7 (0x258) BUCR8 (0x298) BUCR9 (0x2D8) BUCR10 (0x318) BUCR11 (0x358) BUCR12 (0x398) BUCR13 (0x3D8) BUCR14 (0x418) BUCR15 (0x458)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	BU_CNT [15: 0]															
	W																
CCNR0 (0x09C) CCNR1 (0x0DC) CCNR2 (0x11C) CCNR3 (0x15C) CCNR4 (0x19C) CCNR5 (0x1DC) CCNR6 (0x21C) CCNR7 (0x25C) CCNR8 (0x29C) CCNR9 (0x2DC) CCNR10 (0x31C) CCNR11 (0x35C) CCNR12 (0x39C) CCNR13 (0x3DC) CCNR14 (0x41C) CCNR15 (0x45C)	R	0	0	0	0	0	0	0	0	CCNR[23:16]							
	W																
	R	CCNR [15: 0]															
	W																

18.3.1 General Registers

This section describes the function of the general registers.

18.3.1.1 DMA Control Register

The DMA Control Register (DCR) controls the input of the system clock and the resetting of the DMA module.

DCR — DMA Control Register — Addr 0x10001000

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														DAM	DRST	DEN
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	rw	Slf Clr	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-3. DMA Control Register Description

Name	Description	Settings
Reserved Bits 31–3	Reserved—These bits are reserved and should read 0.	
DAM Bit 2	**DMA Access Mode**—Specifies user or privileged access to be performed by DMA.	1 = User access 0 = Privileged access
DRST Bit 1	**DMA Soft Reset**—Generates a 3-cycle reset pulse that resets the entire DMA module, bringing the module to its reset condition. DRST always reads 0.	0 = No effect 1 = Generates a 3-cycle reset pulse
DEN Bit 0	**DMA Enable**—Enables/Disables the system clock to the DMA module**.** However the bit is not used for clock gating in i.MX21 as the clock is controlled from CRM in i.MX21.	0 = DMA disable 1 = DMA enable

18.3.1.2 DMA Interrupt Status Register

The DMA Interrupt Status Register (DISR) contains the interrupt status of each channel in the DMAC. The status bit is set whenever the corresponding DMA channel data transfer is complete. When any bit in the DMA Interrupt Status Register (DISR) is set and the corresponding bit in the interrupt mask register is cleared, a dma_int is asserted to the interrupt controller (AITC). When an interrupt occurs, the interrupt service routine must check the DISR to determine the interrupting channel. Clear each bit by writing a value of 1 to it.

DISR — DMA Interrupt Status Register — Addr 0x10001004

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH15	CH 14	CH 13	CH 12	CH 11	CH 10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
TYPE	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-4. DMA Interrupt Status Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
CH15–CH0 Bits 15–0	**Channel 15 to 0 Interrupt Status**—Indicates the interrupt status for each DMA channel.	0 = No interrupt 1 = Interrupt is pending

18.3.1.3 DMA Interrupt Mask Register

The DMA Interrupt Mask Register (DIMR) masks both normal interrupts and error interrupts generated by the corresponding channel. There is one control bit for each channel. When an interrupt is masked, the interrupt controller does not generate an interrupt request to the AITC, however the status of the interrupt can be observed from the interrupt status register, burst time-out status register, request time-out status register, or the transfer error status register. At reset, all the interrupts are masked and all the bits in this register are set to 1.

DIMR — DMA Interrupt Mask Register — Addr 0x10001008

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH15	CH 14	CH 13	CH 12	CH 11	CH 10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0xFFFF															

Table 18-5. DMA Interrupt Mask Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
CH15–CH0 Bits 15–0	**Channel 15 to 0**—Controls the interrupts for each DMA channel.	0 = Enables interrupts 1 = Disables interrupts

18.3.1.4 DMA Burst Time-Out Status Register

A burst time-out is set when a DMA burst cannot be completed within the number of clock cycles specified in the DMA Burst Time-Out Control Register (DBTOCR) of the channel. When any bit is set in this register and the corresponding bit in the interrupt mask register is cleared, a DMA Error interrupt is asserted to the interrupt controller (AITC). The DMA burst time-out status register (DBTOSR) indicates the channel, if any, that is currently being serviced and whether a burst time-out was detected. Each bit is cleared by writing 1 to it.

DBTOSR — DMA Burst Time-Out Status Register — Addr 0x1000100C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH15	CH 14	CH 13	CH 12	CH 11	CH 10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
TYPE	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-6. DMA Burst Time-Out Status Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
CH15–CH0 Bits 15–0	**Channel 15 to 0**—Indicates the burst time-out status of each DMA channel.	0 = No burst time-out 1 = Burst time-out

18.3.1.5 DMA Request Time-Out Status Register

A DMA request time-out is set when there is no DMA burst started on the channel (when REN =1, either due to no DMA Request or the DMA channel not acquiring the bus) within the pre-assigned number of clock cycles specified in the Request Time-Out control register (RTOR) for the channel. When any bit is set in this register and the corresponding bit in the interrupt mask register is cleared, a DMA Error Interrupt is asserted to the AITC. The DMA Request Time-Out Status Register (DRTOSR) indicates the enabled channel, if any, that detected a DMA request time-out. Clear each bit by writing 1 to it.

DRTOSR — DMA Request Time-Out Status Register — Addr 0x10001010

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH15	CH 14	CH 13	CH 12	CH 11	CH 10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
TYPE	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-7. DMA Request Time-Out Status Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
CH15–CH0 Bits 15–0	**Channel 15 to 0**—Indicates the request time-out status of each DMA channel.	0 = No DMA request time-out 1 = DMA request time-out

18.3.1.6 DMA Transfer Error Status Register

A DMA transfer error is set when the DMA data transfer results in an error. When any bit is set in this register and the corresponding bit in the interrupt mask register is cleared, a DMA Error Interrupt is asserted to the AITC. The DMA Transfer Error Status Register (DSESR) indicates the channel, if any, detected a transfer error during a DMA burst. Clear each bit by writing 1 to it.

DSESR — DMA Transfer Error Status Register — Addr 0x10001014

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH15	CH 14	CH 13	CH 12	CH 11	CH 10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
TYPE	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-8. DMA Transfer Error Status Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
CH15–CH0 Bits 15–0	**Channel 15 to 0**—Indicates the DMA transfer error status of each DMA channel.	0 = No transfer error 1 = Transfer error

18.3.1.7 DMA Buffer Overflow Status Register

The DMA Buffer Overflow Status Register (DBOSR) indicates whether the internal FIFO buffer of the DMA Controller overflowed during a data transfer. Before a channel can be enabled for DMA, the corresponding bit in this register must be cleared. When any bit in this register is set and the corresponding bit in the interrupt mask register is cleared, a DMA Error Interrupt is asserted to the AITC.

DBOSR — DMA Buffer Overflow Status Register — Addr 0x10001018

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH15	CH 14	CH 13	CH 12	CH 11	CH 10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
TYPE	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-9. DMA Buffer Overflow Status Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
CH15–CH0 Bits 15–0	**Channel 15 to 0**—Indicates the buffer overflow error status of each DMA channel.	0 = No buffer overflow occurred 1 = Buffer overflow occurred

18.3.1.8 DMA Burst Time-Out Control Register

This register sets the time-out for a DMA burst (common for all DMA channels), so that the DMA controller can release the AHB and IP buses in the event of an error. An internal counter starts counting when a DMA burst starts, and resets to zero when the burst is completed. When the counter reaches the count value set in the register, it asserts an interrupt and sets the corresponding error bit in the DMA Burst Time-Out Status register. The system clock is used as the input clock to the counter.

DBTOCR — DMA Burst Time-Out Control Register — Addr 0x1000101C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EN	CNT														
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-10. DMA Burst Time-Out Control Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
EN Bit 15	**Enable**—Enables/Disables the burst time-out.	0 = Disables burst time-out 1 = Enables burst time-out
CNT Bits 14–0	**Count**—This count is the number of system clock cycles to be used for the time-out value	

18.3.2 2D Memory Registers (A and B)

There are two sets of 2D memory registers that allow every channel to select any register set to define the respective 2D memory size. Each data transfer performed by DMA is strictly as per Source and destination sizes specified in the Channel Control Register (this is valid for ALL 4 modes—that is, Linear memory,2D memory, FIFO mode.

In the case of a transfer to, or a transfer from 2D Memory, the Channel Count register value is ignored and number of bytes transferred is equal to the size of the 2D Memory. The Size of the 2D Memory is computed as follows.

Size (in number of bytes) = No of bytes per row (value in X register) * No. of rows (Value in Y Register)

At a time any number of channels can be programmed for 2D Memory (even all 16 Channels). 2D Memory can be selected for a channel as source or destination or even for both source and destination. In the later condition, only the selected set of the 2D Registers (selected as per the setting of the MSEL bit in the Channel Control Register) is used for both source and destination.

The advantage of having 2 sets of registers, which are usable by all DMA channels, is that this allows the developer to have two different window size settings for 2D memory. Each channel can be programmed to use any one of the 2 settings.

In Figure 18-4, the shaded portion shows the data transfer zone in the 2D memory for X= 4, Y = 4, and W (display size) = 6 with memory increment option and starting address 0x001.

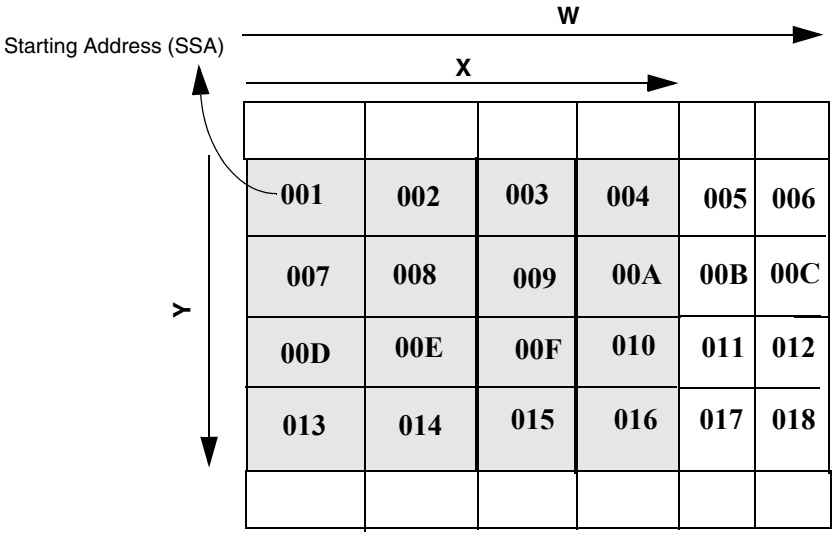


Figure 18-4. 2D Memory Increment Diagram

In Figure 18-5, the shaded portion shows the data transfer zone in the 2D memory for X= 4, Y = 5, and W = 6 with memory decrement option and starting address as 0x11C.

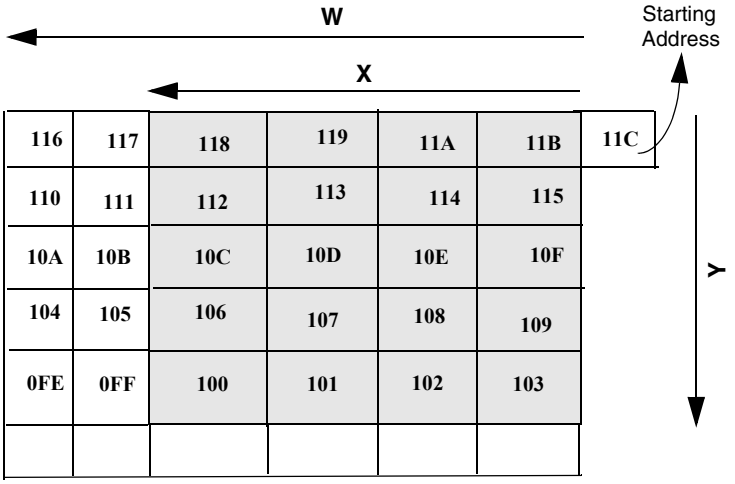


Figure 18-5. 2D Memory Decrement Diagram

18.3.2.1 W-Size Registers

The W-Size registers (WSRA and WSRB) define the number of bytes that make up the width of the display. This allows the DMA controller to calculate the next starting address of another row by adding the source/destination address to the contents of the W-Size register.

WSRA WSRB	W-Size Register A W-Size Register B	Addr 0x10001040 0x1000104C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WS															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-11. W-Size Registers Description

Name	Description
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.
WS Bits 15–0	**W-Size**—Contains the number of bytes that make up the display width. W and X must follow the relation: W ≥ X W and Access Size must follow the relation: W ≥ access size. Wsize needs to be a multiple of Source or Destination Access size whichever is a 2D memory.

18.3.2.2 X-Size Registers

The X-Size registers (XSRA and XSRB) contain the number of bytes per row of the window. The value of this register is used by the DMA controller to determine when to jump to the next row.

XSRA XSRB	X-Size Register A X-Size Register B												Addr 0x10001044 0x10001050			
BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	XS															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-12. X-Size Registers Description

Name	Description
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.
XS Bits 15–0	**X-Size**—Contains the number of bytes per row that define the X-Size of the 2D memory. The value in the X Register should follow the following 2 rules: • X ≥ Burst Length (BL) • X/ BL = Whole number.

18.3.2.3 Y-Size Registers

The Y-Size registers (YSRA and YSRB) contain the number of rows in the 2D memory window. This setting is used by the DMA controller to calculate the total size of the transfer.

YSRA YSRB	Y-Size Register A Y-Size Register B												Addr 0x10001048 0x10001054			
BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	YS															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-13. Y-Size Registers Description

Name	Description
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.
YS Bits 15–0	**Y-Size**—Contains the number of rows that make up the 2D memory window.

18.3.3 Channel Registers

Channels 0 to 15 support linear memory, 2D memory, FIFO transfer.

The DMA request ($\overline{\text{dma\_req}}$ [31:0]) signals do not have any configurable priority. The only priority available is the priority that is defined for each channel: channel 15 has the highest priority and channel 0 has the lowest priority. The channel priority is used only when more than one request occurs at the same time. Otherwise, channels are serviced on a first come, first serve basis.

Each channel generates a normal interrupt to the interrupt handler when the data count reaches the selected value. Each channel generates an error interrupt to the interrupt handler when any of the following conditions exist:

- A DMA request time-out is true
- A DMA burst time-out is true during a burst cycle
- The internal buffer overflows during a burst cycle
- A transfer error acknowledge is asserted during a burst cycle

18.3.3.1 Channel Source Address Registers

Each of the channel source address registers contain the source address for the DMA cycle. The implementation must ensure that the source address register's value is stored internally before use to allow the software to modify the register value for DMA chaining (see section 18.4 "DMA Chaining", on page 35). The value should be stored when the CEN bit is set or at the end of the transfer when the RPT bit is found set (before initiating the new transfer). If the memory direction bit (MDIR) in the channel control register (CCR) is clear (indicating a memory address increment), then the channel source address register contains the starting address of the memory block. If MDIR is set (indicating a memory address decrement), then the channel source address register contains the ending address of the memory block.

SAR0	Channel 0 Source Address Register	0x10001080
SAR1	Channel 1 Source Address Register	0x100010C0
SAR2	Channel 2 Source Address Register	0x10001100
SAR3	Channel 3 Source Address Register	0x10001140
SAR4	Channel 4 Source Address Register	0x10001180
SAR5	Channel 5 Source Address Register	0x100011C0
SAR6	Channel 6 Source Address Register	0x10001200
SAR7	Channel 7 Source Address Register	0x10001240
SAR8	Channel 8 Source Address Register	0x10001280
SAR9	Channel 9 Source Address Register	0x100012C0
SAR10	Channel 10 Source Address Register	0x10001300
SAR11	Channel 11 Source Address Register	0x10001340
SAR12	Channel 12 Source Address Register	0x10001380
SAR13	Channel 13 Source Address Register	0x100013C0
SAR14	Channel 14 Source Address Register	0x10001400
SAR15	Channel 15 Source Address Register	0x10001440

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SA															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SA															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-14. Channel Source Address Register Description

Name	Description
SA Bits 31–0	**Source Address**—Contains the source address from where data is read during a DMA transfer. DMA will not perform misaligned accesses. That is to say, for 32-bit transfers, the lower two bits of this address are ignored. 8-bit accesses begin from the address in this register. Software must take care if the system does not support non-word aligned accesses in this case.

18.3.3.2 Destination Address Registers

Each of the destination address registers (DARx) contain the destination address for a DMA cycle. The implementation needs to ensure that the Destination Address Register's value is stored internally before use to allow the software to modify the register value for DMA Chaining (see section 18.4 "DMA Chaining", on page 35). The value should be stored when the CEN bit is set or at the end of the transfer when the RPT bit is found set (before initiating the new transfer). If the memory direction bit (MDIR) in the channel control register (CCR) is clear (indicating a memory address increment), then the destination address register contains the starting address of the memory block. If MDIR is set (indicating a memory address decrement), then the destination address register contains the ending address of the memory block.

		Addr
DAR0	Channel 0 Destination Address Register	0x10001084
DAR1	Channel 1 Destination Address Register	0x100010C4
DAR2	Channel 2 Destination Address Register	0x10001104
DAR3	Channel 3 Destination Address Register	0x10001144
DAR4	Channel 4 Destination Address Register	0x10001184
DAR5	Channel 5 Destination Address Register	0x100011C4
DAR6	Channel 6 Destination Address Register	0x10001204
DAR7	Channel 7 Destination Address Register	0x10001244
DAR8	Channel 8 Destination Address Register	0x10001284
DAR9	Channel 9 Destination Address Register	0x100012C4
DAR10	Channel 10 Destination Address Register	0x10001304
DAR11	Channel 11 Destination Address Register	0x10001344
DAR12	Channel 12 Destination Address Register	0x10001384
DAR13	Channel 13 Destination Address Register	0x100013C4
DAR14	Channel 14 Destination Address Register	0x10001404
DAR15	Channel 15 Destination Address Register	0x10001444

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DA															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DA															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-15. Channel Destination Address Registers Description

Name	Description
DA Bits 31–0	**Destination Address**—Contains the destination address to which data is written to during a DMA transfer. The DMAC will not perform misaligned accesses. That is, for a 32-bit transfers the lower two bits of this address are ignored. 8-bit accesses begin from the address in this register. Software must take care if the system does not support non-word aligned accesses in this case.

18.3.3.3 Channel Count Registers

The DEN bit in the DCR should be set to enable write to this register.

The implementation needs to ensure that the Count register's value is stored internally before use to allow the software to modify the register value for DMA Chaining (see section 18.4 "DMA Chaining", on page 35). The value should be stored in an Internal Count Register when the CEN bit is set or at the end of the transfer when the RPT bit is found set (before initiating the new transfer).

Each of the channel count registers (CNTRx) contain the number of bytes of data to be transferred. There is an internal counter that counts up (number of bytes—4 for word, 2 for halfword and 1 for byte) for every DMA transfer. The internal counter is compared with the Internal Count Register after every transfer. When the counter value matches with the register value, the channel is disabled until the CEN bit is cleared and set again, or the RPT bit in the corresponding channel control register is set to 1. The internal counter is reset to 0 when the channel is enabled again.

The length of the last DMA burst can be shorter than the regular burst length specified in the burst length register. However, when data is transferred out from an I/O FIFO and the last burst is less than BL, the I/O device must generate a DMA request for the last transfer. When data is transferred to an I/O FIFO and the last burst is less than BL, only the remaining number of data is transferred.

		Addr
CNTR0	Channel 0 Count Register	0x10001088
CNTR1	Channel 1 Count Register	0x100010C8
CNTR2	Channel 2 Count Register	0x10001108
CNTR3	Channel 3 Count Register	0x10001148
CNTR4	Channel 4 Count Register	0x10001188
CNTR5	Channel 5 Count Register	0x100011C8
CNTR6	Channel 6 Count Register	0x10001208
CNTR7	Channel 7 Count Register	0x10001248
CNTR8	Channel 8 Count Register	0x10001288
CNTR9	Channel 9 Count Register	0x100012C8
CNTR10	Channel 10 Count Register	0x10001308
CNTR11	Channel 11 Count Register	0x10001348
CNTR12	Channel 12 Count Register	0x10001388
CNTR13	Channel 13 Count Register	0x100013C8
CNTR14	Channel 14 Count Register	0x10001408
CNTR15	Channel 15 Count Register	0x10001448

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									CNT							
TYPE	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNT															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-16. Channel Count Registers Description

Name	Description
Reserved Bits 31–24	Reserved—These bits are reserved and should read 0.
CNT Bits 23–0	**Count**—Contains the number of bytes of data to be transferred during a DMA cycle.

18.3.3.4 Channel Control Registers

Each of the channel control registers (CCRx) controls and displays the status of a DMA channel operation. The DMA controller has the capability to perform burst transfers of byte and half word data types while the SDRAM controller and EIM support is restricted to burst transfers of word (32-bit) data types. Therefore, when using the DMA in conjunction with the SDRAM controller and EIM, ensure that all burst transfers to/from the SDRAM controller and EIM are of word data types. This is configured in the DMA Channel Control Register. When choosing SDRAM memory as the source or destination address, set the SDRAMC and EIM as a 32-bit port.

		Addr
CCR0	Channel 0 Control Register	0x1000108C
CCR1	Channel 1 Control Register	0x100010CC
CCR2	Channel 2 Control Register	0x1000110C
CCR3	Channel 3 Control Register	0x1000114C
CCR4	Channel 4 Control Register	0x1000118C
CCR5	Channel 5 Control Register	0x100011CC
CCR6	Channel 6 Control Register	0x1000120C
CCR7	Channel 7 Control Register	0x1000124C
CCR8	Channel 8 Control Register	0x1000128C
CCR9	Channel 9 Control Register	0x100012CC
CCR10	Channel 10 Control Register	0x1000130C
CCR11	Channel 11 Control Register	0x1000134C
CCR12	Channel 12 Control Register	0x1000138C
CCR13	Channel 13 Control Register	0x100013CC
CCR14	Channel 14 Control Register	0x1000140C
CCR15	Channel 15 Control Register	0x1000144C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		ACRPT	DMOD		SMOD		MDIR	MSEL	DSIZ		SSIZ		REN	RPT	FRC	CEN
TYPE	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-17. Channel Control Registers Description

Name	Description	Settings
Reserved Bits 31–15	Reserved—These bits are reserved and should read 0.	
ACRPT Bits 14	**Auto Clear RPT**—This bit is to be sampled at the end of the transfer along with the RPT bit. When this bit and RPT are set, a new transfer is initiated and RPT is reset before issuing any interrupts.	0 = Do not modify RPT 1 = Reset RPT at end of current transfer.

Table 18-17. Channel Control Registers Description (continued)

Name	Description	Settings
DMOD Bits 13–12	**Destination Mode**—Selects the destination transfer mode.	00 = Linear memory 01 = 2D memory 10 = FIFO 11 = Reserved
SMOD Bits 11–10	**Source Mode**—Selects the source transfer mode.	00 = Linear memory 01 = 2D memory 10 = FIFO 11 = Reserved
MDIR Bit 9	**Memory Direction**—Selects the memory address direction. **Note:** When address increment is chosen, the data transfer starts from the values in the source and destination address registers. When address decrement is chosen, the data transfer will be done till the addresses mentioned in source and destination address register—that is, no data read or write will be done at the address mentioned in source and destination address registers.	0 = Memory address increment 1 = Memory address decrement
MSEL Bit 8	**Memory Select**—Selects the 2D memory register set when either source and/or destination is programmed to 2D memory mode.	0 = 2D memory register set A selected 1 = 2D memory register set B selected
DSIZ Bits 7–6	**Destination Size**—Selects the destination size of a data transfer. 1. If the number of bytes to be written is less than the DSIZ setting, then only that many bytes will be valid in the DMA write cycle to the AHB. However all DMA write cycles to the destination will be of DSIZ size. 2. DMA always writes data as per DSIZ in all modes —that is, Linear memory, 2D memory and FIFO mode.	00 = 32-bit destination port 01 = 8-bit destination port 10 = 16-bit destination port 11 = Reserved
SSIZ Bits 5–4	**Source Size**—Selects the source size of data transfer. 1. If the number of bytes to be read is less than the SSIZ setting, then only that many bytes will be used by the DMA. However, all DMA read cycles to the source will be of “SSIZ” size. 2. DMA always reads data as per SSIZ in all modes—that is, Linear memory, 2D memory and FIFO mode.	00 = 32-bit source port 01 = 8-bit source port 10 = 16-bit source port 11 = Reserved
REN Bit 3	**Request Enable**—Enables/Disables the DMA request signal. When REN is set, the DMA burst is initiated by the $\overline{\text{dma\_req}}$ signal from the I/O FIFO. When REN is cleared, DMA transfer is initiated by CEN.	0 = Disables the DMA request signal (when the peripheral asserts a DMA request, no DMA transfer is triggered); DMA transfer is initiated by CEN only 1 = Enables the DMA request signal (when the peripheral asserts a DMA request, a DMA transfer is triggered)

Table 18-17. Channel Control Registers Description (continued)

Name	Description	Settings
RPT Bit 2	**Repeat**—This is a status/control bit. The software has a priority and can write to this bit at any time. This bit Enables/Disables the data transfer repeat function. When enabled and when the counter reaches the value set in Internal Count Register: a. The Source addr, Destination addr and Count Register values are stored (reloaded) for the next DMA Burst. b. If the ACRPT bit is set, RPT bit is cleared. c. The next DMA cycle is enabled. After this an interrupt is asserted, if the corresponding channel bit in the Interrupt Mask Register is cleared. Data transfer is carried out continuously until the channel is disabled or it completes the last DMA burst after RPT is cleared. The status information in this bit is that it gets cleared when ACRPT is set as described above. **Note:** Implementation must ensure that the RPT bit is sampled only at the time the counter reaches the value in the Internal Count Register before asserting the interrupt. The RPT bit should be allowed to be modified at all other times by the software (for example in the interrupt subroutine).	0 = Disables repeat function 1 = Enables repeat function
FRC Bit 1	**Force a DMA Cycle**—Forces a DMA burst to occur when the DMA cycle is software enabled or DMA Request Enabled. When FRC bit is set, it will remain set till a DMA burst for this channel starts as per channel priority, and will get cleared after the DMA burst for the channel starts. When set, software will read this bit as'1' till it gets cleared automatically or cleared by software.	0 = No effect 1 = Force DMA cycle
CEN Bit 0	**DMA Channel Enable**—Enables/Disables the DMA channel. **Note:** 1. To re-program a particular channel after completion of a DMA cycle refer Section 24.11 Application Note. 2. Disabling CEN during an ongoing burst on the AHB will stop the burst in between the transfer	0 = Disables the DMA channel 1 = Enables the DMA channel

18.3.3.5 Channel Request Source Select Registers

Each of the 32-bit channel request source select registers (RSSRx) selects one of the 32 DMA request signals (DMA_REQ [31:0]) to initiate a DMA transfer for the corresponding channel.

RSSR0	Channel 0 Request Source Select Register	0x10001090
RSSR1	Channel 1 Request Source Select Register	0x100010D0
RSSR2	Channel 2 Request Source Select Register	0x10001110
RSSR3	Channel 3 Request Source Select Register	0x10001150
RSSR4	Channel 4 Request Source Select Register	0x10001190
RSSR5	Channel 5 Request Source Select Register	0x100011D0
RSSR6	Channel 6 Request Source Select Register	0x10001210
RSSR7	Channel 7 Request Source Select Register	0x10001250
RSSR8	Channel 8 Request Source Select Register	0x10001290
RSSR9	Channel 9 Request Source Select Register	0x100012D0
RSSR10	Channel 10 Request Source Select Register	0x10001310
RSSR11	Channel 11 Request Source Select Register	0x10001350
RSSR12	Channel 12 Request Source Select Register	0x10001390
RSSR13	Channel 13 Request Source Select Register	0x100013D0
RSSR14	Channel 14 Request Source Select Register	0x10001410
RSSR15	Channel 15 Request Source Select Register	0x10001450

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												RSS				
TYPE	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-18. Channel Request Source Select Registers Description

Name	Description	Settings
Reserved Bits 31–5	Reserved—These bits are reserved and should read 0.	
RSS Bits 4–0	**Request Source Select**—Selects one of the 32 $\overline{\text{dma\_req}}$ signals that initiates a DMA transfer cycle for the channel.	00000 = select $\overline{\text{dma\_req}}$[0] 00001 = select $\overline{\text{dma\_req}}$ [1] ... 11111 = select $\overline{\text{dma\_req}}$ [31]

18.3.3.6 Channel Burst Length Registers

The Channel Burst Length registers (BLRx) control the burst length of a DMA cycle. For a FIFO channel setting, the burst length is normally assigned according to the FIFO size of the selected I/O device, or by the FIFO level at which its dma_req signal is asserted.

For example, when the UART RxD FIFO is 12 ¥ 8 and it asserts dma_req when it receives more than 8 bytes of data, BL is 8. When the memory port size also is 8-bit, the DMA burst is 8-byte reads followed by 8-byte writes.

When the memory port access size is smaller than the I/O port access size, the burst length of the byte writes is doubled. For example, the I/O port is 32-bit, the memory port is 16-bit, and the burst length is set to 32. In this configuration, the DMA performs 8 word burst reads and 16 halfword burst writes for I/O to memory transfer.

When the burst length is not programmed as a multiple of access sizes of source and destination, please see section 21.9 for the behavior of DMA.

		Addr
BLR0	Channel 0 Burst Length Register	0x10001094
BLR1	Channel 1 Burst Length Register	0x100010D4
BLR2	Channel 2 Burst Length Register	0x10001114
BLR3	Channel 3 Burst Length Register	0x10001154
BLR4	Channel 4 Burst Length Register	0x10001194
BLR5	Channel 5 Burst Length Register	0x100011D4
BLR6	Channel 6 Burst Length Register	0x10001214
BLR7	Channel 7 Burst Length Register	0x10001254
BLR8	Channel 8 Burst Length Register	0x10001294
BLR9	Channel 9 Burst Length Register	0x100012D4
BLR10	Channel 10 Burst Length Register	0x10001314
BLR11	Channel 11 Burst Length Register	0x10001354
BLR12	Channel 12 Burst Length Register	0x10001394
BLR13	Channel 13 Burst Length Register	0x100013D4
BLR14	Channel 14 Burst Length Register	0x10001414
BLR15	Channel 15 Burst Length Register	0x10001454

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											BL					
TYPE	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-19. Channel Burst Length Registers Description

Name	Description	Settings
Reserved Bits 31–6	Reserved—These bits are reserved and should read 0.	
BL Bits 5–0	**Burst Length**—Contains the number of data bytes that are transferred in a DMA burst.	000000 = 64 bytes read follow 64 bytes write 000001 = 1byte read follow 1 byte write 000010 = 2 bytes read follow 2 bytes write .... 111111 = 63 bytes read follow 63 bytes write

18.3.3.7 Channel Request Time-Out Registers

The Channel Request Time-Out registers (RTOx) set the time-out for DMA Request from the selected request source of the channel, which detects any discontinuity of data transfer. The request time-out takes effect only when the corresponding request enable (REN) bit in the channel control register (CCR) is set. An Internal Request Time-out Counter starts counting when a DMA channel is enabled and a burst on that channel ends. The Internal Request Time-out Counter is reset to zero when another burst for that channel starts. When the counter reaches the count value set in this register, it asserts an interrupt (if it not masked) and sets its error bit in the DMA request time-out status register (RTOSR). The input clock of the counter is selectable from either the system clock (HCLK) or input crystal (CLK32K).

The Internal Request Time-out Counter will not generate an error status (or count) for the first burst of a DMA cycle. It can be programmed to count (and can generate an error status as described above) for all other bursts in the DMA cycle.

NOTE

This register shares the same address as the bus utilization control register.

RTOR0	Channel 0 Request Time-Out Register	0x10001098
RTOR1	Channel 1 Request Time-Out Register	0x100010D8
RTOR2	Channel 2 Request Time-Out Register	0x10001118
RTOR3	Channel 3 Request Time-Out Register	0x10001158
RTOR4	Channel 4 Request Time-Out Register	0x10001198
RTOR5	Channel 5 Request Time-Out Register	0x100011D8
RTOR6	Channel 6 Request Time-Out Register	0x10001218
RTOR7	Channel 7 Request Time-Out Register	0x10001258
RTOR8	Channel 8 Request Time-Out Register	0x10001298
RTOR9	Channel 9 Request Time-Out Register	0x100012D8
RTOR10	Channel 10 Request Time-Out Register	0x10001318
RTOR 11	Channel 11 Request Time Out Register	0x10001358
RTOR 12	Channel 12 Request Time Out Register	0x10001398
RTOR 13	Channel 13 Request Time Out Register	0x100013D8
RTOR 14	Channel 14 Request Time Out Register	0x10001418
RTOR 15	Channel 15 Request Time Out Register	0x10001458

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EN	CLK	PSC	CNT												
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-20. Channel Request Time-Out Registers Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
EN Bit 15	**Enable**—Enables/Disables the DMA request time-out.	0 = Disables DMA request time-out 1 = Enables DMA request time-out
CLK Bit 14	**Clock Source**—Selects the counter of input clock source.	0 = HCLK 1 = 32.768 kHz
PSC Bit 13	**Prescaler Count**—Sets the prescaler of the input clock.	0 = Divide by 1 1 = Divide by 256
CNT Bits 12–0	**Request Time-Out Count**—Contains the time-out count down value for the internal counter in number of clocks. This value remains unchanged through out the DMA cycle.	

18.3.3.8 Channel Bus Utilization Control Registers

The Bus Utilization Control register (BUCRx) controls the bus utilization of an enabled channel when the request enable (REN) bit in channel control register (CCR) is cleared. The channel does not request a DMA transfer until the internal bus_untilization_counter reaches the count value set in this register except for the very first burst. This counter is cleared when the channel burst is started. When this count value is set to zero, the DMA carries on burst transfers one after another until it reaches the value set in Channel Count register. In this case, the user must be careful not to violate the maximum bus request latency of other devices.

NOTE

This register shares the same address of request time-out register.

BUCR0	Channel 0 Bus Utilization Control Register	0x10001098
BUCR1	Channel 1 Bus Utilization Control Register	0x100010D8
BUCR2	Channel 2 Bus Utilization Control Register	0x10001118
BUCR3	Channel 3 Bus Utilization Control Register	0x10001158
BUCR4	Channel 4 Bus Utilization Control Register	0x10001198
BUCR5	Channel 5 Bus Utilization Control Register	0x100011D8
BUCR6	Channel 6 Bus Utilization Control Register	0x10001218
BUCR7	Channel 7 Bus Utilization Control Register	0x10001258
BUCR8	Channel 8 Bus Utilization Control Register	0x10001298
BUCR9	Channel 9 Bus Utilization Control Register	0x100012D8
BUCR10	Channel 10 Bus Utilization Control Register	0x10001318
BUCR11	Channel 11 Bus Utilization Control Register	0x10001358
BUCR12	Channel 12 Bus Utilization Control Register	0x10001398
BUCR13	Channel 13 Bus Utilization Control Register	0x100013D8
BUCR14	Channel 14 Bus Utilization Control Register	0x10001418
BUCR15	Channel 15 Bus Utilization Control Register	0x10001458

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BU_CNT															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-21. Channel Bus Utilization Control Register Description

Name	Description
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.
BU_CNT Bits 15–0	**Bus Utilization Clock Count**—Sets the number of system clocks that must occur before the channel starts the next burst.

18.3.3.9 Channel Counter Registers

The Channel Counter Registers indicates the number of bytes transferred for the channel. It is reset to zero after channel is enabled and keeps incrementing for each transfer during the DMA burst. This counter will retain its value after the channel is disabled, till it is enabled again. If the RPT bit is found set at the end of the last burst of the DMA cycle, this counter retains its value and will be reset to zero only at the start of the another DMA burst—that is, the first burst of the new DMA cycle. If a DMA channel is disabled before the completion of the DMA cycle, this counter will retain the value of the number of data transferred in that DMA cycle. When the peripheral responds with a error response during a DMA data transfer, the CCNR value will not be increment for that AHB cycle as no data was transferred in that cycle.

		Addr
CCNR0	Channel 0 Channel Counter Register	0x1000109C
CCNR1	Channel 1 Channel Counter Register	0x100010DC
CCNR2	Channel 2 Channel Counter Register	0x1000111C
CCNR3	Channel 3 Channel Counter Register	0x1000115C
CCNR4	Channel 4 Channel Counter Register	0x1000119C
CCNR5	Channel 5 Channel Counter Register	0x100011DC
CCNR6	Channel 6 Channel Counter Register	0x1000121C
CCNR7	Channel 7 Channel Counter Register	0x1000125C
CCNR8	Channel 8 Channel Counter Register	0x1000129C
CCNR9	Channel 9 Channel Counter Register	0x100012DC
CCNR10	Channel 10 Channel Counter Register	0x1000131C
CCNR11	Channel 11 Channel Counter Register	0x1000135C
CCNR12	Channel 12 Channel Counter Register	0x1000139C
CCNR13	Channel 13 Channel Counter Register	0x100013DC
CCNR14	Channel 14 Channel Counter Register	0x1000141C
CCNR15	Channel 15 Channel Counter Register	0x1000145C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									CCNR[23:16]							
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CCNR[15:0]															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-22. Channel Counter Register Description

Name	Description
Reserved Bits 31–24	Reserved—These bits are reserved and should read 0.
CCNR Bits 23–0	**Channel Counter**—Indicates the number of bytes transferred for the channel.

18.4 DMA Chaining

This refers to the use of the same DMA Channel to automatically transfer a second data buffer (of maybe a different length) between another 2 sets of Source and Destination addresses, with an increase in the allowable value of the interrupt service time. This is possible because the ISR execution (that is, setup for next transfer) can go in parallel to the next buffer transfer from the DMA (when RPT bit is set).

To achieve DMA Chaining:

- Source Address Register for each Channel are double buffered internally.
- Destination Address Register for each Channel are double buffered internally.
- Channel Count Register for each Channel are double buffered internally.

With this the Host can update the values in these 3 register during an ongoing DMA Transfer for the same channel in preparation for the next DMA transfer. With the use of RPT and ACRPT bits the second transfer can occur for different source, destination addresses and different amount of data.

As an example, consider a Data Transfer of 14 K bytes from memory to a FIFO using 4K buffers.

- The driver writes 4K of data into buffer 1, sets the source register to buffer 1 and count to 4K, sets ACRPT, then enables the transfer. The DMA hardware will immediately latch the registers and start the transfer.
- The driver immediately writes 4K of data into buffer 2, sets the same source register to buffer 2, count to 4K, and sets the RPT bit.
- Transfer of buffer 1 completes, the DMA hardware samples the RPT bit, finds it set, latches the register (now set for buffer 2), clears the RPT bit because ACRPT is set, and starts the next transfer.
- It then generates the 1st interrupt.
- Driver ISR writes 4K of new data to buffer 1, sets the source register to buffer 1 and count to 4K, and sets the RPT bit again.
- Transfer of buffer 2 completes, the DMA hardware samples the RPT bit, finds it set, latches the registers (now set for buffer 1), clears the RPT bit because ACRPT is still set, and starts the next transfer.
- It then generates the 2nd interrupt.
- Driver ISR writes 2K of new data to buffer 2, sets the source register to buffer 2 and count to 2K, and sets the RPT bit again.
- Transfer of buffer 1 completes, the DMA hardware samples the RPT bit, finds it set, latches the register (now set for buffer 2), clears the RPT bit because ACRPT is still set, and starts the next transfer.
- It then generates the 3rd interrupt.
- Driver ISR has no more data to send so does nothing.
- Transfer of buffer 2 completes, the DMA hardware samples the RPT bit, finds it clear so it stops the transfer.
- It then generates the 4th interrupt.
- Driver ISR disables the DMA and the transfer is complete.

18.5 Special Cases of Burst Length and Access Size Settings

DMA burst length should normally be programmed as a multiple of the source and destination access sizes. The following sub-sections discuss the behavior that occurs when the burst length is not a multiple of access size.

18.5.1 Memory Increment

The following are the possible adverse effects:

1. Unknown data can be written at some locations, however, there is no data loss.
2. Number of bytes transferred can be more than the count value set.

These effects are explained in the examples below:

Example 1: Source is Linear memory with access size of 1 byte. Destination is linear memory with access size of 2 bytes. Burst Length is programmed as 3 bytes with memory increment. Source Addr Register: 0x0000_1000. Destination Addr Reg: 0x0000_2000

For the first burst DMA would read 3 bytes from addresses: 1000, 1001, and 1002. During the write cycle of the 1st burst DMA would write 2 bytes each at addresses 2000 and 2002. One extra memory location (0x2003) is written with unknown data from the DMA internal FIFO (8'h00 after hardware reset).

Example 2: Source is Linear Memory with access size of 2 bytes. Destination is linear memory with access size of 2 bytes. Burst Length is programmed as 3 bytes with memory increment. Source Addr Register: 0x0000_1000. Destination Addr Reg: 0x0000_2000

For the first burst, DMA would read 2 bytes each from addresses 1000 and 1002. During the write cycle of the 1st burst DMA would write 2 bytes each at addresses 2000 and 2002. One extra data byte is transferred per burst. When programmed with a count of say 9 bytes, DMA would perform data transfer of 12 bytes.

18.5.2 Memory Decrement

Possible Adverse Effects:

1. Unknown data can be written at some locations. In certain cases there can be data loss.

These effects are explained in the examples below:

For Example: Source is linear memory with access size of 1 byte. Destination is linear memory with access size of 2 bytes. Burst Length is programmed as 3 bytes with memory decrement. Source Addr Register: 0x0000_1000. Destination Addr Reg: 0x0000_2000

For the first burst DMA would read 3 bytes from addresses: 0FFD, 0FFE, and 0FFF. During write cycle of the 1st burst, DMA would write 2 bytes each at addresses 1FFC and 1FFE. An extra byte is written at the address 1FFF with unknown data from the DMA internal FIFO (8'h00 after hardware reset).

For the second burst DMA would read 3 bytes from the addresses: 0FFA, 0FFB, and 0FFC. During write cycle of the 2nd burst, DMA would write 2 bytes each at addresses 1FFA and 1FFC. In this case data

written at 1FFC and 1FFD in the first burst have been overwritten. Data written at 1FFD will be unknown data from the DMA internal FIFO (8'h00 after hardware reset).

Please note the following:

1. In the case of 2D Memory writing extra bytes would mean writing beyond the limits of X-Size programmed in a row.
2. Similarly for linear memory, this can lead data overflowing the allocated buffer for DMA.

18.6 Special Cases When CCNR and CNTR Values Differ

There are two combinations of events that can cause the values of the CCNR and the CNTR to differ. This situations are discussed in greater detail in the following sections.

18.6.1 CNTR Not A Multiple of Destination Access Size

If Counter (CNTR) register value is not a multiple of destination access size then the CCNR value will not match the value programmed in CNTR after completion of the DMA cycle for the channel. Table 18-23 illustrates the values of CCNR with different combinations of source and destination access sizes when CNTR = 5 bytes. This table holds good when BL = 3 bytes or BL = 4 bytes.

Table 18-23. CNNR Value Combinations

Source Size (Bytes)	Destination Size (Bytes)	No. of Bytes Read by DMA		No. of Bytes Written by DMA		CCNR (Bytes)
		Memory Increment	Memory Decrement	Memory Increment	Memory Decrement	
2	2	6	6	6	6	6
4	4	8	8	8	8	8
2	4	6	6	8	8	8
4	2	8	8	6	6	6
1	2	5	5	6	6	6
1	4	5	5	8	8	8

18.6.2 BL is Not a Multiple of Destination Access Size, CNTR Is

If BL register value is not a multiple of destination access size but CNTR is then the value of CCNR will not match the value programmed in CNTR after completion of the DMA cycle for the channel. Table 18-24 illustrates the values of CCNR with different combinations of source and destination access sizes when BL = 3 bytes and CNTR = 4 bytes.

Table 18-24. CCNR Value Combinations

Source Size (Bytes)	Destination Size (Bytes)	No. of Bytes Read by DMA		No. of Bytes Written by DMA		CCNR (Bytes)
		Memory Increment	Memory Decrement	Memory Increment	Memory Decrement	
2	2	6	6	6	6	6
4	4	8	8	8	8	8
2	4	6	6	8	8	8
4	2	8	8	6	6	6
1	2	4	4	6	6	6
1	4	4	4	8	8	8

NOTE

In case of memory decrement there might be some cases where DMA overwrites data written by itself so the number of bytes seen by the user can be different than those mentioned in the tables above.

18.7 Application Note

Following is the sequence to re-program a channel for data transfer:

1. Clear the status register bit corresponding to that channel (DISR,DBTOSR,DSESR,DRTOSR,DBOSR) after the DMA cycle is completed.
2. Change SMOD (source mode) to 2'b00 and clear CEN.
3. Re-program all registers corresponding to that particular channel, except CCR.
4. Program CCR and set CEN bit to 1.

NOTE

This sequence applies to all 16 channels in all modes—that is, Linear memory, 2D memory, and FIFO.

18.8 DMA Burst Termination

The DMAC needs to terminate its burst in case of:

- Transfer error response from slave
- Burst time-out error.
- Buffer overflow error.
- Channel disable (by software using CEN bit).

CAUTION

DMA burst termination may not occur immediately.

The burst termination occurs immediately in DMAC only on occurrence of Transfer Error response from Slave.

In other cases—that is, Burst time-out, Buffer overflow and Channel disable (by software using CEN bit) it takes about 2 more AHB transfers to terminate the burst after these are sensed. The burst termination is not done immediately to avoid AHB protocol violation.

In case the burst hangs and hready is not asserted for a large number of cycles then this must be handled by the watchdog timer in the ABCD Module in i.MX21 or by the system software.

18.9 Glossary of Terms Used

DMA burst—This refers to the burst cycles on the AHB Bus performed by the DMA.

DMA cycle—A DMA cycle can consists of a number of DMA bursts depending on the Channel Burst Length and Channel Count register settings. For example, For BL = 4 and CNTR = 8, the DMA cycle will consist of 2 DMA bursts.

Chapter 19
NAND Flash Memory Controller

This chapter describes the NAND Flash Controller on i.MX21. The NAND Flash Controller includes the following distinctive features.

- NAND Flash Interface: 8-bits / 16-bits (Pin Option)
- Internal RAM buffer (2 Kbyte) that is used as Boot RAM at booting and for buffering at normal operation
- Internal RAM buffer is memory mapped to the same AHB region as the registers
- Supports all 512B and 2 Kbyte page sized NAND Flash products regardless of density and/or organization

Along with these features is a host interface with the following functional features:

- Internal Bootcode loader during power-up (that can be enabled or disabled) Supports burst mode read and write using 16-bits or 32-bits bus transfers
- Error Correction Code (ECC) mode with detection and can be bypassed using single-bit auto-correction
- Multiple resets
  - Cold reset/ Warm reset / Hot reset (Reset of NAND Flash Controller and NAND Flash)
- Data Protection
  - Write Protection mode to the lower 1 Kbyte of RAM buffer space
  - Block based write protection of NAND Flash
  - Automatic Write protection of RAM buffer and NAND Flash during power-up

These features are combined with the ability to handshake with a NAND Flash through the use of interrupt pins that indicate if the NAND Flash is Ready or Busy.

Figure 19-2 shows a block diagram of the NAND Flash Controller.

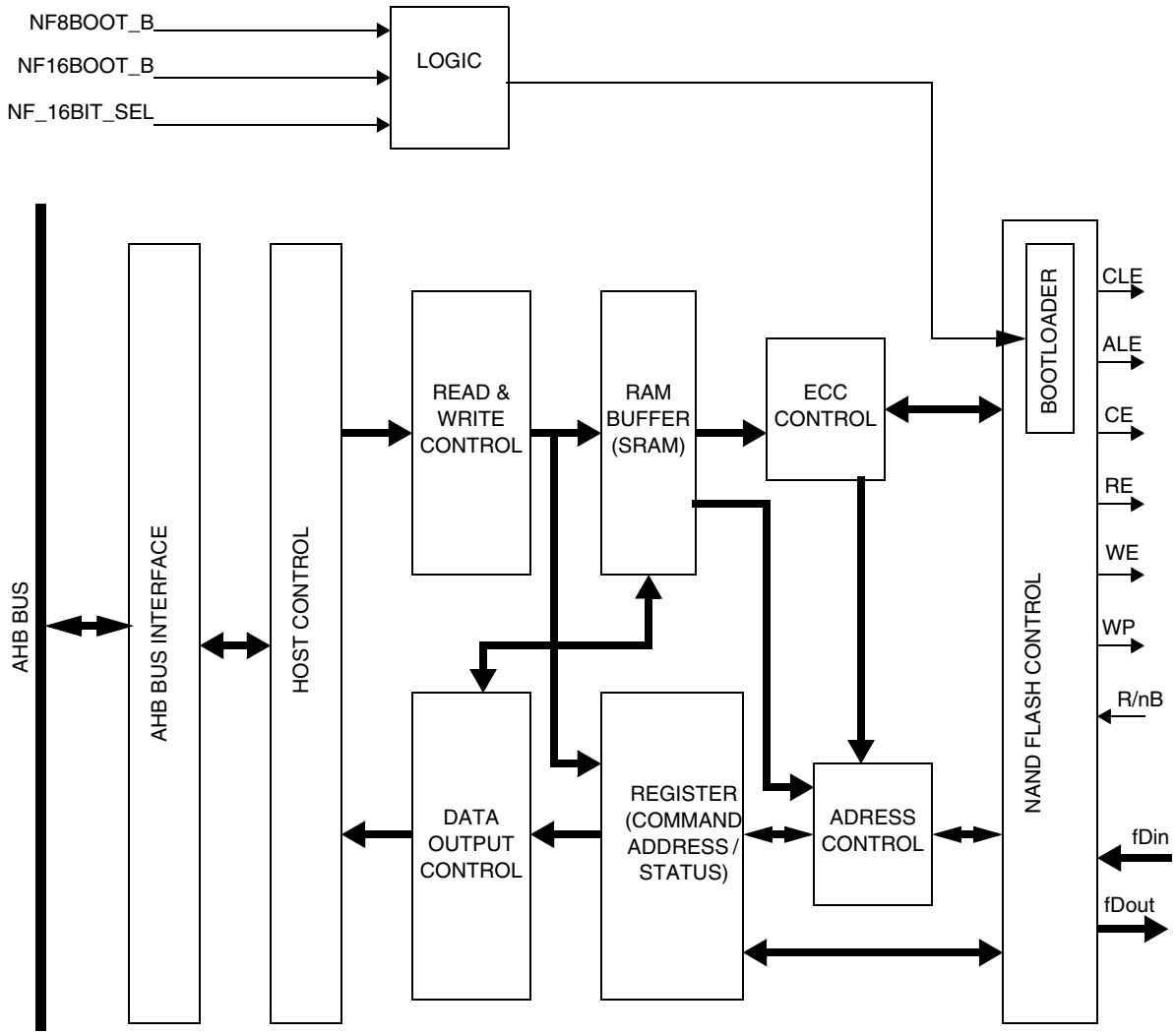


Figure 19-1. NAND Flash Controller Block Diagram

19.1 Functional Overview

The NAND Flash Controller interface with standard NAND Flash parts to the i.MX21 and hides the complexities of accessing the NAND Flash. It provides a glueless interface to both 8-bit and 16-bit NAND Flash parts with page sizes of 512 bytes or 2 Kbytes. The maximum density the NAND Flash Controller can support is limited by the Write Protection Unlock registers (Unlock_Start_Blk_Add and Unlock_End_Blk_Add). The limitation comes from the fact that these are 16-bit registers. Assuming the Unlock_Start_Blk_Add register starts at address 0x0000 and the Unlock_End_Blk_Add ends at address 0xFFFF, this yields a maximum of 64 k blocks (65,536 blocks). To calculate that maximum supported density, one also has to take into account the NAND Flash page size, either 512 bytes or 2 Kbytes. Thus the maximum densities that can be supported according to the respective page size are as follows:

512 byte page size with 32 pages/block: 512 bytes/page × 32 pages/block × 8bits/byte × 64 k-blocks = 8 Gbit

2 Kbyte page size with 64 pages/block: 2 Kbyte/page × 64 pages/block × 8bits/byte × 64 k-blocks = 64 Gbit.

The NAND Flash Controller is comprised of control logic and 2 Kbytes of internal RAM buffer. This 2 Kbytes buffer is used as BootRAM during cold reset (when boot from NAND Flash is chosen), and is used as regular RAM buffer after the boot procedure (to cache data to/from flash). The internal RAM buffers can be accessed as half-words (16-bits) or words (32-bits) and byte mode access is not supported.

The NAND Flash Controller provides a host transparent mechanism to input commands and addresses into the NAND Flash to read, write (program), erase or query status from the NAND Flash.

When the host needs to read from NAND Flash, it configures the controller with the appropriate read command and address and waits for an interrupt. The controller then transfers a page from the NAND Flash into the RAM buffer and generates an interrupt to signal that the read operation has completed. When this interrupt occurs the host can read the content from the internal RAM buffer.

When the host needs to program (write) the NAND Flash, it configures the controller and fills the RAM buffer with the content to be programmed, followed by a program command to the controller. The controller then performs the programming, and at the end it will assert an interrupt to the host. The host then checks the status of the operation by reading the status registers.

19.1.1 BOOTLOADER on Cold Reset Operation

Cold reset of the NFC and NAND Flash takes place during Power-On-Reset (POR).

The i.MX21 can be externally configured via its BOOT[3:0] pins to boot off the NAND Flash. The controller does this by copying 2 Kbytes of data from Block 0 of the NAND Flash to the internal RAM buffer (called BootRAM when used for booting).

The BOOTLOADER hardware module in the controller is responsible for automatically loading 2 Kbytes from the NAND Flash device into the internal 2 Kbyte RAM buffer. After exiting from the reset state, the ARM926EJ-S™ can read the first instruction from the BootRAM.

19.1.2 NAND Flash Control

The NAND Flash Control generates all the control signals that control the NAND Flash: nCE (Flash Chip Enable), nRE (Read Enable for read operations), nWE (Flash Write Enable), CLE (Flash Command Latch Enable, ALE (Flash Address Latch Enable). It monitors the R/nB (Flash Ready/Busy indication) signal to check if the NAND Flash is in the middle of an operation.

19.1.3 ECC Control

The ECC (Error Correction Code) block performs singe-bit error correction within each 512byte/2 Kbyte page and multi-bit error detection.

While the NFC accesses NAND Flash for a Program operation, it generates a code (24bits for the main area data and 10bits for the spare area data). For Read operations, it generates an ECC code and detects the number of errors and the error position, and corrects when there is a single-bit error. The ECC code is updated by the NFC automatically. After a Read operation, the host can know whether there is error or not by reading the status register (see section Table 19-3. "NFC Module Register Summary", on page 9). The

Error type is divided into three types: no error, a single bit error (correctable), and a 2 or more bits of error (uncorrectable).

The NFC generates the ECC for verification during a Read or Program operation. The generated ECC is never available in the internal RAM buffer. The ECC is directly programmed into the NAND Flash spare area during Program operation. The host can find out the generated ECC only by reading back the programmed page. Likewise, a Read operation will always read the NAND Flash spare area data and hence the generated ECC for verification is never available in the RAM buffer. Only the programmed ECC is available, always.

The ECC hardware always performs error detection. However error correction can be controlled (bypassed). This allows the host to examine the content in error, before software based correction or to employ proprietary ECC algorithms, if necessary.

19.1.4 Address Control

This module is responsible for address control and generation. It defines RAM buffer address generation (RAM buffer address for Data In/Data Out). It takes into account the Lock State Sequence (see section 19.5.4 "Write Protection", on page 31) and contains the Flash Memory Lock Address Comparator and RAM buffer Lock Address Comparator that are used to determine if the area is protected. It also generates the RAM buffer address for Bootload and the RAM buffer address for error correction.

19.1.5 RAM Buffer (SRAM)

The internal RAM Buffer is a 2,112 byte single port RAM buffer that is based on a synchronous high performance design. The full single port architecture share I/O and control for both NAND flash side and AHB side.

This memory has 528 words of 32 bits each, of which 512 words are used for the 4 main buffers, and the remaining 16 words form the spare area used for ECC and other data. This reflects the page organization of the NAND Flash device.

19.1.6 Register (Command/Address/Status)

This module contains 15 registers of 16-bits each. With these registers the host can control the NFC, read status on various operations, and perform direct command or address insertion to the NAND Flash.

19.1.7 Read and Write Control

The Read and Write Control block contains a connection to the internal bus which is connected to the RAM buffer and registers. On this internal bus, the block handles synchronous read and asynchronous write operations. It supports synchronous burst read lengths of 16-bit and 32-bit words.

19.1.8 Data Output Control

This module controls the data output (32-bits on the internal bus) which is driven to the AHB interface. It includes RAM buffer data output, register data output, and RAM buffer synchronization of the read mode pipe line.

19.1.9 Host Control

This module defines Host control which is connected to the AHB Interface through the internal bus.

19.1.10 AHB Interface

The AHB Interface is an adapter between AMBA AHB and the internal bus.

On the AHB side it supports 16-bit and 32-bit bus widths, burst and non-burst operations and on the internal bus side it supports 16-bit and 32-bit bus width with synchronous burst read and write.

This bus interface supports Little Endian byte ordering only.

When the transaction from the AHB is 16 bits, the interface translates the transaction to the internal bus. When the transaction is 32bits to/from registers, the interface breaks the transaction into two 16-bit transactions on the internal bus.

When the transaction from the AHB is a burst-read or write transaction, the interface creates a synchronous burst read cycle on the internal bus.

19.2 External

This section describes the input and output signals between the NAND Flash Controller.

The external interface is connected to the external memory device and is summarized in Table 19-2 and detailed in Section 19.2.1, “Flash Chip Enable (NFCE),” on page -5.

19.2.1 Flash Chip Enable ($\overline{\text{NFCE}}$)

This output signal is the NAND Flash selection control. When NAND Flash is in the Busy state, $\overline{\text{NFCE}}$ high is ignored, and the device does not return to standby mode.

19.2.2 Flash Read Enable ($\overline{\text{NFRE}}$)

This output signal is the NAND Flash serial data-out control, and when active drives the flash data onto the NAND Flash I/O bus. When writing a burst into the NAND Flash, $\overline{\text{NFRE}}$ increments the NAND Flash internal column address counter by one.

19.2.3 Flash Write Enable ($\overline{\text{NFWE}}$)

This output signal controls writes to the NAND Flash I/O port. Commands, address, and data are latched on the rising edge of the $\overline{\text{NFWE}}$ signal.

19.2.4 Flash Command Latch Enable (NFCLE)

The CLE output signal controls the activating path for commands sent to the command register of the NAND Flash. This signal is active high, and commands are latched into the command register of NAND Flash through the I/O ports on the rising edge of the $\overline{\text{NFWE}}$ signal.

19.2.5 Flash Address Latch Enable (NFALE)

The NFALE output signal controls the activating path for address to the internal address registers of the NAND Flash. Addresses are latched on the rising edge of $\overline{\text{NWFWE}}$ when NFALE is high.

19.2.6 Flash Write Protect ($\overline{\text{NFWP}}$)

The $\overline{\text{NFWP}}$ signal provides inadvertent program and erase protection during power transitions and is automatically controlled by the NFC. This signal status is activated to Low only during power-up.

19.2.7 NFRB—Flash Ready/Busy

NFRB input indicates the status of the NAND Flash operation. When low, it indicates that a Program, Erase, or random Read operation of the NAND Flash is in progress and returns to high state upon completion. It is an open drain output and a 100 K Ohm pull-up resistor is internally connected (Internal to the memory device). It does not float to a high-impedance condition when the device is deselected or when outputs are disabled.

19.2.8 WARM Reset Operation

A warm reset occurs when the RESET pin of i.MX21 is asserted or when an internal WATCHDOG timeout occurs. This makes the controller and NAND Flash stop the current operation and all internal registers go to the default state. The device is guaranteed to be reset in case RESET pulse is longer than 20 times the clock period of the NFC. Warm reset has no effect on the contents of main and spare area buffers.

CAUTION

When a warm reset is triggered by the RESET pin (RESET_IN) or an internal WATCHDOG timeout, the system bootloader does not reload 2 Kbytes of data from Block 0 of the NAND Flash to the internal RAM buffer (BootRAM). To reboot the system successfully, the bootcode in BootRAM loaded during a cold (POR) reset period must remain unchanged. Therefore, system designers are advised to minimize the bootcode to 512 bytes (1 Kbyte) and always reserve the first 512 bytes (1 Kbyte) in the RAM buffer for maintaining the bootcode. The remaining RAM Buffer can still be used for subsequent NAND Flash read or re-programming and does not affect normal NAND Flash operation. Note that the NAND Flash controller reads a maximum of 528 bytes per page read and it is recommended when reading from 2 Kbytes page size NAND Flash memories to transfer each 528 byte read from the NAND Flash controller RAM to another memory location such as SDRAM. Refer to Section 23.3.19, *NAND Flash Operation Configuration* (Configuration 2), bit FDO for more details on NAND Flash data output (reads).

19.2.9 Pin Configuration for the NAND Flash Controller (NFC)

Table 19-1shows the required GPIO configuration to enable the NFIO pins. The GPR (General Purpose Register) and GPIO In Use register (GIUSR) for Port F are configured to select 8-bit NFIO function after POR.

Table 19-1. GPIO Configuration for NAND Flash Controller

NFC Pin	Pin Description	GPIO Port	Configuration to Enable NFIO
NFIO15–11	Alternate function of Address bus A25–A21	–	Set GPR bits 31–27 if 16-bit NAND Flash device is used
NFIO10–8	Alternate function of Address bus A15–A13	–	Set GPR bits 25–23 if 16-bit NAND Flash device is used
NFIO7–0	Primary function	PF14–PF7	Clear GPR bits 14–7
NFWE_B	Primary function	PF6	Clear GPR bit 6
NFRE_B	Primary function	PF5	Clear GPR bit 5
NFALE	Primary function	PF4	Clear GPR bit 4
NFCLE	Primary function	PF3	Clear GPR bit 3
NFWP_B	Primary function	PF2	Clear GPR bit 2
NFCE_B	Primary function	PF1	Clear GPR bit 1
NFRB	Primary function	PF0	Clear GPR bit 0

19.3 Programming Model

There are 10 NFC Control (Write) Registers and 5 Status (Read) registers. This section describes the memory map of the NFC. In general it is divided into four parts:

1. Main area RAM buffer memory which is used for data loading from the NAND Flash or programming the NAND Flash.
2. Spare area RAM buffer memory that is used for ECC (Error Correction) Code and can also be used for other functions like Logical Sector Number, Bad Block Information, and Wrap Count.
3. Registers which control the NFC and provide a current status of the last operation.
4. Reserved area.

Note that all registers and internal memory are memory mapped to the same AHB region.

19.3.1 Memory Mapping

The NFC internal memory arrays are mapped to the address `0xDF00_3000-0xDF00_3FFF`. Both Registers and Internal memory are mapped to this region. The memory mapping arrays are mapped according to Table 19-2.

Table 19-2. NFC Array—Internal Register Summary

Address	Use	Access
0xDF00_3000h – 0xDF00_31FEh	Main area Buffer 0	R/W
0xDF00_3200h – 0xDF00_33FEh	Main area Buffer 1	R/W
0xDF00_3400h – 0xDF00_35FEh	Main area Buffer 2	R/W

Table 19-2. NFC Array—Internal Register Summary (continued)

Address	Use	Access
0xDF00_3600h – 0xDF00_37FEh	Main area Buffer 3	R/W
0xDF00_3800h – 0xDF00_380Eh	Spare area Buffer 0	R/W
0xDF00_3810h – 0xDF00_381Eh	Spare area Buffer 1	R/W
0xDF00_3820h – 0xDF00_382Eh	Spare area Buffer 2	R/W
0xDF00_3830h – 0xDF00_383Eh	Spare area Buffer 3	R/W
0xDF00_3840h – 0xDF00_3BFEh	Reserved	–
0xDF00_3E00 – 0xDF00_3E1C	Registers	R/W

19.3.2 Spare Area Buffer

The main area buffer is a data block whereas the spare area buffer is used for several functions including Error Correction. There is a difference in the memory organization of this memory region depending on whether the NFC is interfaced to an 8-bit or 16-bit NAND Flash bus width. In Figure 19-2, 8-bit organization is shown and in Figure 19-3, 16 bit configuration is shown.

Address	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
800h (SB0)	LSN(2nd)								LSN(1st)							
802h (SB0)	WC(1st)								LSN(3rd)							
804h (SB0)	BI								WC(2nd)							
806h (SB0)	ECC Code for Main area data (2nd)								ECC Code for Main area data (1st)							
808h (SB0)	ECC Code for Spare area data (1st)								ECC Code for Main area data (3rd)							
80Ah (SB0)	Reserved								ECC Code for Spare area data (2nd)							
80Ch (SB0)	Reserved								Reserved							
80Eh (SB0)	Reserved								Reserved							
810h–81Eh (SB1)	SB1 – SB3 have same assignment like SB0.															
820h–82Eh (SB2)																
830h–83Eh (SB3)																

Figure 19-2. Spare Area Buffer (NAND Flash with 8-Bit I/O Bus)

Address	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
800h (SB0)	LSN(2nd)								LSN(1st)							
802h (SB0)	WC(1st)								LSN(3rd)							
804h (SB0)	Reserved								WC(2nd)							
806h (SB0)	ECC Code for Main area data (2nd)								ECC Code for Main area data (1st)							
808h (SB0)	ECC Code for Spare area data (1st)								ECC Code for Main area data (3rd)							
80Ah (SB0)	BI								ECC Code for Spare area data (2nd)							
80Ch (SB0)	Reserved								Reserved							
80Eh (SB0)	Reserved								Reserved							
810h–81Eh (SB1)	SB1 – SB3 have same assignment like SB0.															
820h–82Eh (SB2)																
830h–83Eh (SB3)																

Figure 19-3. Spare Area Buffer (NAND Flash with 16-Bit I/O Bus)

Note: LSN—Logical Sector Number
Note: WC—Wrap Count and other bytes have same wrap count information and used as error correction for wrap count itself.
Note: BI—Bad block Information

The host can use all of the spare area except BI and ECC code areas. This includes reserved (unused) locations in the spare area.

The NFC automatically generates ECC code for both main and spare areas during data transfer to the NAND Flash. Though, the ECC code is written to the spare area of the NAND Flash, it is not updated into the spare area in RAM.

When programming or reading the spare area, the spare area buffer number (SB 0–3) is chosen through the Start buffer register.

The NFC module includes 15 registers of 16 bits each. Table 19-3 summarizes these registers and their addresses. Registers are accessible in supervisor mode only and result in an exception otherwise.

Table 19-3. NFC Module Register Summary

Description	Name	Address
Internal SRAM Size	NFC_BUFSIZE	0xDF00_3E00
NAND Flash Block Address for Lock Check	Block_Add_Lock	0xDF00_3E02
Buffer Number for Page Data Transfer To/From Flash Memory	RAM_Buffer_Address	0xDF00_3E04
NAND Flash Address	NAND_Flash_Add	0xDF00_3E06
NAND Flash Command	NAND_Flash_CMD	0xDF00_3E08
NFC Internal Buffer Lock Control	NFC_Configuration	0xDF00_3E0A
Controller Status/Result of Flash Operation	ECC_Status_Result	0xDF00_3E0C
ECC Error Position of Main Area Data Error	ECC_Rslt_Main_area	0xDF00_3E0E
ECC Error Position of Spare Area Data Error	ECC_Rslt_Spare_area	0xDF00_3E10

Table 19-3. NFC Module Register Summary (continued)

Description	Name	Address
Nand Flash Write Protection	NF_WR_Prot	0xDF00_3E12
Start Address for Write Protection Unlock	Unlock_Start_Blk_Add	0xDF00_3E14
End Address for Write Protection Unlock	Unlock_End_Blk_Add	0xDF00_3E16
NAND Flash Write Protection Status	NAND_Flash_WR_Pr_St	0xDF00_3E18
NAND Flash Operation Configuration (Configuration 1)	NAND_Flash_Config1	0xDF00_3E1A
NAND Flash Operation Configuration (Configuration 2)	NAND_Flash_Config2	0xDF00_3E1C

19.3.3 Internal SRAM Size

This is a read only register.

NFC_BUFSIZE — NFC_BUFSIZE — Addr 0xDF00_3E00

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													BUFSIZE			
TYPE													r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	0x0001															

Table 19-4. Internal SRAM Size Register Description

Name	Description	Settings
Reserved Bits 15–4	Reserved	
BUFSIZE Bits 3–0	**Buffer Size**—The size of internal RAM buffer.	0000= 1 kbyte 0001= 2 kbytes (Default) 0010= 3 kbytes 0011= 4 kbytes 0100–1111 = Reserved

19.3.4 NAND Flash Block Address for Lock Check

Block_Add_Lock — NAND Flash Block Address for Lock Check — Addr 0xDF00_3E02

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 19-5. NAND Flash Block Address for Lock Check Register Description

Name	Description
Reserved Bits 15–0	Reserved—These bits are reserved and should read 0.

19.3.5 Buffer Number for Page Data Transfer To/From Flash Memory

RAM_Buffer_Address — Buffer Number for Page Data Transfer To/From Flash Memory — Addr 0xDF00_3E04

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															RBA	
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 19-6. Buffer Number for Page Data Transfer To/From Flash Memory Register Description

Name	Description	Settings
Reserved Bits 15–2	Reserved	
RBA Bits 1–0	**RAM Buffer Address**—Specifies the RAM buffer number to use for data transfer to or from the NAND Flash.	00= 1st internal RAM buffer 01= 2nd internal RAM buffer 10= 3rd internal RAM buffer 11= 4th internal RAM buffer

19.3.6 NAND Flash Address

NAND_Flash_Add — NAND Flash Address — Addr 0xDF00_3E06

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADD															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 19-7. NAND Flash Address Register Description

Name	Description
ADD Bits 15–0	**NAND Flash Address**—NAND Flash address which will be read, programmed or erased. This address is entered into NAND Flash.

19.3.7 NAND Flash Command

NAND_Flash_CMD — NAND Flash Command — Addr 0xDF00_3E08

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMD															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 19-8. NAND Flash Command Register Description

Name	Description
CMD Bits 15–0	**NAND Flash Command**—This CMD is entered into NAND Flash.

19.3.8 NFC Internal Buffer Lock Control

NFC_Configuration — NFC Internal Buffer Lock Control — Addr 0xDF00_3E0A

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															BLS	
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	0x0001															

Table 19-9. NFC Internal Buffer Lock Control Register Description

Name	Description	Settings
Reserved Bits 15–2	Reserved—These bits are reserved and should read 0.	
BLS Bits 1–0	**Buffer Lock Set**—This field specifies the buffer lock status of first 2 page buffer.	00 = Locked 01 = Locked (default) 10 = Unlocked 11 = Locked

19.3.9 Controller Status/Result of Flash Operation

ECC_Status_Result — Controller Status/Result of Flash Operation — Addr 0xDF00_3E0C

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													ERm		ERs	
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 19-10. Controller Status/Result of Flash Operation Register Description

Name	Description	Settings
Reserved Bits 15–4	Reserved—These bits are reserved and should read 0.	
ERm Bits 3–2	**ECC Error for Main Area Data**—Shows the number of errors in a page as a result of ECC check at the page read operation. ECC algorithm of NFC cannot correct more than 1 error bit per page and 2 or more error bits are considered uncorrectable.	00 = No Error 01 = 1-bit Error (Correctable Error) 10 = 2-bits Error (Uncorrectable Error) 11 = Reserved
ERs Bits 1–0	**ECC Error for Spare Area Data**—Shows the number of errors in a page as a result of ECC check at the page read operation. ECC algorithm of NFC cannot correct more than 1 error bit per page and 2 or more error bits are considered uncorrectable.	00 = No Error 01 = 1-bit Error (Correctable Error) 10 = 2-bits Error (Uncorrectable Error) 11 = Reserved

19.3.10 ECC Error Position of Main Area Data Error (8-bit NAND Flash)

ECC_Rslt_Main_area — ECC Error Position of Main Area Data Error — Addr 0xDF00_3E0E

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					ECC Result 1									ECC Result 2		
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 19-11. ECC Error Position of Main Area Data Error (8-bit NAND Flash) Register Description

Name	Description
Reserved Bits 15–12	Reserved—These bits are reserved and should read 0.
ECC Result 1 Bits 11–3	**ECC Error Address of Main area Register** ECC Result 1—Byte position of ECC error within the page (512 bytes).
ECC Result 2 Bits 2–0	**ECC Error Address of Main area Register** ECC Result 2—Bit position of ECC error within the byte for 8-bit NAND Flash.

19.3.11 ECC Error Position of Main Area Data Error (16-Bit NAND Flash)

ECC_Rslt_Main_area — ECC Error Position of Main Area Data Error — Addr 0xDF00_3E0E

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					ECC Result 1								ECC Result 2			
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 19-12. ECC Error Position of Main Area Data Error (16-Bit NAND Flash) Register Description

Name	Description
Reserved Bits 15–12	Reserved
ECC Result 1 Bits 11–4	**ECC Error Address of Main area Register** ECC Result 1—half-word position of ECC error within the page (256 half-words).
ECC Result 2 Bits 3–0	**ECC Error Address of Main area Register** ECC Result 2—Bit position of ECC error within the half-word.

19.3.12 ECC Error Position of Spare Area Data Error (8-bit NAND Flash)

ECC_Rslt_Spare_area — ECC Error Position of Spare Area Data Error — Addr 0xDF00_3E10

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												ECC Result 4		Result 3		
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 19-13. ECC Error Position of Spare Area Data Error (8-Bit NAND Flash) Register Description

Name	Description	Settings
Reserved Bits 15–5	Reserved—These bits are reserved and should read 0.	
ECC Result 4 Bits 4–3	**ECC Error Address of Spare area Register** ECC Result 4—Byte position of ECC error in the Logical Sector Number.	00 = 1st byte 01 = 2nd byte 10 = 3rd byte
ECC Result 3 Bits 2–0	**ECC Error Address of Spare area Register** ECC Result 3—Bit position of ECC error within the byte.	–

19.3.13 ECC Error Position of Spare Area Data Error (16-bit NAND Flash)

ECC_Rslt_Spare_area — ECC Error Position of Spare Area Data Error — Addr 0xDF00_3E10

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												Result 4	Result 3			
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 19-14. ECC Error Position of Spare Area Data Error (16bit NAND Flash) Register Description

Name	Description
Reserved Bits 15–5	Reserved—These bits are reserved and should read 0.
ECC Result 4 Bit 4	**ECC Error Address of Spare Area Register** ECC Result 4—Half-word position of ECC error in the Logical Sector Number.
ECC Result 3 Bits 3–0	**ECC Error Address of Spare Area Register** ECC Result 3—Bit position of ECC error within the half-word.

19.3.14 Nand Flash Write Protection

NF_WR_Prot — Nand Flash Write Protection — Addr 0xDF00_3E12

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														WPC		
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	0x0002															

Table 19-15. Nand Flash Write Protection Register Description

Name	Description	Settings
Reserved Bits 15–3	Reserved—These bits are reserved and should read 0.	
WPC Bits 2–0	**Write Protection Command**—The Command field specifies the operation which the controller will perform.	0004h (0000 0100) = Unlock NAND Flash block(s) according to given block address range. 0002h (0000 0010) = Lock all NAND Flash block(s). 0001h (0000 0001) = Lock-tight locked block(s).

For more information on write protection of the NAND Flash, refer to Section 19.5.4, “Write Protection,” on page -31.

19.3.15 Start Address for Write Protection Unlock

Unlock_Start_Blk_Add	Start Address for Write Protection Unlock	Addr 0xDF00_3E14

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	USBA															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 19-16. Start Address for Write Protection Unlock Register Description

Name	Description
USBA Bits 15–0	**Unlock Start Block Address**—Starting address of blocks that are in Write Protection mode and to be unlocked. This address is used in the Unlock Block command.

19.3.16 End Address for Write Protection Unlock

Unlock_End_Blk_Add	End Address for Write Protection Unlock	Addr 0xDF00_3E16

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UEBA															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 19-17. End Address for Write Protection Unlock Register Description

Name	Description
UEBA Bits 15–0	**Unlock End Block Address**—Ending address of blocks that are in Write Protection mode and to be unlocked.This address is used in the Unlock Block command.

19.3.17 NAND Flash Write Protection Status

NAND_Flash_WR_Pr_St	NAND Flash Write Protection Status	Addr 0xDF00_3E18

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														US	LS	LTS
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	0x0002															

Table 19-18. NAND Flash Write Protection Status Register Description

Name	Description	Settings
Reserved Bits 15–3	Reserved—These bits are reserved and should read 0.	
US Bit 2	**Unlocked Status**—Specifies whether there are any unlocked blocks in the NAND Flash.	0 = No unlocked block in the NAND Flash 1 = There are unlocked blocks the in NAND Flash
LS Bit 1	**Locked Status**—Specifies that all NAND Flash blocks are in locked status.	0 = Not all NAND Flash blocks are in locked status 1 = All NAND Flash blocks are in locked status
LTS Bit 0	**Lock-Tighten Status**—Specifies that Locked block(s) is (are) lock-tightened.	0 = Locked block(s) is (are) not lock-tightened 1 = Locked block(s) is (are) lock-tightened

19.3.18 NAND Flash Operation Configuration (Configuration 1)

NAND Flash Operation Configuration (Configuration 1) Addr

NAND_Flash_Config1 0xDF00_3E1A

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												INT_MASK	ECC_EN	SP_EN		
TYPE	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

0x0008

Table 19-19. NAND Flash Operation Configuration (Configuration 1) Register Description

Name	Description	Settings
Reserved Bits 15–5	Reserved—These bits are reserved and should read 0.	
INT_MASK Bit 4	**Interrupt Mask**—This field determines whether NFC generate interrupt or mask interrupt.	0 = Unmask interrupt 1= Mask interrupt
ECC_EN Bit 3	**ECC Operation Enable**—This field determines whether ECC auto-correction is executed or bypassed	0 = ECC auto-correction is bypassed 1 = ECC auto-correction is executed
SP_EN Bit 2	**Spare Area Enable**—This field determines whether NFC reads/writes only NAND Flash spare area data or both NAND Flash main and spare area data	0 = NAND Flash main and spare area data is enabled. 1 = NAND Flash spare only data is enabled.
Reserved Bits 1–0	Reserved—These bits are reserved and should read 0.	

19.3.19 NAND Flash Operation Configuration (Configuration 2)

Note that INT (Bit 15) reset value is zero, but soon after power-up it will change to a value of one. When performing boot from NAND Flash, the INT bit will change from zero to one after the transfer of bootcode has been accomplished. For more information take a look on Section 19.1.1, “BOOTLOADER on Cold Reset Operation,” on page -3.

NAND_Flash_Config2 — NAND Flash Operation Configuration (Configuration 2) — Addr 0xDF00_3E1C

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT										FDO			FDI	FADD	FCMD
TYPE	rw	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 19-20. NAND Flash Operation Configuration (Configuration 2) Register Description

Name	Description	Settings
INT Bit 15	**Interrupt**—This bit is set to 1 by the NFC when basic operation and bootloading is done, or when warm or hot reset is released. An interrupt event occurs when it is set to 1 and interrupts from NFC are unmasked. This bit is cleared by writing a 0. The host should clear this bit before initiating a new operation. *Basic operation:* NAND Flash command input NAND Flash address input NAND Flash Data input NAND Flash Data output	0 = Basic operation or bootloading is still running 1 = Basic operation or bootloading is done.
Reserved Bits 14–6	Reserved—These bits are reserved and should be set to 0.	
FDO Bits 5–3	**NAND Flash Data Output**—This field activates NAND Flash Data Output to NFC. **Note:** Output is with respect to the NAND Flash device.	Use only 1 of the following options: FDO[2:0] = 001 = One page data out [1]. 010 = NAND Flash ID data out 100 = NAND Flash status register dataout
FDI Bit 2	**NAND Flash Data Input**—This field activates NAND Flash Data Input from NFC. **Note:** Input is with respect to NAND Flash device.	0 = To activate another basic operation or change interrupt bit. 1 = To activate NAND Flash Data Input operation.
FADD Bit 1	**NAND Flash Address Input**—This field specifies the NAND Flash Address Input.	0 = To activate another basic operation or change interrupt bit. 1 = To activate NAND Flash Address Input operation.
FCMD Bit 0	**NAND Flash Command Input**—This field specifies the NAND Flash Command Input.	0 = To activate another basic operation or change interrupt bit. 1 = To activate NAND Flash Command Input operation.

[1] One page size is determined by SP_EN register bit and NFC_FMS input. It is 528bytes or 16bytes no matter if NFC_FMS is 0 or 1.

NOTE

1) When the basic operation is completed, the FCMD/FADD/FDI/FDO bits will be reset to 0 automatically.

2) Only one operation among FCMD/FADD/FDI/FDO can be activated. Activate only one option (INT can be activated with one of FCMD/FADD/FDI/FDO operations).

19.4 Operating Modes

The NAND Flash Controller operating modes are described in this section. Table 19-21 shows the possible operating modes of the NAND Flash Controller. The BOOT[3:0] are external pins that select system boot options. NFC_FMS and NF_16BIT_SEL are bits that are automatically set or cleared by the hardware when NAND Flash boot options are chosen. When other boot options are chosen, these bits are set by software to indicate to the NAND Flash Controller on the configuration and organization of the NAND Flash attached to the NAND Flash Controller. NFC_FMS and NF_16BIT_SEL are bits in the Function Multiplexing Control Register (FMCR). This register is described in the System Control, chapter.

Table 19-21. NAND Flash Operating Modes

BOOT[3:0]	NFC_FMS	NF_16BIT_SEL	Description
0011	1	1	Boot from a 16-bit NAND Flash (2 kbytes per page). NFC_FMS and NF_16BIT_SEL are configured internally by hardware.
0100	0	1	Boot from a 16-bit NAND Flash (512 bytes per page). NFC_FMS and NF_16BIT_SEL are configured internally by hardware.
0010	1	0	Boot from an 8-bit NAND Flash (2 kbytes per page). NFC_FMS and 16BIT_SEL are configured internally by hardware.
0111	0	0	Boot from an 8-bit NAND Flash (512 bytes per page). NFC_FMS and NF_16BIT_SEL are configured internally by hardware.
X	0	0	No boot from NAND Flash. Host configures NAND Flash as 8-bit with 512 bytes per page.
X	0	1	No boot from NAND Flash. Host configures NAND Flash as 16-bit with 512 bytes per page.
X	1	0	No boot from NAND Flash. Host configures NAND Flash as 8-bit with 2 kbytes per page.
X	1	1	No boot from NAND Flash. Host configures NAND Flash as 16-bit with 2 kbytes per page.

19.5 General Operation

This section describes how to operate the NFC using the NFC control registers and interrupts. The NFC operations are divided into the following operations.

- Basic Operations
- Normal operations
- ECC operations to the internal memory and the flash device
- Write protection operation to the internal memory and the flash device

Normal Operations are used to operate the NAND Flash. These operations are composed of a sequence of Basic Operations. Basic and Normal Operations are described in the following sections along with flow charts.

This section also provides memory configuration examples.

19.5.1 Basic Operation

Normal operation (for example Read NAND Flash data, etc.) are composed of Basic Operations. A Basic Operation is used to build higher levels of operations. The basic operations are as follows.

- Preset Operation
- NAND Flash Command Input Operation
- NAND Flash Address Input Operation
- NAND Flash Data Input Operation
- NAND Flash Data Output Operation

Each of these operations are detailed in the following sections.

19.5.1.1 Preset Operation

The sequence of events of the NAND Flash controller during Preset operation is shown in Figure 19-4.

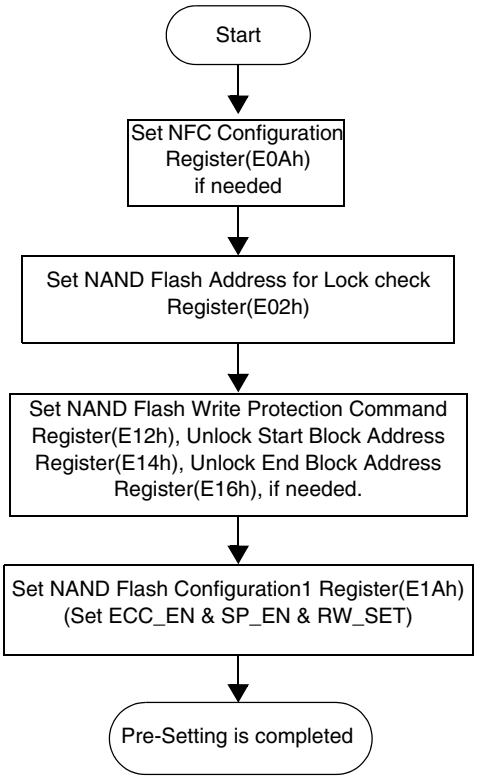


Figure 19-4. Preset Operation

19.5.1.2 NAND Flash Command Input Operation

The sequence of events of the NAND Flash controller during Input operation is shown in Figure 19-5.

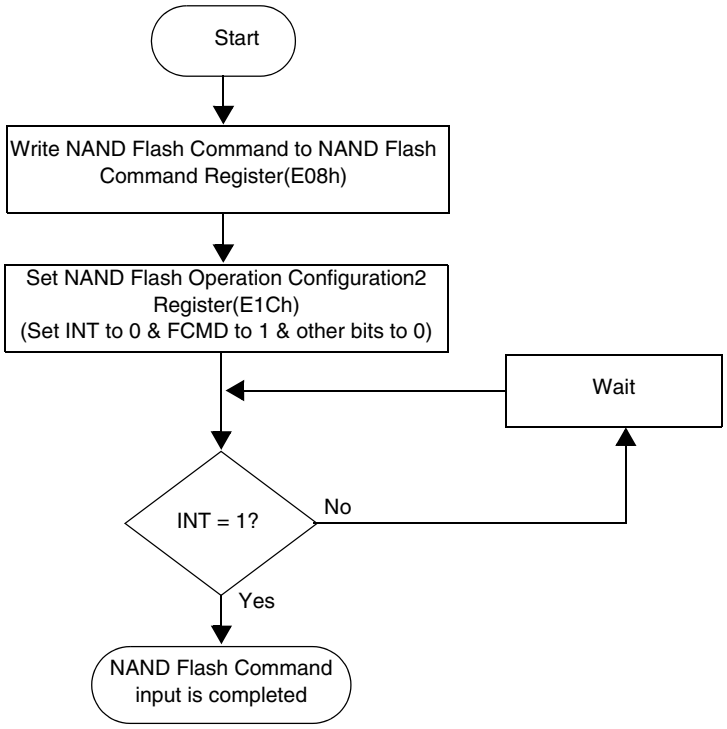


Figure 19-5. NAND Flash Command Input Operation

19.5.1.3 NAND Flash Address Input Operation

The sequence of events of the NAND Flash controller during Flash Address Input operation is shown in Figure 19-6.

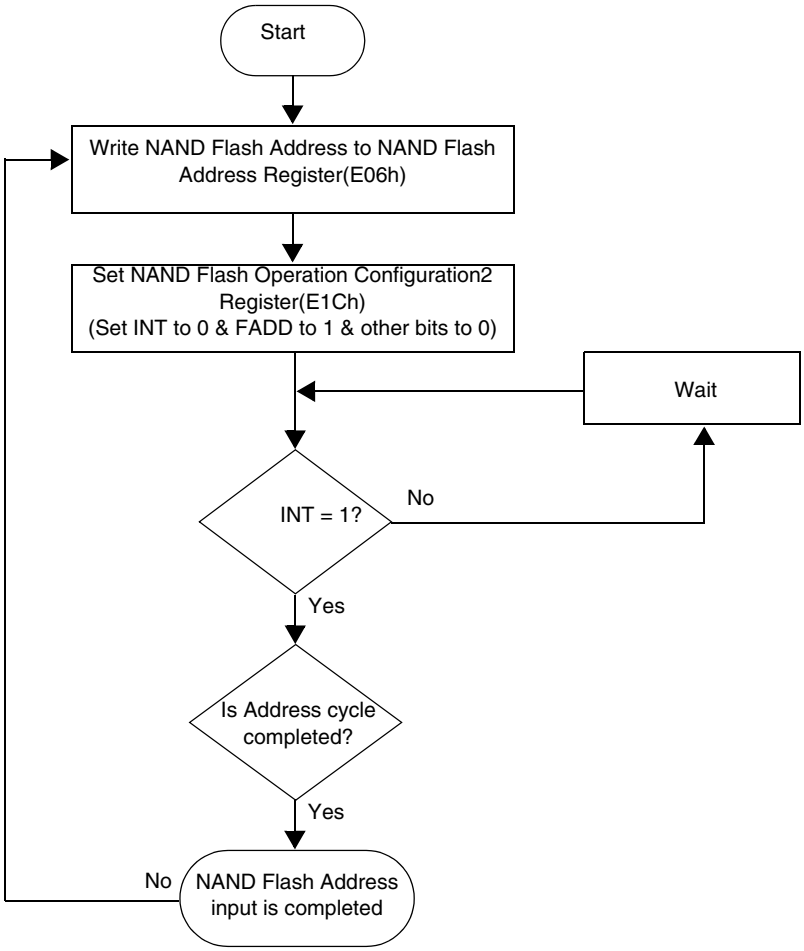


Figure 19-6. NAND Flash Address Input Operation

19.5.1.4 NAND Flash Data Input Operation

The sequence of events of the NAND Flash controller during Flash Data Input operation is shown in Figure 19-7.

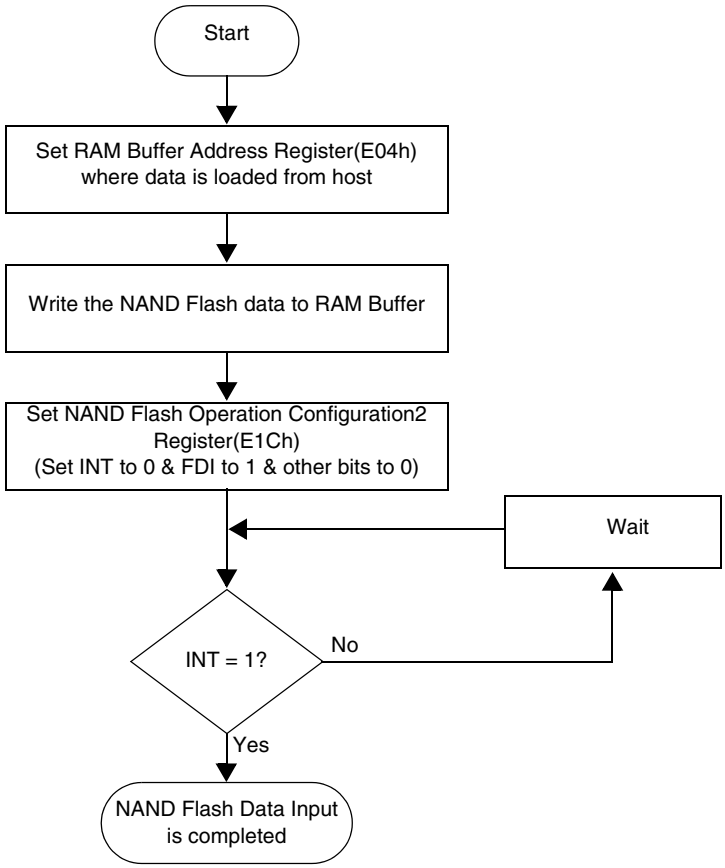


Figure 19-7. NAND Flash Data Input Operation

19.5.1.5 NAND Flash Data Output Operation

The sequence of events of the NAND Flash controller during Flash Data Output operation is shown in Figure 19-8.

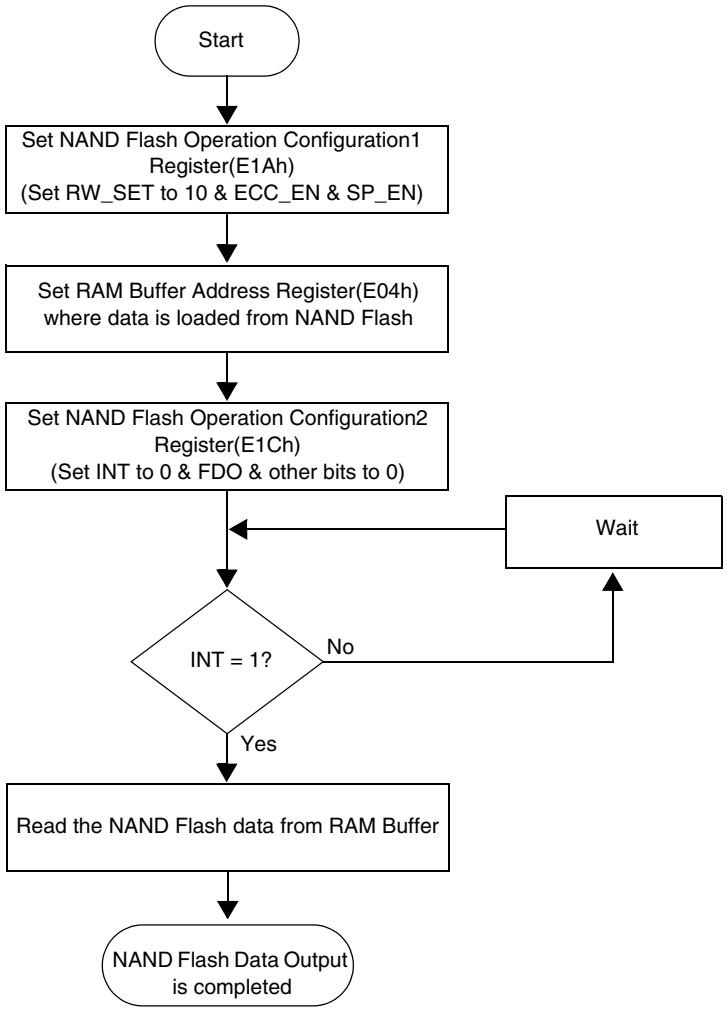


Figure 19-8. NAND Flash Data Output Operation

19.5.2 Normal Operation

Normal Operations are composed of Basic Operations. The following is a list of Normal Operations:

- NAND Flash Read ID Operation
- NAND Flash Read Status Operation
- NAND Flash Read Data Operation
- Program NAND Flash Data Operation
- Erase NAND Flash Data Operation
- Hot Reset

Each of these operations are detailed in the following sections.

19.5.2.1 NAND Flash Read ID Operation

The sequence of events of the NAND Flash controller during Flash Read ID operation is shown in Figure 19-9.

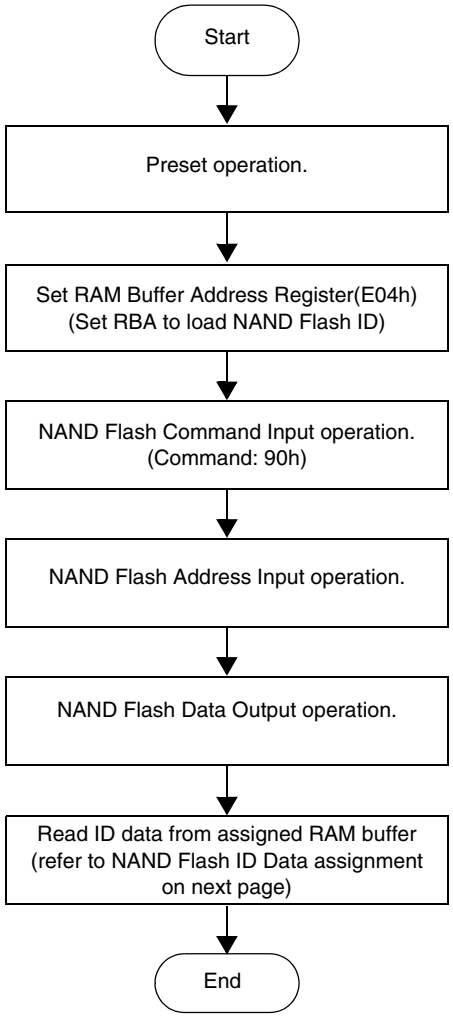


Figure 19-9. Read NAND Flash ID Operation

The assignment of NAND Flash ID data is stored in RAM buffer (8-bit NAND Flash) and is formatted as shown in Figure 19-10.

	1st half-word		2nd half-word		3rd half-word	
RAM Buffer of RBA address	**1st byte of ID**	**2nd byte of ID**	**3rd byte of ID**	**4th byte of ID**	**5th byte of ID**	**6th byte of ID**
	LSB	MSB				

Figure 19-10. NAND (8-Bit) Flash ID Data Format in RAM Buffer

The assignment of NAND Flash ID data stored in RAM buffer (16-bit NAND Flash) is formatted as shown in Figure 19-11.

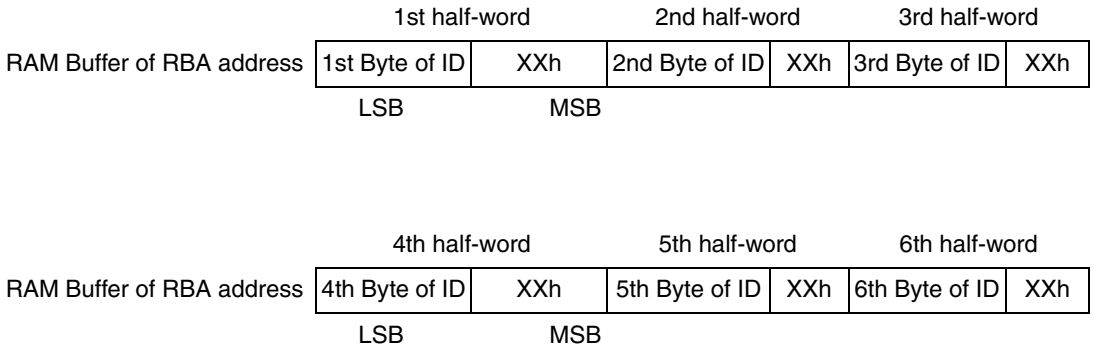


Figure 19-11. NAND (16-Bit) Flash ID Data Format in RAM Buffer

19.5.2.2 NAND Flash Read Status Operation

The sequence of events of the NAND Flash controller during Flash Read Status operation is shown in Figure 19-12.

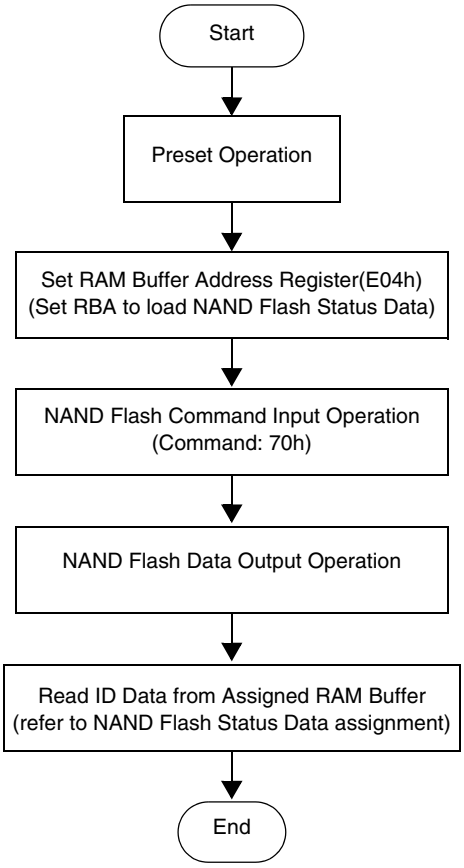


Figure 19-12. Read NAND Flash Status Operation

The assignment of NAND Flash Status data is stored in RAM buffer (8-bit/16-bit NAND Flash) formatted as shown in Figure 19-13.

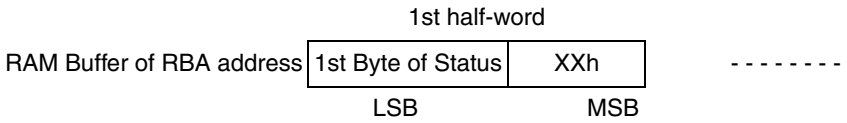


Figure 19-13. NAND (8-Bit/16-Bit) NAND Flash Status Format in RAM Buffer

19.5.2.3 NAND Flash Read Data Operation

The sequence of events of the NAND Flash controller during Flash Read Data operation is shown in Figure 19-14.

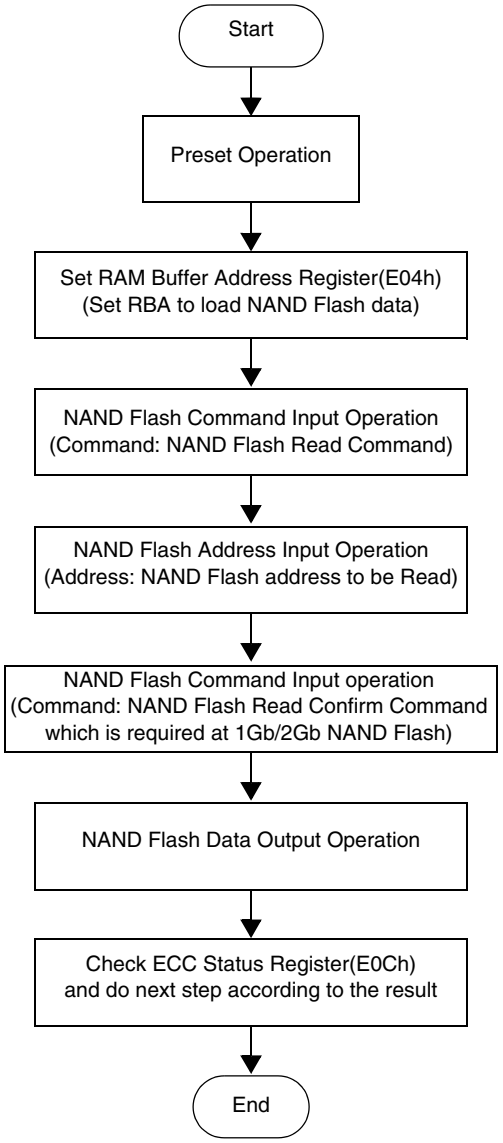


Figure 19-14. Read NAND Flash Data Operation

19.5.2.4 Program NAND Flash Data Operation

The sequence of events of the NAND Flash controller during Program NAND Flash Data operation is shown in Figure 19-15.

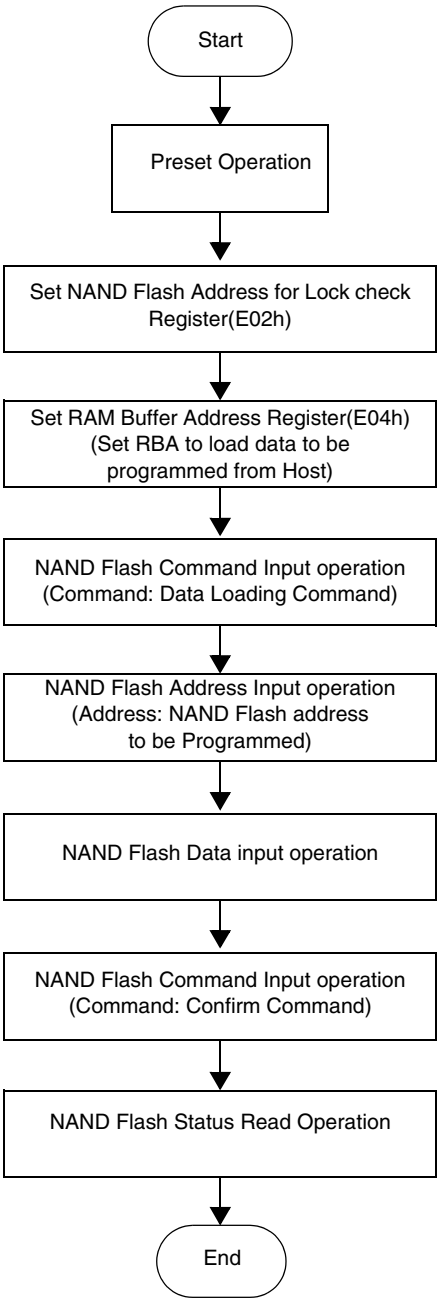


Figure 19-15. Program NAND Flash Data Operation

19.5.2.5 Erase NAND Flash Data Operation

The sequence of events of the NAND Flash controller during Erase NAND Flash Data operation is shown in Figure 19-16.

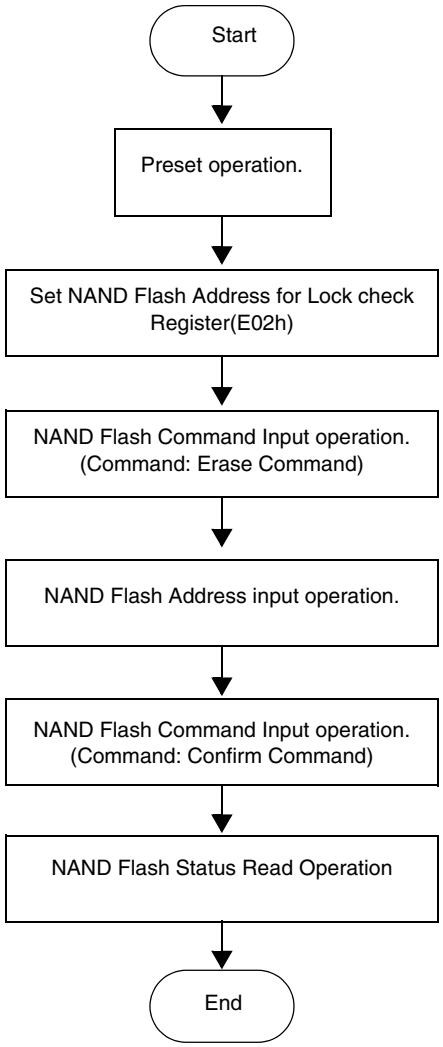


Figure 19-16. Erase NAND Flash Operation

19.5.2.6 Hot Reset

The Hot Reset operation resets the NAND Flash device. The NAND Flash device stops its current operation and its internal registers go to their default state.

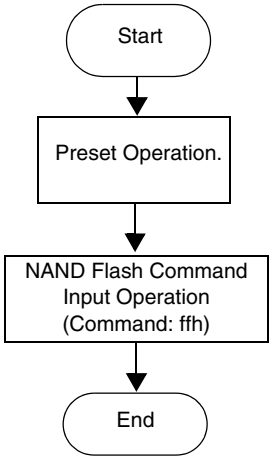


Figure 19-17. Hot Reset Operation

19.5.3 ECC Operation

The following sections discuss the details of ECC operation.

19.5.3.1 ECC Operation Case

When NFC accesses NAND Flash for program operation, it generates ECC code (24 bits for main area data and 10 bits for LSN of spare area data). For a read operation it generates ECC code, detects number of error bits, and corrects a 1-bit error.

Table 19-22 shows the ECC code assignment of the NAND Flash spare area. This ECC code is updated by the NFC automatically. After a read operation, a host can know whether there was an error by reading the status register (Table 19-3). There are three error types: no error, 1-bit error (correctable), and 2 or more bit errors (uncorrectable).

The generated ECC code during a read or program operation is not updated to the internal buffer RAM. It is updated to the NAND Flash spare area directly during a program operation. A host can read the generated ECC code by reading back the programmed NAND Flash spare area.

Table 19-22. ECC Code / Result Readability

	Read Operation		Program Operation	
Operation	ECC code from spare area buffer	ECC result from register	ECC code from spare area buffer	ECC result from register
ECC Operation	Invalid (Pre-written ECC code[1])	Valid	Invalid (old data[2])	–
ECC Auto-correction Bypass	Invalid (Pre-written ECC code)	Valid	Invalid (old data)	–

1 Pre-written ECC code—ECC code previously written to NAND Flash spare area in program operation.

2 Old data—ECC code is not updated to spare buffer, so ECC code placement of spare buffer remains old data.

19.5.3.2 ECC Bypass Operation Case

In ECC bypass operation, the NFC generates ECC result that indicate the error position (refer to Table 19-22), but does not correct the error. After a Read operation, a host can know if there was an error by reading the status register (see Table 19-3).

As stated above, there are three error types: no error, 1-bit error (correctable), and 2 or more bit errors (uncorrectable). In the 1-bit error case, the host can correct the error after reading the ECC Result register. The ECC Bypass operation does not affect the Program operation. ECC is always generated and written to the NAND Flash Spare area always.

19.5.3.3 ECC Operation Guidance

For ECC generation and correction done by the NFC, it is recommended that the user program with an ECC operation and Read with an ECC operation. For ECC generation by the NFC and correction done by the Host, it is recommended that the user program with an ECC operation and Read with ECC bypassed operation.

The ECC result from the ECC result register after a read operation in both ECC operation and ECC bypass cases are always valid.

NOTE

When NFC reads NAND Flash data (page or Spare area data), the ECC code in the RAM buffer (Main or Spare) is the ECC Code from the NAND Flash and not what was internally generated by the NFC. The NFC generated ECC code is never updated to the internal buffer areas.

19.5.4 Write Protection

The following sections discuss the details of the Write Protection mechanism within the NFC.

19.5.4.1 Write Protection for RAM Buffer (LSB 1KB)

NFC offers a software Write Protection feature for the first two pages (main and spare area data) of the RAM buffer, which protects the RAM buffer data. This software Write Protection of the RAM buffer feature is used by setting a two bit value of the NFC configuration register ([1:0]). The default state is the Locked state; These two pages of RAM buffer (LSB) move to the locked state after a cold or warm reset. When this occurs, Write protection is enabled.

Write protection for the main or spare memory regions in the RAM buffer are described in Table 19-23. A state diagram of RAM buffer Write Protection is shown in Figure 19-18.

Table 19-23. Write Protection for Main/Spare RAM Buffer

Main Area	Spare Area	
1st page RAM buffer		Available for Write Protection
2nd page RAM buffer		
3rd page RAM buffer		Not Available for Write Protection Operation
4th page RAM buffer		

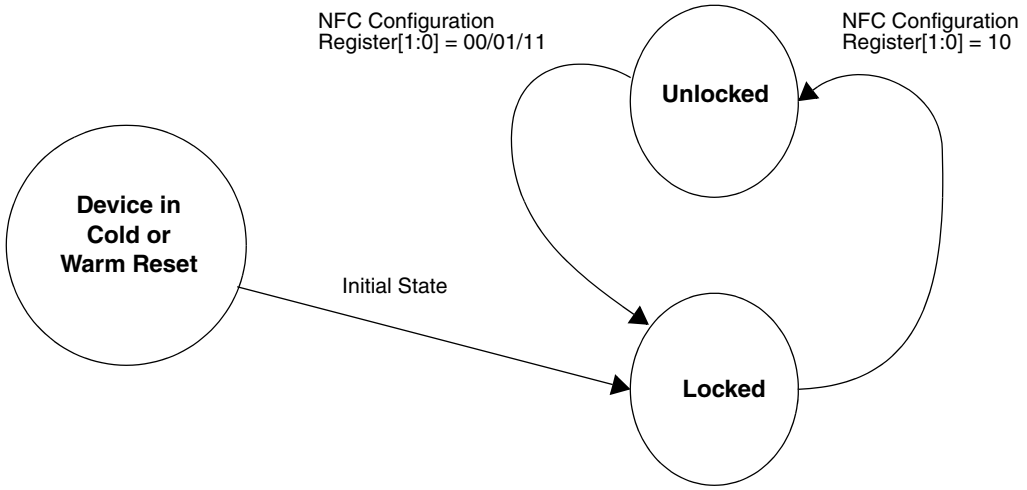


Figure 19-18. State Diagram of RAM Buffer Write Protection

19.5.4.2 Write Protection for NAND Flash

19.5.4.2.1 Write Protection Modes

NFC offers both hardware and software Write Protection features for NAND Flash. The software Write Protection feature is used by executing the Lock block command or Lock-tight block command, and the hardware Write Protection feature is used by executing cold reset or warm reset.

19.5.4.2.2 Write Protection Commands

Individual instant secured blocks protect code and data by allowing any block to be locked or lock-tightened. This Write Protection scheme offers two levels of protection. The first allows software-only control of Write Protection (useful for frequently changed data blocks), while the second requires hardware interaction before locking can be changed (protects infrequently changed code blocks).

The following list summarizes the locking functionality.

- All blocks power-up in a locked state. Unlock commands can unlock these blocks.
- The Lock-tight command locks blocks and prevents it from being unlocked. Lock-tight state can be returned to lock state only when Cold/Warm reset is executed.
- Writing to Unlock the start/end address register during Lock-tighten status does not affect the unlock address, since the NFC has another Unlock address register internally to prevent this.

19.5.4.2.3 Write Protection Status

The current NFC Write Protection status can be read in the NAND Flash Write Protection status register (NAND_Flash_WR_Pr_St). There are three bits—US, LS, LTS, which are not cleared by hot reset. These Write Protection status registers are updated as soon as the Write Protection command is executed. Figure 19-19 shows the state diagram of the NAND Flash Write Protection.

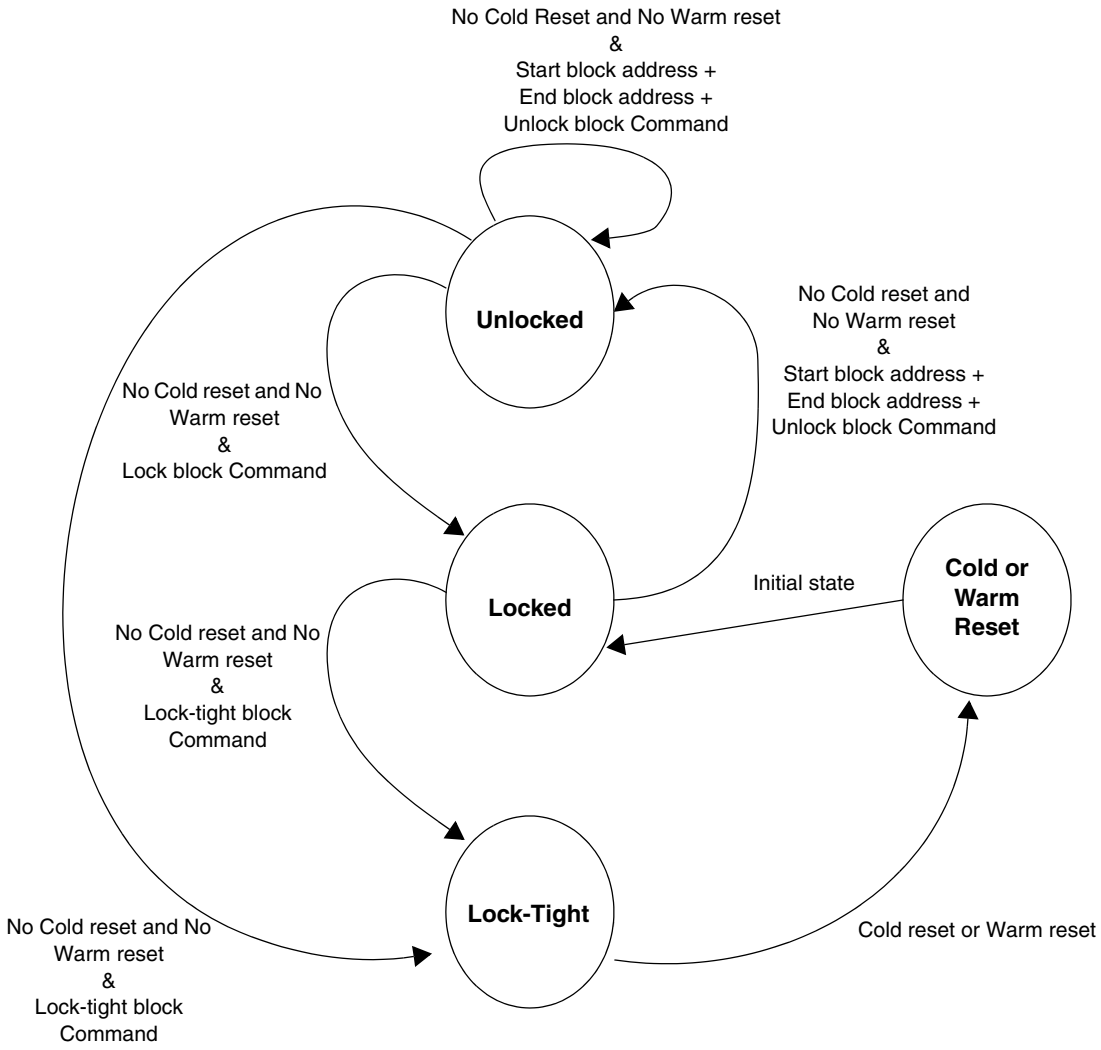


Figure 19-19. State Diagram of NAND Flash Write Protection

19.5.4.3 Lock Sequence

Use the following command sequence to Lock a block.

1. Command sequence: Lock block Command (02h).
2. All blocks default to locked after initial cold reset or warm reset.
3. Partial block lock is not available; Lock block operation is based on all block units.
4. Unlocked blocks can be locked by using the Lock block command, and a block's lock status can be changed to unlock or lock-tight using the appropriate software commands.

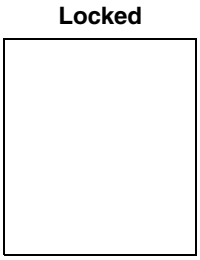


Figure 19-20. Lock Sequence

19.5.4.4 Unlock Sequence

Use the following command sequence:

1. Start block address + End block address + Unlock Block Command(04h)
2. Unlocked blocks can be programmed or erased
3. An unlocked block's status can be changed to the locked or lock-tighten state using the appropriate software commands
4. Only one sequential area can be released to from locked state to unlocked state; Unlocking multiple areas is not available

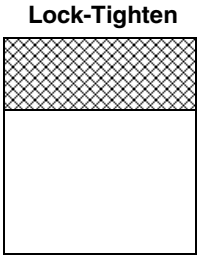


Figure 19-21. Unlocked Blocks During Command Sequence

19.5.4.5 Lock-Tight Sequence

Command Sequence, Lock-tight block Command (01h):

1. Lock-tightened blocks offer the user an additional level of write protection beyond that of a regular lock block. A block that is lock-tightened cannot have its state changed by software and can only be unlocked by a Cold or Warm reset. Unlocking multiple areas is not available.
2. Only locked blocks can be lock-tighten by lock-tight command. Lock-tighten blocks revert to the locked state at cold or warm reset.

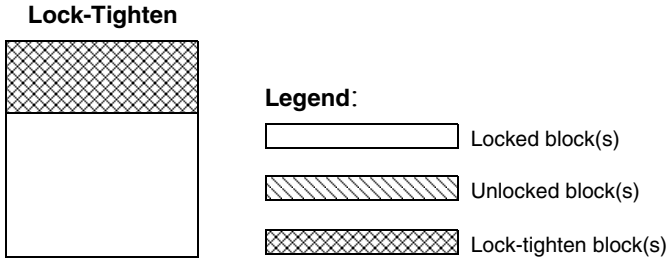


Figure 19-22. Sequential Area Affected by Lock-Tight Command

19.6 AHB Bus Operation

The connection of the Host is done through the AHB Interface. This AHB Interface supports Reads and Write, Burst or Non-Burst operations and 16-bits or 32-bits bus transfers. The bus is Little Endian byte ordering only.

An operation on the AHB bus starts the NON-SEQ command on the *htrans* signals, and with the *hsel* signal of the NFC. It ends on the next IDLE or NON-SEQ command on the *htrans* signals.

There are a different number of Wait States on the AHB bus depending on the current operation, and the previous operation. Table 19-24 shows the number of wait states on the AHB bus for Non-Burst operations or the first operation of a Burst.

Table 19-24. Single (Non-Burst) AHB Operation Wait-States on AHB Bus

Previous Operation Next Operation	IDLE AHB IDLE (No AHB Operations)	WRITE WRITE of 16 or 32 bits (AHB Word or Half Word) to REGISTER or MEMORY Region	READ READ of 16 or 32 bits (AHB Word or Half Word) to REGISTER or MEMORY Region
WR16 WRITE of 16 bits (AHB Half Word) to REGISTER or MEMORY region	2	2	2
WR32 WRITE of 32 bits (AHB Half Word) to REGISTER or MEMORY region	5	5	5
R16-REG READ of 16 bits (AHB Half Word) to REGISTER region	5	5	6
R16-MEM READ of 16 bits (AHB Half Word) to MEMORY region	7	7	8
R32-REG READ of 32bits (AHB Word) to REGISTER region	6	6	7
R32-MEM READ of 32 bits (AHB Word) to MEMORY region	8	8	9

An extension to burst operation would be as follows:

1. For WRITE BURST the first cycle would be according to Table 16-25 and the rest would take 0 wait states For example Writing a burst of Four 32 bit words to the Internal Memory buffer (no matter what was the previous operation) would result in 2-1-1-1 access cycles.
2. For READ BURST the first cycle would be according to Table 19-24 and the rest of the burst would be ZERO wait states. For example Reading a burst of Four 32 bit words from Internal Memory buffer that occurs after any NFC read would result in 3-1-1-1 access cycles.

Table 19-25 summarizes the number of Wait States for all kinds of burst modes available in the NFC module. It is divided into the latency of the first cycle, and the latency of the remaining cycles.

Table 19-25. Burst Wait-States on AHB Bus

AHB Bus Operation	LATENCY (Number of Wait States for the First Cycle in Burst)	BURST W.S. Number of Wait States for the Second and Above Cycles in Burst
WR16 WRITE of 16 bits (AHB Half Word) to REGISTER or MEMORY region	Always 2 W.S.	2 W.S.
WR32 WRITE of 32 bits (AHB Half Word) to REGISTER or MEMORY region	Always 5 W.S.	5 W.S.
R16 READ of 16 bits (AHB Half Word) to REGISTER or MEMORY region	4, 5, 6 or 7 W.S. (According to Table 19-24)	No W.S. (Zero W.S.)
R32 READ of 16 bits (AHB Half Word) to REGISTER or MEMORY region	5, 6, 7 or 8 W.S. (According to Table 19-24)	1 W.S. (One W.S.)

Table 19-26 shows example memory parts that are suitable for use with the corresponding configurations. These bits are configured by hardware when NAND Flash boot is selected and configured by host software otherwise.

Table 19-26. Settings for Samsung Flash Memory

NFC_FMS	NF_16BIT_SEL	Memory Device
0	0	SAMSUNG K9F5608 (32M × 8-bit) Page size is 528 bytes (512 bytes + 16 bits)
0	1	SAMSUNG K9F5616 (16M × 16-bit) Page size is 528 bytes (512 bytes + 16 bits)
1	0	SAMSUNG K9F1G08 (128M × 8-bit) Page size is 2112 bytes (2048 bytes + 64 bits)
1	1	SAMSUNG K9F1G16 (64M × 16-bit) Page size is 2112 bytes (2048 bytes + 64 bits)

Chapter 20
External Interface Module (EIM)

20.1 Overview

The External Interface Module (EIM) handles the interface to devices external to the i.MX21 chip, including generation of chip selects for external peripherals and memory, and provides the following features:

- Six chip selects for external devices: $\overline{CS}$[1:0], covering a range of 64 Mbytes, and $\overline{CS}$[5:2], covering a range of 16 MBytes each
- Selectable protection for each chip select
- Reset programmable data port size for $\overline{CS}$[0] depending on BOOT mode
- Programmable data port size for each chip select
- Address suppression during burst mode operations
- Synchronous burst mode support for burst flash devices
- Synchronous burst mode support for PSRAM (Pseudo-SRAM) devices
- Programmable wait-state generator for each chip select
- Supports positive edge triggered $\overline{DTACK}$ signal operations for $\overline{CS}$[5]
- Supports level sensitive $\overline{DTACK}$ signal operations for $\overline{CS}$[3:0] and $\overline{CS}$[5]

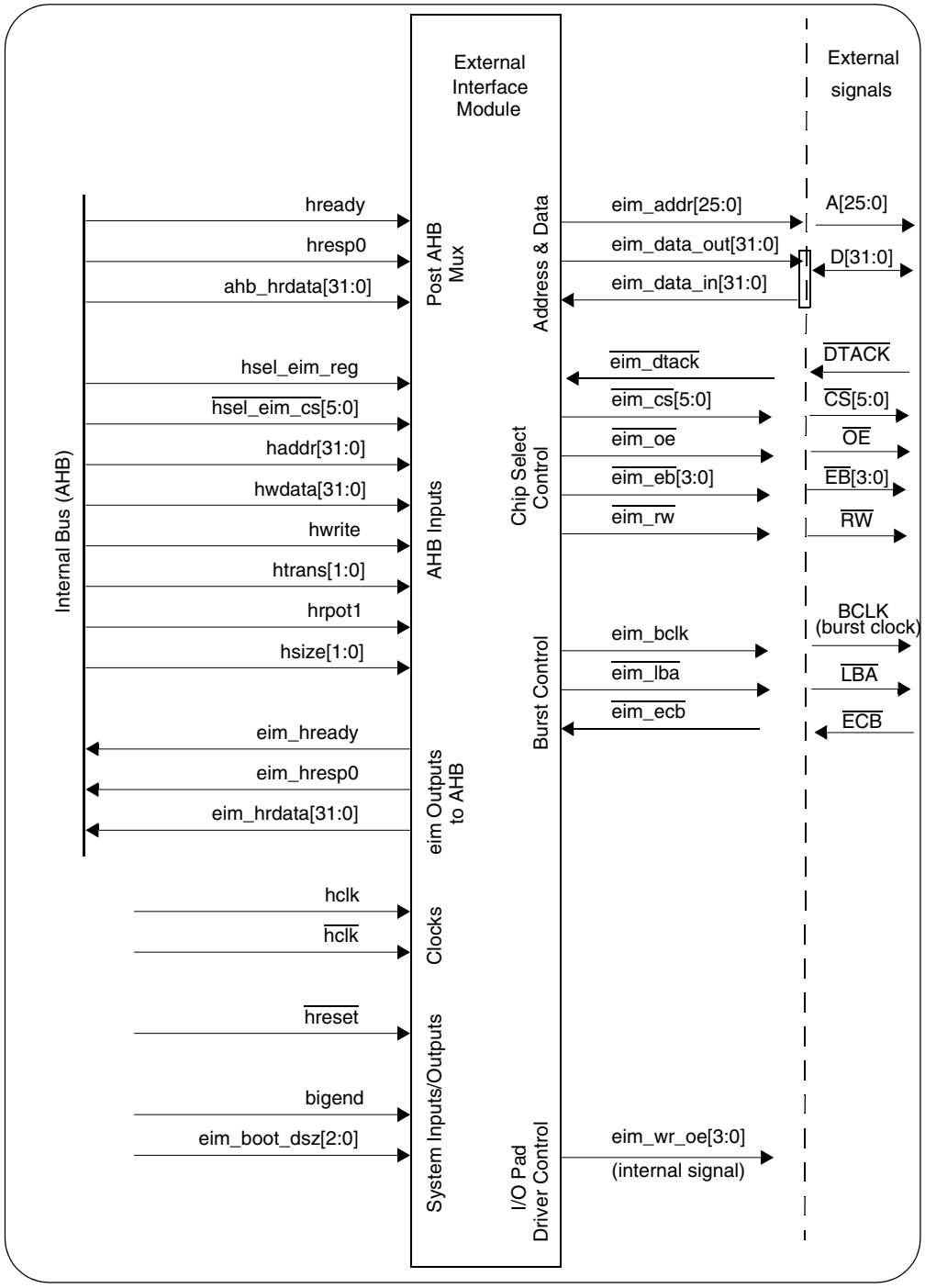


Figure 20-1. EIM Block Diagram

20.2 EIM I/O Signals

The EIM I/O signals provide communication and control pathways between external devices and the i.MX21 chip. A summary of the I/O signal pins is provided in Table 20-1. Each signal is described in the following sections.

Table 20-1. EIM I/O Signal Descriptions

Pin name	Direction	Description
D[31:0]	Bidirectional	External Data bus
A[25:0]	Output	External Address bus outputs
$\overline{CS}$[5:0]	Output	Active Low External Chip Selects
$\overline{EB}$[3:0]	Output	Active low external Enable Byte signals. $\overline{EB}$[0] controls D[31:24] data bits, $\overline{EB}$[1] controls D[23:16] data bits, $\overline{EB}$[2] controls D[15:8] data bits, $\overline{EB}$[3] controls D[7:0] data bits.
$\overline{OE}$	Output	Active Low output enable for external data bus
BCLK (burst clock)	Output	Clock used for external synchronous memories
$\overline{LBA}$	Output	Active low Load Burst Address indicates to external synchronous memories that Address is valid for latching
$\overline{RW}$	Output	Indicates if external access is a read (high) or write (low) cycle
$\overline{ECB}$	Input	Input signal that identifies when to end an external access, active low
$\overline{DTACK}$	Input	This signal is the external input data acknowledge signal multiplexed with $\overline{CS}$[4].

20.2.1 Chip Select Signals

The $\overline{CS}$[5] output signals are active-low and are asserted based on a decode of the internal address bus bits [31:24] of the accessed address. Table 20-2 specifies the address range for each Chip Select output provided by external address decoder.

Table 20-2. Chip Select Address Range (Provided by External Address Decoder)

[x]	CSEN Bit in CS[x] Lower Register	Internal Address [31:24] Decode	Chip Select Active	Memory map Address range
all	cleared	–	inactive	–
0	set	0xC8, 0xC9,0xCA,0xCB	$\overline{CS}$[0]	0xC8000000–0xCBFFFFFF
1	set	0xCC,0xCD,0xCE,0xCF	$\overline{CS}$[1]	0xCC000000–0xCFFFFFFF
2	set	0xD0	$\overline{CS}$[2]	0xD0000000–0xD0FFFFFF
3	set	0xD1	$\overline{CS}$[3]	0xD1000000–0xD1FFFFFF
4	set	0xD2	$\overline{CS}$[4]	0xD2000000–0xD2FFFFFF
5	set	0xD3	$\overline{CS}$[5]	0xD3000000–0xD3FFFFFF

20.2.2 Burst Mode Signals

The burst mode signals for the EIM module are described in Table 20-3.

Table 20-3. Burst Mode Signal Description

Name	Description
BCLK (burst clock)	**Burst Clock**—The BCLK output signal is used to clock external burst capable devices to synchronize the loading and incrementing of addresses, and delivery of burst read data to the EIM. This signal is derived from the internal system clock HCLK and its behavior is affected by the BCM bit in the EIM configuration register and the SYNC, BCD, PME, and BCS bits in the EIM chip select control registers. *Note, this signal is not to be confused with the internal ARM CPU BCLK signal. The BCLK signal name used throughout this chapter is referenced only to the external burst clock.*
$\overline{\text{LBA}}$	**Load Burst Address**—The $\overline{\text{LBA}}$ active-low output signal is asserted during burst mode accesses to cause the external burst capable device to load a new starting burst address. Assertion of $\overline{\text{LBA}}$ indicates that a valid address is present on the address bus. Its behavior is affected by the SYNC, BCD, PME and BCS bits in the EIM chip select control registers.
$\overline{\text{ECB}}$	**End Current Burst**—The $\overline{\text{ECB}}$ active-low input signal is asserted by external burst capable devices to indicate the end of the current (continuous) burst sequence. Following assertion, the EIM terminates the current burst sequence and initiates a new one. For synchronous PSRAM devices, this input may be used as the $\overline{\text{WAIT}}$ input during burst transfers. In this case, the memory device asserts this signal to insert wait states during refresh collisions or during a row boundary crossing. For this scenario the EIM "WAIT" mode should be used (EW=1) in order not to terminate the burst. The EIM then samples the data once $\overline{\text{WAIT}}$ is negated. The EW bit, which selects the EIM WAIT mode operation for the $\overline{\text{ECB}}$ signal, can be found in the EIM chip select control register. For burst memories the $\overline{\text{ECB}}$/$\overline{\text{WAIT}}$ output should be configured to change one cycle before data is ready (before delay).

20.3 Pin Configuration for EIM

Table 20-4 lists the pins used for the EIM module. Some of them are multiplexed with other functions on the device.

Table 20-4. Pin Configuration

Internal EIM signals	i.MX21 Pins	Setting
eim_data_out[31:0]	D [31:0]	Dynamically multiplexed
eim_data_in[31:0]	D [31:0]	Dynamically multiplexed
eim_addr[24:0]	A [24:0]	Some of the Address signals are multiplexed with the NFIO signals (refer to the Signal and Pins Chapter)
$\overline{\text{eim\_cs}}$[5]	$\overline{\text{CS5}}$	Multiplexed with GPIO PF22
$\overline{\text{eim\_cs}}$[4]	$\overline{\text{CS4}}$	Multiplexed with GPIO PF21 and $\overline{\text{DTACK}}$
$\overline{\text{eim\_cs}}$[3]	$\overline{\text{CS3}}$	Multiplexed with $\overline{\text{CSD1}}$
$\overline{\text{eim\_cs}}$[2]	$\overline{\text{CS2}}$	Multiplexed with $\overline{\text{CSD0}}$
$\overline{\text{eim\_cs}}$[1]	$\overline{\text{CS1}}$	Not multiplexed
$\overline{\text{eim\_cs}}$[0]	$\overline{\text{CS0}}$	Not multiplexed
$\overline{\text{eim\_eb}}$[3:0]	$\overline{\text{EB3}}$ … $\overline{\text{EB0}}$	Dynamically multiplexed with the SDRAM DQM signals

Table 20-4. Pin Configuration (continued)

Internal EIM signals	i.MX21 Pins	Setting
$\overline{\text{eim\_oe}}$	$\overline{\text{OE}}$	Not Multiplexed
$\overline{\text{eim\_dtack}}$	$\overline{\text{CS4}}$	System GPIO muxing (multiplexed with $\overline{\text{DTACK}}$)
eim_bclk	BCLK	Not multiplexed
$\overline{\text{eim\_lba}}$	$\overline{\text{LBA}}$	Not multiplexed
$\overline{\text{eim\_rw}}$	$\overline{\text{RW}}$	Not multiplexed
$\overline{\text{eim\_ecb}}$	$\overline{\text{ECB}}$	Not multiplexed

20.4 Typical EIM System Connections

Figure 20-2 shows an example of a typical set of EIM interfaces to external memory and peripherals. Figure 20-3 on page 20-7 shows the EIM interface to two supported external burst flash devices.

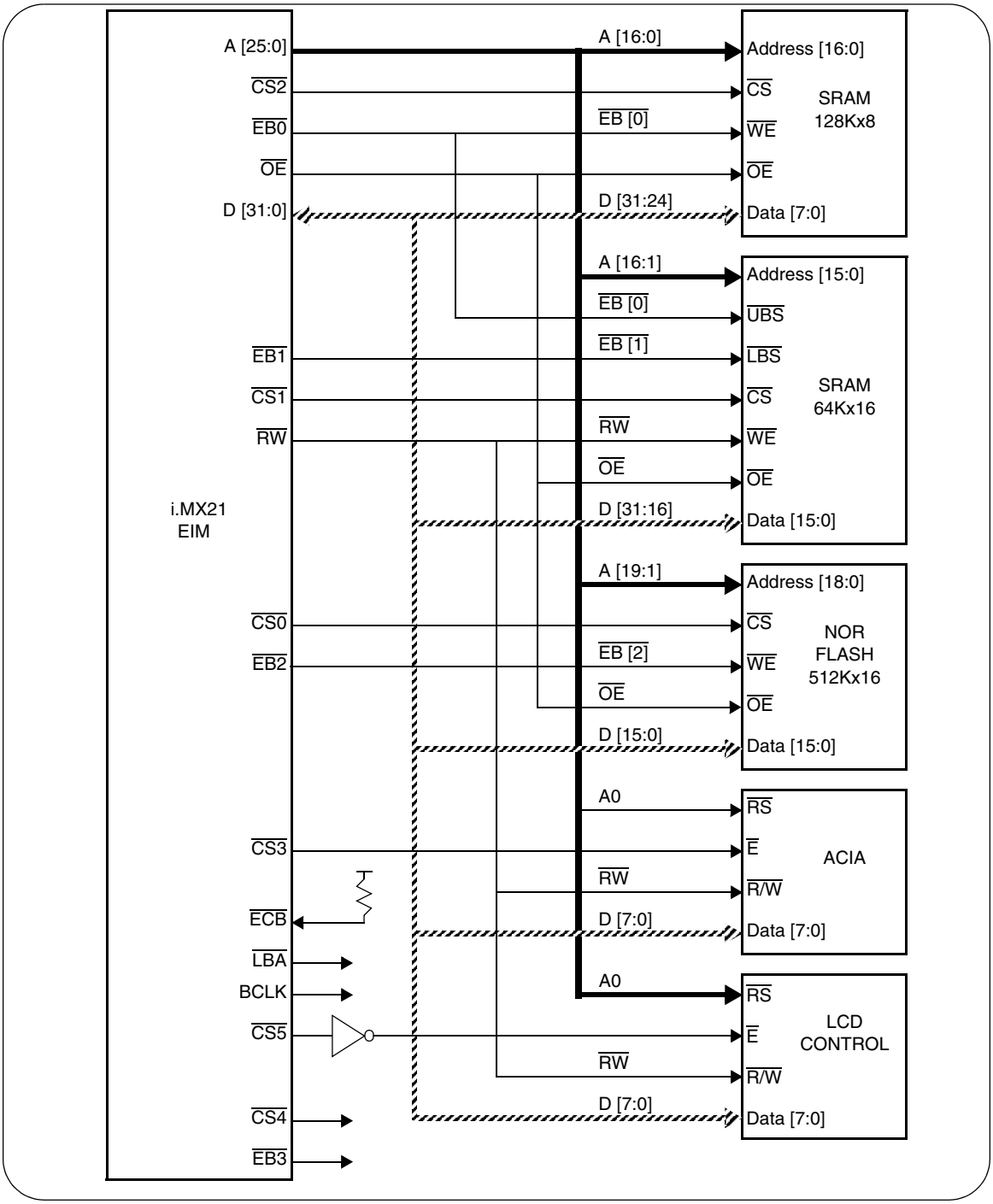


Figure 20-2. Example of EIM Interface to Memory and Peripherals

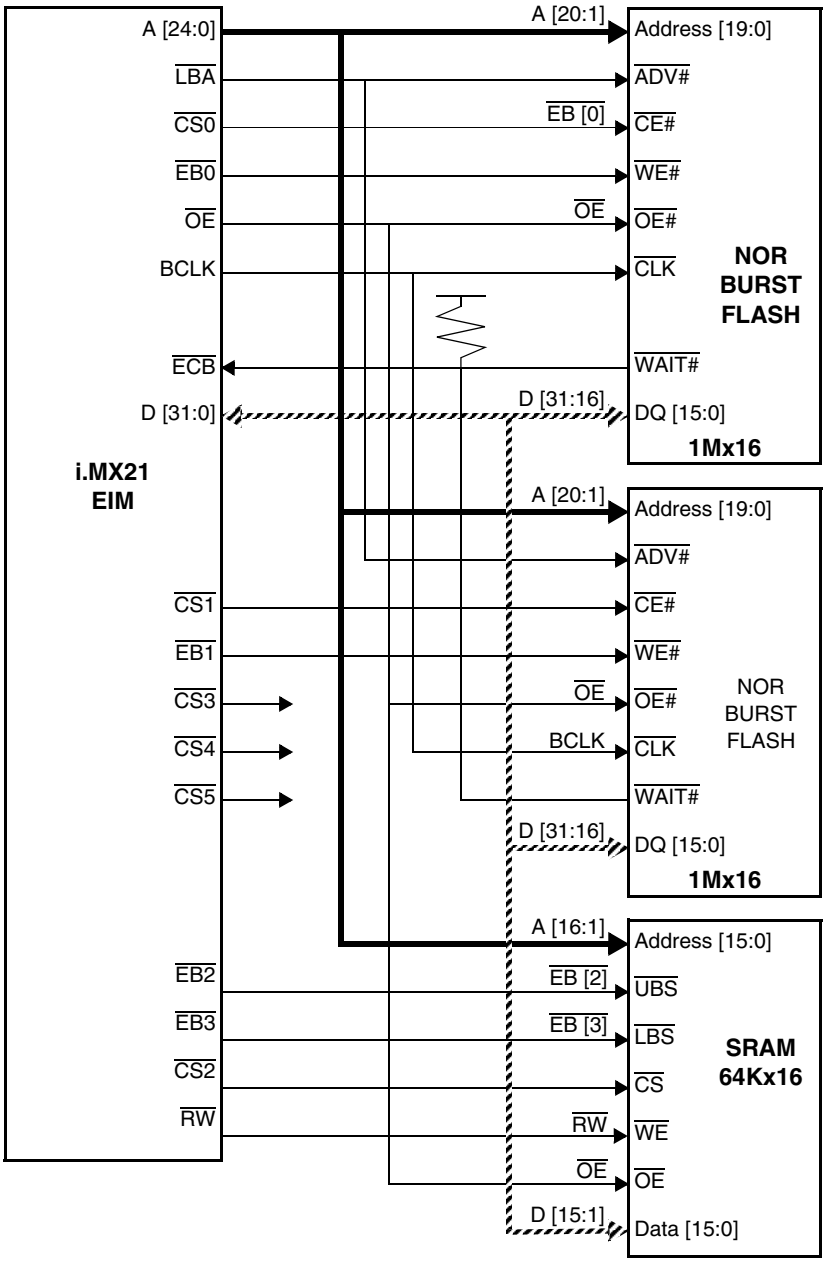


Figure 20-3. Example for EIM Interface to Burst Memory

20.5 EIM Functionality

20.5.1 Configurable Bus Sizing

The EIM supports byte, halfword, and word operands, allowing access to 8-bit ports, 16-bit ports, and 32-bit ports. It does not support misaligned transfers.

The port size is programmable via the DSZ bits in the corresponding chip select control register. In addition, the portion of the data bus used for transfer to or from an 8-bit port or 16-bit port is programmable

via the same bits. An 8-bit port can reside on external data bus bits D[31:24], D[23:16], D[15:8] or D[7:0]. A 16-bit port can reside on external data bus bits D[31:16] or D[15:0].

Word access to or from an 8-bit port requires four external bus cycles to complete the transfer. Word access to or from a 16-bit port requires two external bus cycles to complete the transfer. Halfword access to or from an 8-bit port requires two external bus cycles to complete the transfer. In the case of a multi-cycle transfer, the lower two address bits [1:0] are incremented appropriately.

The EIM has a data multiplexer that routes the four bytes of the AHB interface data bus to the required positions to allow proper interfacing to memory and peripherals.

NOTE

The EIM does not support 8-bit or 16-bit internal AHB burst accesses. The ARM core only issues 32-bit word burst accesses, but the bus masters such as the DMA are also capable of 8-bit and 16-bit internal burst transfers. When programming these bus masters, the user must configure them to access the EIM using 32-bit burst transfers. Hence, internal byte and half-word burst transfers to/from the EIM are not allowed.

20.5.2 Burst Mode Operation

When there are sequential accesses, the EIM attempts to burst read data from as many sequential address locations as possible, limited only by the length of the burst flash internal page buffer, or the non-sequential nature of the ARM926EJ-S processor code or data stream. The EIM only displays the first address accessed in a burst sequence unless the page mode emulation (PME) bit is set. To enable burst accesses from the EIM, the SYNC bit must be set, along with proper settings for BCD, DOL, WSC, and BCS.

For the first access in a burst sequence, the EIM asserts load burst address ($\overline{\text{LBA}}$), causing the external burst device to latch the starting burst address, and then toggles the burst clock (BCLK) for a predefined number of cycles in order to latch the first unit of data. Subsequently read data units are burst from the external device in fewer clock cycles, realizing an overall increase in bus bandwidth.

Burst accesses is terminated by the EIM when it detects that the next ARM926EJ-S processor access is not sequential, or when the external burst device needs additional cycles to retrieve the next requested memory location. In the latter case, the burst flash device provides an $\overline{\text{ECB}}$ signal to the EIM whenever it either terminates the on-going burst sequence and initiates a new (long first access) burst sequence or waits for $\overline{\text{ECB}}$ negation (see EW bit description for the details).

The synchronous mode is also used for burst PSRAM, which supports burst writes. For this, the PSR bit needs to be set in the chip select control register.

20.5.3 Burst Clock Divisor

In some cases it is necessary to slow the external bus in relation to the internal bus to allow accesses to burst devices that have a maximum operating frequency that is lower than the operating frequency of the internal AHB bus. The internal bus frequency (HCLK) can be divided by 2, 3, or 4 for presentation on the external bus in burst mode operation. In fact, when connecting to a 32-bit data bus burst flash, the minimum value BCD should be set for is divide by two. This is due to the one wait state latency required

by the EIM to capture data and present it to the AHB bus. For a 16-bit data bus, BCD can be set to divide by one since it takes two external data fetches to form one 32-bit data word. In either case, check the burst flash data sheet to ensure the burst clock does not violate the burst flash maximum burst clock timing.

Programming the BCD bits to various values (see Table 20-6) affects two signals on the external bus, $\overline{\text{LBA}}$ (load burst address) and BCLK (burst clock). The $\overline{\text{LBA}}$ signal is asserted immediately and remains asserted until the first falling edge of the BCLK signal. The BCLK signal runs with a 50% duty cycle until a non-sequential internal request is received or an external $\overline{\text{ECB}}$ signal is recognized.

When programming these bits, ensure that the WSC and DOL bit fields are coordinated to provide the desired external bus waveforms. For example, when the BCD bits are programmed to 01, the DOL bits should be programmed to 0001, 0011, 0101, … When the BCD bits are programmed to 10, the DOL should be programmed to 0010, 0101, 1000, …

20.5.4 Burst Clock Start

To allow greater flexibility in achieving the minimum number of wait states on bursted accesses, the user can determine when they want the BCLK (burst clock) to start toggling. This allows the BCLK to be skewed from the point of data capture on the system clock by any number of system clock phases. Use caution when setting these bits in conjunction with the BCD, WSC, and DOL bits. See the external timing diagrams for some examples of how to use the BCS, BCD, WSC, and DOL bits together.

20.5.5 Page Mode Emulation

Setting the PME bit along with SYNC bit causes the EIM to perform bursted accesses by emulating page mode operation. The $\overline{\text{LBA}}$ signal remains asserted for the entire access, the burst clock does not send a signal, and the external address asserts for each access made. The initial access timing is dictated by the WSC bits and the page mode access timing is dictated by the DOL bits.

The EIM can take advantage of improved page timing for sequential accesses only. Accesses that are on the page but are not sequential in nature have their timing dictated by the WSC bits. The page size can be set via the PSZ bits to 4, 8, 16, or 32 words (the word size is determined by the data width of the external memory, such as determined by the DSZ bits).

20.5.6 PSRAM mode operation

A control bit PSR is provided to enable PSRAM operation. Bit CRE and an unused address line A[25] can be used to drive the control register enable (CRE) PSRAM input for PSRAM configuration registers loading. The PSRAM CRE signal should be connected to A[25] for this purpose and the bit CRE set when programming the PSRAM control registers.

20.5.7 Mixed Burst Mode Support

To provide mixed sequential/wrap accesses of different lengths, the EIM tests the internal AHB hburst signal to indicate the burst access type and size and will generate additional $\overline{\text{LBA}}$ signals whenever there appears to be an unequal wrap condition. The chip select control register bits PSZ and WRAP should be programmed to notify EIM about the current memory wrap condition for proper external address

generation. In other words, the EIM chip select control register bits PSZ and WRAP should be programmed to match the external device's memory wrap condition. In the case where the wrap boundaries in both the PSZ and AHB access do not match, the EIM puts the address of the next access on the address bus and generates LBA signal. Burst accesses from AHB must be word width. The EIM does not support burst accesses of byte or half-word width.

20.5.8 DTACK Signal Operation

When enabled, the DTACK input signal is used to externally terminate a data transfer. For DTACK enabled operations, a bus time-out monitor generates a bus error when an external bus cycle is not terminated by the DTACK input signal after 1024 HCLK clock cycles have elapsed, where HCLK is the internal system clock driven from the PLL module. For a 133MHz HCLK setting, this time equates to 7.7us. Refer to the PLL Control Chapter for more information on how to generate different HCLK frequencies.

There are two modes of operation for the DTACK input signal: rising edge detection or level sensitive detection with a programmable insensitivity time. DTACK is only used during external asynchronous data transfers, thus the SYNC bit in the chip select control registers must be cleared.

During edge detection mode, the EIM will terminate an external data transfer following the detection of the DTACK signal's rising edge, so long as it occurs within the 1024 HCLK cycle time. Edge detection mode is used for devices that follow the PCMCIA standard. Note that DTACK rising edge detection mode can only be used for CS[5] operations. To configure CS[5] for DTACK rising edge detection, the following bits must be programmed in the Chip Select 5 Control Register and EIM Configuration Register:

- WSC bit field set to 0x3F and CSA (or CSN) set to 1 or greater in the Chip Select 5 Control Register
- AGE bit set in the EIM Configuration Register

Other bits such as DSZ, OEA, OEN, and so on, may be set according to system and timing requirements of the external device. The requirement of setting CSA or CSN is needed to allow the EIM to wait for the rising edge of DTACK during back-to-back external transfers, such as during DMA transfers or an internal 32-bit access through an external 16-bit data port.

During level sensitive detection, the EIM will first hold off sampling the DTACK signal for at least 2 HCLK cycles, and up to 5 HCLK cycles as programmed by the DCT bits in the Chip Select Control Register. After this insensitivity time, the EIM will sample DTACK and if it detects that DTACK is logic high, it will continue the data transfer at the programmed number of wait states. However, if the EIM detects that DTACK is logic low, it will wait until DTACK goes to logic high to continue the access, so long as this occurs within the 1024 HCLK cycle time. If at anytime during an external data transfer DTACK goes to logic low, the EIM will wait until DTACK returns to logic high to resume the data transfer. Level detection is often used for asynchronous devices such graphic controller chips. Level detection may be used with any chip select except CS[4] as it is multiplexed with the DTACK signal. To configure a chip select for DTACK level sensitive detection, the following bits must be programmed in the Chip Select Control Register and EIM Configuration Register:

- EW bit set, WSC set to > 1, and CSN set to < 3 in the Chip Select Control Register

- BCD/DCT set to desired “insensitivity time” in the Chip Select Control Register. The “insensitivity time” is dictated by the external device’s timing requirements.
- AGE bit cleared in the EIM Configuration Register

Other bits such as DSZ, OEA, OEN, and so on, may be set according to system and timing requirements of the external device.

20.5.9 Error Conditions

The following conditions cause an error condition to be asserted to the ARM926EJ-S processor:

- Access to a disabled chip select (access to a mapped chip select address space where the CSEN bit in the corresponding chip select control register is clear)
- Write access to a write-protected chip select address space (the WP bit in the corresponding chip select control register is set)
- User access to a supervisor-protected chip select address space (the SP bit in the corresponding chip select control register is set)
- User read or write access to a chip select control register or the EIM configuration register
- Byte or halfword access to a chip select control register or the EIM configuration register
- $\overline{\text{DTACK}}$ acknowledge is absent for more then 1024 HCLK clock cycles.

20.6 Programming Model

The EIM module includes thirteen user-accessible 32-bit registers. There is a common register called the EIM Configuration Register that contains control bits that configure the EIM for certain operation modes. The other twelve registers are pairs of control registers for each chip select. This registers are accessible only in supervisor mode with word (32-bit) reads and writes.

Decoding is provided by external address decoder, so shadowing can occur with these registers. The user should not attempt to address these registers at any other address location other than those listed in Table 20-5.

Table 20-5. EIM Module Register Summary (Provided by External Address Decoder)

Description	Name	Address
Chip Select 0 Upper Control Register	CS0U	0xDF001000
Chip Select 0 Lower Control Register	CS0L	0xDF001004
Chip Select 1 Upper Control Register	CS1U	0xDF001008
Chip Select 1 Lower Control Register	CS1L	0xDF00100C
Chip Select 2 Upper Control Register	CS2U	0xDF001010
Chip Select 2 Lower Control Register	CS2L	0xDF001014
Chip Select 3 Upper Control Register	CS3U	0xDF001018
Chip Select 3 Lower Control Register	CS3L	0xDF00101C
Chip Select 4 Upper Control Register	CS4U	0xDF001020

Table 20-5. EIM Module Register Summary (Provided by External Address Decoder) (continued)

Description	Name	Address
Chip Select 4 Lower Control Register	CS4L	0xDF001024
Chip Select 5 Upper Control Register	CS5U	0xDF001028
Chip Select 5 Lower Control Register	CS5L	0xDF00102C
EIM Configuration Register	EIM_CNF	0xDF001030

20.6.1 Chip Select 0 Control Registers

The layout of the Chip Select 0 control registers are slightly different from the other Chip Select control registers because it reset state depends on the internal eim_boot_dsz[2:0] input, which is derived from the external BOOT[3:0] settings.

The 64 bits of control are divided into two registers, Chip Select 0 Upper Control Register and Chip Select 0 Lower Control Register.

- Bits [63:32] are located in Chip Select 0 Upper Control Register.
- Bits [31:0] are located in Chip Select 0 Lower Control Register.

20.6.1.1 Chip Select 0 Upper Control Register

CS0U — Chip Select 0 Upper Control Register[1] — Addr 0xDF001000

BIT	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	SP	WP	BCD/DCT		BCS/RWA				PSZ		PME	SYNC	DOL/RWN			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	CNC		WSC						EW	WWS			EDC			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	0x3E00															

[1]—For bit descriptions, see Table 20-6.

20.6.1.2 Chip Select 0 Lower Control Register

CS0L	Chip Select 0 Lower Control Register[1]	Addr 0xDF001004

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OEA				OEN				WEA				WEN			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x2000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSA				EBC	DSZ			CSN				PSR	CRE	WRAP	CSEN
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	1	*	*	*	0	0	0	0	0	0	0	1
	0x0801															

[1]—For bit descriptions, see Table 20-6.

*—Configurable on reset, depends on BOOTMODE signals setting.

20.6.2 Chip Select 1 through Chip Select 5 Control Registers

The layout of the control registers for Chip Selects 1 through 5 are identical.

The 64 bits of control per chip select are divided into an Upper and a Lower register.

- Bits [63:32] are located in Chip Select x Upper Control Register.
- Bits [31:0] are located in Chip Select x Lower Control Register.

20.6.2.1 Chip Select 1 through Chip Select 5 Upper Control Registers

CS1U	Chip Select 1 Upper Control Register[1]	0xDF001008
CS2U	Chip Select 2 Upper Control Register[1]	0xDF001010
CS3U	Chip Select 3 Upper Control Register[1]	0xDF001018
CS4U	Chip Select 4 Upper Control Register[1]	0xDF001020
CS5U	Chip Select 5 Upper Control Register[1]	0xDF001028

BIT	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	SP	WP	BCD/DCT		BCS/RWA				PSZ		PME	SYNC	DOL/RWN			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	CNC		WSC						EW	WWS			EDC			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

[1]—For bit descriptions, see Table 20-6.

20.6.2.2 Chip Select 1 through Chip Select 5 Lower Control Registers

CS1L	Chip Select 1 Lower Control Register[1]	0xDF00100C
CS2L	Chip Select 2 Lower Control Register[1]	0xDF001014
CS3L	Chip Select 3 Lower Control Register[1]	0xDF00101C
CS4L	Chip Select 4 Lower Control Register[1]	0xDF001024
CS5L	Chip Select 5 Lower Control Register[1]	0xDF00102C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OEA				OEN				WEA				WEN			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSA				EBC	DSZ			CSN				PSR	CRE	WRAP	CSEN
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	0x0800															

[1]—For bit descriptions, see Table 20-6.

Table 20-6. Chip Select Control Registers Description

Name	Description	Settings
SP Bit 63	**Supervisor Protect**—Prevents accesses to the address range defined by the corresponding chip select when the access is attempted in the User mode of ARM9 core operation. SP is cleared by a hardware reset.	0 = User mode accesses are allowed in the range of chip select 1 = User mode accesses are prohibited; attempts to access an address mapped by this chip select in User mode results in a error respond on the AHB and no assertion of the chip select output
WP Bit 62	**Write Protect**—Prevents writes to the address range defined by the corresponding chip select. WP is cleared by a hardware reset.	0 = Writes are allowed in the range of chip select 1 = Writes are prohibited; attempts to write to an address mapped by this chip select result in a error respond on the AHB and no assertion of the chip select output

Table 20-6. Chip Select Control Registers Description (continued)

Name	Description	Settings
BCD/DCT Bits 61–60	**Burst Clock Divisor/ DTACK Check Time**— SYNC = 1 (BCD): Contains the value used to program the burst clock divisor. See Section 20.5.3, “Burst Clock Divisor,” for more information on the burst clock divisors. SYNC = 0 (DCT): Contains the value used to program the $\overline{\text{DTACK}}$ “insensitivity” capture time (number of HCLK cycles the EIM holds off sampling the $\overline{\text{DTACK}}$ signal) when EW = 1, when level sensitive $\overline{\text{DTACK}}$ has been enabled. The DCT setting determines the number of HCLK cycles between the $\overline{\text{CS}}$ assertion and the first sampling of the $\overline{\text{DTACK}}$ signal by the EIM. A setting of “00” dictates a minimum HCLK time of 2 HCLK cycles. BCD/DCT is cleared by a hardware reset.	SYNC = 1 (BCD): 00 = Divisor is 1 01 = Divisor is 2 10 = Divisor is 3 11 = Divisor is 4 SYNC = 0 (DCT): 00 = 2 HCLK cycles between the $\overline{\text{CS}}$ assertion and the first sampling of the $\overline{\text{DTACK}}$ signal 01 = 3 HCLK cycles between the $\overline{\text{CS}}$ assertion and the first sampling of the $\overline{\text{DTACK}}$ signal 10 = 4 HCLK cycles between the $\overline{\text{CS}}$ assertion and the first sampling of the $\overline{\text{DTACK}}$ signal 11 = 5 HCLK cycles between the $\overline{\text{CS}}$ assertion and the first sampling of the $\overline{\text{DTACK}}$ signal
BCS/RWA Bits 59–56	**Burst Clock Start / Read/Write Assertion** — If SYNC = 1 it determines the number of half cycles (HCLK cycles) after $\overline{\text{LBA}}$ assertion before the first rising edge of burst clock (BCLK) is seen and is controlled by the BCS bit field. A value of 0 results in a half clock delay, not an immediate assertion. When the BCM bit is set (BCM = 1) in the EIM configuration register, this overrides the BCS bits. If SYNC = 0 this bit field determines when $\overline{\text{RW}}$ is asserted during write cycles and is controlled by the RWA bit field. BCS/RWA is cleared by a hardware reset. Note: If RWA=0 and RWN is set to any value greater than or equal to 1, then the $\overline{\text{RW}}$ signal will assert one half HCLK cycle after start of access (as if RWA were set to 1). Programming RWA to a value other than zero in this condition will follow the settings indicated in the bit settings description.	*SYNC = 1 (BCS):* 0000 = 1 half HCLK cycle before burst clock (BCLK) 0001 = 2 half HCLK cycles before BCLK ... 1111 = 16 half HCLK cycles before BCLK *SYNC = 0 (RWA):* 0000 = $\overline{\text{RW}}$ asserts 0 half HCLK cycles after start of external access 0001 = $\overline{\text{RW}}$ asserts 1 half HCLK cycle after start of external access ... 1111 = $\overline{\text{RW}}$ asserts 15 half HCLK cycles after start of external access Note: The $\overline{\text{RW}}$ assertion time (as affected by RWA, RWN, and WSC bits) must be programmed to be greater than or equal to one HCLK cycle.
PSZ Bits 55–54	**Page Size**—Indicates the number of words (where “word” is defined by the port size or DSZ bits) in a page in memory. This ensures that the EIM does not burst pass a page boundary when the PME bit is set (along with SYNC=1 to enable page mode). When PME = 0: If WRAP = 1 the PSZ bits notify the EIM about the current memory word burst length (where “word” is defined by the port size or DSZ bits) and where to “wrap” in case of an unaligned access. If WRAP = 0, the PSZ bits simply notify the EIM the current memory word burst length. In either case, when SYNC = 1 and PME = 0, a PSZ setting of ‘11’ indicates a continuous burst. PSZ is cleared by a hardware reset.	00 = 4 words in a page 01 = 8 words in a page 10 = 16 words in a page 11 = 32 words in a page

Table 20-6. Chip Select Control Registers Description (continued)

Name	Description	Settings
PME Bit 53	**Page Mode Emulation**—Enables/Disables page mode emulation in burst mode (SYNC=1). When PME is set, the external address asserts for each piece of data requested. Additionally, the $\overline{\text{LBA}}$ and BCLK signals behave as they do when an asynchronous access is performed. PME is cleared by a hardware reset.	0 = Disables page mode emulation 1 = Enables page mode emulation
SYNC Bit 52	**Synchronous Burst Mode Enable**—Enables/Disables synchronous burst mode. When enabled, the EIM is capable of interfacing to burstable flash devices through additional burst control signals: BCLK, $\overline{\text{LBA}}$, and $\overline{\text{ECB}}$. The sequencing of these additional I/Os is controlled by other EIM configuration register bit settings as defined below. SYNC is cleared by a hardware reset.	0 = Disables synchronous burst mode 1 = Enables synchronous burst mode
DOL/RWN Bits 51–48	**Data Output Length / Read/Write Negation**—if SYNC = 1 it specifies the expected number of system clock cycles (HCLK) required for burst read data to be valid on the data bus before it is latched by the EIM. As system clock frequencies increase, it may become necessary to delay sampling the data for multiple system clock periods in order to meet burst flash max frequency specifications and/or EIM data setup time requirements. If SYNC = 0 this bit field determines when $\overline{\text{RW}}$ is negated (deasserted) during a write cycle. DOL has no effect on EIM data latching when SYNC = 0. DOL/RWN is cleared by a hardware reset.	*SYNC = 1 (DOL):* 0000 = 1 system clock HCLK delay 0001 = 2 system clock HCLK delay 0010 = 3 system clock HCLK delay ... 1111 = 16 system clock HCLK delay *SYNC = 0 (RWN):* 0000 = $\overline{\text{RW}}$ deasserts 0 half HCLK clocks before end of external access 0001 = $\overline{\text{RW}}$ deasserts 1 half HCLK clock before end of external access ... 1111 = $\overline{\text{RW}}$ deasserts 15 half HCLK clocks before end of external access Note: The $\overline{\text{RW}}$ assertion time (as affected by RWA, RWN, and WSC bits) must be programmed to be greater than or equal to one HCLK cycle.
CNC Bits 47–46	**Chip Select Negation Clock Cycles**—Specifies the minimum number of HCLK clock cycles a chip select must remain negated (deasserted) after it is negated. CNC has no effect on write accesses when any CSA or CSN bit is set. CNC is cleared by a hardware reset.	00 = Minimum Negation is 0 clock cycles 01 = Minimum Negation is 1 clock cycle 10 = Minimum Negation is 2 clock cycles 11 = Minimum Negation is 3 clock cycles

Table 20-6. Chip Select Control Registers Description (continued)

Name	Description	Settings
WSC Bits 45–40	**Wait State Control**— *For SYNC = 0:* WSC programs the number of wait-states for an access to the external device connected to the chip select. Table 20-7 shows the encoding of this field. When WWS is cleared, setting: WSC = 000000 results in 2 clock transfers WSC = 000001 results in 2 clock transfers WSC = 001110 results in 15 clock transfers WSC = 111110 results in 63 clock transfers WSC = 111111 is not available (reserved) for $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$ and $\overline{CS4}$, but it is available (the external DTACK) for $\overline{CS5}$. *For SYNC = 1:* WSC programs the number of system clock cycles required for the initial access of a burst sequence initiated by the EIM to an external burst device. See Table 20-7 and to the EIM synchronous burst read timing diagrams for further detail. WSC is set to 111110 by a hardware reset for $\overline{CS0}$. WSC is cleared by a hardware reset for $\overline{CS1}$–$\overline{CS5}$. For SYNC=0, and for $\overline{CS5}$: A WSC setting of 0x3F in conjunction with the AGE bit set to 1 in the EIM Configuration Register will enable $\overline{CS5}$ for DTACK rising edge triggered operation. **Note:** WSC bits should be configured to more than one wait states.	See Table 20-7
EW Bit 39	**ECB/WAIT**—determines how the EIM supports the $\overline{ECB}$ input. There are two modes: *For SYNC = 1:* $\overline{ECB}$ mode (EW=0): if $\overline{ECB}$ (End Current Burst) goes to low state in the middle of access then the EIM starts a new access by asserting the current AHB address to ADDR pins and along with an $\overline{LBA}$ assertion. WAIT mode (EW=1): if $\overline{ECB}$ goes to low state in the middle of access then the EIM will wait until $\overline{ECB}$/$\overline{WAIT}$ goes high to continue current access. *For SYNC = 0:* Setting EW while SYNC = 0 will enable the chip select for $\overline{DTACK}$ level sensitive operation. In addition, the AGE bit in the EIM Configuration Register must be cleared and the BCD/DCT bits set to the desired insensitivity time. EW is cleared by a hardware reset.	*SYNC = 1:* 0 = ECB mode 1 = WAIT mode *SYNC = 0:* 0 = $\overline{DTACK}$ level sensitive operation disabled 1 = $\overline{DTACK}$ level sensitive operation enabled
WWS Bits 38–36	**Write Wait State**—Determines whether additional wait-states are required for write cycles. This is useful for writing to memories that require additional data setup time. WWS is cleared by a hardware reset.	See Table 20-7

Table 20-6. Chip Select Control Registers Description (continued)

Name	Description	Settings
EDC Bits 35–32	**Extra Dead Cycles**—Determines whether idle HCLK cycles are inserted after a read cycle for back-to-back external transfers to eliminate data bus contention. This is useful for slow memory and peripherals that use long $\overline{CS}$ or $\overline{OE}$ to output data three-state times. Idle cycles are not inserted for back-to-back external reads from the same chip select. EDC is cleared by a hardware reset.	0000 = 0 Idle Cycles Inserted 0001 = 1 Idle Cycle Inserted ... 1111 = 15 Idle Cycles Inserted
OEA Bits 31–28	**$\overline{OE}$ Assert**—Determines when $\overline{OE}$ is asserted during read cycles. For SYNC = 0: OEA determines the number of half HCLK clocks before $\overline{OE}$ asserts during a read cycle. A setting of 0 causes $\overline{OE}$ to assert when the address is placed on the external bus, a setting of 1 causes $\overline{OE}$ to assert one half HCLK cycles after the address is placed on the external bus, etc. For SYNC = 1: After the initial burst access, $\overline{OE}$ is asserted continuously for subsequent burst accesses, and is not affected by OEA (see burst read timing diagrams for more detail). The behavior of $\overline{OE}$ on the initial burst access is the same as when SYNC = 0. When the EBC bit in the corresponding register is clear, the $\overline{EB}$ [3:0] outputs are similarly affected. OEA does not affect the cycle length. OEA is cleared by a hardware reset.	0000 = $\overline{OE}$ asserts 0 half clocks (HCLK) after start of external access 0001 = $\overline{OE}$ asserts 1 half clock (HCLK) after start of external access ... 1111 = $\overline{OE}$ asserts 15 half clocks (HCLK) after start of external access Note: The $\overline{OE}$ assertion time (as affected by OEA, OEN, and WSC bits) must be programmed to be greater than or equal to one HCLK cycle.
OEN Bits 27–24	**$\overline{OE}$ Negate**—Determines when $\overline{OE}$ is negated (deasserted) during a read cycle. Setting the SYNC bit (SYNC = 1) overrides OEN and $\overline{OE}$ deasserts at the end of a read access and no sooner. When EBC in the corresponding register is clear, the $\overline{EB}$ [3:0] outputs are similarly affected. OEN does not affect the cycle length. OEN is cleared by a hardware reset.	0000 = $\overline{OE}$ deasserts 0 half clocks (HCLK) before end of access 0001 = $\overline{OE}$ deasserts 1 half clock (HCLK) before end of access ... 1111 = $\overline{OE}$ deasserts 15 half clocks (HCLK) before end of access Note: The $\overline{OE}$ assertion time (as affected by OEA, OEN, and WSC bits) must be programmed to be greater than or equal to one HCLK cycle.
WEA Bits 23–20	**$\overline{EB}$ [3:0] Assert**—Determines when $\overline{EB}$ [3:0] is asserted during write cycles. This is useful to meet data setup time requirements for slow memories. WEA does not affect the cycle length. WEA is cleared by a hardware reset.	0000 = $\overline{EB}$[3:0] asserts 0 half clocks (HCLK) after start of external access 0001 = $\overline{EB}$[3:0] asserts 1 half clock (HCLK) after start of external access ... 1111 = $\overline{EB}$[3:0] asserts 15 half clocks (HCLK) after start of external access Note: The $\overline{EB}$[3:0] assertion time (as affected by WEA, WEN, and WSC bits) must be programmed to be greater than or equal to one HCLK cycle.

Table 20-6. Chip Select Control Registers Description (continued)

Name	Description	Settings
WEN Bits 19–16	**$\overline{EB}$ [3:0] Negate During Write**—Determines when $\overline{EB}$ [3:0] outputs are negated (deasserted) during a write cycle. This is useful to meet data hold time requirements for slow memories. WEN does not affect the cycle length. WEN is cleared by a hardware reset.	0000 = $\overline{EB}$[3:0] deasserts 0 half clocks (HCLK) before end of access 0001 = $\overline{EB}$[3:0] deasserts 1 half clock (HCLK) before end of access ... 1111 = $\overline{EB}$[3:0] deasserts 15 half clocks (HCLK) before end of access Note: The $\overline{EB}$[3:0] assertion time (as affected by WEA, WEN, and WSC bits) must be programmed to be greater than or equal to one HCLK cycle.
CSA Bits 15–12	**Chip Select Assert**—Determines when chip select is asserted or devices that require additional address setup time. CSA affects only external writes, and is ignored on external reads. However, during $\overline{DTACK}$ rising edge triggered operations ($\overline{CS5}$ only), CSA affects both read and write cycles. CSA does not affect the cycle length. CSA is cleared by a hardware reset.	0000 = $\overline{CS}$ asserts 0 HCLK clocks after start of external access 0001 = $\overline{CS}$ asserts 1 HCLK clocks after start of external access ... 1111 = $\overline{CS}$ asserts 15 HCLK clocks after start of external access
EBC Bit 11	**Enable Byte Control**—Indicates the access types that assert the enable byte outputs ($\overline{EB}$ [3:0]). The EBC bits are set by default for all CS0 to CS5	0 = Both read and write accesses assert the $\overline{EB}$ [3:0] outputs, thus configuring the access as byte enables. 1 = Only write accesses assert the $\overline{EB}$ [3:0] outputs, thus configuring the access as byte write enables; the $\overline{EB}$ [3:0] outputs are configured as byte write enables for accesses to dual x16 or quad x8 memories.
DSZ Bits 10–8	**Data Port Size**—Defines the width of the external device's data port size. At hardware reset, the value of DSZ is cleared for $\overline{CS1}$– $\overline{CS5}$. For $\overline{CS0}$, DSZ is mapped based on the value of the BOOT[3:0] signals.	000 = 8-bit port, resides on D [31:24] pins 001 = 8-bit port, resides on D [23:16] pins 010 = 8-bit port, resides on D [15:8] pins 011 = 8-bit port, resides on D [7:0] pins 100 = 16-bit port, resides on D [31:16] pins 101 = 16-bit port, resides on D [15:0] pins 110 = 32-bit port 111 = Reserved
CSN Bits 7–4	**Chip Select Negate**—Determines when chip select is negated for devices that require additional address/data hold times. CSN affects only external writes, and is ignored on external reads. CSN has no affect when synchronous mode is enabled (SYNC=1). During $\overline{DTACK}$ rising edge triggered operations ($\overline{CS5}$ only), CSN affects both read and write cycles. CSN does not affect the cycle length. CSN is cleared by a hardware reset.	0000 = $\overline{CS}$ deasserts 0 HCLK clocks before end of external access 0001 = $\overline{CS}$ deasserts 1 HCLK clocks before end of external access ... 1111 = $\overline{CS}$ deasserts 15 HCLK clocks before end of external access

Table 20-6. Chip Select Control Registers Description (continued)

Name	Description	Settings
PSR Bit 3	**PSRAM Enable**—Enables the PSRAM mode that allows burst writes and refresh wait. If SYNC = 0, asynchronous mode is used. If SYNC = 1, synchronous mode is used. In PSRAM mode the $\overline{ECB}$ signal is continually sampled after wait count programmed in the BCD/DCT expires (EIM waits until the $\overline{ECB}$ negation to continue accesses). The WRAP bit during writes is automatically masked (according the first generation CellularRAM™ Specification, WRAP is supported only for read accesses) and wait state on read is automatically incremented. PSR also needs to be set when programming the PSRAM control register. PSR is cleared by a hardware reset.	0 = PSRAM mode is disabled 1 = PSRAM mode is enabled
CRE Bit 2	**Control Register Enable**—Indicates the CRE memory pin state while writing to $\overline{CS}$ address space, for PSRAM control register write. This bit will be driven on pin ADDR[25]. For CRE to drive ADDR[25], the PSR bit also needs to be set. CRE is cleared by a hardware reset.	0 = CRE pin (A[25]) low 1 = CRE pin (A[25]) high
WRAP Bit 1	**WRAP memory mode**—Indicates to the EIM that the memory is in wrap mode. The size of wrap should be set in PSZ bits. In case wrap boundaries in both the PSZ and AHB access do not match, the EIM issues the next address on address bus and generates $\overline{LBA}$ signal. WRAP is cleared by a hardware reset.	0 = memory is in linear addressing mode 1 = memory is in wrap mode, PSZ indicates memory burst length
CSEN Bit 0	**Chip Select Enable**—Controls the operation of the chip select pin. CSEN in the CS0 control register is set at reset to allow $\overline{CS0}$ to select from an external boot ROM or NOR Flash. CSEN is set by a hardware reset for $\overline{CS0}$. CSEN is cleared by a hardware reset for $\overline{CS1}$–$\overline{CS5}$.	0 = Chip select function is disabled; attempts to access an address mapped by this chip select results in an error respond on the AHB and no assertion of the chip select output. 1 = Chip select is enabled, and is asserted when presented with a valid AHB access.

Table 20-7. Chip Select Wait State and Burst Delay Encoding

WSC [5:0]	Number of Wait-States					
	WWS = 0		WWS = 1		WWS = 7	
	Read Access	Write Access	Read Access	Write Access	Read Access	Write Access
000000	1	1	1	1	1	7
000001	1	1	1	2	1	8
000010	2	2	2	3	2	9
000011	3	3	3	4	3	10
...						
110111	55	55	55	56	55	62
111000	56	56	56	57	56	63
111001	57	57	57	58	57	63
111010	58	58	58	59	58	63
111011	59	59	59	60	59	63
111100	60	60	60	61	60	63
111101	61	61	61	62	61	63
111110	62	62	62	63	62	63
111111	Reserved or External DTACK	Reserved or External DTACK	Reserved or External DTACK	Reserved or External DTACK	Reserved or External DTACK	Reserved or External DTACK

Note: WSC = 111111 is not available (reserved) for $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$ and $\overline{CS4}$, but it is available (the external rising edge triggered $\overline{DTACK}$) for $\overline{CS5}$.

20.6.3 EIM Configuration Register

The EIM Configuration Register contains the bit that controls the operation of the burst clock.

EIM_CNF — EIM Configuration Register — 0xDF001030

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														BCM	AGE	
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 20-8. EIM Configuration Register Description

Name	Description	Settings
Reserved Bits 31–3	Reserved—These bits are reserved and should read 0.	
BCM Bit 2	**Burst Clock Mode**—Selects the burst clock mode of operation. BCM is cleared by a hardware reset.	0 = The burst clock runs only when accessing a chip select range with the SYNC bit set; when the burst clock is not running it remains in a logic 0 state; when the burst clock is running it is configured by the BCD and BCS bits in the chip select control register 1 = The burst clock runs only during chip select accesses. The frequency of the burst clock is also affected by the setting of the BCD bits in the chip select control register.
AGE Bit 1	Acknowledge Glue Enable—Is used to enable / disable glue logic between external $\overline{\text{DTACK}}$ and internal control logic. This glue logic is used to synchronize with the rising edge of the external $\overline{\text{DTACK}}$. Hence, when using $\overline{\text{DTACK}}$ for rising edge triggered operations, the AGE bit should be set to enable the glue logic. For $\overline{\text{DTACK}}$ level sensitive operations, the AGE bit should be cleared. AGE is cleared by a hardware reset.	0 = Disable glue logic (for $\overline{\text{DTACK}}$ level sensitive operations) 1 = Enable glue logic (for $\overline{\text{DTACK}}$ rising edge triggered operations)
Reserved Bit 0	Reserved—These bits are reserved and should read 0.	

Chapter 21
Bus Master Interface (BMI)

This chapter describes the unique interface requirement for the Bus Master Interface (BMI) module. The BMI module enables high speed connection between i.MX21 and the alternate bus master devices in the system.

The BMI provides support for the following functions:

- 8- or 16-bit data bus mode
- External bus master read or write to the CPU using memory access timing
- CPU reads or writes to the external bus slave using memory access timing
- ATI graphic chip burst read/write accesses timing
- High communication speed
- DMA support

21.1 BMI Block Diagram

Figure 21-1 shows the BMI block diagram.

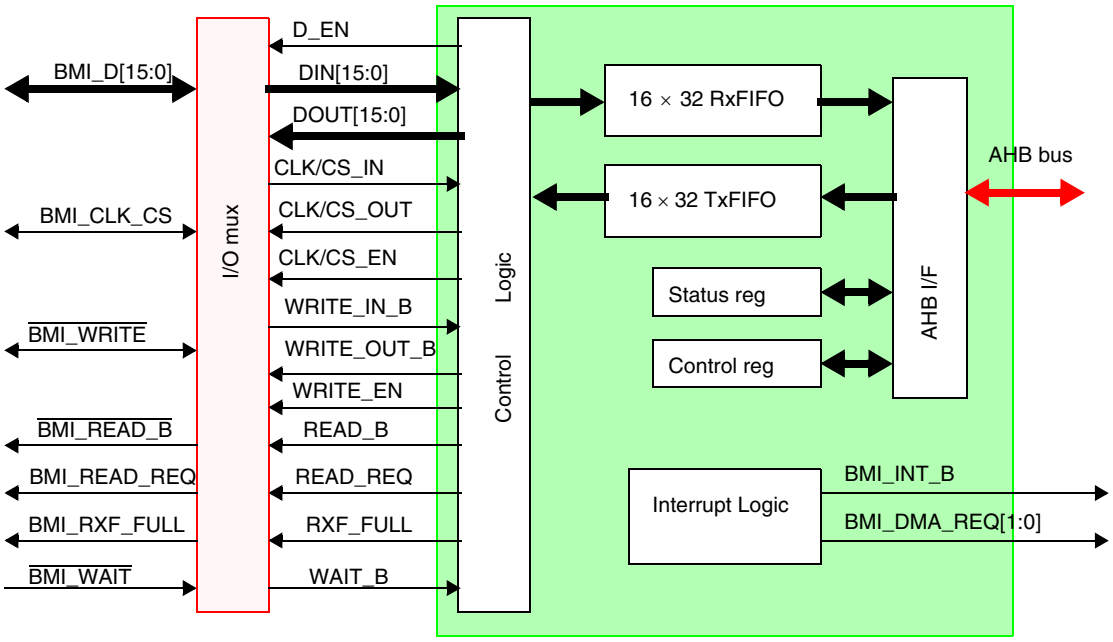


Figure 21-1. BMI Block Diagram

21.2 Signal Description

- BMI_D[15:0]—BMI data bus, bus width can be 8- to 16-bit selectable.
- BMI_READ_REQ—Inform external bus master that data is ready for a read operation. This signal is active when the data in the TXFIFO is larger or equal to the data transfer size of a single external BMI access.
- BMI_CLK/CS—When the BMI is connecting to an ATI graphic chip, this is MMDCLK, which is used to clock in/out data to/from the data FIFO. When the BMI is connecting to external bus master, this becomes a chip select signal. This clock frequency must be less than hclk/3. Please refer to the respective timing diagrams and tables to get the reasonable frequency.
- BMI_WRITE—Active low bidirectional signal to control data direction. When BMI is a slave, this is input to BMI to determine read or write to BMI data FIFO. When BMI is a bus master, this is an output write enable signal to external slave device.
- BMI_READ—Active low output signal to enable external slave data output. this signal is not used and is driven high when BMI is a slave.
- BMI_RXF_FULL—Active high output signal to reflects if RxFIFO reaches water mark value.
- BMI_WAIT—Active low signal to wait for data ready (read cycle) or accepted (write_cycle). This signal is only used when the BMI is in master mode and the WAIT bit is set. This signal is connected with the PA29 AOUT.
- AHB bus—Internal ARM AHB bus.
- BMI_INT_B—Active low interrupt signal to AIPI interrupt controller. This signal is asserted when there is an Interrupt event and is negated when there is no more interrupt event is pending.
- BMI_DMA_REQ[1]—Active low DMA request signal to DMA controller. This signal is asserted whenever the RxF_FULL status bit is set, and it is negated when RxF_FULL condition is no longer true.
- BMI_DMA_REQ[0]—Active low DMA request signal to DMA controller. This signal is asserted whenever the TxF_EMPTY status bit is set, and it is negated when TxF_EMPTY condition is no longer true.

21.3 Pin Configuration for BMI

Table 21-1 lists the pin used for the BMI module. These pins are multiplexed with other functions on the device and must be configured for BMI operation.

Table 21-1. BMI Pin Configuration

Pin	Setting	Configuration Procedure
BMI_D[15:0]	Alternate Function of GPIO PortA[21:6]	1. Clear bit 21–6of the Port A GPIO In Use Register (GIUS_A) 2. Set bit 21–6 of the Port A General Purpose Register (GPR_A)
BMI_CLK_CS	Alternate Function of GPIO PortA[5]	1. Clear bit 5 of the Port A GPIO In Use Register (GIUS_A) 2. Set bit 5 of the Port A General Purpose Register (GPR_A)
$\overline{\text{BMI\_WRITE}}$	Alternate Function of GPIO PortA[23]	1. Clear bit 23 of the Port A GPIO In Use Register (GIUS_A) 2. Set bit 23 of the Port A General Purpose Register (GPR_A)

Table 21-1. BMI Pin Configuration (continued)

Pin	Setting	Configuration Procedure
$\overline{\text{BMI\_READ}}$	Alternate Function of GPIO PortA[30]	1. Clear bit 30of the Port A GPIO In Use Register (GIUS_A) 2. Set bit 30 of the Port A General Purpose Register (GPR_A)
BMI_READ_REQ	Alternate Function of GPIO PortA[22]	1. Clear bit 22 of the Port A GPIO In Use Register (GIUS_A) 2. Set bit 22 of the Port A General Purpose Register (GPR_A)
BMI_RXF_FULL	Alternate Function of GPIO PortA[29]	1. Clear bit 29 of the Port A GPIO In Use Register (GIUS_A) 2. Set bit 29 of the Port A General Purpose Register (GPR_A)
$\overline{\text{BMI\_WAIT}}$	AOUT of GPIO PortA[29]	1. Set bit 29 of the Port A GPIO In Use Register (GIUS_A) 2. Clear bit27-26 of the PortA GPIO Input Configuration Register A2(ICONFA2)

21.4 Programming Model

The BMI module includes five user-accessible 32-bit registers. Only the BMI TxFIFO register (BMITXD) can a byte or halfword write, all other access is always word size. Table 21-2 summarizes these registers and their addresses.

Table 21-2. System Control Module Register Summary

Description	Name	Address
BMI Control Register 1	BMICTLR1	0xA000_0000
BMI Control Register 2	BMICTLR2	0xA000_0004
BMI Status Register	BMISTR	0xA000_0008
BMI RxFIFO	BMIRXD	0xA000_000C
BMI TxFIFO	BMITXD	0xA000_0010

21.4.1 BMI Control Register 1

This 32-bit register controls how the BMI operates. The bit assignments for the register are shown in the following register display. The settings for the bits in the register are listed in Table 21-3.

BMICTLR1 — BMI Control Register 1 — Addr 0xA0000000

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			WAIT	WS			DIV		TxF_Water_Mark				RxF_Water_Mark			
TYPE	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
	0x000F															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA_LATCH	MMD_CLKOUT	READ	MASTER_SEL	RxF_OV_INT_EN	BRDY_INT_EN	WRDY_INT_EN	RxFF_INT_EN	TxFE_INT_EN	MMD_MODE_SEL	CLK_CS_POL	READ_REQ_POL	16BIT_SEL	RxFIFO_CLR	TxFIFO_CLR	BMI_EN
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 21-3. BMI Control Register 1 Description

Name	Description	Settings
Reserved Bits 31–30	Reserved—These data bits are reserved and should always read 0.	
WAIT Bit 29	**WAIT**—This bit is only used in master mode. When this bit is cleared, the CS cycle is terminated by Wait State (WS) control bits. When this bit is set, the CS cycle is terminated upon sampling a logic high WAIT in signal	0 = ignore $\overline{\text{BMI\_WAIT}}$ signal 1 = detect BMI_WAIT signal
WS Bits 28–26	**Wait State**—In master mode the number of wait states can be inserted to the BMI read/write cycle. The WS is inserted to both chip low for longer access time and chip select high for longer idle time between read/write cycles	000 = 0 WS 001 = 1WS 010 = 2WS … 111 = 7 WS
DIV Bits 25–24	**Int_Clk Divider**—In master mode HCLK is divide by DIV to generate Int_Clk. This clock is used to clock all output signals for the read/write cycles.When MMD_MODE_SEL bit and MMD_CLKOUT bit both set, HCLK is divided by DIV to generate BMI_CLK_CS.	00 = divide by 2 01 = divide by 4 10 = divide by 8 11 = divide by 16
TxF_Water_Mark Bits 23–20	**TxFIFO Interrupt and DMA Request Water Mark**—Sets the interrupt and DMA request trigger level. When the number of empty slots is equal or greater than the water mark value, TxF_EMPTY status bit is set and DMA request signal is driving to active level. BMI interrupt is generated if TxFE_INT_EN is set.	0000 = Tx_EMPTY status bit is always set 0001 = At least 1 empty slot 0010 = At least 2 empty slots ... 1111 = At least 15 empty slots

Table 21-3. BMI Control Register 1 Description (continued)

Name	Description	Settings
RxF_Water_Mark Bits 19–16	**RxFIFO Interrupt and DMA Request Water Mark**—Sets the interrupt and DMA request trigger level. When the number of data words (32-bit) in the RxFIFO is equal or greater than the water mark value, RxF_FULL status bit is set and DMA request signal is driving to active level. BMI interrupt is generated if RxFF_INT_EN is set.	0000 = RxFIFO is full 0001 = At least 1 data word ready 0010 = At least 2 data words ready ... 1111 = At least 15 data words ready
DATA_LATCH Bit 15	**BMI Latch Data Edge**—This bit is only useful when the MMD_CLKOUT bit set to 1. When ATI MMD devices write to BMI and BMI drives clock, the BMI can latch data at both edge according to this bits. If the BMI latch data on the next rising edge, the last data is latched by internal clock.	0 = BMI latch data on the falling edge 1 = BMI latch data on the next rising edge
MMD_CLKOUT Bit 14	**BMI Clock Direction**—This bit is only useful when the MMD_MODE_SEL bit set to 1. When this bit is cleared, both BMI_WRITE and BMI_CLK/CS are input signals to BMI driving by ATI MMD device. When this bit is set, the BMI_CLK/CS is output signals driven by BMI and the BMI_WRITE is input signals to BMI driving by ATI MMD device.	0 = MMD drives clock 1 = BMI drives clock
READ Bit 13	**Force a READ Cycle**—When BMI is configured as master mode (MASTER_SEL bit is set) or MMD_MODE_SEL bit and MMD_CLKOUT bit are both set, writing a “1” to this bit will enable BMI to generate a read cycle. The COUNT bits in the BMICTLR2 control the read count. This bit is clear automatically when the read cycle is completed. This bit is ignored in all other mode. User can read this bit to see if the read cycle is completed before issuing the next WRITE or READ command.	0 = No READ cycle or READ command is completed. 1= READ command is ongoing or pending.
MASTER_SEL Bit 12	**Master Mode Select**—Configure the BMI to master mode or slave mode. This bit is clear automatically to select slave mode when BMI is connecting to ATI graphic chip (MMD mode). This bit is also used to control the direction of BMI_WRITE signals. It is output in master mode and is input in slave mode.	0 = Slave mode 1 = Master mode
RxF_OV_INT_EN Bit 11	**RxFIFO Over Flow Interrupt Enable**—Enables data over flow interrupt.	0 = Disable 1 = Enable
BRDY_INT_EN Bit 10	**Data Byte Ready Interrupt Enable**—Enables data byte interrupt. Interrupt is generated when there is at least one data byte in the RxFIFO.	0 = Disable 1 = Enable
WRDY_INT_EN Bit 9	**Data Word Ready Interrupt Enable**—Enables data word (32-bit) interrupt. Interrupt is generated when there is at least one data word in the RxFIFO.	0 = Disable 1 = Enable
RxFF_INT_EN Bit 8	**RxFIFO Full Interrupt Enable**—Set this bit to enable RxFIFO full interrupt if RxFIFO reaches or greater than water mark value.	0 = Disable 1 = Enable
TxFE_INT_EN Bit 7	**TxFIFO Empty Interrupt Enable**—Set this bit to enable TxFIFO empty Interrupt when the number of empty slots in TxFIFO reaches or greater than water mark value.	0 = Disable 1 = Enable
MMD_MODE_SEL Bit 6	**ATI MMD Interface Timing Select**—Set this bit to select the ATI graphic chip interface timing mode.	0 = Memory interface timing mode 1 = ATI graphic chip interface timing mode

Table 21-3. BMI Control Register 1 Description (continued)

Name	Description	Settings
CLK_CS_POL Bit 5	**Clock/Chip Select Polarity**—Set this bit to invert the clock/chip select polarity.	0 = Not inverted 1 = Inverted
READ_REQ_POL Bit 4	**Read Request Output Polarity**—Set this bit to invert the READ_REQ signal output polarity.	0 = Not inverted 1 = Inverted
16BIT_SEL Bit 3	**16-Bit Data Bus Select**—Set this bit to select 16-bit data bus width.	0 = 8-bit 1 = 16-bit
RxFIFO_CLR Bit 2	**Clear RxFIFO**—Write a 1 to this bit to clear RxFIFO. This bit is always read 0.	0 = No action 1 = Clear RxFIFO (This bit is clear automatically after the write cycle)
TxFIFO_CLR Bit 1	**Clear TxFIFO**—Write a 1 to this bit to clear TxFIFO. This bit is always read 0. When the TxFIFO is written to some data to be transfer, BMI asserts READ_REQ to signal external bus master to read data from BMI. If there is no read operation performed, TxFIFO is never emptied. A clear TxFIFO will reset the TxFIFO to Empty position.	0 = No action 1 = Clear TxFIFO (This bit is clear automatically after the write cycle)
BMI_EN Bit 0	**BMI Module Enable**—Set this bit to enable BMI module. The status register will go to reset value when this bit is cleared.	0 = Disable 1 = Enable

21.4.2 BMI Control Register 2

The BMI Control Register(BMICTLR2) controls how the BMI operates. The bit assignments for the register are shown in the following register display. The settings for the bits in the register are listed in Table 21-4.

BMICTLR2 — BMI Control Register 2 — Addr 0xA0000004

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											COUNT					
TYPE	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 21-4. BMI Control Register 2 Description

Name	Description	Settings
Reserved Bits 31–6	Reserved—These bits are reserved and should read 0.	
COUNT Bits 5–0	**Read Cycle Count**—Reflects number of data the BMI read from the external bus. It is valid when the BMI is configured as master or MMD_MODE_SEL and MMD_CLKOUT bits are both set. When a read cycle is issued, this count bits control how much data the BMI will read from the external bus.	000000 = read one data 000001 = read two data 000010 = read three data 000011 = read four data ..... 111111 = read 64 data

21.4.3 BMI Status Register

The BMI Status Register (BMISR) reflects the BMI status. See Table 21-5 for detailed description of status bit.

BMISR — BMI Status Register — Addr 0xA0000008

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
											0x0000					

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									TA	RxF_OV	BRDY	WRDY	RxF_FULL	TxF_EMPTY	BCNT	
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
											0x0004					

Table 21-5. BMI Status Register Description

Name	Description	Settings
Reserved Bits 31–8	Reserved—These bits are reserved and should read 0.	
TA Bit 7	**Transfer Activity**—Reflects if the BMI doing READ or WRITE operation. This bit is used for master mode only.	0 = No READ or WRITE activity 1 = In READ or WRITE activity
RxF_OV Bit 6	**RxFIFO Over Flow**—Reflects if receiving data has ever been over flow. When there is over flow, the new data is discarded.	0 = No data over flow 1 = At least one data byte lost
BRDY Bit 5	**Data Byte Ready**—Reflects if there is any data byte ready in the RxFIFO. This bit is used when WRDY is clear and there could be 1,2 or 3 bytes in the RxFIFO.	0 = No data byte ready 1 = At least one data byte is ready
WRDY Bit 4	**Data Word Ready**—Reflects if there is any complete data word ready in the RxFIFO.	0 = No data word ready 1 = At least one data word is ready
RxF_FULL Bit 3	**RxFIFO Full**—Reflects if RxFIFO reaches the water mark value	0 = Not full 1 = RxFIFO at or over water mark value
TxF_EMPTY Bit 2	**TxFIFO Empty**—Reflects if TxFIFO reaches the water mark value	0 = Not empty 1 = TxFIFO has empty slot equal or greater than water mark value
BCNT Bits1–0	**Byte Count**—Reflects number of data bytes the in the last data word read. BCNT is updated after each 32-bit read from the RxFIFO. When WRDY is clear and BRDY is set, CPU or DMA performs a final 32-bit data read from the RxFIFO, then read this 2 status bits to determine how many valid data bytes are in the last 32-bit read. If user sets the data packet to be always word size, BRDY and BCNT are not used.	00 = No valid data in the last read 01 = 1 valid data byte, D[7:0]] 10 = 2 valid data bytes D[15:0] 11 = 3 valid data bytes D[23:0]

21.4.4 BMI RxFIFO Register

The BMI RxFIFO Register (BMIRXD) contains received data from external bus master. CPU or DMA data read from this register is always word size regardless number of bytes left in the FIFO.

BMIRXD	BMI RxFIFO Register												Addr 0xA000000C			
BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	D[31:16]															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D[15:0]															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 21-6. BMI RxFIFO Register Description

Name	Description
D[31:16] Bits 31–16	**Data**—Upper half data word.
D[15:0] Bits 15–0	**Data**—Lower half data word.

21.4.5 BMI TxFIFO Register

The BMI TxFIFO Register (BMTXD) is to store data to be sent to external bus devices upon BMI read access. The DMA or CPU data write to this register can be a byte, haft word, or full word size, but all these write operation must align to the 0xA0000010 address.

BMITXD — BMI TxFIFO Register — Addr 0xA0000010

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	D[31:16]															
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D[15:0]															
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 21-7. BMI TxFIFO Register Description

Name	Description
D[31:16] Bits 31–16	**Data**—Upper half data word.
D[15:0] Bits 15–0	**Data**—Lower half data word.

Part 5
InterChip Connectivity

Chapter 22
I²C Module

I²C is a two-wire, bidirectional serial bus that provides a simple, efficient method of data exchange, minimizing the interconnection between devices. This bus is suitable for applications requiring occasional communications over a short distance between many devices. The flexible I²C allows additional devices to be connected to the bus for expansion and system development.

The I²C operates up to 400 kbps but it depends on the pad loading and timing (for pad requirement details please refer to Philips I²C Bus Specification, Version 2.1). The I²C system is a true multiple-master bus including arbitration and collision detection that prevents data corruption if multiple devices attempt to control the bus simultaneously. This feature supports complex applications with multiprocessor control and can be used for rapid testing and alignment of end products through external connections to an assembly-line computer.

The I²C module has the following key features:

- Compatibility with I²C bus standard
- Multiple-master operation
- Software-programmable for one of 64 different serial clock frequencies
- Software-selectable acknowledge bit
- Interrupt-driven, byte-by-byte data transfer
- Arbitration-lost interrupt with automatic mode switching from master to slave
- Calling address identification interrupt
- Start and stop signal generation/detection
- Repeated START signal generation
- Acknowledge bit generation/detection
- Bus-busy detection

Figure 22-1 is the block diagram of the I²C module.

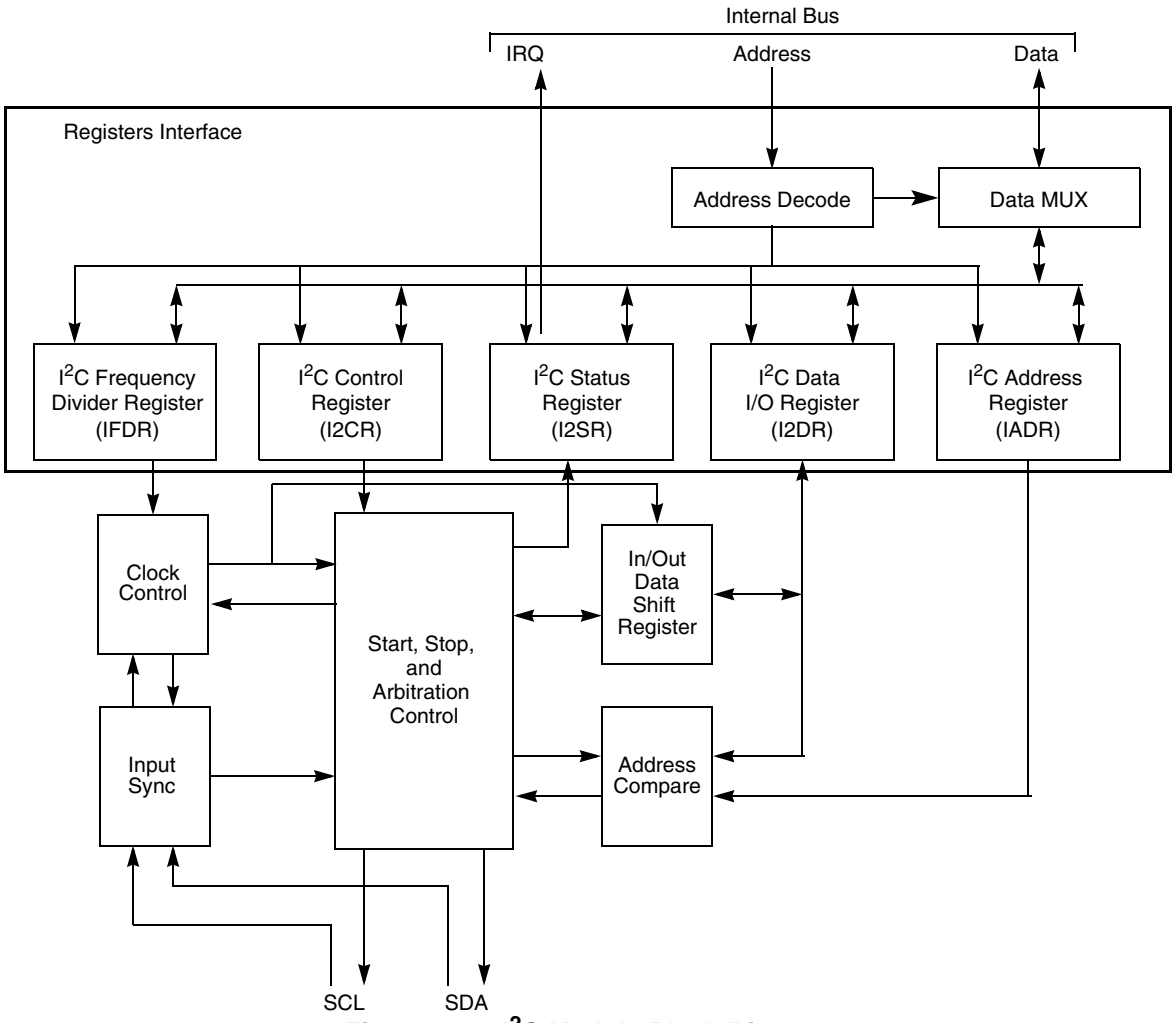


Figure 22-1. I²C Module Block Diagram

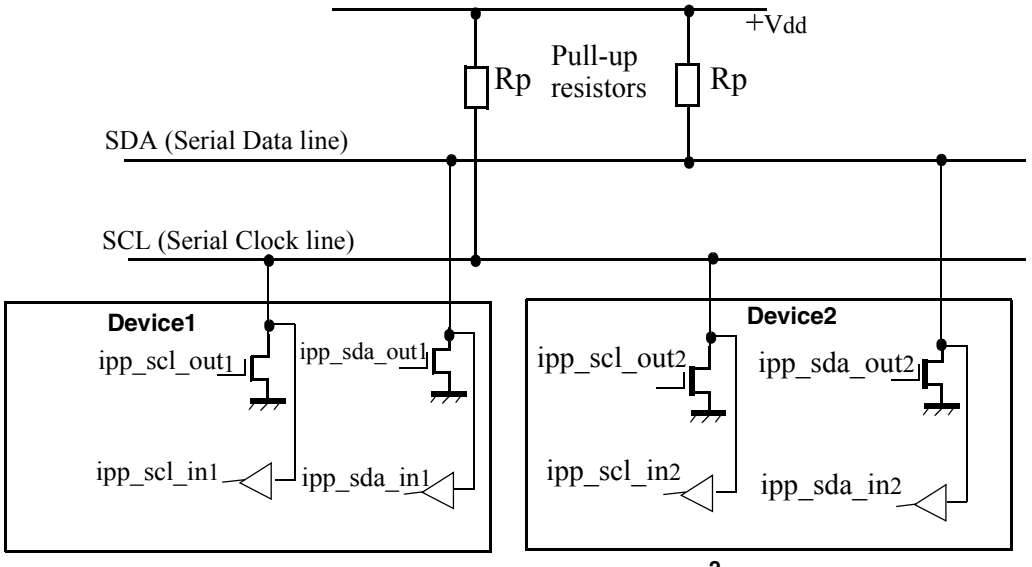


Figure 22-2. Connection of Devices to I²C Bus

22.1 I²C System Configuration

The I²C module uses a serial data line (SDA) and a serial clock line (SCL) for data transfer. For I²C compliance, all devices connected to these two signals must have open drain or open collector outputs. (There is no such requirement for inputs.) The logic AND function is exercised on both lines with external pull-up resistors.

Out of reset, the I²C default is as slave receiver. Thus, when not programmed to be a master or responding to a slave transmit address, the I²C module should return to the default slave receiver state. See Section 22.6.1, "Initialization Sequence," for exceptions.

NOTE

The I²C module is designed to be compatible with the Philips I²C bus protocol. For information on system configuration, protocol, and restrictions, see *The I²C Bus Specification, Version 2.1*.

22.2 I²C Protocol

The I²C communication protocol consists of six components: START, Data Source/Recipient, Data Direction, Slave Acknowledge, Data, Data Acknowledge and STOP. These are shown in Figure 22-3 and described in the text following the figure.

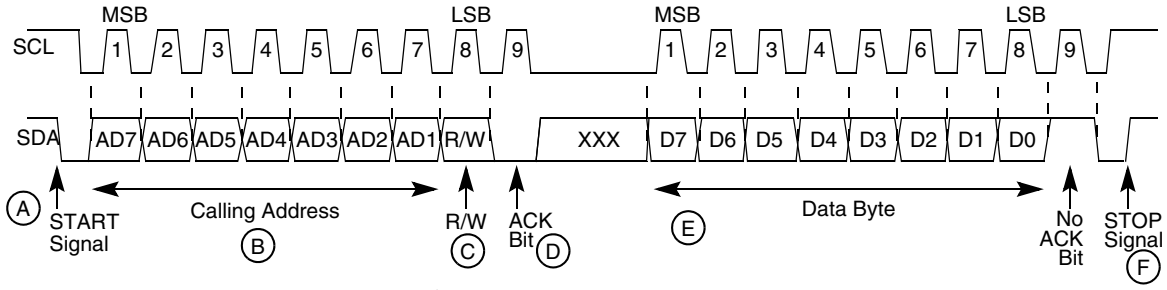


Figure 22-3. I²C Standard Communication Protocol

1. START signal—When no other device is bus master (both SCL and SDA lines are at logic high), a device can initiate communication by sending a START signal (see A in Figure 22-3). A START signal is defined as a high-to-low transition of SDA while SCL is high. This signal denotes the beginning of a data transfer (each data transfer can be several bytes long) and awakens all slaves.
2. Slave address transmission—The master sends the slave address in the first byte after the START signal (B). After the seven-bit calling address, it sends the R/W bit (C), which tells the slave data transfer direction.

   Each slave must have a unique address. An I²C master must not transmit an address that is the same as its slave address; it cannot be master and slave at the same time.

   The slave whose address matches that sent by the master pulls SDA low at the ninth clock (D) to return an acknowledge bit.
3. Data transfer—When successful slave addressing is achieved, the data transfer can proceed (E) on a byte-by-byte basis in the direction specified by the R/W bit sent by the calling master.

Data can be changed only while SCL is low and must be held stable while SCL is high, as Figure 22-3 shows. SCL is pulsed once for each data bit, with the MSB being sent first. The receiving device must acknowledge each byte by pulling SDA low at the ninth clock; therefore, a data byte transfer takes nine clock pulses.

If it does not acknowledge the master, the slave receiver must leave SDA high. The master can then generate a STOP signal to abort the data transfer or generate a START signal (repeated start, shown in Figure 22-4) to start a new calling sequence.

If the master receiver does not acknowledge the slave transmitter after a byte transmission, it means end-of-data to the slave. The slave releases SDA for the master to generate a STOP or START signal.

4. STOP signal The master can terminate communication by generating a STOP signal to free the bus. A STOP signal is defined as a low-to-high transition of SDA while SCL is at logical high (F). Note that a master can generate a STOP even if the slave has made an acknowledgment, at which point the slave must release the bus.
5. Instead of signalling a STOP, the master can repeat the START signal, followed by a calling command, (A in Figure 22-4). A repeated START occurs when a START signal is generated without first generating a STOP signal to end the communication. The master uses a repeated START to communicate with another slave or with the same slave in a different mode (transmit/receive mode) without releasing the bus.

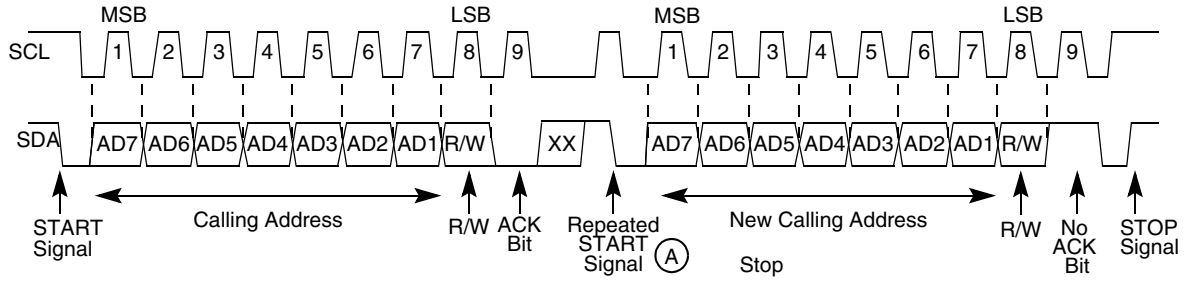


Figure 22-4. Repeated START

22.2.1 Arbitration Procedure

If multiple devices simultaneously request the bus, the bus clock is determined by a synchronization procedure in which the low period equals the longest clock-low period among the devices and the high period equals the shortest. A data arbitration procedure determines the relative priority of competing devices. A device loses arbitration if it sends logic high while another sends logic low; it immediately switches to slave-receive mode and stops driving SDA. In this case, the transition from master to slave mode does not generate a STOP condition. Meanwhile, hardware sets I2SR[IAL] to indicate loss of arbitration.

22.2.2 Clock Synchronization

Because wire-AND logic is used, a high-to-low transition on SCL affects devices connected to the bus. Devices start counting their low period when the master drives SCL low. When a device clock goes low, it holds SCL low until the clock high state is reached. However, the low-to-high change in this device clock

may not change the state of SCL if another device clock is still in its low period. Therefore, the device with the longest low period holds the synchronized clock SCL low. Devices with shorter low periods enter a high wait state during this time (See Figure 22-5). When all devices involved have counted off their low period, the synchronized clock SCL is released and pulled high. There is then no difference between device clocks and the state of SCL, so all of the devices start counting their high periods. The first device to complete its high period pulls SCL low again.

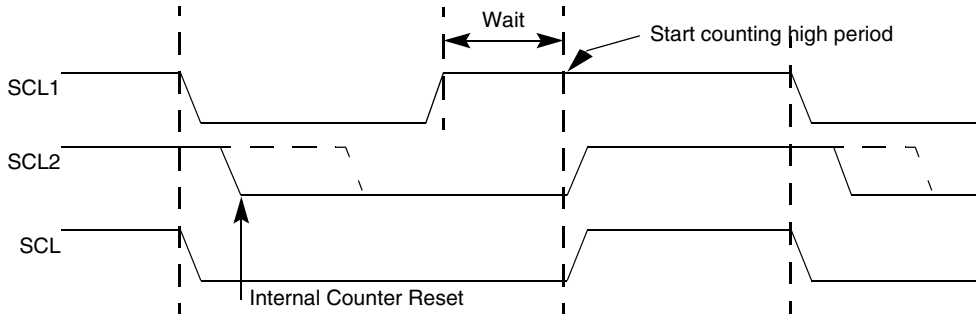


Figure 22-5. Synchronized Clock SCL

22.2.3 Handshaking

The clock synchronization mechanism can be used as a handshake in data transfers. Slave devices can hold SCL low after completing one byte transfer (9 bits). In such a case, the clock mechanism halts the bus clock and forces the master clock into wait states until the slave releases SCL.

22.2.4 Clock Stretching

Slaves can use the clock synchronization mechanism to slow down the transfer bit rate. After the master has driven SCL low, the slave can drive SCL low for the required period and then release it. If the slave SCL low period is longer than the master SCL low period, the resulting SCL bus signal low period is stretched.

22.3 Pin Configuration for I²C

Two pins are required for I²C.

- I2C_CLK Bidirectional clock Pin
- I2C_DATA Bi-directional Data Pin

22.4 IP Bus Accesses

I²C module expects all reads and writes to be aligned to 32-bit boundary.

Only lower 1 byte [7:0] is written for write accesses. If write access is for more than lower 1 byte—that is, 16-bit or 32-bit write access, upper bytes [31:24], [23:16], [15:8] will be simply ignored and writes happens only for the lower one byte [7:0]. Module takes only [7:0] bits of write data bus as input.

For reads I²C drives 16-bit data bus as output. For both 8-bit and 16-bit access, it returns lower 1 byte [7:0] out of respective register and 2nd byte [15:8] is always returned as 0. For a 32-bit read access module drives lower 16-bits as above and upper bits are not driven from the module.

22.4.1 Generation of Transfer Error on IP Bus

On getting an IP access to an address which is not implemented transfer error is signalled—that is, ips_xfr_err is generated. Input pin *resp_sel* provides configuration capability for the generation of this error. When resp_sel is low only then above error is signalled else no error is signalled.

22.5 Programming Model

Table 22-1 lists the configuration registers used in the I²C interface.

0xBA is the base address of I²C module. For i.MX21, 0xBA=0x10012000.

Table 22-1. I²C Interface Register Summary

Address Offset	[31:24]	[23:16]	[15:8]	[7:0]
0x000	Reserved			I²C address register (IADR)
0x004	Reserved			I²C frequency divider register (IFDR)
0x008	Reserved			I²C control register (I2CR)
0x00C	Reserved			I²C status register (I2SR)
0x010	Reserved			I²C data I/O register (I2DR)

22.5.1 I²C Address Register (IADR)

The IADR holds the address the I²C responds to when addressed as a slave. Note that it is not the address sent on the bus during the address transfer. The register does not get reset by software reset. Table 22-2 describes IADR fields.

IADR — I²C Address Register — 0xBA+0x000

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									ADR							
TYPE	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 22-2. I²C Address Register Description

Name	Description	Settings
Reserved Bits 15–8	Reserved—These bits are reserved and should read 0.	
ADR Bits 7–1	**Slave Address**—Contains the specific slave address to be used by the I²C module. Slave mode is the default I²C mode for an address match on the bus.	I²C module slave address
Reserved Bit 0	Reserved—This bit is reserved and should read 0.	

22.5.2 I²C Frequency Divider Register (IFDR)

The IFDR provides a programmable prescaler to configure the clock for bit-rate selection. The register does not get reset by software reset. Table 22-3 describes IFDR[IC].

IFDR — I²C Frequency Divider Register — $BA+0x004

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											IC					
TYPE	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 22-3. IFDR Register Description

Name	Description	Settings
Reserved Bits 15–6	Reserved—These bits are reserved and should read 0.	
IC Bits 5–0	**I²C Clock Rate**—Pre-scales the clock for bit-rate selection. Due to potentially slow SCL and SDA rise and fall times, bus signals are sampled at the prescaler frequency. The serial bit clock frequency is equal to ipg_clk divided by the divider shown below. Note that IC can be changed anywhere in a program. **Note:** Software should set the desired frequency divider	See note.

IC	Divider	IC	Divider	IC	Divider	IC	Divider
0x00	30	0x10	288	0x20	22	0x30	160
0x01	32	0x11	320	0x21	24	0x31	192
0x02	36	0x12	384	0x22	26	0x32	224
0x03	42	0x13	480	0x23	28	0x33	256
0x04	48	0x14	576	0x24	32	0x34	320
0x05	52	0x15	640	0x25	36	0x35	384
0x06	60	0x16	768	0x26	40	0x36	448
0x07	72	0x17	960	0x27	44	0x37	512
0x08	80	0x18	1152	0x28	48	0x38	640
0x09	88	0x19	1280	0x29	56	0x39	768
0x0A	104	0x1A	1536	0x2A	64	0x3A	896
0x0B	128	0x1B	1920	0x2B	72	0x3B	1024
0x0C	144	0x1C	2304	0x2C	80	0x3C	1280
0x0D	160	0x1D	2560	0x2D	96	0x3D	1536
0x0E	192	0x1E	3072	0x2E	112	0x3E	1792
0x0F	240	0x1F	3840	0x2F	128	0x3F	2048

22.5.3 I²C Control Register (I2CR)

The I2CR is used to enable the I²C module and the I²C interrupt. It also contains bits that govern operation as a slave or a master. Table 22-4 describes I2CR control register fields.

I2CR — I²C Control Register — $BA+0x008

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									IEN	IIEN	MSTA	MTX	TXAK	RSTA		
TYPE	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 22-4. I²CR Register Description

Name	Description	Settings
Reserved Bits 15–8	Reserved—These bits are reserved and should read 0.	
IEN Bit 7	**I²C Enable**—Also controls the software reset of the entire I²C module. Resetting the bit generates internal reset to the module. If the module is enabled in the middle of a byte transfer, slave mode ignores the current bus transfer and starts operating when the next start condition is detected. Master mode is not aware that the bus is busy; so initiating a start cycle may corrupt the current bus cycle, ultimately causing either the current master or the I²C module to lose arbitration, after which bus operation returns to normal. **Note:** IEN bit is not used for gating main clock to the module in i.MX21	0 = The module is disabled, but registers can still be accessed. 1 = The I²C module is enabled. This bit must be set before any other I2CR bits have any effect.
IIEN Bit 6	**I²C Interrupt Enable**—This bit enables or disables I²C module interrupts	0 = I²C module interrupts are disabled, but currently pending interrupt condition are not cleared. 1 = I²C module interrupts are enabled. An I²C interrupt occurs if I2SR[IIF] is also set. Interrupt remains asserted as long as IIF[I2SR] and IIEN remains set together.
MSTA Bit 5	**Master/Slave Mode Select**—If the master loses arbitration, MSTA is cleared without generating a STOP signal. **Note:** Module clock should be on for writing to MSTA bit.	0 = Slave mode. Changing MSTA from 1 to 0 generates a STOP and selects slave mode. 1 = Master mode. Changing MSTA from 0 to 1 signals a START on the bus and selects master mode.
MTX Bit 4	**Transmit/Receive Mode Select**—Selects the direction of master and slave transfers.	0 = Receive 1 = Transmit. When a slave is addressed, software should set MTX according to I2SR[SRW]. In master mode, MTX should be set according to the type of transfer required. Therefore, for address cycles, MTX is always 1.
TXAK Bit 3	**Transmit Acknowledge Enable**—Specifies the value driven onto SDA during acknowledge cycles for both master and slave receivers. **Note:** Writing TXAK applies only when the I²C bus is a receiver.	0 = An acknowledge signal is sent to the bus at the ninth clock bit after receiving one byte of data. 1 = No acknowledge signal response is sent (that is, acknowledge bit = 1).

Table 22-4. I²CR Register Description (continued)

Name	Description	Settings
RSTA Bit 2	**Repeat Start**—Always read as 0. Attempting a repeat start without bus mastership causes loss of arbitration.	0 = No repeat start 1 = Generates a repeated START condition.
Reserved Bits 1–0	Reserved—These bits are reserved and should read 0.	

22.5.4 I²C Status Register (I2SR)

This I2SR contains bits that indicate transaction direction and status. Table 22-5 describes I2SR fields.

I2SR — I²C Status Register — $BA+0x00C

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									ICF	IAAS	IBB	IAL		SRW	IIF	RXAK
TYPE	r	r	r	r	r	r	r	r	r	r	r	rw	r	r	rw	r
RESET	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1

Table 22-5. I²SR Register Description

Name	Description	Settings
Reserved Bits 15–8	Reserved—These bits are reserved and should read 0.	
ICF Bit 7	**Data Transferring**—While one byte of data is transferred, ICF is cleared.	0 = Transfer in progress 1 = Transfer complete. Set by the falling edge of the ninth clock of a byte transfer.
IAAS Bit 6	**I²C Addressed as Slave**—The CPU is interrupted if I2CR[IIEN] is set. Next, the CPU must check SRW and set its TX/RX mode accordingly. Writing to I2CR clears this bit.	0 = Not addressed. 1 = Addressed as a slave. Set when its own address (IADR) matches the calling address.
IBB Bit 5	**I²C Bus Busy**—Indicates the status of the bus.	0 = Bus is idle. If a STOP signal is detected, IBB is cleared. 1 = Bus is busy. When START is detected, IBB is set.
IAL Bit 4	**Arbitration Lost**—	Set by hardware in the following circumstances: (IAL must be cleared by software by writing zero to it.) • SDA sampled low when the master drives high during an address or data-transmit cycle. • SDA sampled low when the master drives high during the acknowledge bit of a data-receive cycle. • For above 2 cases bit is set at the falling edge of 9th SCL clock during ACK cycle. • A start cycle is attempted when the bus is busy. • A repeated start cycle is requested in slave mode. • A stop condition is detected when the master did not request it.
Reserved Bit 3	Reserved—This bit is reserved and should read 0.	

Table 22-5. I²SR Register Description (continued)

Name	Description	Settings
SRW Bit 2	**Slave Read/Write**—When IAAS is set, SRW indicates the value of the R/W command bit of the calling address sent from the master. SRW is valid only when a complete transfer has occurred, no other transfers have been initiated, and the I²C module is a slave and has an address match.	0 = Slave receive, master writing to slave. 1 = Slave transmit, master reading from slave.
IIF Bit 1	**I²C Interrupt**— This bit indicates that one of the I²C interrupts is pending.	Must be cleared by software by writing a zero to it in the interrupt routine. 0 = No I²C interrupt pending 1 = An interrupt is pending, which causes a processor interrupt request (if IIEN = 1). Set when one of the following occurs: Complete one byte transfer (set at the falling edge of the ninth clock) Reception of a calling address that matches its own specific address in slave-receive mode Arbitration lost
RXAK Bit 0	**Received Acknowledge**—The value of SDA during the acknowledge bit of a bus cycle.	0 = An acknowledge signal was received after the completion of 8-bit data transmission on the bus 1 = No acknowledge signal was detected at the ninth clock.

22.5.5 I²C Data I/O Register (I2DR)

In master-receive mode, reading the I2DR allows a read to occur and initiates next byte data receiving. In slave mode, the same function is available after it is addressed.

I2DR — I²C Data Register — Addr $BA+0x010

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									D							
TYPE	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 22-6. I²C Address Register Description

Name	Description	Settings
Reserved Bits 15–8	Reserved—These bits are reserved and should read 0.	
D Bits 7–0	**Data**—Holds last data byte received or next data byte to be transferred	Write the next data byte to be transmitted

NOTE

Core written value in I2DR can not be read back by Core, only data written by I²C bus side can be read.

22.6 I²C Programming Examples

The following examples show programming for initialization, signalling START, post-transfer software response, signalling STOP, and generating a repeated START.

22.6.1 Initialization Sequence

Before the interface can transfer serial data, registers must be initialized, as follows:

1. Set IFDR[IC] to obtain SCL frequency from the system bus clock. See Section 22.5.2, "I²C Frequency Divider Register (IFDR)."
2. Update the IADR to define its slave address.
3. Set I2CR[IEN] to enable the I²C bus interface system.
4. Modify the I2CR to select master/slave mode, transmit/receive mode, and interrupt-enable or not.

NOTE

If IBSR[IBB] is set when the I²C bus module is enabled, execute the following code sequence before proceeding with normal initialization code. This issues a STOP command to the slave device, placing it in idle state as if it were just power-cycled on.

22.6.2 Generation of START

After completion of the initialization procedure, serial data can be transmitted by selecting the master transmitter mode. On a multiple-master bus system, IBSR[IBB] must be tested to determine whether the serial bus is free. If the bus is free (IBB = 0), the START signal and the first byte (the slave address) can be sent. The data written to the data register comprises the address of the desired slave and the LSB indicates the transfer direction.

The free time between a STOP and the next START condition is built into the hardware that generates the START cycle. Depending on the relative frequencies of the system clock and the SCL period, it may be necessary to wait until the I²C is busy after writing the calling address to the I2DR before proceeding with data writing to I2DR.

22.6.3 Post-Transfer Software Response

Sending or receiving a byte sets the I2SR[ICF], which indicates one byte communication is finished. I2SR[IIF] is also set. An interrupt is generated if the interrupt function is enabled during initialization by setting I2CR[IIEN]. Software must first clear IIF in the interrupt routine. ICF is cleared either by reading from I2DR in receive mode or by writing to I2DR in transmit mode.

Software can service the I²C I/O in the main program by monitoring IIF if the interrupt function is disabled. Polling should monitor IIF rather than ICF because that operation is different when arbitration is lost.

When an interrupt occurs at the end of the address cycle, the master is always in transmit mode; that is, the address is sent. If master receive mode is required (I2DR[R/W], I2CR[MTX] should be toggled.

During slave-mode address cycles (I2SR[IAAS] = 1), I2SR[SRW] is read to determine the direction of the next transfer. MTX is programmed accordingly. For slave-mode data cycles (IAAS = 0), SRW is invalid. MTX should be read to determine the current transfer direction.

22.6.4 Generation of STOP

A data transfer ends when the master signals a STOP, which can occur after all data is sent.

For a master receiver to terminate a data transfer, it must inform the slave transmitter by not acknowledging the last data byte. This is done by setting I2CR[TXAK] before reading the next-to-last byte. Before the last byte is read, a STOP signal must be generated.

22.6.5 Generation of Repeated START

After the data transfer, if the master still wants the bus, it can signal another START followed by another slave address without signalling a STOP.

22.6.6 Slave Mode

In the slave interrupt service routine, the module addressed as slave bit (IAAS) should be tested to check if a calling of its own address has just been received. If IAAS is set, software should set the transmit/receive mode select bit (I2CR[MTX]) according to the I2SR[SRW]. Writing to the I2CR clears the IAAS automatically. The only time IAAS is read as set is from the interrupt at the end of the address cycle where an address match occurred; interrupts resulting from subsequent data transfers will have IAAS cleared. A data transfer can now be initiated by writing information to I2DR for slave transmits, or read from I2DR in slave-receive mode. A dummy read of I2DR in slave/receive mode releases SCL, allowing the master to send data.

In the slave transmitter routine, I2SR[RXAK] must be tested before sending the next byte of data. Setting RXAK means an end-of-data signal from the master receiver, after which software must switch it from transmitter to receiver mode. Reading I2DR then releases SCL so that the master can generate a STOP signal.

22.6.7 Arbitration Lost

If several devices try to engage the bus at the same time, one becomes master. Hardware immediately switches devices that lose arbitration to slave receive mode. Data output to SDA stops, but SCL is still generated until the end of the byte during which arbitration is lost. An interrupt occurs at the falling edge of the ninth clock of this transfer with I2SR[IAL] = 1 and I2CR[MSTA] = 0.

If a device that is not a master tries to transmit or do a START, hardware inhibits the transmission, clears MSTA without signalling a STOP, generates an interrupt to the CPU, and sets IAL to indicate a failed attempt to engage the bus. When considering these cases, the slave service routine should first test IAL and software should clear it if it is set.

For Multi master mode, when the I²C module is enabled and when the bus is busy and as assertions begin, the IAL bit is set for all combinations of SDA and SCL except for SDA=1 and SCA=1 which is same as bus idle state.

22.6.8 Timing Section

This section shows the timing diagram and description table for devices on the I²C bus.

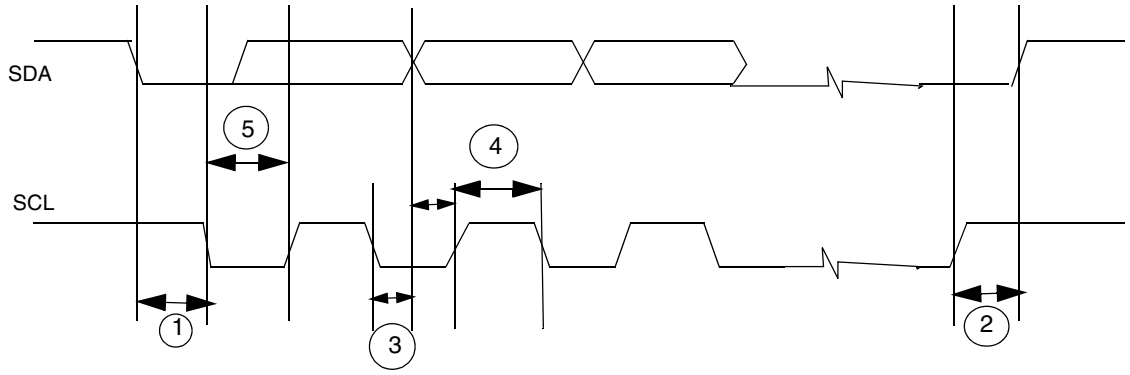
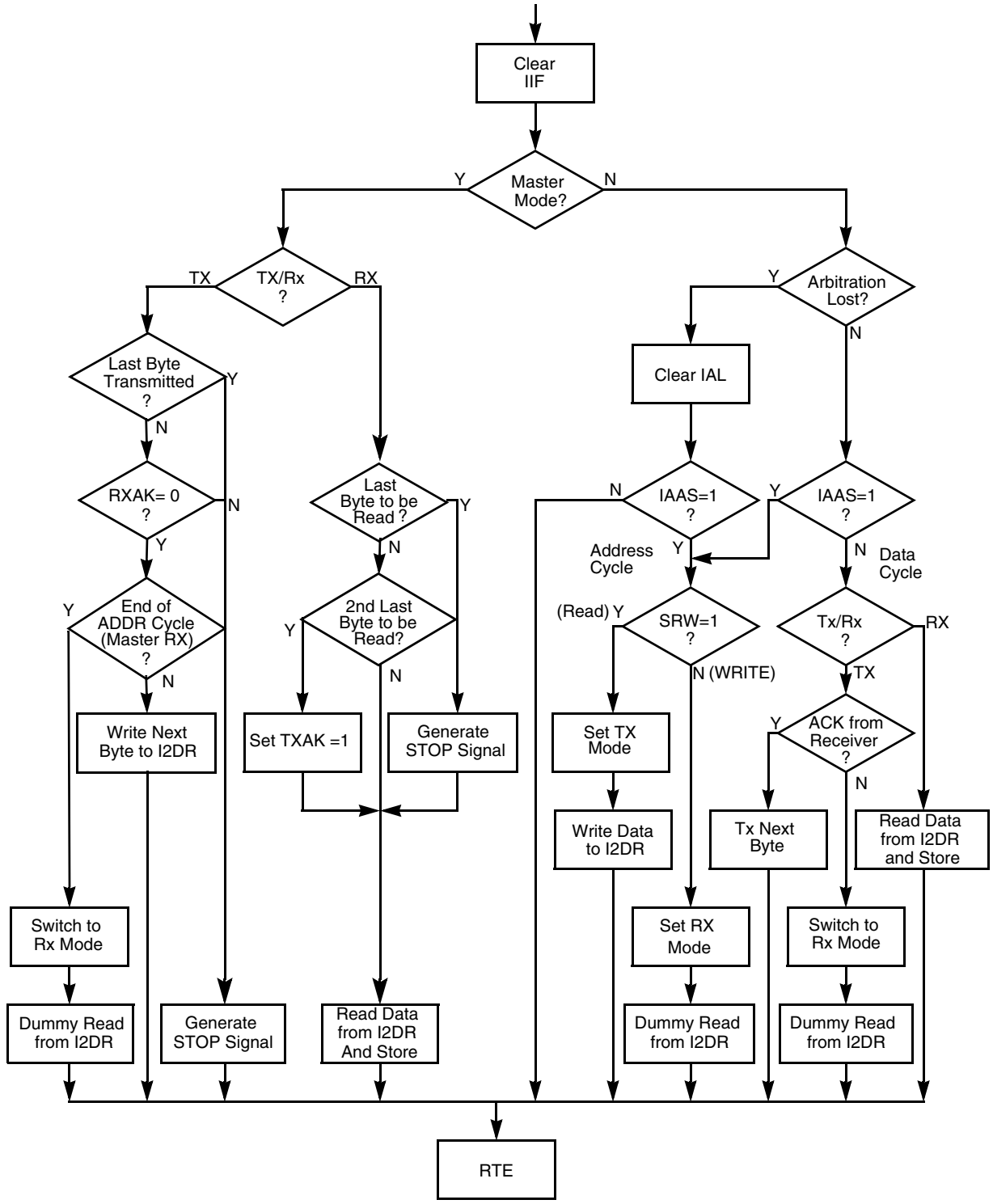


Figure 22-6. Timing Definition for Devices on the I²C Bus

Table 22-7. I²C Bus Device Timing Definitions[1]

Ref No.	Parameter	Maximum (w.r.t ipg_clk)	Minimum (w.r.t ipg_clk)
1	Hold time (repeated) START condition	–	4
2	Setup time for STOP condition	–	4
3	Data hold time	(0.27) * Divider	–
4	HIGH of the SCL Period	–	(0.4) * Divider (Master mode)
5	LOW period of the SCL Clock	–	(0.4) * Divider (Master mode)

1 Refer to Table 22-3 for details for divider values.



Note: In receiver mode, after a byte transfer is complete, it is not required to set TXAK = 1. The value of TXAK depends on the requirement of the transmitter device.

Figure 22-7. Flow-Chart of Typical I²C Interrupt Routine

Chapter 23
Configurable Serial Peripheral Interface (CSPI)

The i.MX21 processor has three Configurable Serial Peripheral Interface (CSPI) modules that allow rapid data communication with fewer software interrupts than conventional serial communications.

The primary features of the CSPIs include:

- Master/Slave configurable (CSPI1 and CSPI2 only)
- Three available chip-selects (CSPI1 and CSPI2) for master mode operation (SS0–SS2)
- Up to 32-bit programmable data transfer
- 8 × 32-bit FIFO for both Tx and Rx data

CSPI1 and CSPI2 are equipped with two data FIFOs and is a master/slave configurable serial peripheral interface module, allowing i.MX21 to interface with both external SPI master and slave devices. CSPI3 is also equipped with two data FIFOs however is only a master.

This section describes how CSPI communicates with external devices. CSPI has one 8 × 32-bit data-in FIFO and one 8 × 32-bit data-out FIFO. Incorporating the $\overline{\text{CSPI1\_RDY}}$ and SS control signals, it enables fast data communication with fewer software interrupts. Figure 23-1 illustrates the configurable serial peripheral interface block diagram.

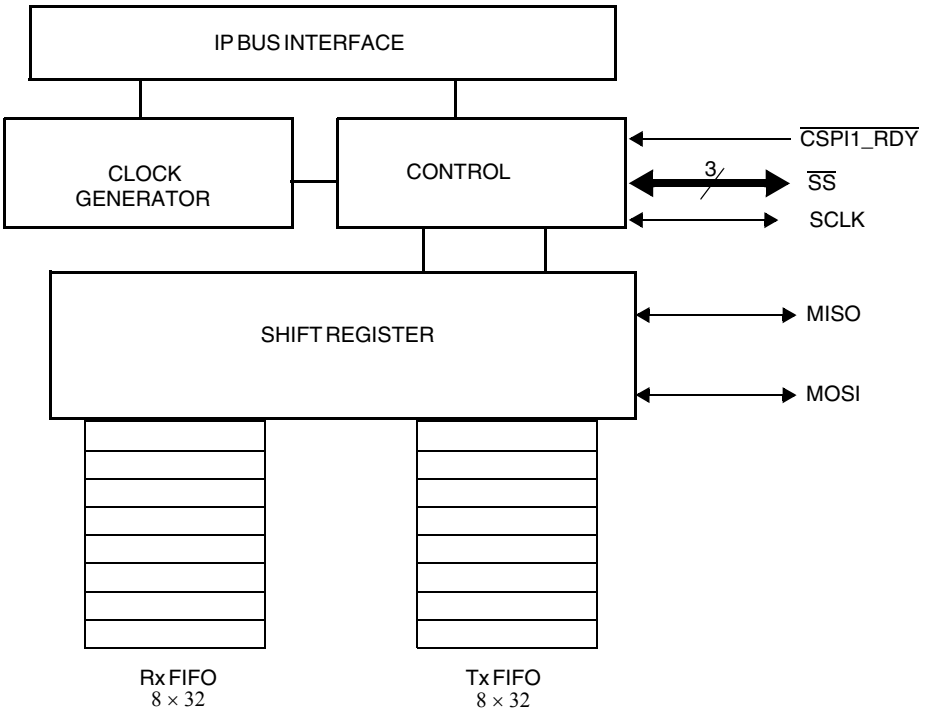


Figure 23-1. Configurable Serial Peripheral Interface Block Diagram

23.1 Operation

When CSPI is configured as master, SS (output) and $\overline{\text{CSPI1\_RDY}}$ (input) signals, are used for data transfer rate control. The sample period control register can be set if a fixed data transfer rate is required.

When CSPI is configured as slave, SS becomes an input signal and can optionally be used for data latching and loading to the internal data shift registers, as well as incrementing internal data FIFO pointers. Figure 23-2 shows the generic CSPI timing.

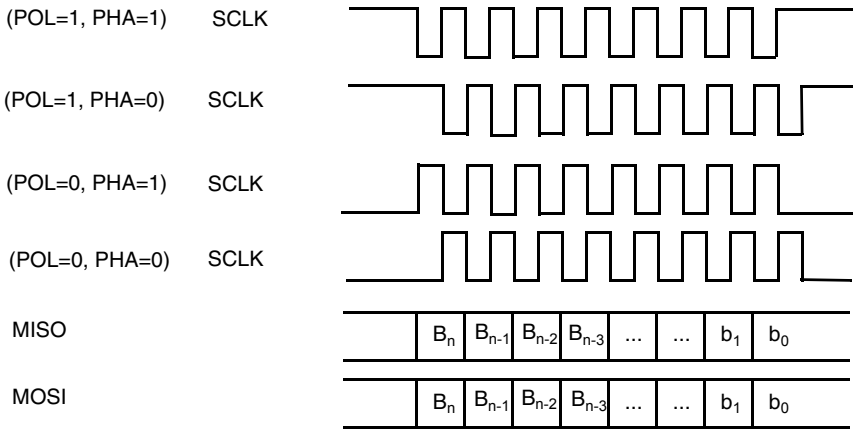


Figure 23-2. Generic CSPI Timing

23.1.1 Phase and Polarity Configurations

The serial peripheral interface master uses the SCLK signal to transfer data in and out of the shift register. Data is clocked using any one of four programmable clock phase and polarity combinations. During Phase 0, Polarity 0 and Phase 1, Polarity 1 operations, output data changes on the falling clock edge and input data is shifted in on the rising edge. The most-significant bit is output when the CPU loads the transmit data. During Phase 1, Polarity 0 and Phase 0, Polarity 1 operations, output data changes on the rising edges of the clock and is shifted in on falling edges. The most-significant bit is output on the first rising SCLK edge. Polarity inverts SCLK, but does not change the edge-triggered events that are internal to the serial peripheral interface master. This flexibility allows it to operate with most serial peripheral devices.

23.1.2 Signals

The following signals are used to control the serial peripheral interface master:

- **MOSI**—Master Out Slave In bidirectional signal, which is TxD output signal from the data shift register in master mode. In Slave mode it is RxD input to the data shift register.
- **MISO**—Master In Slave Out bidirectional signal, which is RxD input signal to the data shift register in master mode. In Slave mode it is TxD output from the data shift register.
- **SCLK**—CSPI Clock bidirectional signal, which is CSPI clock output in master mode. In slave mode it is an input CSPI clock signal.
- **SS[2:0]**, Slave Select bidirectional signal, output in master mode, and input in slave mode.
- **$\overline{\text{CSPI1\_RDY}}$**—This input signal is used only in master mode. It will edge or level trigger a CSPI burst if used. This signal is only available for CSPI1; it is not present on CSPI2 and CSPI3.

23.2 Programming Model

The following sections provide the register summary and the programming model for the 3 CSPI modules in the i.MX21. The Register Summary in Table 23-1 lists all registers of the CSPI module by ascending address. The absolute address of each register is given, as is the value of each bit for reads and writes.

NOTE

CSPI3 is a master mode port only.

Table 23-1. CSPI Register Summary

Name		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RxDataReg	R	RxData[31:16]															
	W																
0x1000 E000		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1000 F000	R	RxData[15:0]															
0x1001 7000	W																
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TxData Reg	R																
	W	Tx Data [31:16]															
0x1000 E004		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1000 F004	R																
0x1001 7004	W	Tx Data [15:0]															
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Control Reg	R	0	0	0	0	0	0	0	0	BURST	SDHC_SPIEN	SWAP	CS[1:0]		DataRate[4:2]		
	W																
0x1000 E008		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1000 F008	R	DataRate[1:0]		DR CTL[1:0]		MODE	SPIEN	XCH	SSPOL	SSCTL	PHA	POL	BIT COUNT[4:0]				
0x1001 7008	W																
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTREG	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BOEN	ROEN
	W																
0x1000 E00C		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1000 F00C	R	RFEN	RHEN	RREN	TSHFEEN	TFEN	THEN	TEEN	BO	RO	RF	RH	RR	TSHFE	TF	TH	TE
0x1001 700C	W																
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Test Reg	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
0x1000 E010		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1000 F010	R		LBC	INIT	SS_ASSERT	SSTATUS[3:0]				RXCNT[3:0]				TXCNT[3:0]			
0x1001 7010	W																
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Period Reg	R																
	W																
0x1000 E014		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1000 F014	R	CSRC	WAIT[14:0]														
0x1001 7014	W																

Table 23-1. CSPI Register Summary (continued)

Name		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAREG **0x1000 E018 0x1000 F018 0x1001 7018**	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	THDEN	TEDEN	RFDEN	RHDEN	0	0	0	0	TH DMA	TE DMA	RF DMA	RH DMA	0	0	0	0
	W																
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reset Reg **0x1000 E01C 0x1000 F01C 0x1001 701C**	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	START
	W																

23.2.1 Rx Data Registers

		Addr
RxData1	Receive Data Register 1	0x1000E000
RxData2	Receive Data Register 2	0x1000F000
RxData3	Receive Data Register 3	0x10017000

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RxData															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RxData															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 23-2. RxData Register Description

Name	Description
RxData Bits 31–0	**Rx Data**—This is a 32-bit read-only register.This holds the top of the 8 × 32 RxFIFO received from external CSPI device during data transaction. It is not a valid value if the RR bit in the Interrupt control/status register is cleared.

23.2.2 Tx Data Registers

These 32-bit write only registers form the top of each respective 8 × 32 TxFIFO. Writing to the associated TxFIFO is permitted as long as it is not full even though XCH bit is set—that is, the user can write to TXFIFO during CSPI data exchange process. Writes to this register are ignored while the SPIEN bit is clear.

		Addr
TxData1	Transmit Data Register 1	0x1000E004
TxData2	Transmit Data Register 2	0x1000F004
TxData3	Transmit Data Register 3	0x10017004

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TxData															
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TxData															
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 23-3. TxData Register Description

Name	Description
TxData Bits 31–0	**Tx Data**—This register forms the top CSPI data to be loaded to the 8 × 32 TxFIFO. In master or slave mode, maximum of 8 data words are loaded. Data write to this register is 32-bit size only. Number of bits to be shifted out of a 32-bit FIFO element is determined by the BIT COUNT of control register. The unused MSBs are don't care. For example, to transfer 10-bit data, 32-bit word is written to this register and the 22 MSBs are don't care and will not be shifted out. In slave mode, if no data is loaded to this Tx data FIFO, zeroes are shifted out on the associated MISO signal.

23.2.3 Control Registers

Each Control Register is 32 bits. The 8 MSBs are reserved and read 0.

		Addr
ControlReg1	Control Register 1	0x1000E008
ControlReg2	Control Register 2	0x1000F008
ControlReg3	Control Register 3	0x10017008

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									BURST	SDHC_SPIEN	SWAP	CS		DataRate		
TYPE	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DataRate		DR CTL		MODE	SPIEN	XCH	SSPOL	SSCTL	PHA POL	POL	BIT COUNT				
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 23-4. Control Register Description

Name	Description	Settings
Reserved Bits 31–24	Reserved—These bits are reserved and should read 0.	
BURST Bit 23	**Burst or Continuous Bit**—Used to program the CSPI data in continuous mode or insertion an idle time between two consecutive data transfers in Master mode.	0 = Idle time inserted between two transfers 1 = No idle time between consecutive data transfers.
SDHC_SPIEN Bit 22	**SDHC SPI Mode Enable Bit**—Used to enable the SDHC SPI mode **Note:** In SDHC SPI mode the maximum frequency of SCLK is PERCLK2 divided by 3. The setting of the DATARATE[4:0] field to %00001 is valid for SDHC SPI mode only. With this mode disabled, the DATARATE field should be set to %00010 or higher.	0= SDHC SPI mode disabled 1= SDHC SPI mode enable
SWAP Bit 21	**TxFIFO Swap Bit**—Used for byte swapping of Tx FIFO and RxFIFO data	0= No Swapping 1= Swap data
CS Bits 20–19	**Chip Select**—In master mode, these two bits select the Slave by asserting SSn. • In **Master Mode** SSn are the outputs. Only the selected SSn signal is active the remaining 3 signals are tristated • In **Slave mode**, these two bits select the master. SSn are the inputs and only one is asserted at a time. **Note:** With CSPI1 and CSPI2, a setting of 11 is reserved (because SS3 isn't available on MX21). For CSPI3, only a setting of 00 is valid (because it only has one SS). This only works on master mode for CSPI3.	**In Master Mode:** 00 = SS0 is asserted 01 = SS1 is asserted 10 = SS2 is asserted 11 = Reserved **In Slave Mode:** SS0,SS1, and SS2 become inputs 00 = SS0 is selected 01 = SS1 is selected 10 = SS2 is selected 11 = Reserved

Table 23-4. Control Register Description (continued)

Name	Description	Settings
Data Rate Bits 18–14	**Data Rate**—This field selects the bit rate (frequency of SPICLK) based on the division of PERCLK2. The equations used are: • For even states other than 0: Divide ratio is $2 \times 2^{n/2}$ • For odd states: Divide ratio is $3 \times 2^{(n-1)/2}$ **Note:** Divide ratio of 3 is valid only if SDHC_SPIEN bit in CONTROLREG is set.	00000 = Reserved 00001 = Divide by 3 00010 = Divide by 4 00011 = Divide by 6 00100 = Divide by 8 00101 = Divide by 12 00110 = Divide by 16 00111 = Divide by 24 01000 = Divide by 32 01001 = Divide by 48 01010 = Divide by 64 01011 = Divide by 96 01100 = Divide by 128 01101 = Divide by 192 01110 = Divide by 256 01111 = Divide by 384 10000 = Divide by 512 10001 = Divide by 768 10010 = Divide by 1024 All other = Reserved
DRCTL Bits 13–12	**$\overline{\text{CSPI1\_RDY}}$ Control**—In master mode, these two bits control the response of the $\overline{\text{CSPI1\_RDY}}$ input. Note that RDY is only available for CSPI1. In slave mode, it is ignored.	00 = Ignored 01 = Falling edge trigger input 10 = Active low level trigger input 11 = Reserved
MODE Bit 11	**CSPI Mode Select**—This bit select the Master/Slave mode of CSPI. CSPI3 only works in master mode.	0 = CSPI is in slave mode 1 = CSPI is in master mode
SPIEN Bit 10	**CSPI Module Enable**—This bit enables the serial peripheral interface. This bit must be asserted before initiating an exchange. Writing a 0 to this bit will flush the Rx and Tx FIFO. However, the PCCR0 register contains the master enable for the CSPI.	0 = The serial peripheral interface is disabled 1 = The serial peripheral interface is enabled
XCH Bit 9	**Data Exchange**—In master mode, writing a 1 to this bit triggers data exchange. This bit remains set while either the exchange is in progress, or CSPI is waiting for active $\overline{\text{CSPI1\_RDY}}$ input if $\overline{\text{CSPI1\_RDY}}$ is enabled through DRCTL. This bit is cleared automatically when all data in TxFIFO and shift register are shifted out. In slave mode this bit must be clear. Before setting the XCH bit to initiate an exchange, the user must ensure that it has been automatically cleared which indicates that the exchange is over.	1 = Initiates exchange (write) or busy (read) 0 = Idle
SSPOL Bit 8	**SS Polarity Select**—In both master and slave mode, this bit selects the polarity of SS signal.	0 = Active low 1 = Active high

Table 23-4. Control Register Description (continued)

Name	Description	Settings
SSCTL Bit 7	**SS Wave Form Select**—In **master mode**, this bit select the output wave form for SS signal. In **slave mode**, this bit controls RxFIFO advancement.	**In Master Mode** 0 = SS stays low between CSPI bursts 1 = Insert pulse between CSPI bursts **In Slave Mode** 0 = RxFIFO advanced by Bit Count 1 = RxFIFO advanced by SS rising edge
PHA Bit 6	**Phase**—This bit controls the clock/data phase relationship.	0 = Phase 0 operation 1 = Phase 1 operation
POL Bit 5	**Polarity**—This bit controls the polarity of the SPICLK signal.	0 = Active high polarity (0 = idle) 1 = Active low polarity (1 = idle)
BIT COUNT Bits 4–0	**Bit Count**—This field selects the length of the transfer. A maximum of 32 bits can be transferred. In **master mode**, a 32-bit data word is loaded from TxFIFO to shift register and only the least n bits (n=BIT COUNT) are shifted out. The next 32-bit word is then loaded to shift register. In **slave mode** and when SSCTL bit is 0 this field controls the number of bits received as a data word loaded to RxFIFO. When SSCTL bit is 1 and SS rising edge is faster than BIT COUNT is treated, then this field is don't care.	00000 = 1-bit transfer 00001 = 2-bit transfer … 01110 = 15-bit transfer 01111 = 16-bit transfer … 11110 = 31-bit transfer 11111 = 32-bit transfer

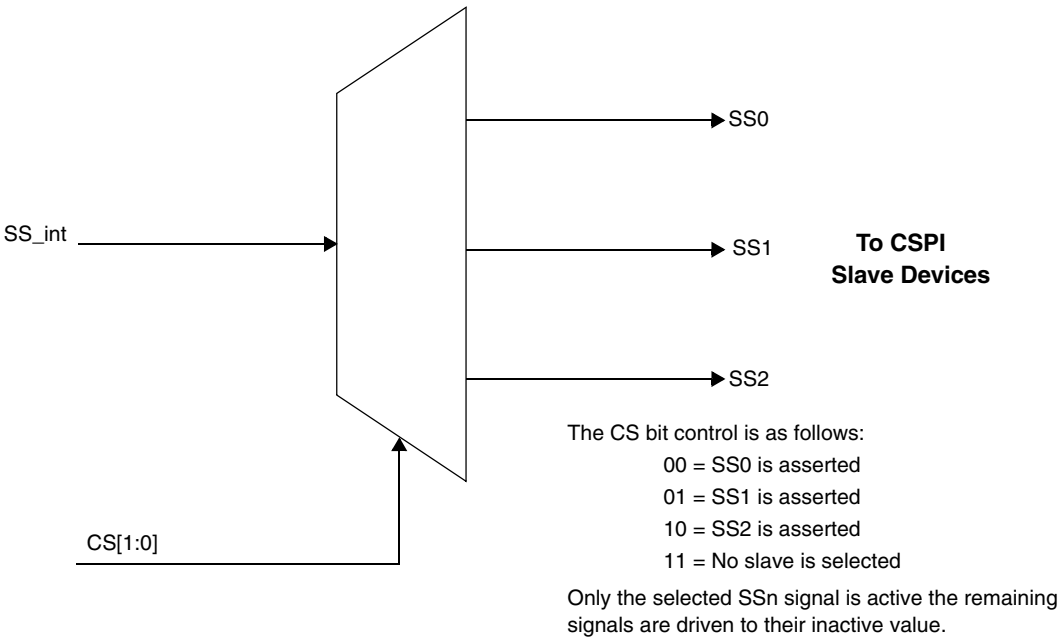


Figure 23-3. In Master Mode

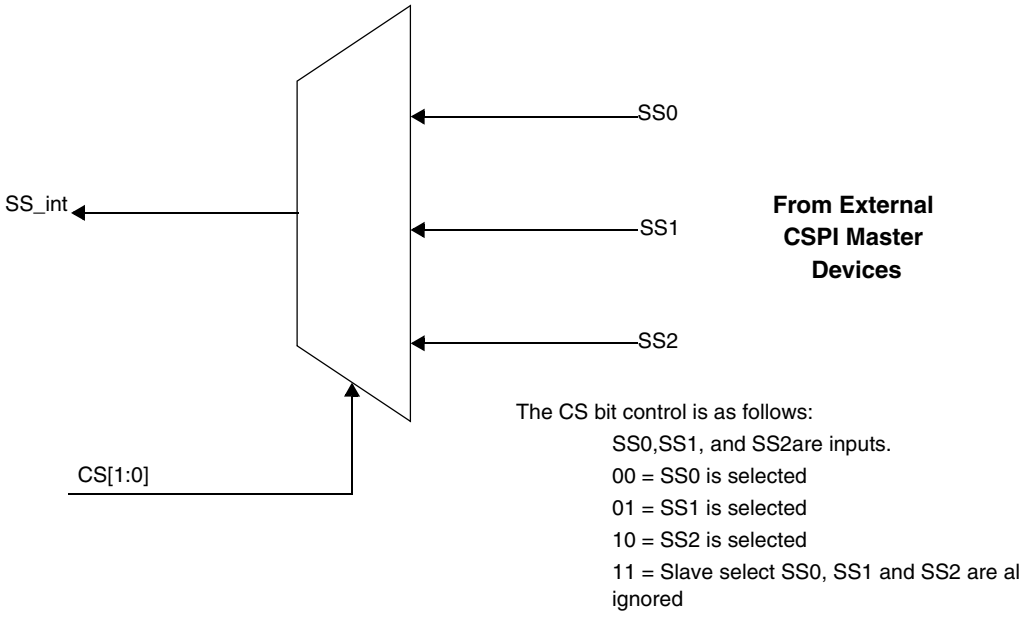


Figure 23-4. In Slave Mode

23.2.4 Interrupt Control and Status Register

The configurable serial peripheral interface control/status (INTREG) register controls how the serial peripheral interface operates and reports its status.The register is 32 bits. The 14 MSBs are reserved and always read as 0.

INT1	Interrupt Control/Status Register 1	0x1000E00C
INT2	Interrupt Control/Status Register 2	0x1000F00C
INT3	Interrupt Control/Status Register 3	0x1001700C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															BOEN	ROEN
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFEN	RHEN	RREN	TSHFEEN	TFEN	THEN	TEEN	BO	RO	RF	RH	RR	TSHFE	TF	TH	TE
TYPE	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 23-5. Interrupt Control/Status Register Description

Name	Description	Settings
Reserved Bits 31–18	Reserved—These bits are reserved and should read 0.	
BOEN Bit 17	**Bit Count Overflow Interrupt Enable**— This bit enables an interrupt when bit count overflow flag BO is set.	0 = Disable 1 = Enable
ROEN Bit 16	**RxFIFO Overflow Interrupt Enable**— This bit enables an interrupt when receive overflow flag “RO” is set.	0 = Disable 1 = Enable
RFEN Bit 15	**RxFIFO Full Interrupt Enable**—This bit enables an interrupt when RxFIFO full flag “RF” is set.	0 = Disable 1 = Enable
RHEN Bit 14	**RxFIFO Half Interrupt Enable**— This bit enables an interrupt when RxFIFO half flag “RH” is set.	0 = Disable 1 = Enable
RREN Bit 13	**RxFIFO Data Ready Interrupt Enable**— This bit enables an interrupt when RxFIFO ready flag “RR” is set.	0 = Disable 1 = Enable
TSHFEEN Bit 12	**Tx FIFO and Tx Shift Register Empty Interrupt Enable**—If enabled an interrupt is generated when both the TxFIFO and Tx Shift register are empty. This interrupt flag is cleared when a write operation is performed on the TxFIFO.In **Master mode**, this interrupt is generated only after the completion of a transmission and if SPIEN is set. In **Slave mode** this interrupt is generated only after the completion of a transmission and does not care for the $\overline{\text{CSPI1\_RDY}}$ signal. This interrupt will not be asserted before any transmission has occurred since in this scenario also both TxFIFO and TxShift register are empty.	0 = Disable 1 = Enable
TFEN Bit 11	**TxFIFO Full Interrupt Enable**— This bit enables an interrupt when TxFIFO full flag TF is set. When the TXFIFO full interrupt bit is set, it means there are 8 words in the TXFIFO and 1 word in shift register.	0 = Disable 1 = Enable
THEN Bit 10	**TxFIFO Half Interrupt Enable**—This bit enables an interrupt when TxFIFO half flag TH is set. When the TXFIFO half full interrupt bit is set, it means there are 4 or more words in the TXFIFO and 1 word in shift register.	0 = Disable 1 = Enable
TEEN Bit 9	**TxFIFO Empty Interrupt Enable**—This bit enables an interrupt when TxFIFO empty flag TE is set. When the CSPI TXFIFO empty interrupt bit is set, it means no words are in the TXFIFO, however 1 or 0 words in the shift register.	0 = Disable 1 = Enable
BO Bit 8	**Bit Count Overflow**—This bit is set when CSPI is in "Slave CSPI FIFO advanced by SS rising edge” mode and the slave is receiving more than 32 bits in one burst. This bit is clear after a data read from SPIRXD register. However, there is nothing to remember which data word is overflowed, hence, the bad data word may still be in the FIFO if it is not empty.	0 = No bit count overflow 1 = At least one data word in RxFIFO has bit count overflow error
RO Bit 7	**RxFIFO Overflow**—This bit indicates that the RxFIFO is overflow. At least one new written data word is lost. The RO flag is automatically cleared after a data read.	0 = RxFIFO is not overflow 1 = RxFIFO is overflow. At least one data word in RxFIFO is over written.
RF Bit 6	**RxFIFO Full Status**—This flag is set when RxFIFO has 8 data words.	0 = Less than 8 data words in RxFIFO. 1 = 8 data words in RxFIFO.

Table 23-5. Interrupt Control/Status Register Description (continued)

Name	Description	Settings
RH Bit 5	**RxFIFO Half Status**—This flag is set when RxFIFO has more than or equal to 4 data words.	0 = Less than 4 data words in RxFIFO. 1 = More than or equal to 4 data words in RxFIFO.
RR Bit 4	**RxFIFO Data Ready Status**—This flag is set when RxFIFO has at least one data word.	0 = RxFIFO is empty. 1 = At least one data word is ready in Rx FIFO.
TSHFE Bit 3	**TxFIFO and TxShift Register Empty**—This flag is set when both the TxFIFO and Tx Shift register are empty after the completion of data transfer.When TSHFE bit is set and a write is performed on the TxDATA register, the first data is immediately transferred to the Tx shift register. This flag will not be asserted before any transmission has occurred since in this scenario also both TxFIFO and TxShift register are empty.	0 = At least one data word is in TxFIFO or at least one data bit is in Tx Shift Register. 1 = TxFIFO and Tx Shift Register are empty.
TF Bit 2	**TxFIFO Full Status**—This flag is set when Transmit buffer has 9 data words (8 in TxFIFO, 1 in Tx Shift register).	0 = Less than 9 data words in buffer. 1 = 9 data words in buffer.
TH Bit 1	**TxFIFO Half Status**—This flag is set when TxFIFO has more than or equal to 4 empty slots (less than or equal to 5 words left for transmit—that is, 4 or more in TxFIFO, 1 in Tx Shift register).	0 = Less than 4 empty slots in TxFIFO. 1 = More than or equal to 4 empty slots in TxFIFO
TE Bit 0	**TxFIFO Empty Status**—This flag is set when the TxFIFO is empty (0 in TxFIFO, 1 or 0 words in Tx Shift register). When CSPI is enabled and TE bit is set, the first data written to the TxFIFO is immediately loaded into the Tx shift register.	0 = At least one data word is in Tx FIFO. 1 = TxFIFO is empty, but data shifting may still be on-going. To make sure no data transaction is on-going, read XCH bit in control register.

23.2.5 CSPI Test Register

These registers are used for test purpose and is used to report the State machine status as well as Rx and Tx FIFO counter status.The register is 32 bits. The high 17 bits are reserved bits and always be read as 0.

Test1	Test Register 1	0x1000E010
Test2	Test Register 2	0x1000F010
Test3	Test Register 3	0x10017010

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		LBC	INIT	SS_ASSERT	SSTATUS				RXCNT				TXCNT			
TYPE	r	rw	rw	rw	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 23-6. Test Register Description

Name	Description	Settings
RSV Bit 15–31	Reserved	N/A
LBC Bit 14	Loop Back Control	0 = Disable 1 = The TX and RX are connected internally for test purpose
INIT Bit 13	**Initialize**—This bit initializes the State Machine	0 = No initialization 1 = Initialize
SS_ASSERT Bit 12	**SS Assert/Deassert**—Setting this bit causes SSn to remain asserted throughout the data transmission and remains asserted after the TxFIFO empty and TxSHFD flags are set. If this bit is not set SSn assertion/deassertion is controlled by the SSPOL,SSCTL bits in the CONTROLREG.	0 = SSn is asserted/deasserted as per settings of the SSPOL,SSCTL bits in the CONTROLREG. 1 = SSn remains asserted after TxSHFD flag is set.
SSTATUS Bits 11–8	State Machine Status—This field indicates the state machine status. These bits are used for test purpose only	N/A

Table 23-6. Test Register Description (continued)

Name	Description	Settings
RXCNT Bits 7–4	**RxFIFO Counter**—This field indicates the number of data words in RxFIFO.	0000 = RXFIFO is empty 0001 = 1 data word in RXFIFO 0010 = 2 data word in RXFIFO 0011 = 3 data word in RXFIFO 0100 = 4 data word in RXFIFO 0101 = 5 data word in RXFIFO 0110 = 6 data word in RXFIFO 0111 = 7 data word in RXFIFO 1000 = 8 data word in RXFIFO
TXCNT Bits 3–0	**TxFIFO Counter**—This field indicates the number of data words in TxFIFO. When this field is 0, there may be 1 or 0 words in the TX Shift register. When this field is equal or larger than 1, because the first data is moved to TX Shift register, TXCNT is equal to the actual number minus one.	0000 = TxFIFO is empty 0001 = 1 data word in TxFIFO 0010 = 2 data word in TxFIFO 0011 = 3 data word in TxFIFO 0100 = 4 data word in TxFIFO 0101 = 5 data word in TxFIFO 0110 = 6 data word in TxFIFO 0111 = 7 data word in TxFIFO 1000 = 8 data word in TxFIFO

23.2.6 CSPI Sample Period Control Register

This register controls the time inserted between data transaction in master mode. The time inserted between samples can be from 0 to about 1 second at the resolution of bit clock or 32.768 KHz clock. The register is 32 bits. The high 16 bits are reserved bits and always be read as 0.

Period1	Period Control Register 1	0x1000E014
Period2	Period Control Register 2	0x1000F014
Period3	Period Control Register 3	0x10017014

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSRC	WAIT														
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 23-7. Period Control Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	

Table 23-7. Period Control Register Description (continued)

Name	Description	Settings
CSRC Bit 15	**Clock Source to Counter**—This bit selects between the CSPI bit clock and the 32Khz clock as the clock source for the wait period insertion counter.	0 = Bit clock 1 = 32.68KHz
WAIT Bits 14–0	**Wait**—Number of clocks (either bit clock or 32Khz clock) inserted between data transaction.	0000 = 0 clock 0001 = 1 clock 0002 = 2 clocks … 7FFF = 32,767 clocks

23.2.7 DMA Control Register

The CSPI's DMA control/status (DMAREG) register controls how the DMA interface of the CSPI operates and reports its status. The register is 32 bits.

		Addr
DMA1	DMA Register 1	0x1000E018
DMA2	DMA Register 2	0x1000F018
DMA3	DMA Register 3	0x10017018

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	THDEN	TEDEN	RFDEN	RHDEN					THDMA	TEDMA	RFDMA	RHDMA				
TYPE	rw	rw	rw	rw	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 23-8. DMA Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
THDEN Bit 15	**TxFIFO Half DMA Request Enable**—This bit enables the DMA Tx request if THDMA flag is set.	0 = Disable 1 = Enable
TEDEN Bit 14	**TxFIFO Empty DMA Request Enable**—This bit enables the DMA Tx request if TEDMA flag is set.	0 = Disable 1 = Enable
RFDEN Bit 13	**RxFIFO Full DMA Request Enable**—This bit enables the DMA Rx request if RFDMA flag is set.	0 = Disable 1 = Enable
RHDEN Bit 12	**RxFIFO Half DMA Request Enable**—This bit enables the DMA Rx request if RHDMA flag is set.	0 = Disable 1 = Enable
Reserved Bits 11–8	Reserved—These bits are reserved and should read 0.	

Table 23-8. DMA Register Description (continued)

Name	Description	Settings
THDMA Bit 7	**TxFIFO Half Status**—This flag is set when TxFIFO has more than or equal to 4 empty slots.	0 = Less than 4 empty slots in TxFIFO. 1 = More than or equal to 4 empty slots in TxFIFO.
TEDMA Bit 6	**TxFIFO Empty Status**—This flag is set when TxFIFO is empty.	0 = At least one data word is in Tx FIFO. 1 = TxFIFO is empty, but data shifting may still be on-going. To ensure no data transaction is on-going, read XCH bit in control register.
RFDMA Bit 5	**RxFIFO Full Status**—This flag is set when RxFIFO is full i.e it has 8 data words.	0 = Less than 8 data words in RxFIFO. 1 = 8 data words in RxFIFO.
RHDMA Bit 4	**RxFIFO Half Status**—This flag is set when RxFIFO has more than or equal to 4 data words.	0 = Less than 4 data words in RxFIFO. 1 = More than or equal to 4 data words in RxFIFO
Reserved Bits 3–0	Reserved—These bits are reserved and should read 0.	

23.2.8 CSPI Soft Reset Register

The CSPI Soft Reset registers are 32-bit registers. Soft Reset is generated by setting the start bit of the Soft Reset register to 1.

Reset1	Soft Reset Register 1	0x1000E01C
Reset2	Soft Reset Register 2	0x1000F01C
Reset3	Soft Reset Register 3	0x1001701C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																START
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 23-9. Soft Reset Register Description

Name	Description	Settings
Reserved Bits 31–1	Reserved—These bits are reserved and should read 0.	
START Bit 0	**Soft Reset Bit**—Soft Reset is generated by setting this bit to 1.The soft reset is asserted for 3 ipg_clk clock cycles and it is automatically cleared. This resets the following registers: a. CONTROLREG b. INTREG c. DMAREG d. TESTREG e. PERIODREG f. RESETREG	0 = No soft reset 1 = Soft reset

Chapter 24
Synchronous Serial Interface (SSI)

This chapter discusses the architecture, programming model, operating modes, and initialization of the Synchronous Serial Interface (SSI). The i.MX21 processor contains two SSI modules (SSI1 and SSI2). These are full-duplex, serial ports that allow the i.MX21 to communicate with a variety of serial devices. These serial devices can be standard codecs, Digital Signal Processors (DSPs), microprocessors, peripherals that implement the Serial Peripheral Interface (SPI), and popular industry audio codecs that implement the inter-IC sound bus standard (I^2S) and Intel AC97 standard.

SSI typically are used to transfer samples in a periodic manner. The i.MX21 processor's SSI consists of independent transmitter and receiver sections with independent clock generation and frame synchronization.

The capabilities of each of the SSI include:

- Independent (asynchronous) or shared (synchronous) transmit and receive sections with separate or shared internal/external clocks and frame syncs, operating in Master or Slave mode.
- Normal mode operation using frame sync
- Network mode operation allowing multiple devices to share the port with as many as thirty-two time slots
- Gated Clock mode operation requiring no frame sync
- 2 sets of Transmit and Receive FIFOs. Each of the four FIFOs is 8 × 24 bits. The two sets of Tx/Rx FIFOs can be used in Network mode to provide 2 independent channels for transmission and reception
- Programmable data interface modes such as I^2S, LSB, MSB aligned
- Programmable word length (8, 10, 12, 16, 18, 20, 22, or 24 bits)
- Program options for frame sync and clock generation
- Programmable I^2S modes (Master, Slave or Normal). Oversampling clock, ccm_ssi_clk available as output from SRCK in I^2S Master mode
- AC97 support
- Completely separate clock and frame sync selections for the receive and transmit sections. In AC97 standard, the clock is taken from an external source and frame sync is generated internally.
- External CCM_SSI_CLK input for use in I^2S Master mode. Programmable oversampling clock (SYS_CLK) of the sampling frequency available as output in master mode at SRCK, when operated in sync mode.
- Programmable internal clock divider
- Time Slot Mask Registers for reduced CPU overhead (for both Tx and Rx)
- SSI power-down feature

- SSI signals are connected to IO pads through the Digital Audio Mux (AUDMUX)
- Programmable wait states for CPU accesses
- IP Interface for register access, compliant to SRS 3.0.2 standard

24.1 References

The following documents are identified for further reading.

- [1] *Audio Codec '97 Specification*, Revision 2.3 Rev 1.0, April 2002
- [2] I^2S *Bus Specification*, Revised: June 5, 1996

24.2 SSI Signal Description

Table 24-1. SSI Signal Description

Name	I/O	Function
$\overline{\text{HARD\_ASYNC\_RESET}}$	Input	Global hardware reset signal
IPG_CLK	Input	IP Interface working clock
IPG_CLK_S	Input	IP Interface register access clock
SSI_SRXD	Input	SSI receive data input
SSI_STXD	Output	SSI transmit data output
SSI_STCK	Input/Output	SSI transmit clock output
SSI_STFS	Input/Output	SSI transmit frame sync output
SSI_SRCK	Input/Output	SSI receive clock output
SSI_SRFS	Input/Output	SSI receive frame sync output

Note: For i.MX21, the clock gating functionality is controlled by the PLL Clock Controller (PCCR0 Register) and this signal is not used. However, SSI still generates this signal based on SSIEN (SCR) and CLKOFF (SOR) bits.

24.3 SSI Architecture

The Synchronous Serial Interface (SSI) is connected to chip pads through the Digital Audio Mux (AUDMUX) module. The AUDMUX can be configured to connect the two SSIs together or to external SSIs or codecs. For more information see Chapter 30, "Digital Audio Mux (AUDMUX)," Section 30.7 on page -11.

Figure 24-1 shows a block diagram of the SSI. It consists of three control registers to set up the port, one status register, separate transmit and receive circuits with FIFO registers, and separate serial clock and frame sync generation for the transmit and receive sections. The second set of Tx and Rx FIFOs, replicates the logic used for the first set of FIFOs.

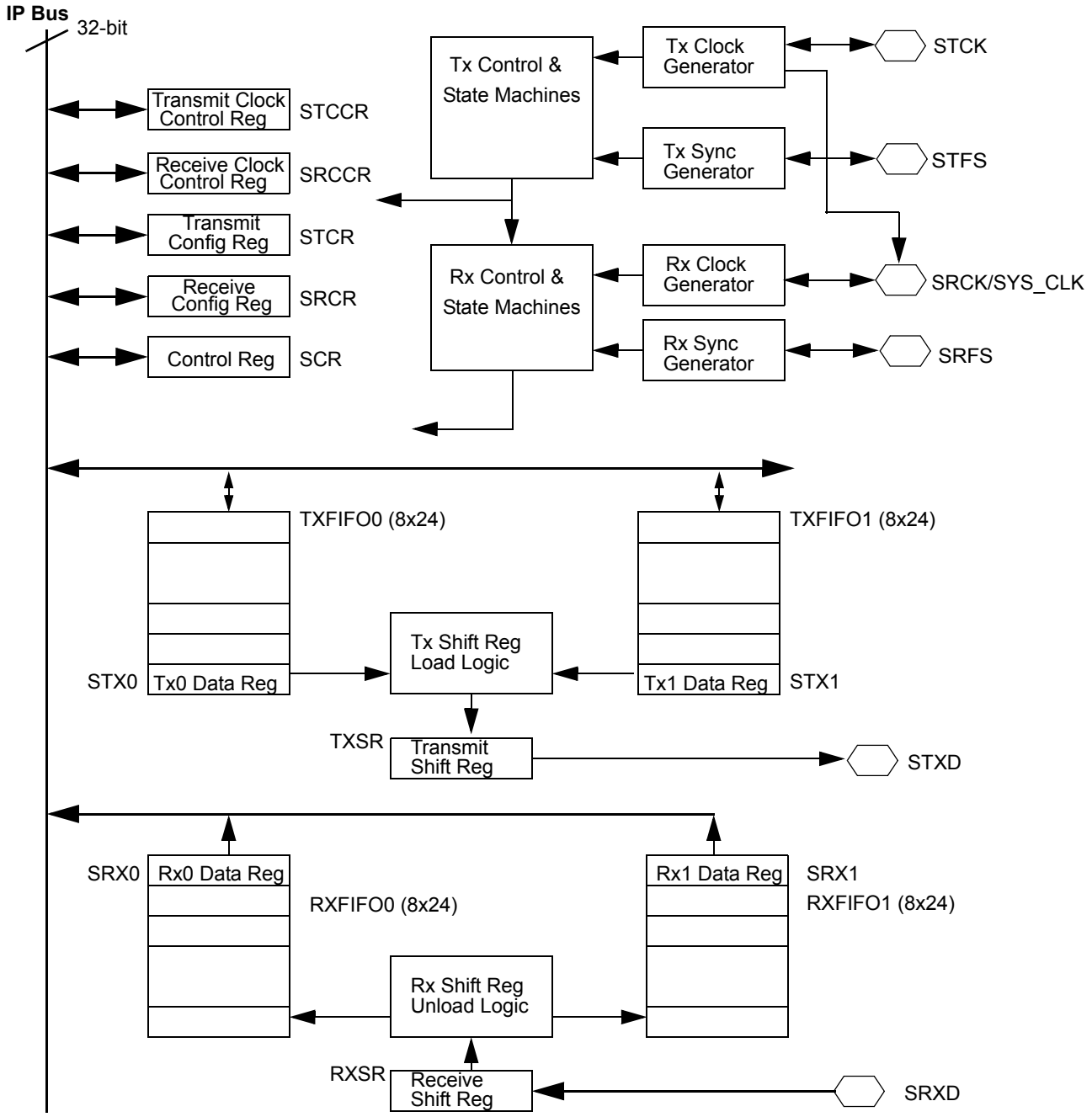


Figure 24-1. SSI Block Diagram

24.3.1 SSI Clocking

The SSI uses the following clocks:

- Bit clock—Used to serially clock the data bits in and out of the SSI port. This clock is either generated internally or taken from an external clock source (through the Tx/Rx clock ports).
- Word clock—Used to count the number of data bits per word (8, 10, 12, 16, 18, 20, 22 or 24 bits). This clock is generated internally from the bit clock.

- Frame clock (Frame Sync)—Used to count the number of words in a frame. This signal can be generated internally from the bit clock, or taken from external source (from the Tx/Rx frame sync ports).
- Sys clock—In master mode, this is an integer multiple of frame clock. It is used in cases when SSI must provide the clock.

Care should be taken to ensure that the bit clock frequency (either internally generated or sourced from external device through Tx/Rx clock ports) is never greater than 1/4 of the ipg_clk frequency.

In Normal mode (SCR[6:5]=00), the bit clock used to serially clock the data, is visible on the Serial Transmit Clock (STCK) and Serial Receive Clock (SRCK) ports. The word clock is an internal clock used to determine when transmission of an 8, 10, 12, 16, 18, 20, 22 or 24 bit word has completed. The word clock in turn then clocks the frame clock, which counts the number of words in the frame. The frame clock can be viewed on the STFS and SRFS frame sync ports, because a frame sync is generated after the correct number of words in the frame have passed. In master and synchronous mode, the unused port SRCK is used as Serial System Clock (SYS_CLK) enabled by the SCR register bit 15, SYS_CLK_EN. This Serial System Clock is an oversampling clock of the frame sync clock (STFS). In this mode, the word length (WL), Prescaler Range (PSR), Prescaler Modulus (PM) and Frame rate (DC) selects the ratio of SYS_CLK to sampling clock STFS. In case of I2S mode, the oversampling clock ccm_ssi_clk can be made available on this port if the SYS_CLK_EN bit is set. This (oversampling) clock is not routed through the AUDMUX, it is routed directly from the CCM (through the GPIO) to the external codec. The relationship between the clocks and the dividers is shown in Figure 24-2. The bit clock can be received from an SSI clock port or can be generated from the ccm_ssi_clk (or the IP Interface working clock) through a divider, as shown in Figure 24-3.

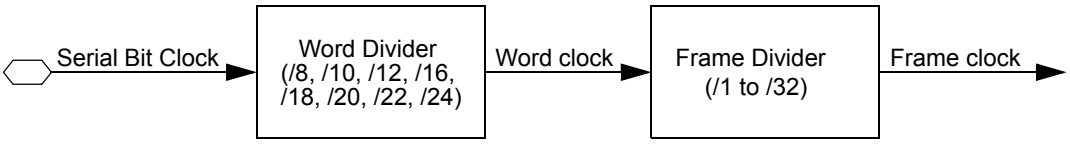


Figure 24-2. SSI Clocking

24.3.2 SSI Clock and Frame Sync Generation

Data clock and frame sync signals can be generated internally by i.MX21 or can be obtained from external sources. If internally generated, the SSI clock generator is used to derive bit clock and frame sync signals from the SSInCLK, n=1 or 2, generated by the PLL and Clock Controller module, or the IP Interface working clock coming as input to the SSI. The SSI clock generator consists of a selectable, fixed prescaler and a programmable prescaler for bit rate clock generation. In Gated Clock mode, the data clock is valid only when data is being transmitted. Otherwise the clock port is pulled to the inactive state. A programmable frame rate divider and a word length divider are used for frame rate sync signal generation.

Figure 24-3 shows a block diagram of the clock generator for the transmit section. The serial bit clock can be internal or external, depending on the Transmit Direction (TXDIR) bit in the SSI Transmit Configuration Register (STCR). The receive section contains an equivalent clock generator circuit.

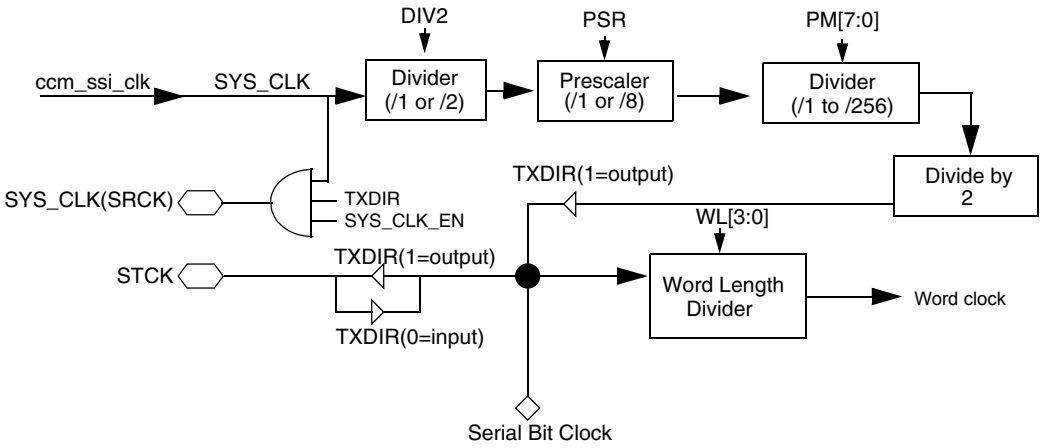


Figure 24-3. SSI Transmit Clock Generator Block Diagram

Figure 24-4 shows the Frame Sync Generator block for the transmit section. When internally generated, both receive and transmit frame sync are generated from the word clock and are defined by the Frame Rate Divider (DC[4:0]) bits and the Word Length (WL[3:0]) bits of the SSI Transmit Clock Control Register (STCCR). The receive section contains an equivalent circuit for the Frame Sync Generator.

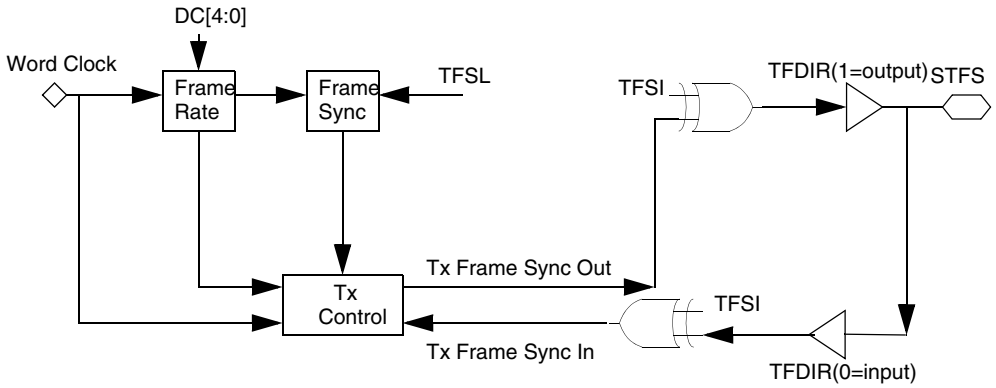


Figure 24-4. SSI Transmit Frame Sync Generator Block Diagram

24.4 Programming Model

Table 24-2 is a summary listing of the registers associated with the operation of the SSI.

The control registers associated with the SSI shows the programming information for SSI interrupts. The Transmit Shift Register (TXSR), Receive Shift Register (RXSR), Receive FIFO, and Transmit FIFO are not user-accessible module.

The base addresses used for the registers are as follows:

- SSI1 = 0x1001 0000
- SSI2 = 0x1001 1000

This section provides a detailed description of the SSI registers.

Table 24-2. SSI Register Summary

Description	Name	Address
SSI1 Transmit Data Register 0	STX10	0x1001 0000
SSI1 Transmit Data Register 1	STX11	0x1001 0004
SSI2 Transmit Data Register 0	STX20	0x1001 1000
SSI2 Transmit Data Register 1	STX21	0x1001 1004
SSI1/2 Transmit FIFO Register 0 (not user-accessible)		—
SSI1/2 Transmit FIFO Register 1 (not user-accessible)		—
SSI Transmit Shift Register (not user-accessible)	TXSR	—
SSI1 Receive Data (read-only) Register 0	SRX10	0x1001 0008
SSI1 Receive Data (read-only) Register 1	SRX11	0x1001 000C
SSI2 Receive Data (read-only) Register 0	SRX20	0x1001 1008
SSI2 Receive Data (read-only) Register 1	SRX21	0x1001 100C
SSI Receive FIFO Register 0 (not user-accessible)		—
SSI Receive FIFO Register 1 (not user-accessible)		—
SSI Receive Shift Register (not user-accessible)	RXSR	—
SSI1 Control Register	SCR1	0x10010010
SSI2 Control Register	SCR2	0x10011010
SSI1 Interrupt Status Register (read-only)	SISR1	0x10010014
SSI2 Interrupt Status Register (read-only)	SISR2	0x10011014
SSI1 Interrupt Enable Register	SIER1	0x10010018
SSI2 Interrupt Enable Register	SIER2	0x10011018
SSI1 Transmit Configuration Register	STCR1	0x1001001C
SSI2 Transmit Configuration Register	STCR2	0x1001101C
SSI1 Receive Configuration Register	SRCR1	0x10010020
SSI2 Receive Configuration Register	SRCR2	0x10011020
SSI1 Transmit Clock Control Register	STCCR1	0x10010024
SSI2 Transmit Clock Control Register	STCCR2	0x10011024
SSI1 Receive Clock Control Register	SRCCR1	0x10010028
SSI2 Receive Clock Control Register	SRCCR2	0x10011028
SSI1 FIFO Control Status Register	SFCSR1	0x1001002C
SSI2 FIFO Control Status Register	SFCSR2	0x1001102C
SSI1 Test Register	STR1	0x10010030
SSI2 Test Register	STR2	0x10011030
SSI1 Option Register	SOR1	0x10010034

Table 24-2. SSI Register Summary (continued)

Description	Name	Address
SSI2 Option Register	SOR2	0x10011034
SS1I AC97 Control Register	SACNT1	0x10010038
SS12 AC97 Control Register	SACNT2	0x10011038
SSI1 AC97 Command Address Register	SACADD1	0x1001003C
SSI2 AC97 Command Address Register	SACADD2	0x1001103C
SSI1 AC97 Command Data Register	SACDAT1	0x10010040
SSI2 AC97 Command Data Register	SACDAT2	0x10011040
SSI1 AC97 Tag Register	SATAG1	0x10010044
SSI2 AC97 Tag Register	SATAG2	0x10011044
SSI1 Transmit Time Slot Mask Register	STMSK1	0x10010048
SSI2 Transmit Time Slot Mask Register	STMSK2	0x10011048
SSI1 Receive Time Slot Mask Register	SRMSK1	0x1001004C
SSI2 Receive Time Slot Mask Register	SRMSK2	0x1001104C

24.4.1 SSI Transmit Data Registers 0 and 1 (STX0/1)

STX0	SSI1 Transmit Data Register 0	0x10010000
STX1	SSI1 Transmit Data Register 1	0x10010004
STX0	SSI2 Transmit Data Register 0	0x10011000
STX1	SSI2 Transmit Data Register 1	0x10011004

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									STX0/1							
TYPE	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STX0/1															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 24-3. STX0/1 Register Description

Name	Description	Settings
Reserved Bits 31–24	Reserved—These bits are reserved and should not be used.	
STX0/1 Bits 23–0	**Transmit Data 0/1 Bits**—These bits store the data to be transmitted by the SSI. These are implemented as the first word of their respective Tx FIFOs. Data written to these registers is transferred to the TXSR, when shifting of the previous data is complete. If both FIFOs are in use, data is alternately transferred from STX0 and STX1, to TXSR. Multiple writes to the STX registers will not result in the previous data being over-written by the subsequent data. STX1 can only be used in Network Two-Channel mode of operation. Protection from over-writing is present irrespective of whether the transmitter is enabled or not. For example: If Tx FIFO0 is in use and user writes Data1–Data9 to STX0, Data9 will not over-write Data1. Data1–Data8 are stored in the FIFO while Data9 is discarded. For example: If Tx FIFO0 is not in use and user writes Data1, Data2 to STX0, then Data2 will not over-write Data1 and will be discarded.	These bits contain the data to be transmitted.

Note: Enable SSI (SSIEN=1) before writing to SSI Transmit Data Registers.

24.4.2 SSI Transmit FIFO 0 and 1 Registers

The SSI Transmit FIFO registers are 8 × 24-bit registers. Transmit Shift Register (TXSR) receives its values from these FIFO registers. Transmitted data is first-in-first-out. When the Transmit Interrupt Enable (TIE) bit in the STCR and either of the Transmit FIFO Empty (TFE0 or 1) bits in the SISR are set, the ARM 9 platform is interrupted whenever the data level in either of the SSI Transmit FIFOs falls below the selected threshold.

24.4.3 SSI Transmit Shift Register (TXSR)

The SSI Transmit Shift Register (TXSR) is a 24-bit shift register that contains the data being transmitted. When a continuous clock is used, data is shifted out to the Serial Transmit Data (STXD) port by the selected (internal/external) bit clock when the associated (internal/external) frame sync is asserted. When a gated clock is used, data is shifted out to the STXD port by the selected (internal/external) gated clock. The Word Length control bits (WL[3:0]) in the STCCR (described in SSI Transmit and Receive Clock Control Registers) determine the number of bits to be shifted out of the TXSR before it is considered empty and can be written to again. This word length can be 8, 10, 12, 16, 18, 20, 22 or 24 bits. The data to be transmitted occupies the most significant portion of the shift register if TXBIT0 is 0, otherwise it occupies the least significant portion. The unused portion of the register is ignored. Data is always shifted out of this register with the Most Significant Bit (MSB) first when the SHFD bit of the STCR is cleared. If this bit is set, the Least Significant Bit (LSB) is shifted out first. Figure 24-5 through Figure 24-8 show the transmitter loading and shifting operation. The figures show the working for some WL values, the same can be extended for other values.

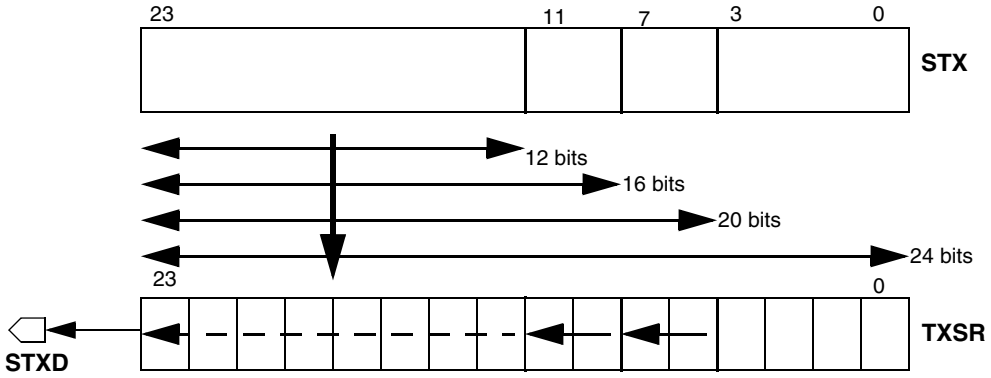


Figure 24-5. Transmit Data Path (TXBIT0=0, TSHFD=0)

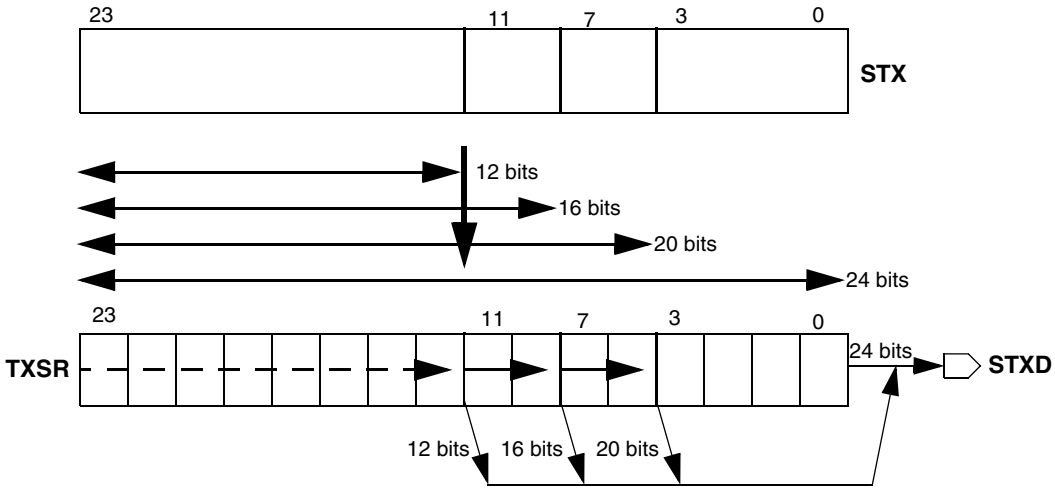


Figure 24-6. Transmit Data Path (TXBIT0=0, TSHFD=1)

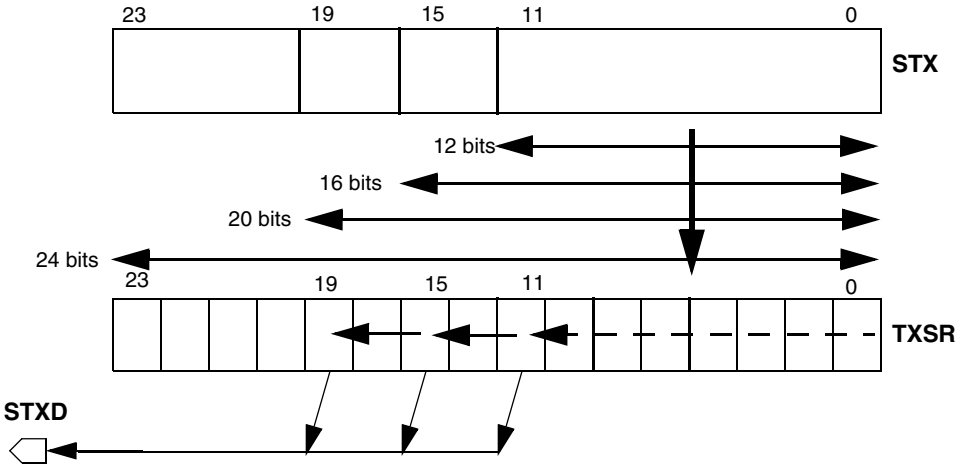


Figure 24-7. Transmit Data Path (TXBIT0=1, TSHFD=0)

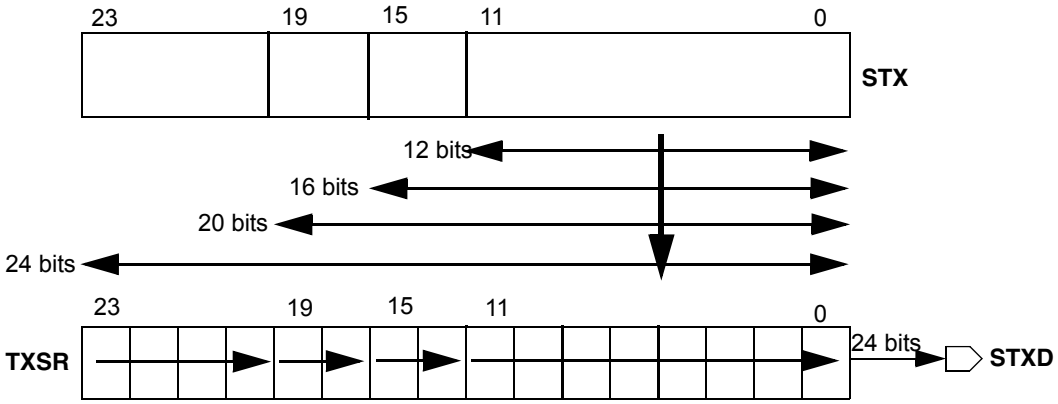


Figure 24-8. Transmit Data Path (TXBIT0=1, TSHFD=1)

24.4.4 SSI Receive Data Registers 0 and 1 (SRX0/1)

SRX0	SSI1 Receive Data Register 0	0x10010008
SRX1	SSI1 Receive Data Register 1	0x1001000C
SRX0	SSI2 Receive Data Register 0	0x10011008
SRX1	SSI2 Receive Data Register 1	0x1001100C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									SRX0/1							
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRX0/1															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 24-4. SRX0/1 Register Description

Name	Description	Settings
Reserved Bits 31–24	Reserved—These bits are reserved and should not be used.	
SRX0/1 Bits 23–0	**Receive Data 0/1 Bits**—These bits store the data received by the SSI. These are implemented as the first word of their respective Rx FIFOs. These bits receive data from the RXSR depending on the mode of operation. In case both FIFOs are in use, data is transferred to each data register alternately. SRX1 can only be used in Network Two-Channel mode of operation.	These bits contain the received data.

24.4.5 SSI Receive FIFO 0 and 1 Registers

The SSI Receive FIFO Registers are 8 × 24-bit registers. They always accept data from the Receive Shift Register (RXSR). The ARM 9 platform is interrupted when the data level in either of the SSI Receive FIFOs reaches the selected threshold, if the associated interrupt is enabled.

24.4.6 SSI Receive Shift Register (RXSR)

The SSI Receive Shift Register (RXSR) is a 24-bit, shift register that receives incoming data from the serial receive data SRXD port. When a continuous clock is used, data is shifted in by the selected (internal/external) bit clock when the associated (internal/external) frame sync is asserted. When a gated clock is used, data is shifted in by the selected (internal/external) gated clock. Data is assumed to be received MSB first if the SHFD bit of the SRCR is cleared. If this bit is set, the data is received LSB first. Data is transferred to the appropriate SSI Receive Data Register (SRX0/1) or Receive FIFOs (if the receive FIFO is enabled and the corresponding SRX is full) after 8, 10, 12, 16, 18, 20, 22 or 24 bits have been shifted in depending on the WL[3:0] control bits. For receiving less than 24 bits of data, LSB bits are appended with zero. Figure 24-9 through Figure 24-12 show the receiver loading and shifting operation. The figures show the working for some WL values, the same can be extended for other values.

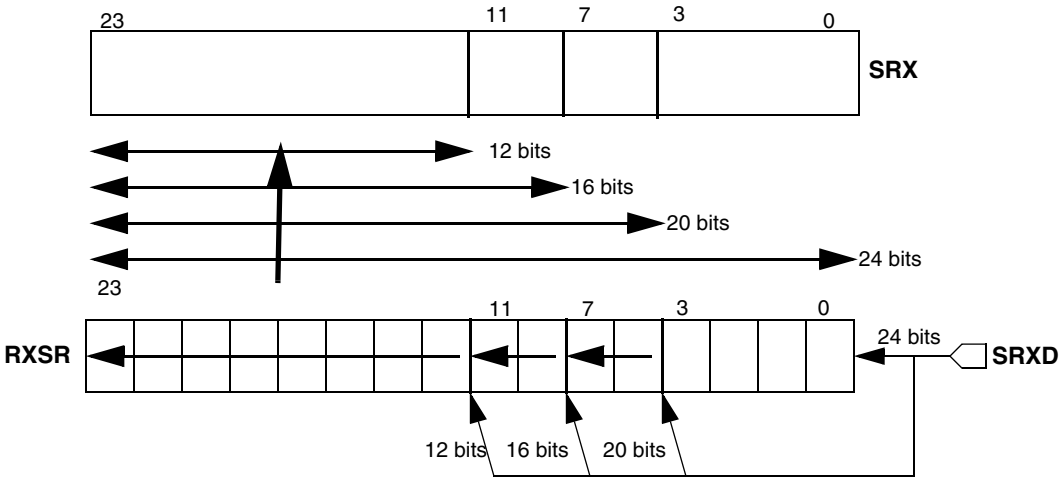


Figure 24-9. Receive Data Path (RXBIT0=0, RSHFD=0)

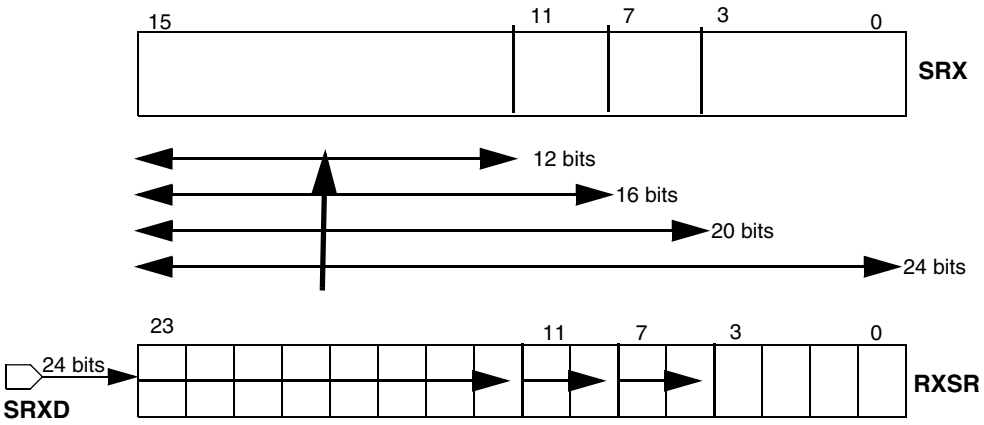


Figure 24-10. Receive Data Path (RXBIT0=0, RSHFD=1)

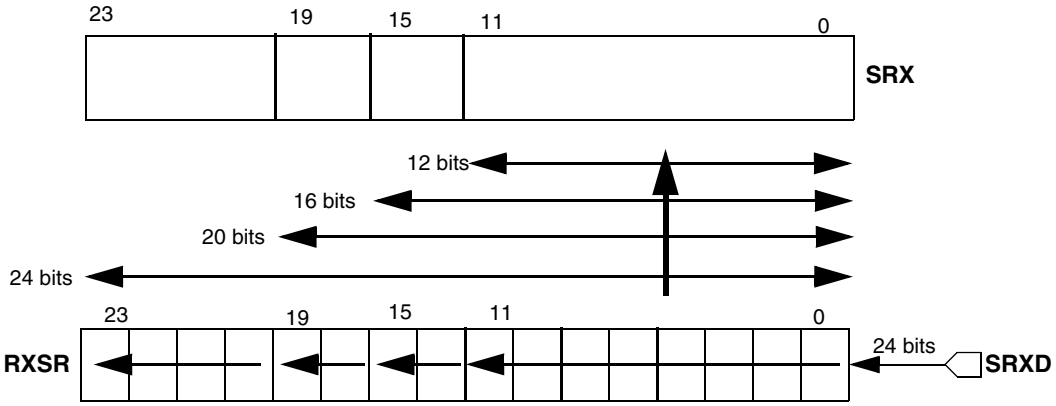


Figure 24-11. Receive Data Path (RXBIT0=1, RSHFD=0)

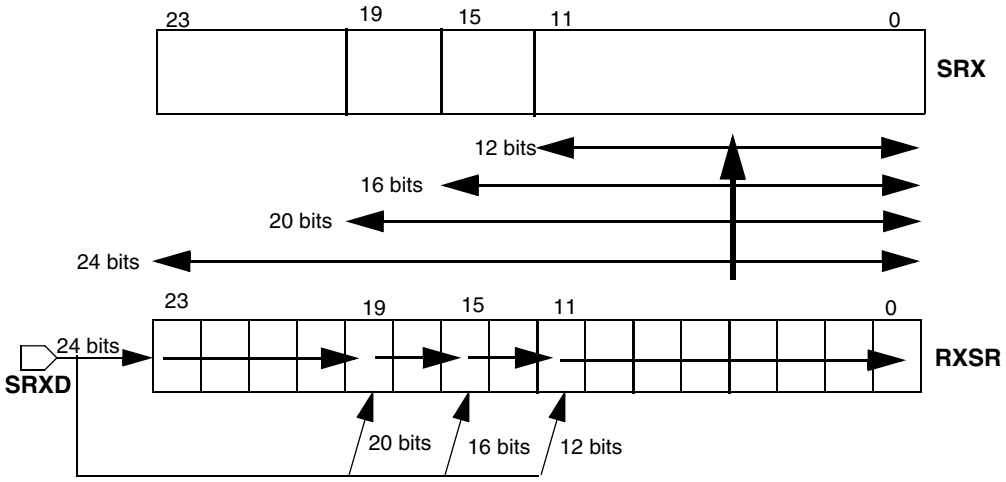


Figure 24-12. Receive Data Path (RXBIT0=1, RSHFD=1)

24.4.7 SSI Control Register (SCR)

The SSI Control Register (SCR) is a 10-bit register used to set up the SSI. SSI reset is controlled by bit 0 in the SCR. SSI operating modes are also selected in this register (except AC97 mode, which is selected in SACNT register). The control bits are described in the following table.

SCR1 SSI1 Control Register 0x10010010
SCR2 SSI2 Control Register 0x10011010

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							CLK_IST	TCH_EN	SYS_CLK_EN	I2S_MODE	SYN	NET	RE	TE	SSIEN	
TYPE	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 24-5. SCR Register Description

Name	Description	Settings
Reserved Bits 31–10	Reserved—These bits are reserved and should not be used.	
CLK_IST Bit 9	**Clock Idle State**—This bit controls the idle state of the transmit clock port during SSI internal gated mode.	0 = Clock idle state is 0. 1 = Clock idle state is 1.
TCH_EN Bit 8	**Two Channel Operation Enable**—This bit allows SSI to operate in the two channel mode. In this mode, 2 time slots should be used out of the possible 32. Any 2 time slots (from 0 to 31) can be selected. The data in the two time slots is alternately handled by the two data registers (0 and 1). While receiving, the RXSR transfers data to SRX0 and SRX1 alternately and while transmitting, data is alternately transferred from STX0 and STX1 to TXSR. If more than 2 slots are to be enabled, then for odd number of slots Two Channel Operation is deprecated and TCH_EN should be cleared. This will ensure that all data is received and transmitted from one FIFO, FIFO0. For an even number of slots, Two Channel Operation can be enabled to optimize usage of both FIFOs or disabled as in the case of odd number of active slots.	0 = Two channel mode disabled. 1 = Two channel mode enabled.
SYS_CLK_EN Bit 7	**System Clock Enable**—When set, this bit allows the SSI to output the input clock used to generate the bit clock at the SRCK port, provided that network mode, synchronous mode and transmit internal clock mode are set. The relationship between bit clock and SYS_CLK is determined by PSR, PM and WL bits. This bit can be used to output the oversampling clock on SRCK port, in I2S Master mode.	0 = SYS_CLK not output on SRCK port. 1 = SYS_CLK output on SRCK port.
I2S_MODE Bits 6–5	**I2S Mode Select**—These bits allow the SSI to operate in Normal, I2S Master or I2S Slave mode. Refer to section 24.9 "I2S Mode Operation" for a detailed description of I2S Mode of operation.	Refer to Table 24-36 for details regarding settings.
SYN Bit 4	**Synchronous Mode**—This bit controls whether SSI is in synchronous mode or not. In synchronous mode, the transmit and receive sections of SSI share a common clock port (STCK) and frame sync port (STFS).	0 = Asynchronous mode selected. 1 = Synchronous mode selected.
NET Bit 3	**Network Mode**—This bit controls whether SSI is in network mode or not.	0 = Network mode not selected. 1 = Network mode selected.
RE Bit 2	**Receive Enable**—This control bit enables the receive section of the SSI. If data is being received when this bit is cleared, the rest of the word is not shifted in and nor is it transferred to the SRX registers. If this bit is set again during a time slot before the second to last bit, then the next word will be received.	0 = Receive section disabled. 1 = Receive section enabled.

Table 24-5. SCR Register Description (continued)

Name	Description	Settings
TE Bit 1	**Transmit Enable**—This control bit enables the transmit section of the SSI. It enables the transfer of the contents of the STX registers to the TXSR and also enables the internal transmit clock. The transmit section is enabled when this bit is set and a word boundary is detected. When this bit is cleared, the transmitter continues to send the data currently in the TXSR and then disables the transmitter. Data can be written to the STX registers with the TE bit cleared (the corresponding TDE bit will be cleared). If the TE bit is cleared and then set during the same transmitted word, the data continues to be transmitted. If the TE bit is set again during a different time slot, data is not transmitted until the next word boundary. The normal transmit enable sequence is to write data to the STX register(s) and then set the TE bit. The normal disable sequence is to clear the TE and TIE bits after the TDE bit is set. In gated clock mode, clearing the TE bit results in the clock stopping after the data currently in TXSR has shifted out. When the TE bit is set, the clock starts immediately.	0 = Transmit section disabled. 1 = Transmit section enabled.
SSIEN Bit 0	**SSI Enable**—This bit is used to enable/disable the SSI. When disabled, all SSI status bits are preset to the same state produced by the Power-on Reset ($\overline{\text{POR}}$), all control bits are unaffected, the contents of Tx and Rx FIFOs are cleared. When SSI is disabled, all internal clocks are disabled (except register access clock). However, the functionality of clock gating in i.MX21 is controlled by PLL Clock controller (PCCR0 Register) and not by SSIEN bit.	0 = SSI is disabled. 1 = SSI is enabled.

24.4.8 SSI Interrupt Status Register (SISR)

The SSI Interrupt Status Register (SISR) is a 19-bit register used to monitor the SSI. This register is read-only and is used by the ARM 9 platform to interrogate the status of the SSI. The status bits are described in the following table. Refer to Table 24-7 for a list of SSI interrupt sources.

NOTE

SSI Status flags are updated when SSI is enabled.

NOTE

All the flags in the SISR are updated after the first bit of the next SSI word has completed transmission or reception. Some status bits (ROE0/1 and TUE0/1) are cleared by reading the SISR followed by a read or write to either the SRX0/1 or STX0/1 registers.

SISR1	SSI1 Interrupt Status Register													0x10010014		
SISR2	SSI2 Interrupt Status Register													0x10011014		
BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
														CMDAU	CMDDU	RXT
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDR1	RDR0	TDE1	TDE0	ROE1	ROE0	TUE1	TUE0	TFS	RFS	TLS	RLS	RFF1	RFF0	TFE1	TFE0
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1

Table 24-6. SISR Register Description

Name	Description	Settings
Reserved Bits 31–19	Reserved—These bits are reserved and should not be used.	
CMDAU Bit 18	**Command Address Register Updated**—This bit causes the Command Address Updated interrupt (when CMDAU_EN bit is set). This status bit is set each time there is a difference in the previous and current value of the received Command Address. This bit is cleared on reading the SACADD register.	0 = No change in SACADD register. 1 = SACADD register updated with a different value.
CMDDU Bit 17	**Command Data Register Updated**—This bit causes the Command Data Updated interrupt (when CMDDU_EN bit is set). This status bit is set each time there is a difference in the previous and current value of the received Command Data. This bit is cleared on reading the SACDAT register.	0 = No change in SACDAT register. 1 = SACDAT register updated with different value.
RXT Bit 16	**Receive Tag Updated**—This status bit is set each time there is a difference in the previous and current value of the received tag. It causes the Receive Tag Interrupt (if RXT_EN bit is set). This bit is cleared on reading the SATAG register.	0 = No change in SATAG register. 1 = SATAG register updated with different value.
RDR1 Bit 15	**Receive Data Ready 1**—This flag bit is set when SRX1 or Rx FIFO 1 is loaded with a new value and Network Two-Channel mode is selected. RDR1 is cleared when the ARM 9 platform reads the SRX1 register. If Rx FIFO 1 is enabled, RDR1 is cleared when the FIFO is empty. If RIE and RDR1_EN are set, a Receive Data 1 interrupt request is issued on setting of RDR1 bit in case Rx FIFO1 is disabled, if the FIFO is enabled, the interrupt is issued on RFF1 assertion. The RDR1 bit is cleared by $\overline{\text{POR}}$ and SSI reset.	0 = No new data for ARM 9 platform to read. 1 = New data for ARM 9 platform to read.
RDR0 Bit 14	**Receive Data Ready 0**—This flag bit is set when SRX0 or Rx FIFO 0 is loaded with a new value. RDR0 is cleared when the ARM 9 platform reads the SRX0 register. If Rx FIFO 0 is enabled, RDR0 is cleared when the FIFO is empty. If RIE and RDR0_EN are set, a Receive Data 0 interrupt request is issued on setting of RDR0 bit in case Rx FIFO0 is disabled, if the FIFO is enabled, the interrupt is issued on RFF0 assertion. The RDR0 bit is cleared by POR and SSI reset.	0 = No new data for ARM 9 platform to read. 1 = New data for ARM 9 platform to read.

Table 24-6. SISR Register Description (continued)

Name	Description	Settings
TDE1 Bit 13	**Transmit Data Register Empty 1**—This flag is set whenever data is transferred to TXSR from STX1 register and Two-Channel mode is selected. If Tx FIFO1 is enabled, this occurs when there is at least one empty slot in STX1 or Tx FIFO1. If Tx FIFO1 is not enabled, this occurs when the contents of STX1 are transferred to TXSR. The TDE1 bit is cleared when the ARM 9 platform writes to STX1. If TIE and TDE1_EN are set, an SSI Transmit Data 1 interrupt request is issued on setting of TDE1 bit. TheTDE1 bit is cleared by POR and SSI reset.	0 = Data available for transmission. 1 = Data needs to be written by the ARM 9 platform for transmission.
TDE0 Bit 12	**Transmit Data Register Empty 0**—This flag is set whenever data is transferred to TXSR from STX0 register. If Tx FIFO 0 is enabled, this occurs when there is at least one empty slot in STX0 or Tx FIFO 0. If Tx FIFO 0 is not enabled, this occurs when the contents of STX0 are transferred to TXSR. The TDE0 bit is cleared when the ARM 9 platform writes to STX0. If TIE and TDE0_EN are set, an SSI Transmit Data 0 interrupt request is issued on setting of TDE0 bit. TheTDE0 bit is cleared by POR and SSI reset.	0 = Data available for transmission. 1 = Data needs to be written by the ARM 9 platform for transmission.
ROE1 Bit 11	**Receiver Overrun Error 1**—Receiver Overrun Error 1—This flag is set when the RXSR is filled and ready to transfer to SRX1 register or to Rx FIFO 1 (when enabled) and these are already full and Network Two-Channel mode is selected. If Rx FIFO 1 is enabled, this is indicated by RFF1 flag, else this is indicated by the RDR1 flag. The RXSR is not transferred in this case. The ROE1 flag causes an interrupt if RIE and ROE1_EN are set. The ROE1 bit is cleared by POR and SSI reset. It is also cleared by reading the SISR with ROE1 bit set, followed by reading the SRX1 register. Clearing the RE bit does not affect the ROE1 bit.	0 = Default interrupt issued to the ARM 9 platform. 1 = Exception interrupt issued to the ARM 9 platform.
ROE0 Bit 10	**Receiver Overrun Error 0—**This flag is set when the RXSR is filled and ready to transfer to SRX0 register or to Rx FIFO 0 (when enabled) and these are already full. If Rx FIFO 0 is enabled, this is indicated by RFF0 flag, else this is indicated by the RDR0 flag. The RXSR is not transferred in this case. The ROE0 flag causes an interrupt if RIE and ROE0_EN are set. The ROE0 bit is cleared by POR and SSI reset. It is also cleared by reading the SISR with ROE0 bit set, followed by reading the SRX0 register. Clearing the RE bit does not affect the ROE0 bit.	0 = Default interrupt issued to the ARM 9 platform. 1 = Exception interrupt issued to the ARM 9 platform.
TUE1 Bit 9	**Transmitter Underrun Error 1—**This flag is set when the TXSR is empty (no data to be transmitted), the TDE1 flag is set, a transmit time slot occurs and the SSI is in Network Two-Channel mode. When a transmit underrun error occurs, the previous data is retransmitted. In Network mode, each time slot requires data transmission (unless masked through STMSK register), when the transmitter is enabled (TE is set). The TUE1 flag causes an interrupt if TIE and TUE1_EN are set. The TUE1 bit is cleared by POR and SSI reset. It is also cleared by reading the SISR with TUE1 bit set, followed by writing to the STX1 register.	0 = Default interrupt issued to the ARM 9 platform. 1 = Exception interrupt issued to the ARM 9 platform.

Table 24-6. SISR Register Description (continued)

Name	Description	Settings
TUE0 Bit 8	**Transmitter Underrun Error 0—**This flag is set when the TXSR is empty (no data to be transmitted), the TDE0 flag is set and a transmit time slot occurs. When a transmit underrun error occurs, the previous data is retransmitted. In Network mode, each time slot requires data transmission (unless masked through STMSK register), when the transmitter is enabled (TE is set). The TUE0 flag causes an interrupt if TIE and TUE0_EN are set. The TUE0 bit is cleared by POR and SSI reset. It is also cleared by reading the SISR with TUE0 bit set, followed by writing to the STX0 register.	0 = Default interrupt issued to the ARM 9 platform. 1 = Exception interrupt issued to the ARM 9 platform.
TFS Bit 7	**Transmit Frame Sync—**This flag indicates the occurrence of transmit frame sync. Data written to the STX registers during the time slot when the TFS flag is set, is sent during the second time slot (in Network mode) or in the next first time slot (in Normal mode). In Network mode, the TFS bit is set during transmission of the first time slot of the frame and is then cleared when starting transmission of the next time slot. In Normal mode, this bit is always high. This flag causes an interrupt if TIE and TFS_EN are set. The TFS bit is cleared by POR and SSI reset. This flag should be ignored and masked in I^2S modes.	0 = No Occurrence of Transmit frame sync. 1 = Transmit frame sync occurred during transmission of last word written to STX registers.
RFS Bit 6	**Receive Frame Sync—**This flag indicates the occurrence of receive frame sync. In Network mode, the RFS bit is set when the first slot of the frame is being received. It is cleared when the next slot begins to be received. In Normal mode, this bit is always high. This flag causes an interrupt if RIE and RFS_EN are set. The RFS bit is cleared by POR and SSI reset. This flag should be ignored and masked in I^2S modes.	0 = No Occurrence of Receive frame sync. 1 = Receive frame sync occurred during reception of next word in SRX registers.
TLS Bit 5	**Transmit Last Time Slot**—This flag indicates the last time slot in a frame. When set, it indicates that the current time slot is the last time slot of the frame. TLS is set at the start of the last transmit time slot and causes the SSI to issue an interrupt (if TIE and TLS_EN are set). TLS is cleared when the SISR is read with this bit set. The TLS bit is cleared by POR and SSI reset. This flag should be ignored and masked in I^2S modes.	0 = Current time slot is not last time slot of frame. 1 = Current time slot is the last transmit time slot of frame.
RLS Bit 4	**Receive Last Time Slot**—This flag indicates the last time slot in a frame. When set, it indicates that the current time slot is the last receive time slot of the frame. RLS is set at the end of the last time slot and causes the SSI to issue an interrupt (if RIE and RLS_EN are set). RLS is cleared when the SISR is read with this bit set. The RLS bit is cleared by POR and SSI reset. This flag should be ignored and masked in I^2S modes.	0 = Current time slot is not last time slot of frame. 1 = Current time slot is the last receive time slot of frame.
RFF1 Bit 3	**Receive FIFO Full 1**—This flag is set when Rx FIFO1 is enabled, the data level in Rx FIFO1 reaches the selected Rx FIFO WaterMark 1 (RFWM1) threshold and the SSI is in Network Two-Channel mode. The setting of RFF1 only causes an interrupt when RIE and RFF1_EN are set, Rx FIFO1 is enabled and the Network Two-Channel mode is selected. RFF1 is automatically cleared when the amount of data in Rx FIFO1 falls below the threshold. The RFF1 bit is cleared by POR and SSI reset. When Rx FIFO1 is full, all further data received (for storage in this FIFO) is ignored until the FIFO contents are read.	0 = Space available in Receive FIFO1. 1 = Receive FIFO1 is full.

Table 24-6. SISR Register Description (continued)

Name	Description	Settings
RFF0 Bit 2	**Receive FIFO Full 0**—This flag is set when Rx FIFO0 is enabled and the data level in Rx FIFO0 reaches the selected Rx FIFO WaterMark 0 (RFWM0) threshold. The setting of RFF0 only causes an interrupt when RIE and RFF0_EN are set and Rx FIFO0 is enabled. RFF0 is automatically cleared when the amount of data in Rx FIFO0 falls below the threshold. The RFF0 bit is cleared by POR and SSI reset. When Rx FIFO0 is full, all further data received (for storage in this FIFO) is ignored until the FIFO contents are read.	0 = Space available in Receive FIFO0. 1 = Receive FIFO0 is full.
TFE1 Bit 1	**Transmit FIFO Empty 1**—This flag is set when Tx FIFO1 is enabled, the data level in Tx FIFO1 falls below the selected Tx FIFO WaterMark 1 (TFWM1) threshold and the Network Two-Channel mode is selected. The setting of TFE1 only causes an interrupt when TIE and TFE1_EN are set, Tx FIFO1 is enabled and Network Two-Channel mode is selected. The TFE1 bit is automatically cleared when the data level in Tx FIFO1 becomes more than the amount specified by the watermark bits. The TFE1 bit is set by POR and SSI reset.	0 = Transmit FIFO1 has data for transmission. 1 = Transmit FIFO1 is empty.
TFE0 Bit 0	**Transmit FIFO Empty 0**—This flag is set when Tx FIFO0 is enabled and the data level in Tx FIFO0 falls below the selected Tx FIFO WaterMark 0 (TFWM0) threshold. The setting of TFE0 only causes an interrupt when TIE and TFE0_EN are set and Tx FIFO0 is enabled. The TFE0 bit is automatically cleared when the data level in Tx FIFO0 becomes more than the amount specified by the watermark bits. The TFE0 bit is set by POR and SSI reset.	0 = Transmit FIFO0 has data for transmission. 1 = Transmit FIFO0 is empty.

Table 24-7. SSI Interrupt Sources

SSI Interrupt Sources
SSI Receive Data with Exception Status 0/1
SSI Receive Data 0/1
SSI Receive Last Time Slot
SSI Transmit Data with Exception Status 0/1
SSI Transmit Data 0/1
SSI Transmit Last Time Slot
SSI AC97 Command Address Updated
SSI AC97 Command Data Updated
SSI AC97 Receive Tag Updated
SSI Receive Frame Sync
SSI Transmit Frame Sync

24.4.9 SSI Interrupt Enable Register (SIER)

The SSI Interrupt Enable Register (SIER) is a 23-bit register used to set up the SSI interrupts. The register bits are described in the following table.

SIER1 SSI1 Interrupt Enable Register 0x10010018
SIER2 SSI2 Interrupt Enable Register 0x10011018

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										RDMAE	RIE	TDMAE	TIE	CMDAU_EN	CMDDU_EN	RXT_EN
TYPE	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDR1_EN	RDR0_EN	TDE1_EN	TDE0_EN	ROE1_EN	ROE0_EN	TUE1_EN	TUE0_EN	TFS_EN	RFS_EN	TLS_EN	RLS_EN	RFE1_EN	RFE0_EN	TFE1_EN	TFE0_EN
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1

Table 24-8. SIER Register Description

Name	Description	Settings
Reserved Bits 31–23	Reserved—These bits are reserved and should not be used.	
RDMAE Bit 22	**Receive DMA Enable**—This bit allows SSI to request for DMA transfers. When enabled, DMA requests are generated when any of the RFF0/1 bits in the SISR are set and if the corresponding RFEN bit is also set. If the corresponding FIFO is disabled, a DMA request is generated when the corresponding RDR bit is set.	0 = SSI DMA requests disabled. 1 = SSI DMA requests enabled.
RIE Bit 21	**Receive Interrupt Enable**—This control bit allows the SSI to issue receiver related interrupts to the ARM 9 platform. Refer to section 1.6.9.1 “Receive Interrupt Enable Bit Description” for a detailed description of this bit.	0 = SSI Receiver Interrupt requests disabled. 1 = SSI Receiver Interrupt requests enabled.
TDMAE Bit 20	**Transmit DMA Enable**—This bit allows SSI to request for DMA transfers. When enabled, DMA requests are generated when any of the TFE0/1 bits in the SISR are set and if the corresponding TFEN bit is also set. If the corresponding FIFO is disabled, a DMA request is generated when the corresponding TDE bit is set.	0 = SSI DMA requests disabled. 1 = SSI DMA requests enabled.
TIE Bit 19	**Transmit Interrupt Enable**—This control bit allows the SSI to issue transmitter data related interrupts to the ARM 9 platform. Refer to section 24.4.9.2 "Transmit Interrupt Enable Bit Description" for a detailed description of this bit.	0 = SSI Transmitter Interrupt requests disabled. 1 = SSI Transmitter Interrupt requests enabled.
Enable Bit Bits 18–0	**Enable Bit**—Each bit controls whether the corresponding status bit in SISR can be issue an interrupt to the ARM 9 platform or not.	0 = Status bit cannot issue interrupt. 1 = Status bit can issue interrupt.

24.4.9.1 Receive Interrupt Enable Bit Description

When the RIE and RE bit are set the program controller is interrupted when either of the SSI Receive FIFO Full (RFF0/1) bits in SISR is set (if the corresponding Receive FIFO is enabled). If Receive FIFO is not enabled, interrupt is generated when the corresponding SSI Receive Data Ready (RDR0/1) bit in the SISR is set. When the receive FIFO is enabled, maximum 8 values are available to be read (8 values per channel in Two Channel mode). If not enabled, then one value can be read from the SRX register (one each in case of Two Channel mode). If the RIE bit is cleared, these interrupt are disabled. However, the RFF0/1 and RDR0/1 bits still indicate the receive data register full condition. Reading the SRX registers clears the RDR bits, thus clearing the pending interrupt. Two receive data interrupts (two per channel in case of Two Channel mode) are available: receive data with exception status, and receive data without exception. Table 24-9 and Table 24-10 show the conditions under which these interrupts are generated.

Table 24-9. SSI Receive Data 1 Interrupts

Interrupt	RIE	ROE1	RFF1/RDR1
Receive Data 1 with Exception Status	1	1	1
Receive Data 1 (without exception)	1	0	1

Table 24-10. SSI Receive Data 0 Interrupts

Interrupt	RIE	ROE0	RFF0/RDR0
Receive Data 0 with Exception Status	1	1	1
Receive Data 0 (without exception)	1	0	1

Table 24-11. Clock Pin Configuration

SYN	RXDIR	TXDIR	RFDIR	TFDIR	SRFS	STFS	SRCK	STCK
Asynchronous Mode								
0	0	0	0	0	RFS in	TFS in	RCK in	TCK in
0	0	1	0	1	RFS in	TFS out	RCK in	TCK out
0	1	0	1	0	RFS out	TFS in	RCK out	TCK in
0	1	1	1	1	RFS out	TFS out	RCK out	TCK out
Synchronous Mode								
1	0	0	x	0	–	FS in	–	CK in
1	0	1	x	1	–	FS out	–	CK out
1	1	0	x	x	–	–	–	Gated in
1	1	1	x	x	–	–	–	Gated out

24.4.9.2 Transmit Interrupt Enable Bit Description

The SSI Transmit Interrupt Enable (TIE) control bit allows interrupting the program controller for data transfer requirements of the SSI transmitter. When the TIE and TE bits are set, the program controller is

interrupted when either of the SSI Transmit FIFO Empty (TFE0/1) flags in SISR are set (if corresponding Transmit FIFO is enabled). If the corresponding Transmit FIFO is not enabled, interrupt is generated when the corresponding SSI Transmit Data Register Empty (TDE0/1) flag in the SISR is set and Transmit Enable (TE) bit is set.

When Transmit FIFO 0 is enabled, 8 values can be written to the SSI (8 per channel in case of Two Channel mode, using Tx FIFO 1). If not enabled, then one value can be written to the STX0 register (one per channel in case of Two Channel mode, using STX1). When the TIE bit is cleared, all transmit interrupts are disabled. However, the TDE0/1 bits always indicate the corresponding STX register empty condition, even when the transmitter is disabled by the Transmit Enable (TE) bit (in the SCR). Writing data to the STX clears the corresponding TDE bit, thus clearing the interrupt. Two transmit data interrupts are available (four in case of Two Channel mode, two per channel): transmit data with exception status and transmit data without exceptions. Table 24-12 and Table 24-13 show the conditions under which these interrupts are generated.

Table 24-12. SSI Transmit Data 1 Interrupts

Interrupt	TIE	TUE1	TFE1/TDE1
Transmit Data 1 with Exception Status	1	1	1
Transmit Data 1 (without exception)	1	0	1

Table 24-13. SSI Transmit Data 0 Interrupts

Interrupt	TIE	TUE0	TFE0/TDE0
Transmit Data 0 with Exception Status	1	1	1
Transmit Data 0 (without exception)	1	0	1

24.4.10 SSI Transmit Configuration Register (STCR)

The SSI Transmit Configuration Register (STCR) is a read/write control registers used to direct the transmit operation of the SSI. STCR controls the direction of the bit clock and frame sync ports, STCK and STFS. Interrupt enable bit for the transmit sections is provided in this control register. The Power-On Reset ($\overline{POR}$) clears all STCR bits. However, SSI reset does not affect the STCR bits. The STCR bits are described in the following paragraphs. See Section 24.4 on page -5 for the programming model of the STCR. As with all on-chip peripheral interrupts for i.MX21, the Control / Status Register (STCR) must first be set to enable maskable interrupts. Next, the SSI interrupt bit in the Interrupt Enable Register (SIER) must be set to enable the interrupt. Finally the interrupt can be enabled from within the SSI.

STCR1 SSI1 Transmit Configuration Register 0x1001001C
STCR2 SSI2 Transmit Configuration Register 0x1001101C

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							TXBIT0	TFEN1	TFEN0	TFDIR	TXDIR	TSHFD	TSCKP	TFSI	TFSL	TEFS
TYPE	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 24-14. STCR Register Description

Name	Description	Settings
Reserved Bits 31–10	Reserved—These bits are reserved and should not be used.	
TXBIT0 Bit 9	**Transmit Bit 0**—This control bit allows SSI to transmit the data word from bit position 0 or 23 in the transmit shift register. The shifting data direction can be MSB or LSB first, controlled by the TSHLD bit.	0 = Shifting with respect to bit 23 of transmit shift register. 1 = Shifting with respect to bit 0 of transmit shift register.
TFEN1 Bit 8	**Transmit FIFO Enable 1**—This bit enables transmit FIFO 1. When enabled, the FIFO allows 8 samples to be transmitted by the SSI (per channel) (a 9th sample can be shifting out) before TDE1 bit is set. When the FIFO is disabled, an interrupt is generated when a single sample is transferred to the transmit shift register (provided the interrupt is enabled).	0 = Transmit FIFO 1 disabled. 1 = Transmit FIFO 1 enabled.
TFEN0 Bit 7	**Transmit FIFO Enable 0**—This bit enables transmit FIFO 0. When enabled, the FIFO allows 8 samples to be transmitted by the SSI (per channel) (a 9th sample can be shifting out) before TDE0 bit is set. When the FIFO is disabled, an interrupt is generated when a single sample is transferred to the transmit shift register (provided the interrupt is enabled).	0 = Transmit FIFO 0 disabled. 1 = Transmit FIFO 0 enabled.
TFDIR Bit 6	**Transmit Frame Direction**—This bit controls the direction and source of the transmit frame sync signal. Internally generated frame sync signal is sent out through the STFS port and external frame sync is taken from the same port.	0 = Frame Sync is external. 1 = Frame Sync generated internally.
TXDIR Bit 5	**Transmit Direction**—This bit controls the direction and source of the clock signal used to clock the TXSR. Internally generated clock is output through the STCK port. External clock is taken from this port. Refer to Table 24-11 for details of clock port configuration.	0 = Transmit Clock is external. 1 = Transmit Clock generated internally.
TSHFD Bit 4	**Transmit Shift Direction**—This bit controls whether the MSB or LSB will be transmitted first in a sample. **Note:** The codec device labels the MSB as bit 0, whereas i.MX21 labels the LSB as bit 0. Therefore, when using a standard codec, i.MX21 MSB (codec LSB) is shifted in first (TSHFD cleared).	0 = Data transmitted MSB first. 1 = Data transmitted LSB first.
TSCKP Bit 3	**Transmit Clock Polarity**—This bit controls which bit clock edge is used to latch in data for the transmit section.	0 = Data clocked out on rising edge of bit clock. 1 = Data clocked out on falling edge of bit clock.
TFSI Bit 2	**Transmit Frame Sync Invert**—This bit controls the active state of the frame sync I/O signal for the transmit section of SSI.	0 = Transmit frame sync is active high. 1 = Transmit frame sync is active low.
TFSL Bit 1	**Transmit Frame Sync Length**—This bit controls the length of the frame sync signal to be generated or recognized for the transmit section. The length of a word-long frame sync is same as the length of the data word selected by WL[3:0].	0 = Transmit frame sync is one-word long. 1 = Transmit frame sync is one-clock-bit long.
TEFS Bit 0	**Transmit Early Frame Sync**—These bit controls when the frame sync is initiated for the transmit section. The frame sync signal is deasserted after one bit-for-bit length frame sync and after one word-for-word length frame sync. In case of synchronous operation, the frame sync can also be initiated on receiving the first bit of data.	0 = Transmit frame sync initiated as the first bit of data is transmitted. 1 = Transmit frame sync is initiated one bit before the data is transmitted.

24.4.11 SSI Receive Configuration Register (SRCR)

The SSI Receive Configuration Register (SRCR) is a read/write control registers used to direct the receive operation of the SSI. SRCR controls the direction of the bit clock and frame sync ports, SRCK and SRFS. Interrupt enable bit for the transmit sections is provided in this control register. The Power-On Reset ($\overline{POR}$) clears all SRCR bits. However, a SSI reset does not affect the SRCR bits.

SRCR1 SSI1 Receive Configuration Register 0x10010020
SRCR2 SSI2 Receive Configuration Register 0x10011020

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							RXBIT0	RFEN1	RFEN0	RFDIR	RXDIR	RSHFD	RSCKP	RFSI	RFSL	REFS
TYPE	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 24-15. SRCR Register Description

Name	Description	Settings
Reserved Bits 31–10	Reserved—These bits are reserved and should not be used.	
RXBIT0 Bit 9	**Receive Bit 0**—This control bit allows SSI to receive the data word at bit position 0 or 23 in the receive shift register. The shifting data direction can be MSB or LSB first, controlled by the RSHLD bit.	0 = Shifting with respect to bit 23 of receive shift register. 1 = Shifting with respect to bit 0 of receive shift register.
RFEN1 Bit 8	**Receive FIFO Enable 1**—This bit enables receive FIFO 1. When enabled, the FIFO allows 8 samples to be received by the SSI (per channel) (a 9th sample can be shifting in) before RDR1 bit is set. When the FIFO is disabled, an interrupt is generated when a single sample is received by the SSI (provided the interrupt is enabled).	0 = Receive FIFO 1 disabled. 1 = Receive FIFO 1 enabled.
RFEN0 Bit 7	**Receive FIFO Enable 0**—This bit enables receive FIFO 0. When enabled, the FIFO allows 8 samples to be received by the SSI (per channel) (a 9th sample can be shifting in) before RDR0 bit is set. When the FIFO is disabled, an interrupt is generated when a single sample is received by the SSI (provided the interrupt is enabled).	0 = Receive FIFO 0 disabled. 1 = Receive FIFO 0 enabled.
RFDIR Bit 6	**Receive Frame Direction**—This bit controls the direction and source of the receive frame sync signal. Internally generated frame sync signal is sent out through the SRFS port and external frame sync is taken from the same port.	0 = Frame Sync is external. 1 = Frame Sync generated internally.
RXDIR Bit 5	**Receive Direction**—This bit controls the direction and source of the clock signal used to clock the RXSR. Internally generated clock is output through the SRCK port. External clock is taken from this port. Refer to Table 24-11 for details of clock port configuration.	0 = Receive Clock is external. 1 = Receive Clock generated internally.
RSHFD Bit 4	**Receive Shift Direction**—This bit controls whether the MSB or LSB will be received first in a sample. **Note:** The codec device labels the MSB as bit 0, whereas i.MX21 labels the LSB as bit 0. Therefore, when using a standard codec, i.MX21 MSB (codec LSB) is shifted in first (RSHFD cleared).	0 = Data received MSB first. 1 = Data received LSB first.

Table 24-15. SRCR Register Description (continued)

Name	Description	Settings
RSCKP Bit 3	**Receive Clock Polarity**—This bit controls which bit clock edge is used to latch in data for the receive section.	0 = Data latched on falling edge of bit clock. 1 = Data latched on rising edge of bit clock.
RFSI Bit 2	**Receive Frame Sync Invert**—This bit controls the active state of the frame sync I/O signal for the receive section of SSI.	0 = Receive frame sync is active high. 1 = Receive frame sync is active low.
RFSL Bit 1	**Receive Frame Sync Length**—This bit controls the length of the frame sync signal to be generated or recognized for the receive section. The length of a word-long frame sync is same as the length of the data word selected by WL[3:0].	0 = Receive frame sync is one-word long. 1 = Receive frame sync is one-clock-bit long.
REFS Bit 0	**Receive Early Frame Sync**—These bit controls when the frame sync is initiated for the receive section. The frame sync is disabled after one bit-for-bit length frame sync and after one word-for-word length frame sync.	0 = Receive frame sync initiated as the first bit of data is received. 1 = Receive frame sync is initiated one bit before the data is received.

24.4.12 SSI Transmit and Receive Clock Control Registers (STCCR and SRCCR)

The SSI Transmit and Receive Control (STCCR and SRCCR) registers are 18-bit, read/write control registers used to direct the operation of the SSI. These registers control the SSI clock generator bit and frame sync rates, word length, and number of words per frame for the serial data. The STCCR register is dedicated to the transmit section, and the SRCCR register is dedicated to the receive section except in Synchronous mode, in which the STCCR register controls both the receive and transmit sections. Power-On Reset ($\overline{\text{POR}}$) clears all STCCR and SRCCR bits. SSI reset does not affect the STCCR and SRCCR bits. The control bits are described in the following paragraphs. Although the bit patterns of the SRCCR and STCCR registers are the same, the contents of these two registers can be programmed differently. See below for the programming models of the STCCR and SRCCR registers.

STCCR1 SSI1 Transmit Clock Control Register 0x10010024
STCCR2 SSI2 Transmit Clock Control Register 0x10011024

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
														DIV2	PSR	WL3
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WL2	WL1	WL0	DC4	DC3	DC2	DC1	DC0	PM7	PM6	PM5	PM4	PM3	PM2	PM1	PM0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRCCR1	SSI1 Receive Clock Control Register												0x10010028			
SRCCR2	SSI2 Receive Clock Control Register												0x10011028			
BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
														DIV2	PSR	WL3
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WL2	WL1	WL0	DC4	DC3	DC2	DC1	DC0	PM7	PM6	PM5	PM4	PM3	PM2	PM1	PM0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 24-16. STCCR/SRCCR Register Description

Name	Description	Settings
Reserved Bits 31–19	Reserved—These bits are reserved and should not be used.	
DIV2 Bit 18	**Divide By 2**—This bit controls a divide-by-two divider in series with the rest of the pre-scalers.	0 = Divider bypassed. 1 = Divider used to divide clock by 2.
PSR Bit 17	**Prescaler Range**—This bit controls a fixed divide-by-eight prescaler in series with the variable prescaler. It extends the range of the prescaler for those cases where a slower bit clock is required.	0 = Prescaler bypassed. 1 = Prescaler used to divide clock by 8.
WL Bits 16–13	**Word Length Control**—These bits are used to control the length of the data words being transferred by the SSI. These bits control the Word Length Divider in the Clock Generator. They also control the frame sync pulse length when the FSL bit is cleared. In I2S Master mode, the SSI works with a fixed word length of 32, and the WL bits are used to control the amount of valid data in those 32 bits.	Refer to Table 24-17 for details regarding settings.
DC Bits 12–8	**Frame Rate Divider Control**—These bits are used to control the divide ratio for the programmable frame rate dividers. The divide ratio works on the word clock. In Normal mode, this ratio determines the word transfer rate. In Network mode, this ration sets the number of words per frame. The divide ratio ranges from 1 to 32 in Normal mode and from 2 to 32 in Network mode. In Normal mode, a divide ratio of 1 (DC=00000) provides continuous periodic data word transfer. A bit-length frame sync must be used in this case.	These bits can be programmed with values ranging from “00000” to “11111” to control the number of words in a frame.
PM Bits 7–0	**Prescaler Modulus Select**—These bits control the prescale divider in the clock generator. This prescaler is used only in Internal Clock mode to divide the internal clock (SYS_CLK). The bit clock output is available at the clock port.	A divide ratio from 1 to 256 (PM[7:0] = $00 to $FF) can be selected. Refer to Section 24.4.12.1, “Prescale Modulus Select Bit Description,” for details regarding settings.

Table 24-17. SSI Data Word Lengths

WL3	WL2	WL1	WL0	Number of bits/word	Supported in Implementation
0	0	0	0	2	No
0	0	0	1	4	No
0	0	1	0	6	No
0	0	1	1	8	Yes
0	1	0	0	10	Yes
0	1	0	1	12	Yes
0	1	1	0	14	No
0	1	1	1	16	Yes
1	0	0	0	18	Yes
1	0	0	1	20	Yes
1	0	1	0	22	Yes
1	0	1	1	24	Yes
1	1	0	0	26	No
1	1	0	1	28	No
1	1	1	0	30	No
1	1	1	1	32	No

24.4.12.1 Prescale Modulus Select Bit Description

The bit clock frequency can be calculated from the SSI Serial System Clock (SYS_CLK), using the equation in Equation 24-1.

NOTE

All three (DIV2, PSR and PM) should not be set to zero at the same time.

Eqn. 24-1

$$f_{INT\_BIT\_CLK} = f_{SYS\_CLK} / [(DIV2 + 1) \times (7 \times PSR + 1) \times (PM + 1) \times 2]$$

where PM=PM[7:0]

$$f_{FRAME\_SYN\_CLK} = (f_{INT\_BIT\_CLK}) / [(DC + 1) \times WL]$$

where DC = DC[4:0] and WL = 8, 10, 12, 16, 18, 20, 22, 24

For example, if the SSI working clock SYS_CLK (ccm_ssi_clk) is 19.2 MHz, in 8-bit word Normal mode with DC[4:0] set to 1 (00001), PM[7:0] set to 74 (0100 1010), the PSR bit cleared, DIV2 bit set to 1, a bit clock rate of 19.2 MHz /[1 x 4 x 75] = 64 kHz is generated. Since the 8-bit word rate is equal to two, the sampling rate (FS rate) would then be 64 kHz / [2 x 8] = 4 kHz.

In next example, the SYS_CLK clock is 12 MHz. A16-bit word Network mode with DC[4:0] set to 1 (00001), PM[7:0] set to 1(0000 0001), the PSR bit is set to 0, DIV2 bit set to 1, and a 12 MHz SYS_CLK clock, a bit clock rate of 12 MHz ³ [1 × 4 × 2] = 1.5MHz is generated. Since the 16-bit word rate is equal to two, the sampling rate (FS rate) would be 1.5 MHz / [2 × 16] = 46.875 kHz.

Table 24-18 shows the example to program the PSR and PM bits to generate the different bit clock (STCK) frequencies.

Table 24-18. SSI Bit Clock, Frame Clock as a Function of PSR/PM in Normal Mode

SYS_CLK clock (MHz)	PSR (0 or 1)	PM[7:0]	Actual Bit_Clk Frequency (kHz) STCK, SRCK	Ideal Bit_Clk Frequency (kHz)
6.4	1	24($18)	8.0	8.0
9.6	1	24($18)	12.0	12.0
4.8	0	74($4A)	16.0	16.0
19.2	0	74($4A)	64.0	64.0
19.2	0	37($25)	126.3	128.0
48.0	0	46($2E)	255.3	256.0
96.0	0	46($2E)	510.6	512.0
96.0	0	23($0F)	1000.0	1024
96.0	0	11($0B)	2000.0	2048
96.0	0	5($05)	4000.0	4096

Table 24-19 and Table 24-20 show an example of programming the clock controller divider ratio to generate the SYS_CLK and BIT_CLK frequencies that are close to the ideal sampling rates. In these examples, the master mode is selected either by setting I2S master bit (SCR[6:5]=01) or individually programming the SSI in network, synchronous, transmit internal mode (the table specifically illustrates the I2S mode frequencies/sample rates). The SYS_CLK clock is ccm_ssi_clk. The fixed I2S frame rate of 64 bits per frame (word length (WL) can be any value) and DC of 1 are assumed.

Table 24-19. SSI Sys Clock, Bit Clock, Frame Clock in Master Mode (Table 1)

Ideal Sampling Rate (kHz)	Over sampling Rate	SP DPLL Freq (MHz)	SSIDIV (in PLL Clock controller)	Ideal MCLK Freq (MHz)	Actual MCLK Freq (MHz) SRCK	DIV2	PSR	PM	Actual Bit_Clk Freq (kHz) STCK	Actual Frame Rate (kHz) STFS	Error (Hz)
44.10	384	288.00	17	16.9344	16.9411	0	0	2	2823.53	44.117	17.65
22.05	384	288.00	17	16.9344	16.9411	0	0	5	1411.87	22.058	8.82
11.025	384	288.00	17	16.9344	16.9411	0	0	11	705.83	11.029	4.41

Table 24-20. SSI Sys Clock, Bit Clock, Frame Clock in Master Mode (Table 2)

Ideal Sampling Rate (kHz)	Over sampling Rate	MCU PLL Freq (MHz)	SSIDIV (in PLL Clock controller)	Ideal MCLK Freq (MHz)	Actual MCLK Freq (MHz) SRCK	DIV2	PSR	PM	Actual Bit_Clk Freq (kHz) STCK	Actual Frame Rate (kHz) STFS	Error (Hz)
48.00	256	258.048	21	12.288	12.288	0	0	1	3072.00	48.000	0

24.4.13 SSI FIFO Control/Status Register (SFCSR)

SFCSR1 SSI1 FIFO Control/Status Register 0x1001002C
SFCSR2 SSI2 FIFO Control/Status Register 0x1001102C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFCNT1				TFCNT1				RFWM1				TFWM1			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFCNT0				TFCNT0				RFWM0				TFWM0			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1

Table 24-21. SFCSR Register Description

Name	Description	Settings
RFCNT1 Bits 31–28	**Receive FIFO Counter 1**—These bits indicate the number of data words in Receive FIFO 1.	Refer to Table 24-22 for details regarding settings.
TFCNT1 Bits 27–24	**Transmit FIFO Counter 1**—These bits indicate the number of data words in Transmit FIFO 1.	Refer to Table 24-23 for details regarding settings.
RFWM1 Bits 23–20	**Receive FIFO Full WaterMark 1**—These bits control the threshold at which the RFF1 flag will be set. The RFF1 flag is set whenever the data level in Rx FIFO 1 reaches the selected threshold.	Refer to Table 24-24 for details regarding settings.
TFWM1 Bit 19–16	**Transmit FIFO Empty WaterMark 1**—These bits control the threshold at which the TFE1 flag will be set. The TFE1 flag is set whenever the data level in Tx FIFO 1 falls below the selected threshold.	Refer to Table 24-25 for details regarding settings.
RFCNT0 Bits 15–12	**Receive FIFO Counter 0**—These bits indicate the number of data words in Receive FIFO 0.	Refer to Table 24-22 for details regarding settings.
TFCNT0 Bits 11–8	**Transmit FIFO Counter 0**—These bits indicate the number of data words in Transmit FIFO 0.	Refer to Table 24-23 for details regarding settings.

Table 24-21. SFCSR Register Description (continued)

Name	Description	Settings
RFWM0 Bits 7–4	**Receive FIFO Full WaterMark 0**—These bits control the threshold at which the RFF0 flag will be set. The RFF0 flag is set whenever the data level in Rx FIFO 0 reaches the selected threshold.	Refer to Table 24-24 for details regarding settings.
TFWM0 Bit 3–0	**Transmit FIFO Empty WaterMark 0**—These bits control the threshold at which the TFE0 flag will be set. The TFE0 flag is set whenever the data level in Tx FIFO 0 falls below the selected threshold.	Refer to Table 24-25 for details regarding settings.

Table 24-22. Receive FIFO Counter Bit Description

Bits	Description
0000	0 data word in receive FIFO
0001	1 data word in receive FIFO
0010	2 data word in receive FIFO
0011	3 data word in receive FIFO
0100	4 data word in receive FIFO
0101	5 data word in receive FIFO
0110	6 data word in receive FIFO
0111	7 data word in receive FIFO
1000	8 data word in receive FIFO

Table 24-23. Transmit FIFO Counter Bit Description

Bits	Description
0000	0 data word in TxFIFO
0001	1 data word in TxFIFO
0010	2 data word in TxFIFO
0011	3 data word in TxFIFO
0100	4 data word in TxFIFO
0101	5 data word in TxFIFO
0110	6 data word in TxFIFO
0111	7 data word in TxFIFO
1000	8 data word in TxFIFO

Table 24-24. Receive FIFO WaterMark Bit Description

Bits	Description
0000	Reserved
0001	RFF set when at least one data word have been written to the Receive FIFO Set when RxFIFO = 1,2,3,4,5,6,7,8 data words.
0010	RFF set when more than or equal to 2 data word have been written to the Receive FIFO. Set when RxFIFO = 2,3,4,5,6,7,8 data words.
0011	RFF set when more than or equal to 3 data word have been written to the Receive FIFO. Set when RxFIFO = 3,4,5,6,7,8 data words.
0100	RFF set when more than or equal to 4 data word have been written to the Receive FIFO. Set when RxFIFO = 4,5,6,7,8 data words.
0101	RFF set when more than or equal to 5 data word have been written to the Receive FIFO. Set when RxFIFO = 5,6,7,8 data words.
0110	RFF set when more than or equal to 6 data word have been written to the Receive. Set when RxFIFO = 6,7,8 data words.
0111	RFF set when more than or equal to 7 data word have been written to the Receive FIFO. Set when RxFIFO = 7,8 data words.
1000	RFF set when 8 data word have been written to the Receive FIFO (default). Set when RxFIFO = 8 data words.

Table 24-25. Transmit FIFO WaterMark Bit Description

Bits	Description
0000	Reserved
0001	TFE set when there are more than or equal to 1 empty slots in Transmit FIFO. (default) Transmit FIFO empty is set when TxFIFO ≤ 7 data.
0010	TFE set when there are more than or equal to 2 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO ≤ 6 data.
0011	TFE set when there are more than or equal to 3 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO ≤ 5 data.
0100	TFE set when there are more than or equal to 4 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO ≤ 4 data.
0101	TFE set when there are more than or equal to 5 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO ≤ 3 data.
0110	TFE set when there are more than or equal to 6 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO ≤ 2 data.
0111	TFE set when there are more than or equal to 7 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO ≤ 1 data.
1000	TFE set when there are 8 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO=0 data.

The following table indicates the status of the Transmit FIFO Empty flag, with different settings of the Transmit FIFO WaterMark bits and varying amounts of data in the Tx FIFO.

Table 24-26. Status of Transmit FIFO Empty Flag

Transmit FIFO Watermark (TFWM)	Number of data in TXFIFO								
	0	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1	0
2	1	1	1	1	1	1	1	0	0
3	1	1	1	1	1	1	0	0	0
4	1	1	1	1	1	0	0	0	0
5	1	1	1	1	0	0	0	0	0
6	1	1	1	0	0	0	0	0	0
7	1	1	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0

24.4.14 SSI Test Register (STR)

STR1 SSI1 Test Register 0x10010030
STR2 SSI2 Test Register 0x10011030

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEST	RCK2TCK	RFS2TFS	RXSTATE					TXD2RXD	TCK2RCK	TFS2RFS	TXSTATE				
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1

Table 24-27. STR Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should not be used.	
TEST Bit 15	**Test Mode**—This bit enables the test features of SSI. When set, the RXSTATE and TXSTATE bit values get transferred to the state machine. When in test mode, the user can read the contents of a specific FIFO by writing an appropriate value to the RFCNT0/1 or TFCNT 0/1 bits.	0 = No effect on SSI operation. 1 = SSI in Test Mode.
RCK2TCK Bit 14	**Receive Clock to Transmit Clock Loop Back**—This bit connects SRCK (used as output) to STCK (used as input).	0 = No effect on SSI operation. 1 = SRCK to STCK loop back enabled.
RFS2TFS Bit 13	**Receive Frame to Transmit Frame Loop Back**—This bit connects SRFS (used as output) to STFS (used as input).	0 = No effect on SSI operation. 1 = SRFS to STFS loop back enabled.
RXSTATE Bits 12–8	**Receiver State Machine Status**—These bits indicate the current status of the Receive State Machine.	Status bits indicating current status of receiver state machine.
TXD2RXD Bit 7	**Transmit Data to Receive Data Loop Back**—This bit connects STXD to SRXD.	0 = No effect on SSI operation. 1 = STXD to SRXD loop back enabled.

Table 24-27. STR Register Description (continued)

Name	Description	Settings
TCK2RCK Bit 6	**Transmit Clock to Receive Clock Loop Back**—This bit connects STCK (used as output) to SRCK (used as input).	0 = No Frame Synchronization. 1 = STCK to SRCK loop back enabled.
TFS2RFS Bit 5	**Transmit Frame to Receive Frame Loop Back**—This bit connects STFS (used as output) to SRFS (used as input).	0 = No Frame Synchronization. 1 = STFS to SRFS loop back enabled.
TXSTATE Bits 4–0	**Transmitter State Machine Status**—These bits indicate the current status of the Transmit State Machine.	Status bits indicating current status of receiver state machine.

24.4.15 SSI Option Register (SOR)

SOR1 SSI1 Option Register 0x10010034
SOR2 SSI2 Option Register 0x10011034

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										CLKOFF	RX_CLR	TX_CLR	INIT	WAIT		SYNRST
TYPE	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 24-28. SOR Register Description

Name	Description	Settings
Reserved Bits 31–7	Reserved—These bits are reserved and should not be used.	
CLKOFF Bit 6	**Clock Off**—This bit is used to turn off the IP Interface working clock of SSI to further reduce power consumption. However, the functionality of clock gating in i.MX21 is controlled by PLL Clock Controller (PCCR0 Register) and not by the CLKOFF bit.	0 = No effect on SSI operation. 1 = Turn off IP Working Clock.
RX_CLR Bit 5	**Receiver Clear**—This bit flushes the contents of Rx FIFOs. It is always read as cleared (0).	0 = No effect on SSI operation. 1 = Flush Rx FIFOs.
TX_CLR Bit 4	**Transmitter Clear**—This bit flushes the contents of Tx FIFOs. It is always read as cleared (0).	0 = No effect on SSI operation. 1 = Flush Tx FIFOs.
INIT Bit 3	**Initialize**—The setting of this bit causes the SSI state machine to reset.	0 = No effect on SSI operation. 1 = Initialize SSI state machine.
WAIT Bits 2–1	**Wait**—These bits control the number wait states to be added to all transactions between the ARM 9 platform and SSI.	The value of these bits ranges from 00 (no wait states) to 11 (three wait states).
SYNRST Bit 0	**Frame Syn Reset**—This bit automatically resets the accumulation of data in Receive Data Registers (SRX0/1) and Receive FIFOs (RXFIFO 0/1) on the next frame synchronization	0 = Data accumulation not affected. 1 = Reset data accumulation on Frame Synchronization.

24.4.16 SSI AC97 Control Register (SACNT)

SACNT1 SSI1 AC97 Control Register 0x10010038
SACNT2 SSI2 AC97 Control Register 0x10011038

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						FRDIV						WR	RD	TIF	FV	A97EN
TYPE	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 24-29. SACNT Register Description

Name	Description	Settings
Reserved Bits 31–11	Reserved—These bits are reserved and should not be used.	
FRDIV Bits 10–5	**Frame Rate Divider**—These bits control the frequency of AC97 data transmission/reception. They are programmed with the number of frames for which the SSI should be idle, after operating in one frame. Through these bits, AC97 frequency of operation, from 48 KHz (000000) to 1 KHz (101111) can be achieved.	Sample Value: 001010 (10 Decimal) = SSI will operate once every 11 frames.
WR Bit 4	**Write Command**—This bit specifies whether the next frame will carry an AC97 Write Command or not. The programmer should take care that only one of the bits (WR or RD) is set at a time. When this bit is set, the corresponding tag bits (corresponding to Command Address and Command Data slots of the next Tx frame) are automatically set. This bit is automatically cleared by the SSI after completing transmission of a frame.	0 = Next frame will not have a Write Command. 1 = Next frame will have a Write Command.
RD Bit 3	**Read Command**—This bit specifies whether the next frame will carry an AC97 Read Command or not. The programmer should take care that only one of the bits (WR or RD) is set at a time. When this bit is set, the corresponding tag bit (corresponding to Command Address slot of the next Tx frame) is automatically set. This bit is automatically cleared by the SSI after completing transmission of a frame.	0 = Next frame will not have a Read Command. 1 = Next frame will have a Read Command.
TIF Bit 2	**Tag in FIFO**—This bit controls the destination of the information received in AC97 tag slot (Slot #0).	0 = Tag info stored in SATAG register. 1 = Tag info stored in Rx FIFO 0.
FV Bit 1	**Fixed/Variable Operation**—This bit selects whether the SSI is in AC97 Fixed mode or AC97 Variable mode. Fix mode only supports 48kHz sampling rate.	0 = AC97 Fixed Mode. 1 = AC97 Variable Mode.
AC97EN Bit 0	**AC97 Mode Enable**—This bit is used to enable SSI AC97 operation. Refer to section 24.10 "AC97 Operation" for details of AC97 operation.	0 = AC97 mode disabled. 1 = SSI in AC97 mode.

24.4.17 SSI AC97 Command Address Register (SACADD)

SACADD1	SSI1 AC97 Command Address Register	0x1001003C
SACADD2	SSI2 AC97 Command Address Register	0x1001103C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
														SACADD18:16]		
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SACADD															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 24-30. SACADD Register Description

Name	Description	Settings
Reserved Bits 31–19	Reserved—These bits are reserved and should not be used.	
SACADD Bits 18–0	**AC97 Command Address**—These bits store the Command Address Slot information (bit 19 of the slot is sent in accordance with the Read and Write Command bits in SACNT register). Writing to this register (by the ARM 9 platform) sets the value for the Slot #1 data to be transmitted. On a read, the ARM 9 platform gets the last received Slot #1 value. If a different value is received from the AC97 codec (different than the previously received value), the CMDDU bit in SISR is set.	These bits reflect the Command Address information.

24.4.18 SSI AC97 Command Data Register (SACDAT)

SACDAT1	SSI1 AC97 Command Data Register	0x10010040
SACDAT2	SSI2 AC97 Command Data Register	0x10011040

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													SACDAT			
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SACDAT															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 24-31. SACDAT Register Description

Name	Description	Settings
Reserved Bits 31–20	Reserved—These bits are reserved and should not be used.	
SACDAT Bits 19–0	**AC97 Command Data**—The outgoing Command Data Slot carries the information contained in these bits. Writing to this register (by the ARM 9 platform) sets the value for the Slot #2 data to be transmitted. On a read, the ARM 9 platform gets the last received Slot #2 value. If a different value is received from the AC97 codec (different than the previously received value), the CMDAU bit in SISR is set. In case of an AC97 Read Command, these bits are cleared for transmission in Time Slot #2 of AC97 frame.	These bits reflect the Command Data information.

24.4.19 SSI AC97 Tag Register (SATAG)

SATAG1 SSI1 AC97 Tag Register 0x10010044
SATAG2 SSI2 AC97 Tag Register 0x10011044

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SATAG															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 24-32. SATAG Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should not be used.	
SATAG Bits 15–0	**AC97 Tag**—These bits store the tag information if the tag is not stored in the FIFO 0—that is, when TIF bit in SACNT register is cleared. Writing to this register (by the ARM 9 platform) sets the value of the Tx Tag (in AC97 fixed mode). On a read, the ARM 9 platform gets the last Rx Tag value. It is updated at the start of each received frame. The contents of this register are also used to generate the transmit tag in AC97 Variable mode of operation. In case the received tag value changes, the RXT bit in SISR register is set, if enabled.	These bits reflect the received tag information.

24.4.20 SSI Transmit Time Slot Mask Register (STMSK)

STMSK1	SSI1 Transmit Time Slot Mask Register	0x10010048
STMSK2	SSI2 Transmit Time Slot Mask Register	0x10011048

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STMSK															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STMSK															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 24-33. STMSK Register Description

Name	Description	Settings
STMSK Bits 31–0	**Receive Mask**—These bits indicate which slot has been masked in the current frame. The ARM 9 platform can write to this register to control the time slots in which the SSI transmits data. Each bit has info corresponding to the respective time slot in the frame. If a change is made to the register contents, the transmission pattern is updated from the next frame.	For each bit: 0 = Valid Time Slot. 1 = Time Slot masked (no data transmitted in this time slot).

24.4.21 SSI Receive Time Slot Mask Register (SRMSK)

SRMSK1	SSI1 Receive Time Slot Mask Register	0x1001004C
SRMSK2	SSI2 Receive Time Slot Mask Register	0x1001104C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SRMSK															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRMSK															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 24-34. SRMSK Register Description

Name	Description	Settings
SRMSK Bits 31–0	**Receive Mask**—These bits indicate which slot has been masked in the current frame. The ARM 9 platform can write to this register to control the time slots in which the SSI receives data. Each bit has info corresponding to the respective time slot in the frame. If a change is made to the register contents, the reception pattern is updated from the next frame.	For each bit: 0 = Valid Time Slot. 1 = Time Slot masked (no data received in this time slot).

24.5 SSI Port Configuration

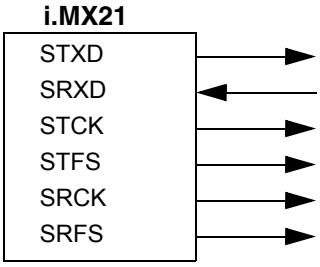
The six SSI ports can be connected to various i.MX21 chip pins by programming the Digital Audio Mux (AUDMUX) in the appropriate manner. Please refer to the AUDMUX specification for details regarding this programming.

24.6 SSI Date and Control Ports

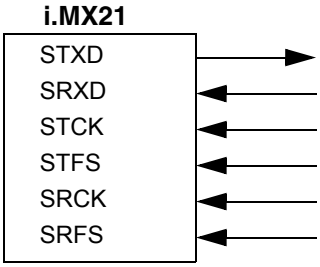
The SSI has the following six ports:

- Serial Transmit Data (STXD)
- Serial Receive Data (SRXD)
- Serial Transmit Clock (STCK)
- Serial Transmit Frame Sync (STFS)
- Serial Receive Clock (SRCK)
- Serial Receive Frame Sync (SRFS)

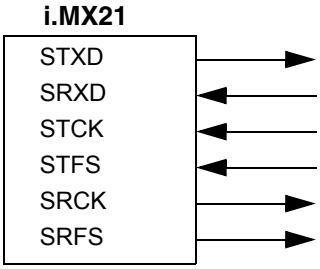
Figure 24-13 and Figure 24-14 show the main SSI configurations. These ports support all transmit and receive functions with continuous or gated clock as shown. Note that gated clock implementations do not require the use of the frame sync ports (STFS and SRFS).



SSI Internal Continuous Clock for TX/RX (RXDIR=1,TXDIR=1,RFDIR=1,TFDIR=1,SYN=0)

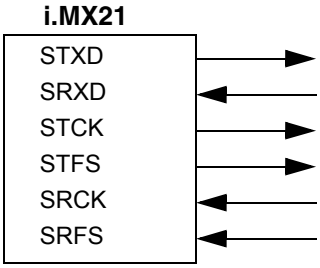


SSI External Continuous Clock for TX/RX (RXDIR=0,TXDIR=0,RFDIR=0,TFDIR=0,SYN=0)



SSI Internal Continuous Clock for RX (RXDIR=1, TXDIR=0,RFDIR=1,TFDIR=0, SYN=0)

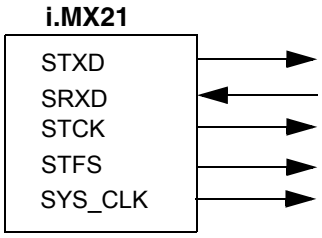
SSI External Continuous Clock for TX



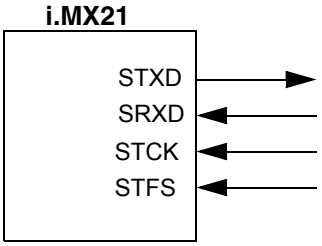
SSI Internal Continuous Clock for TX (RXDIR=0, TXDIR=1, RFDIR=0, TFDIR=1, SYN=0)

SSI External Continuous Clock for RX

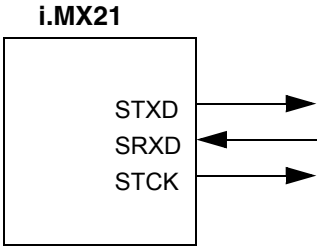
Figure 24-13. Asynchronous (SYN=0) SSI Configurations—Continuous Clock



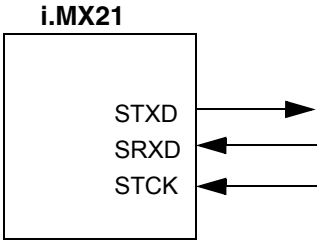
SSI Internal Continuous Clock (RXDIR=0, TXDIR=1, RFDIR=X, TFDIR=1, SYN=1, SYS_CLK_EN = 1)
SSI I2S Master Mode(I2S_Mode=01, SYS_CLK_EN)



SSI External Continuous Clock (RXDIR=0, TXDIR=0, RFDIR=X, TFDIR=0,SYN=1)
SSI I2S Slave Mode(I2S_Mode=10)



SSI Internal Gated Clock (RXDIR=1, TXDIR=1, SYN=1)



SSI External Gated Clock (RXDIR=1, TXDIR=0, SYN=1)

Figure 24-14. Synchronous SSI Configurations—Continuous and Gated Clock

The following paragraphs describe the configuration of the SSI ports.

- STXD (Serial Transmit Data)—The STXD port transmits data from the Serial Transmit Shift Register. The STXD port is an output port when data is being transmitted and is disabled between data word transmissions and on the trailing edge of the bit clock after the last bit of a word is transmitted.

- SRXD (Serial Receive Data)—The SRXD port brings serial data into the Receive Data Shift Register.
- STCK (Serial Transmit Clock)—The STCK port can be used as either an input or an output. This clock signal is used by the transmitter and can be either continuous or gated. During Gated Clock mode, data on the STCK port is valid only during the transmission of data, otherwise it is pulled to the inactive state. In Synchronous mode, this port is used by both the transmit and receive sections.
- STFS (Serial Transmit Frame Sync)—The STFS port can be used as either an input or an output. The frame sync is used by the transmitter to synchronize the transfer of data. The frame sync signal can be one bit or one word in length and can occur one bit before the transfer of data or right at the transfer of data. In Synchronous mode, this port is used by both the transmit and receive sections. In Gated Clock mode, frame sync signals are not used. If STFS is configured as input, the external device should drive STFS during rising edge of STCK or SRCK.
- SRCK (SYS_CLK) (Serial Receive Clock)—The SRCK port can be used as either an input or an output. This clock signal is used by the receiver and is always continuous. During Gated Clock mode, the STCK port is used instead for clocking in data. In SSI master mode, this port is used as an output port for the oversampling clock, SYS_CLK. In I2S master mode, this port can be used to output the oversampling clock, ccm_ssi_clk.
- SRFS (Serial Receive Frame Sync)—The SRFS port can be used as either an input or an output. The frame sync is used by the receiver to synchronize the transfer of data. The frame sync signal can be one bit or one word in length and can occur one bit before the transfer of data or right at the transfer of data. If SRFS is configured as input, the external device should drive SRFS during rising edge of STCK or SRCK.

An example of the port signals for an 8-bit data transfer is shown in Figure 24-15. Continuous and gated clock signals are shown, as well as the bit-length frame sync signal and the word-length frame sync signal. Note that the shift direction can be defined as MSB first or LSB first, and that there are other options on the clock and frame sync.

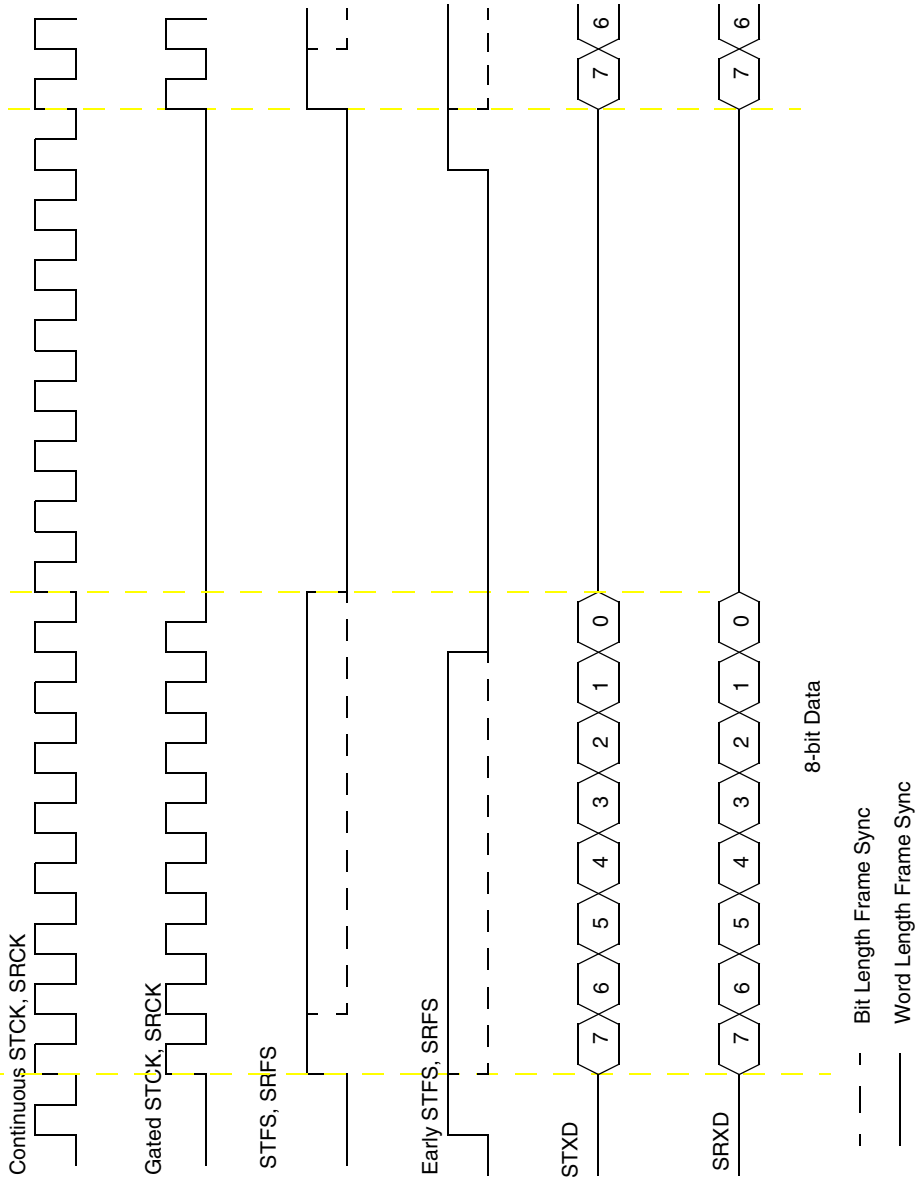


Figure 24-15. Serial Clock and Frame Sync Timing

24.7 SSI Operating Modes

The SSI has three basic operating modes, with the option of asynchronous or synchronous protocol, as listed in the following:

- Normal mode
  - Asynchronous protocol
  - Synchronous protocol
- Network mode
  - Asynchronous protocol
  - Synchronous protocol

- Gated Clock mode
  - Synchronous protocol only

These modes can be programmed by several bits in the SSI control registers. Table 24-35 lists these operating modes and some of the typical applications in which they can be used.

Table 24-35. SSI Operating Modes

TX, RX Sections	Serial Clock	Mode	Typical Application
Asynchronous	Continuous	Normal	Multiple synchronous codecs
Asynchronous	Continuous	Network	TDM codec or DSP networks
Synchronous	Continuous	Normal	Multiple synchronous codecs
Synchronous	Continuous	Network	TDM codec or DSP network
Synchronous	Gated	Normal	SPI-type devices; DSP to MCU

The transmit and receive sections of the SSI can be synchronous or asynchronous. In Synchronous mode, the transmitter and the receiver use a common clock and frame synchronization signal. In Asynchronous mode, the transmitter and receiver each has its own clock and frame synchronization signals. Continuous or Gated Clock mode can be selected. In Continuous mode, the clock runs continuously. In Gated Clock mode, the clock is only functioning during transmission.

Normal or Network mode can also be selected. In Normal mode, the SSI functions with one data word of I/O per frame. In Network mode, any number from two to thirty-two data words of I/O per frame can be used. Network mode is typically used in star or ring time division multiplex networks with other processors or codecs, allowing interface to time division multiplexed networks without additional logic. Use of the gated clock is not allowed in Network mode. These distinctions result in the basic operating modes that allow the SSI to communicate with a wide variety of devices.

The SSI supports both Normal and Network modes, and these can be selected independently of whether the transmitter and receiver are synchronous or asynchronous. Typically these protocols are used in a periodic manner, where data is transferred at regular intervals, such as at the sampling rate of an external codec. Both modes use the concept of a frame. The beginning of the frame is marked with a frame sync when programmed with continuous clock. The frame sync occurs at a periodic interval. The length of the frame is determined by the DC[4:0] bits in either the SRCCR or STCCR register, depending on whether data is being transferred or received. The number of words transferred per frame depends on the mode of the SSI.

In Normal mode, one data word is transferred per frame. In Network mode, the frame is divided into anywhere between two and thirty-two time slots, where in each time slot one data word can optionally be transferred.

In (non-I2S) slave modes (external frame sync), SSI's programmed word length setting should be equal to the word length setting of the master. In I2S slave mode, SSI's programmed word length setting can be lesser than or equal to the word length setting of the I2S master (external codec).

In slave modes, SSI's programmed frame length setting (DC bits) can be lesser than or equal to the frame length setting of the master (external codec).

24.7.1 Normal Mode

Normal mode is the simplest mode of the SSI. It is used to transfer data in one time slot per frame. A time slot is a unit of data and the WL[3:0] bits define the number of bits in a time slot. In Continuous Clock mode, a frame sync occurs at the beginning of each frame. The length of the frame is determined by the following factors:

- The period of the Serial Bit Clock (PSR, PM[7:0] bits for internal clock or the frequency of the external clock on the STCK port)
- The number of bits per time slot (WL[3:0] bits)
- The number of time slots per frame (DC[4:0] bits)

If Normal mode is configured to provide more than one time slot per frame, data is transmitted only in the first time slot. No data is transmitted in subsequent time slots. In Normal mode, DC[4:0] values corresponding to more than a single time slot in a frame, only result in lengthening the frame. Data transfer only takes place during the first time slot of the frame.

24.7.1.1 Normal Mode Transmit

The conditions for data transmission from the SSI in Normal mode are:

1. SSI enabled (SSIEN = 1)
2. Enable FIFO and configure Transmit and Receive Watermark if FIFO is used.
3. Write data to Transmit Data Register (STX)
4. Transmitter enabled (TE = 1)
5. Frame sync active (for continuous clock case)
6. Bit clock begins (for gated clock case)

When the above conditions occur in Normal mode, the next data word is transferred into the Transmit Shift Register (TXSR) from the Transmit Data Register 0 (STX0), or from the Transmit FIFO 0 Register, if transmit FIFO 0 is enabled. The new data word is transmitted immediately.

If transmit FIFO 0 is not enabled and the transmit data register empty (TDE0) bit is set, transmit interrupt 0 occurs if the transmit interrupt enable (TIE) and TDE0_EN bits are set.

If transmit FIFO 0 is enabled and the Transmit FIFO Empty (TFE0) bit is set, transmit interrupt 0 occurs if the transmit interrupt enable (TIE) and TFE0_EN bits are set. If transmit FIFO 0 is enabled and filled with data, 8 data words can be transferred before the core must write new data to the STX0 register.

The STXD port is disabled except during the data transmission period. For a continuous clock, the optional frame sync output and clock outputs are not disabled, even if both receiver and transmitter are disabled.

24.7.1.2 Normal Mode Receive

The conditions for data reception from the SSI are:

1. SSI enabled (SSIEN = 1)
2. Receiver enabled (RE = 1)
3. Frame sync active (for continuous clock case)

4. Bit clock begins (for gated clock case)

With the above conditions in Normal mode with a continuous clock, each time the frame sync signal is generated (or detected) a data word is clocked in. With the above conditions and a gated clock, each time the clock begins, a data word is clocked in.

If receive FIFO 0 is not enabled, the received data word is transferred from the Receive Shift Register (RXSR) to the Receive Data Register 0 (SRX0), the Receive Data Ready 0 (RDR0) flag is set. Receive Interrupt 0 occurs if RIE and RDR0_EN bits are set.

If receive FIFO 0 is enabled, the received data word is transferred to the Receive FIFO 0. The Receive FIFO Full 0 (RFF0) flag is set if the Receive Data Register (SRX0) is full and Receive FIFO 0 reaches the selected threshold. Receive Interrupt 0 occurs if Receive Interrupt Enable (RIE) and RFF0_EN bits are set.

The ARM 9 platform program must read the data from the Receive Data Register 0 (SRX0) before a new data word is transferred from the Receive Shift Register (RXSR), otherwise the Receive Overrun Error 0 (ROE0) bit is set. If receive FIFO 0 is enabled, the Receive Overrun Error 0 (ROE0) bit is set when the Receive FIFO 0 data level reaches the selected threshold and a new data word is ready to be transferred to the Receive FIFO 0.

Figure 24-16 shows transmitter and receiver timing for an 8-bit word with two words per time slot in Normal mode, continuous clock with a late word length frame sync.

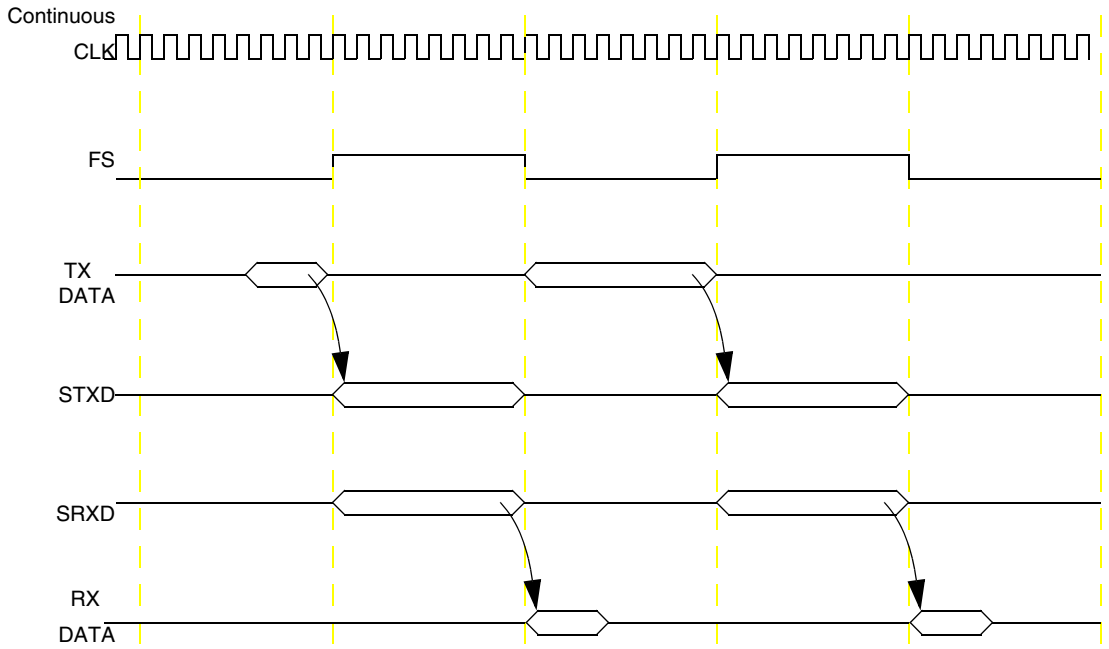


Figure 24-16. Normal Mode Timing—Continuous Clock

Figure 24-17 shows a similar case for gated clock. Note that a pull-down resistor is required in the gated clock case because the clock port is disabled between transmissions.

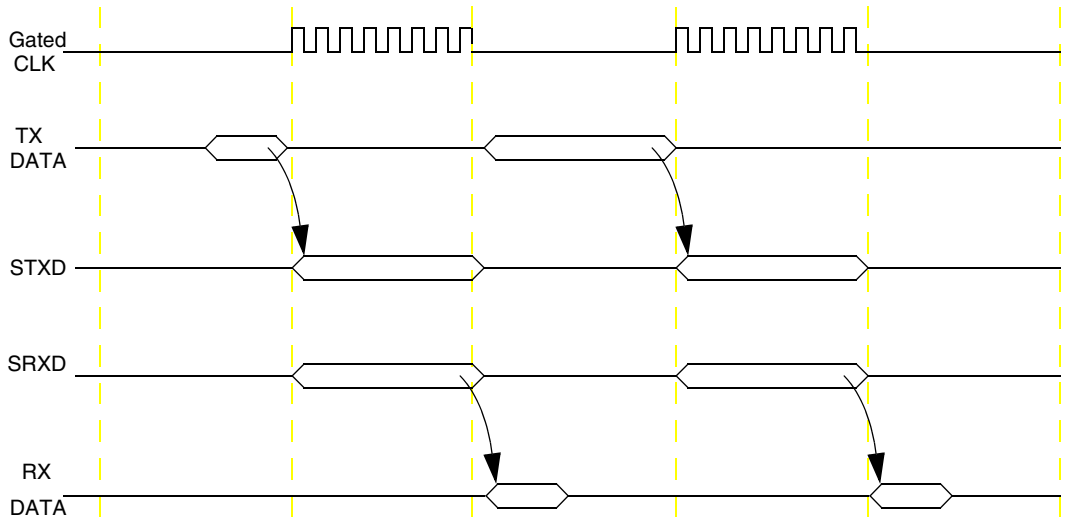


Figure 24-17. Normal Mode Timing—Gated Clock

24.7.2 Network Mode

Network mode is used for creating a Time Division Multiplexed (TDM) network, such as a TDM codec network or a network of DSPs. In Continuous Clock mode, a frame sync occurs at the beginning of each frame. In this mode, the frame is divided into more than one time slot. During each time slot, one data word can be transferred. Each time slot is then assigned to an appropriate codec or DSP on the network. The DSP can be a master device that controls its own private network, or a slave device that is connected to an existing TDM network and occupies a few time slots.

In the Two Channel mode of operation, the second set of Transmit and Receive FIFOs and Data Registers are used to create two separate channels. These channels are completely independent, with a their own set of ARM 9 platform interrupts and DMA requests, which are identical to the ones available for the default channel. In this mode, data is transmitted/received in enabled time slots alternately from/to FIFO 0 and FIFO 1, starting from FIFO 0. The first data word is taken from FIFO 0 and transmitted in the first enabled time slot and subsequently, data is loaded from FIFO 1 and FIFO 0 alternately and transmitted. Similarly, the first received data is sent to FIFO 0 and subsequent data is sent to FIFO 1 and FIFO 0 alternately. Time slots can be selected through the Transmit and Receive Time Slot Mask registers (STMSK and SRMSK). For using this mode of operation, the TCH_EN bit (SCR[8]) needs to be set.

The frame sync signal indicates the beginning of a new data frame. Each data frame is divided into time slots and transmission and/or reception of one data word can occur in each time slot (rather than in just the frame sync time slot as in Normal mode). The frame rate dividers, controlled by the DC[4:0] bits, select two to thirty-two time slots per frame. The length of the frame is determined by the following factors:

- The period of the serial bit clock (PSR, PM[7:0] bits for internal clock, or the frequency of the external clock on the STCK port)
- The number of bits per sample (WL[3:0] bits)
- The number of time slots per frame (DC[4:0] bits)

In Network mode, data can be transmitted in any time slot. The distinction of the Network mode is that each time slot is identified with respect to the frame sync (data word time). This time slot identification allows the option of transmitting data during the time slot by writing to the STX registers or ignoring the time slot by writing to STMSK. The receiver is treated in the same manner and received data is only transferred to the receive data register/fifo if the corresponding time slot is enabled (through SRMSK).

By utilizing the STMSK and SRMSK registers, software only must service the SSI during valid time slots. This eliminates any overhead associated with unused time slots.

24.7.2.1 Network Mode Transmit

The transmit portion of SSI is enabled when the SSIEN and the TE bits in the SCR are both set. However, for continuous clock, when the TE bit is set, the transmitter is enabled only after detection of a new time slot (transmission starts from the next frame boundary).

Normal start-up sequence for transmission is to do the following:

1. Write the data to be transmitted to the STX register. This clears the TDE flag.
2. Set the TE bit to enable the transmitter on the next word boundary (for continuous clock case).
3. Enable transmit interrupts.

Alternatively, the programmer may decide not to transmit in a time slot by writing to the STMSK. The TDE flag is not cleared, but the STXD port remains disabled during the time slot. When the frame sync is detected or generated (continuous clock), the first enabled data word is transferred from the STX register to the TXSR and is shifted out (transmitted). When the STX register is empty, the TDE bit is set, which causes a transmitter interrupt to be sent if the TIE bit is set. Software can poll the TDE bit or use interrupts to reload the STX register with new data for the next time slot. Failing to reload the STX register before the TXSR is finished shifting (empty) causes a transmitter underrun, the TUE error bit to be set, and the STXD port is disabled for the next time slot.

The operation of clearing the TE bit disables the transmitter after completion of transmission of the current data word. Setting the TE bit enables transmission of the next word. During that time the STXD port is disabled. The TE bit should be cleared after the TDE bit is set to ensure that all pending data is transmitted.

To summarize, the Network mode transmitter generates interrupts every enabled time slot and requires the DSP program to respond to each enabled time slot. These responses may be one of the following:

- Write data in data register to enable transmission in the next time slot.
- Write in the time slot register to disable transmission in the next time slot (unless time slot is already masked by STMSK register bit).
- Do nothing—transmit underrun occurs at the beginning of the next time slot and the previous data is re-transmitted.

In the Two Channel Network mode of operation, both the channels (Data Registers, FIFOs, Interrupts and DMA requests) operate in the same manner, as described above. The only difference in case of the second channel is that the Interrupts related to this channel are generated only in case this mode of operation is selected (TDE1 is low by default).

24.7.2.2 Network Mode Receive

The receiver portion of the SSI is enabled when both the SSIEN and the RE bits in the SCR are set. However, the receive enable only takes place during that time slot if RE is enabled before the second to last bit of the word. If the RE bit is cleared, the receiver is disabled immediately. SSI is capable of finding the start of the next frame automatically. When the word is completely received, it is transferred to the SRX register, which sets the RDR bit (Receive Data Ready). Setting the RDR bit causes a receive interrupt to occur if the receiver interrupt is enabled (the RIE bit is set). The second data word (second time slot in the frame), begins shifting in immediately after the transfer of the first data word to the SRX register. The DSP program must read the data from the Receive Data Register (which clears RDR) before the second data word is completely received (ready to transfer to RX data register) or a receive overrun error occurs (the ROE bit is set).

An interrupt can occur after the reception of each enabled data word or the programmer can poll the RDR flag. The ARM9 platform program response can be one of the following:

- Read RX and use the data.
- Read RX and ignore the data.
- Do nothing—the receiver overrun exception occurs at the end of the current time slot.

NOTE

For a continuous clock, the optional frame sync output and clock output signals are not affected, even if the transmitter or receiver is disabled. TE and RE do not disable the bit clock or the frame sync generation. The only way to disable the bit clock and the frame sync generation is to disable the SSIEN bit in the SCR.

In the Two Channel Network mode of operation, both the channels (Data Registers, FIFOs, Interrupts and DMA requests) operate in the same manner, as described above. The only difference in case of the second channel is that the Interrupts related to this channel are generated only in case this mode of operation is selected.

The transmitter and receiver timing for an 8-bit word with continuous clock, FIFO disabled, three words per frame sync in Network mode is shown in Figure 24-18.

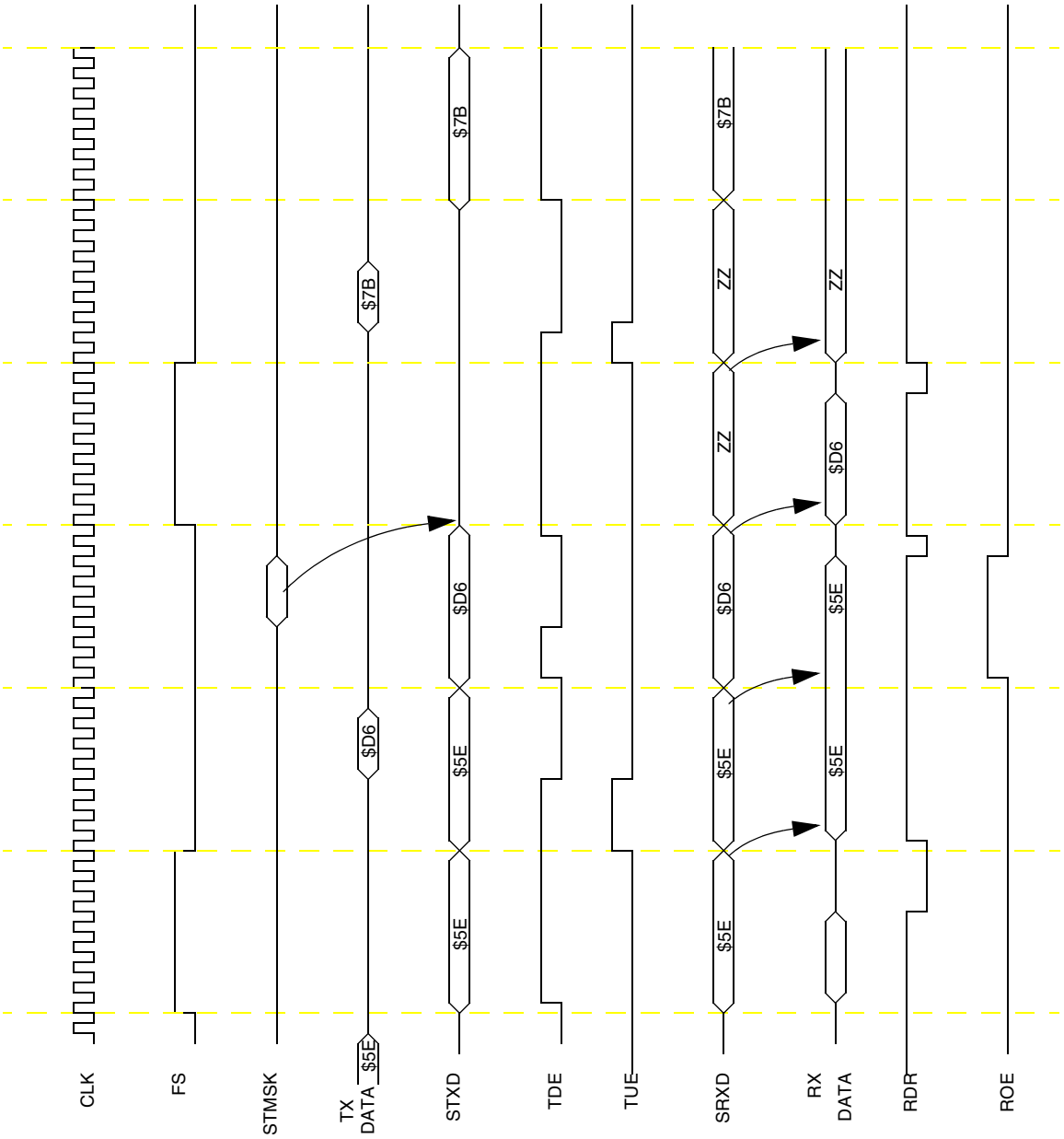


Figure 24-18. Network Mode Timing—Continuous Clock

24.8 Gated Clock Operation

Gated Clock mode is often used to hook up to SPI-type interfaces on Microcontroller Units (MCUs) or external peripheral chips. In Gated Clock mode, the presence of the clock indicates that valid data is on the STXD or SRXD ports. For this reason, no frame sync is needed in this mode. Once transmission of data has completed, the clock is pulled to the inactive state. Gated clocks are allowed for both the transmit and receive sections with either internal or external clock and in Normal mode. Gated clocks are not allowed in Network mode. Refer to Table 24-11 for SSI configuration for Gated mode operation.

The clock runs when the TE bit and/or the RE bit are appropriately enabled. For the case of internally generated clock, all internal bit clocks, word clocks, and frame clocks continue to operate. When a valid time slot occurs (such as the first time slot in Normal mode), the internal bit clock is enabled onto the appropriate clock port. This allows data to be transferred out in periodic intervals in Gated Clock mode. With an external clock, the SSI waits for a clock signal to be received. Once the clock begins, valid data is shifted in. Care should be taken to clear all DC bits (0x00000) when SSI is used in Gated mode.

For Gated clock operated in external clock mode, a proper clock signalling must be apply to the SSI STCK for it to function properly. If the SSI uses rising edge transition to clock data (TSCKP=0) and falling edge transition to latch data (RSCKP=0), the clock must be in an active low state when idle. If the SSI uses falling edge transition to clock data (TSCKP=1) and rising edge transition to latch data (RSCKP=1), the clock must be in a active high state when idle. The following diagrams illustrate the different edge clocking/latching.

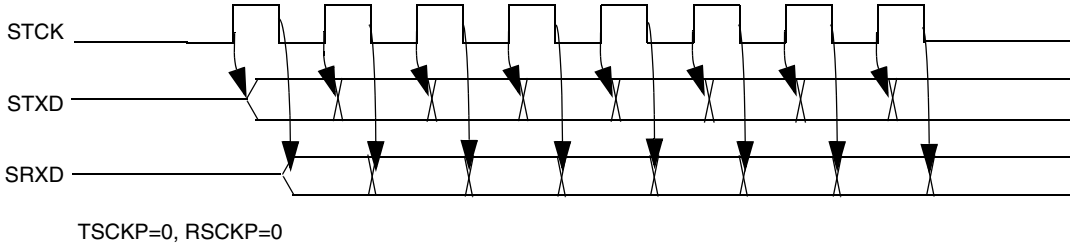


Figure 24-19. Rising Edge Clocking and Falling Edge Latching

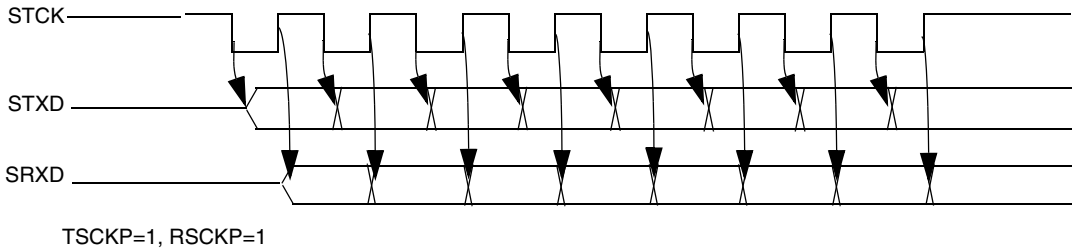


Figure 24-20. Falling Edge Clocking and Rising Edge Latching

NOTE

The bit clock ports must be kept free of timing glitches. If a single glitch occurs, all ensuing transfers will be out of synchronization.

24.9 I²S Mode Operation

Table 24-36. I²S Mode Selection

I2S_MODE[1]	I2S_MODE[0]	Remark
0	0	Normal mode
0	1	I2S master mode
1	0	I2S slave mode
1	1	Normal Mode

In normal mode operation, all register bits are not forced to any particular state internally and user can program the SSI to work in any operating condition.

When I2S modes are entered (I2S master (01) or I2S slave (10)), the following settings are recommended:

- Sync mode (SCR[4] =1)
- Tx shift direction: MSB transmitted first (STCR[4]=0)
- Rx shift direction: MSB received first (SRCR[4]=0)
- Tx data clocked at falling edge of the clock (STCR[3]=1)
- Rx data latched at rising edge of the clock (SRCR[3]=1)
- Tx frame sync active low (STCR[2]=1)
- Rx frame sync active low (SRCR[2]=1)
- Tx frame sync initiated one bit before data is transmitted (STCR[0]=1)
- Rx frame sync initiated one bit before data is received (SRCR[0]=1)

In I2S master mode(SCR[6:5]=01), the following additional settings are recommended:

TXDIR bit (STCR[5]) set to 1 to select internal generated bit clock

TFDIR bit (STCR[6]) set to 1 to select internal generated frame sync

In I2S master mode(SCR[6:5]=01), the following settings are done automatically by the hardware internally:

Network mode is selected (SCR[3]=1)

Tx frame sync length set to one-word-long-frame (STCR[1]=0)

Rx frame sync length set to one-word-long-frame (SRCR[1]=0)

Tx shifting w.r.t. bit 0 of TXSR (STCR[9]=1)

Rx shifting w.r.t. bit 0 of RXSR (SRCR[9]=1)

Further control bits needed to set by user to configure the bit clock and frame sync are the PM (STCCR[7:0]), PSR (STCCR[17]), DIV2(STCCR[18], WL (STCCR[16:13]) and DC (STCCR[12:8]) bits. The word length is fixed to 32 in I2S Master mode and the WL bits determine the number of bits that will contain valid data (out of the 32 transmitted/received bits in each channel). The fixing of word

duration as 32 simplifies the relation between oversampling clock (ccm_ssi_clk) and Frame Sync (ccm_ssi_clk becomes an integer multiple of Frame Sync).

In I2S slave mode(SCR[6:5]=10), the following additional settings are recommended:

TXDIR bit(STCR[5]) set to 0 to select external generated bit clock

TFDIR bit(STCR[6]) set to 0 to select external generated frame sync

In I2S slave mode(SCR[6:5]=10), the following settings are done automatically by the hardware internally:

Normal mode is selected (SCR[3]=0)

Tx frame sync length set to one-bit-long-frame (STCR[1]=1)

Rx frame sync length set to one-bit-long-frame (SRCR[1]=1)

Tx shifting w.r.t. bit 0 of TXSR (STCR[9]=1)

Rx shifting w.r.t. bit 0 of RXSR (SRCR[9]=1)

Further control bits needed to set by user to configure the data transmission are the WL (STCCR[16:13]) and DC (STCCR[12:8]) bits. The word length is variable in I2S slave mode and the WL bits determine the number of bits that will contain valid data. The actual word length is determined by the external codec. The external I2S Master still sends frame sync according to the I2S protocol (early, word wide and active low), the SSI internally operates so that each frame sync transition is the start of a new frame (the WL bits determine the number of bits to be transmitted/received). After one data word has been transferred, the SSI waits for the next frame sync transition to start operation in the next time slot.

24.10 AC97 Operation

In AC97 mode of operation, the SSI transmits a 16-bit Tag Slot at the start of a frame and the rest of the slots (in that frame) are all 20-bits wide. The same sequence is followed while receiving data. Refer to the AC97 specification for details regarding transmit and receive sequences and data formats.

When the AC97 mode is entered, the following settings are automatically configured by the hardware internally. The register contents do not reflect the changes anymore. The register bits within the bracket are the equivalent setting.

- Sync mode is entered (SCR[4] =1)
- Network mode is selected (SCR[3]=1)
- Tx shift direction is MSB transmitted first (STCR[4]=0)
- Rx shift direction is MSB received first (SRCR[4]=0)
- Tx data is clocked at rising edge of the clock (STCR[3]=0)
- Rx data is latched at falling edge of the clock (SRCR[3]=0)
- Tx frame sync is active high (STCR[2]=0)
- Rx frame sync is active high (SRCR[2]=0)
- Tx frame sync length is one-word-long-frame (STCR[1]=0)

- Rx frame sync length is one-word-long-frame (SRCR[1]=0)
- Tx frame sync initiated one bit before data is transmitted (STCR[0]=1)
- Rx frame sync initiated one bit before data is transmitted (SRCR[0]=1)
- Tx FIFO is enabled (STCR[7]=1)
- Rx FIFO is enabled (SRCR[7]=1)
- TFDIR bit (STCR[6]) is forced to 1 internally to select internal generated frame sync
- TXDIR bit(STCR[5]) is forced to 0 internally to select external generated bit clock

Any alternation of these bits individually will not affect the operational conditions of the SSI unless AC97 mode is deselected.

Hence the only control bits needed to set by user to configure the data transmission/reception are the WL (STCCR[16:13]) and DC (STCCR[12:8]) bits. In AC97 mode, the WL bits can only legally take the values corresponding to 16-bit (truncated data) or 20-bit time slots. In case WL bits are set to select 16-bit time slots, the SSI pads the transmit data (four least significant bits) with zeros and while receiving, stores only the most significant 16 bits in the Rx FIFO.

The following sequence should be followed for programming the SSI to work in AC97 mode:

- Program the WL bits to a value corresponding to either 16 or 20 bits
- Select the number of time slots through DC bits
- Write data to be transmitted, in Tx FIFO 0 (through Tx Data Register 0)
- Program the FV, TIF, RD, WR and FRDIV bits in SACNT register
- Update the contents of SACADD, SACDAT and SATAG (for Fixed mode only) registers
- Enable the AC97 mode of operation (AC97EN bit in SACNT register)

Once the SSI starts transmitting and receiving data (after being configured in AC97 mode), the programmer needs to service the interrupts, as and when they are raised (updates to command address/data or tag registers, reading of received data and writing more data for transmission). Further details regarding fixed and variable mode implementation are provided in the following sections.

While using AC97 in Two-Channel Mode (TCH_EN=1), it is recommended that the received tag is not stored in the Rx FIFO (TIF=0). In case the programmer needs to update the SATAG register and also issue a RD/WR command (in a single frame), it is recommended that the SATAG register be updated prior to issuing a RD/WR command.

24.10.1 AC97 Fixed Mode Operation (SACNT[1]=0)

The AC97 fixed mode operation only supports 48kHz sampling rate. During this operation, the SSI transmits at the start of a frame in accordance with the Frame Rate Divider bits. To transmit, select the frame rate (SACNT[10:5]) and at the start of appropriate frames, tag (slot #0), command address (slot #1), command data (slot #2), and two data samples from the Tx FIFO are transmitted (slots #3, #4).

When receiving, the received Tag slot (slot #0) bit 15 is checked to see if the Codec is ready. If this bit is set, the rest of the frame is received and the relevant registers updated. Otherwise, the entire frame is ignored.

24.10.2 AC97 Variable Mode Operation (SACNT[1]=1)

In variable mode operation, the SSI transmits at the start of a frame only if there was a request for data in the previous frame. While transmitting, the AC97 slot request bits received during the previous frame are examined and the appropriate data samples are transmitted from the Tx FIFO, the software does not need to write the Tx Tag information, only data to be transmitted, needs to be written to the appropriate location.

While receiving, if the Codec is ready, the frame is received and the slot request bits are stored for scheduling transmission in the next frame.

24.11 External Frame and Clock Operation

When applying external frame sync and clock signals to SSI, there should be at least 4 clock cycle before the rising edge of frame sync signal. The transition of STFS or SRFS should be synchronized with the rising edge of external clock signal, STCK or SRCK.

24.12 SSI Reset and Initialization Procedure

The SSI is affected by three types of reset:

- Power-on Reset—The Power-on reset ($\overline{\text{POR}}$) is generated by asserting the RESET port. The Power-on reset clears the SSIEN bit in SCR, which disables the SSI. All other status and control bits in the SSI are affected as described in SSI Programming Model in Section 24.4 on page -5.
- SSI Reset—The SSI reset is generated when the SSIEN bit in the SCR is cleared. The SSI status bits are preset to the same state produced by the Power-On Reset ($\overline{\text{POR}}$). The SSI control bits are unaffected. The control bits in the SCR are also unaffected. The SSI reset is useful for selective reset of the SSI without changing the present SSI control bits and without affecting the other peripherals.

The correct sequence to initialize the SSI is as follows:

1. Issue a Power-On Reset ($\overline{\text{POR}}$) or SSI reset
2. Set the SSIEN bit in SCR
3. Set all other control bits, for example TXDIR, RXDIR, and so on
4. Set TE, RE bits in SCR

To ensure proper operation of the SSI, the programmer should use the Power-On Reset ($\overline{\text{POR}}$) or SSI reset before changing any of the following control bits listed in Table 24-37.

NOTE

These control bits should not be changed during SSI operation.

Table 24-37. SSI Control Bits Requiring Reset Before Change

Control Register	Bit
SRCCR STCCR	[16]=WL3 [15]=WL2 [14]=WL1 [13]=WL0
SCR	[9]=CLK_IST [8]=TCH_EN [7]=SYS_CLK_EN [6:5]=I2S_MODE [4]=SYN [3]=NET
SACNT	[0]=AC97EN [1]=FV
SIER	[22]=RDMAE [20]=TDMAE
SRCR STCR	[9]=RXBIT0 [9]=TXBIT0 [8]=RFEN1 [8]=TFEN1 [7]=RFEN0 [7]=TFEN0 [6]=RFDIR [6]=TFDIR [5]=RXDIR [5]=TXDIR [4]=RSHFD [4]=TSHFD [3]=RSCKP [3]=TSCKP [2]=RFSI [2]=TFSI [1]=RFSL [1]=TFSL [0]=REFS [0]=TEFS

Part 6
Peripherals

Chapter 25
CMOS Sensor Interface (CSI)

This chapter presents the CMOS Sensor Interface (CSI) architecture, operation principles, and programming model. The CSI enables the i.MX21 to connect directly to external CMOS image sensors. CMOS image sensors are separated into two classes, dumb and smart. Dumb sensors are those that support only traditional sensor timing (Vertical SYNC and Horizontal SYNC) and output only Bayer and statistics data, while smart sensors support CCIR656 video decoder formats and perform additional processing of the image (for example, image compression, image pre-filtering, and various data output formats).

The capabilities of the CSI include:

- Configurable interface logic to support most commonly available CMOS sensors.
- Support for CCIR656 video interface as well as traditional sensor interface.
- 8-bit data port for YCC, YUV, Bayer, or RGB data input.
- Full control of 8-bit and 16-bit data to 32-bit FIFO packing.
- 32 × 32 FIFO to store received image pixel data that can be read through programmed IO or DMA.
- Direct interface to eMMA Preprocessing block (PrP).
- Single interrupt source to interrupt controller from maskable sensor interrupt sources: Start of Frame, End of Frame, Change of Field, FIFO full.
- Configurable master clock frequency output to sensor.
- Statistic data generation for Auto Exposure (AE) and Auto White Balance (AWB) control of the camera (for Bayer data only).

25.1 CSI Architecture

Figure 25-6 shows the block diagram of the CMOS Sensor Interface. It consists of 2 control registers (Control Register 1 and 3) to set up the interface timing and interrupt generation, a control register (Control Register 2) for statistic data generation, a status register, interface logic, data packing logic, CCIR timing decoder, interrupt controller, master clock generator, statistical data generator, 32 × 32 image data receive FIFO (RxFIFO), and a 16 × 32 statistic data FIFO (StatFIFO).

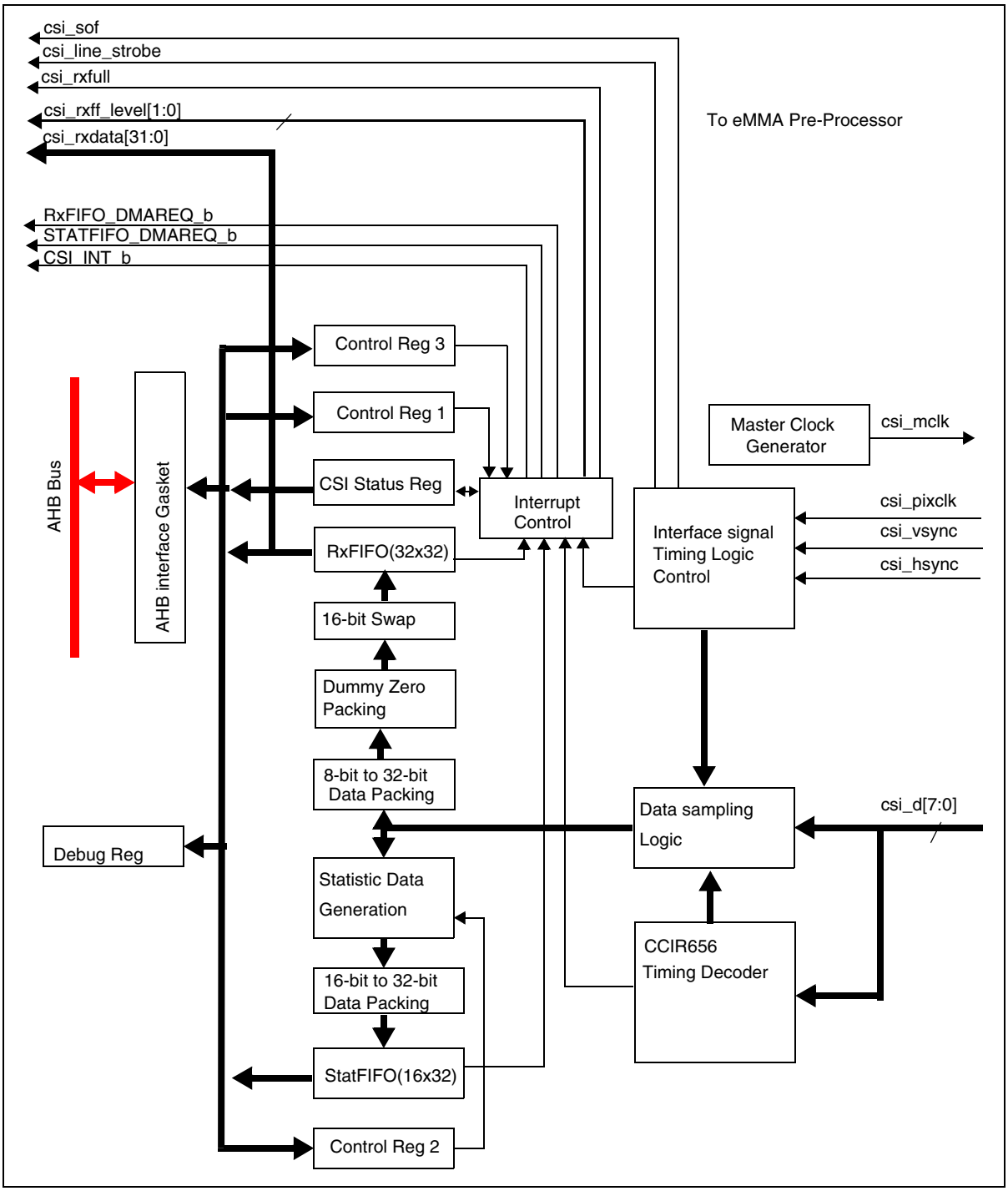


Figure 25-1. CSI Block Diagram

25.2 CSI Interface Signal Description

Table 25-1 provides a listing of the input and output signals between the CSI module of the MC9328MX21 and an external CMOS sensor.

Table 25-1. Signals Between CSI and Sensor

CSI Signals	Direction	Description
CSI_VSYNC	Input	Vertical Sync (Start Of Frame)
CSI_HSYNC	Input	Horizontal Sync (Blank Signal)
CSI_D[7:0]	Input	8-bit Sensor Data Bus (Bayer, YUV, YCrCb, RGB)
CSI_MCLK	Output	Sensor Master Clock
CSI_PIXCLK	Input	Pixel Clock

25.2.1 Signals from CSI to eMMA Pre-Processor Block (PrP)

There is a dedicated bus from CSI RxFIFO to the eMMA Pre-Processor Block (PrP) for fast data transfer.

The bus can be enabled or disabled. When it is enabled, the RxFIFO is detached from the AHB bus and connected to the PrP. Any CPU read or DMA access to the RxFIFO register is ignored. All CSI Interrupts are also masked to prevent software access to the FIFO and status registers.

Users select the RxFIFO full level according to the data format and line width, each of the burst from CSI RxFIFO to PrP must use a size that equal to the RxFIFO full level. To ensure complete transfer of the whole frame, the size of the image (in words) must be integer multiples of the RxFIFO full level. A simple calculation is shown in Table 25-2.

Table 25-2. Integer Multiples of RxFIFO Full Levels

Data Format	Byte Per Pixel	Pixel Per Word	Options for RxFIFO Full Level (Words)	Requirement on Line Width (Pixels)
YUV422	2	2	4 / 8 / 16	Multiple of 8 / 16 / 32
YCC422	2	2		Multiple of 8 / 16 / 32
RGB565	2	2		Multiple of 8 / 16 / 32
RGB888	4	1		Multiples of 4 / 8 / 16
Bayer	1	4		Multiple of 16 / 32 / 64

NOTE

FIFO full level of 24 words is not supported in the CSI-PrP interface. If a 24-word full setting is used the internal logic will regard it as 8 words.

25.3 Principles of Operation

This section describes the modes of operation of the sensor interface.

The CSI is designed to support generic sensor interface timing as well as CCIR656 video interface timing. Traditional CMOS sensors typically use SOF, HSYNC (BLANK), and PIXCLK signals to output Bayer or YUV data. Smart CMOS sensors, that come with on-chip imaging processing, usually support video

mode transfer. They use an embedded timing codec to replace the SOF and BLANK signal. The timing codec is defined by the CCIR656 standard.

25.3.1 Gated Clock Mode

VSYNC, HSYNC, and PIXCLK signals are used in gated clock mode.

Figure 25-2. Gated Clock Mode Timing Diagram

A frame starts with a rising edge on VSYNC, then HSYNC goes to HIGH and holds for the entire line. The Pixel clock is valid as long as HSYNC is HIGH. Data is latched at the rising edge of the valid pixel clocks. HSYNC goes to LOW at the end of line. Pixel clocks then become invalid and CSI stops receiving data from the stream. For the next line the HSYNC timing repeats. For the next frame the VSYNC timing repeats.

25.3.2 Non-Gated Clock Mode

In non-gated clock mode, only the VSYNC and PIXCLK signals are used; the HSYNC signal is ignored.

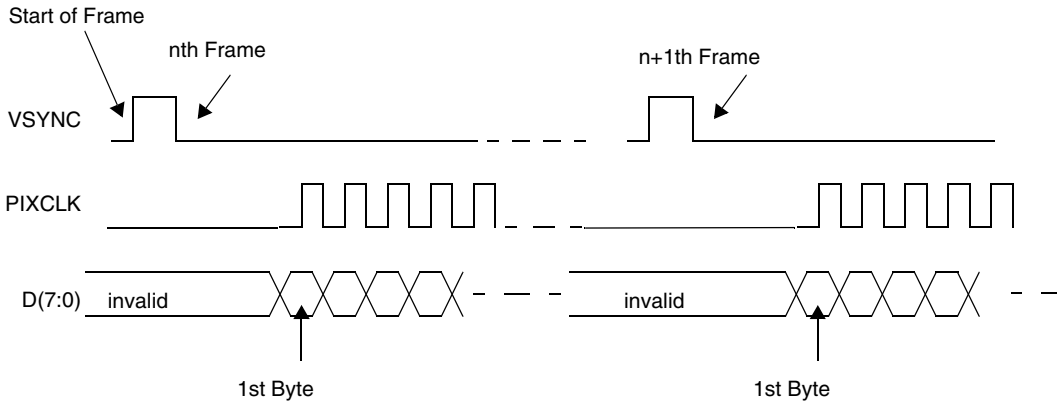


Figure 25-3. Non-Gated Clock Mode Timing Diagram

The overall timing of non-gated mode is the same as the gated-clock mode, except for the HSYNC signal. HSYNC signal is ignored by the CSI. All incoming pixel clocks are valid and cause data to be latched into RxFIFO. The PIXCLK signal is inactive (states low) until valid data is ready to be transmitted over the bus.

Figure 25-3 shows the timing using a typical sensor, other sensors may have the slightly different timing from that shown. The CSI should be programed to support rising / falling-edge triggered VSYNC; active-high / low HSYNC; and rising / falling-edge triggered PIXCLK.

25.3.3 CCIR656 Interlace Mode

In CCIR656 mode, only the PIXCLK and DATA[7:0] signals are used. The start of frame and blank signals are replaced by a timing codec which is embedded in the data stream. Each active line starts with a SAV code and ends with a EAV code. In some cases, digital blanking is inserted in between EAV and SAV code. The CSI decodes and filters out the timing-coding from the data stream, thus recovering VSYNC and HSYNC signals for internal use, such as statistical block control and CSI-to-PrP interconnection. Data is

forwarded to the data receive and packing block in a *sequential* manner without re-ordering—that is, field 1 followed by field 2. The fields must be re-ordered in software to get back the original image.

Change Of Field interrupt (COF) is triggered upon every field change. The interrupt service routine reads the status register to check for the current field.

According to the CCIR656 specification, the image must be in 625/50 PAL or 525/60 NTSC format. In addition, the image is interlaced into odd and even fields, with vertical and horizontal blank data being filled into certain lines. Data must be in YCC422 format, each pixel contains 2 bytes, either Y + Cr or Y + Cb. These requirements are set for TV systems. The CSI module supports PAL and NTSC format only. Figure 25-4 shows the frame structure in PAL system, showing vertical blanking and horizontal blanking.

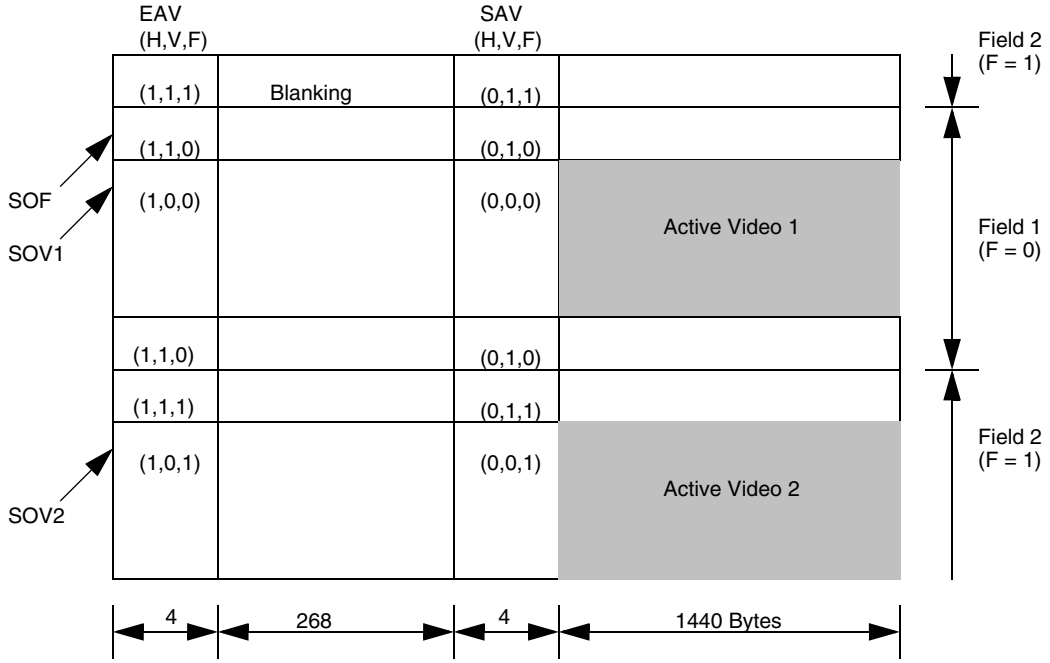


Figure 25-4. CCIR656 Interlace Mode (PAL)

Figure 25-5 shows the general timing for a single line, showing SAV and EAV.

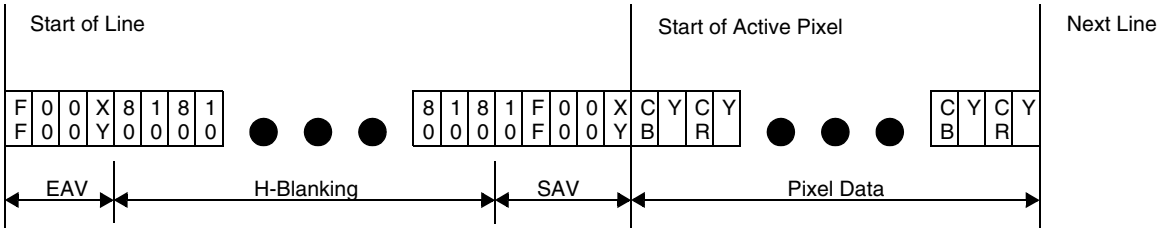


Figure 25-5. CCIR656 General Line Timing

The coding tables recommended by the CCIR656 specification are shown in Table 25-3, Table 25-4 and Table 25-5. It is used in the CCIR656 mode to decode the video stream. An interrupt is generated for SOF, which is decoded from the embedded timing codec.

Table 25-3. Coding for SAV and EAV

Data Bit Number	1st Byte 0xFF	2nd Byte 0x00	3rd Byte 0x00	4th Byte 0xXY
7 (MSB)	1	0	0	1
6	1	0	0	F
5	1	0	0	V
4	1	0	0	H
3	1	0	0	P3
2	1	0	0	P2
1	1	0	0	P1
0	1	0	0	P0

Table 25-4. Coding for Protection Bits

F	V	H	P3	P2	P1	P0
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1

Table 25-5. Representations by F-Bit

F-Bit	Representations
0	ODD FIELD (FIELD 1)
1	EVEN FIELD (FIELD 2)

25.3.4 CCIR656 Progressive Mode

For a CMOS camera system of VGA or CIF resolution, strict adherence to the interlace requirements stated in the CIR standard is not required. The image is considered to have only 1 active field which is scanned in a progressive manner. This active field is regarded as field 1 and the F-bit in the timing codec is ignored by the decoder. Most sensors support CCIR timing in this mode (progressive) by default.

Figure 25-6 shows the typical flow of progressive mode.

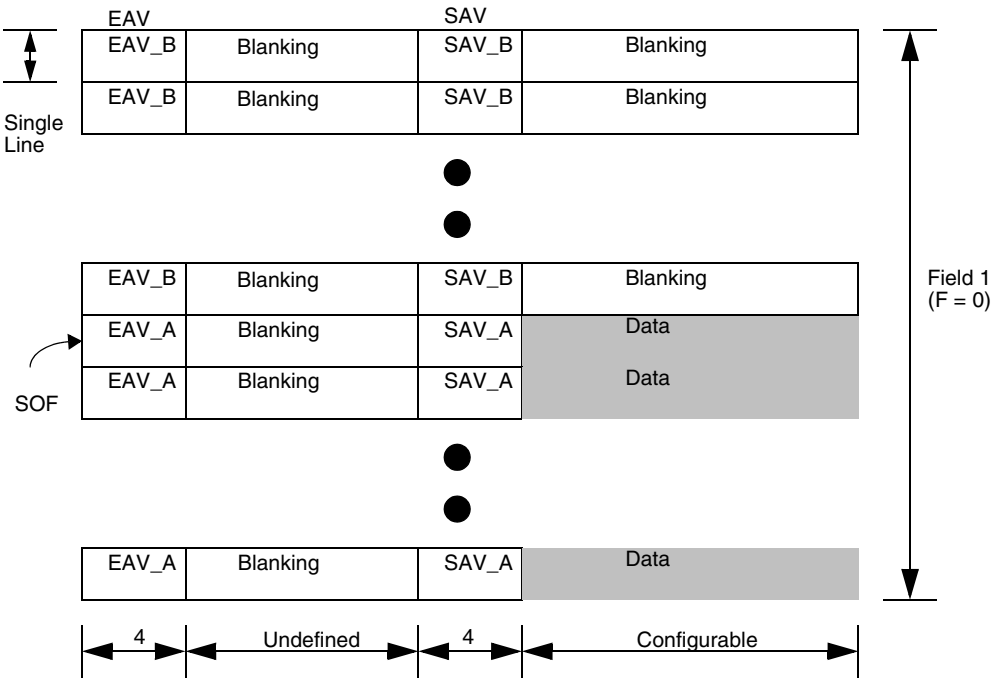


Figure 25-6. CCIR656 Progressive Mode (General Case)

An interrupt is generated for SOF but not for COF. In the general case, when SOF information is retrieved from the embedded coding, it is known as internal VSYNC mode. In other cases, when the VSYNC signal is provided by the sensor, it is known as external VSYNC mode. The CSI can be operated in internal or external VSYNC mode.

25.3.5 Error Correction for CCIR656 Coding

According to the algorithm for CCIR coding, protection bits in the SAV and EAV are encoded in the way that allows a 1-bit error to be corrected, or a 2-bit error to be detected by the decoder. This feature is supported by the CCIR decoder in CSI, for interlace mode only.

For the 1-bit error case, users can select the error to be corrected automatically, or simply shown as a status flag instead. For the 2-bit error case, because the decoder is unable to make a correction, the error would be shown as a status flag only.

An interrupt can be generated upon the detection of an error. This signal can be enabled or disabled without affecting the operation of the status bit.

25.4 Interrupt Generation

This section describes CSI events that generate interrupts.

25.4.1 Start Of Frame Interrupt (SOF_INT)

The source of an SOF interrupt is dependent on the mode of operation.

In traditional mode, VSYNC signal is taken from sensor and SOF_INT is generated at the rising or falling edge (programmable) of VSYNC.

In CCIR interlace mode, the SOF interrupt information is retrieved from the embedded coding and SOF_INT is generated.

In CCIR progressive mode, there are two sources of an SOF interrupt:

- In *internal* VSYNC mode, SOF is retrieved from the embedded coding.
- In *external* VSYNC mode, VSYNC is taken from the sensor and SOF is generated at the rising edge of VSYNC.

25.4.2 End Of Frame Interrupt (EOF_INT)

An EOF interrupt is generated when the frame ends and the complete frame data in RXFIFO is read. The EOF interrupt does not work in CSI PreProcessing mode.

The EOF event triggering works with the RX count register (CSIRXCNT). Software sets the RX count register to the frame size (in bytes). The CSI RX logic then counts the number of pixel data being received and compares it with the RX count. If the preset value is reached, then an EOF interrupt is generated and the data in the RXFIFO are read. If an SOF event is detected before this happens, then the EOF interrupt is not generated.

25.4.3 Change Of Field Interrupt (COF_INT)

The Change of Field interrupt is only valid in CCIR Interlace mode. The COF interrupt is generated when the field toggles, either from field 1 to field 2, or field 2 to field 1.

Software should first check on COF_INT bit in the CSI Status Register (CSISTAT), before checking that F1_INT or F2_INT is turned on.

In PAL systems, the field changes at the beginning of the frame and coincides with SOF. For the first field, a COF interrupt is not generated, only an SOF is generated. The COF interrupt is generated for the second field.

25.4.4 CCIR Error Interrupt (ECC_INT)

The CCIR Error Interrupt is only valid for CCIR Interlace mode. An ECC interrupt is generated when an error is found on the SAV or EAV codes in the incoming stream. When this happens, the ECC_INT status bit is set.

25.4.5 Data Packing Style

Owing to different port sizes at different stages of the image capture path, the endianess of data is important. To enable flexible packing of image data, the CSI module provides data swapping through the PACK_DIR and the SWAP16_EN bits in CSI Control Register 1 (CSICR1) which enables data swapping before it is presented to the FIFOs.

Data is packed from 8-bit to 32-bit according to the setting of PACK_DIR bit, and then put into the RX FIFO according to the setting of the SWAP16_EN bit.

25.4.6 RX FIFO Path

Bayer data is a type of raw data from the image sensor. This byte-wide data must be converted to the RGB space or YUV space by software. The data path for Bayer data is from the CSI to memory. If the system is in little endian, then the PACK_DIR bit should be set to 0. Doing so results in the data being packed to 32-bit as P3P2P1P0, where P0 is the pixel coming in time slot 0 (first data), while P3 is the pixel coming in time slot 3 (last data). When the data is addressed as bytes by software, P0 goes out first, and ends up with P3.

25.4.6.1 RGB565 Data

RGB565 data is processed data from the image sensor, which can be put directly into the display buffer. The data is 16 bits wide. The data path is from CSI to memory, memory to LCDC. On the sensor side, data must be output as P0 first, followed by P1, and so on. Within each pixel, either MSB or LSB will come out first. This is controlled by the endian style of the sensor. Data is 16 bits wide with the MSB labeled RG, and the LSB labeled GB. So for P0, it is represented as RG0, GB0, and so on for P1.

CSI receives data in one of the following sequence:

- RG0, GB0, RG1, GB1, while RG0 comes out at time slot 0 (first data), and GB1 comes out at time slot 3 (last data), or
- GB0, RG0, GB1, RG1.

Using the first sequence as an example, and assuming the system is running in little endian the data is presented as:

- 8-bit data from sensor: RG0, GB0, RG1, GB1, …
- 32-bit data before CSI RX FIFO (PACK_DIR bit = 1): RG0GB0RG1GB1
- 32-bit data in CSI RX FIFO (SWAP16_EN bit enabled): RG1GB1RG0GB0
- 32-bit transfer to system memory: RG1GB1RG0GB0
- 16-bit read by LCDC: RG0GB0, RG1GB1

25.4.6.2 RGB888 Data

This is another kind of processed data from image sensor, which can be used for further image processing directly. Each of the data consist of 8-bit Red, 8-bit Green, and 8-bit Blue data. An example of a possible timing scheme is shown in Figure 25-7.

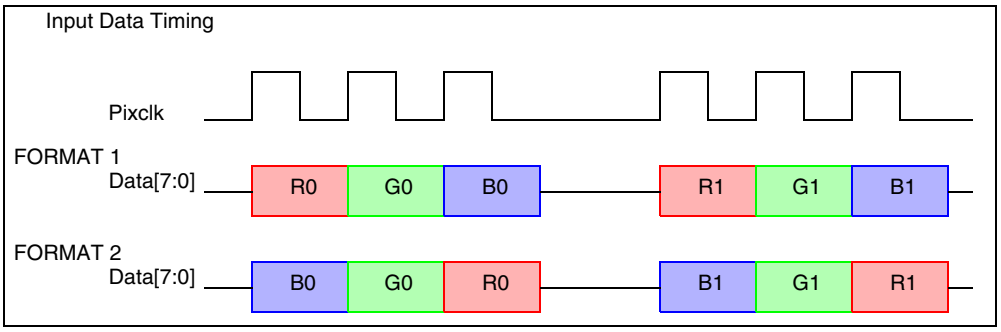


Figure 25-7. Sample Timing Diagram for RGB888 Data

The 3-byte per pixel structure is not an optimum choice for a CSI to PRP path. To improve the data transfer from CSI to PRP, an optional dummy byte packing scheme is provided. For every group of 3 bytes data, a dummy zero is packed to form a 32-bit word as shown in Figure 25-8. The dummy zero is always packed at the LSB position. This byte will be ignored by the PRP

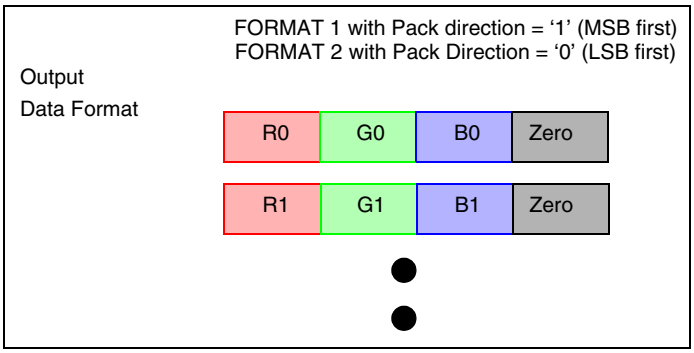


Figure 25-8. Optional Dummy Byte Packing Scheme

25.4.7 STAT FIFO Path

Statistics only works for Bayer data. It generates 16-bit statistical output from the 8-bit Bayer input. The outputs are Sum of Green (G), Sum of Red (R), Sum of Blue (B), and Auto Focus (F). Each output is 16-bits wide.

The settings of PACK_DIR and SWAP16_EN bits in the CSICR1 register have no effect on the input path. The PACK_DIR only controls how the 16-bit stat output is packed into the 32-bit STAT FIFO.

When the PACK_DIR bit = 1, the stat data is packed as:

First 32-bit: RG

Second 32-bit: BF

…

When the PACK_DIR bit = 0, the stat data is packed as:

First 32-bit GR

Second 32-bit: FB

…

25.5 Programming Model

This section provides the register summary of the CSI module. The figures used in the register description and associated text give detailed descriptions of the various module registers.

The CSI module contains six 32-bit registers which are summarized in Table 25-6.

Table 25-6. CSI Register Summary

Description	Name	Address
CSI Control Register 1	CSICR1	0x8000 0000
CSI Control Register 2	CSICR2	0x8000 0004
CSI Control Register 3	CSICR3	0x8000 001C
CSI Status Register	CSISR	0x8000 0008
CSI Statistic FIFO Register	CSISTATFIFO	0x8000 000C
CSI RX FIFO Register	CSIRXFIFO	0x8000 0010
CSI RX Count Register	CSIRXCNT	0x8000 0014

25.5.1 CSI Control Register 1 (CSICR1)

This register controls the sensor interface timing, CSI-to-PrP bus interface and interrupt generation. The CSI module is enabled through the Peripheral Clock Control Register 0 (PCCR0). Please refer to Chapter 6, “Phase-Locked Loop (PLL), Clock and Reset Controller.”

CSICR1 — CSI Control Register 1 — Addr 0x8000 0000

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SWAP16_EN	EXT_VSYNC	EOF_INT_EN	PrP_IF_EN	CCIR_MODE	COF_INT_EN	SF_OR_INTEN	RF_OR_INTEN	STATFF_LEVEL		STATFF_INTEN	RxFF_LEVEL		RxFF_INTEN	SOF_POL	SOF_INTEN
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x4000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCLKDIV				HSYNC_POL	CCIR_EN	MCLKEN	FCC	PACK_DIR	CLR_STATFIFO	CLR_RxFIFO	GCLK_MODE	INV_DATA	INV_PCLK	REDGE	
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r
RESET	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	0x0800															

Table 25-7. CSI Control Register 1 Description

Name	Description	Settings
SWAP16_EN Bit 31	**SWAP 16-Bit Enable**—This bit enables the swapping of 16-bit data. Data is packed from 8-bit to 32-bit first (according to the setting of PACK_DIR and then swapped as 16-bit words before putting into the RX FIFO. The action of the bit only affects the RX FIFO and has no affect on the STAT FIFO. **Note:** Example of swapping enabled: Data input to FIFO = 0x11223344 Data in RX FIFO = 0x 33441122 **Note:** Example of swapping disabled: Data input to FIFO = 0x11223344 Data in RX FIFO = 0x11223344	0 = Disable swapping 1 = Enable swapping
EXT_VSYNC Bit 30	**External VSYNC Enable**—This bit controls the operational VSYNC mode. **Note:** This only works when the CIS is in CCIR progressive mode.	0 = Internal VSYNC mode 1 = External VSYNC mode

Table 25-7. CSI Control Register 1 Description (continued)

Name	Description	Settings
EOF_INT_EN Bit 29	**End-of-Frame Interrupt Enable**—This bit enables and disables the EOF interrupt.	0 = EOF interrupt is disabled. 1 = EOF interrupt is generated when RX count value is reached.
PrP_IF_EN Bit 28	**CSI - PrP Interface Enable**—This bit controls the CSI to Prp bus. When enabled the RxFIFO is detached from the AHB bus and connected to PrP. All CPU reads or DMA accesses to the RxFIFO register are ignored. All CSI interrupts are also masked.	0 = CSI to PrP bus is disabled 1 = CSI to PrP bus is enabled
CCIR_MODE Bit 27	**CCIR Mode Select**—This bit controls the CCIR mode of operation. This bit only works in CCIR interface mode.	0 = Progressive mode is selected 1 = Interlace mode is selected
COF_INT_E Bit 26	**Change Of Image Field (COF) Interrupt Enable**—This bit enables the COF interrupt. This bit works only in CCIR interlace mode which is when CCIR_EN = 1 and CCIR_MODE = 1.	0 = COF interrupt is disabled 1 = COF interrupt is enabled
SF_OR_INTEN Bit 25	**STAT FIFO Overrun Interrupt Enable**—This bit enables the STATFIFO overrun interrupt.	0 = STATFIFO overrun interrupt is disabled 1 = STATFIFO overrun interrupt is enabled
RF_OR_INTEN Bit 24	**RxFIFO Overrun Interrupt Enable**—This bit enables the RX FIFO overrun interrupt.	0 = RxFIFO overrun interrupt is disabled 1 = RxFIFO overrun interrupt is enabled
STATFF_LEVEL Bits 23–22	**STATFIFO Full Level**—When the number of data in STATFIFO reach this level, STATFIFO full interrupt is generated, or STATFIFO DMA request is sent.	00 = 4 Words 01 = 8 Words 10 = 12 Words 11 = 16 Words
STATFF_INTEN Bit 21	**STATFIFO Full Interrupt Enable**—This bit enables the STAT FIFO interrupt.	0 = STATFIFO full interrupt disable 1 = STATFIFO full interrupt enable
RXFF_LEVEL Bits 20–19	**RxFIFO Full Level**—When the number of data in RxFIFO reach this level, a RxFIFO full interrupt is generated, or an RXFIFO DMA request is sent, or CSI-PrP burst cycle is issued.	00 = 4 Words 01 = 8 Words 10 = 16 Words 11 = 24 Words **Note:** In the case when PrP I/F is enabled, 24-words option is not supported, internal logic will regard it as 8-word. This is not reflected in the register value.
RXFF_INTEN Bit 18	**RxFIFO Full Interrupt Enable**—This bit enables the RxFIFO full interrupt.	0 = RxFIFO full interrupt disable 1 = RxFIFO full interrupt enable
SOF_POL Bit 17	**SOF Interrupt Polarity**—This bit controls the condition that generates an SOF interrupt.	0 = SOF interrupt is generated on SOF falling edge 1 = SOF interrupt is generated on SOF rising edge
SOF_INTEN Bit 16	**Start Of Frame (SOF) Interrupt Enable**—This bit enables the SOF interrupt.	0 = SOF interrupt disable 1 = SOF interrupt enable

Table 25-7. CSI Control Register 1 Description (continued)

Name	Description	Settings
MCLKDIV Bits 15–12	**Sensor Master Clock (MCLK) Divider**—This field contains the divisor MCLK. The MCLK is derived from the PERCLK4.	0000 = Divided by 2 0001 = Divided by 4 0010 = Divided by 6 ... 1111 = Divided by 32
HSYNC_POL Bit 11	**HSYNC Polarity Select**—This bit controls the polarity of HSYNC. **Note:** This bit only works in gated-clock—that is, GCLK_MODE = 1 and CCIR_EN = 0.	0 = HSYNC is active low 1 = HSYNC is active high
CCIR_EN Bit 10	**CCIR656 Interface Enable**—This bit selects the type of interface used. When the CCIR656 timing decoder is enabled, it replaces the function of timing interface logic.	0 = Traditional interface is selected. Timing interface logic is used to latch data. 1 = CCIR656 interface is selected.
MCLKEN Bit 9	**Sensor Master Clock (MCLK) Enable**—This bit enables or disables the MCLK input to the sensor.	0 = MCLK disable 1 = MCLK enable
FCC Bit 8	**FIFO Clear Control**—This bit determines how the RXFIFO and STATFIFO are cleared. When Synchronous FIFO clear is selected the RXFIFO and STATFIFO are cleared, and STAT block is reset, on every SOF. FIFOs and STAT block restarts immediately after reset. For a description of the operation when Asynchronous FIFO clear is selected refer to descriptions for the CLR_RXFIFO and CLR_STATFIFO bits.	0 = Asynchronous FIFO clear is selected 1 = Synchronous FIFO clear is selected
PACK_DIR Bit 7	**Data Packing Direction**—This bit Controls how 8-bit image data is packed into 32-bit RX FIFO, and how 16-bit statistical data is packed into 32-bit STAT FIFO.	0 = Pack from LSB first. For image data, 0x11, 0x22, 0x33, 0x44, it will appear as 0x44332211 in RX FIFO. For stat data, 0xAAAA, 0xBBBB, it will appear as 0xBBBBAAAA in STAT FIFO. 1 = Pack from MSB first. For image data, 0x11, 0x22, 0x33, 0x44, it will appear as 0x11223344 in RX FIFO. For stat data, 0xAAAA, 0xBBBB, it will appear as 0xAAAABBBB in STAT FIFO.
CLR_STATFIFO Bit 6	**Asynchronous STATFIFO Clear**—This bit clears the STATFIFO and Reset STAT block. **Note:** This bit works only in async FIFO clear mode—that is, FCC = 0. Otherwise this bit is ignored.	Writing 1 will clear STATFIFO and reset STAT block immediately, STATFIFO and STAT block then wait and restart after the arrival of next SOF. The bit is restored to 0 automatically after finish. Normally reads 0
CLR_RXFIFO Bit 5	**Asynchronous RXFIFO Clear**—This bit clears the RXFIFO. This bit works only in async FIFO clear mode—that is, FCC = 0. Otherwise this bit is ignored.	Writing 1 clears the RXFIFO immediately, RXFIFO restarts immediately after that. The bit is restore to 0 automatically after finish. Normally reads 0

Table 25-7. CSI Control Register 1 Description (continued)

Name	Description	Settings
GCLK_MODE Bit 4	**Gated Clock Mode Enable**—Controls if CSI is working in gated or non-gated mode. **Note:** This bit works only in traditional mode—that is, CCIR_MODE = 0. Otherwise this bit is ignored.	0 = Non-gated clock mode. All incoming pixel clocks are valid. HSYNC is ignored 1 = Gated clock mode. Pixel clock signal is valid only when HSYNC is high
INV_DATA Bit 3	**Invert Data Input**—This bit enables or disables internal inverters on the data lines.	0 = CSI_D[7:0] data lines are directly applied to internal circuitry 1 = CSI_D[7:0] data lines are inverted before applied to internal circuitry
INV_PCLK Bit 2	**Invert Pixel Clock Input**—This bit determines if the Pixel Clock (CSI_PIXCLK) is inverted before it is applied to the CSI module.	0 = CSI_PIXCLK is directly applied to internal circuitry 1 = CSI_PIXCLK is inverted before applied to internal circuitry
REDGE Bit 1	**Valid Pixel Clock Edge Select**—Selects which edge of the CSI_PIXCLK is used to latch the pixel data.	0 = Pixel data is latched at the falling edge of CSI_PIXCLK 1 = Pixel data is latched at the rising edge of CSI_PIXCLK
Reserved Bit 0	Reserved—This bit is reserved and should read 0.	

25.5.2 CSI Control Register 2 (CSICR2)

This register indicates to the statistic block which live view resolution is being used, and the starting sensor pixel of the Bayer pattern. It also contains the horizontal and vertical count used to determine the number of pixels to skip between the 64 × 64 blocks of statistics when generating statistics on live view image that are greater than 512 × 384.

CSICR2 — CSI Control Register 2 — Addr 0x8000 0004

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						DRM	AFS		SCE			BTS		LVRM		
TYPE	r	r	r	r	r	rw	rw	rw	rw	r	r	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VSC								HSC							
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 25-8. CSI Control Register 2 Description

Name	Description	Settings
Reserved Bits 31–27	Reserved—These bits are reserved and should read 0.	
DRM Bit 26	**Double Resolution Mode**—Controls size of statistics grid.	0 = Stats grid of 8 × 6 1 = Stats grid of 8 × 12
AFS Bit 25–24	**Auto Focus Spread**—Selects which green pixels are used for auto-focus.	00 = Abs Diff on consecutive green pixels 01 = Abs Diff on every third green pixels 1x = Abs Diff on every four green pixels
SCE Bit 23	**Skip Count Enable**—Enables or disables the skip count feature.	0 = Skip count disable 1 = Skip count enable
Reserved Bits 22–21	Reserved—These bits are reserved and should read 0.	
BTS Bits 20–19	**Bayer Tile Start**—Controls the Bayer pattern starting point.	00 = GR 01 = RG 10 = BG 11 = GB
LVRM Bits 18–16	**Live View Resolution Mode**—Selects the grid size used for live view resolution.	0 = 512 × 384 1 = 448 × 336 2 = 384 × 288 3 = 384 × 256 4 = 320 × 240 5 = 288 × 216 6 = 400 × 300
VSC Bits 15–8	**Vertical Skip Count**—Contains the number of rows to skip. SCE must be 1, otherwise VSC is ignored.	0–255 = Number of rows to skip minus 1
HSC Bits 7–0	**Horizontal Skip Count**—Contains the number of pixels to skip. SCE must be 1, otherwise HSC is ignored.	0–255 = Number of pixels to skip minus 1

25.5.3 CSI Control Register 3 (CSICR3)

This read/write register acts as an extension of the functionality of the CSI Control register 1 adding additional control and features.

CSICR3 — CSI Control Register 3 — Addr 0x8000 001C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FRMCNT															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FRMCNT_RST												CSI_SUP	ZERO_PACK_EN	ECC_INT_EN	ECC_AUTO_EN
TYPE	rw	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 25-9. CSI Control Register 3 Description

Name	Description	Settings
FRMCNT Bits 31–16	**Frame Counter**—This is a 16-bit Frame Counter (Wrap around automatically after reaching the maximum)	
FRMCNT_RST Bit 15	**Frame Count Reset**—Resets the Frame Counter.	0 = Do not reset 1 = Reset frame counter immediately
Reserved Bits 14–4	Reserved—These bits are reserved and should read 0.	
CSI_SVR Bit 3	**Supervisor Mode Access Control**—This bit enables and disables ARM9 supervisor mode access.	0 = Module can be accessed in any ARM9 mode 1 = Module can only be accessed in ARM9 supervisor mode. Accessing the module in non-supervisor mode will cause a Data Abort exception.
ZERO_PACK_EN Bit 2	**Dummy Zero Packing Enable**—This bit causes a dummy zero to be packed with every 3 incoming bytes, forming a 32-bit word. The dummy zero is always packed to the LSB position.	0 = Zero packing disabled 1 = Zero packing enabled

Table 25-9. CSI Control Register 3 Description (continued)

Name	Description	Settings
ECC_INT_EN Bit 1	**Error Detection Interrupt Enable**—This bit enables and disables the error detection interrupt. This feature only works in CCIR interlace mode.	0 = No interrupt is generated when error is detected. Only the status bit ECC_INT is set. 1 = Interrupt is generated when error is detected.
ECC_AUTO_EN Bit 0	**Automatic Error Correction Enable**—This bit enables and disables the automatic error correction. If an error occurs and error correction is disabled only the ECC_INT status bit is set. This feature only works in CCIR interlace mode.	0 = Auto Error correction is disabled 1 = Auto Error correction is enabled

25.5.4 CSI Status Register (CSISR)

This read/write register shows sensor interface status, and which kind of interrupt is being generated. The corresponding interrupt bits must be set for the status bit to function. Status bits should function normally even if the corresponding interrupt enable bits are not enabled

CSISR — CSI Status Register — Addr 0x8000 0008

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							SFF_ OR_INT	RFF_ OR_INT			STATFF_INT			RxFF_ INT	EOF_ INT	SOF_ INT
TYPE	r	r	r	r	r	r	rw	rw	r	r	r	r	r	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	F2_INT	F1_INT	COF_INT												ECC_ INT	DRDY
TYPE	rw	rw	rw	r	r	r	r	r	r	r	r	r	r	r	r	rw
RESET	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x4000															

Table 25-10. CSI Status Register Description

Name	Description	Setting
Reserved Bits 31–26	Reserved—These bits are reserved and should read 0.	
SF_OR_INT Bit 25	**STATFIFO Overrun Interrupt Status**—Indicates the overflow status of the STATFIFO register.	0 = STATFIFO has not overflowed 1 = STATFIFO has overflowed (Cleared by writing 1)
RF_OR_INT Bit 24	**RxFIFO Overrun Interrupt Status**—Indicates the overflow status of the RxFIFO register.	0 = RXFIFO has not overflowed 1 = RXFIFO has overflowed (Cleared by writing 1)

Table 25-10. CSI Status Register Description (continued)

Name	Description	Setting
Reserved Bits 23–22	Reserved—These bits are reserved and should read 0.	
STATFF_INT Bit 21	**STATFIFO Full Interrupt Status**—Indicates whether the STATFIFO condition is either full or not full /overflowed.	0 = STATFIFO is not full, or has overflowed 1 = STATFIFO is full (this bit is cleared automatically by reading the STATFIFO)
Reserved Bits 20–19	Reserved—These bits are reserved and should read 0.	
RxFF_INT Bit 18	**RXFIFO Full Interrupt Status**—Indicates whether the RxFIFO condition is either full or not full /overflowed.	0 = RxFIFO is not full, or has overflowed 1 = RxFIFO is full (this bit is cleared automatically by reading the RxFIFO)
EOF_INT Bit 17	**End of Frame (EOF) Interrupt Status**—Indicates when EOF is detected.	0 = EOF is not detected 1 = EOF is detected (Cleared by writing 1)
SOF_INT Bit 16	**Start of Frame Interrupt Status**—Indicates when SOF is detected.	0 = SOF is not detected 1 = SOF is detected (Cleared by writing 1)
F2_INT Bit 15	**CCIR Field 2 Interrupt Status**—Indicates the presence of field 2 of video in CCIR mode. **Note:** Only works in CCIR Interlace mode.	0 = Field 2 of video is not detected 1 = Field 2 of video is about to start (Cleared automatically when current field does not match)
F1_INT Bit 14	**CCIR Field 1 Interrupt Status**—Indicates the presence of field 1 of video in CCIR mode. **Note:** Only works in CCIR Interlace mode.	0 = Field 1 of video is not detected 1 = Field 1 of video is about to start (Cleared automatically when current field does not match)
COF_INT Bit 13	**Change Of Field Interrupt Status**—Indicates that a change of the video field has been detected. Only works in CCIR Interlace mode. Software should read this bit first and then dispatch the new field from F1_INT and F2_INT.	1 = Change of video field is detected 0 = Video field has no change (Cleared by writing 1)
Reserved Bits 12– 2	Reserved—These bits are reserved and should read 0.	
ECC_INT Bit 1	**CCIR Error Interrupt**—This bit indicates an error has occurred. This only works in CCIR Interlace mode.	0 = No error detected (Cleared by writing 1) 1 = Error is detected in CCIR coding
DRDY Bit 0	**RXFIFO Data Ready**—Indicates the presence of data that is ready for transfer in the RxFIFO.	0 = No data (word) is ready 1 = At least 1 data (word) is ready in RXFIFO (Cleared automatically by reading FIFO)

25.5.5 CSI STATFIFO Register (CSISTATFIFO)

The StatFIFO is a read-only register containing statistic data from the sensor. Writing to this register has no effect.

CSISTATFIFO — CSI Statistic FIFO Register — 0x8000 000C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STAT															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STAT															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

25.5.6 CSI RxFIFO Register (CSIRFIFO)

This read-only register contains received image data. Writing to this register has no effect.

CSIRFIFO — CSI RX FIFO Register — Addr 0x8000 0010

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IMAGE															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IMAGE															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Because the incoming image data is 8-bit while the FIFO size is 32-bit, the incoming byte size data will be packed into 32-bit in the following sequence.

25.5.7 CSI RX Count Register (CSIRXCNT)

This register works for EOF interrupt generation. It should be set to the number of words to receive that would generate an EOF interrupt.

There is an internal counter that counts the number of words read from the RX FIFO. Whenever the RX FIFO is being read, by either the CPU or DMA, the counter value is updated and compared with this register. If the values match, then an EOF interrupt is triggered.

CSIRXCNT — CSI RX COUNT Register — 0x8000 0014

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											RXCNT					
TYPE	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXCNT															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 25-11. CSI RX Count Register Description

Name	Description
Reserved Bits 31–22	Reserved—These bits are reserved and should read 0.
RXCNT Bits 21–0	**RxFIFO Count**—This 22-bit counter for RXFIFO is updated each time the RXFIFO is read by CPU or DMA.This counter should be set to the expected number of words to receive that would generate an EOF interrupt.

Chapter 26
Liquid Crystal Display Controller (LCDC)

The Liquid Crystal Display Controller (LCDC) provides display data for external gray-scale or color LCD panels. The LCDC is capable of supporting black-and-white, gray-scale, passive-matrix color (passive color or CSTN), and active-matrix color (active color or TFT) LCD panels.

The LCDC provides the following features:

- Support for single (non-split) screen monochrome or color LCD panels and self-refresh type LCD panels
- 16 simultaneous gray-scale levels from a palette of 16 for monochrome display
- Support for:
  - Maximum resolution of 800 × 600
  - Passive color panel:
    - 4 (mapped to RGB444) / 8 (mapped to RGB444) / 12 (RGB444) bits per pixel (bpp)
  - TFT panel:
    - 4 (mapped to RGB666) / 8 (mapped to RGB666) / 12 (RGB444) / 16 (RGB565) / 18 (RGB666) bpp
  - 16 and 256 colors out of a palette of 4096 colors for 4 bpp and 8 bpp CSTN display respectively
  - 16 and 256 colors out of a palette of 256K colors for 4 bpp and 8 bpp TFT display respectively
  - True 4096 colors for a 12 bpp display
  - True 64K colors for 16 bpp
  - True 256K colors for 18 bpp
  - Additional support details are shown in Table 26-1

Table 26-1. Supported Panel Characteristics

Panel Type	Bit/Pixel	Panel Interface (Bits)	Number of Gray Level/Color
Monochrome	1	1, 2, 4, 8	black-and-white
	2	1, 2, 4, 8	4 out of palette of 16
	4	1, 2, 4, 8	16
CSTN	4, 8	8	16, 256 out of palette of 4096
	12	8	4096
TFT	4, 8	18	16, 256 out of palette of 256K
	12, 16, 18	12, 16, 18	4096, 64K, 256K

- Standard panel interface for common LCD drivers
- Panel interface of 1-, 2-, 4-, 8-bit for monochrome panels

- Panel interface of 12-, 16-, 18-bit for color panels
- For 4 bpp and 8 bpp a palette table is used for re-mapping of data from memory, independent of type of panel used. For the 1 bpp, 2 bpp, 12 bpp, 16 bpp, and 18 bpp the palette table is by-passed.
- Interface to passive and active color panel (TFT)
- Supports timing requirements for Sharp 240 × 320 HR-TFT panel
- Hardware-generated cursor with blink, color, and size programmability
- Logical operation between color hardware cursor and background
- Hardware panning (soft horizontal scrolling)
- 8-bit pulse-width modulator for software contrast control
- Graphic window support for viewfinder function in color display
- Graphic window color keying for graphical hardware cursor
- 256 transparency levels for alpha blending between graphic window and background plane

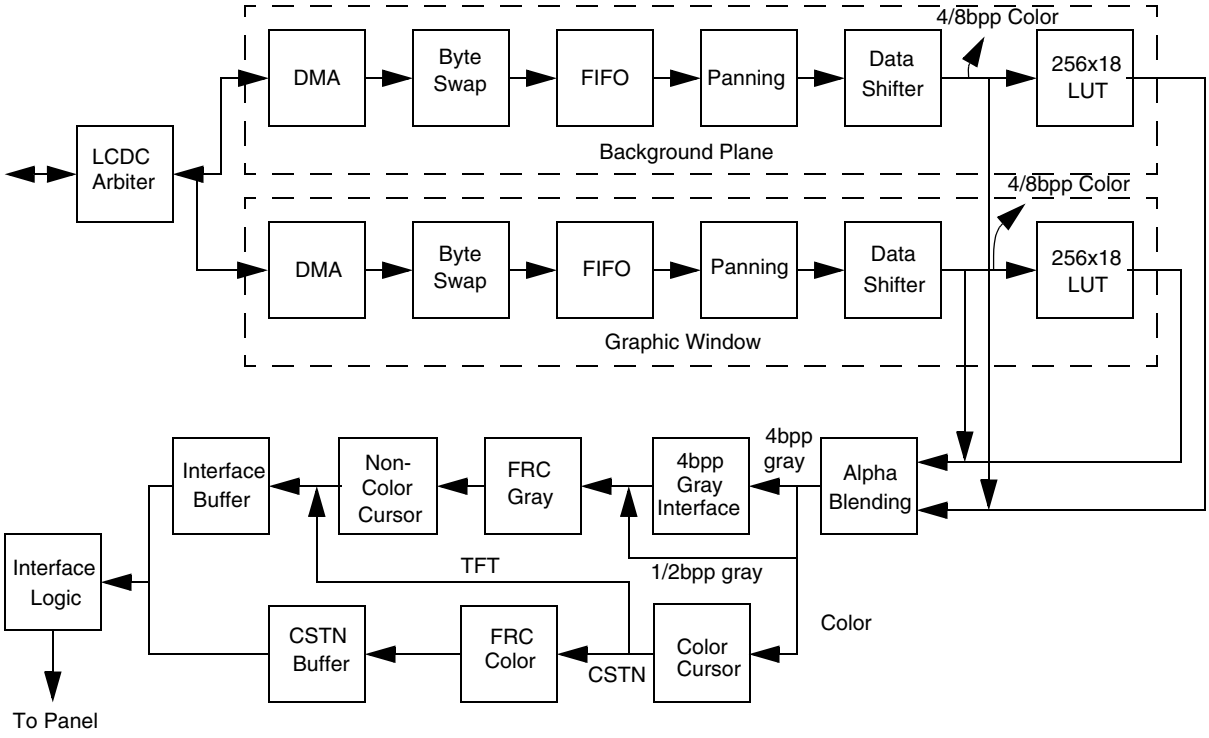


Figure 26-1. LCDC Block Diagram

26.1 LCDC Operation

The following sections describe the operation of the LCD Controller with various industry standard LCD displays.

26.1.1 LCD Screen Format

The number of pixels forming the screen width and screen height of the LCD panel are software programmable. Figure 26-2 shows the relationship between the screen size and memory window.

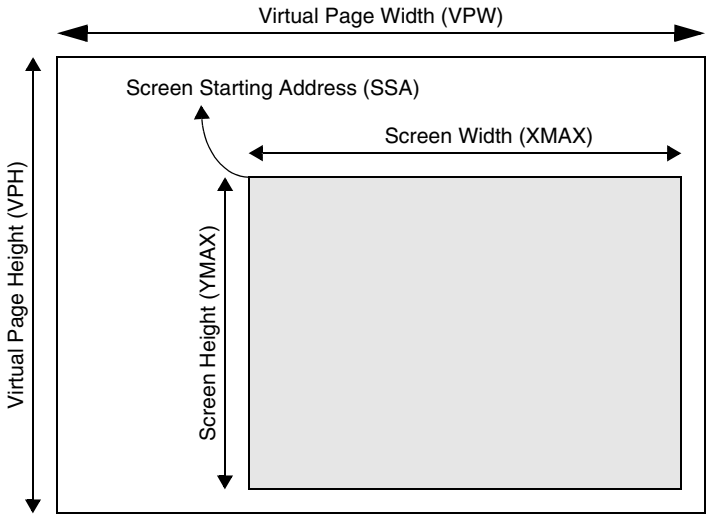


Figure 26-2. LCD Screen Format

The screen width (XMAX) and screen height (YMAX) parameters specify the LCD panel size. The LCDC begins scanning the display memory at the location pointed to by the screen starting address (SSA) register, represented by the shaded area in Figure 26-2, for display on the LCD panel.

The maximum page width is specified by the virtual page width (VPW) parameter. Virtual page height (VPH) does not affect the LCDC and is limited only by memory size. By changing the SSA register, a screen-sized window can be vertically or horizontally scrolled (panned) anywhere inside the virtual page boundaries. The software must control the starting address in the SSA properly so that the scanning logic's system memory pointer (SMP) stays within the VPW and VPH limits to prevent the display of strange artifacts on the screen.

VPH is used by the programmer only for boundary checks. There is no VPH parameter internal to the LCDC.

VPW is used in calculating the RAM starting address representing the beginning of each displayed line. SSA sets the address of data for the first line of a frame. For each subsequent line, VPW is added to an accumulation initialized by the SSA to yield the starting address of that line.

26.1.2 Graphic Window on Screen

Graphic window is supported in LCD color panel screen for viewfinder and graphic hardware cursor functions. Similar to the screen, the virtual page width, graphic window start address and graphic window width and height are software programmable. The position of the graphic window on screen is specified by the graphic window position register. Figure 26-3 shows how the graphic window is configured and placed on the screen.

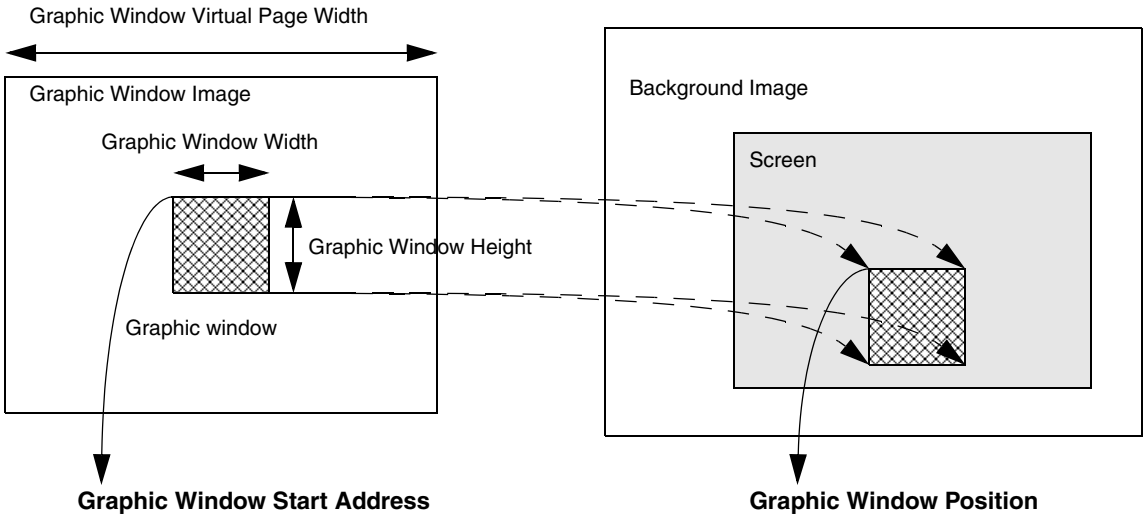


Figure 26-3. Graphic Window on Screen

The graphic window and background plane can be alpha blended. The alpha value is window basis which means all the pixels in the graphic window have the same level of transparency. There are a total of 256 levels of transparency to be configured. In addition, one of the pixel colors can be chosen for color keying in which the selected pixel color in the graphics window is made totally transparent. One of the applications can be a graphical hardware cursor.

NOTE

It is important to note that the graphic window and background images must have the same bpp setting.

26.1.3 Panning

Panning offset (POS) is expressed in bits, not pixels, so when operating in any mode other than 1 bpp, only even pixel boundaries are valid. In 12 bpp mode, the pixels are aligned to 16-bit boundaries, and POS also must align to these boundaries.

SSA and POS are located in isolated registers and are double buffered because they are dynamic parameters likely to change while the LCDC is running. New values of SSA and POS do not take effect until the beginning of the next frame. A typical panning algorithm includes an interrupt at the beginning of the frame. In the interrupt service routine, POS and/or SSA are updated (the old values are internally latched). The updates take effect on the next frame.

26.1.4 Display Data Mapping

The LCDC supports 1/2/4 bpp in monochrome mode and 4/8/12/16/18 bpp in color mode. System memory data mapping in 2/4/8/12/16/18 bpp modes is shown in Figure 26-4.

NOTE

In 12 bpp mode, 16 bits of memory are used for each set of 12 bits, to leave 4 bits unused. In 18 bpp mode, 32 bits of memory are used for each pixel, leaving 14 bits unused. Refer to Table 26-5.

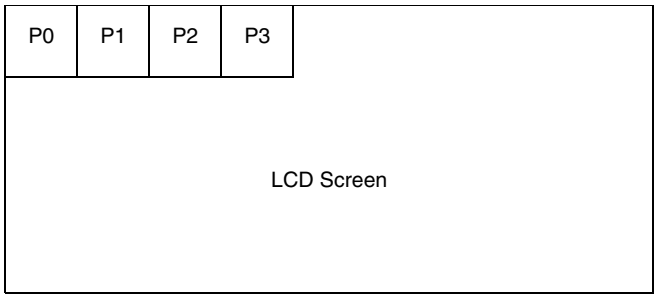


Figure 26-4. Pixel Location on Display Screen

1 bpp Mode

Byte Address	Sample Bit-to-Pixel Mapping							
3	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	P24	P25	P26	P27	P28	P29	P30	P31
2	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	P16	P17	P18	P19	P20	P21	P22	P23
1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	P8	P9	P10	P11	P12	P13	P14	P15
0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	P0	P1	P2	P3	P4	P5	P6	P7

2 bpp Mode

Byte Address	Sample Bit-to-Pixel Mapping							
3	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	P12		P13		P14		P15	
2	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	P8		P9		P10		P11	
1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	P4		P5		P6		P7	
0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	P0		P1		P2		P3	

4 bpp Mode

Byte Address	Sample Bit-to-Pixel Mapping							
3	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	P6				P7			
2	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	P4				P5			
1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	P2				P3			
0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	P0				P1			

8 bpp Mode

Byte Address	Sample Bit-to-Pixel Mapping							
3	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	P3							
2	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	P2							
1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	P1							
0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	P0							

12 bpp Mode

Byte Address	Sample Bit-to-Pixel Mapping							
3	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
					Red1 [3]	Red1 [2]	Red1 [1]	Red1 [0]
2	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Green1 [3]	Green1 [2]	Green1 [1]	Green1[0]	Blue1 [3]	Blue1 [2]	Blue1 [1]	Blue1 [0]
1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
					Red0 [3]	Red0 [2]	Red0 [1]	Red0 [0]
0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Green0 [3]	Green0 [2]	Green0 [1]	Green0 [0]	Blue0 [3]	Blue0 [2]	Blue0 [1]	Blue0 [0]

16 bpp Mode

Byte Address	Sample Bit-to-Pixel Mapping							
3	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	Red1 [4]	Red1 [3]	Red1 [2]	Red1 [1]	Red1 [0]	Green1 [5]	Green1 [4]	Green1 [3]
2	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Green1 [2]	Green1 [1]	Green1 [0]	Blue1 [4]	Blue1 [3]	Blue1 [2]	Blue1 [1]	Blue1 [0]
1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Red0 [4]	Red0 [3]	Red0 [2]	Red0 [1]	Red0 [0]	Green0 [5]	Green0 [4]	Green0 [3]
0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Green0 [2]	Green0 [1]	Green0 [0]	Blue0 [4]	Blue0 [3]	Blue0 [2]	Blue0 [1]	Blue0 [0]

Figure 26-5. Display Data Mapping 1 bpp through 16 bpp Modes

18bpp Mode

Byte Address	Sample Bit-to-Pixel Mapping							
3	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
2	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Red [5]	Red [4]	Red [3]	Red [2]	Red [1]	Red [0]		
1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Green [5]	Green [4]	Green [3]	Green [2]	Green [1]	Green [0]		
0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Blue [5]	Blue [4]	Blue [3]	Blue [2]	Blue [1]	Blue [0]		

Microsoft PAL_BGR 18bpp Mode

Byte Address	Sample Bit-to-Pixel Mapping							
3	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
2	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	Blue [5]	Blue [4]	Blue [3]	Blue [2]	Blue [1]	Blue [0]		
1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Green [5]	Green [4]	Green [3]	Green [2]	Green [1]	Green [0]		
0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Red [5]	Red [4]	Red [3]	Red [2]	Red [1]	Red [0]		

Figure 26-6. Display Data Mapping for 18 bpp Mode

26.1.5 Black-and-White Operation

The 1 bpp mode is also known as black-and-white mode because each pixel is always either fully on or fully off.

26.1.6 Gray-Scale Operation

The LCDC generates a maximum of 16 gray levels. These gray levels are defined by 2 or 4 bits of display data for each pixel. Using 2 bpp, the LCDC displays 4 shades of gray, and using 4 bpp, the LCDC displays all 16 shades. The shades of gray are obtained by controlling the number of frames in which the pixel is "on" over a period of 16 frames. This method is known as frame rate control (FRC). For more information on FRC, see Section 26.1.8, "Frame Rate Modulation Control (FRC)."

The use of the mapping RAM is shown in Figure 26-7. When using 2 bpp, the 2-bit code is mapped to one of four gray levels, and when using 4 bpp, the 4-bit code is mapped to one of 16 gray levels. Because crystal formulations and driving voltages vary, the visual gray effect may or may not be linearly related to the frame rate. A logarithmic scale such as 0, 1/4, 1/2 and 1 might be more pleasing than a linearly spaced scale such as 0, 5/16, 11/16 and 1 for certain graphics.

Figure 26-7 illustrates gray-scale pixel generation. The flexible mapping scheme allows the user to optimize the visual effect for a specific panel or application.

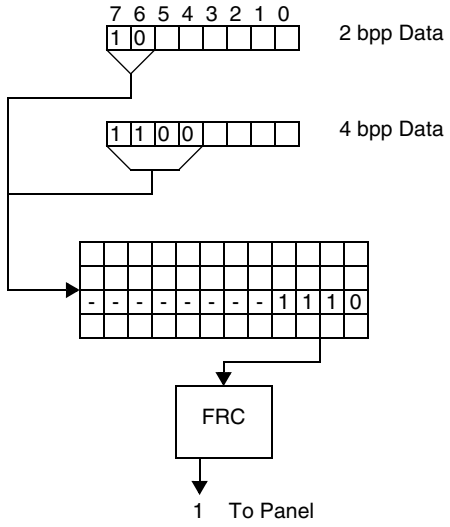


Figure 26-7. Gray-Scale Pixel Generation

26.1.7 Color Generation

The value corresponding to each color pixel on the screen is represented by a 4-, 8-, 12-, 16- or 18-bit code in the display memory.

For the 4- and 8- bit modes the LCDC's color mapping RAM is used to map the data to a 12-bit and 18-bit RGB code for passive and active matrix color displays respectively. For 4-bit and 8-bit passive matrix color displays, the 12-bit RGB code from the mapping RAM is output to the FRC blocks that independently process the code corresponding to the red, green, and blue components of each pixel to generate the required shade and intensity.

For 4-bit and 8-bit active matrix display, the 18-bit output from the mapping RAM is output to the panel.

For 12-bit mode for passive matrix color display, the mapping RAM is by-passed and output directly to the FRC block.

For 12-, 16- and 18-bit active matrix color display, pixel data is simply moved from display memory to the LCDC output bus.

Figure 26-8 and Figure 26-9 illustrate passive matrix and active matrix color pixel generation.

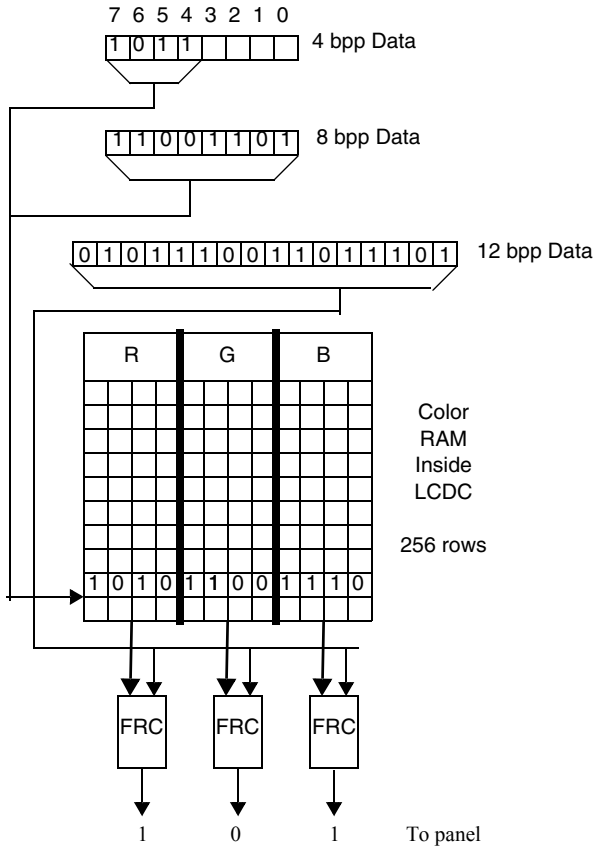


Figure 26-8. Passive Matrix Color Pixel Generation

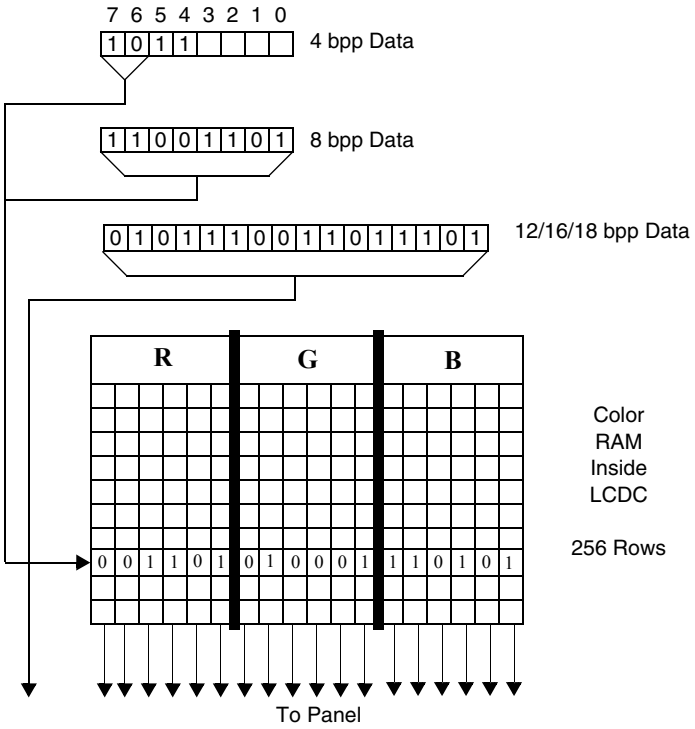


Figure 26-9. Active Matrix Color Pixel Generation

26.1.8 Frame Rate Modulation Control (FRC)

Circuitry inside the LCDC generates intermediate gray-scale colors on the panel by adjusting the density of zeroes and ones that appear over the frames. The LCDC can generate 16 simultaneous gray-scale levels.

Table 26-2. Gray Palette Density

Gray Code (Hexadecimal)	Density	Density (Decimal)
0	0	0
1	1/8	0.125
2	1/5	0.2
3	1/4	0.25
4	1/3	$0.\overline{333}$
5	2/5	0.4
6	4/9	$0.\overline{444}$
7	1/2	0.5
8	5/9	$0.\overline{555}$
9	3/5	0.6
A	2/3	$0.\overline{666}$
B	3/4	0.75

Table 26-2. Gray Palette Density (continued)

Gray Code (Hexadecimal)	Density	Density (Decimal)
C	4/5	0.8
D	7/8	0.875
E	14/15	$0.9\overline{33}$
F	1	1
Note: Overbars indicate repeating decimal numbers.		

26.1.9 Panel Interface Signals and Timing

The LCDC continuously provides pixel data to the LCD panel via the LCD panel interface. Panel interface signals are illustrated in Figure 26-10.

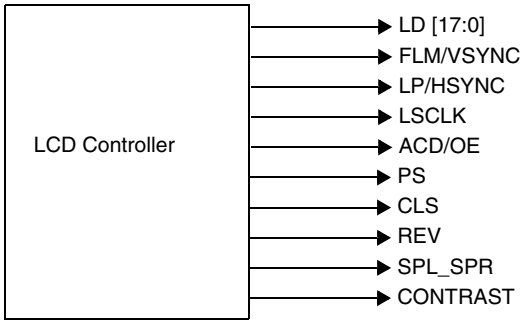


Figure 26-10. LCDC Interface Signals

The format, timing, and polarity of the panel interface signals are programmable. There are two basic modes, passive and active, selected by the TFT register bit. The user must also select either grayscale mode or color mode.

SPL_SPR, PS, CLS and REV are other interface signals from the LCDC. However, these signals are dedicated for Sharp HR-TFT 240 × 320 panels only.

26.1.9.1 Pin Configuration for LCDC

Figure 26-10 shows the signals used for the LCDC. These pins are multiplexed with other functions on the device, and must be configured for LCDC operation before they can be used.

NOTE

The user must ensure that the data direction bits in the GPIO are set to the correct direction for proper operation. See Section 15.5.1, “Data Direction Register,” for details.

Table 26-3. Pin Configuration

Pin	Setting	Configuration Procedure
ACD/OE	Primary function of GPIO Port A[31]	1. Clear bit 31 of Port A GPIO In Use Register (GIUS_A) 2. Clear bit 31 of Port A General Purpose Register (GPR_A)
CONTRAST	Primary function of GPIO Port A[30]	1. Clear bit 30 of Port A GPIO In Use Register (GIUS_A) 2. Clear bit 30 of Port A General Purpose Register (GPR_A)
FLM/VSYNC	Primary function of GPIO Port A[29]	1. Clear bit 29 of Port A GPIO In Use Register (GIUS_A) 2. Clear bit 29 of Port A General Purpose Register (GPR_A)
LP/HSYNC	Primary function of GPIO Port A[28]	1. Clear bit 28 of Port A GPIO In Use Register (GIUS_A) 2. Clear bit 28 of Port A General Purpose Register (GPR_A)
SPL_SPR	Primary function of GPIO Port A[27]	1. Clear bit 27 of Port A GPIO In Use Register (GIUS_A) 2. Clear bit 27 of Port A General Purpose Register (GPR_A)
PS	Primary function of GPIO Port A[26]	1. Clear bit 26 of Port A GPIO In Use Register (GIUS_A) 2. Clear bit 26 of Port A General Purpose Register (GPR_A)
CLS	Primary function of GPIO Port A[25]	1. Clear bit 25 of Port A GPIO In Use Register (GIUS_A) 2. Clear bit 25 of Port A General Purpose Register (GPR_A)
REV	Primary function of GPIO Port A[24]	1. Clear bit 24 of Port A GPIO In Use Register (GIUS_A) 2. Clear bit 24 of Port A General Purpose Register (GPR_A)
LD [17:0]	Primary function of GPIO Port A[23:6]	1. Clear bits [23:6] of Port A GPIO In Use Register (GIUS_A) 2. Clear bits [23:6] of Port A General Purpose Register (GPR_A)
LSCLK	Primary function of GPIO Port A[5]	1. Clear bit 5 of Port A GPIO In Use Register (GIUS_A) 2. Clear bit 5 of Port A General Purpose Register (GPR_A)

26.1.9.2 Passive Matrix Panel Interface Signals

Figure 26-11 shows the LCD interface timing for monochrome panels and Figure 26-12 shows the LCD interface timing for passive matrix color panels. Signal polarities are shown positive, however it can be reversed by clearing the bits in the Panel Configuration Register (PCR). The data bus timing for passive panels is controlled by the shift clock (LSCLK), line pulse (LP), first line marker (FLM), alternate crystal direction (ACD), and line data (LD) signals.

Operation of the panel interface is accomplished in the following steps:

1. LSCLK clocks the pixel data into the display driver's internal shift register.
2. LP signifies the end of the current line of serial data and latches the shifted pixel data into a wide latch.
3. FLM marks the first line of the displayed page. The LD (and the associated LP), enclosed by the FLM signal, marks the first line of the current frame.
4. ACD toggles after a pre-programmed number of FLM pulses. This signal refreshes the LCD panel.

NOTE

The LD bus width is programmable to 1, 2, 4, or 8 bits in monochrome mode (the COLOR bit in the Panel Configuration register is set to 0). Data is justified to the least significant bits of the LD [17:0] bus. Passive color displays use a fixed 2-2/3 pixels of data per 8-bit vector as shown in Figure 26-12.

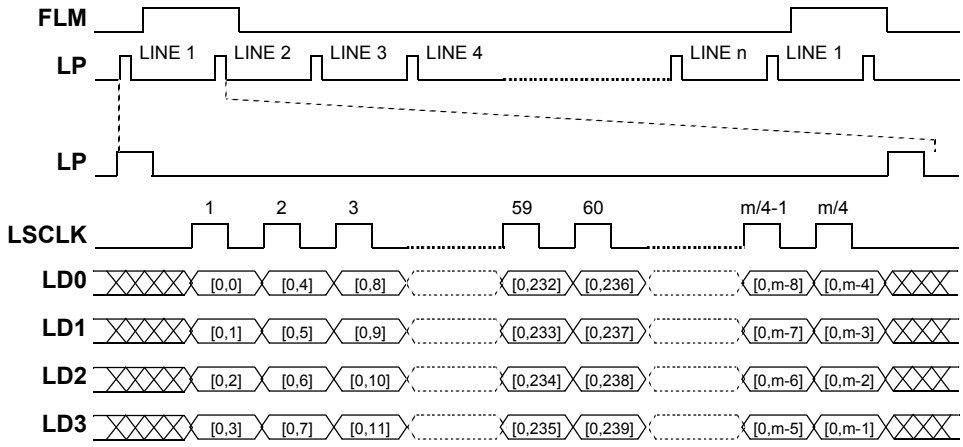


Figure 26-11. LCDC Interface Timing for 4-bit Data Width Gray-Scale Panels

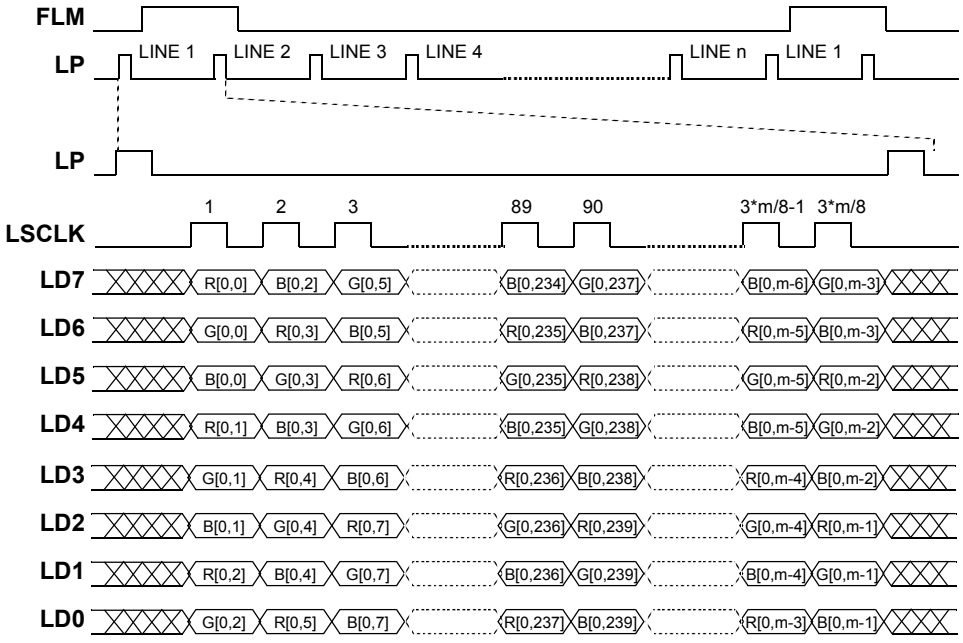


Figure 26-12. LCDC Interface Timing for 8-bit Data Passive Matrix Color Panels

26.1.9.3 Passive Panel Interface Timing

Figure 26-13 shows the horizontal timing (timing of one line), including both the line pulse (LP) and the data. The width of LP and delays both before and after LP are programmable.

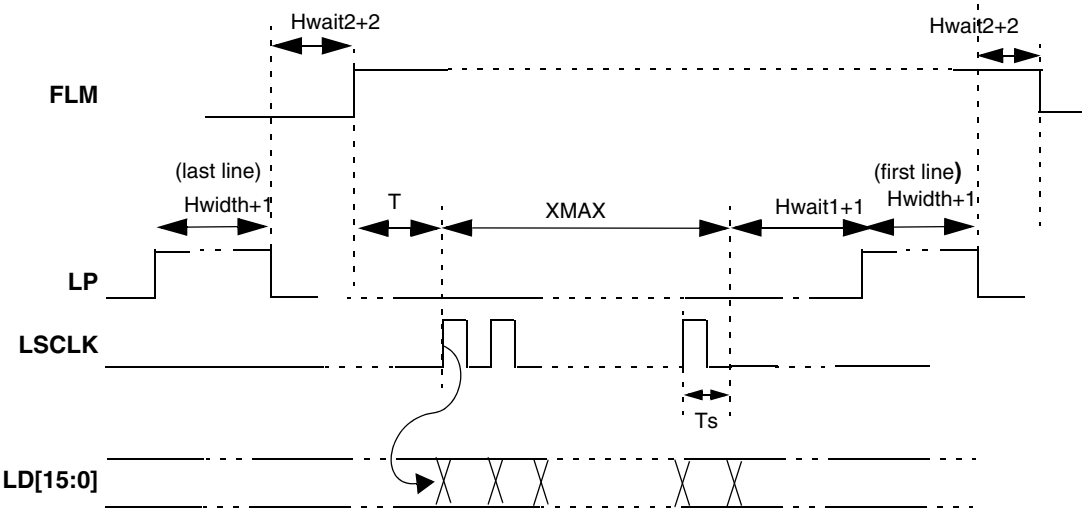
The parameters used for panel interface timing are:

- XMAX (X size) defines the number of pixels per line. XMAX is the total number of pixels per line.
- H_WAIT_1 defines the delay from the end of data output to the beginning of LP.
- H_WIDTH (horizontal sync pulse width) defines the width of the FLM pulse, and H_WIDTH must be at least 1.
- H_WAIT_2 defines the delay from the end of LP to the beginning of data output.

NOTE

All parameters are defined in unit of pixel clock period, unless stated otherwise.

26.1.10 8 bpp Mode Color STN Panel



When it is in CSTN mode or monochrome mode with bus width = 1, T = 1 SCLK period.
When it is in monochrome mode with bus width = 2, 4 and 8, T = 1, 2 and 4 SCLK period respectively.

Figure 26-13. Horizontal Sync Pulse Timing in Passive Mode

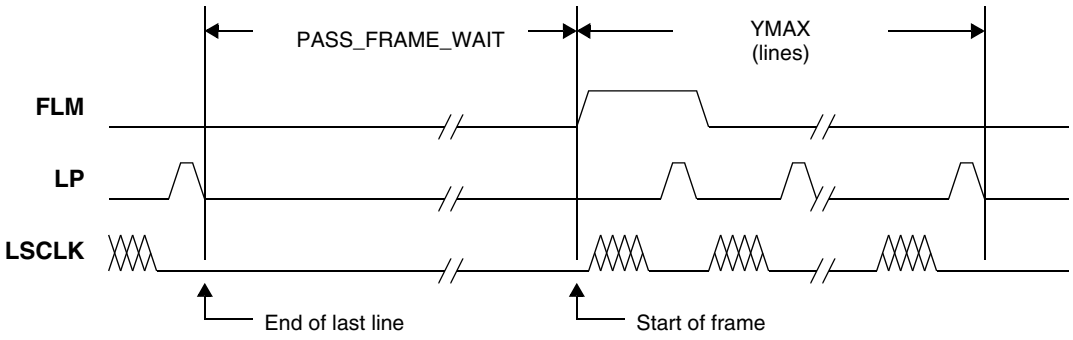


Figure 26-14. Vertical Sync Pulse Timing in Passive Mode

26.1.10.1 Active Matrix Panel Interface Signals

Figure 26-15 shows the LCD interface timing for an active matrix color TFT panel. In this figure, signals are shown with negative polarity (FLMPOL=1, LPPOL=1, CLKPOL=0, OEPOL=1). In TFT mode, the LSCLK is automatically inverted. The panel interface timing for active matrix panels is sometimes referred to as a “digital CRT” and is controlled by the shift clock (LSCLK), horizontal sync pulse (HSYNC, the LP pin in passive mode), vertical sync pulse (VSYNC, the FLM pin in passive mode), output enable (OE, the ACD pin in passive mode), and line data (LD) signals. The sequence of events for active matrix interface timing is:

1. LSCLK latches data into the panel on its negative edge (when positive polarity is selected). In active mode, LSCLK runs continuously.
2. HSYNC causes the panel to start a new line.
3. VSYNC causes the panel to start a new frame. It always encompasses at least one HSYNC pulse.
4. OE functions as an output enable signal to the CRT. This output enable signal is similar to the blanking output in a CRT and enables the data to be shifted onto the display. When disabled, the data is invalid and the trace is off.

In 4- and 8-bit mode, the LD [17:12] bits define red, the LD [11:6] bits define green, and the LD [5:0] bits define blue. In 12-bit mode, the LD[17:14] bits define red, the LD[11:8] bits define green, and the LD[5:2] bits define blue. In 16-bit mode, the LD [17:13] bits define red, the LD [11:6] bits define green, and the LD [5:1] bits define blue.

The actual TFT color channel assignments are shown in Table 26-4. The unused bits are fixed at 0.

Table 26-4. TFT Color Channel Assignments

	LD 17	LD 16	LD 15	LD 14	LD 13	LD 12	LD 11	LD 10	LD 9	LD 8	LD 7	LD 6	LD 5	LD 4	LD 3	LD 2	LD 1	LD 0
4 bpp	R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
8 bpp	R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
12 bpp	R3	R2	R1	R0	–	–	G3	G2	G1	G0	–	–	B3	B2	B1	B0	–	–
16 bpp	R4	R3	R2	R1	R0	–	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0	–
18 bpp	R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0

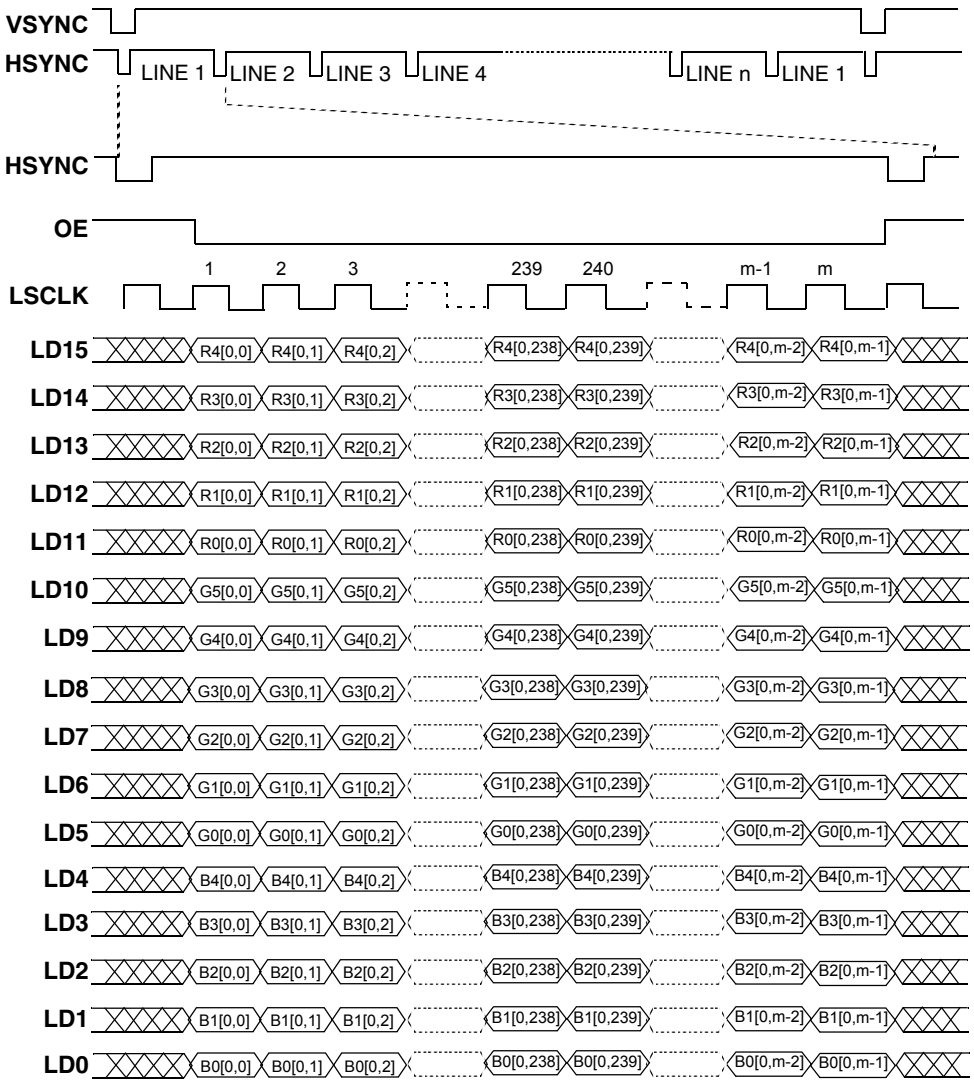


Figure 26-15. LCDC Interface Timing for Active Matrix Color Panels

26.1.10.2 Active Panel Interface Timing

Figure 26-16 shows the horizontal timing (timing of one line), including both the horizontal sync pulse and the data. The width of HSYNC and delays both before and after HSYNC are programmable.

The timing signal parameters are defined as follows:

- H_WIDTH defines the width of the HSYNC pulse and must be at least 1.
- H_WAIT_2 defines the delay from the end of HSYNC to the beginning of the OE pulse.
- H_WAIT_1 defines the delay from end of OE to the beginning of the HSYNC pulse.
- XMAX defines the (total) number of pixels per line.

NOTE

All parameters are defined in pixel periods, not LSCLK periods.

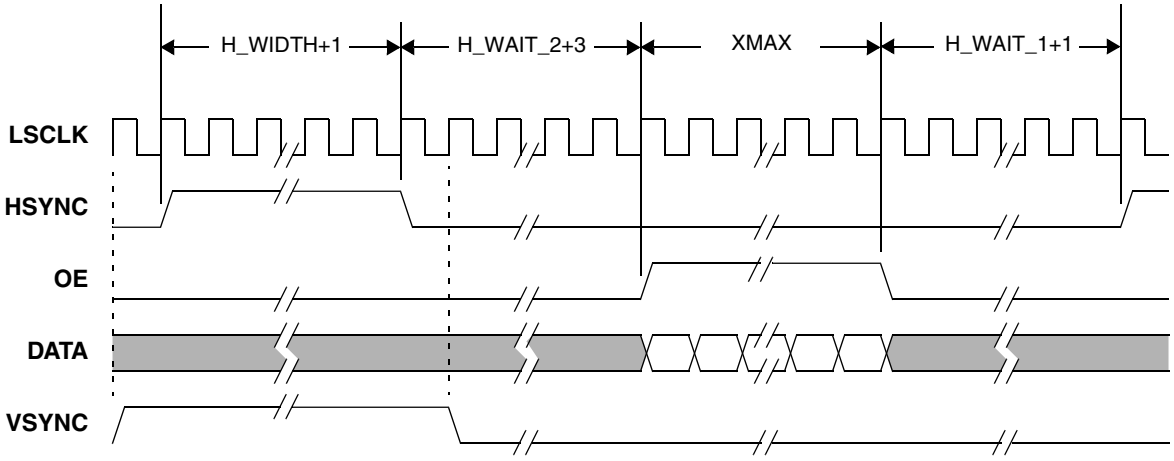


Figure 26-16. Horizontal Sync Pulse Timing in TFT Mode

Figure 26-17 shows the vertical timing (timing of one frame). The delay from the end of one frame until the beginning of the next is programmable. The memory timing signal parameters are:

- V_WAIT_1 is a delay measured in lines. For V_WAIT_1= 1 there is a delay of one HSYNC (time = one line period) before VSYNC. The HSYNC pulse is output during the V_WAIT_1 delay.
- For V_WIDTH (vertical sync pulse width) = 0, VSYNC encloses one HSYNC pulse. For V_WIDTH = 2, VSYNC encloses two HSYNC pulses.
- V_WAIT_2 is a delay measured in lines. For V_WAIT_2 = 1, there is a delay of one HSYNC (time = one line period) after VSYNC. The HSYNC pulse is output during the V_WAIT_2 delay.

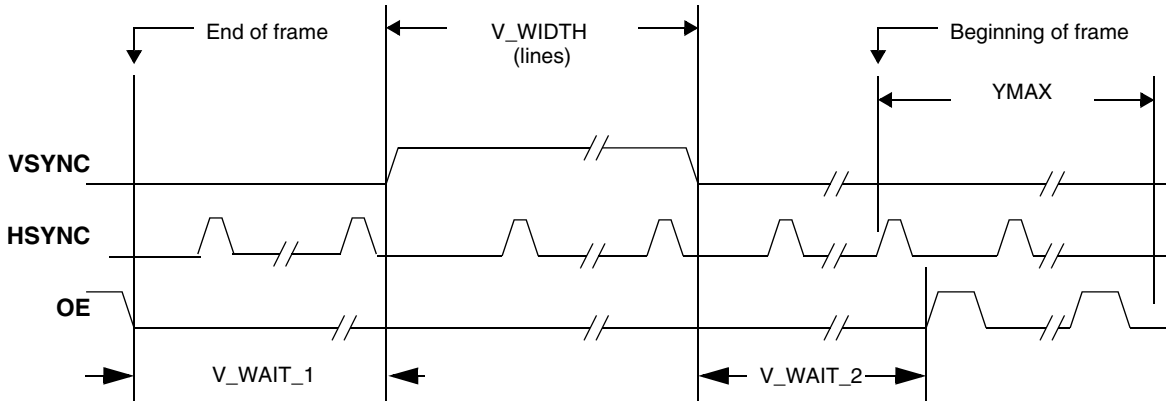


Figure 26-17. Vertical Sync Pulse Timing TFT Mode

26.2 Programming Model

The LCDC memory space contains 21 32-bit registers for display parameters, a read-only status register, and 2 256 × 18 Color Mapping RAMs—one for graphic window and the other for background plane. The color mapping RAMs are physically located inside the Palette Lookup Table module.

Table 26-5 summarizes these registers and their addresses. Only WORD access is supported. Byte and halfword access is undefined.

Table 26-5. LCDC Register Summary

Description	Name	Address
Screen Start Address Register	LSSAR	0x1002 1000
LCDC Size Register	LSR	0x1002 1004
LCDC Virtual Page Width Register	LVPWR	0x1002 1008
LCDC Cursor Position Register	LCPR	0x1002 100C
LCDC Cursor Width Height and Blink Register	LCWHBR	0x1002 1010
LCDC Color Cursor Mapping Register	LCCMR	0x1002 1014
LCDC Panel Configuration Register	LPCR	0x1002 1018
LCDC Horizontal Configuration Register	LHCR	0x1002 101C
LCDC Vertical Configuration Register	LVCR	0x1002 1020
LCDC Panning Offset Register	LPOR	0x1002 1024
LCDC Sharp Configuration Register	LSCR	0x1002 1028
LCDC PWM Contrast Control Register	LPCCR	0x1002 102C
LCDC DMA Control Register	LDCR	0x1002 1030
LCDC Refresh Mode Control Register	LRMCR	0x1002 1034
LCDC Interrupt Configuration Register	LICR	0x1002 1038
LCDC Interrupt Enable Register	LIER	0x1002 103C
LCDC Interrupt Status Register	LISR	0x1002 1040
LCDC Graphic Window Start Address Register	LGWSAR	0x1002 1050
LCDC Graphic Window Size Register	LGWSR	0x1002 1054
LCDC Graphic Window Virtual Page Width Register	LGWVPWR	0x1002 1058
LCDC Graphic Window Panning Offset Register	LGWPOR	0x1002 105C
LCDC Graphic Window Position Register	LGWPR	0x1002 1060
LCDC Graphic Window Control Register	LGWCR	0x1002 1064
LCDC Graphic Window DMA Control Register	LGWDCR	0x1002 1068

Table 26-6 provides an overview of the fields for all of the registers.

Table 26-6. LCDC Register Summary

Register Name	Register Location	Register Bits															
		31 (15)	30 (14)	29 (13)	28 (12)	27 (11)	26 (10)	25 (9)	24 (8)	23 (7)	22 (6)	21 (5)	20 (4)	19 (3)	18 (2)	17 (1)	16 (0)
LSSAR	0x1002 1000	Screen Start Address High – SSA H															
		Screen Start Address Low – SSA L															
LSR	0x1002 1004							Screen Width – XMAX									
								Screen Height – YMAX									
LVPWR	0x1002 1008																
								Virtual Page Width – VPW									
LCPR	0x1002 100C	CC			OP			Cursor X Position – CXP									
								Cursor Y Position – CYP									
LCWHBR	0x1002 1010	BK_EN			Cursor Width – CW								Cursor Height – CH				
										BD							
LCCMR	0x1002 1014															CUR_COL_R[5:4]	
		Cursor Red – CUR_COL_R[3:0]				Cursor Green – CUR_COL_G						Cursor Blue – CUR_COL_B					
LPCR	0x1002 1018	TFT	COLOR	Bus Width PBSIZ		Bits Per Pixel BPIX			PIX POL	FLM POL	LP POL	CLK POL	OE POL	SCLK IDLE	END_ SEL	END_ BYTE_ SWAP	REV _VS
		ACD SEL	Crystal Direction Toggle – ACD							SCLK SEL	SHARP	Pixel Clock Divider – PCD					
LHCR	0x1002 101C	Horizontal Sync Width – H_WIDTH															
		Horizontal Wait 1 – H_WAIT_1								Horizontal Wait 2 – H_WAIT_2							
LVCR	0x1002 1020	Vertical Sync Width – V_WIDTH															
		Vertical Wait 1 – V_WAIT_1								Vertical Wait 2 – V_WAIT_2							
LPOR	0x1002 1024																
													Panning Offset – POS				
LSCR	0x1002 1028	PS_RISE_DELAY								CLS_RISE_DELAY							
						REV_TOGGLE_DELAY				Gray 2				Gray 1			
LPCCR	0x1002 102C								CLS High Width								
		LD MSK					SCR		CC _EN	Pulse Width – PW							
LDCR	0x1002 1030	BURST											DMA High Mark – HM				
													DMA Trigger Mark – TM				
LRMCR	0x1002 1034																
																	SELF _REF
LICR	0x1002 1038																
													GW_ INT_ CON		INT SYN		INT CON
LIER	0x1002 103C																
										GW_U DR_E RR_E N	GW_ER R_RES_ EN	GW_ EOF _EN	GW_ BOF _EN	UDR_ ERR_ EN	ERR_ RES_ EN	EOF_E N	BOF_EN

Table 26-6. LCDC Register Summary (continued)

Register Name	Register Location	Register Bits															
		31 (15)	30 (14)	29 (13)	28 (12)	27 (11)	26 (10)	25 (9)	24 (8)	23 (7)	22 (6)	21 (5)	20 (4)	19 (3)	18 (2)	17 (1)	16 (0)
LISR	0x1002 1040									GW_UDR_ERR	GW_ERR_RES	GW_EOF	GW_BOF	UDR_ERR	ERR_RES	EOF	BOF
LGWSAR	0x1002 1050	Graphic Window Start Address High – GWSA H															
		Graphic Window Start Address Low – GWSA L															
LGWSR	0x1002 1054							Graphic Window Width – GWW									
								Graphic Window Height – GWH									
LGWVPWR	0x1002 1058																
		Graphic Window Virtual Page Width – GWVPW															
LGWPOR	0x1002 105C																
													Graphic Window Panning Offset – GWPO				
LGWPR	0x1002 1060	Graphic Window X Position – GWXP															
		Graphic Window Y Position – GWYP															
LGWCR	0x1002 1064	Graphic Window Alpha Value – GWAV								GWCKE	GWE	GW_RVS				Graphic Window Color Key Red – GWCKR[5:4]	
		Graphic Window Color Key Red – GWCKR[3:0]				Graphic Window Color Key Green – GWCKG						Graphic Window Color Key Blue – GWCKB					
LGWDCR	0x1002 1068	GWBT											Graphic Window High Mark – GWHM				
													Graphic Window Low Mark – GWLM				
BPLUT	0x1002 1800															(R[5:4])	
		First RAM Location of Background Plane LUT (R [3:0], G [5:0], B [5:0])															
	0x1002 1BFC															(R[5:4])	
		Last RAM Location of Background Plane LUT (R [3:0], G [5:0], B [5:0])															
GWLUT	0x1002 1C00																
		First RAM Location of Graphic Window LUT (R[3:0], G[5:0], B[5:0])															
	0x1002 1FFC															(R[5:4])	
		Last RAM location of Graphic Window LUT (R[3:0], G[5:0], B[5:0])															

26.2.1 LCDC Screen Start Address Register

The Screen Start Address Register specifies the start address of the LCD screen. See Figure 26-2 on page 26-3.

LSSAR — Screen Start Address Register — Addr 0x10021000

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SSA															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SSA															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-7. Screen Start Address Register Description

Name	Description
SSA Bits 31–2	**Screen Start Address of LCD Panel**—Holds pixel data for a new frame from the SSA address. This field must start at a location that enables a complete picture to be stored in a 4 Mbyte memory boundary (A [21:0]). A [31:22] has a fixed value for a picture's image.
Reserved Bits 1–0	Reserved—These bits are reserved and should read 0.

26.2.2 LCDC Size Register

The Size Register defines the height and width of the LCD screen.

LSR — LCDC Size Register — Addr 0x10021004

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							XMAX									
TYPE	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							YMAX									
TYPE	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-8. LCDC Size Register Description

Name	Description
Reserved Bits 31–26	Reserved—These bits are reserved and should read 0.
XMAX Bits 25–20	**Screen Width Divided by 16**—Holds screen x-axis size, divided by 16. For black-and-white panels (1 bpp), XMAX [20] is ignored, forcing the x-axis of the screen size to be a multiple of 32 pixels/line.
Reserved Bits 19–10	Reserved—These bits are reserved and should read 0.
YMAX Bits 9–0	**Screen Height**—Specifies the height of the LCD panel in terms of pixels or lines. The lines are numbered from 1 to YMAX for a total of YMAX lines.

26.2.3 LCDC Virtual Page Width Register

The Virtual Page Width Register defines the width of the virtual page for the LCD panel. See Figure 26-2 on page 26-3.

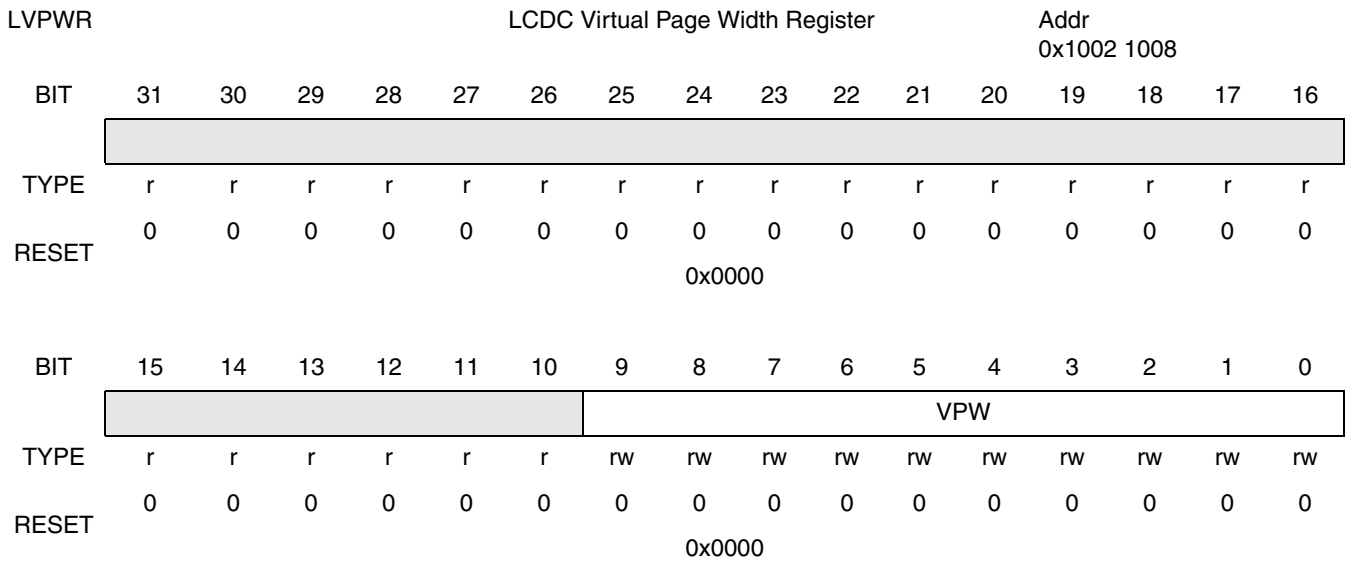


Table 26-9. LCDC Virtual Page Width Register Description

Name	Description
Reserved Bits 31–10	Reserved—These bits are reserved and should read 0.
VPW Bits 9–0	**Virtual Page Width**—Defines the virtual page width of the LCD panel. The VPW bits represent the number of 32-bit words required to hold the data for one virtual line. VPW is used in calculating the starting address representing the beginning of each displayed line.

26.2.4 LCDC Panel Configuration Register

The Panel Configuration Register defines all of the properties of the LCD screen.

LPCR — LCDC Panel Configuration Register — Addr 0x10021018

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TFT	COLOR	PBSIZ		BPIX			PIX POL	FLM POL	LP POL	CLK POL	OE POL	SCLK IDLE	END_ SEL	SWAP_ SEL	REV_ VS
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ACD SEL	ACD							SCLK SEL	SHARP	PCD					
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-10. LCDC Panel Configuration Register Description

Name	Description	Settings
TFT Bit 31	**Interfaces to TFT Display**—Controls the format and timing of the output control signals. Active and passive displays use different signal timing formats as described in Section 26.1.9, “Panel Interface Signals and Timing.” TFT also controls the use of the FRC in color mode. Please refer to Table 26-11 for TFT/COLOR setting usage.	0 = The LCD panel is a passive display 1 = The LCD panel is an active display: “digital CRT” signal format, FRC is bypassed. **Table 26-11. TFT/COLOR Settings** TFT / Color / LCD Display 0 / 0 / monochrome 0 / 1 / CSTN 1 / 0 / – 1 / 1 / TFT
COLOR Bit 30	**Interfaces to Color Display**—Activates three channels of FRC in passive mode to allow use of the special 2 2/3 pixels per output vector format. Please refer to Table 26-11 for TFT/COLOR setting usage.	0 = The LCD panel is a monochrome display 1 = The LCD panel is a color display
PBSIZ Bits 29–28	**Panel Bus Width**—Specifies the panel bus width. Applicable for monochrome or passive matrix color monitors. For passive color panels, only an 8-bit panel bus width is supported.	00 = 1-bit 10 = 4-bit 11 = 8-bit

Table 26-10. LCDC Panel Configuration Register Description (continued)

Name	Description	Settings
BPIX Bits 27–25	**Bits Per Pixel**—Indicates the number of bits per pixel in memory.	000 = 1 bpp, FRC bypassed 001 = 2 bpp 010 = 4 bpp 011 = 8 bpp 100 = 12 bpp (16 bits of memory used) 101 = 16 bpp 110 = 18bpp (32 bits of memory used) 111 = reserved **Note:** *To set normal 18bpp mode:* BPIX = 110 END_SEL = 0 SWAP_SEL = X (don't care) *To set Microsoft PAL_BGR 18bpp mode:* BPIX = 110 END_SEL = 1 SWAP_SEL = 1
PIXPOL Bit 24	**Pixel Polarity**—Sets the polarity of the pixels.	0 = Active high 1 = Active low
FLMPOL Bit 23	**First Line Marker Polarity**—Sets the polarity of the first line marker symbol.	0 = Active high 1 = Active low
LPPOL Bit 22	**Line Pulse Polarity**—Sets the polarity of the line pulse signal.	0 = Active high 1 = Active low
CLKPOL Bit 21	**LCD Shift Clock Polarity**—Sets the polarity of the active edge of the LCD shift clock.	0 = Active negative edge of LSCLK (in TFT mode, active on positive edge of LSCLK) 1 = Active positive edge of LSCLK (in TFT mode, active on negative edge of LSCLK)
OEPOL Bit 20	**Output Enable Polarity**—Sets the polarity of the output enable signal.	0 = Active high 1 = Active low
SCLKIDLE Bit 19	**LSCLK Idle Enable**—Enables/disables LSCLK when VSYNC is idle in TFT mode.	0 = Disable LSCLK 1 = Enable LSCLK
END_SEL Bit 18	**Endian Select**—Selects the image download into memory as big or little endian format.	0 = Little endian 1 = Big endian
SWAP_SEL Bit 17	**Swap Select**—LCDC operates in big endian mode internally. Swap Select controls the swap of data before operation in little endian mode.	0 = 16 bpp, 12 bpp mode 1 = 8 bpp, 4 bpp, 2 bpp, 1 bpp mode **Note:** When SWAP_SEL = 0, byte 3 (bits 31–24), byte 2 (bits 23–16), byte 1 (bits 15–8), byte 0 (bits 7–0) data swapped to byte 1, byte 0, byte 3 and byte 2 respectively. When SWAP_SEL = 1, byte 3, byte 2, byte 1, byte 0 data swapped to byte 0, byte 1, byte 2 and byte 3 respectively.

Table 26-10. LCDC Panel Configuration Register Description (continued)

Name	Description	Settings
REV_VS Bit 16	**Reverse Vertical Scan**—Selects the vertical scan direction as normal or reverse (the image flips along the x-axis). The SSA register must be changed accordingly.	0 = Vertical scan in normal direction 1 = Vertical scan in reverse direction
ACDSEL Bit 15	**ACD Clock Source Select**—Selects the clock source used by the alternative crystal direction counter.	0 = Use FRM as clock source for ACD count 1 = Use LP/HSYN as clock source for ACD count
ACD Bits 14–8	**Alternate Crystal Direction**—Toggles the ACD signal once every 1–16 FLM cycles based on the value specified in this field. The actual number of FLM cycles between toggles is the programmed value plus one.	For active mode (TFT=1), this parameter is not used. For passive mode (TFT=0), see description.
SCLKSEL Bit 7	**LSCLK Select**—Selects whether to enable or disable LSCLK in TFT mode when there is no data output.	0 = Disable OE and LSCLK in TFT mode when no data output 1 = Always enable LSCLK in TFT mode even if there is no data output
SHARP Bit 6	**Sharp Panel Enable**—Enables/disables signals for Sharp HR-TFT 240 x 320 panels.	0 = Disable Sharp signals 1 = Enable Sharp signals
PCD Bits 5–0	**Pixel Clock Divider**—Holds clock divider value. The LCDC_CLK (PERCLK3) is divided by N (PCD plus one) to yield the pixel clock rate. Values of 1 to 63 yield N=2 to 64. The pixel clock rate is faster than LSCLK by a factor equal to the data bus width for monochrome display. For all other displays, the pixel clock rate is the same as LSCLK.	The value of PCD must be set such that the LSCLK frequency is at least one third or one fourth of the HCLK frequency in TFT and CSTN modes respectively, otherwise the LD will be incorrect.

26.2.5 LCDC Horizontal Configuration Register

The Horizontal Configuration Register defines the horizontal sync pulse timing. For detail settings, please refer to MC9328MX21 data sheet.

LHCR — Horizontal Configuration Register — Addr 0x1002101C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	H_WIDTH															
TYPE	rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	0x0400															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	H_WAIT_1								H_WAIT_2							
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-12. LCDC Horizontal Configuration Register Description

Name	Description
H_WIDTH Bits 31–26	**Horizontal Sync Pulse Width**—Specifies the number of SCLK periods that HSYNC is activated. The active time is equal to (H_WIDTH + 1) of the SCLK periods.
Reserved Bits 25–16	Reserved—These bits are reserved and should read 0.
H_WAIT_1 Bits 15–8	**Wait Between OE and HSYNC**—In TFT mode, it specifies the number of SCLK periods between the end of OE signal and the beginning of the HSYNC. Total delay time equals (H_WAIT_1 + 1) of SCLK periods. In CSTN mode, it specifies the number of SCLK periods between the last display data and the beginning of the HSYNC signal. Total delay time equals (H_WAIT_1 + 1) of SCLK periods.
H_WAIT_2 Bits 7–0	**Wait Between HSYNC and Start of Next Line**—In TFT mode, it specifies the number of SCLK periods between the end of HSYNC and the beginning of the OE signal. Total delay time equals (H_WAIT_2 + 3). In CSTN mode, it specifies the number of SCLK periods between the end of HSYNC and the first display data in each line. Total delay time equals (H_WAIT_2 + 2) of SCLK periods.

26.2.6 LCDC Vertical Configuration Register

The Vertical Configuration Register defines the vertical sync pulse timing. For detail settings, please refer to MX9328MX1 data sheet.

LVCR — Vertical Configuration Register — Addr 0x10021020

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	V_WIDTH															
TYPE	rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	0x0400															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	V_WAIT_1								V_WAIT_2							
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-13. LCDC Vertical Configuration Register Description

Name	Description
V_WIDTH Bits 31–26	**Vertical Sync Pulse Width**—Specifies the width, in lines, of the VSYNC pulse for active (TFT =1) mode. For a value of “000001”, the vertical sync pulse encompasses one HSYNC pulse. For a value of “000002”, the vertical sync pulse encompasses two HSYNC pulses, and so on. For passive (TFT=0) mode and non-color mode, see Figure 26-13.
Reserved Bits 25–16	Reserved—These bits are reserved and should read 0.

Table 26-13. LCDC Vertical Configuration Register Description (continued)

Name	Description
V_WAIT_1 Bits 15–8	**Wait Between Frames 1**—Defines the delay, in lines, between the end of the OE pulse and the beginning of the VSYNC pulse for active (TFT=1) mode. This field has no meaning in passive non-color mode. The actual delay is (V_WAIT_1). In passive color mode, this field is the delay, measured in virtual clock periods, between the last line of the frame to the beginning of the next frame.
V_WAIT_2 Bits 7–0	**Wait Between Frames 2**—Defines the delay, in lines, between the end of the VSYNC pulse and the beginning of the OE pulse of the first line in active (TFT=1) mode. The actual delay is V_WAIT_2) lines. Set this field to zero for passive non-color mode. The minimum value of this field is 0x01.

26.2.7 LCDC Panning Offset Register

The Panning Offset Register sets up the panning for the image.

LPOR — Panning Offset Register — Addr 0x10021024

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												POS				
TYPE	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-14. LCDC Panning Offset Register Description

Name	Description	Settings
Reserved Bits 31–5	Reserved—These bits are reserved and should read 0.	
POS Bits 4–0	**Panning Offset**—Defines the number of bits that the data from memory is panned to the left before processing. POS is read by the LCDC once at the beginning of each frame. For example, in 4bpp mode, setting POS = 16 shifts 16bits of which means panning the image by 4 pixels left. **Note:** 1. Shifting data more than 32 bits should use LSSAR register setting. 2. 18 bpp panning should use LSSAR register setting.	To achieve panning of the final image by N bits: **Table 26-1.** **Bits Per Pixel** / **POS** / **Effective # of pixels Panned on Image** 1 / N / N 2 / 2N / N 4 / 4N / N 8 / 8N / N 12/16 / 16N / N

26.2.8 LCDC Cursor Position Register

The LCD Cursor Position Register is used to determine the starting position of the cursor on the LCD panel.

LCPR — LCDC Cursor Position Register — 0x1002100C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CC			OP			CXP									
TYPE	rw	rw	r	rw	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							CYP									
TYPE	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-15. LCDC Cursor Position Register Description

Name	Description	Settings
CC Bits 31–30	**Cursor Control**—Controls the format of the cursor and the type of arithmetic operations.	*When OP = 0:* 00 = Transparent, cursor is disabled 01 = 1 for non-color displays; color defined in LCDC Color Cursor Mapping Register for color displays 10 = Reversed, INV background for non-color displays; INV color defined in LCDC Color Cursor Mapping Register for color displays 11 = 0 for non-color displays; AND between background and cursor for color displays *When OP = 1, for color mode only:* 00 = Transparent, cursor is disabled 01 = OR between background and cursor 10 = XOR between background and cursor 11 = AND between background and cursor
Reserved Bit 29	Reserved—This bit is reserved and should read 0.	
OP Bit 28	**Arithmetic Operation Control**—Enables/disables arithmetic operations between the background and the cursor.	0 = Disable arithmetic operation 1 = Enable arithmetic operation
Reserved Bits 27–26	Reserved—These bits are reserved and should read 0.	
CXP Bits 25–16	**Cursor X Position**—Represents the cursor's horizontal starting position X in pixel count (from 0 to XMAX).	

Table 26-15. LCDC Cursor Position Register Description (continued)

Name	Description	Settings
Reserved Bits 15–10	Reserved—These bits are reserved and should read 0.	
CYP Bits 9–0	**Cursor Y Position**—Represents the cursor's vertical starting position Y in pixel count (from 0 to YMAX).	

26.2.9 LCDC Cursor Width Height and Blink Register

The LCD Cursor Width Height and Blink Register is used to determine the width and height of the cursor, and how it blinks.

LCWHB — LCDC Cursor Width Height and Blink Register — Addr 0x10021010

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BK_EN			CW								CH				
TYPE	rw	r	r	rw	rw	rw	rw	rw	r	r	r	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
	0x0101															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									BD							
TYPE	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	0x00FF															

Table 26-16. LCDC Cursor Width Height and Blink Register Description

Name	Description	Settings
BK_EN Bit 31	**Blink Enable**—Determines whether the blink enable cursor will blink or remain steady.	0 = Blink is disabled 1 = Blink is enabled
Reserved Bits 30–29	Reserved—These bits are reserved and should read 0.	
CW Bits 28–24	**Cursor Width**—Specifies the width of the hardware cursor in pixels.	This field can be any value between 1 and 31 (setting this field to zero disables the cursor)
Reserved Bits 23–21	Reserved—These bits are reserved and should read 0.	
CH Bits 20–16	**Cursor Height**—Specifies the height of the hardware cursor in pixels.	This field can be any value between 1 and 31 (setting this field to zero disables the cursor)

Table 26-16. LCDC Cursor Width Height and Blink Register Description (continued)

Name	Description	Settings
Reserved Bits 15–8	Reserved—These bits are reserved and should read 0.	
BD Bits 7–0	**Blink Divisor**—Sets the cursor blink rate. A 32 Hz clock from RTC is used to clock the 8-bit up counter. When the counter value equals BD, the cursor toggles on/off. Hence the larger the BD, the slower the cursor is blinking. The faster cursor blinking rate is when BD is 0.	

26.2.10 LCDC Color Cursor Mapping Register

The LCD Color Cursor Mapping Register defines the color of the cursor in passive or TFT color modes. If the bpp mode setting is smaller than 18bpp. The cursor color component bits should be put in the MSBs. For example, if the color cursor of RGB=(0x1a, 0x26, 0x05) is wanted, the CUR_COL_R, CUR_COL_G and CUR_COL_B should be set to 0x34, 0x26 and 0x0a.

LCCMR — LCDC Color Cursor Mapping Register — Addr 0x10021014

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															CUR_COL_R[5:4]	
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CUR_COL_R[3:0]				CUR_COL_G						CUR_COL_B					
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-17. LCDC Color Cursor Mapping Register Description

Name	Description	Settings
Reserved Bits 31–18	Reserved—These bits are reserved and should read 0.	
CUR_COL_R Bits 17–12	**Cursor Red Field**—Defines the red component of the cursor color in color mode.	000000 = No red … 111111 = Full red
CUR_COL_G Bits 11–6	**Cursor Green Field**—Defines the green component of the cursor color in color mode.	000000 = No green … 111111 = Full green
CUR_COL_B Bits 5–0	**Cursor Blue Field**—Defines the blue component of the cursor color in color mode.	000000 = No blue … 111111 = Full blue

26.2.11 LCDC Sharp Configuration Register

For 2 bpp modes, full black and full white are the two predefined display levels. The other two intermediate gray-scale shading densities can be adjusted within the LCDC Sharp Configuration Register. The LCDC Sharp Configuration register also controls the relative delay timing of the CLS, REV, and PS. For detail Sharp panel settings, please refer to MX9328MX1 data sheet. The TFT timing diagram that shows the relationship between these signals is shown in Figure 26-18 on page 26-33.

LSCR — LCDC Sharp Configuration Register — Addr 0x10021028

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PS_RISE_DELAY								CLS_RISE_DELAY							
TYPE	rw	rw	rw	rw	rw	rw	r	r	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0
	0x400c															

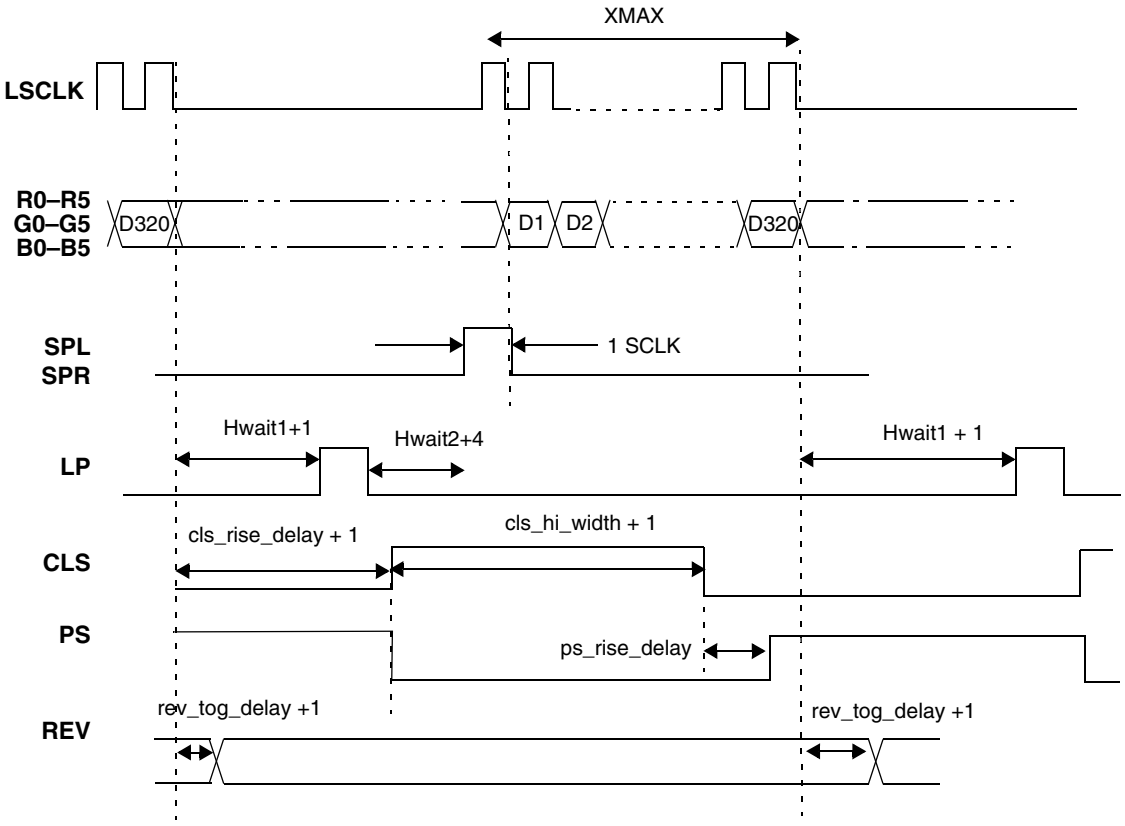
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					REV_TOGGLE_DELAY				GRAY 2				GRAY 1			
TYPE	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	1	1	0	1	1	1	0	0	1	1
	0x0373															

Table 26-18. LCDC Sharp Configuration Register Description

Name	Description	Settings
PS_RISE_DELAY Bits 31–26	**PS Rise Delay**—Controls the delay of the rising edge of PS relative to the falling edge of CLS. Total delay time equals to PS_RISE_DELAY SCLK periods.	00000 = 0 LSCLK period 00001 = 1 LSCLK period … 11111 = 63 LSCLK periods
Reserved Bits 25–24	Reserved—These bits are reserved and should read 0.	
CLS_RISE_DELAY Bits 23–16	**CLS Rise Delay**—Controls the delay of the rising edge of CLS relative to the last LD of the line. Total delay time equals to (CLS_RISE_DELAY + 1) SCLK periods	00000000 = 10 LSCLK periods 00000001 = 2 LSCLK period … 11111111 = 256 LSCCLK periods
Reserved Bits 15–12	Reserved—These bits are reserved and should read 0.	
REV_TOGGLE_DELAY Bits 11–8	**REV Toggle Delay**—Controls the transition delay of REV relative to the last LD of the line. Total delay time equals to (REV_TOGGLE_DELAY + 1) SCLK periods	0000 = 1 LSCLK period 0001 = 2 LSCLK period … 1111 = 16 LSCLK periods

Table 26-18. LCDC Sharp Configuration Register Description (continued)

Name	Description	Settings
GRAY 2 Bits 7–4	**Gray-Scale 2**—Represents one of the two gray-scale shading densities.	This field is programmable to any value between 0 and 16 (0 and 16 are already defined as two of the four colors).
GRAY 1 Bits 3–0	**Gray-Scale 1**—Represents the other gray-scale shading density.	This field is programmable to any value between 0 and 16 (0 and 16 are already defined as two of the four colors).



Falling edge of PS aligns with rising edge of CLS
The rising edge delay of PS is programmed by PS_RISE_DELAY
CLS_HI_WIDTH is equal to PWM_SCR0 • 256 + PWM_WIDTH in units of LSCLK.
SPL/SPR pulse width is fixed and aligned to the first data of the line.
REV toggles every LP period.

Figure 26-18. Horizontal Timing in MC9328MX21

26.2.12 LCDC PWM Contrast Control Register

The PWM Contrast Control Register is used to control the signal output at the contrast pin, which controls the contrast of the LCD panel.

LPCCR — LCDC PWM Contrast Control Register — Addr 0x1002102C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								CLS_HI_WIDTH								
TYPE	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LDMSK					SCR		CC_EN	PW							
TYPE	rw	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-19. LCDC PWM Contrast Control Register Description

Name	Description	Settings
Reserved Bits 31–25	Reserved—These bits are reserved and should read 0.	
CLS_HI_WIDTH Bit 24–16	**CLS High Pulse Width**—Controls the pulse width of CLS in units of SCLK. The actual pulse width = CLS_HI_WDITH +1.	
LDMSK Bit 15	**LD Mask**—Enables/disables the LD output to zero for the Sharp TFT panel power-off sequence.	0 = LD [15:0] is normal 1 = LD [15:0] always equals 0
Reserved Bits 14–11	Reserved—These bits are reserved and should read 0.	
SCR Bits 10–9	**Source Select**—Selects the input clock source for the PWM counter. The PWM output frequency is equal to the frequency of the input clock divided by 256.	00 = Line pulse 01 = Pixel clock 10 = LCD clock 11 = Reserved
CC_EN Bit 8	**Contrast Control Enable**—Enables/disables the contrast control function.	0 = Contrast control is off 1 = Contrast control is on
PW Bits 7–0	**Pulse-Width**—Controls the pulse-width of the built-in pulse-width modulator, which controls the contrast of the LCD screen.	

26.2.13 LCDC Refresh Mode Control Register

The Refresh Mode Control Register is used to control refresh characteristics.

LRMCR — LCDC Refresh Mode Control Register — Addr 0x10021034

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SELF_REF
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-20. LCDC Refresh Mode Control Register Description

Name	Description	Settings
Reserved Bits 31–1	Reserved—These bits are reserved and should read 0.	
SELF_REF Bit 0	**Self-Refresh**—Enables/disables self-refresh mode.	0 = Disable self-refresh 1 = Enable self-refresh

NOTE

1. On entering self-refresh mode, the LSCLK and LD [17:0] signals stay low. HYSN and VSYN operate normally.
2. Except for the SSA, BGLUT, and GWLUT registers, all configurations must be performed before enabling the LCDC to avoid a malfunction.
3. The SSA must always match the address range of the RAM selected. If the user wants to switch between various types of RAM, the LCDC must be disabled before switching.

26.2.14 LCDC DMA Control Register

There is a 32 × 32 bit line buffer in the LCDC that stores DMA data from system memory. The LCDC DMA Control register controls the DMA burst length and when to trigger a DMA burst in terms of the number of data bytes left in the pixel buffer.

LCDR — LCDC DMA Control Register — Addr 0x1002 1030

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BURST											HM				
TYPE	rw	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw
RESET	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	0x8010															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												TM				
TYPE	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	0x0004															

Table 26-21. LCDC DMA Control Register Description

Name	Description	Settings
BURST Bit 31	**Burst Length**—Determines whether the burst length is fixed or dynamic.	0 = Burst length is dynamic 1 = Burst length is fixed
Reserved Bits 30–21	Reserved—These bits are reserved and should read 0.	
HM Bits 20–16	**DMA High Mark**—Establishes the high mark for DMA requests. For dynamic burst length, after the DMA request is made, data is loaded and the pixel buffer continues to be filled until the number of empty words left in the DMA FIFO is equal to the high mark minus 2. The minimum HM setting in dynamic burst is 3. For fixed burst length, the burst length (in words) of each request is equal to the DMA high mark setting and its value must be larger than TM.	
Reserved Bits 15–5	Reserved—These bits are reserved and should read 0.	
TM Bits 4–0	**DMA Trigger Mark**—Sets the low-level mark in the pixel buffer to trigger a DMA request. The low-level mark equals the number of words left in the pixel buffer.	

NOTE

For SDRAM access, a fixed burst length of 8 is recommended:

fixed burst length = 1; high mark = 8; low mark = 4

For bus that is heavy loaded that requires SDRAM access, a dynamic burst length is recommended:

fixed burst length = 0; high mark = 4; low mark = 8

26.2.15 LCDC Interrupt Configuration Register

The Interrupt Configuration Register is used to configure the interrupt conditions.

LICR — LCDC Interrupt Configuration Register — Addr 0x10021038

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												GW_INT_CON		INT SYN		INT CON
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	rw	r	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-22. LCDC Interrupt Configuration Register Description

Name	Description	Settings
Reserved Bits 31–5	Reserved—These bits are reserved and should read 0.	
GW_INT_CON Bit 4	**Graphic Window Interrupt Condition**—Determines if an interrupt condition is set at the beginning or the end of graphic window condition.	0 = Interrupt flag is set when the end of graphic window is reached 1 = Interrupt flag is set when the beginning of graphic window is reached
Reserved Bit 3	Reserved—This bit is reserved and should read 0.	

Table 26-22. LCDC Interrupt Configuration Register Description (continued)

Name	Description	Settings
INTSYN Bit 2	**Interrupt Source**—Determines if an interrupt flag is set during last data/first data of frame loading or on last data/first data of frame output to the LCD panel. Please refer to Table 26-23 for INTSYN/INTCON setting usage. **Note:** There is a latency between loading the last/first data of frame to output to LCD panel.	0 = Interrupt flag is set on loading the last data/first data of frame from memory 1 = Interrupt flag is set on output of the last data/first data of frame to LCD panel **Table 26-23. INTSYN/INTCON Settings** INTSYN / INTCON / Description 0 / 0 / Interrupt flag is set on loading last data of frame from memory. 0 / 1 / Interrupt flag is set on loading first data of frame from memory. 1 / 0 / Interrupt flag is set on output of last data of frame to LCD panel. 1 / 1 / Interrupt flag is set on output of first data of frame to LCD panel.
Reserved Bit 1	Reserved—This bit is reserved and should read 0.	
INTCON Bit 0	**Interrupt Condition**—Determines if an interrupt condition is set at the beginning or the end of frame condition. Please refer to Table 26-23 for INTSYN/INTCON setting usage.	0 = Interrupt flag is set when the End of Frame (EOF) is reached 1 = Interrupt flag is set when the Beginning of Frame (BOF) is reached

26.2.16 LCDC Interrupt Enable Register

The LCDC Interrupt Enable Register is used to enable the LCDC interrupt pin generated to the ARM926EJ-S Interrupt Controller (AITC). When the interrupt is masked, the LCDC will not generate the interrupt request to AITC, but its status can still be observed in the Interrupt Status Register.

LIER — LCDC Interrupt Enable Register — Addr 0x1002103C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									GW_UDR_ERR_EN	GW_ERR_RES_EN	GW_EOF_EN	GW_BOF_EN	UDR_ERR_EN	ERR_RES_EN	EOF_EN	BOF_EN
TYPE	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-24. LCDC Interrupt Enable Register Description

Name	Description	Settings
Reserved Bits 31–8	Reserved—These bits are reserved and should read 0.	
GW_UDR_ERR_EN Bit 7	**Graphic Window Under Run Error Interrupt Enable**—Used to enable or mask the graphic window under-run error interrupt to AITC.	0 = mask interrupt 1 = enable interrupt
GW_ERR_RES_EN Bit 6	**Graphic Window Error Response Interrupt Enable**—Used to enable or mask the graphic window error response interrupt to AITC.	0 = mask interrupt 1 = enable interrupt
GW_EOF_EN Bit 5	**Graphic Window End of Frame Interrupt Enable**—Used to enable or mask the graphic window end of frame interrupt to AITC.	0 = mask interrupt 1 = enable interrupt
GW_BOF_EN Bit 4	**Graphic Window Beginning of Frame Interrupt Enable**—Used to enable or mask the graphic window beginning of frame interrupt to AITC.	0 = mask interrupt 1 = enable interrupt
UDR_ERR_EN Bit 3	**Under Run Error Interrupt Enable**—Used to enable or mask the background plane under-run error interrupt to AITC.	0 = mask interrupt 1 = enable interrupt
ERR_RES_EN Bit 2	**Error Response Interrupt Enable**—Used to enable or mask the background plane error response interrupt to AITC.	0 = mask interrupt 1 = enable interrupt
EOF_EN Bit 1	**End of Frame Interrupt Enable**—Used to enable or mask the background plane end of frame interrupt to AITC.	0 = mask interrupt 1 = enable interrupt
BOF_EN Bit 0	**Beginning of Frame Interrupt Enable**—Used to enable or mask the background plane beginning of frame interrupt to AITC.	0 = mask interrupt 1 = enable interrupt

26.2.17 LCDC Interrupt Status Register

The read-only Interrupt Status Register indicates whether an interrupt has occurred. The status bit is set whenever the interrupt condition is met. If any bit in this register is set and the corresponding bit in the LCDC Interrupt Enable Register is set, LCDC interrupt pin is asserted to AITC. The status bit is cleared by reading the register.

LISR — LCDC Interrupt Status Register — Addr 0x10021040

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									GW_UDR_ERR	GW_ERR_RES	GW_EOF	GW_BOF	UDR_ERR	ERR_RES	EOF	BOF
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-25. LCDC Interrupt Status Register Description

Name	Description	Settings
Reserved Bits 31–8	Reserved—These bits are reserved and should read 0.	
GW_UDR_ERR Bit 7	**Graphic Window Under Run Error**—Indicates whether the LCDC FIFO in graphic window plan has hit an under-run condition. This is when the data output rate is faster than the data input rate to the FIFO of the graphic window plan. Under-run can cause erroneous data output to LD. The LD data output rate must be adjusted to prevent this error.	0 = Interrupt has not occurred 1 = Interrupt has occurred
GW_ERR_RES Bit 6	**Graphic Window Error Response**—Indicates whether the LCDC has issued a read data request in graphic window and has received a response from the memory controller not equal to ‘OK.’ It is cleared by reading the status register, at power on reset, or when the LCDC is disabled.	0 = Interrupt has not occurred 1 = Interrupt has occurred
GW_EOF Bit 5	**Graphic Window End of Frame**—Indicates whether the end of graphic window has been reached. It is cleared by reading the status register, at power on reset, or when the LCDC is disabled.	0 = Interrupt has not occurred 1 = Interrupt has occurred
GW_BOF Bit 4	**Graphic Window Beginning of Frame**—Indicates whether the beginning of graphic window has been reached. It is cleared by reading the status register, at power on reset, or when the LCDC is disabled.	0 = Interrupt has not occurred 1 = Interrupt has occurred
UDR_ERR Bit 3	**Under Run Error**—Indicates whether the LCDC FIFO has hit an under-run condition. This is when the data output rate is faster than the data input rate to the FIFO. Under-run can cause erroneous data output to LD. The LD data output rate must be adjusted to prevent this error.	0 = Interrupt has not occurred 1 = Interrupt has occurred
ERR_RES Bit 2	**Error Response**—Indicates whether the LCDC has issued a read data request and has received a response from the memory controller not equal to OK. It is cleared by reading the status register, at power on reset, or when the LCDC is disabled.	0 = Interrupt has not occurred 1 = Interrupt has occurred

Table 26-25. LCDC Interrupt Status Register Description (continued)

Name	Description	Settings
EOF Bit 1	**End of Frame**—Indicates whether the end of frame has been reached. It is cleared by reading the status register, at power on reset, or when the LCDC is disabled.	0 = Interrupt has not occurred 1 = Interrupt has occurred
BOF Bit 0	**Beginning of Frame**—Indicates whether the beginning of frame has been reached. It is cleared by reading the status register, at power on reset, or when the LCDC is disabled.	0 = Interrupt has not occurred 1 = Interrupt has occurred

26.2.18 LCDC Graphic Window Start Address Register

The LCDC Graphic Window Start Address Register defines the starting address of the graphic window image. See Figure 26-3 on page 26-4.

LGWSAR — LCDC Graphic Window Start Address Register — Addr 0x10021050

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GWSA															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GWSA															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-26. LCDC Graphic Window Start Address Register Description

Name	Description
GWSA Bits 31–2	**Graphic Window Start Address on LCD Screen**—Holds the starting address of the graphic window picture. This field must start at a location that enables a complete graphic window picture to be stored in a 4 Mbyte memory boundary (A[21:0]). A[31:22] has a fixed value for the graphic window picture's image.
Reserved Bits 1–0	Reserved—These bits are reserved and should read 0.

26.2.19 LCDC Graphic Window Size Register

The LCDC Graphic Window Size Register defines the height and width of the graphic window on LCD screen.

LGWSR — LCDC Graphic Window Size Register — Addr 0x10021054

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							GWW									
TYPE	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							GWH									
TYPE	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-27. LCDC Graphic Window Size Register Description

Name	Description
Reserved Bits 31–26	Reserved—These bits are reserved and should read 0.
GWW Bits 25–20	**Graphic Window Width Divided by 16**—Holds graphic window x-axis size, divided by 16. For black-and-white panels (1 bpp), GW_XMAX [20] is ignored, forcing the x-axis of the screen size to be a multiple of 32 pixels/line. Please note that graphic window size cannot be set to 0.
Reserved Bits 19–10	Reserved—These bits are reserved and should read 0.
GWH Bits 9–0	**Graphic Window Height**—Specifies the height of the graphic window in terms of pixels or lines. The lines are numbered from 1 to GW_YMAX for a total of GW_YMAX lines. Please note that the graphic window size cannot be set to 0.

26.2.20 LCDC Graphic Window Virtual Page Width Register

The Graphic Window Virtual Page Width Register defines the width of the virtual page for the graphic window picture on LCD screen. See Figure 26-3 on page 26-4.

LGWVPWR — LCDC Graphic Window Virtual Page Width Register — Addr 0x10021058

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							GWVPW									
TYPE	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-28. LCDC Graphic Window Virtual Page Width Register Description

Name	Description
Reserved Bits 31–10	Reserved—These bits are reserved and should read 0.
GWVPW Bits 9–0	**Graphic Window Virtual Page Width**—Defines the virtual page width of the graphic window picture. The VPW bits represent the number of 32-bit words required to hold the data for one virtual line. GWVPW is used in calculating the starting address representing the beginning of each line of the graphic window picture.

26.2.21 LCDC Graphic Window Panning Offset Register

The Panning Offset Register sets up the panning for the image.

LGWPOR — LCDC Graphic Window Panning Offset Register — Addr 0x1002105C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												GWPO				
TYPE	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-29. LCDC Graphic Window Panning Offset Register Description

Name	Description	Settings
Reserved Bits 31–5	Reserved—These bits are reserved and should read 0.	
GWPO Bits 4–0	**Graphic Window Panning Offset**—Defines the number of bits that the graphic window data from memory is panned to the left before processing. POS is read by the LCDC once at the beginning of each frame. For example, in 4 bpp mode, setting POS = 16 shifts 16 bits which means panning the image by 4 pixels left. **Note:** 1. Shifting data more than 32 bits in graphic window should use LGWSAR register setting. 2. 18 bpp graphic window panning should use LGWSAR register setting.	To achieve panning of the final image by N bits: **Table 26-2.** **Bits Per Pixel** / **GWPO** / **Effective number of pixels panned** 1 / N / N 2 / 2N / N 4 / 4N / N 8 / 8N / N 12/16 / 16N / N

26.2.22 LCDC Graphic Window Position Register

The LCDC Graphic Window Position Register is used to determine the starting position of the graphic window on the LCD panel.

LGWPR — LCDC Graphic Window Position Register — Addr 0x10021060

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							GWXP									
TYPE	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							GWYP									
TYPE	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-30. LCDC Graphic Window Position Register Description

Name	Description
Reserved Bits 31–26	Reserved—These bits are reserved and should read 0.
GWXP Bits 25–16	**Graphic Window X Position**—Represents the graphic window's horizontal starting position in pixel count (from 0 to XMAX).
Reserved Bits 15–10	Reserved—These bits are reserved and should read 0.
GWYP Bits 9–0	**Graphic Window Y Position**—Represents the graphic window's vertical starting position in line (from 0 to YMAX).

26.2.23 LCDC Graphic Window Control Register

The LCD Color Cursor Mapping Register defines the color of the cursor in passive or TFT color modes.

NOTE

The graphics window can only be enabled while the HCLK to the LCDC is disabled. The graphics window can be disabled at any time

LGWCR — LCDC Graphic Window Control Register — 0x10021064

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GWAV								GWCKE	GWE	GW_RVS				GWCKR	
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GWCKR				GWCKG						GWCKB					
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 26-31. LCDC Graphic Window Control Register Description

Name	Description	Settings
GWAV Bits 31–24	**Graphic Window Alpha Value**—Defines the alpha value of graphic window used for alpha blending between graphic window and background plane	0 = Graphic window totally transparent—that is, not displayed on LCD screen 1= Graphic window totally opaque—that is, completely visible on LCD screen
GWCKE Bit 23	**Graphic Window Color Keying Enable**—Enable or disable graphic window color keying.	0 = Disable color keying of graphic window 1 = Enable color keying of graphic window
GWE Bit 22	**Graphic Window Enable**—Enable or disable graphic window displayed on screen.	0 = Disable graphic window on screen 1 = Enable graphic window on screen
GW_RVS Bit 21	**Graphic Window Reverse Vertical Scan**—Selects the graphic window vertical scan direction as normal or reverse (graphic window image flips along the x-axis). LGWSAR must change accordingly.	0 = Vertical scan in normal direction 1 = Vertical scan in reverse direction
Reserved Bits 20–18	Reserved—These bits are reserved and should read 0.	
GWCKR Bits 17–12	**Graphic Window Color Keying Red Component**—Defines the red component of graphic window color keying.	000000 = No red … 111111 = Full red
GWCKG Bits 11–6	**Graphic Window Color Keying Green Component**—Defines the green component of graphic window color keying.	000000 = No green … 111111 = Full green
GWCKB Bits 5–0	**Graphic Window Color Keying Blue Component**—Defines the blue component of graphic window color keying.	000000 = No blue … 111111 = Full blue

26.2.24 LCDC Graphic Window DMA Control Register

There is a 32 × 32 bit line buffer in the LCDC that stores graphic window data from system memory. The Graphic Window DMA Control Register controls the DMA burst length and when to trigger a DMA burst in terms of the number of data bytes left in the pixel buffer.

LGWDCR — LCDC Graphic Window DMA Control Register — Addr 0x10021068

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GWBT											GWHM				
TYPE	rw	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw
RESET	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	0x8010															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												GWTM				
TYPE	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	0x0004															

Table 26-32. LCDC Graphic Window DMA Control Register Description

Name	Description	Settings
GWBT Bit 31	**Graphic Window DMA Burst Type**—Determines whether the burst length is fixed or dynamic in graphic window plane.	0 = Burst length is dynamic 1 = Burst length is fixed
Reserved Bits 30–21	Reserved—These bits are reserved and should read 0.	
GWHM Bits 20–16	**Graphic Window DMA High Mark**—Establishes the high mark for DMA requests. For dynamic burst length, once the DMA request is made, data is loaded and the graphic window FIFO continues to be filled until the number of empty words left in the graphic window FIFO is equal to the high mark minus 2. The minimum HM setting in dynamic burst is 3. For fixed burst length, the burst length (in words) of each request is equal to the DMA high mark setting and its value must be larger than TM.	
Reserved Bits 15–5	Reserved—These bits are reserved and should read 0.	
GWTM Bits 4–0	**Graphic Window DMA Low Mark**—Sets the low level mark in the graphic window FIFO to trigger a DMA request. The low level mark equals the number of words left in the pixel buffer.	

26.2.25 BGLUT and GWLUT

There are two separate mapping RAMs in the LCD controller, the background lookup table (BGLUT) and the graphic window lookup table (GWLUT). The BGLUT is for the background plane and the mapping table is addressable from 0x10021800–0x10021BFC. The GWLUT is for the graphic window and its mapping table is addressable from 0x10021C00–0x10021FFC.

These mapping RAMs are used for mapping 4-bit codes for grayscale to 16 gray shades, and for mapping 4-bit color and 8-bit color to 16 colors and 256 colors, respectively, out of either a palette of 4096 (passive panels) or a palette of 256K (active panels).

The mapping RAM contains 256 entries and each entry is 18 bits wide. Each RAM entry uses 4 bytes of address space. The RAM is accessed with word transactions only and the address must be word aligned. Unimplemented bits are read as 0. Byte or halfword access to the RAM corrupts its contents. All read and write data use the least significant 12 or 18 bits.

In 4 bpp mode, the first sixteen RAM entries are used. In 8 bpp mode, all 256 RAM entries are used. The color RAM is not initialized at reset. Only the following settings use the mapping RAMs:

- 4 bpp gray-scale mode
- 4 bpp passive matrix color mode
- 8 bpp passive matrix color mode
- 4 bpp active matrix color mode
- 8 bpp active matrix color mode

26.2.25.1 Four Bits Per Pixel Gray-Scale Mode

In four bits per pixel gray-scale mode, a 4-bit code represents a gray-scale level. The first 16 mapping RAM entries must be written to define the codes for all 16 combinations.

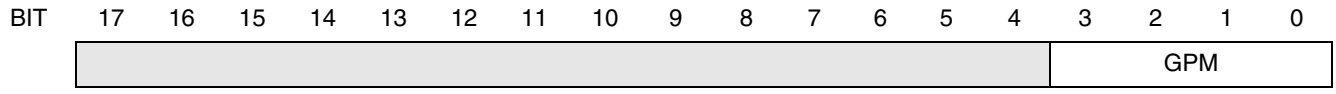


Table 26-33. Four Bits Per Pixel Gray-Scale Mode

Name	Description
Reserved Bits 11–4	Reserved—These bits are reserved and should read 0.
GPM Bits 3–0	**Gray Palette Map**—Represents the gray-scale level for a given pixel code.

26.2.25.2 Four Bits Per Pixel Passive Matrix Color Mode

In four bits per pixel passive matrix color mode, a 4-bit code represents a 12-bit color. Because just four bits are used to encode the color, a maximum of 16 colors can be selected out of a palette of 4096. The first 16 mapping RAM entries must be written to define the codes for the 16 available combinations.

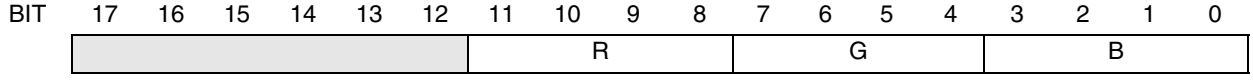


Table 26-34. Four Bits Per Pixel Passive Matrix Color Mode

Name	Description
R Bits 11–8	**Red Level (color display)**—Represents the red component level in the color.
G Bits 7–4	**Green Level (color display)**—Represents the green component level in the color.
B Bits 3–0	**Blue Level (color display)**—Represents the blue component level in the color.

26.2.25.3 Eight Bits Per Pixel Passive Matrix Color Mode

In eight bits per pixel passive matrix color mode, an 8-bit code represents a 12-bit color. Because eight bits are used to encode the color, a maximum of 256 colors can be selected out of a palette of 4096. All 256 mapping RAM entries must be written to define the codes for the 256 available combinations.

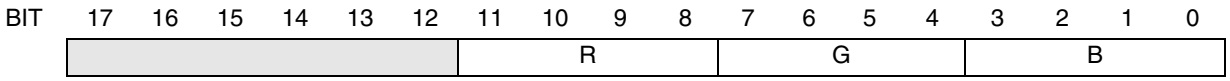


Table 26-35. Eight Bits Per Pixel Passive Matrix Color Mode

Name	Description
R Bits 11–8	**Red Level (color display)**—Represents the red component level in the color.
G Bits 7–4	**Green Level (color display)**—Represents the green component level in the color.
B Bits 3–0	**Blue Level (color display)**—Represents the blue component level in the color.

26.2.25.4 Four Bits Per Pixel Active Matrix Color Mode

In four bits per pixel active color mode, a 4-bit code represents a 18-bit color. Because just four bits are used to encode the color, a maximum of 16 colors can be selected out of a palette of 256K. The first 16 mapping RAM entries must be written to define the codes for the 16 available combinations.

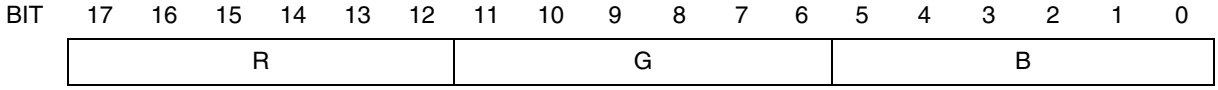


Table 26-36. Four Bits Per Pixel Active Matrix Color Mode

Name	Description
R Bits 17–12	**Red Level (color display)**—Represents the red component level in the color.
G Bits 11–6	**Green Level (color display)**—Represents the green component level in the color.
B Bits 5–0	**Blue Level (color display)**—Represents the blue component level in the color.

26.2.25.5 Eight Bits Per Pixel Active Matrix Color Mode

In eight bits per pixel active color mode, an 8-bit code represents an 18-bit color. Because eight bits are used to encode the color, a maximum of 256 colors can be selected out of a palette of 256K. All 256 mapping RAM entries must be written to define the codes for the 256 available combinations.

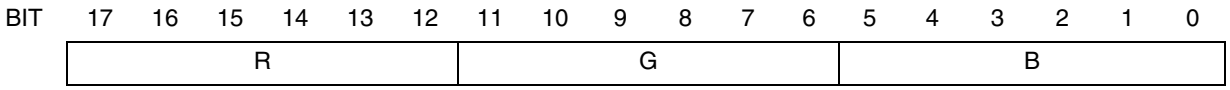


Table 26-37. Eight Bits Per Pixel Active Matrix Color Mode

Name	Description
R Bits 17–12	**Red Level (color display)**—Represents the red component level in the color.
G Bits 11–6	**Green Level (color display)**—Represents the green component level in the color.
B Bits 5–0	**Blue Level (color display)**—Represents the blue component level in the color.

Chapter 27
Smart Liquid Crystal Display Controller (SLCDC)

The Smart Liquid Crystal Display Controller module transfers data from the display memory buffer to the external display device. Direct Memory Access (DMA) transfers the data transparently with minimal software intervention. Bus utilization of the DMA is controllable and deterministic.

As cellular phone displays become larger and more colorful, demands on the processor increase. More CPU power is needed to render and manage the image. The role of the display controller is to reduce the CPU's involvement in the transfer of data from memory to the display device so the CPU can concentrate on image rendering. Direct Memory Access (DMA) is used to optimize the transfer. Embedded control information needed by the display device is automatically read from a second buffer in system memory and inserted into the data stream at the proper time to completely eliminate the CPU's role in the transfer.

A typical scenario for a cellular phone display is to have the display image rendered in main system memory. After the image is complete, the CPU triggers the SLCDC module to transfer the image to the display device. Image transfer is accomplished by burst DMA which steals bus cycles from the CPU. Cycle stealing behavior is programmable so bus use is kept within predefined bounds. After the transfer is complete, a maskable interrupt is generated indicating the status. For animated displays it is suggested that a two-buffer ping-pong scheme be implemented so that the DMA is fetching data from one buffer while the next image is rendered into the other.

Several display sizes and types are used in the various products that use the SLCDC. The SLCDC module has the capability of directly interfacing to the selected display devices. Both serial and parallel interfaces are supported. The SLCDC module only supports writes to the display controller. SLCDC read operations from the display controller are not supported.

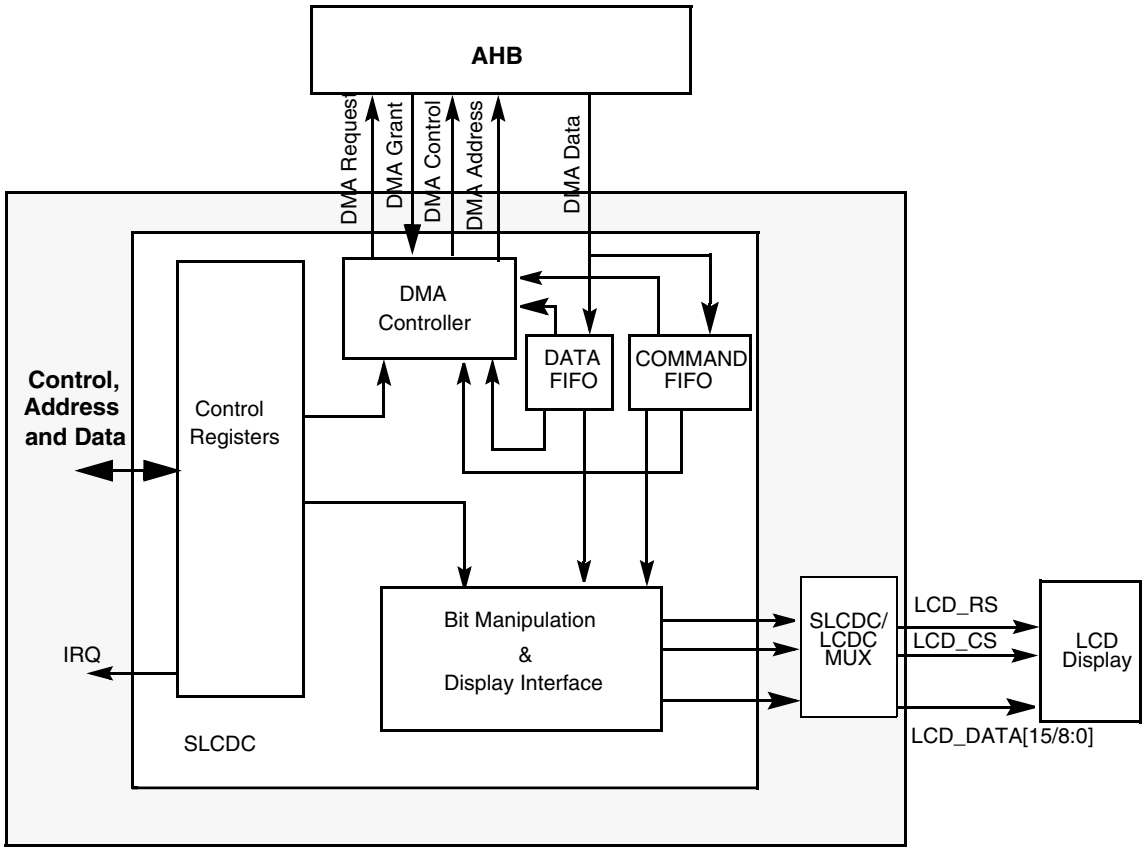


Figure 27-1. SLCDC System Diagram

27.1 SLCDC Module Pin List

Table 27-1 is a list of the SLCDC module pins.

Table 27-1. SLCDC Module Pin List

Pin Name	Direction	Description
LCD Interfaces		
SLCDC_LCD_DATA[15:0]	Output	Data bus used to write information to external LCD controller.
SLCDC_LCD_CS	Output	Used as a chip select for the external display controller in serial mode. In parallel mode, used as write strobe for the external display controller. The polarity of this signal is programmable.
SLCDC_LCD_RS	Output	LCD Register Select signal that indicates to the LCD device whether the data being written is display data or control data. The polarity of this signal is programmable.

27.2 Functional Description

The purpose of the SLCDC is to transparently and efficiently transfer image data from system memory to an external LCD controller. The system memory can be either internal or external memory. Figure 27-1 shows the block diagram of the SLCDC module within the system. The SLCDC module contains a DMA

controller that is designed to operate as an alternate master. The SLCDC DMA interface requests control of the AHB to do 32-bit reads from system memory.

The purpose of the DMA within the SLCDC is to transfer image and control data from system memory to the SLCDC FIFO where it is formatted and sent out to the external LCD controller. The DMA only performs read accesses from system memory. The data is collected in 32-bit words and stored in two internal FIFOs. Data is pulled from the FIFOs, as needed, and put in the correct format for the external LCD controller.

The SLCDC has both control and status registers which are accessible via the IP bus. These registers are used to store information about the type of LCD controller attached to the system.

The SLCDC can be configured to write image data to an external LCD controller via a 4- line serial, 3-line serial, an 8- or 16-bit parallel interface.

During automatic transfers, the SLCDC is responsible for properly sequencing display data and control strings in a data stream that is sent to the LCD controller to fill its internal display RAM. Display data is typically written to the external controller in pages. Each write to the controller fills one column of the current page. The SLCDC assumes that the controller automatically increments its display RAM pointers after each write.

NOTE

Changing the SLCDC configuration in the middle of a transfer can cause corruption of the data stream output to the LCD. The SLCDC configuration must only be changed when the GO bit and BUSY bit of the SLCDC control/status register are cleared.

27.2.1 Word Size Definition

Due to the increased complexity of color displays, the SLCDC supports 16- or 8-bit transfers of command or display data. This is controlled by the WRD_DEF_COM and WRD_DEF_DAT bits in the LCD transfer configuration register. All registers in the SLCDC are defined in words.

WRD_DEF_DAT controls the definition of word for all data registers:

- DATA_BUFFER_SIZE
- LCD_CONFIG

WRD_DEF_COM controls the definition of word for all command registers:

- COMMAND_BUFFER_SIZE

WRD_DEF_WRITE controls the definition of LCDDAT for the register:

- LCD_WRITE_DATA

27.2.2 Image Endianess

In the examples of automatic transfers shown in this document all of the memory images are either big endian, or 32-bit little endian. However, it is possible that the user may have an image that is stored in half word (16-bit) or byte (8-bit little endian). The SLCDC can compensate for this by the IMGEND bits in the

LCD_TRANSFER_CONFIG register. These bits cause the SLCDC to convert the 16- or 8-bit little endian image to a big endian image. The resultant big endian decode is used for the rest of the data processing.

Table 27-2. Image Endianess

IMGEND	Conversion	32-Bit Word Data Stored in Memory	32-Bit Data After Conversion
00	Big endian or 32-bit little endian to big endian	0x12345678	0x12345678
01	16-bit little endian to big endian	0x56781234	0x12345678
10	8-bit little endian to big endian	0x78563412	0x12345678

27.2.3 Accessing the LCD Controller

The SLCDC provides two methods of accessing the external display device. The first, and preferred method, is through automatic writes handled by the SLCDC. The second, is through direct IP register access via the LCD write data register.

27.2.3.1 Automatic SLCDC Transfers

The SLCDC is designed to take a block of data from system memory and write it, without software intervention, to an external display controller. This is done by giving the SLCDC a starting address and a data block size (in words). This method of data transfer can be used for both LCD display data as well as command data. Transfers are initiated by setting the GO bit in the SLCDC control/status register. The data transfer is done in the background. Upon completion, a maskable interrupt is generated.

Automatic transfers are accomplished in one of three ways:

- A block of data is transferred to the LCD controller with control strings automatically inserted to navigate through the LCD controller's internal RAM. This mode makes use of a second buffer in system memory to store the software commands necessary for LCD controller RAM addressing.
- A block of data is transferred from system memory to the display controller while tying the LCD_RS pin low.
- A block of data is transferred from system memory to the display controller while tying the LCD_RS pin high.

The automatic transfer mode control bits (AUTOMODE[1:0]) of the SLCDC control/status register configure the SLCDC for one of the three supported automatic transfer modes.

27.2.3.1.1 Automatic Display Data Transfers (AUTOMODE[1:0][1]=10)

The SLCDC is designed to require no software intervention when transferring a display frame buffer from system memory to the external LCD controller. To accomplish this, the SLCDC must automatically transmit control strings for LCD controller RAM addressing at appropriate times in the frame buffer data stream. LCD controller RAMs are typically organized into pages. After the page of RAM is filled, the controller's page address must be set to the next page. The SLCDC contains counters that monitor the number of display data words written to the LCD controller to determine when the current page of

1. Automode [1:0] = 11 is reserved.

controller RAM is filled. After the page is filled, the SLCDC inserts a control string into the data stream to set the controller RAM's page address to the next page. The SLCDC requires that this control string be stored in a command buffer in system memory. The organization of the command buffer is discussed later in this section.

To correctly move a complete frame of image data from system memory to the LCD controller, the SLCDC must be programmed with the following LCD information:

- Command word definition (8- or 16-bit words)
- Data word definition (8- or 16-bit words)
- Number of words in a page of LCD image data
- Number of words in the LCD frame data buffer
- Type of LCD interface (serial or parallel)
- Data clock polarity of the LCD (for serial interfaces only)
- Endianess of the image (big endian, little 32-bit, little 16-bit, or little 8-bit)
- Chip-select polarity for the LCD
- Transfer clock speed requirements of the LCD
- Control string needed at the start of each page of LCD display data (stored in a separate command buffer in system memory)

The SLCDC uses the number of words per page of LCD image data information (stored in the WORDPPAGE[12:0] bits of the LCD CONFIG register) to determine when a set of addressing commands must be sent to the LCD. When the current display RAM page is full, the SLCDC must increment the page address and reset the column address of the LCD controller to 0.

This is accomplished by reading the next control string from the command buffer and transmitting it to the LCD controller. The SLCDC continues to send display data and command strings as needed until the controller's display RAM is filled. The transfer is finished when the number of data words transferred equals the number of words in the buffer as defined by the DATABUFSIZ bits in the DATA BUFFER SIZE registers.

Figure 27-2 on page 27-7 shows an example of how the external LCD controller RAM is filled during automatic SLCDC writes for a monochrome display. Command and data word sizes are both defined as bytes. In this example, the LCD controller requires three "command" words (bytes) to set the page and column address. Setting the GO bit of the control/status register begins the transfer of data from system memory to the LCD controller. The first word written to the LCD sets the page address to 0. The next two words written are control strings that set the LCD column address to 0. Word write #4 starts the transfer of image data to the LCD. The first word of display data maps to column 0 of the display. The most-significant bit of the word maps to row 7 and the LSB maps to row 0.

Words of display data continue to be written to the LCD until the SLCDC's internal word counter equals the value stored in the WORDPPAGE[12:0] bits of the LCD CONFIG register. At this point, the SLCDC issues a command string, taken from the command buffer, to increment the page address and reset the column address of the LCD.

This sequence continues until the LCD controller's data RAM is filled. After the display buffer has been filled, the IRQ bit of the SLCDC control/status register becomes set. Also, the BUSY bit and GO bit clear. An interrupt is generated if the IRQEN bit is set and the system is configured for SLCDC interrupts.

Figure 27-3 on page 27-8 shows the mapping for a 2-bit color/gray scale display. Command and data word sizes are defined as bytes. For this configuration, each column of data within the page requires 2 word writes. The sequencing remains the same as the 1-bit per pixel case show in Figure 27-2 on page 27-7, however, twice as many display data writes are required to fill the page.

The SLCDC is designed to deal with a large variety of LCD controllers from multiple vendors. While the hardware interface to the external controller is somewhat standard, the software interface can vary from chip to chip. To deal with the software differences between LCD controllers, the SLCDC makes use of a separate command buffer stored in system memory to go along with the display data buffer. Figure 27-4 on page 27-9 shows the organization of the information in system memory.

The LCD page and column command buffer should be set up to contain the commands necessary to navigate through the LCD controller's internal RAM. Each command in the buffer is preceded by a tag. The SLCDC uses the tags to determine the value of the LCD_RS pin during the transmission of the command word.

Figure 27-5 on page 27-10 shows how the commands and tags are organized in system memory. In this example, the command string length is 3. That is, three words are required to be transmitted to the LCD controller at the start of each page of LCD RAM. Each of the command words have a tag placed adjacent to it in memory. The command is placed immediately after its tag in memory. Currently the SLCDC only uses the least-significant bit of the tag field. This bit, labeled "RS" in Figure 27-5, sets the value of the LCD_RS pin while the command word is being transferred to the LCD controller. If the RS tag bit is set to "1", LCD_RS is driven to 1 during the transfer of the corresponding command byte. Similarly, if the RS tag bit is set to "0", LCD_RS is driven to 0 during the transfer of the corresponding command byte.

Figure 27-6 on page 27-11 shows a 16-bit command and 16-bit data transaction for a 16-bit color display. In this example the command length is 3. Each LCD data transfer is for one display pixel. The mapping of display pixels for 16-bit color is generally "RRRRGGGGBBBBXXX" (R=red G=green B=blue X=ignored). Check your display to see what the 16-bit color mapping is. In this case a page consists of a row of pixels. Three 16-bit words of command data are transferred to the display first, then 16-bit transfers of data will occur until the WORDPPAGE are transferred. Three more command words are sent designating the next page, and the cycle repeats itself.

Figure 27-7 on page 27-12 shows how a 16-bit command word is stored in system memory. The 16-bit command is stored in the least significant half-word of memory (15–0). The tag is stored in the most significant half-word of memory (31–16). Currently only bit 16 of the tag is used for RS.

Figure 27-8 on page 27-13 shows how 16-bit word commands and 16-bit word data is stored in system memory. Because command data is 17 bits long (16 bits of command + 1-bit tag), each command takes up 1 word of system memory. Each 16-bit word of data can be placed in a half-word of system memory.

Figure 27-9 and Figure 27-10 show the sequence in which the words of data are written to the LCD controller when AUTOMODE[1:0] =10. These figures illustrate how command strings, taken from the command buffer, are interleaved with display data taken from the data buffer.

It is important to note that different combinations of word definitions are allowed. For example, there can be 8-bit command words and 16-bit data words. In this case, the commands would be stored in system memory much like Figure 27-5, and data would be stored like Figure 27-8. Refer to Chapter 27.4, “LCD Controller Interface,” to determine how mixed transfers occur.

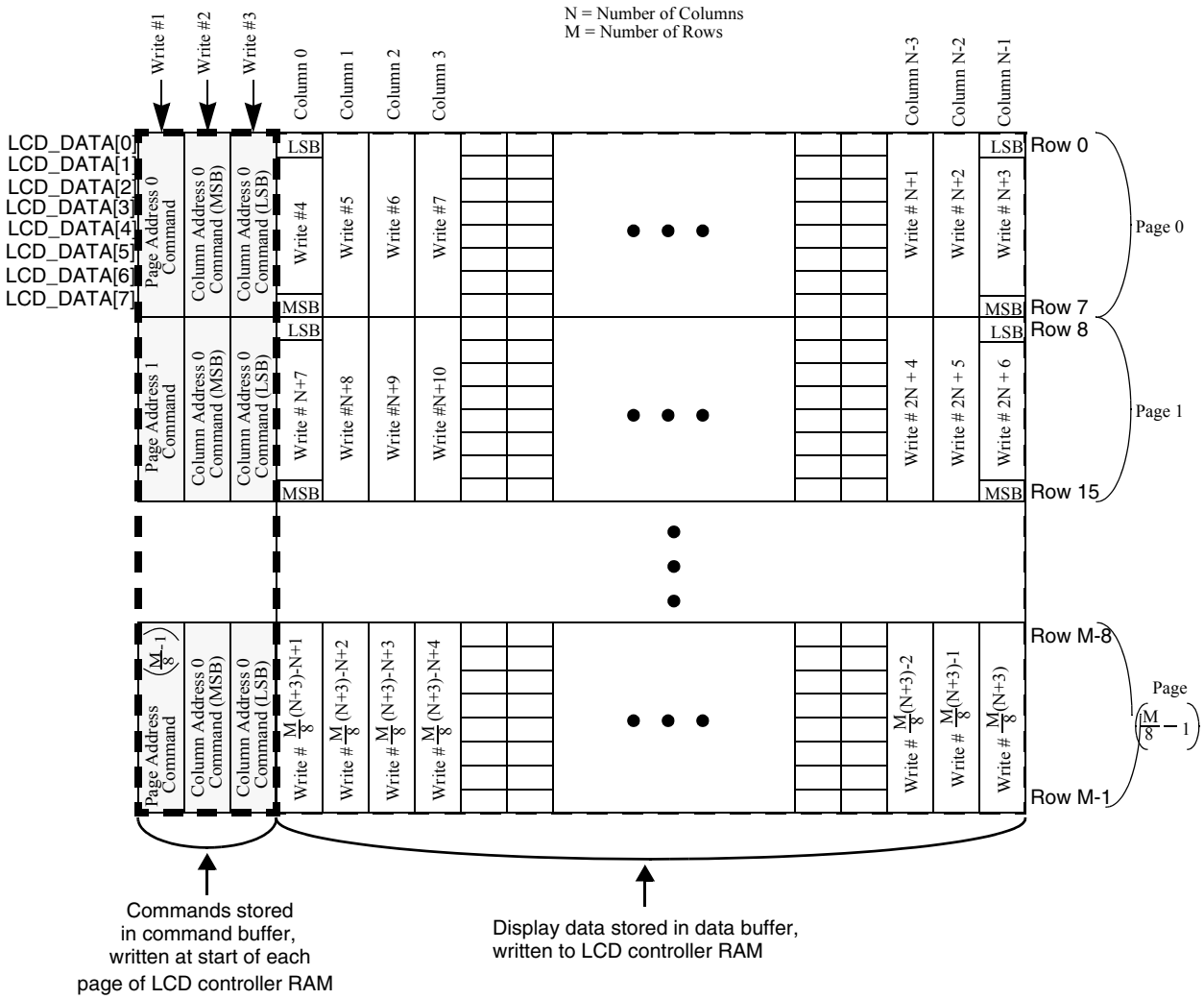


Figure 27-2. Sample LCD Controller Memory Mapping (Monochrome) (8-bit Command/8-bit Data)

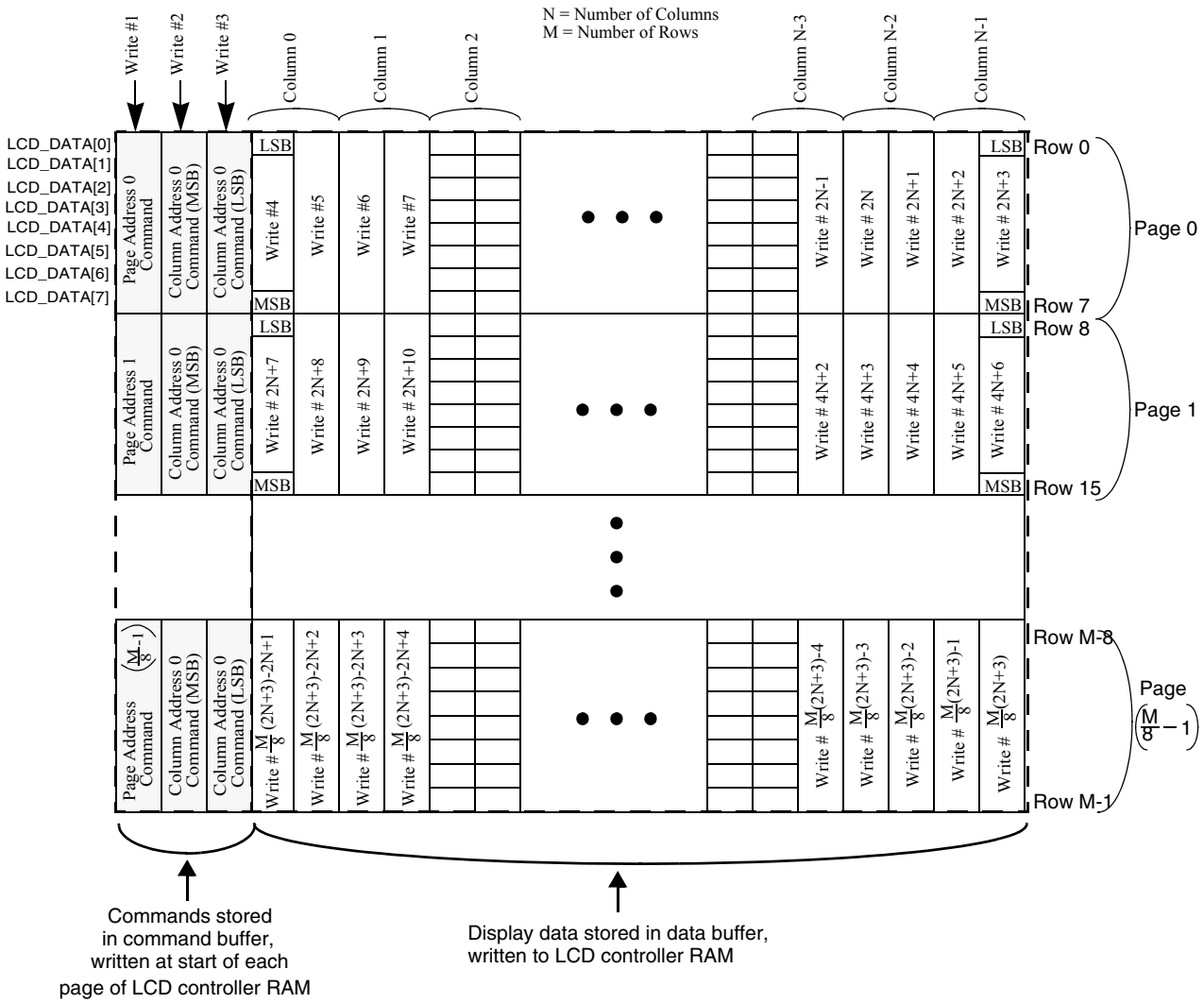


Figure 27-3. SLCDC LCD Controller Memory Mapping (2-bit Color/Gray Scale) (8-bit Command/8-bit Data)

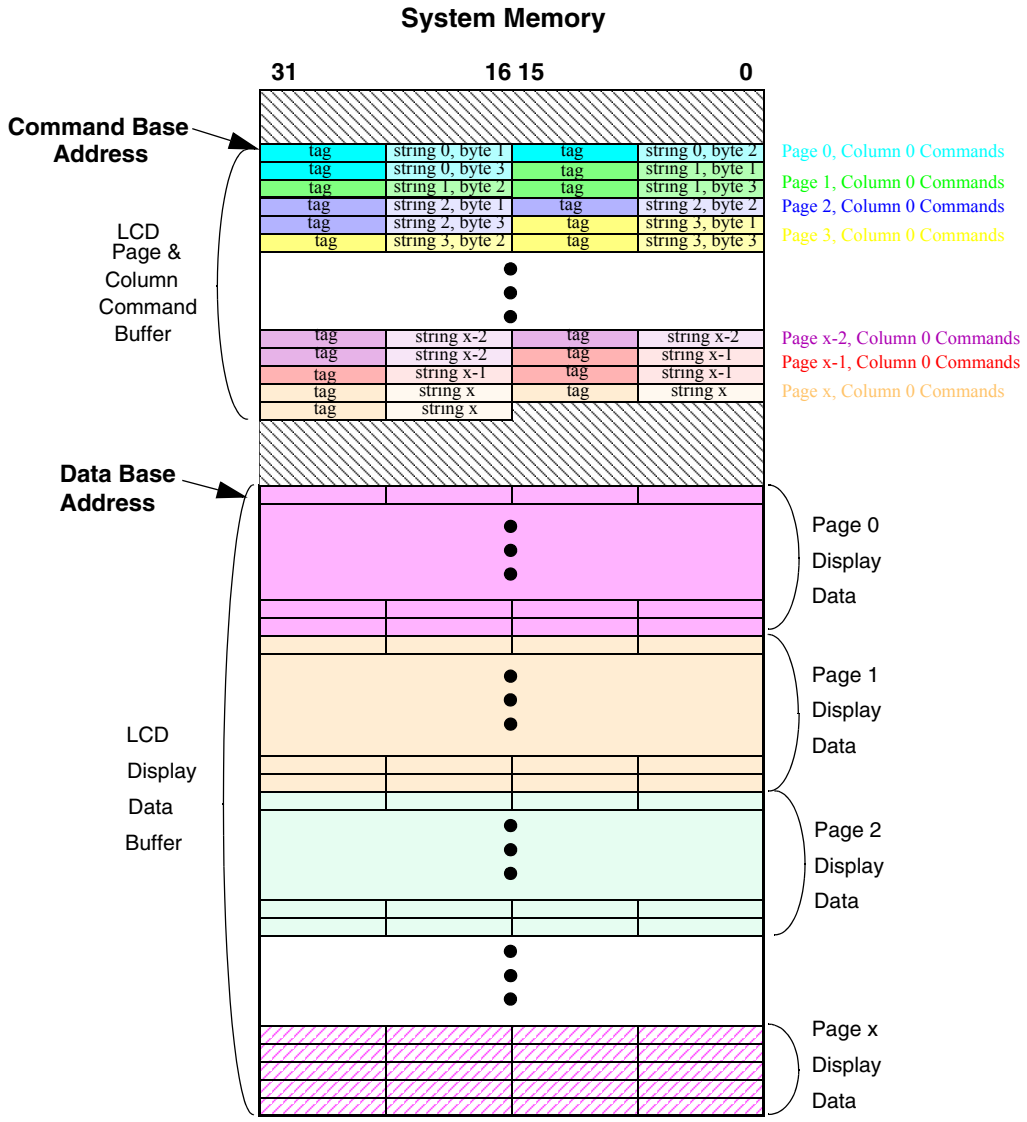


Figure 27-4. Automatic Display Data Transfer Memory Configuration (8-bit Command/8-bit Data)

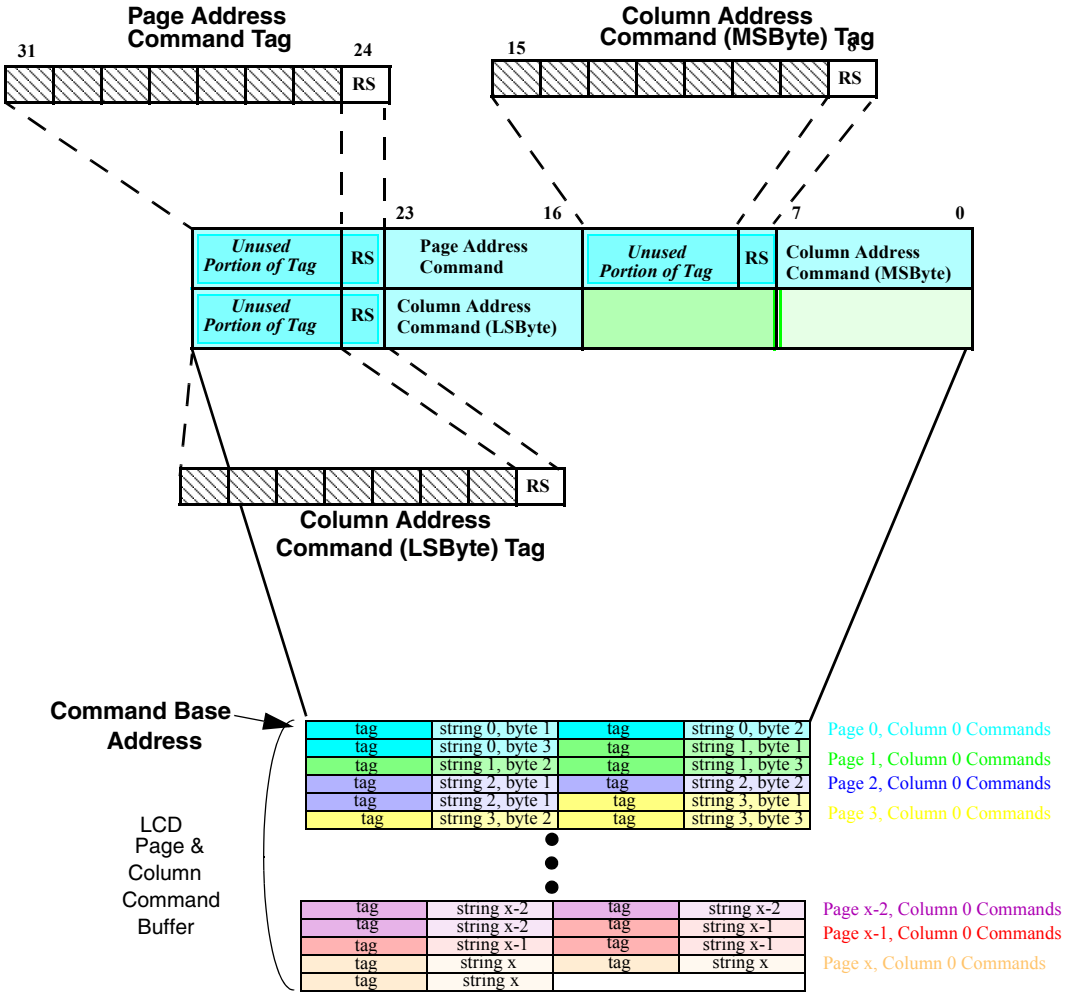


Figure 27-5. Command Buffer Tag Organization (8-bit Words)

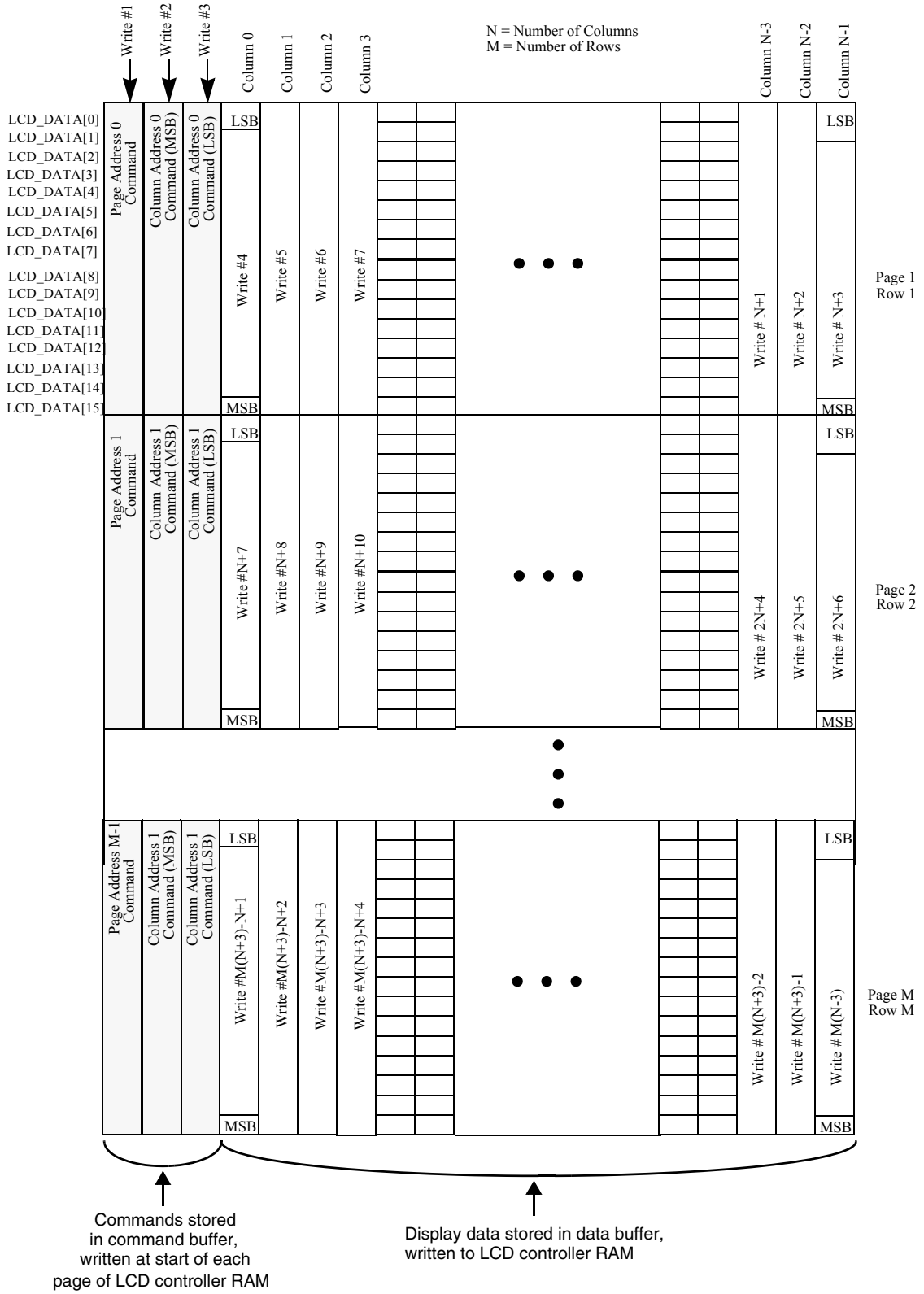


Figure 27-6. Sample LCD Controller Memory Map (16-bit Color) (16-bit Command/16-bit Data)

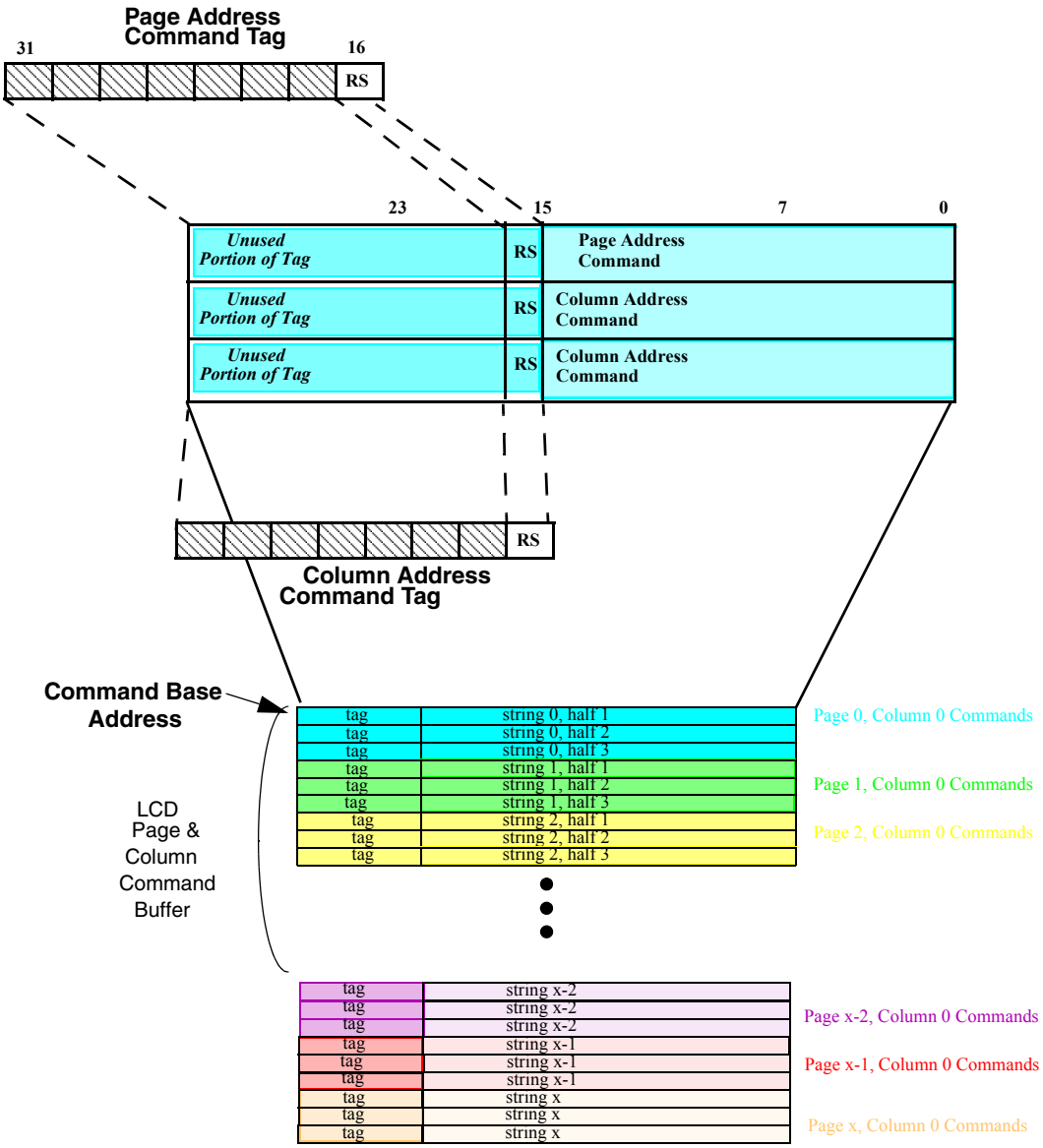


Figure 27-7. Command Buffer Tag Organization (16-bit Words)

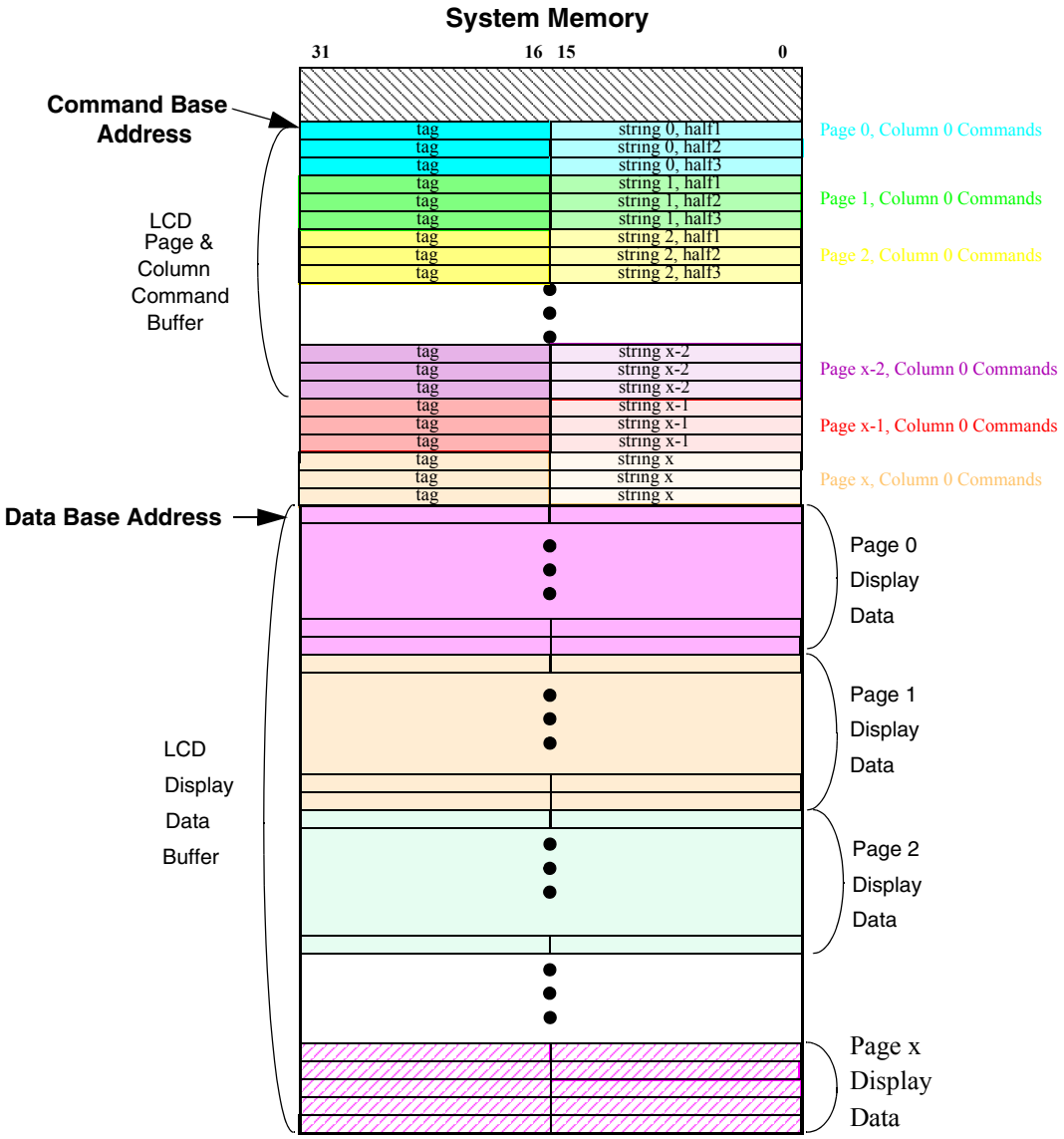


Figure 27-8. Automatic Display Data Transfer Memory Configuration (16-bit Command/16-bit Data)

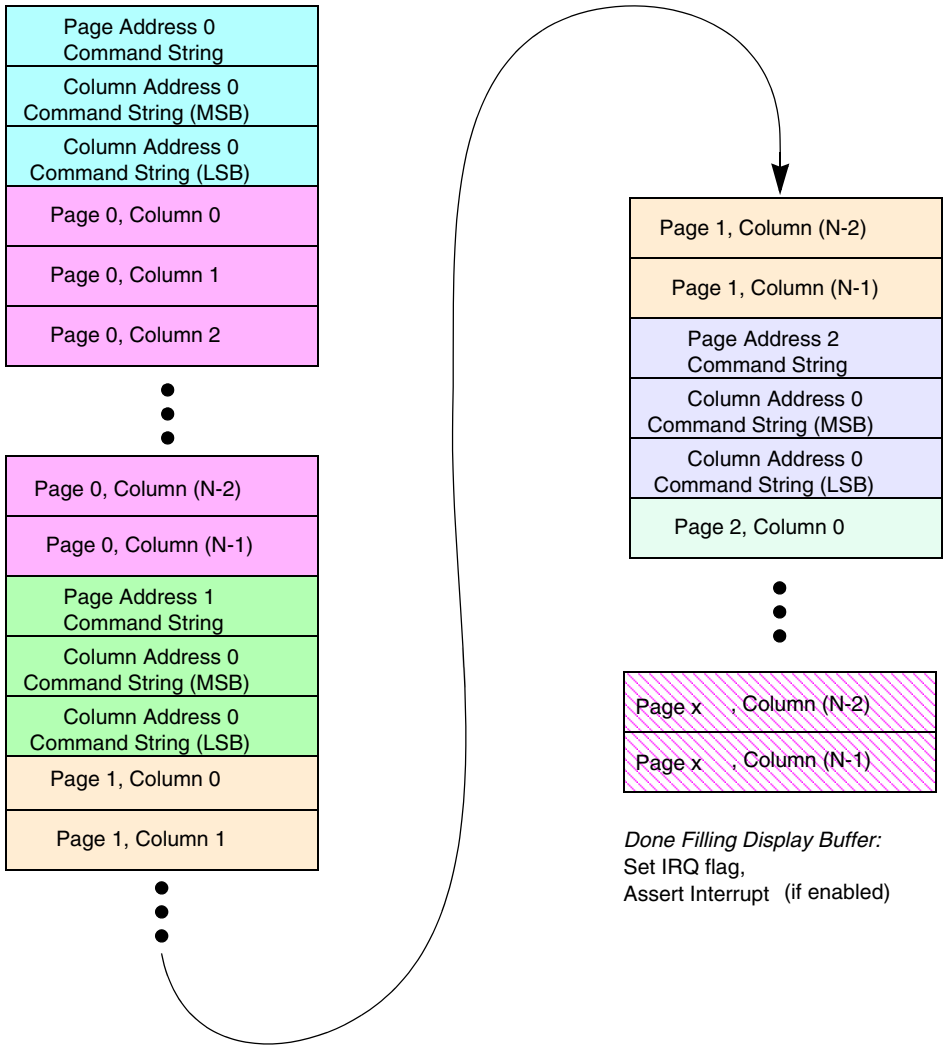


Figure 27-9. SLCDC Automatic Mode Write Sequence (Monochrome Display)

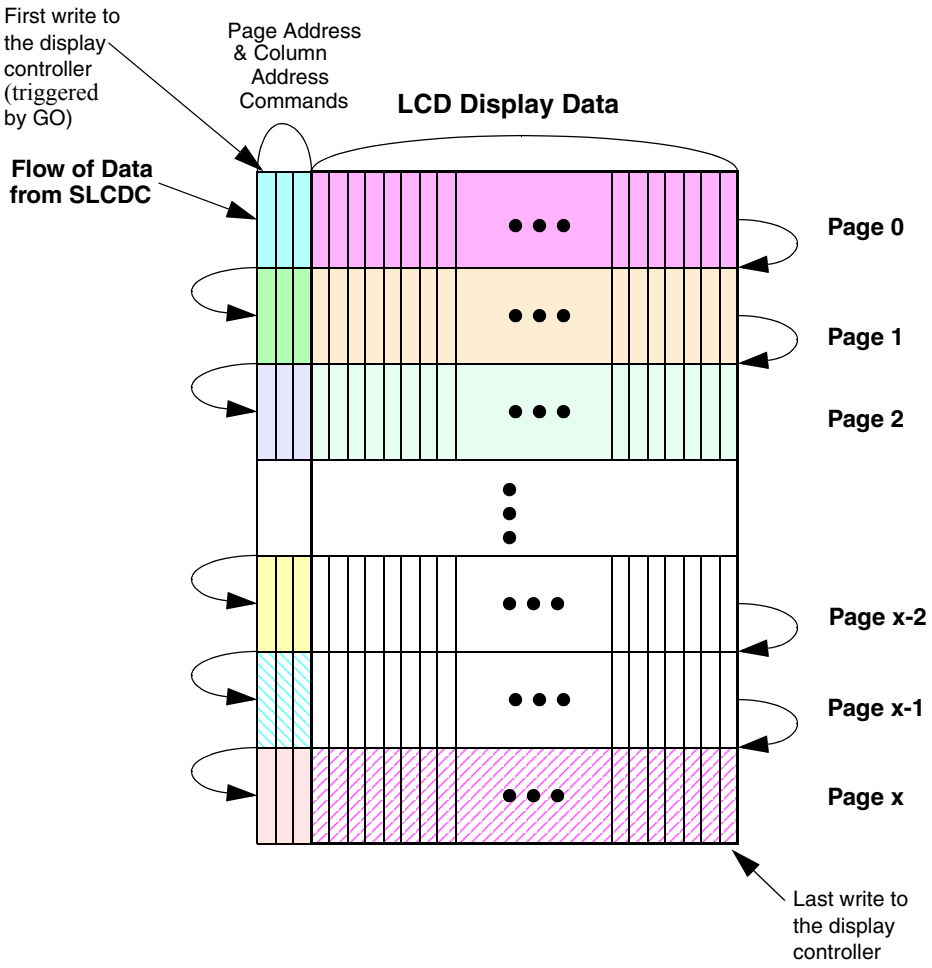


Figure 27-10. SLCDC Automatic Mode Data Flow (AUTOMODE[1:0] = 10)

27.2.3.1.2 Automatic Command Data Transfers (AUTOMODE[1:0]=00)

The SLCDC provides a convenient method of moving data from system memory to an external device, such as an LCD controller. When the AUTOMODE[1:0] bits = 00, the SLCDC ignores the command buffer and transfers the contents of the specified data buffer to the SLCDC output pins without inserting page and column addressing control strings. When AUTOMODE[1:0] = 00, the LCD_RS output is held to 0 during the entire transfer.

This method is used for sending a block of commands to the LCD controller for initialization. The buffer transmitted is defined by the DATABASEADR[16:0] bits of the DATA BASE ADDRESS register and the DATABUFSIZE[16:0] bits of the DATA BUFFER SIZE register. Setting the GO bit of the SLCDC control/status register starts the transfer and sets the BUSY status bit. After the number of words transferred to the external device is equal to the buffer size defined by the DATA BUFFER SIZE register, the BUSY bit will clear, the GO bit will clear, and the IRQ flag will set. A system interrupt is generated if the IRQEN bit is set.

27.2.3.1.3 Automatic Command Data Transfers (AUTOMODE[1:0]=01)

The SLCDC provides a convenient general purpose method of moving data from system memory to an external device, such as an LCD controller. When the AUTOMODE[1:0] bits = 01, the SLCDC ignores the command buffer and transfers the contents of the specified data buffer to the SLCDC output pins without inserting page and column addressing control strings. When AUTOMODE[1:0] = 01, the LCD_RS output is held to 1 during the entire transfer.

The buffer transmitted is defined by the DATABASEADR[31:0] bits of the DATA BASE ADDRESS register and the DATABUFSIZ[16:0] bits of the DATA BUFFER SIZE register. Setting the GO bit of the SLCDC control/status register starts the transfer and sets the BUSY status bit. After the number of words transferred to the external device is equal to the buffer size defined by the DATA BUFFER SIZE register, the BUSY bit clears, the GO bit clears, and the IRQ flag sets. A system interrupt is generated if the IRQEN bit is set.

27.2.3.2 Direct Register Access

Single words are written to the external LCD controller via the 9-bit LCD WRITE DATA register. Any write to the LCD WRITE DATA register results in a write to the external display controller, provided that the SLCDC is not currently in the middle of a transfer (indicated by BUSY=1 in the control/status register). The value written to the LCDDAT[15:0] bits of the LCD write data register are transmitted according to settings stored in the LCD transfer configuration register—that is, the transfer is done in accordance with the settings of the WORDDEFWRITE, XFRMODE, CSPOL, and SCKPOL bits of the LCD transfer configuration register.

Typical LCD controllers use a register select signal to distinguish between display data writes and control command writes. The RS bit of the LCD WRITE DATA register determines whether a write is presented to the external controller as a command string or a display data string. The LCD_RS pin is held to the value stored in the RS bit during the byte transfer. Because the transfer to the external device is slow compared to the system clock rates, the BUSY bit in the control/status register is used to indicate that the transfer is in progress. A transfer initiated by a write to the WRITE DATA register causes an interrupt if the IRQEN bit of the SLCDC control/status register is set.

WORDDEFWRITE in the LCD transfer configuration register determines if an 8- or 16-bit transfer occurs. An 8-bit transfer causes LCDDAT[7:0] bits to be transferred, and LCDDAT[15:8] to be ignored.

Another direct write cannot be done until the previous transfer has completed (BUSY=0).

27.2.4 Aborting SLCDC Transfers

The SLCDC control/status register contains a bit, ABORT, used to terminate a SLCDC transfer that is in progress. Termination of the transfer occurs gracefully. Any byte transfer that is in progress when ABORT is asserted is completed before the SLCDC goes to an idle state. The BUSY status bit clears upon entry to the idle state. The IRQ flag will also assert, indicating that the abort is complete. A SLCDC interrupt is generated, if enabled.

27.2.5 Low-Power Mode Operation

The SLCDC does not contain any control registers to program behavior of the SLCDC while in the low-power modes. Any power savings in the usage of SLCDC are gained by reducing the system clock to a lower rate.

27.3 Programming Model

The section provides detailed descriptions of the various SLCDC registers. Table 27-3 provides the detail summary for the SLCDC registers.

Table 27-3. SLCDC Register Summary

Name		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
DATA BASE ADDRESS (0x10022000)	R	DATABASEADR[31:2]															
	W																
	R	DATABASEADR[31:2]														0	0
	W																
DATA BUFFER SIZE (0x10022004)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	W																
	R	DATABUFSIZ[16:0]															
	W																
COMMAND BASE ADDRESS (0x10022008)	R	COMBASEADR[31:2]															
	W																
	R	COMBASEADR[31:2]														0	0
	W																
COMMAND BUFFER SIZE (0x1002200C)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	W																
	R	COMBUFSIZ[16:0]															
	W																
COMMAND STRING SIZE (0x10022010)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	0	0	0	0	0	0	0	0	COMSTRINGSIZ[7:0]							
	W																
FIFO CONFIG (0x10022014)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BURST[1:0]	
	W																
LCD CONFIG (0x10022018)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	0	0	0	WORDPPAGE[12:0]												
	W																

Table 27-3. SLCDC Register Summary (continued)

Name		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
LCD TRANSFER CONFIG (0x1002201C)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IMGEND	
	W																
	R	0	0	0	0	0	0	0	0	0	0	WORD DEF WRITE	WORD DEF DAT	WORD DEF COM	XFR MODE	CS POL	SCK POL
	W																
DMAC CONTROL/STATUS (0x10022020)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	0	0	0	AUTOMODE [1:0]		0	0	PROT1	IRQEN	IRQ	UNDRFLOW	IRQ	0	BUSY	0	0
	W										W1C	W1C	W1C			ABORT	GO
LCD CLOCK CONFIG (0x10022024)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W																
	R	0	0	0	0	0	0	0	0	0	0	DIVIDE[5:0]					
	W																
LCD WRITE DATA (0x10022028)	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RS
	W																
	R	LCDDAT[15:0]															
	W																

27.3.1 Data Buffer Base Address Register

DATA BASE ADDRESS — Data Buffer Base Address Register — 0x10022000

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATABASEADR															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATABASEADR															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 27-4. Data Base Address Register Description

Name	Description
DATABASEADR Bits 31–2	**Data Buffer Base Address**—This register contains the pointer in the R-AHB address space to the start of the LCD data buffer. This value is used to form the 32-bit data buffer base address. The data buffer base address value is forced to be word-aligned because the 2 LSB's are forced to 0. That is, data buffer base address [1:0] = 00.
Reserved Bits 1–0	Reserved—These bits are reserved and should read zero.

NOTE

The SLCDC DMA address generator width is 17 bits. Therefore, it is possible that a data buffer can be incorrectly addressed by the SLCDC if it extends beyond a 128 kbyte boundary. Therefore, it is recommended that the data buffer base address be set at the beginning of a 128 kbyte boundary. The sum of DATABASEADR[16:0] and DATABUFSIZ[16:0] must not exceed a multiple of 128K.

27.3.2 Data Buffer Size Register

DATA BUFFER SIZE — Data Buffer Size Register — Addr 0x10022004

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATABUFSIZE															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 27-5. Data Buffer Size Register Description

Name	Description
Reserved Bits 31–17	Reserved—These bits are reserved and should read zero.
DATABUFSIZ Bits 16–0	**Data Buffer Size**—This register defines the length (in words) of the LCD image or control buffer stored in system memory. This value is used to determine when the DMA transfer of the data buffer from system memory to the LCD controller is complete.

27.3.3 Command Buffer Base Address Register

COMMAND BASE ADDRESS — Command Buffer Base Address Register — Addr 0x10022008

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	COMBASEADR															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 27-6. Command Base Address Register Description

Name	Description
COMBASEADR Bits 31–2	**Command Buffer Base Address**—This register contains the pointer in the R-AHB address space to the start of the LCD command buffer that contains the LCD controller page and column address commands. This buffer is used when the SLCDC is configured for transfers with automatic page address and column address command insertion. The command buffer base address value is forced to be word-aligned because the 2 LSB's are forced to 0. That is, command buffer base address [1:0] = 00.
Reserved Bits 1–0	Reserved—These bits are reserved and read zero.

NOTE

The SLCDC DMA address generator width is 17 bits. Therefore, it is possible that a command buffer can be incorrectly addressed by the SLCDC if it extends beyond a 128 kbyte boundary. Therefore, it is recommended that the command buffer base address be set at the beginning of a 128 kbyte boundary. The sum of COMBASEADR[16:0] and COMBUFSIZ[16:0] must not exceed a multiple of 128K.

27.3.4 Command Buffer Size Register

COMMAND BUFFER SIZE — Command Buffer Size Register — Addr 0x1002200C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COMBUFSIZE															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 27-7. Command Buffer Size Register Description

Name	Description
Reserved Bits 31–17	Reserved—These bits are reserved and read zero.
COMBUFSIZ Bits 16–0	**Command Buffer Size**—This register defines the length (in words) of the LCD control buffer stored in system memory. This value is used to determine when the DMA transfer of the command buffer from system memory to the LCD controller is complete.

27.3.5 Command String Size Register

COMMAND STRING SIZE — Command String Size Register — Addr 0x10022010

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									COMSTRINGSIZ							
TYPE	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 27-8. Command String Size Register Description

Name	Description
Reserved Bits 31–8	Reserved—These bits are reserved and should read zero.
COMSTRINGSIZ Bits 7–0	**Command String Size**—This register contains COMSTRINGSIZ[7:0] bits, which define the length (in words) of the LCD command string made up of the LCD controller page and column address commands. This command string is used when the SLCDC begins to fill a new page of the LCD controller's RAM. It should be noted that these strings are only transmitted to the LCD controller when the SLCDC is configured for transfers with automatic page and column address command insertion. The command strings are stored along with tags in the LCD command buffer located in system memory. The COMSTRINGSIZ[7:0] bits represent the number of words that must be sent to the LCD controller to set the page and column address, not the number of words needed to store the corresponding commands and tags in system memory.

27.3.6 FIFO Configuration Register

FIFOCONFIG — FIFO Configuration Register — Addr 0x10022014

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														BURST		
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 27-9. FIFO Configuration Register Description

Name	Description	Settings
Reserved Bits 31–3	Reserved—These bits are reserved and should read zero.	
BURST Bits 2–0	**DMA Burst Length**—These bits select the length (in 32-bit words) of the DMA burst. That is, these bits determine the number of 32-bit word reads that occur when the SLCDC has ownership of the R-AHB bus.	000 = Burst length set to 1 32-bit word 001 = Burst length set to 2 32-bit words 010 = Burst length set to 3 32-bit words 011 = Burst length set to 4 32-bit words 100 = Burst length set to 5 32-bit words 101 = Burst length set to 6 32-bit words 110 = Burst length set to 7 32-bit words 111 = Burst length set to 8 32-bit words

27.3.7 LCD Controller Configuration Register

LCDCONFIG — LCD Controller Configuration Register — Addr 0x10022018

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				WORDPPAGE												
TYPE	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 27-10. LCD Controller Configuration Register Description

Name	Description	Settings
Reserved Bits 31–13	Reserved—These bits are reserved and should read zero.	
WORDPPAGE Bits 12–0	**LCD Bytes Per Page**—These 13 bits set the number of words per page for the external LCD controller connected to the SLCDC module. These bits are used by the SLCDC to determine when control characters must be sent to the controller.	0 = 0 words per LCD page 1 = 1 word per LCD page 2 = 2 words per LCD page … 8191 = 8191 words per LCD page

27.3.8 LCD Transfer Configuration Register

LCDTRANS CONFIG — LCD Transfer Configuration Register — Addr 0x1002201C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															IMGEND[1:0]	
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											WORDDEF WRITE	WORDDEF DAT	WORDDEF COM	XFR MODE	CSPOL	SKCPOL
TYPE	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 27-11. LCD Transfer Configuration Register Description

Name	Description	Settings
Reserved Bits 31–18	Reserved—These bits are reserved and should read zero.	
IMGEND Bits 17–16	**Image Endianess**—This field defines the endianess of the image.	00 = Big endian or 32-bit little endian 01 = 16-bit little endian 10 = 8-bit little endian 11 = Reserved
Reserved Bits 15–6	Reserved—These bits are reserved and should read zero.	
WORDDEFWRITE Bit 5	**Word Define, Write Data**—This bit defines word for the LCD WRITE DATA register transfers.	0 = 8-bit data words 1 = 16-bit data words
WORDDEFDAT Bit 4	**Word Define, Data**—This bit defines word for LCD data registers and data transfers. All registers associated with data transfers (DATA BUFFER SIZE, LCD CONFIG) must take this into consideration.	0 = 8-bit data words 1 = 16-bit data words
WORDDEFCOM Bit 3	**Word Define, Command**—This bit defines word for LCD command registers and command data transfers. All registers associated with data transfers (COMMAND BUFFER SIZE) must take this into consideration.	0 = 8-bit command words 1 = 16-bit command words
XFRMODE Bit 2	**Image Data Transfer Width**—This bit selects the width of the LCD display interface databus. Data transfers to the LCD controller can be done serially or in parallel.	0 = word serial transfers 1 = word parallel transfers

Table 27-11. LCD Transfer Configuration Register Description (continued)

Name	Description	Settings
CSPOL Bit 1	**Chip Select Polarity**—This bit selects the polarity of the LCD_CS signal. **Note:** This bit affects the behavior of the LCD_CS pin in both serial and parallel mode.	0 = LCD_CS is asserted low. Therefore, the SLCDC will drive LCD_CS to 0 during a serial transfer of data to the external LCD controller. In parallel mode, LCD_CS will normally be high. LCD data will change when LCD_CS goes low. The rising edge of LCD_CS is used by the LCD controller to latch the parallel data. 1 = LCD_CS is asserted high. Therefore, the SLCDC will drive LCD_CS to 1 during a serial transfer of data to the external LCD controller. In parallel mode, LCD_CS will normally be low. LCD data will change when LCD_CS goes high. The falling edge of LCD_CS is used by the LCD controller to latch the parallel data.
SCKPOL Bit 0	**Serial Data Clock Polarity**—This bit selects whether the serial data transitions on the rising or falling edge of the serial data clock during transfers of data in serial mode. This bit has no effect during parallel data transfers to the external LCD display device.	0 = Serial data will transition on the rising edge of the serial clock. Therefore, data should be latched by the display device on the falling edge of the serial clock. 1 = Serial data will transition on the falling edge of the serial clock. Therefore, data should be latched by the display device on the rising edge of the serial clock.

27.3.9 SLCDC Control/Status Register

SLCDC CONTROL/STATUS — SLCDC Control/Status Register — Addr 0x10022020

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				AUTOMODE				PROT1	IRQEN	IRQ	UNDRFLOW	TEA		BUSY	ABORT	GO
TYPE	r	r	r	rw	rw	r	r	rw	rw	w1c	w1c	w1c	r	r	slfclr	slfclr
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 27-12. SLCDC Control/Status Register Description

Name	Description	Settings
Reserved Bits 31–13	Reserved—These bits are reserved and should read zero.	
AUTOMODE Bits 12–11	**Automatic Transfer Mode**—These bits control how the SLCDC transfers the data buffer to the LCD controller upon being triggered by the GO bit of the SLCDC control/status register. The DATA BASE ADDRESS bits designate the location of the transferred data buffer in system memory. Transfers triggered by the GO bit can be configured to automatically insert page address and column address commands in the data stream to navigate through the LCD controller's RAM. This is done without CPU intervention. The SLCDC can be also be configured to send the data buffer to the LCD controller without inserting any command strings. In this mode, the value of the LCD_RS signal can be configured to be either 1 or 0 during the transfer.	00 = SLCDC transfers the data buffer defined by DATA BASE ADDRESS and DATA BUFFER SIZE without inserting control strings. With this setting, the LCD_RS signal is tied to 0 during the entire transfer. 01 = SLCDC transfers the data buffer defined by DATA BASE ADDRESS and DATA BUFFER SIZE without inserting control strings. With this setting, the LCD_RS signal is tied to 1 during the entire transfer. 10 = When transmitting the data buffer defined by DATA BASE ADDRESS and DATA BUFFER SIZE, the SLCDC inserts command strings to set the LCD controller's page and column addresses. These command strings are taken from the buffer defined by COMMAND BASE ADDRESS and COMMAND STRING SIZE. This insertion of command strings is done at the beginning of each page of LCD controller RAM. 11 = Reserved
Reserved Bits 10–9	Reserved—These bits are reserved and should read zero.	
PROT1 Bit 8	**R-AHB Protection Code**—This bit controls the value of the HPROT[1] signal sent to the R-AHB each time a SLCDC access is done. This value controls the type of access being done on the bus (user or privileged). The SLCDC_hprot[0] output of the SLCDC is tied to 1 to indicate a data access. The PROT1 bit resets to 0.	0 = SLCDC_hprot[1:0] driven to 01 indicating SLCDC accesses will be user data accesses. 1 = SLCDC_hprot[1:0] driven to 11 indicating SLCDC accesses will be privileged data accesses.
IRQEN Bit 7	**Interrupt Enable**—This bit controls whether a SLCDC interrupt is generated upon completion of a SLCDC transfer. SLCDC transfers can end three ways: 1) When a transfer error occurs on the R-AHB. 2) When the SLCDC completes the transfer of data to the LCD controller. 3) When a SLCDC transfer is aborted by setting the ABORT bit in the SLCDC control/status register. SLCDC transfers are initiated by assertion of the GO bit or a write to the LCD WRITE DATA register. Clearing the IRQEN bit prevents an interrupt from being generated when the SLCDC completes a transfer.	0 = SLCDC interrupt is masked. No interrupt is generated from the SLCDC. 1 = SLCDC interrupt is generated upon the completion of a SLCDC transfer.

Table 27-12. SLCDC Control/Status Register Description (continued)

Name	Description	Settings
IRQ Bit 6	**SLCDC Interrupt Flag**—This bit indicates that an interrupt condition occurred in the SLCDC. The interrupt flag is set when a SLCDC transfer is completed or aborted. SLCDC transfers are initiated by assertion of the GO bit or a write to the LCD WRITE DATA register. This bit is cleared by writing a 1 to the IRQ bit of the control/status register.	0 = SLCDC interrupt is not pending 1 = Interrupt condition occurred in the SLCDC.
UNDRFLOW Bit 5	**SLCDC FIFO Underflow**—This bit indicates that the SLCDC FIFO became empty during the transfer of data from system memory to the LCD controller. Although FIFO underflow does not cause an error in the SLCDC transfer, it does indicate the output portion of the SLCDC had to wait for the FIFO to be filled during the transfer. This bit is cleared by writing a 1 to the UNDRFLOW bit of the control/status register. **Note:** The underflow flag will not set until after some data has been read into the SLCDC FIFOs. Underflow only sets after a condition where the FIFOs have data, they empty, and then do not refill fast enough for the SLCDC bit manipulator. In this case, the SLCDC bit manipulator stalls while it waits for data to be read from system memory into the FIFOs. The underflow flag will not set in a case where a SLCDC transfer is initiated, but the SLCDC is never granted control of the bus, and thus able to fill the FIFOs. The FIFOs must go from a non-empty state to an empty state for UNDERFLOW to set.	0 = No underflow occurred in SLCDC FIFO. 1 = SLCDC FIFO became empty during the transfer of data to the LCD controller. This caused the output portion of the SLCDC to wait, delaying transfer of data to the LCD controller.
TEA Bit 4	**SLCDC DMA Transfer Error**—This bit indicates that the SLCDC DMA received a transfer error acknowledge from the R-AHB bus while trying to read data from system memory. This is indicated by HRESP0=1 and HREADY=1 on the R-AHB bus. This is a fatal condition and causes the SLCDC to terminate its transfer. This bit is cleared by writing a 1 to the TEA bit of the SLCDC control/status register.	0 = No transfer error acknowledge error occurred. 1 = A transfer error acknowledge was received by the SLCDC DMA from the R-AHB.
Reserved Bit 3	Reserved—This bit is reserved and should read zero.	
BUSY Bit 2	**SLCDC Busy**—This bit indicates that the SLCDC is in the process of transferring the image buffer from system memory to the LCD controller. Data transfer is initiated by setting the GO bit of the SLCDC control/status register or by writing data to the LCD write data register. This bit is a read-only bit and clears after the transfer of data to the LCD controller is complete.	0 = SLCDC idle. 1 = SLCDC in process of transferring image buffer.

Table 27-12. SLCDC Control/Status Register Description (continued)

Name	Description	Settings
ABORT Bit 1	**Abort SLCDC Transfer**—This bit causes the SLCDC to abort the data transfer currently underway. Termination of the transfer occurs gracefully. That is, any byte transfer to the LCD controller in progress is completed before the SLCDC goes to an idle state. This bit is cleared automatically when the SLCDC has transitioned to the idle state.	0 = Do not abort current data transfer. 1 = Abort the SLCDC data transfer currently in progress.
GO Bit 0	**Start SLCDC Transfer**—This bits starts the automatic transfer of image data from system memory to the external LCD controller. This bit always read 1 until the data transfer is complete. After the data transfer is complete or has been aborted, the GO bit will read 0. **Note:** Writing 1 to the GO bit has no effect if the BUSY bit of the SLCDC control/status register is set.	0 = Do not start SLCDC transfer. 1 = Start SLCDC image data transfer.

27.3.10 LCD Clock Configuration Register

LCD CLOCK CONFIG — LCD Clock Configuration Register — Addr 0x10022024

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											DIVIDE					
TYPE	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 27-13. LCD Clock Configuration Register Description

Name	Description	Settings
Reserved Bits 31–6	Reserved—These bits are reserved and should read zero.	
DIVIDE Bits 5–0	**LCD Clock Divide Value**—This bit sets the divide ratio used to generate the LCD clock from the HCLK_SLCDC clock (See Section 27.5, “LCD Clock Configuration,” for a detailed explanation of the fractional divider programming). Setting DIVIDE to 0 disables the LCD_CLK by gating HCLK_SLCDC clock from the fractional divider. These bits must be set to some non-zero value before transfers to the LCD can occur.	Frequency relations: 0 = LCD_CLK off 1 = LCD_CLK = HCLK_SLCDC clock/128 2 = LCD_CLK=HCLK_SLCDC clock*2/128 … 62 = LCD_CLK=HCLK_SLCDC clock*62/128 63 = LCD_CLK=HCLK_SLCDC clock*63/128

27.3.11 LCD Write Data Register

LCD WRITE DATA — LCD Write Data Register — Addr 0x10022028

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																RS
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LCDDAT															
TYPE	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 27-14. LCD Write Data Register Description

Name	Description	Settings
Reserved Bits 31–17	Reserved—These bits are reserved and should read zero.	
RS Bit 16	**LCD Register Select**—This bit determines the value placed on the LCD_RS pin during a byte transfer from the WRITE DATA register.	0 = LCD_RS signal is tied to 0 during the transfer. 1 = LCD_RS signal is set to 1 during the transfer.
LCDDAT Bits 15–0	**LCD Controller Data**—Data written is transferred to the external LCD controller. This register gives direct write access to the LCD controller. The data is determined to be command or display data via the RS bit of the LCD Write Data register. Data written to this register is only transferred to the LCD controller if the SLCDC is not in middle of another transfer (indicated by the BUSY bit of the control/status register). A read of these bits gives the last value written to this register (or the reset value), not a value from the LCD controller. The amount of data that is transferred is determined by WORDDEFWRITE bit in the LCD transfer configuration register. During 8-bit word transfers, LCDDAT[7:0] is transferred, and LCDDAT[15:8] is ignored.	

27.4 LCD Controller Interface

The SLCDC transfers data from the display memory buffer in system memory to the external display device. Transfers can be done via a 4-wire serial, 3-wire serial, 8-bit parallel, or a 16-bit parallel interface.

27.4.1 Serial Interface

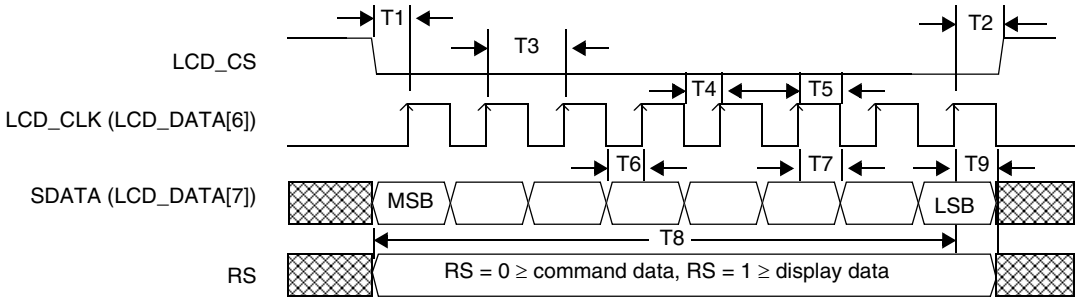
In serial mode (XFRMODE bit in LCD transfer configuration register set to 0), data is transmitted to the display device via four signals: LCD_CS, LCD_DATA[7:6], and LCD_RS. The data is transmitted on LCD_DATA[7], the serial clock toggles on LCD_DATA[6], and LCD_RS is used to tell the display whether the data being transmitted is display data or command data. LCD_CS is used as a chip select signal. The LCD_CS signal is asserted during all transfers, and is held asserted for sequential bytes being transmitted. The polarity of the LCD_CS signal is programmable using the CSPOL bit of the LCD transfer configuration register.

Figure 27-11 shows the timing associated with the transfer of one byte of data to the display. The polarity of the serial data clock on the LCD_DATA[6] pin can be configured using the SCKPOL bit of the LCD transfer configuration register. This bit must be set in accordance with the external display device requirements. If the external device latches the serial data on the rising edge of the serial clock, SCKPOL must be set to 1. If the external device latches the serial data on the falling edge of the serial clock, SCKPOL must be cleared. The command and data word definitions determine how long the data string is in serial mode. Figure 27-11 shows the 8-bit transfers. A serial transfer of 16 bits has the same timing as 8-bit transfers, however 16 bits of data is transferred. It is possible to have a command defined as 8 bits and serial defined as 16 bits (or vice versa). In this case, when a command is being transferred, 8 bits of data are sent, and when data is being transferred, 16 bits of data is sent.

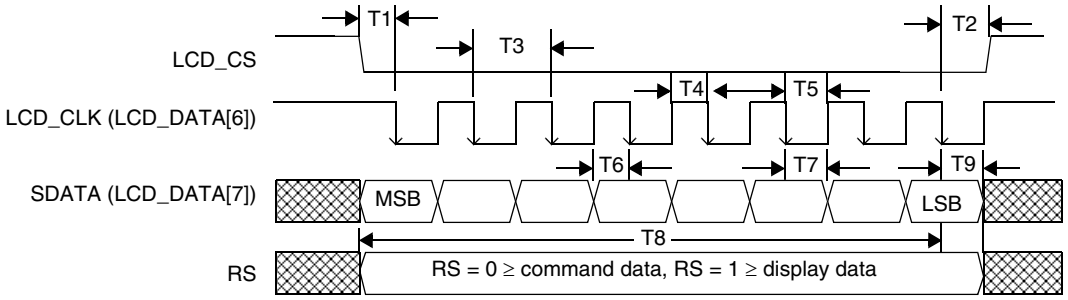
Table 27-15. SLCDC Serial Interface Timing

Symbol	Parameter	Minimum (ns)	Typical (ns)	Maximum (ns)
T1	Chip select setup time	(t_{cyc} / 2) (±) t_{prop}	–	–
T2	Chip select hold time	(t_{cyc} / 2) (±) t_{prop}	–	–
T3	Serial clock cycle time	39 (±) t_{prop}	–	2641
T4	Serial clock low pulse	18 (±) t_{prop}	–	–
T5	Serial clock high pulse	18 (±) t_{prop}	–	–
T6	Data setup time	(t_{cyc} / 2) (±) t_{prop}	–	–
T7	Data hold time	(t_{cyc} / 2) (±) t_{prop}	–	–
T8	Register select setup time	(15 * t_{cyc} / 2) (±) t_{prop}	–	–
T9	Register select hold time	(t_{cyc} / 2) (±) t_{prop}	–	–

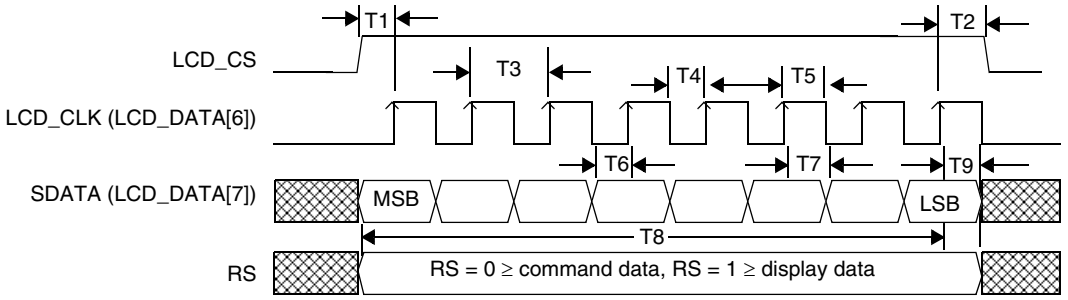
t_{prop} is the propagation delay from the SLCDC outputs to the pin outputs.



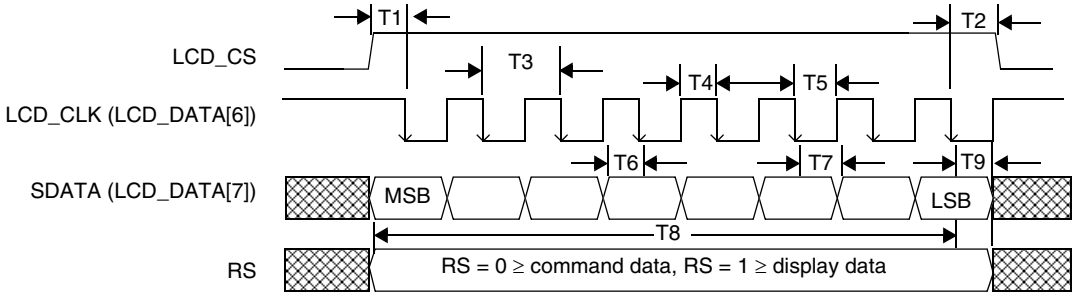
This diagram illustrates the timing when the SCKPOL = 1, CSPOL = 0



This diagram illustrates the timing when the SCKPOL = 0, CSPOL = 0



This diagram illustrates the timing when the SCKPOL = 1, CSPOL = 1



This diagram illustrates the timing when the SCKPOL = 0, CSPOL = 1

Figure 27-11. SLCDC Serial Transfers to LCD Device

27.4.2 Parallel Interface

The SLCDC module can be configured to execute parallel transfers of display data from system memory to an external display device. Figure 27-12 shows the timing associated with an SLCDC parallel transfer to an external LCD device. The timing is based on the frequency of LCD_CLK, which is programmable via the LCD clock configuration register. In parallel mode, the LCD_CS pin is used as a write strobe by the external LCD controller. The latching edge of LCD_CS occurs one LCD_CLK cycle after the data is made available to the display on the LCD_DATA[15:0] pins. The polarity of the LCD_CS signal is programmable using the CSPOL bit in the LCD transfer configuration register. Data transfers of 16- or 8-bit are determined by the word definition bits in the LCD transfer configuration register. All 8-bit data transfers use LCD_DATA[7:0]. LCD_DATA[15:8] shows 0's. It is possible for there to be 8-bit command data transfers and 16-bit display data transfers (and vice versa). In this case, when command data is transferred, only LCD_DATA[7:0] is used, and display data uses LCD_DATA[15:0].

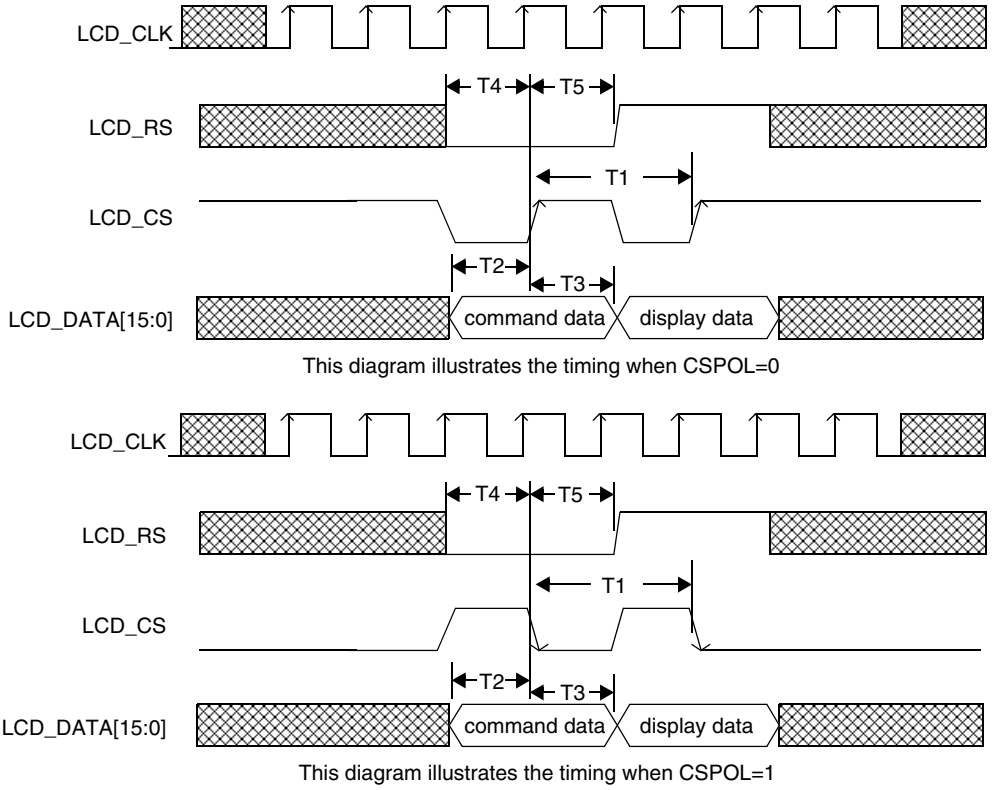


Figure 27-12. SLCDC Parallel Transfers to LCD Device

Table 27-16. SLCDC Parallel Interface Timing

Symbol	Parameter	Minimum (ns)	Typical (ns)	Maximum (ns)
T1	Parallel clock cycle time	78 (±) t_{prop}	–	4923
T2	Data setup time	(t_{cyc} / 2) (±) t_{prop}	–	–
T3	Data hold time	(t_{cyc} / 2) (±) t_{prop}	–	–
T4	Register select setup time	(t_{cyc} / 2) (±) t_{prop}	–	–
T5	Register select hold time	(t_{cyc} / 2) (±) t_{prop}	–	–

t_{prop} is the propagation delay from the SLCDC outputs to the pin outputs.

27.5 LCD Clock Configuration

The SLCDC uses a fractional divider to generate the LCD data clock from the HCLK_SLCDC clock signal. The fractional divider is programmed by setting a divide value using the DIVIDE[5:0] bits of the LCD clock configuration register. The frequency of LCD_CLK is calculated according to the Equation :

Eqn. 27-1

$$LCD\_CLK = \frac{HCLKx(DivideValue)}{128}$$

The divide value is taken directly from the value stored in the DIVIDE[5:0] bits of the LCD clock configuration register. Setting DIVIDE[5:0] to 0 disables the LCD_CLK signal. These bits must be set to some non-zero value before transfers to the LCD can occur.

Table 27-17. LCD_CLK Frequency Range

HCLK_SLCDC Clock Frequency (MHz)	Minimum LCD_CLK Frequency (MHz)	Maximum LCD_CLK Frequency (MHz)	Resolution of Step (kHz)
52	0	25.59	406.25
26	0	12.8	203.13
16.8	0	8.27	131.25

27.6 R-AHB Interface and SLCDC FIFOs

The SLCDC uses a DMA interface to access system memory without CPU intervention. When the SLCDC needs data, the DMA requests control of the AHB and reads 32-bit words from system memory. The number of 32-bit words read during the burst is determined by the value stored in the BURST[2:0] bits of the FIFO configuration register. Control of the AHB is relinquished by the DMA after the burst is complete. The SLCDC generates idle cycles when the data transfer is complete. The amount of R-AHB bandwidth required by the SLCDC can be controlled by system software. The frequency of LCD frame updates and the rate of the LCD_CLK affect how often the SLCDC requests use of the R-AHB.

The data read by the DMA interface is stored in two 32-bit × 8 word FIFOs that are used to buffer data between the DMA and the SLCDC display interface. One FIFO is used to buffer data from the command buffer and the other buffers data from the display data buffer. When one of the FIFOs has enough room to store the data read during a DMA burst, it will request servicing. Because the DMA burst length is programmable, the occupancy level of the FIFOs can vary when servicing is requested.

Chapter 28
enhanced Multimedia Accelerator (eMMA)

The enhanced MultiMedia Accelerator (eMMA) consists of the video Pre-processor (PrP), Encoder (ENC), Decoder (DEC), and Post-processor (PP). These blocks work together to provide video acceleration and off-load the CPU from computation intensive tasks.

While the Encoder and Decoder support only MPEG4-SVIP, the PrP and PP can be used for generic video pre and post processing such as scaling, resizing, and color space conversions.

eMMa includes the following features:

- Pre-processor:
  - Data input:
    - System memory
    - Private DMA between CMOS Sensor Interface module and pre-processor
  - Data input formats:
    - Arbitrarily formatted RGB pixels (16 or 32 bits)
    - YUV 4:2:2 (Pixel interleaved)
    - YUV 4:2:0 (IYUV, YV12)
  - Input image size: 32 × 32 to 2044 × 2044
  - Image scaling:
    - Programmable independent CH-1 and CH-2 resizer. Can program to be in cascade or parallel.
    - Each resizer supports downscaling ratios from 1:1 to 8:1 in fractional steps.
  - Channel-1 output data format
    - Channel 1
    - RGB 16 and 32 bpp
    - YUV 4:2:2 (YUYV, YVYU, UYVY, VYUY)
  - Channel-2 output data format
    - YUV 4:2:2 (YUYV)
    - YUV 4:4:4
    - YUV 4:2:0 (IYUV, YV12)
    - RGB data and YUV data format can be generated concurrently
- Encoder
  - Supports MPEG4 and H.263 (Short Video Header)
  - Full conformance to ISO/IEC 14496-2 Visual Simple Profiles Levels 0 to 3

  - Supports real-time encoding images of sizes from 32 × 32 up to CIF at 30 fps
  - Input data format is YUV 4:2:0 (IYUV) from system memory
  - Supports camera stabilization
- Decoder
  - Supports MPEG4 and H.263 (Short Video Header)
  - Full conformance to ISO/IEC 14496-2 Visual Simple Profile Levels 0 to 3
  - Supports real-time decoding of images of sizes from 32 × 32 up to CIF at 30 fps
  - Output data format is YUV 4:2:0 (IYUV) to system memory
- Post-processor
  - Input data:
    - From system memory
  - Input format:
    - YUV 4:2:0 (IYUV, YV12)
  - Image Size: 32 × 32 to 2044 × 2044
  - Output format:
    - YUV 4:2:2 (YUYV)
    - RGB16 and RGB32 bpp
  - Image Resize
    - Upscaling ratios ranging from 1:1 to 1:4 in fractional steps
    - Downscaling ratios ranging from 1:1 to 2:1 in fractional steps and a fixed 4:1
    - Ratios provide scaling between QCIF, CIF, QVGA (320 × 240, 240 × 320)

28.1 eMMA Architecture

Figure 28-1 shows the block diagram of eMMA.

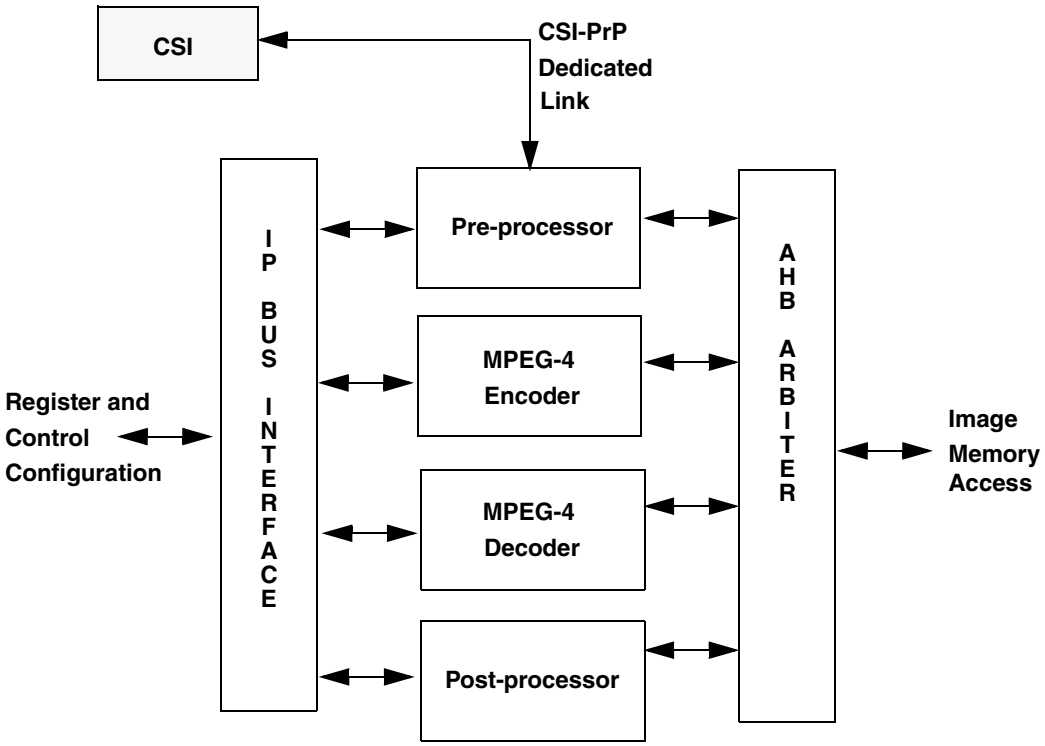


Figure 28-1. eMMA Block Diagram

The eMMA consists of the Pre-processor, MPEG-4 Encoder, MPEG-4 Decoder and Post-processor modules. Each module has individual control and configuration registers which are accessed via the IP interface and are capable of bus mastering the AMBA bus to independently access system memory without any CPU intervention. This allows each module to be used independently of each other and enables the Pre-processor and Post-processor modules to provide acceleration features for other software codec implementations and image processing software.

28.1.1 Pre-Processor (PrP)

The Pre-processor (PrP) module accepts input from main memory or from the dedicated link that connects it to camera input via the CMOS Sensor Interface (CSI) module. The PrP can be operated in two modes; single or continuous frame (loop) mode. In single frame mode, a single frame is processed either from memory or from the dedicated CSI-PrP link. In this mode, the PrP must be re-enabled each time a frame is to be processed. This mode is suitable for still image capture, processing and display and for very low frame rate operation. In continuous frame or loop mode, the PrP processes input frames from the dedicated CSI-PrP link continuously until it is disabled or an error occurs.

The PrP has two output channels (Channel-1 and Channel-2) and both channels store processed frames to main memory. The output from Channel-1 is dedicated for display purposes and the output from Channel-2 for input to a hardware encoder (MPEG-4 Encoder module) or a software encoder or image compressor. The PrP resizes input frames from memory of from the CSI and performs color space conversion.

28.1.2 Encoder and Decoder

The Encoder and Decoder, together with additional control software, implement a MPEG-4 Simple Visual Profile Codec (Levels 0-3).

28.1.3 Post-Processor (PP)

The Post-processor module takes decoded frames from memory and performs additional processing to de-block, de-ring, resize, and color space convert the decoded frames for display. The decoded input can be either from the Decoder module or a software decoder module.

The following sections provide more detailed information on the PP and PrP blocks.

28.2 Post-Processor (PP)

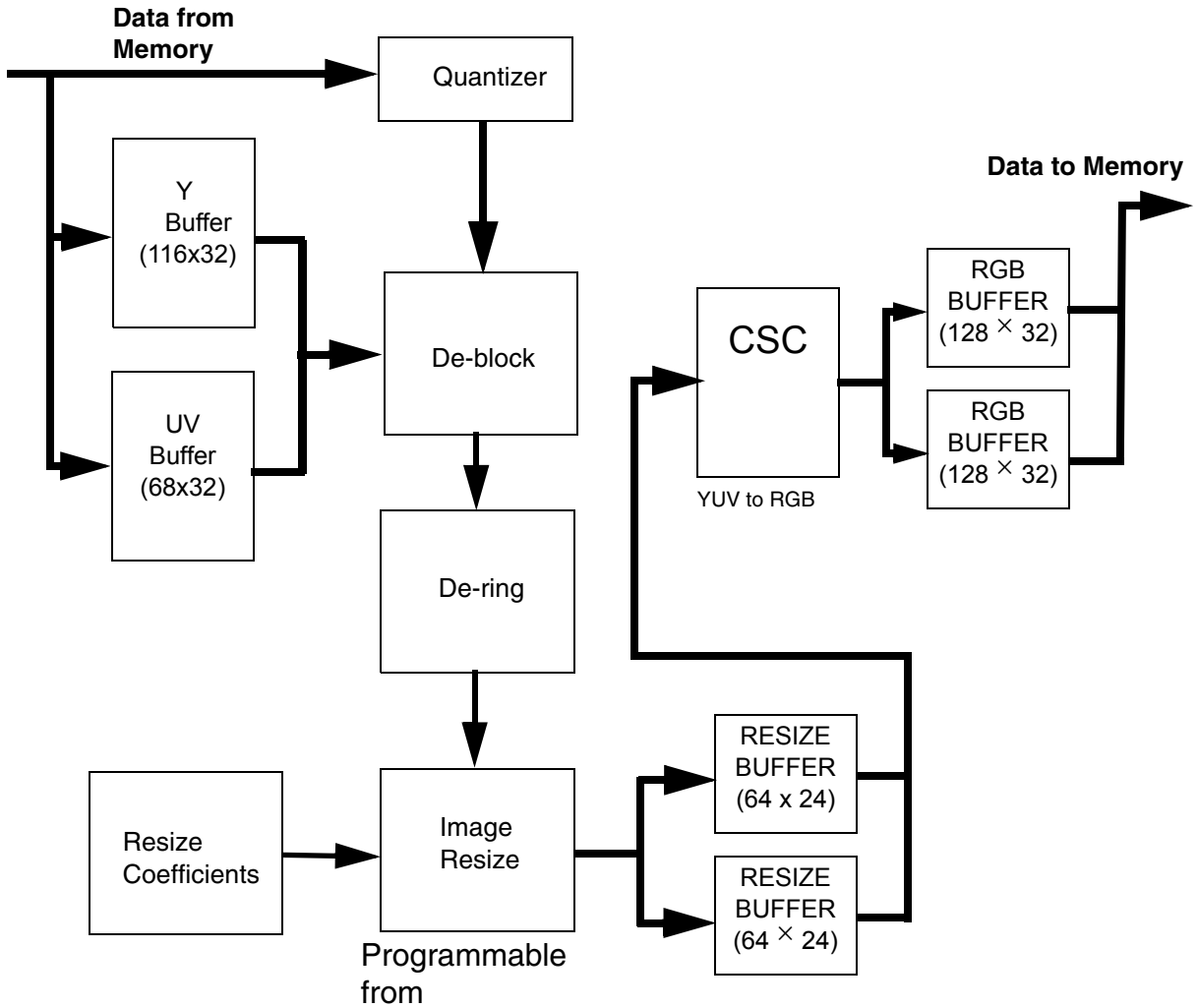


Figure 28-2. Post-Processor (PP) Block Diagram

The Post-processor (PP) performs postprocessing functions after video decoding.

The key modules in the PP are:

De-block—Removes blocking artifacts while preserving natural edges in the images. Deblock processing is bypassed if not selected.

De-ring—Removes ringing artifacts from decoded images caused by the truncation of high spatial frequencies. Subjective tests have shown that performing both de-ringing and de-blocking improves slightly the visual quality than performing de-blocking alone. De-ring processing is bypassed if not selected.

Image Resize—Scales input image to a different size—for example, from QCIF (176 × 144) to QVGA (320 × 240) to match the size of the display or to scale from one aspect ratio to another ratio—for example, from 1:1 to 4:3. It supports programmable resize ratios from 2:1 to 1:4, and a fixed 4:1. Horizontal resizing and vertical resizing are independent and can be set to different resizing ratios.

Bilinear interpolation algorithm is implemented and used for both upscaling and downscaling—that is, two adjacent pixels are loaded and multiplied by respective weighting coefficients to produce an output pixel. The weighting coefficients for a particular resize ratio are calculated by software and pre-loaded into the resize coefficient table of the PP from which the resize block reads the coefficients to use.

For example, the output samples for 3:5 bilinear interpolation can be calculated as follows:

out[0] = in[0]

out[1] = 2/5 × in[0] + 3/5 × in[1]

out[2] = 4/5 × in[1] + 1/5 × in[2]

out[3] = 1/5 × in[1] + 4/5 × in[2]

out[4] = 3/5 × in[2] + 2/5 × in[3]

The output samples for the 5:3 bilinear interpolation can be calculated as follows:

out[0] = 2/3 × in[0] + 1/3 × in[1]

out[1] = 0/3 × in[1] + 3/3 × in[2]

out[2] = 1/3 × in[3] + 2/3 × in[4]

A programmable resize engine is implemented in hardware, which reads instructions from the resize coefficient table (Register eMMA PP RESIZE COEF TABLE).

An output pixel will be generated with the value (w1 × in1 + w2 × in2)/32 and the resize engine will then read in *n* new input pixels, where in1 and in2 are two adjacent pixels. If *n* is zero, then no new pixels are read and the in1 and in2 pixel values are reused.

Each instruction in the table is in the form of (w1, n, o) where each coefficient (w1) is represented with 5 bits and n with 2 bits and ‘o’ in 1-bit. w2 is calculated as 32-w1.

NOTE

Coefficient value of 31 (5’b11111) is treated as 32 (6’b100000), consequently, coefficient values of 1 and 31 are not possible.

The Table 28-1 through Table 28-3 show examples of resize coefficients.

Table 28-1. Resize Coefficients for 3:5

w1	w2	n	Right Coefficient	Left Coefficient	in1	in2	Out
1	0	0	5'b11111 (32)	5'b00000 (0)	in[0]	–	in[0]
2/5	3/5	1	5'b01101 (13)	5'b10011 (19)	in[0]	in[1]	13/32 × in[0] + 19/32 × in[1]
4/5	1/5	1	5'b11010 (26)	5'b00110 (6)	in[1]	in[2]	26/32 × in[1] + 6/32 × in[2]
1/5	4/5	0	5'b00110 (6)	5'b11010 (26)	in[1]	in[2]	6/32 × in[1] + 26/32 × in[2]
3/5	2/5	1	5'b10011 (19)	5'b01101 (13)	in[2]	in[3]	19/32 × in[2] + 13/32 × in[3]

Table 28-2. Resize Coefficients for 5:3

w1	w2	n	Left Coefficient	Right Coefficient	in1	in2	Out
2/3	1/3	1	5'b10101 (21)	5'b01011 (11)	in[0]	in[1]	21/32 × in[0] + 11/32 × in[1]
0	1	2	5'b00000 (0)	5'b11111 (32)	in[2]	in[3]	0 × in[0] + 1 × in[1]
1/3	2/3	2	5'b01011 (11)	5'b10101 (21)	in[4]	in[5]	11/32 × in[0] + 21/32 × in[1]

Table 28-3. Resize Coefficients for 4:1

w1	w2	n	Left Coefficient	Right Coefficient	in1	in2	Out
1/2	1/2	1	5'b10000 (16)	5'b10000 (16)	in[0]	in[1]	1/2 × in[0]+ 1/2 × in[1]
0	0	1	5'b00000 (0)	5'b00000 (0)	in[1]	in[2]	–
0	0	1	5'b00000 (0)	5'b00000 (0)	in[2]	in[3]	–
0	0	1	5'b00000 (0)	5'b00000 (0)	in[3]	in[4]	–

28.2.1 Color Space Conversion (CSC)

The color space conversion block converts input images from YUV to RGB color space needed for display. The CSC block is fully programmable.

Equation used for YCbCr to RGB calculation:

```
R = C0*(Y - X0) + C1*(Cr-128)
G = C0*(Y - X0) - C2*(Cb-128) - C3*(Cr-128)
B = C0*(Y - X0) + C4*(Cb-128)
```

Equation used for YUV to RGB calculation:

```
R = C0*(Y - X0) + C1*(U-128)
G = C0*(Y - X0) - C2*(U-128) - C3*(V-128)
B = C0*(Y - X0) + C4*(U-128)
```

X0, C0, C1, C2, C3 and C4 are coefficients that can be programmed through registers PP_CSC_COEF_0123 and PP_CSC_COEF_4X.

```
C0[7:0] Range from 0 to 1.9921875 in steps of (1/128)
C1[7:0] Range from 0 to 1.9921875 in steps of (1/128)
C2[7:0] Range from 0 to 1.9921875 in steps of (1/128)
C3[7:0] Range from 0 to 1.9921875 in steps of (1/128)
C4[8:0] Range from 0 to 3.9921875 in steps of (1/128)
X0 1 - 16, 0 - 0
```

All the 10 formats defined by MPEG-4, including YUV to RGB, YCbCr to RGB and ITU-R BT 709 to RGB are supported through this programmable CSC block. The MPEG-4 standard allows for a number of conversion scenarios. The particular type of color space used by an MPEG-4 encoder is signaled in the bit stream syntax and is determined by two fields, matrix_coefficients and video_range. 5 color space conversion equations (matrix coefficients) and 2 video ranges for each equation (matrix) are defined in MPEG-4. This gives a total of 10 (5x2) color space conversion possibilities.

The 5 matrix coefficients can be categorized into two sets. The matrices represented by matrix_coefficients field values of 4, 5 and 6 are similar and can be grouped together into one set (Set A). The two remaining matrices, represented by matrix_coefficients field values of 1 and 7, are similar and hence can be grouped together into a second set (Set B). Set A and Set B represent CSC matrices in accordance with Recommendation ITU-R BT.470 and Recommendation ITU-R BT.709, respectively. For each CSC matrix, there are two possible video ranges, indicated by the video_range field which is set to either 1 or 0. Therefore there are now 2 video ranges × 2 sets = 4 CSC scenarios and each of these are summarized in Table 28-4.

Table 28-4. YUV to RGB CSC Equations

Eqn	Matrix Co-efficient	Video Range	Input to CSC and Notation	Matrix	Register Values
A1	4,5 or 6 (Set A)	1	YUV Y ranges from 0–255 U ranges from 0–255 V ranges from 0–255	R = Y + 1.4026 × (V-128) G = Y – 0.3444 × (U-128) – 0.7144 × (V-128) B = Y + 1.7730 × (U-128)	C0 = 8’b1000 0000 C1 = 8’b1011 0011 C2 = 8’b0010 1100 C3 = 8’b0101 1011 C4 = 9’b0 1110 0010 X0 = 1’b0
A0	4,5 or 6 (Set A)	0	YCrCb Y ranges from 16–235 Cr ranges from 16–240 Cb ranges from 16–240	R = 1.164*(Y – 16) + 1.596*(Cr-128) G = 1.164*(Y – 16) – 0.813*(Cr-128) – 0.391*(Cb-128) B = 1.164*(Y – 16) + 2.018*(Cb-128)	C0 = 8’b1001 0100 C1 = 8’b1100 1100 C2 = 8’b0011 0010 C3 = 8’b0110 1000 C4 = 9’b1 0000 0010 X0 = 1’b1

Table 28-4. YUV to RGB CSC Equations (continued)

Eqn	Matrix Co-efficient	Video Range	Input to CSC and Notation	Matrix	Register Values
B1	1 or 7 (Set B)	1	Y'U'V' Y' ranges from 0–255 U' ranges from 0–255 V' ranges from 0–255	R = Y' + 1.5749 × (V'-128) G = Y' – 0.1875 × (U'-128) – 0.4682 × (V'-128) B = Y' + 1.8554 × (U'-128)	C0 = 8'b1000 0000 C1 = 8'b1100 1001 C2 = 8'b0001 1000 C3 = 8'b0011 1100 C4 = 9'b0 1110 1101 X0 = 1'b0
B0	1 or 7 (Set B)	0	Y'Cr'Cb' Y' ranges from 16–235 Cr' ranges from 16–240 Cb' ranges from 16–240	R = 1.164(Y'-16) + 1.793 × (Cr'-128) G = 1.164(Y'-16) – 0.533 × (Cr'-128) – 0.213 × (Cb'-128) B = 1.164(Y'-16) + 2.112 × (Cb'-128)	C0 = 8'b1001 0100 C1 = 8'b1110 0110 C2 = 8'b0001 1011 C3 = 8'b0100 0100 C4 = 9'b1 0000 1110 X0 = 1'b1

The correct matrix must be selected based on the video_range and matrix_coefficient for color space conversion.

28.2.2 Input Interface

The PP reads IYUV or YV12 data from external memory. Quantization Parameter (QP) data is required if Deblock and/or De-ring operations are selected. Figure 28-3 shows an example layout for QCIF frames to be processed by the PP.

The input Y, U, V, and QP data are expected in 4 sections of memory. The first data in every row starts with a new word. When the row size is not a multiple of 4 bytes, the last few pixels in a row will not occupy a full word and the extra bytes, if any, in the last word are ignored.

The PP expects one QP byte per macroblock (Mbyte). Only the lower 5 bits of a QP byte are used and the 3 most significant bits must be set to 0. The first QP in every row starts with a new word and four QP bytes from 4 adjacent MBs in a row are packed into one word. When the number of MBs in a row is not a multiple of 4, then the extra QP bytes in the last word of a QP row are ignored.

Data is stored in the memory in the order of natural scan lines. For Y, U, and V data, it is permitted that the distance between the start of two neighboring lines (line stride) is greater than the number of pixels in a line—that is, there could be fixed number of unused words between the end and the beginning of every two neighboring lines. However, unused words are not permitted for QP data.

Each row in Figure 28-3 represents a word in memory. Y (j, i) denotes Y data of pixel row j and column i. Layout of U and V data is similar to that of Y data. The figure shows there can be unused space in memory between rows. This parameter is controlled by the Input Line Stride parameter and applies only to Y, U, and V data in the frame. The start of Y, U, and V data can be anywhere in addressable memory.

In a QCIF image, there are 11 × 9 = 99 MBs, representing all the QPs for a QCIF image. However, due to the packing constraints specified above, the 11 QPs of a row are packed into 3 words with one unused byte in the last word. This is repeated for each row and there can be no optional space between QP rows.

LSB			MSB	
Y(0,0)	Y(0,1)	Y(0,2)	Y(0,3)	Start of Y frame; start of line 0
Y(0,4)	Y(0,5)	Y(0,6)	Y(0,7)	
Other Y data in the current row				
Y(0,172)	Y(0,173)	Y(0,174)	Y(0,175)	End of valid data in line 0
Optional unused memory				
Y(1,0)	Y(1,1)	Y(1,2)	Y(1,3)	Start of line 1
Other Y data in the current frame				
Y(143,172)	Y(143,173)	Y(143,174)	Y(143,175)	End of valid data in line 143
Optional unused memory				
Optional gap				End of Y frame buffer
U frame buffer				
Optional gap				
V frame buffer				
Optional gap				
QP0	QP1	QP2	QP3	Start of QP frame; start of row 0 of MB
QP4	QP5	QP6	QP7	
QP8	QP9	QP10	Unused	End of row 0 of MB
QP11	QP12	QP13	QP14	Start of row 1 of MB
Other QP data in the current frame				
QP96	QP97	QP98	unused	End of QP frame buffer

Figure 28-3. PP Input Data Layout (QCIF)

28.2.3 Output Interface

The output of the PP is selectable between RGB and YUV 4:2:2 (YUYV).

RGB data is internally represented with 24 bits resolution (8 bits per color component) and color bits are truncated according to the programmed color widths. This truncation is done at the last stage of color space conversion by discarding the least significant bits. For 8 bpp output only 1:1 resize ratio is supported. Table 28-5 shows some examples for 16 bpp and unpacked 24 bpp settings.

Table 28-5. RGB Color Width and Offsets

Pixel Format	bpp	RGB Width	RGB Offset
Packed 16-bit RGB565	16	Red Width = 5 Green Width = 6 Blue Width = 5	Red Offset = 11 Green Offset = 5 Blue Offset = 0
Unpacked 32-bit RGB888	32	Red Width = 8 Green Width = 8 Blue Width = 8	Red Offset = 16 Green Offset = 8 Blue Offset = 0

28.2.4 Data Flow

The Post-Processing block reads YUV 4:2:0 data from external memory and writes processed RGB or YUYV 4:2:2 data into external memory. A typical use case of the PP in i.MX21 is as follows:

1. Software or hardware decoder decodes one frame
2. PP is programmed with frame buffer address and other ancillary information
3. Software enables PP
4. For every Mbyte read from memory, PP performs Deblock, Dering, Resize and CSC and writes to output buffer for display
5. When all MBs of the current frame are processed, PP signals frame completion to the core
6. When PP completes frame processing, it sets the frame completion status and interrupts the CPU.

28.2.5 Relationship of Register Fields Related to the Input Frame

Figure 28-4 shows how the Input Line Stride affects the area of frame that is processed by the PP. There are two rectangular areas in the diagram. The inner rectangle shows the frame area to be processed and the outer rectangle indicates the actual memory allocated for the total frame. PP_Y_SOURCE, PP_CR_SOURCE, and PP_CB_SOURCE are pointers to the start addresses of frame data. The Input Line Stride, Width, and Height parameters are automatically divided by 2 when processing the Cr and Cb frame components.

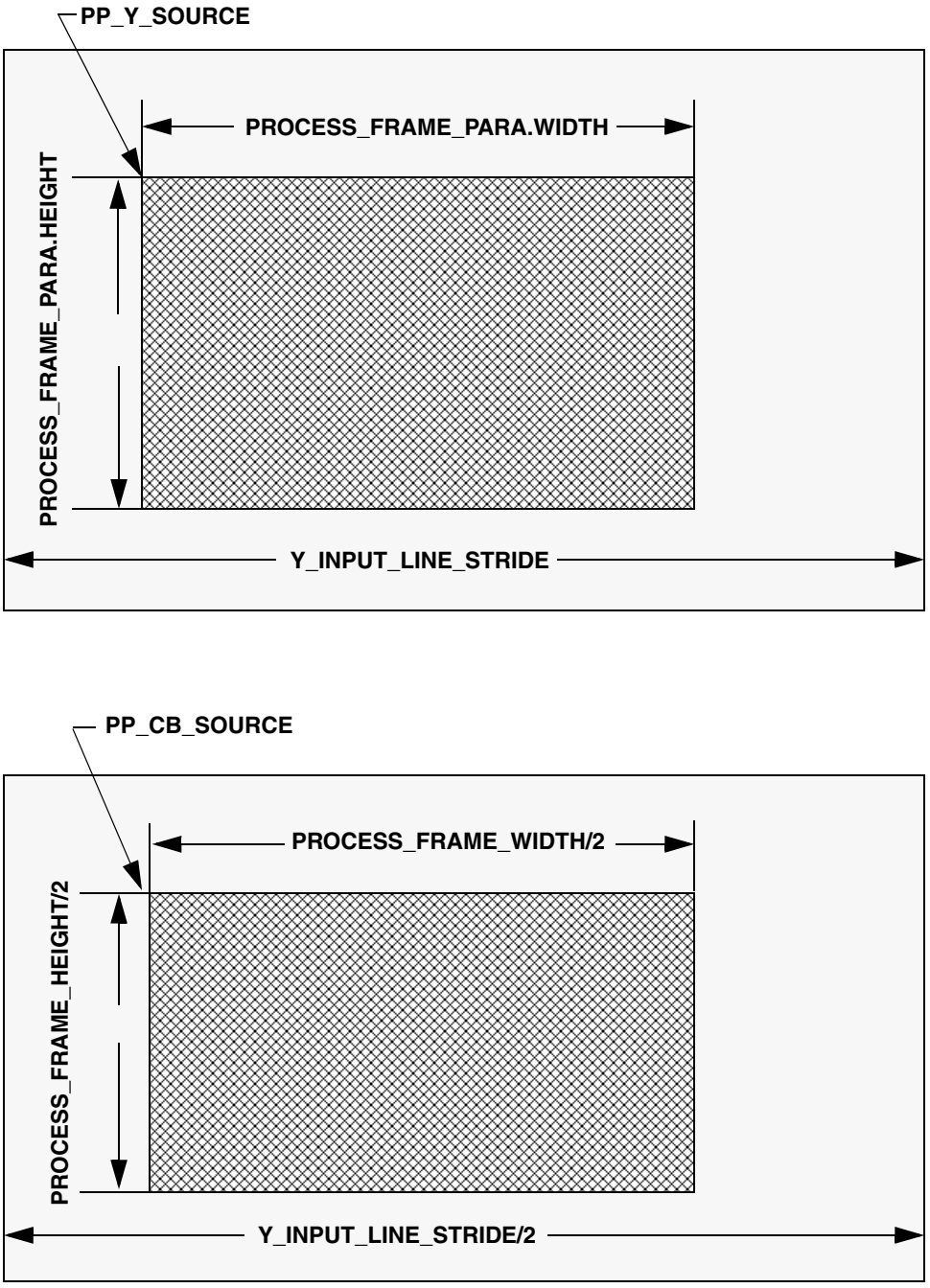


Figure 28-4. Input Line Stride

28.2.6 Relationship of Register Fields Related to Output Frame

Figure 28-5 shows the effect of the Output Line Stride parameter on the output frame. The Output Line Stride can be used to select a smaller area of the processed and resized frame. The figure shows two rectangular areas. The outer rectangle shows the memory allocated for display. The inner rectangle shows the size of the final output image. The output image size is defined by the Output Line Stride, IMAGE_WIDTH, and IMAGE_HEIGHT parameters.

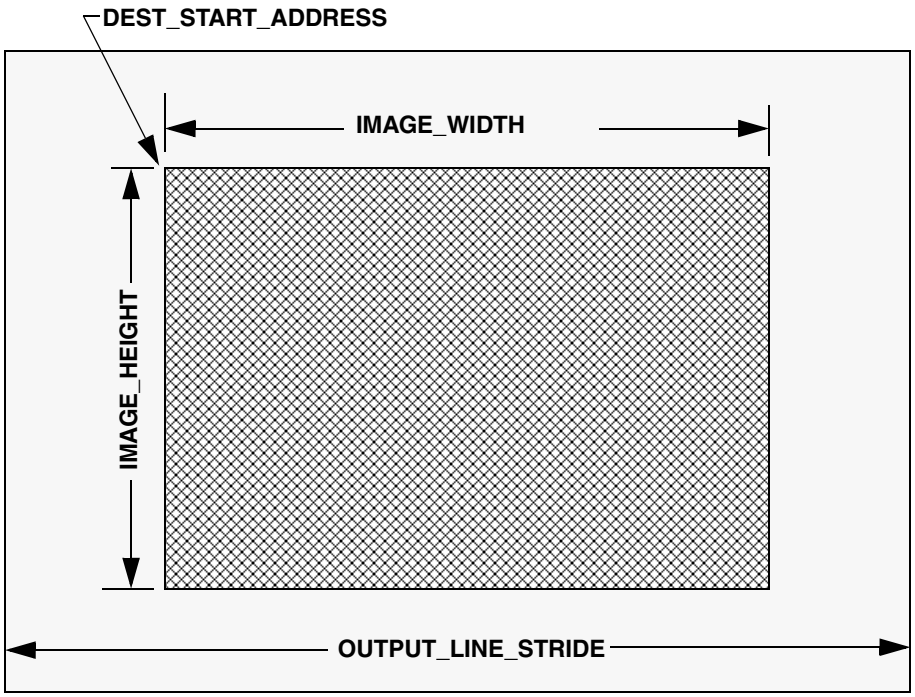


Figure 28-5. Output Line Stride

NOTE

Output Line Stride is specified in bytes while IMAGE_WIDTH and IMAGE_HEIGHT are specified in pixels.

28.3 Post Processor (PP) Programming Model

Only 32-bit accesses (read/write) is supported. All reserved bits should always be written with 0 and all registers are R/W unless specified.

Table 28-6. PP Register Summary

Description	Name	Address
PP Control Register	PP_CNTL	0x10026000
PP Interrupt Control	PP_INTRCNTL	0x10026004
PP interrupt Status	PP_INTRSTATUS	0x10026008
PP Source Y Frame data pointer	PP_SOURCE_Y_PTR	0x1002600C
PP Source CB Frame data pointer	PP_SOURCE_CB_PTR	0x10026010
PP Source CR Frame data pointer	PP_SOURCE_CR_PTR	0x10026014
PP Destination RGB Frame start address	PP_DEST_RGB_PTR	0x10026018
PP Quantizers start address	PP_QUANTIZER_PTR	0x1002601C
PP Process frame parameter, width and height	PP_PROCESS_FRAME_PARA	0x10026020
PP Source Frame width	PP_SOURCE_FRAME_WIDTH	0x10026024

Table 28-6. PP Register Summary (continued)

Description	Name	Address
PP Destination Display width	PP_DEST_DISPLAY_WIDTH	0x10026028
PP Destination Image Size	PP_DEST_IMAGE_SIZE	0x1002602C
PP Destination Frame Format Control	PP_DEST_FRAME_FMT_CNTL	0x10026030
PP Resize Table Index	PP Resize Table index Reg	0x10026034
PP CSC coefficients	PP_CSC_COEF_0123	0x10026038
PP CSC coefficients	PP_CSC_COEF_4	0x1002603C
PP Resize Coefficient Table	PP_RESIZE_COEF_TBL	0x10026100 – 0x1002619C

28.3.1 PP Control Register

PP_CNTL — PP Control Register — 0x10026000

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME																
TYPE	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME				BSDI	CSC_OUT		MB_MODE	SWRST		CSC TABLE SEL		CSCEN		DERINGEN	DEBLOCKEN	PP_EN
TYPE	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	1	0	0	0	0	1	1	1	0	1	1	0

Table 28-7. PP Control Register Description

Name	Description	Settings
Reserved Bits 31–13	Reserved—These bits are reserved and should read 0.	
BSDI Bit 12	**Byte Swap Input Data**—The input data word from memory is byte swapped (32-bit little endian to big endian or vice versa) before use.	0 = Swap disabled 1 = Swap enabled
CSC_OUT Bits 11–10	**CSC Output**—Sets RGB output resolution.	00 = 32-bit (unpacked RGB888) 01 = Reserved 10 = 16-bit 11 = 32-bit (unpacked RGB888)
Reserved Bit 9	Reserved—This bit should always read 0.	
SWRST Bit 8	**Software Reset**—Resets entire module, all registers return to their reset default values.	0 = No reset 1 = Reset
Reserved Bit 7	Reserved—This bit is reserved and should read 0.	

Table 28-7. PP Control Register Description (continued)

Name	Description	Settings
CSC_TABLE_SEL Bits 6–5	**CSC Table Select**—Selects one of the 4 CSC matrices. Please refer to Table 28-4 for more information.	00 = A1 01 = A0 10 = B1 11 = B0
CSCEN Bit 4	**CSC Enable**—Enables CSC to output RGB data else YUV 4:2:2 data when disabled. YUV 4:2:2 output is in YUYV interleaved format.	0 = YUV 4:2:2 1 = RGB
Reserved Bit 3	Reserved—This bit is reserved and should read 0.	
DERINGEN Bit 2	**De-ring Enable**—Enable or disable Dering operation.	0 = No Dering 1 = Enable Dering
DEBLOCKEN Bit 1	**De-block Enable**—Enable or disable Deblock operation.	0 = No Deblock 1 = Enable Deblock
PP_EN Bit 0	**PP Enable**—Start frame processing. Once enabled, this bit cannot be reset unless one of the following has occurred; Frame is completely processed or SWRST is set or Data abort error.	0 = Not enabled 1 = Enabled (self-clearing)

28.3.2 PP Interrupt Control Register

PP_INTRCNTL PP Interrupt Control Register 0x10026004

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME																
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME														ERRINTR_EN		FRAMECOMPINTREN
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-8. PP Interrupt Control Register Description

Name	Description	Settings
Reserved Bits 31–3	Reserved—This bit is reserved and should read 0.	
ERR_INTR_EN Bit 2	**Error Interrupt Enable**—If set, enables interrupt on error.	0 = interrupt disabled 1 = interrupt enabled
Reserved Bit 1	Reserved—This bit is reserved and should read 0.	
FRAME_COMP_INTR_EN Bit 0	**Frame Complete Interrupt Enable**—If set and in Frame mode (MB_MODE=0), enables interrupt on completion of frame processing.	0 = interrupt disabled 1 = interrupt enabled

28.3.3 PP Interrupt Status Register

PP_INTRSTATUS — PP Interrupt Status Register — 0x10026008

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME																
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME														ERR_INTR		FRAME_COM_INTR
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w1c	ro	w1c
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-9. PP Interrupt Status Register Description

Name	Description
Reserved Bits 31–3	Reserved—These bits are reserved and should be set to 0.
ERR_INTR Bit 2	**Error Interrupt Status**—If set an error has occurred. The PP must be reset (SWRST = 1) before further operations can be initiated.
Reserved Bit 1	Reserved—This bit is reserved.
FRAME_COMP_INTR Bit 0	**Frame Complete Interrupt Status**—If set, a frame has been processed.

28.3.4 PP Source Y Address Register

PP_SOURCE_Y_PTR — PP Source Y Address Register — 0x1002600C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	PP_Y_SOURCE															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	PP_Y_SOURCE															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-10. PP Source Y Address Register Description

Name	Description	Settings
PP_Y_SOURCE Bits 31–0	**PP Y Source Parameter**—32-bit frame start address of Y data (Luminance).	Bits 1–0 are always set to 0 (word aligned)

28.3.5 PP Source Cb Address Register

PP_SOURCE_CB_PTR — PP Source Cb Address Register — 0x10026010

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	PP_CB_SOURCE															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	PP_CB_SOURCE															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-11. PP Source Cb Address Register Description

Name	Description	Settings
PP_CB_SOURCE Bits 31–0	**PP CB Source Parameter**—32-bit frame start address of Cb data (U or Chrominance).	Bit 1–0 are always set to 0 (word aligned)

28.3.6 PP Source Cr Address Register

PP_SOURCE_CR_PTR — PP Source Cr Address Register — 0x10026014

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	PP_CR_SOURCE															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	PP_CR_SOURCE															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-12. PP Source Cr Address Register Description

Name	Description	Settings
PP_CR_SOURCE Bits 31–0	**PP CR Source Parameter**—32-bit frame start address of Cr data (V or Chrominance).	Bits1–0 are always set to 0 (word aligned)

28.3.7 PP Destination RGB Frame Start Address Register

PP_DEST_RGB_PTR — PP Destination RGB Frame Start Address Register — 0x10026018

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	RGB_START_ADDR															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	RGB_START_ADDR															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-13. PP Destination RGB Frame Start Address Register Description

Name	Description	Settings
RGB_START_ADDR Bits 31–0	**RGB Start Address**—Sets the destination frame start address. If CSCEN = 0, then these bits point to the start of YUV 4:2:2 (YUYV interleaved) data, else it points to RGB data.	Bit 1–0 are always set to 0 (word aligned)

28.3.8 PP Quantizer Start Address Register

PP_QUANTIZER_PTR — PP Quantizer Start Address Register — 0x1002601C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	QUANTIZER_PTR															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	QUANTIZER_PTR															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-14. PP Quantizer Start Address Register Description

Name	Description	Settings
QUANTIZER_PTR Bits 31–0	**Quantizer Parameter**—Sets the start address of the Quantization Parameter in memory. This register is ignored if Deblock and De-ring are both disabled.	Bits 1–0 are always set to 0 (word aligned)

28.3.9 PP Process Frame Parameter Register

PP_PROCESS_PARA — PP Process Parameter Register — 0x10026020

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME							PROCESS_FRAME_WIDTH									
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME							PROCESS_FRAME_HEIGHT									
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-15. PP Process Parameter Register Description

Name	Description	Settings
Reserved Bits 31–26	Reserved—These bits are reserved and should read 0.	
PROCESS_FRAME_WIDTH Bits 25–16	**Process Frame Width**—Sets the input window width to be processed based on Y frame pixel count. PROCESS_FRAME_WIDTH/2 is used for Cb and Cr window width. PROCESS_FRAME_WIDTH must always be less than or equal to INPUT_LINE_STRIDE.	This value should be a multiple of 8 pixels.
Reserved Bits 15–10	Reserved—These bits are reserved and should read 0.	
PROCESS_FRAME_HEIGHT Bits 9–0	**Process Frame Height**—Sets the input window height to be processed based on Y frame line count. PROCESS_FRAME_HEIGHT/2 is used for Cb and Cr window height.	This value should be a multiple of 8 lines.

28.3.10 PP Source Frame Width Register

PP_FRAME_WIDTH — PP Source Frame Width Register — 0x10026024

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME									QUANTIZER_FRAME_WIDTH							
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME					Y_INPUT_LINE_STRIDE											
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-16. PP Source Frame Width Register Description

Name	Description	Settings
Reserved Bits 31–24	Reserved—These bits are reserved and should read 0.	
QUANTIZER_FRAME_WIDTH Bits 23–16	**Quantizer Frame Width**—These bits set the number of bytes used to represent quantizers from all the Mbyte in a row. QP data is packed into words with possibly unused bytes in the last word when the number of Mbyte in a row in not a multiple of 4. In such cases, QUANTIZER_FRAME_WIDTH is rounded up to the nearest 4-byte multiple (word) that represents all the QP data for one row of MBs. For example, If there are 7 MBs in a row, then 2 words are required to represent the 7 QP data bytes with one unoccupied byte. The QUANTIZER_FRAME_WIDTH in this example is set as 8 (7 bytes rounded up to the nearest multiple of 4).	This value should be a multiple of 4 bytes.
Reserved Bits 15–12	Reserved—These bits are reserved and should read 0.	
Y_INPUT_LINE_STRIDE Bits 11–0	**Y Input Line Stride**—These bits set the number of pixels between adjacent rows of pixels for Y input data. Y_INPUT_LINE_STRIDE/2 is used for Cb and Cr input data.	This value must be a multiple of 8 pixels.

28.3.11 PP Destination Display Width Register

PP_DISPLAY_WIDTH — PP Destination Display Width Register — 0x10026028

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME																
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME				OUTPUT_LINE_STRIDE												
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-17. PP Destination Display Width Register Description

Name	Description	Settings
Reserved Bits 31–13	Reserved—These bits are reserved and should read 0.	
OUTPUT_LINE_STRIDE Bits 12–0	**Output Line Stride**—These bits set the distance in bytes between the start addresses of adjacent lines in the output frame. If the stride is equal to the OUT_IMAGE_WIDTH, then the stride should be calculated as: OUT_IMAGE_WIDTH × Bytes Per Pixel.	This value should be a multiple of 4 bytes.

28.3.12 PP Destination Image Size Register

PP_IMAGE_SIZE — PP Destination Image Size Register — 0x1002602C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME					OUT_IMAGE_WIDTH											
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r
RESET	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME					OUT_IMAGE_HEIGHT											
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r
RESET	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

Table 28-18. PP Destination Image Size Register Description

Name	Description	Settings
Reserved Bits 31–28	Reserved—These bits are reserved and should read 0.	
OUT_IMAGE_WIDTH Bits 27–16	**Out Image Width**—These bits set the width of the output in pixels (not bytes).	This value must always be a multiple of 2. OUT_IMAGE_WIDTH[0] is read-only and always 0.
Reserved Bits 15–12	Reserved—These bits are reserved and should read 0.	
OUT_IMAGE_HEIGHT Bits 11–0	**Out Image Height**—These bits set the number of lines in the output.	This value must always be a multiple of 2. OUT_IMAGE_HEIGHT[0] is read-only and always 0.

28.3.13 PP Destination Frame Format Control Register

PP_DEST_FRAME_FORMAT_CNTL — PP Destination Frame Format Control Register — 0x10026030

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME		RED_OFFSET					GREEN_OFFSET					BLUE_OFFSET				
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0
	0x2ca0															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME					RED_WIDTH				GREEN_WIDTH				BLUE_WIDTH			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	1	0	1	0	1	1	0	0	1	0	1
	0x0565															

NOTE

This register is used only when the output is in RGB format.

Table 28-19. PP Destination Frame Format Control Register Description

Name	Description	Settings
Reserved Bit 31	Reserved—These bits are reserved and should read 0.	
RED_OFFSET Bits 30–26	**Red Offset**—Specifies the bit offset of the Red color or Luminance component in the output pixel	The offset is specified with respect to Bit 0.
GREEN_OFFSET Bits 25–21	**Green Offset**—Specifies the bit offset of the Green color or Chrominance (U) component in the output pixel.	The offset is specified with respect to Bit 0.
BLUE_OFFSET Bits 20–16	**Blue Offset**—Specifies the bit offset of the Blue color or Chrominance (V) component in the output pixel.	The offset is specified with respect to Bit 0.
Reserved Bits 15–12	Reserved—These bits are reserved and should read 0.	
RED_WIDTH Bits 11–8	**Red Width**—Specifies the number of bits in the Red color component in the output pixel. The width of the Luminance component is fixed at 8 bits, always.	Allowed values are 0 to 8. Any value greater than 8 is fixed to 8 internally.
GREEN_WIDTH Bits 7–4	**Green Width**—Specifies the number of bits in the Green color component in the output pixel. The width of the Chrominance (Cb or U) component is fixed at 8 bits, always.	Allowed values are 0 to 8. Any value greater than 8 is fixed to 8 internally.
BLUE_WIDTH Bits 3–0	**Blue Width**—Specifies the number of bits in the Blue color component in the output pixel.The width of the Chrominance (Cr or V) component is fixed at 8 bits, always.	Allowed values are 0 to 8. Any value greater than 8 is fixed to 8 internally.

Table 28-20 shows example configurations for the YUV 4:2:2 combinations.

Table 28-20. YUV 4:2:2 Configuration Settings

YUV Format	Offsets	Width	Settings
YUYV	RED_OFFSET=24 GREEN_OFFSET=16 BLUE_OFFSET=0	RED_WIDTH=8 GREEN_WIDTH=8 BLUE_WIDTH=8	0x6200_0888
YVYU	RED_OFFSET=24 GREEN_OFFSET=0 BLUE_OFFSET=16		0x6010_0888
UYVY	RED_OFFSET=16 GREEN_OFFSET=24 BLUE_OFFSET=8		0x4308_0888
VYUY	RED_OFFSET=16 GREEN_OFFSET=8 BLUE_OFFSET=24		0x4118_0888

28.3.14 PP Resize Table Index Register

This register sets the start and end indices of horizontal and vertical resize tables. The two resize tables share a memory of 40 locations. The minimum start index is 0 and the maximum end index is 39. If the horizontal and vertical resize ratios are the same, then one resize table can be used and the horizontal and vertical start and end indices can be set to the same value. However, if the resize ratios are different, the horizontal and vertical resize tables need to be programmed differently. Either the horizontal resize table or the vertical resize table can start from address 0.

PP_RESIZE_INDEX — PP Resize Table Index Register — 0x10026034

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME			HORI_TBL_START_INDEX								HORI_TBL_END_INDEX					
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME			VERT_TBL_START_INDEX								VERT_TBL_END_INDEX					
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-21. PP Resize Table Index Register Description

Name	Description	Settings
Reserved Bits 31–30	Reserved—These bits are reserved and should read 0.	
HORI_TBL_START_INDEX Bits 29–24	**Horizontal Table Start Index**—Start index of horizontal resize table.	Valid values: 0–39

Table 28-21. PP Resize Table Index Register Description (continued)

Name	Description	Settings
Reserved Bits 23–22	Reserved—These bits are reserved and should read 0.	
HORI_TBL_END_INDEX Bits 21–16	**Horizontal Table End Index**—End index of horizontal resize table.	Valid values: 0–39
Reserved Bits 15–14	Reserved—These bits are reserved and should read 0.	
VERT_TBL_START_INDEX Bits 13–8	**Vertical Table Start Index**—Start index of vertical resize table.	Valid values: 0–39
Reserved Bits 7–6	Reserved—These bits are reserved and should read 0.	
VERT_TBL_END_INDEX Bits 5–0	**Vertical Table End Index**—End index of vertical resize table.	Valid values: 0–39

28.3.15 PP CSC COEF 123 Register

PP_CSC_COEF_123 PP CSC Coefficient_123 Register 0x10026038

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	C0								C1							
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	C2								C3							
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-22. PP CSC Coefficient_123 Register Description

Name	Description	Settings
C0 Bits 31–24	**CSC Coefficient 0**	Range from 0 to 1.9921875 in steps of (1/128)
C1 Bits 23–16	**CSC Coefficient 1**	Range from 0 to 1.9921875 in steps of (1/128)
C2 Bits 15–8	**CSC Coefficient 2**	Range from 0 to 1.9921875 in steps of (1/128)
C3 Bits 7–0	**CSC Coefficient 3**	Range from 0 to 1.9921875 in steps of (1/128)

28.3.16 PP CSC COEF_4 Register

PP_CSC_COEF_4 — PP CSC Coefficient_4 Register — 0x1002603C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME																
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME							X0	C4								
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-23. PP CSC COEF_4 Register Description

Name	Description	Settings
Reserved Bits 31–10	Reserved—These bits are reserved and should read 0.	
X0 Bit 9	**X0**—Luminance Component Offset	1 = 16 0 = 0
C4 Bits 8–0	CSC Coefficient 4	Range from 0 to 3.9921875 in steps of 1/128

Table 28-24. CSC Coefficient Usage

YUV Format	YUV / RGB Conversions
YCbCr	R = C0*(Y – X0) + C1*(Cr-128) G = C0*(Y – X0) – C2*(Cb-128) – C3*(Cr-128) B = C0*(Y – X0) + C4*(Cb-128)
YUV	R = C0*(Y – X0) + C1*(V-128) G = C0*(Y – X0) – C2*(U-128) – C3*(V-128) B = C0*(Y – X0) + C4*(U-128)

28.3.17 PP Resize Coefficient Table

PP_RESIZE_COEF_TBL	PP Resize Coefficient Table (array of 40 resize coefficients)	0x10026100 – 0x1002619C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME																
TYPE	wo	wo	wo	wo	wo	wo	wo	woo	wo	wo	wo	wo	wo	wo	wo	wo
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME									w					n		OP
TYPE	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NOTE

This is a write-only register.

Table 28-25. PP Resize Coefficient Table Register Description

Name	Description	Settings
Reserved Bits 31–8	Reserved—These bits are reserved and should read 0.	
w Bits 7–3	**Weighting Coefficient**—These bits set the weighting coefficient applied to the older of the two pixels used in the resize equation.	Valid values for weight are 0, 2 to 30, and 31. A value of 31 is treated as 32 and therefore 31 is an invalid co-efficient. The resizing algorithm uses w as the Weighting Coefficient-1, w1. w1 = w Weighting coefficient-2, w2, is calculated as: w2 = 32 - w1
n Bits 2–1	**Number of Pixels to Read**—These bits set the number of new pixels to read.	00 = No pixels are read 01 = 1 new pixel is read 10 = 2 new pixels are read 11 = 3 new pixels are read
OP Bit 0	**Output Pixels**—This bit controls if pixels are output.	1 = Pixels are output 0 = No pixels are output

28.4 Pre-Processor

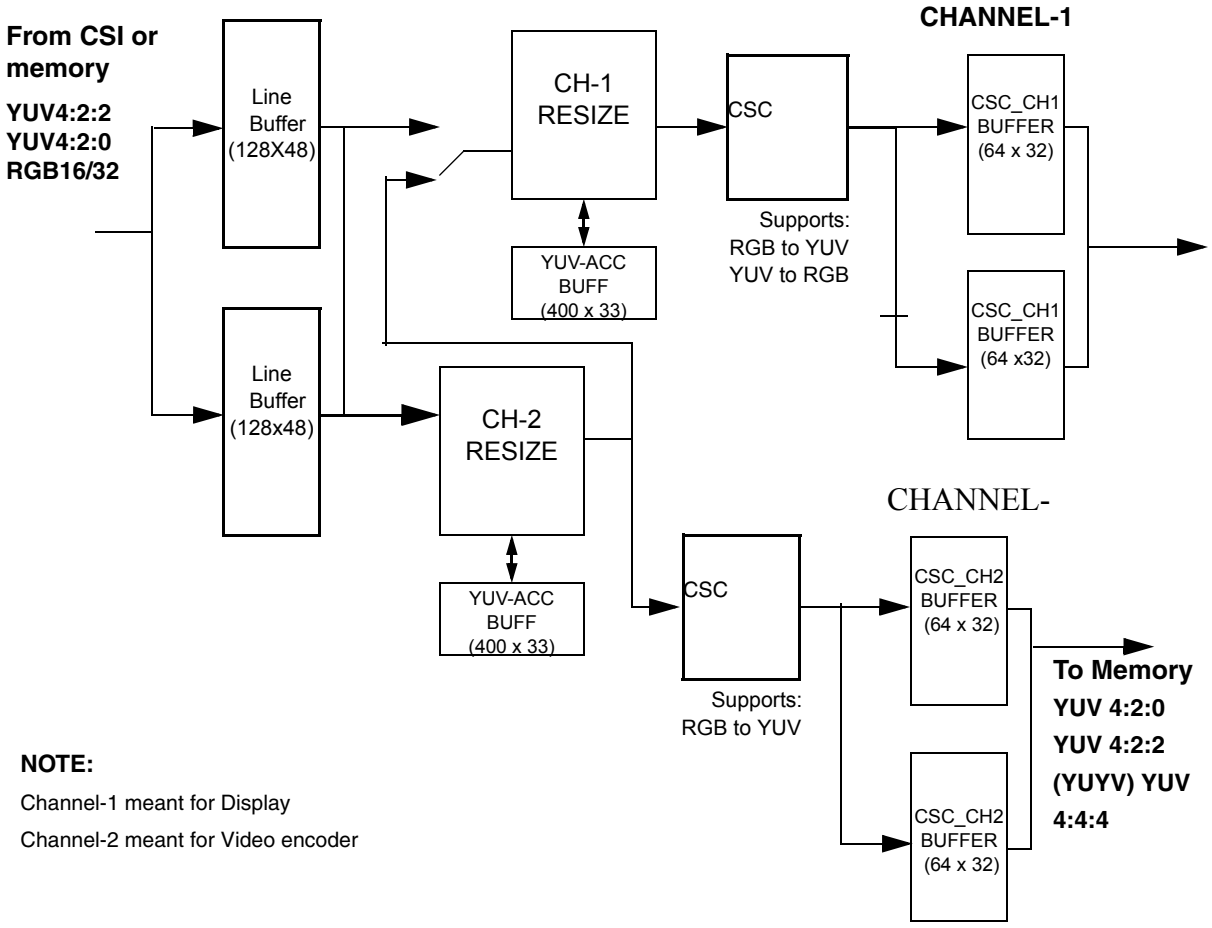


Figure 28-6. Pre-Processor Block Diagram

The Pre-processor receives input from main memory or from the camera sensor via the CMOS Sensor Interface (CSI) module and outputs two channels of data, one for video encoding and another for video display. Input data undergoes resize in a Channel-1 and Channel-2 resize block which provides programmable downscaling. The Channel-1 resize can be connected in cascade or parallel to Channel-2 resize. This is followed by programmable Color Space Conversion. The CSC block provides conversion from YUV to RGB and RGB to YUV for CH-1 and RGB to YUV for Channel-2.

After CSC, the data is channelled into memory. Channel-1 output is meant for display and both RGB and YUV 4:2:2 (interleaved) formats are supported. Data output on Channel-2 is meant for video encoding and various YUV formats are supported in this path. Namely, the YUV 4:2:0 (planar) format matches most MPEG video encoder inputs and is required by the on-chip MPEG-4 encoder.

28.4.1 Features

28.4.2 Input Data Formats

Table 28-26 shows the input data formats for the Pre-processor.

Table 28-26. Input Data Formats

Source	Format	Resolution
CSI	RGB	16 bpp
	RGB	32 bpp (unpacked RGB888)
	YUV 4:2:2	Pixel interleaved
	YUV 4:4:4	32 bpp—pixel interleaved
Memory	RGB	16 bpp
	RGB	32 bpp (unpacked RGB888)
	YUV 4:2:2	Pixel interleaved
	YUV 4:2:0	Band interleaved (IYUV and YV12)
	YUV 4:4:4	32 bpp—pixel interleaved

28.4.2.1 Input Size

The Pre-processor can accept frames as small as 32 × 32 pixels to as large as 2044 × 2044 pixels. For YUV 4:2:0, the maximum frame size is limited to 2040 × 2040 pixels.

28.4.2.2 Resize Ratios

There are two independent and identical resize blocks in the Pre-processor, called the Channel-1 resize block and the Channel-2 resize block. The Channel-1 resize block can either work in parallel to the Channel-2 resize block—that is, both blocks connect to the input, or in cascade to the output of the Channel-2 resize.

Each resize block supports two resize algorithms: bilinear and averaging. For each resize block, a user needs select one of the algorithms through register bit before starting resizing.

Table 28-27. Resize Ratios

Resize Block	Resize Ratio	Description
Channel-1 Resize	1:1	Data is copied from input to output. No resize is effected.
	Programmable from 1:1 to 8:1 or when cascaded, from 8:1 to 64:1	Downscaling
Channel-2 Resize	1:1	Data is copied from input to output. No resize is effected.
	Programmable from 1:1 to 8:1	Downscaling

28.4.2.3 Output Formats

Table 28-28 shows the output formats supported on Channel-1 and Channel-2.

Table 28-28. Output Formats

Channel	Output Format	Resolution	Description
Channel-1	RGB	8 bpp	4 pixels in an output word
		16 bpp	2 pixels in an output word
		32 bpp	Unpacked RGB888—1 pixel in an output word
	YUV 4:2:2	YUYV,YVYU,UYVY,VYUY	Pixel interleaved—2 pixels in an output word
Channel-2	YUV 4:2:2	YUYV	Pixel interleaved—2 pixels in an output word
	YUV 4:2:0	IYUV, YV12	Band interleaved
	YUV 4:4:4	YUV0	Pixel interleaved—1 pixel in an output word

28.4.2.4 Output Data

Output data is always written to memory pointed to by the destination pointers of Channel-1 and Channel-2.

28.4.2.5 Output Size

Minimum output size on Channel-1 and Channel-2 is 32 × 32 pixels and resize ratios must be calculated accordingly to prevent data truncation at the output.

28.4.3 Resize

The Channel-1 Resize and Channel-2 Resize modules are primarily used to resize captured sensor images and suitably format them to match the viewfinder display and video encoder input requirements. For example, an image or live video input from a 640 x 480 camera sensor can be resized to fit an LCD display of 240 x 320 or 320 x 320. The resizer can also prepare data for video encoding (Channel-2).

Each resize module implements two resize algorithms: bilinear and averaging, and one of the algorithms can be enabled at any single time.

28.4.3.1 Bilinear Resize in PrP

Bilinear interpolation algorithm is implemented and recommended for resize ratios between 1:1 and 2:1. Here two adjacent pixels are loaded and multiplied by respective weighting coefficients to produce an output pixel.

The weighting coefficients for a particular resize ratio are calculated by software and preloaded into the resize coefficient registers (For example, Register eMMA PrP_CH1_RZ_HORI_COEF1) of the PrP from which the resize block reads the coefficients to use.

For example, the output samples for the 5:3 bilinear interpolation can be calculated as follows:

out[0] = 5/8 × in[0] + 3/8 × in[1]

out[1] = 0/8 × in[1] + 8/8 × in[2]

out[2] = 3/8 × in[3] + 5/8 × in[4]

the entries should be (5,1),(0,1),(X,0),(3,1),(X,0).

A programmable resize engine has been implemented in hardware, which reads instructions from the PRP resize coefficient registers (Register eMMA PrP_CH1_RZ_HORI_COEF1, PrP_CH1_RZ_HORI_COEF2, PrP_CH1_RZ_VERT_COEF1, PrP_CH1_RZ_VERT_COEF2, PrP_CH2_RZ_HORI_COEF1, PrP_CH2_RZ_HORI_COEF2, PrP_CH2_RZ_VERT_COEF1, PrP_CH2_RZ_VERT_COEF2). An output pixel will be generated with the value (w1 × in1 + w2 × in2)/8 where in1 and in2 are two adjacent pixels. The resize engine will then read in one new input pixel, if the corresponding VOn bit in the corresponding registers (Register eMMA PrP_CH1_RZ_HORI_VALID, PrP_CH1_RZ_VERT_VALID, PrP_CH2_RZ_HORI_VALID, or PrP_CH2_RZ_VERT_VALID) is true 1. If the corresponding VOn bit is false 0, then no new pixels are read and the in1 and in2 pixel values are reused.

Therefore, each resize instruction is in the form of (w1, n) where each coefficient (w1) is represented with 3 bits and n with 1-bit, and stored in 2 registers, one for w1 coefficient, and one for n valid. w2 is calculated as 8-w1.

- Allowed values of w1 are 0,1,2,3,4,5,6 and 7. w1 coefficient value of 7 (3’b111) is treated as 8 (4’b1000), consequently, w1 coefficient value of 7 is not possible.
- PrP resize weighting coefficients only have 3 bits, not 5 bits as in PP resize.

28.4.3.2 Averaging Resize in PrP

This is a special convolution filter where a weighted average of every N input pixels will produce an output pixel, when the resize ratio is N:1. Suppose in[0], in[1], … in[N] are input pixels, w[i] are weighting coefficients, and out[0] is the corresponding output pixel, then

out[0] = w[0] × in[0] + w[1] × in[1] +... + w[N] × in[N].

The resize instruction for PRP averaging is also in the form of (w, n) where each coefficient (w) is represented with 3 bits and n with 1-bit. The averaging resizer is also implemented as a programmable resize engine. One pixel is loaded every cycle to the resize engine and a multiplication and accumulate operation is applied. A temporary register, T is used to store the interim result.

On reset or at the beginning of a line (for horizontal resize) or a row (for vertical resize), T is reset to 0. Then the process is as follows in every cycle:

1. Load a new input pixel value into the "in" register
2. T=T+ w*in; where $0 \le w \le 7$.
3. If n bit is true "1", then an output pixel is produced as:
   out = T/8. After that T is reset to 0.

The sum of all w coefficients (w[0]+w[1]+...+w[n]) for an output pixel will be 8. The resize instructions (w, n) are stored in 2 registers, one for w coefficient (For example PrP_CH1_RZ_HORI_COEF1), and one for n (For example PrP_CH1_RZ_HORI_VALID).

- w coefficient value of 7 (3'b111) is treated as 8 (4'b1000), consequently, w coefficient value of 7 is not possible.

Following are some example resize tables:

```
3:1: (2, 0) (4, 0) (2, 1)
4:1: (1, 0) (3, 0) (3, 0) (1, 1)
7:1: (1, 0) (1, 0) (1, 0) (2, 0) (1, 0) (1, 0) (1, 1)
```

Non N:1 resize ratios - M:N - (greater than 2:1) can be supported in averaging resize algorithm by converting into N X[i]:1 resizing ratios, where the sum of X[i], i=1…N will be M. For example, 5:2 resize can be converted into 3:1 + 2:1 resize and the resize table could be 5:2 (2, 0) (4, 0) (2, 1) (4, 0) (4, 1).

28.4.3.3 Combined Bilinear and Averaging

Can choose bi-linear and averaging at same time for two different direction. That is horizontal can be averaging and vertical be using bi-linear.

28.4.3.4 Resize Output Image Size

For M:N resize ratio the output image width is

RZOUTWIDTH = [RZINWIDTH × (N/M)] where [] integer operation.

For example if input image width is 176 and resize ratio of 5:3 then

RZWIDTHOUT = [176*3/5] = 105 pixels.

It should be noted that the height calculation will be updated.

28.4.3.5 Channel-1 Output

Table 28-29 summarizes the Channel-1 output requirements and limits. All channel-1 output must be word aligned.

Table 28-29. Channel-1 Output Formats and Sizes

Output Resolution	CH1_WIDTH	Description	Example Output Format
8 bpp	RZWIDTHOUT and ~0x03	Rounded down to a multiple of 4	RGB 332
16 bpp	RZWIDTHOUT and ~0x01	Rounded down to a multiple of 2	RGB 565 or YUV 4:2:2
32 bpp	RZWIDTHOUT	Output is word aligned	Unpacked RGB888

The final output on Channel-1 is controlled by the CH1_OUT_IMAGE_WIDTH and CH1_OUT_IMAGE_HEIGHT settings and the maximum values for these cannot exceed those of CH1_WIDTH and RZHEIGHTOUT.

28.4.3.6 Channel-2 Output

Table 28-30 summarizes the Channel-2 output requirements and limits. The inputs are from CH2 resize.

Table 28-30. Channel-2 Output Formats and Sizes

Output Format	Y_Width	Y_Height	U, V Width, and Height
YUV 4:2:0	RZWIDTHOUT and ~0x07 (multiple of 8)	RZHEIGHTOUT and ~0x01	Band interleaved: U_WIDTH = Y_WIDTH/2 V_WIDTH = Y_WIDTH/2 U_HEIGHT = Y_HEIGHT/2 V_HEIGHT = Y_HEIGHT/2
YUV 4:2:2	RZWIDTHOUT and ~0x01	RZHEIGHTOUT	Pixel interleaved
YUV 4:4:4	RZWIDTHOUT	RZHEIGHTOUT	Pixel interleaved

28.4.4 Color Space Conversion (CSC)

If an image sensor produces RGB output then Color Space Conversion (CSC) is required to convert from RGB to a YUV color space format used for MPEG-4 encoding.

The MPEG-4 standard allows for a number of Color Space Conversion (CSC) scenarios. The particular type of CSC used by an MPEG-4 encoder is signaled in the bit stream syntax and is determined by two fields, these being *matrix_coefficients* and *video_range*. To allow total flexibility, programmable CSC matrices are implemented.

Calculation of each coefficient depend upon precision or steps it need to represent. Support a coefficient is specified in steps of (1/128) and user want to represent 0.345 in that then they need to calculate as INT(128*0.345) = 44.

28.4.5 RGB to YUV

The MPEG-4 encoder only operates in YUV (or also referred to as YCbCr) color space and therefore if an image sensor produces RGB output, there is a need to perform RGB to YUV color space conversion. The MPEG-4 standard allows a number of YUV formats and these can be summarized by four color space conversion equations.

The conversion from RGB to YUV 4:2:0 (and YUV 4:2:2) can be decomposed into two steps. The first step is the conversion from RGB to YUV 4:4:4 and the second step is the down sampling from YUV 4:4:4 to YUV 4:2:0 (or YUV 4:2:2). Down sampling from YUV 4:4:4 to YUV 4:2:0 is done by skipping alternate pixel and lines. Similarly down sampling from YUV 4:4:4: to YUV 4:2:2 is done by skipping alternate pixels.

The conversion matrices for color space conversion (CSC) and the downsampling procedure is discussed below.

The MPEG-4 standard allows four CSC matrices to be employed for RGB to YUV conversion. The actual matrix that is used needs to be indicated in the bit stream to allow the decoder to choose the appropriate inverse matrix for YUV to RGB conversion. Please see Table 28-31.

Table 28-31. YUV to RGB CSC Equations

Equation Name	Matrix Coefficient	Video Range	Input to CSC and Notation	Matrix
A1	4, 5 or 6 (Set A)	1	YUV Y ranges from 0–255 U ranges from 0–255 V ranges from 0–255	Y = 0.299*R + 0.587*G + 0.114*B U = -0.169*R – 0.331*G + 0.5*B + 128 V = 0.5*R - 0.419*G - 0.081*B + 128
A0	4, 5 or 6 (Set A)	0	YCrCb Y ranges from 16–235 Cr ranges from 16–240 Cb ranges from 16–240	Y = 0.2568*R + 0.5041*G + 0.0979*B + 16 Cb = -0.1484*R - 0.2907*G + 0.4392*B + 128 Cr = 0.4392*R - 0.3680*G - 0.0711*B + 128
B1	1 or 7 (Set B)	1	Y'U'V' Y' ranges from 0–255 U' ranges from 0–255 V' ranges from 0–255	Y = 0.2126*R + 0.7152*G + 0.0722*B U = -0.115*R - 0.386*G + 0.5*B + 128 V = 0.5*R - 0.454*G - 0.046*B + 128
B0	1 or 7 (Set B)	0	Y'Cr'Cb' Y' ranges from 16–235 Cr' ranges from 16–240 Cb' ranges from 16–240	Y = 0.1826*R + 0.6142*G + 0.0620*B + 16 Cb = -0.1010*R - 0.3390*G + 0.4392*B + 128 Cr = 0.4392*R - 0.3988*G - 0.0404*B + 128

28.4.5.1 YUV to RGB

For YUV to RGB conversion matrices, please refer to Table 28-4.

28.4.5.2 Clipping of RGB and YUV Outputs

The output RGB or YUV values are clipped to the range of 0 to 255.

28.4.6 Frame Skip

Frame skipping is done to reduce the output frame rate. It can be done at three stages. Firstly at input, secondly at Channel-1 output and thirdly at Channel-2 output by using IN_SKIP,CH1_SKIP and CH2_SKIP bits of PRP_CNTL register, respectively. Frame Skip is valid only when input is from CSI (CSIEN = 1).

Input frames are skipped by using IN_SKIP then passed to Channel-1 and Channel-2. Each skip is controlled independently.

Example 1: Channel-1 and Channel-2 are configured in parallel

1. IN_SKIP = 3'b010
2. CH1_SKIP = 3'b001
3. CH2_SKIP = 3'b100
4. Let the input frame sequence be: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
5. After applying IN_SKIP, the reduced sequence is: 0, 2, 3, 5, 6, 8, 9, 11, 12. These frames are sent to Channel-1 and Channel-2 (since these are configured in parallel).
6. Output from Channel-1 after applying CH1_SKIP is: 0, 3, 6, 9, 12

7. Output from Channel-2 after applying CH2_SKIP is: 0, 2, 3, 6, 8, 9, 12

Example 2: Channel-1 cascaded with Channel-2

1. IN_SKIP = 3'b010
2. CH1_SKIP = 3'b001
3. CH2_SKIP = 3'b100
4. Let the input frame sequence be: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
5. After applying IN_SKIP, the reduced sequence is: 0, 2, 3, 5, 6, 8, 9, 11, 12. These frames are sent to Channel-2
6. Output from Channel-2 after applying CH2_SKIP: 0, 2, 3, 6, 8, 9, 12. These frames are sent to Channel-1
7. Output from Channel-1 after applying CH1_SKIP: 0, 3, 8, 12

28.4.7 LOOP Mode (LEN)

LOOP mode is effective when the CSI-PrP link is enabled and has no effect when input frames are processed from memory. LOOP mode is enabled by setting LEN bit in PRP_CNTL. When this bit is enabled, output data is written into output buffers in ping-pong fashion for each enabled channel. Two memory buffers are used for each channel output. For example, the current frame is written to Buffer 1 and the next frame to Buffer 2 followed by Buffer 1, again.

Table 28-32. Loop Mode Ping Pong Registers

Channel	Buffer-1 Registers	Buffer 2 Registers
Channel-1	PRP_DEST_RGB1_PTR	PRP_DEST_RGB2_PTR
Channel-2	PRP_DEST_Y_PTR[1] PRP_DEST_CB_PTR[2] PRP_DEST_CR_PTR	PRP_SOURCE_Y_PTR[3] PRP_SOURCE_CB_PTR PRP_SOURCE_CR_PTR

1 PRP_DEST_Y_PTR and PRP_SOURCE_Y_PTR are used in YUV 4:2:0, YUV 4:2:2 and YUV 4:4:4 modes.

2 PRP_DEST_CB_PTR, PRP_DEST_CR_PTR and PRP_DEST_CB_PTR, PRP_DEST_CR_PTR are only used when output is in YUV 4:2:0 mode.

3 These registers are re-used as output pointers in LOOP mode.

While in LOOP mode, if Channel-1 or Channel-2 is disabled, the PrP continues to output data for that channel until the frame is completed. When the channel is re-enabled again, output always starts at the next alternate buffer.

28.4.8 Channel-1 and Channel-2 Enable

CH1EN and CH2EN bits of PRP_CNTL are used to enable Channel-1 and Channel-2 processing. If a frame is to be processed by both channels then both bits should be enabled at the same time. If data input is from memory and the two channels are enabled separately one after the other, then the second channel will receive partial frame data. For example, if the input frame size is 320 × 240 and Channel-1 is enabled first and Channel-2 is enabled when 120 lines have already been processed, then lines 121 to 240 will be written to Channel-2. Line 121 will be treated as line 1 for Channel-2.

If data input is from CSI then processing always begins with a new frame. For example, when Channel-1 is enabled and processing of the first frame is in progress when Channel-2 is enabled then Channel-2 will only begin to output frames from the second frame onwards and frame boundaries are maintained in this mode.

If LOOP is not enabled then an enabled channel is automatically disabled after a single frame has been processed. If a channel is disabled in the midst of a frame, it will continue processing and stop when the full frame has been processed.

28.4.9 Channel-2 Flow Control

Flow control is done to control Channel-2 frame processing. Flow control is valid when the CSI input is enabled. Flow control is enabled by setting CH2FEN bit to ‘1’ in PRP_CNTL register. When flow control is enabled, CH2B1EN and CH2B2EN bits indicate if the corresponding buffer is ready to accept new data.

When flow control is disabled CH2B1EN and CH2B2EN are not checked and Channel-2 will write to Buffer-1 and Buffer-2, alternately. Buffer-1 and Buffer-2 are ping pong buffers whose memory address is configured as the Channel-2 destination pointers.

If flow control is enabled, then at the start of frame processing the buffer enable bits (CH2B1EN or CH2B2EN) are checked to determine if the alternate buffer is enabled for writing. If the buffer is enabled then an input frame is processed and at the end of frame, the buffer enable bit is reset. If the buffer is not enabled, then an overflow condition is signaled. The C2FCFO bit is set in PRP_INTRSTATUS. If the C2FCIE bit is enabled in PRP_INTRCNTL, then an interrupt is also raised. When an overflow condition is encountered, input frame processing stops until the buffer is enabled and processing begins at the next start of frame.

28.4.10 Line Buffer Overflow

When PRP processing speed is slower than input data arrival then line buffer overflow can happen. When a line buffer overflow occurs, the LB_OVI bit in PRP_INTRSTATUS is set. If LB_OVIE bit in PRP_INTRCNTL is enabled then an interrupt is raised.

The FRAME_SKIP bit in the PRP_CNTL register determines if processing re-starts at the next start of frame or continues when an overflow occurs. When an overflow occurs and FRAME_SKIP is’1’ then processing of that frame stops and continues at the next start of frame. When FRAME_SKIP is’0’ and an overflow occurs, then processing continues but there will be some missing data in the output frame contributing to errors in the picture.

28.4.11 Relationship of Register Fields Related to the Input Frame

Figure 28-7 shows the relationship between PrP_Y_SOURCE, PICTURE_X_SIZE, PICTURE_Y_SIZE and SOURCE_LINE_STRIDE. There are two rectangles in the diagrams. The inner rectangle shows the processed frame, however a larger amount of memory may be allocated, as bounded by the outside rectangle. This windowing relationship applies when PrP takes input from main memory.

Stride information is not used when CSI-PrP link is enabled and instead cropping takes place.

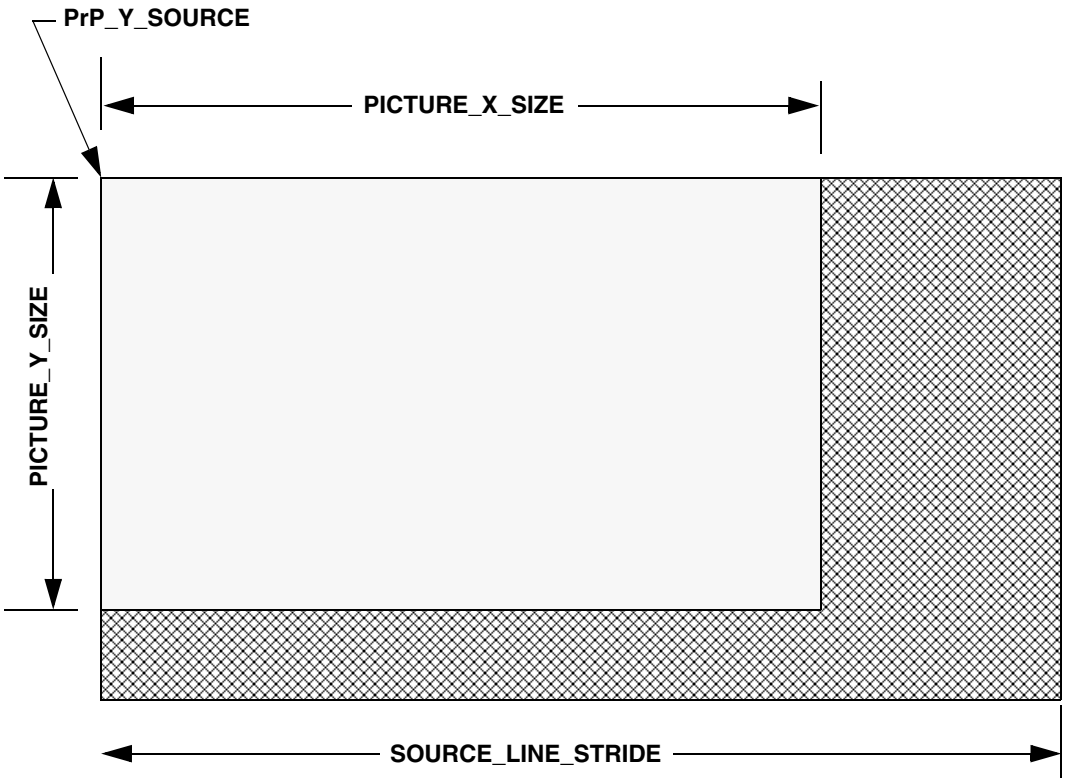


Figure 28-7. Memory Image Size and Source Line Stride

28.4.12 Relationship of Register Fields Related to Channel-1 Output Frame

Figure 28-8 shows the relationship between PrP_DEST_RGB1_PTR or PRP_DEST_RGB2_PTR, CH1_OUT_IMAGE_WIDTH, CH1_OUT_IMAGE_HEIGHT and CH1_OUT_LINE_STRIDE. There are two rectangles in the diagrams. The inner rectangle shows the Channel-1 written frame in memory, however a larger amount of memory may be allocated as bounded by the outer rectangle or the input image may be larger than the output image which is cropped.

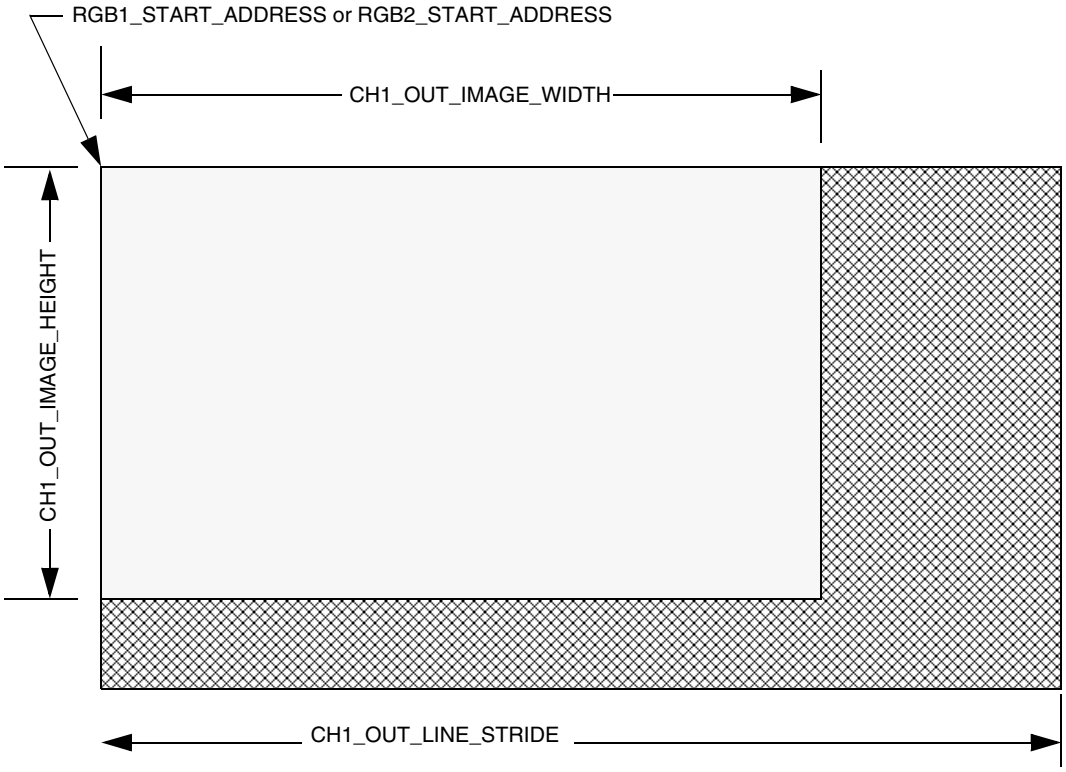


Figure 28-8. Memory Image and Output Line Stride

28.4.13 CSI Frame Cropping

The frame data from CSI can be cropped using CSI_LINE_SKIP and SOURCE_LINE_STRIDE of PRP_SRC_LINE_STRIDE register and PICTURE_X_SIZE and PICTURE_Y_SIZE of PRP_SRC_FRAME_SIZE register. When the input is 16-bit RGB or YUYV, SOURCE_LINE_STRIDE should be a multiple of two. SOURCE_LINE _STRIDE specifies the number of initial pixels to skip in a line and CSI_LINE_SKIP specifies the number of lines to skip.

The cropping parameter should be chosen such that this condition is always met.

SOURCE_LINE_STRIDE + PICTURE_X_SIZE ≤ CSI_FRAME_X_SIZE

CSI_LINE_SKIP + PICTURE_Y_SIZE ≤ CSI_FRAME_Y_SIZE

Cropping is enabled by setting WEN bit to ‘1’ in PRP_CNTL register.

Figure 28-9 shows the effect of the cropping parameters. The figure shows two rectangular areas. The outer rectangle shows the actual frame from CSI. The inner rectangle shows the size of the image user selects.

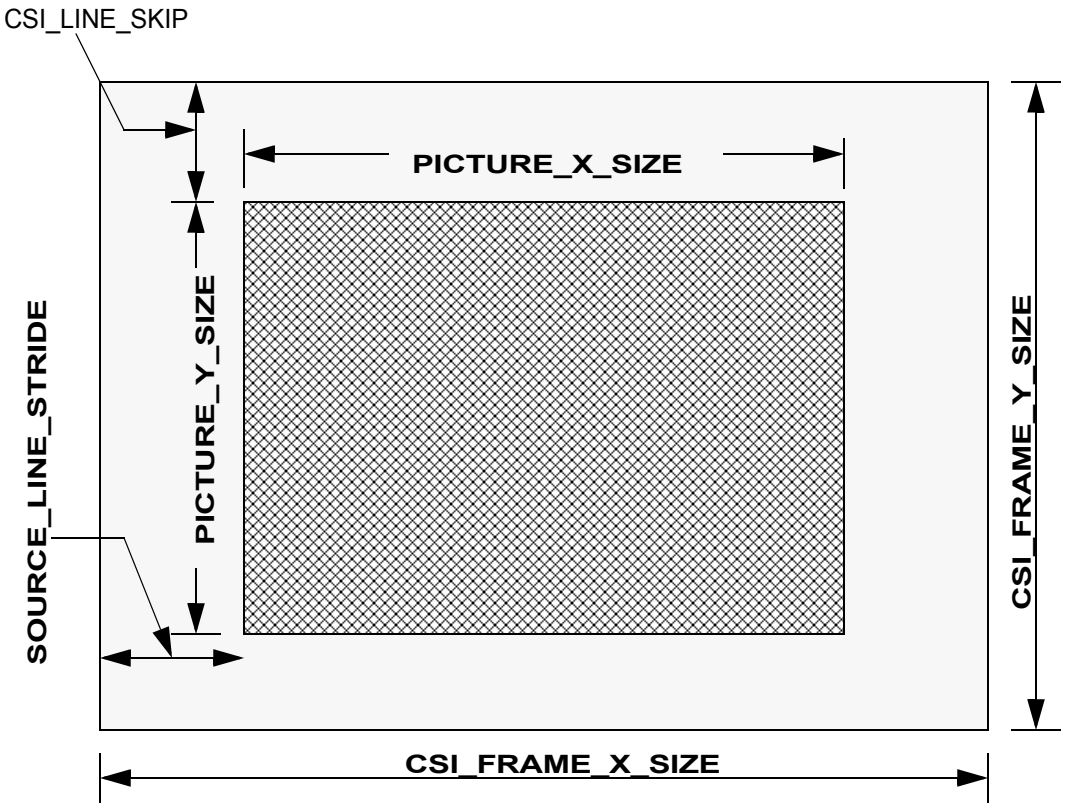


Figure 28-9. CSI Frame Cropping

NOTE

SOURCE_LINE_STRIDE and PICTURE_X_SKIP are specified in pixels and CSI_LINE_SKIP and PICTURE_Y_SIZE are specified in lines.

28.4.14 CSI-PrP Link

Figure 28-10 shows the timing diagram of the CSI-PrP dedicated link. The FIFO data from CSI is transferred to the Pre-processor without any CPU intervention.

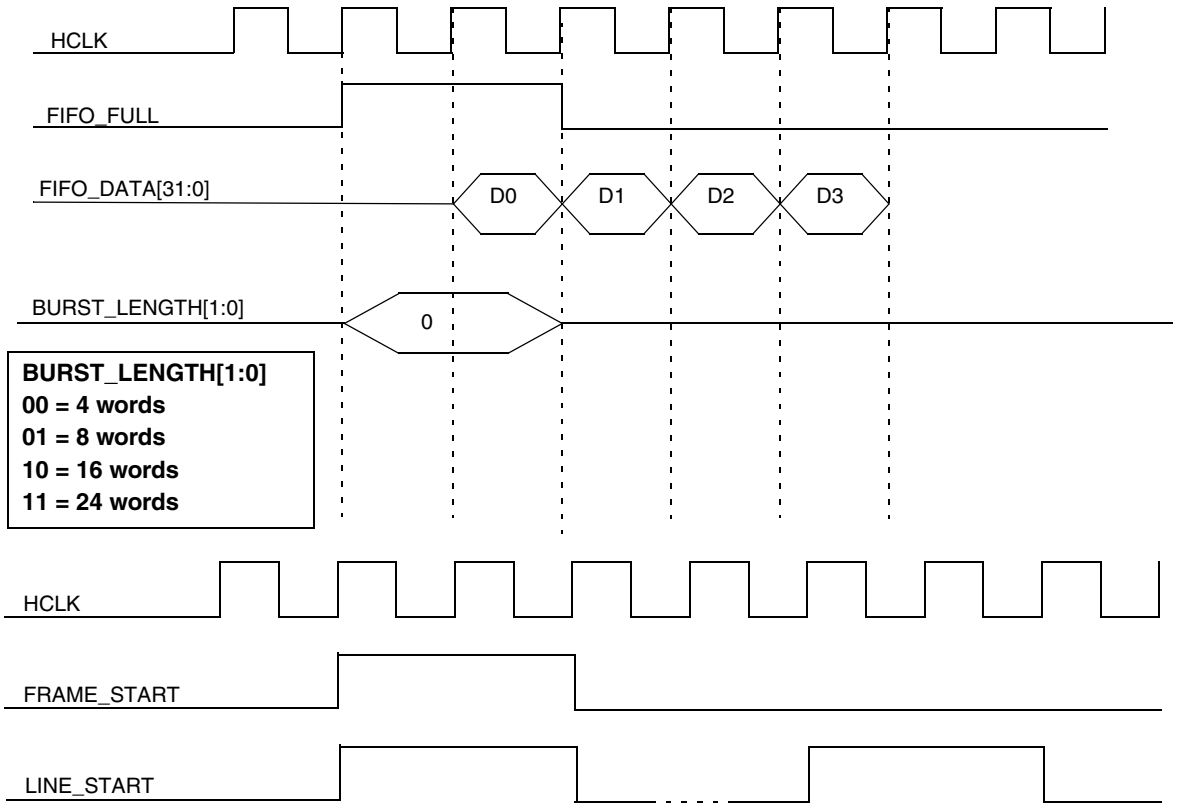


Figure 28-10. CSI-PrP Link

Table 28-33. CSI-PrP Link Internal Signals

Signal	Description
HCLK	AHB HCLK
FIFO_FULL	Asserted high when CSI FIFO has reached the high watermark
FIFO_DATA[31:0]	32-bit unidirectional data bus from CSI to PrP
FRAME_START	Asserted for 2 HCLKs when the CSI detects a Start-Of-Frame (VSYNC) condition
LINE_START	Asserted for 2 HCLKs when the CSI detects a Start-Of-Line (HSYNC) condition
BUSRT_LENGTH[1:0]	2-bit encoded to indicate to PrP how many data words there are in a FIFO_FULL burst. The FIFO high watermark must be chosen such that the PrP PICTURE_X_SIZE is a multiple of the burst length. YUV 4:2:2: Burst_count = PICTURE_X_SIZE / (BURST_LENGTH × 2) RGB (16 bpp): Burst_count = PICTURE_X_SIZE / (BURST_LENGTH × 2) RGB (32 bpp): Burst_count = PICTURE_X_SIZE / (BURST_LENGTH) Burst_count should be exact integer value.

28.5 Pre-Processor (PrP) Programming Model

All register reads and writes must be 32-bits access. Write 0 to reserved bits. All registers are read or write unless specified.

Table 28-34. PrP Register Summary

Description	Name	Address
PrP Control Register	PrP_CNTL	0x10026400
PrP Interrupt Control	PrP_INTRCNTL	0x10026404
PrP interrupt Status	PrP_INTRSTATUS	0x10026408
PrP Source Y Frame Start Address	PrP_SOURCE_Y_PTR	0x1002640C
PrP Source CB Frame Start Address	PrP_SOURCE_CB_PTR	0x10026410
PrP Source CR Frame Start Address	PrP_SOURCE_CR_PTR	0x10026414
PrP Destination RGB Frame-1 Start Address	PrP_DEST_RGB1_PTR	0x10026418
PrP Destination RGB Frame-2 Start Address	PrP_DEST_RGB2_PTR	0x1002641C
PrP Destination Y Frame Start Address	PrP_DEST_Y_PTR	0x10026420
PrP Destination CB Frame Start Address	PrP_DEST_CB_PTR	0x10026424
PrP Destination CR Frame Start Address	PrP_DEST_CR_PTR	0x10026428
PrP Source Frame Size	PrP_SOURCE_FRAME_SIZE	0x1002642C
PrP Channel-1 Line stride	PrP_CH1_LINE_STRIDE	0x10026430
PrP Source Pixel Format Control	PrP_SRC_PIXEL_FORMAT_CNTL	0x10026434
PrP CH1 Pixel Format Control	PrP_CH1_PIXEL_FORMAT_CNTL	0x10026438
PrP CH1 Output Image Size	PrP_CH1_OUT_IMAGE_SIZE	0x1002643C
PrP CH2 Output Image Size	PrP_CH2_OUT_IMAGE_SIZE	0x10026440
PrP Source Line Stride	PrP_SOURCE_LINE_STRIDE	0x10026444
PrP CSC Coefficients C0, C1, C2	PrP_CSC_COEF_012	0x10026448
PrP CSC Coefficients C3 to C5	PrP_CSC_COEF_345	0x1002644C
PrP CSC Coefficients C6 to C8	PrP_CSC_COEF_678	0x10026450
PrP Channel 1 Resize horizontal coefficients.	PrP_CH1_RZ_HORI_COEF1	0x10026454
PrP Channel 1 Resize horizontal coefficients.	PrP_CH1_RZ_HORI_COEF2	0x10026458
PrP Channel 1 Resize horizontal output data valid.	PrP_CH1_RZ_HORI_VALID	0x1002645C
PrP Channel 1 Resize vertical coefficients.	PrP_CH1_RZ_VERT_COEF1	0x10026460
PrP Channel 1 Resize vertical coefficients.	PrP_CH1_RZ_VERT_COEF2	0x10026464
PrP Channel 1 Resize vertical output data valid.	PrP_CH1_RZ_VERT_VALID	0x10026468
PrP Channel 2 Resize horizontal coefficients.	PrP_CH2_RZ_HORI_COEF1	0x1002646C
PrP Channel 2 resize horizontal coefficients.	PrP_CH2_RZ_HORI_COEF2	0x10026470
PrP Channel 2 resize horizontal output data valid.	PrP_CH2_RZ_HORI_VALID	0x10026474
PrP Channel 2 resize vertical coefficients.	PrP_CH2_RZ_VERT_COEF1	0x10026478
PrP Channel 2 resize vertical coefficients.	PrP_CH2_RZ_VERT_COEF2	0x1002647C
PrP Channel 2 resize vertical output data valid.	PrP_CH2_RZ_VERT_VALID	0x10026480

28.5.1 PrP Control Register

PRP_CNTL — PRP Control Register — 0x10026400

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	CH2FEN	CH2B2EN	CH2B1EN	RZ_FIFO _LEVEL		INPUT_FIFO _LEVEL		CH2_TSKIP			CH1_TSKIP			IN_TSKIP		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	CH1BYP	WEN	CLKEN	SWRST	SKIP FRAME	CH2_L EN	CH1_LEN	CH2_OUT _MODE		CH1_OUT _MODE		DATAIN MODE		CSIEN	CH2EN	CH1EN
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
	0x0028															

Table 28-35. PRP Control Register Description

Name	Description	Settings
CH2FEN Bit 31	**Channel 2 Flow Control Enable**—When this bit is set then output is controlled by CH2B1EN and CH2B2EN, else output is written alternately to output buffers.	0 = Disable Flow Control 1 = Enable Flow Control
CH2B2EN Bit 30	**Channel 2 Buffer 2 Enable**—This bit signals that Buffer 2 is ready for writing.	0 = Buffer is not ready for writing 1 = Buffer is ready for writing
CH2B1EN Bit 29	**Channel 2 Buffer 1 Enable**—This bit signals that Buffer 1 is ready for writing.	0 = Buffer is not ready for writing 1 = Buffer is ready for writing
RZ_FIFO_LEVEL Bits 27–28	**Resize Fifo Level**—Selects the depth of resize buffers. The buffers are 24 bits wide.	00 = 64 words 01 = 48 words 10 = 32 words 11 = 16 words
INPUT_FIFO_LEVEL Bits 25–26	**Input Fifo Level**—Selects the depth of input line buffers. The buffers are 48 bits wide. When data is sourced from CSI (CSIEN = 1) this bit should always set to “00”.	00 = 128 words 01 = 96 words 10 = 64 words 11 = 32 words
CH2_TSKIP Bits 22–24	**Channel 2 Skip Control**—Selects frames to skip on Channel 2 output. This bit is valid when CSIEN=1.	000 = no skip 001 = skip 1 out of every 2 (1-0) 010 = skip 1 out of every 3 (1-0-1) 011 = skip 2 out of every 3 (1-0-0) 100 = skip 1 out of every 4 (1-1-1-0) 101 = skip 3 out of every 4 (1-0-0-0) 110 = skip 2 out of every 5 (1-0-1-0-1) 111 = skip 4 out of every 5 (1-0-0-0-0)

Table 28-35. PRP Control Register Description (continued)

Name	Description	Settings
CH1_TSKIP Bits 19–21	**Channel 1 Skip Control**—Selects frames to skip for channel-1 output. This bit is valid when CSIEN=1.	000 = no skip 001 = skip 1 out of every 2 (1-0) 010 = skip 1 out of every 3 (1-0-1) 011 = skip 2 out of every 3 (1-0-0) 100 = skip 1 out of every 4 (1-1-1-0) 101 = skip 3 out of every 4 (1-0-0-0) 110 = skip 2 out of every 5 (1-0-1-0-1) 111 = skip 4 out of every 5 (1-0-0-0-0)
IN_TSKIP Bits 16–18	**Input Frame Skip**—Selects frames to skip from CSI. This bit is valid when CSIEN=1.	000 = no skip 001 = skip 1 out of every 2 (1-0) 010 = skip 1 out of every 3 (1-0-1) 011 = skip 2 out of every 3 (1-0-0) 100 = skip 1 out of every 4 (1-1-1-0) 101 = skip 3 out of every 4 (1-0-0-0) 110 = skip 2 out of every 5 (1-0-1-0-1) 111 = skip 4 out of every 5 (1-0-0-0-0)
CH1BYP Bit 15	**Channel-1 Bypass**—Cascade control of Channel-1 resize	0 = Cascade Channel-1 resize block 1 = Disable cascade of Channel-1 resize
WEN Bit 14	**Window Enable**—Enables input windowing feature from main memory or cropping from CSI.	0 = Disable 1 = Enable
CLKEN Bit 13	**Clock Gating Enable**—This bit controls clock gating to the PrP. When clock gating is disabled, logic in the PrP is always clocked. When clock gating is enabled, then clock is turned on when PrP module is enabled.	1 = Clock gating off 0 = Clock gating on (power saving feature)
SWRST Bit 12	**Software Reset**—Writing 1 to this bit resets entire PrP module. All registers return to their default values.	This bit is self-clearing.
SKIP_FRAME Bit 11	**Frame Skip**—This bit selects whether to continue or stop when we get input FIFO overflow error.	0 = Continue 1 = Stop
CH2_LEN Bit 10	**Channel-2 Loop Enable**—This bit is valid only when input data is from CSI (CSIEN = 1). When enabled, output data is written in ping-pong fashion.	0 = Disable 1 = Enable
CH1_LEN Bit 9	**Channel-1 Loop Enable**—This bit is valid only when input data is from CSI (CSIEN = 1). When enabled, output data is written in ping-pong fashion.	0 = Disable 1 = Enable
CH2_OUT_MODE Bits 7–8	**Channel-2 Output Mode**—These bits select Channel-2 output format.	00 or 11 = YUV 4:2:0 (IYUV, YV12) 01 = YUV 4:2:2 (YUYV) 10 = YUV 4:4:4 (YUV0)
CH1_OUT_MODE Bit 5–6	**Channel-1 Output Mode**—These bits select the Channel-1 output format.	00 = 8 bpp RGB 01 = 16 bpp RGB 10 = 32 bpp RGB 11 = YUV 422
DATA_IN_MODE Bits 3–4	**Data Input Mode**—These bits set the input data format.	00 = YUV 4:2:0 (not supported for CSIEN=1) 01 = YUV 4:2:2 10 = 16-bit RGB data 11 = 32-bit RGB data

Table 28-35. PRP Control Register Description (continued)

Name	Description	Settings
CSIEN Bit 2	**CSI Enable**—If set, enables the CSI-PrP link and input is sourced from the CSI. When cleared, input is sourced from main memory.	0 = input from main memory 1 = input from CSI
CH2EN Bit 1	**Channel-2 Enable**—This bit enables or disables Channel-2 output.	0 = disable 1 = enable When enabled, self clearing on success or error. Does not self clear if in LOOP mode (CH2_LEN = 1)
CH1EN Bit 0	**Channel-1 Enable**—This bit enables or disables Channel-1 output.	0 = disable 1 = enable When enabled, self clearing on success or error. Does not self clear if in LOOP mode (CH1_LEN = 1)

28.5.2 PrP Interrupt Control Register

PRP_INTR_CNTL — PRP Interrupt Control Register — 0x10026404

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME																
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME								CH2 OVFIE	LB OVFIE		CH2FC IE		CH1FCIE	CH2 WERRIE	CH1 WERRIE	RD ERRIE
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-36. PRP Interrupt Control Register Description

Name	Description	Settings
Reserved Bits 31–9	Reserved—These bits are reserved and should read 0.	
CH2OVFIE Bit 8	**Channel-2 Overflow Interrupt Enable**—If set, an interrupt is generated when frame is missed due to flow control process.	0 = disable 1 = enable
LBOVFIE Bit 7	**Line Buffer Overflow Interrupt Enable**—If set, an interrupt is generated when Line buffer overflow occurs.	0 = disable 1 = enable
Reserved Bit 6	Reserved—This bit is reserved and should read 0.	
CH2FCIE Bit 5	**Channel-2 Frame Complete Interrupt Enable**—If set, an interrupt is generated when a frame is completely written out through Channel-2.	0 = disable 1 = enable

Table 28-36. PRP Interrupt Control Register Description (continued)

Name	Description	Settings
Reserved Bit 4	Reserved—This bit is reserved and should read 0.	
CH1FCIE Bit 3	**Channel-1 Frame Complete Interrupt Enable**—If set, an interrupt is generated when a frame is completely written out through Channel-1.	0 = disable 1 = enable
CH2WERRIE Bit 2	**Channel-2 Write Error Interrupt Enable**—If set, an interrupt is generated when an AHB write error is encountered during Channel-2 data output to memory.	0 = disable 1 = enable
CH1WERRIE Bit 1	**Channel-1 Write Error Interrupt Enable**—If set, an interrupt is generated when an AHB write error is encountered during Channel-1 data output to memory.	0 = disable 1 = enable
RDERRIE Bit 0	**Read Error Interrupt Enable**—If set, an interrupt is generated when an AHB read error is encountered during data input from memory.	0 = disable 1 = enable

28.5.3 PrP Interrupt Status Register

PRP_INTRSTATUS PrP Interrupt Status Register 0x10026408

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME																
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME								CH2 OVF	LB OVF	CH1 B1CI	CH1 B2CI	CH2 B1CI	CH2 B2CI	CH2 WRERR	CH1 WRERR	RDERR
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-37. PrP Interrupt Status Register Description

Name	Description	Settings
Reserved Bits 31–9	Reserved—These bits are reserved and should read 0.	
CH2OVF Bit 6	**Channel-2 Buffer Overflow**—When this bit is set, it indicates that a frame was missed because both CH2B1EN and CH2B2EN are disabled.	–
LBOVF Bit 6	**Line Buffer Overflow**—When this bit is set, it indicates that a line buffer overflow has occurred.	–
CH1B1CI Bit 6	**Channel-1 Buffer-1 Complete Interrupt**—If Channel-1 is enabled and when this bit is set, it indicates that a frame has been completely written into Buffer-1 of Channel-1.	This bit is set when input is either from CSI or from memory.
CH1B2CI Bit 5	**Channel-1 Buffer-2 Complete Interrupt**—If Channel-1 is enabled and when this bit is set, it indicates that a frame has been completely written into Buffer-2 of Channel-1.	This bit is set only when CSIEN=1 and LEN=1.

Table 28-37. PrP Interrupt Status Register Description (continued)

Name	Description	Settings
CH2B1CI Bit 4	**Channel-2 Buffer-1 Complete Interrupt**—If Channel-2 is enabled and when this bit is set, it indicates that a frame has been completely written into Buffer-1 of Channel-2.	This bit is set when input is either from CSI or from memory.
C2B2CI Bit 3	**Channel-2 Buffer-2 Complete Interrupt**—If Channel-2 is enabled and when this bit is set, it indicates that a frame has been completely written into Buffer-2 of Channel-2.	This bit is set only when CSIEN=1 and LEN=1.
CH2WRERR Bit 2	**Channel-2 Write Error**—If set, then an AHB write error to memory was encountered during Channel-2 data output. PrP must be reset (SWRST=1) and re-initialized.	–
CH1WRERR Bit 1	**Channel-1 Write Error**—If set, then an AHB write error was encountered during Channel-1 data output. PrP must be reset (SWRST=1) and re-initialized.	–
READERR Bit 0	**Read Error**—If set, then an AHB read error was encountered during data input from memory. PrP must be reset (SWRST=1) and re-initialized.	–

28.5.4 PrP Source Y Address Register

PRP_SOURCE_Y_PTR — PrP Source Y Address Register — 0x1002640C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	PRP_Y_SOURCE															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	PRP_Y_SOURCE															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-38. PrP Source Y Address Register Description

Name	Description
PRP_SOURCE_Y_PTR Bits 31–0	**PRP_SOURCE_Y_PTR**—32-bit pointer to memory. In RGB and YUV 4:2:2 modes, this register sets the frame start address. In YUV 4:2:0 mode (input or output), this register sets the Luminance band start address of the frame. When CSIEN=1 and LEN=1, this register is re-used as the Channel-2 output Buffer-2 destination address for the above formats.

28.5.5 PrP Source Cb Address Register

PRP_SOURCE_CB_PTR — PrP Source Cb Address Register — 0x10026410

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	PRP_CB_SOURCE															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	PRP_CB_SOURCE															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-39. PrP Source Cb Address Register Description

Name	Description
PRP_SOURCE_CB_PTR Bits 31–0	**PRP Source CB Parameter**—32-bit pointer to memory. This register sets the Chrominance (U or Cb) band start address when input is in YUV 4:2:0 band interleaved mode. It is not used for other input modes. When CSIEN=1 and LEN=1, this register is re-used as the Channel-2 U or Cb output Buffer-2 destination address for YUV 4:2:0.

28.5.6 PrP Source Cr Address Register

PRP_SOURCE_CR_PTR — PrP Source Cr Address Register — 0x10026414

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	PRP_CR_SOURCE															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	PRP_CR_SOURCE															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-40. PrP Source Cr Address Register Description

Name	Description
PRP_SOURCE_CR_PTR Bits 31–0	**PRP Source CR Parameter**—32-bit pointer to memory. This register sets the Chrominance (V or Cr) band start address when input is in YUV 4:2:0 band interleaved mode. It is not used for other input modes. When CSIEN=1 and LEN=1, this register is re-used as the Channel-2 Y or Cr output Buffer-2 destination address for YUV 4:2:0.

28.5.7 PrP Destination RGB1 Frame Start Address Register

PRP_DEST_RGB1_PTR — PrP Destination RGB1 Start Address Register — 0x10026418

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	RGB1_START_ADDRESS															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	RGB1_START_ADDRESS															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-41. PrP Destination RGB1 Start Address Register Description

Name	Description
PRP_DEST_RGB1_PTR Bits 31–0	**PRP Destination RGB1 Parameter**—32-bit pointer to memory. This register sets the RGB or YUV 4:2:2 frame start address for Channel-1 output. When CSIEN=1 and LEN=1, this register becomes the output Buffer-1 address for Channel-1.

28.5.8 PrP Destination RGB2 Frame Start Address Register

PRP_DEST_RGB2_PTR — PrP Destination RGB2 Start Address Register — 0x1002641C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	RGB2_START_ADDRESS															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	RGB2_START_ADDRESS															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-42. PrP Destination RGB2 Start Address Register Description

Name	Description
PRP_DEST_RGB2_PTR Bits 31–0	**PRP Destination RGB2 Parameter**—32-bit pointer to memory. This register sets the output Buffer-2 RGB or YUV 4:2:2 frame start address for Channel-1 and is used only when CSIEN=1 and LEN=1.

28.5.9 PrP Destination Y Address Register

PRP_DEST_Y_PTR — PrP Destination Y Address Register — 0x10026420

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	PRP_Y_DEST															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	PRP_Y_DEST															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-43. PrP Destination Y Address Register Description

Name	Description
PRP_Y_DEST Bits 31–0	**PRP Y Destination**—32-bit pointer to memory. This register sets the destination start address for Channel-2 output. This register is the Luminance band start address in YUV 4:2:0 mode and frame start address in YUV 4:2:2 and YUV 4:4:4 pixel interleaved modes. When CSIEN=1 and LEN=1, this register is the Buffer-1 output address.

28.5.10 PrP Destination Cb Address Register

PRP_DEST_CB_PTR — PrP Destination Cb Address Register — 0x10026424

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	PRP_CB_DEST															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	PRP_CB_DEST															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-44. PrP Destination Cb Address Register Description

Name	Description
PRP_CB_DEST Bits 31–0	**PRP CB Destination**—32-bit pointer to memory. This register is used in YUV 4:2:0 band interleaved mode and sets the destination start address for Channel-2 U or Cb band data output. When CSIEN=1 and LEN=1, this register is the Channel-2 Buffer-1 output address for U or Cb band data in YUV 4:2:0 mode.

28.5.11 PrP Destination Cr Address Register

PRP_DEST_CR_PTR — PrP Destination Cr Address Register — 0x10026428

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	PRP_CR_DEST															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	PRP_CR_DEST															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-45. PrP Destination Cr Address Register Description

Name	Description
PRP_CR_DEST Bits 31–0	**PRP CR Destination**—32-bit pointer to memory. This register is used in YUV 4:2:0 band interleaved mode and sets the destination start address for Channel-2 V or Cr band data output. When CSIEN=1 and LEN=1, this register is the Channel-2 Buffer-1 output address for V or Cr band data in YUV 4:2:0 mode.

28.5.12 PrP Source Frame Size Register

PRP_SRC_FRAME_SIZE — PrP Source Frame Size Register — 0x1002642C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME						PICTURE_X_SIZE										
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME						PICTURE_Y_SIZE										
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

Table 28-46. PrP Source Frame Size Register Description

Name	Description	Settings
Reserved Bits 31–27	Reserved—These bits are reserved and should read 0.	
PICTURE_X_SIZE Bits 26–16	**Picture X Size**—These bits set the frame width to be processed in number of pixels. In YUV 4:2:0 mode, Cb and Cr widths are taken as PICTURE_X_SIZE/2 pixels.	In YUV 4:2:0 mode, this value should be a multiple of 8-pixels. In other modes (RGB, YUV 4:2:2 and YUV 4:4:4) it should be a multiple of 4 pixels.
Reserved Bits 15–11	Reserved—These bits are reserved and should read 0.	
PICTURE_Y_SIZE Bits 10–0	**Picture Y Size**—These bits set the frame height in number of lines to be processed. In YUV 4:2:0 mode, Cb and Cr lines to be processed are taken as PICTURE_Y_SIZE/2.	In YUV 4:2:0 mode, this value should be a multiple of 2.

28.5.13 PrP Destination Channel-1 Line Stride Register

PRP_DEST_CH1_LINE_STRIDE PrP Destination Channel-1 Line Stride Register 0x10026430

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME																
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME					CH1_OUT_LINE_STRIDE											
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0

Table 28-47. PrP Destination Channel-1 Line Stride Register Description

Name	Description	Settings
Reserved Bits 31–12	Reserved—These bits are reserved and should read 0.	
CH1_OUT_LINE_STRIDE Bits 11–0	**CH1 Out Line Stride**—These bits sets the distance in bytes between the start addresses of adjacent lines in Channel-1 output.	The stride value should be a multiple of 4-bytes.

28.5.14 PrP Source Pixel Format Control Register

PRP_SRC_PIXEL_FORMAT_CNTL — PrP Source Pixel Format Control Register — 0x10026434

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME		RED_Y_OFFSET					GREEN_U_CB_OFFSET					BLUE_V_CR_OFFSET				
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME					RED_WIDTH				GREEN_WIDTH				BLUE_WIDTH			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0

Table 28-48. PrP Source Pixel Format Control Register Description

Name	Description	Settings
Reserved Bit 31	Reserved—This bit is reserved and should read 0.	
RED_Y_OFFSET Bits 30–26	**Red Y Offset**—These bits set the offset of the Red color or Y luminance component in the input pixel. The offset is calculated with respect to Bit 0.	
GREEN_U_CB_OFFSET Bits 25–21	**Green U CB Offset**—These bits set the offset of the Green color, U or Cb Chrominance component in the input pixel. The offset is calculated with respect to Bit 0.	
BLUE_V_CR_OFFSET Bits 20–16	**Blue V CR Offset**—These bits set the offset of the Blue color, V or Cr Chrominance component in the input pixel. The offset is calculated with respect to Bit 0.	
Reserved Bits 15–12	Reserved—These bits are reserved and should read 0.	
RED_WIDTH Bits 11–8	**Red Width**—These bits set the width in bits of the Red color component in the input pixel.	Valid values are 0 to 8. Any value greater than 8 is set to 8.
GREEN_WIDTH Bits 7–4	**Green Width**—These bits set the width in bits of the Green color component in the input pixel.	Valid values are 0 to 8. Any value greater than 8 is set to 8.
BLUE_WIDTH Bits 3–0	**Blue Width**—These bits set the width in bits of the Blue color component in the input pixel.	Valid values are 0 to 8. Any value greater than 8 is set to 8.

Table 28-49 provides some examples of common input formats and the respective settings.

Table 28-49. Example Source Input Pixel Formats

Input Format	Pixel Arrangement	Offset	Width	PRP_SRC_PIXEL_FORMAT_CNTL
RGB 565	16 bpp	RED_Y_OFFSET=11 GREEN_U_CB_OFFSET=5 BLUE_V_CR_OFFSET=0	RED_WIDTH=5 GREEN_WIDTH=6 BLUE_WIDTH=5	0x2CA0_0565
RGB 888	Unpacked RGB888	RED_Y_OFFSET=16 GREEN_U_CB_OFFSET=8 BLUE_V_CR_OFFSET=0	RED_WIDTH=8 GREEN_WIDTH=8 BLUE_WIDTH=8	0x4100_0888
YUV 4:2:0	IYUV	Not applicable as input is band interleaved		Register is not used.
YUV 4:2:0	YV12			
YUV 4:2:2	YUYV	RED_Y_OFFSET=8 GREEN_U_CB_OFFSET=16 BLUE_V_CR_OFFSET=0	Width is not used, set to 8 by default.	0x2200_0888
YUV 4:2:2	YVYU	RED_Y_OFFSET=8 GREEN_U_CB_OFFSET=0 BLUE_V_CR_OFFSET=16		0x2010_0888
YUV 4:2:2	UYVY	RED_Y_OFFSET=0 GREEN_U_CB_OFFSET=24 BLUE_V_CR_OFFSET=8		0x0308_0888
YUV 4:2:2	VYUY	RED_Y_OFFSET=0 GREEN_U_CB_OFFSET=8 BLUE_V_CR_OFFSET=24		0x0118_0888
YUV 4:4:4	YUV0	RED_Y_OFFSET=24 GREEN_U_CB_OFFSET=16 BLUE_V_CR_OFFSET=8		0x6208_0888

28.5.15 PrP Channel-1 Pixel Format Control Register

PRP_CH1_PIXEL_FORMAT_CNTL — PrP CH1 Pixel Format Control Register — 0x10026438

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME		RED_OFFSET					GREEN_OFFSET					BLUE_OFFSET				
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	1	0	1	1	0	0	1	0	1	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME					RED_WIDTH				GREEN_WIDTH				BLUE_WIDTH			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	1	0	1	0	1	1	0	0	1	0	1

Table 28-50. PrP CH1 Pixel Format Control Register Description

Name	Description	Settings
Reserved Bit 31	Reserved—This bit is reserved and should read 0.	
RED_OFFSET Bits 30–26	**Red Offset**—These bits set the offset of the Red color component in the output pixel. The offset is calculated with respect to Bit 0.	
GREEN_OFFSET Bits 25–21	**Green Offset**—These bits set the offset of the Green color component in the output pixel. The offset is calculated with respect to Bit 0.	
BLUE_OFFSET Bits 20–16	**Blue Offset**—These bits set the offset of the Blue color component in the output pixel. The offset is calculated with respect to Bit 0.	
Reserved Bits 15–12	Reserved—These bits are reserved and should read 0.	
RED_WIDTH Bits 11–8	**Red Width**—These bits set the width in bits of the Red color component in the input pixel.	Valid values are 0 to 8. Any value greater than 8 is set to 8.
GREEN_WIDTH Bits 7–4	**Green Width**—These bits set the width in bits of the Green color component in the input pixel.	Valid values are 0 to 8. Any value greater than 8 is set to 8.
BLUE_WIDTH Bits 3–0	**Blue Width**—These bits set the width in bits of the Blue color component in the input pixel.	Valid values are 0 to 8. Any value greater than 8 is set to 8.

Table 28-51 shows some example output RGB formats and encodings.

Table 28-51. Example Channel-1 RGB Output Pixel Format

Input Format	Pixel Arrangement	Offset	Width	PRP_CH1_PIXEL_FORMAT_CNTL
RGB 332	8 bpp	RED_OFFSET=5 GREEN_OFFSET=2 BLUE_OFFSET=0	RED_WIDTH=3 GREEN_WIDTH=3 BLUE_WIDTH=2	0x1440_0332
RGB 565	16 bpp	RED_OFFSET=11 GREEN_OFFSET=5 BLUE_OFFSET=0	RED_WIDTH=5 GREEN_WIDTH=6 BLUE_WIDTH=5	0x2CA0_0565
RGB 888	Unpacked RGB888	RED_OFFSET=16 GREEN_OFFSET=8 BLUE_OFFSET=0	RED_WIDTH=8 GREEN_WIDTH=8 BLUE_WIDTH=8	0x4100_0888
YUV 4:2:2	YUYV	RED_Y_OFFSET=24 GREEN_U_CB_OFFSET=16 BLUE_V_CR_OFFSET=0	RED_WIDTH=8 GREEN_WIDTH=8 BLUE_WIDTH=8	0x6200_0888
YUV 4:2:2	YVYU	RED_Y_OFFSET=24 GREEN_U_CB_OFFSET=0 BLUE_V_CR_OFFSET=16		0x6010_0888
YUV 4:2:2	UYVY	RED_Y_OFFSET=16 GREEN_U_CB_OFFSET=24 BLUE_V_CR_OFFSET=8		0x4308_0888
YUV 4:2:2	VYUY	RED_Y_OFFSET=16 GREEN_U_CB_OFFSET=8 BLUE_V_CR_OFFSET=24		0x4118_0888

28.5.16 PrP Destination Channel-1 Output Image Size Register

PRP_CH1_OUT_IMAGE_SIZE — PrP Destination Channel-1 Output Image Size Register — 0x1002643C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME						CH1_OUT_IMAGE_WIDTH										
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
	0x0140															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME						CH1_OUT_IMAGE_HEIGHT										
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
	0x00F0															

Table 28-52. PrP Destination Channel-1 Output Image Size Register Description

Name	Description	Settings
Reserved Bits 31–27	Reserved—These bits are reserved and should read 0.	
CH1_OUT_IMAGE_WIDTH Bits 26–16	**CH1 Out Image Width**—These bits sets the output width in number of pixels to display. See 28.4.3 for more details.	8 bpp—this value should be a multiple of 4. 16 bpp or YUV 4:2:2—this value should be a multiple of 2
Reserved Bits 15–11	Reserved—These bits are reserved and should read 0.	
CH1_OUT_IMAGE_HEIGHT Bits 10–0	**CH1 Out Image Height**—These bits set the output height in number of lines to display. See 28.4.3 for more details.	–

28.5.17 PrP Destination Channel-2 Output Image Size Register

PRP_CH2_OUT_IMAGE_SIZE — PrP Destination CH2 Output Image Size Register 0x10026440

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME						CH2_OUT_IMAGE_WIDTH										
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
	0x0140															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME						CH2_OUT_IMAGE_HEIGHT										
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
	0x00F0															

Table 28-53. PrP Destination CH2 Output Image Size Register Description

Name	Description	Settings
Reserved Bits 31–27	Reserved—These bits are reserved and should read 0.	
CH2_OUT_IMAGE_WIDTH Bits 26–16	**Channel-2 Output Image Width**—These bits sets the output width in number of pixels to display.	YUV 422, YUV 444 = this value should be a multiple of 2. YUV 4:2:0 = this value should be a multiple of 8
Reserved Bits 15–11	Reserved—These bits are reserved and should read 0.	
CH2_OUT_IMAGE_HEIGHT[1] Bits 10–0	**Channel 2 Output Image Height**—These bits set the Channel-2 output image height in number of lines in display. In YUV 4:2:0 mode, number of Cr and Cb lines are CH2_OUT_IMAGE_HEIGHT/2. See Section 28.4.3 on page -29 for more details.	In YUV 4:2:0 mode, this value should be in multiples of 2.

1 CH2_OUT_IMAGE_WIDTH is not required. PrP uses PICTURE_X_SIZE/MRR and truncates the output to multiples of 8 pixels for YUV 4:2:0 output and to multiples of 2 pixels for YUV 4:2:2 output.

28.5.17.1 PrP Source Line Stride Register

PRP_SRC_LINE_STRIDE — PrP Source Line Stride Register — 0x10026444

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME				CSI_LINE_SKIP												
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME				SOURCE_LINE_STRIDE												
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-54. PrP Source Line Stride Register Description

Name	Description	Settings
Reserved Bits 31–29	Reserved—These bits are reserved and should read 0	
CSI_LINE_SKIP Bits 28–16	**CSI Line Skip**—These bits set the number lines to skip from start of a frame from CSI. Used only when WINEN=1 and CSIEN=1.	
Reserved Bits 15–13	Reserved—These bits are reserved and should read 0.	
SOUCRE_LINE_STRIDE Bits 12–0	**Source Line Stride**—When CSIEN = 0, These bits set the line stride for the source frame in bytes. In YUV 4:2:0 mode, source line stride for ‘Cb’ and ‘Cr’ are SOURCE_LINE_STRIDE/2. When CSIEN = 1, then these bits sets number of pixels to skip from start of a line. It should be multiple of 2.	YUV 4:2:0 = value should be a multiple of 8. YUV 4:2:2 or RGB = value should be a multiple of 4.

28.5.17.2 PrP CSC Coefficient 012

PrP_CSC_COEF_012 — PrP CSC Coefficient 012 — 0x10026448

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME				C0										C1		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	C1								C2							
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0

Table 28-55. For YUV to RGB Description

Name	Description	Settings
Reserved Bits 31–29	Reserved—These bits are reserved and should read 0.	
C0 Bits 28–21	**Coefficient 0**—All 8 bits are significant.	Range from 0 to 1.9921875 in steps of (1/128)
Reserved Bits 20–19	Reserved—These bits are reserved and should read 0.	
C1 Bits 18–11	**Coefficient 1**—All 8 bits are significant.	Range from 0 to 1.9921875 in steps of (1/128)
Reserved Bits 10–8	Reserved—These bits are reserved and should read 0.	
C2 Bits 7–0	**Coefficient 2**—All 8 bits are significant.	Range from 0 to 1.9921875 in steps of (1/128)

Table 28-56. For RGB to YUV Description

Name	Description	Settings
Reserved Bits 31–29	Reserved—These bits are reserved and should read 0.	
C0 Bits 28–21	**Coefficient 0**—Only C0[6:0], 7 bits are significant.	Range from 0 to 0.49603750 in steps of (1/256)
Reserved Bits 20–19	Reserved—These bits are reserved and should read 0.	
C1 Bits 18–11	**Coefficient 1**—Only C1[6:0], 7 bits are significant.	Range from 0 to 0.9921875 in steps of (1/128)
Reserved Bits 10–8	Reserved—These bits are reserved and should read 0.	
C2 Bits 7–0	**Coefficient 2**—Only C2[6:0], 7 bits are significant.	Range from 0 to 0.248046875 in steps of (1/512)

28.5.17.3 PrP CSC Coefficient 345

PrP_CSC_COEF_345 — PrP CSC coefficient 345 — 0x1002644C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME				C3									C4			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	1	0	1	1	0	1	1	0	0	1	1	1

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	C4									C5						
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0

Table 28-57. For YUV to RGB Description

Name	Description	Settings
Reserved Bits 31–29	Reserved—These bits are reserved and should read 0.	
C3 Bits 28–21	**Coefficient 3**—All 8 bits are significant.	Range from 0 to 1.9921875 in steps of (1/128)
Reserved Bit 20	Reserved—This bit is reserved and should read 0.	
C4 Bits 19–11	**Coefficient 4**—All 8 bits are significant.	Range from 0 to 3.9921875 in steps of (1/128)
Reserved Bits 10–7	Reserved—These bits are reserved and should read 0.	
C5 Bits 6–0	**Coefficient 5**—This coefficient is unused in YUV to RGB conversion.	

Table 28-58. For RGB to YUV Description

Name	Description	Settings
Reserved Bits 31–29	Reserved—These bits are reserved and should read 0.	
C3 Bits 28–21	**Coefficient 3**—Only C3[6:0], 7 bits are significant.	Range from 0.0 to 0.248046875 in steps of (1/512)
Reserved Bit 20	Reserved—This bit is reserved and should read 0.	
C4 Bits 19–11	**Coefficient 4**—Only C4[6:0], 7 bits are significant.	Range from 0 to 0.49603750 in steps of (1/256)
Reserved Bits 10–7	Reserved—These bits are reserved and should read 0.	
C5 Bits 6–0	**Coefficient 5**—Only C5[6:0], 7 bits are significant.	Range from 0 to 0.9921875in steps of (1/128)

28.5.17.4 PrP CSC Coefficient 678

PrP_CSC_COEF_678 — PrP CSC Coefficient 678 — 0x10026450

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	X0				C6										C7	
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	C7									C8						
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28-59. For RGB to YUV Description

Name	Description	Settings
X0 Bit 31	**X0**—Luminance offset	1 = X0 is equal to 16 in CSC 0 = X0 is zero in CSC
Reserved Bits 30–28	Reserved—These bits are reserved and should read 0.	
C6 Bits 27–21	**Coefficient 6**—All 7 bits are significant.	Range from 0.0 to 0.9921875 in steps of (1/128)
Reserved Bits 20–18	Reserved—These bits are reserved and should read 0.	
C7 Bits 17–11	**Coefficient 7**—All 7 bits are significant.	Range from 0 to 0.49603750 in steps of (1/256)
Reserved Bits 10–7	Reserved—These bits are reserved and should read 0.	
C8 Bits 6–0	**Coefficient 8**—All 7 bits are significant.	Range from 0 to 0.248046875 in steps of (1/512)

Table 28-60. CSC Equations

Conversion	CSC Equation
YCbCr to RGB	R = C0*(Y – X0) + C1*(Cr-128) G = C0*(Y – X0) – C2*(Cb-128) – C3*(Cr-128) B = C0*(Y – X0) + C4*(Cb-128)
YUV to RGB	R = C0*(Y – X0) + C1*(U-128) G = C0*(Y – X0) – C2*(U-128) – C3*(V-128) B = C0*(Y – X0) + C4*(U-128)
RGB to YUV	Y = C0 × R + C1 × G + C2 × B + X0 U = -C3 × R - C4 × G + C5 × B + 128 V = C6 × R - C7 × G - C8 × B + 128

28.5.17.5 PrP Channel-1 Horizontal Resize Coefficient-1

PrP_CH1_RZ_HORI_COEF1 — PrP Channel-1 Horizontal Resize Coefficient 1 — 0x10026454

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME		HC9			HC8			HC7			HC6			HC5		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME		HC4			HC3			HC2			HC1			HC0		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

This register selects Channel-1 Horizontal resize coefficient values. The coefficient is 3-bits in width which represent value from 0 to 8. If HCx = 7 then it will treat as ‘8’.

- For bi-linear mode (w0 × a+ w1 × b)/8. Coefficient w0 is programmed.
- For m:n resize ratio number of coefficients will be “m”.

Table 28-61. PrP Channel-1 Horizontal Resize Coefficient-1 Register Description

Name	Description	Settings
Reserved Bit31	Reserved—This bit is reserved and should read 0.	
HC9 Bits 30–28	**Horizontal Resize Coefficient 9**	Valid values are 0–6 and 8. To set a value of 8, use 3’b111 instead.
HC8 Bits 27–25	**Horizontal Resize Coefficient 8**	Valid values are 0–6 and 8. To set a value of 8, use 3’b111 instead.
HC7 Bits 24–22	**Horizontal Resize Coefficient 7**	Valid values are 0–6 and 8. To set a value of 8, use 3’b111 instead.
HC6 Bits 21–19	**Horizontal Resize Coefficient 6**	Valid values are 0–6 and 8. To set a value of 8, use 3’b111 instead.
HC5 Bits 18–16	**Horizontal Resize Coefficient 5**	Valid values are 0–6 and 8. To set a value of 8, use 3’b111 instead.
Reserved Bit15	Reserved—This bit is reserved and should read 0.	
HC4 Bits 14–12	**Horizontal Resize Coefficient 4**	Valid values are 0–6 and 8. To set a value of 8, use 3’b111 instead.
HC3 Bits 11–9	**Horizontal Resize Coefficient 3**	Valid values are 0–6 and 8. To set a value of 8, use 3’b111 instead.
HC2 Bits 8–6	**Horizontal Resize Coefficient 2**	Valid values are 0–6 and 8. To set a value of 8, use 3’b111 instead.
HC1 Bits 5–3	**Horizontal Resize Coefficient 1**	Valid values are 0–6 and 8. To set a value of 8, use 3’b111 instead.
HC0 Bits 2–0	**Horizontal Resize Coefficient 0**	Valid values are 0–6 and 8. To set a value of 8, use 3’b111 instead.

28.5.17.6 PrP Channel-1 Horizontal Resize Coefficient-2

PrP_CH1_RZ_HORI_COEF2 — PrP Channel1 Horizontal Resize Coefficient 2 — 0x10026458

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	RSV	HC19			HC18			HC17			HC16			HC15		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	RSV	HC14			HC13			HC12			HC11			HC10		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register selects Channel-1 Horizontal resize coefficient values. The coefficient is 3-bits in width which represent value from 0 to 8. If HCx = 7 then it will treat as 8.

- For bi-linear mode (w0 × a+ w1 × b)/8. Coefficient w0 is programmed.
- For m:n resize ratio number of coefficients will be m.

Table 28-62. PrP Channel-1 Horizontal Resize Coefficient-2

Name	Description	Settings
Reserved Bit 31	Reserved—This bit is reserved and should read 0.	
HC10 Bits 30–28	**Horizontal Resize Coefficient 10**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC11 Bits 27–25	**Horizontal Resize Coefficient 11**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC12 Bits 24–22	**Horizontal Resize Coefficient 12**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC13 Bits 21–19	**Horizontal Resize Coefficient 13**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC14 Bits 18–16	**Horizontal Resize Coefficient 14**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
Reserved Bit15	Reserved—This bit is reserved and should read 0.	
HC15 Bits 14–12	**Horizontal Resize Coefficient 15**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC16 Bits 11–9	**Horizontal Resize Coefficient 16**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC17 Bits 8–6	**Horizontal Resize Coefficient 17**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC18 Bits 5–3	**Horizontal Resize Coefficient 18**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC19 Bits 2–0	**Horizontal Resize Coefficient 19**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.

28.5.17.7 PrP Channel-1 Horizontal Resize Valid

PrP_CH1_RZ_HORI_VALID — PrP Channel-1 Resize Horizontal Valid — 0x1002645C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	AVG_BIL			HORI_TBL_LEN									HOV			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	HOV															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 28-63. PrP Channel-1 Horizontal Resize Valid

Name	Description	Settings
AVG_BIL Bit 31	**Averaging or Bilinear**—Mode select.	0 = Averaging 1 = Bi-linear
Reserved Bits 30–29	Reserved—These bits are reserved and should read 0.	
HORI_TBL_LEN Bits 28–24	Horizontal Resize Table Length—Selects horizontal resize table length.	For M:N resize ratio for both bilinear and averaging mode table length should be set to “M”. Range from 1 to 20.
Reserved Bits 23–20	Reserved—These bits are reserved and should read 0.	
HOV Bits 19–0	**Horizontal Output Valid**—Bit vector that selects when to output pixel. Skips output pixels when averaging and input pixels when bi-linear.	0 = Pixel is not output 1 = Pixel is output

28.5.17.8 PrP Channel-1 Vertical Resize Coefficient-1

PrP_CH1_RZ_VERT_COEF1 — PrP Channel 1 Vertical Resize Coefficient 1 — 0x10026460

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME		VC9			VC8			VC7			VC6			VC5		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME		VC4			VC3			VC2			VC1			VC0		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

This register selects Channel-1 vertical resize coefficient values. The coefficient is 3-bits in width which represent values from 0 to 8. If Vex = 7 then it will treat it as 8.

- For I-linear mode (w0 × a+ w1 × b)/8. Coefficient w0 is programmed.
- For man resize ratio number of coefficients will be m.

Table 28-64. PrP Channel-1 Vertical Resize Coefficient-1

Name	Description	Settings
Reserved Bit31	Reserved—This bit is reserved and should read 0.	
VC9 Bits 30–28	**Vertical Resize Coefficient 9**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC8 Bits 27–25	**Vertical Resize Coefficient 8**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC7 Bits 24–22	**Vertical Resize Coefficient 7**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC6 Bits 21–19	**Vertical Resize Coefficient 6**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC5 Bits 18–16	**Vertical Resize Coefficient 5**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
Reserved Bit15	Reserved—This bit is reserved and should read 0	
VC4 Bits 14–12	**Vertical Resize Coefficient 4**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC3 Bits 11–9	**Vertical Resize Coefficient 3**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC2 Bits 8–6	**Vertical Resize Coefficient 2**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC1 Bits 5–3	**Vertical Resize Coefficient 1**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC0 Bits 2–0	**Vertical Resize Coefficient 0**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.

28.5.17.9 PrP Channel-1 Vertical Resize Coefficient-2

PrP_CH1_RZ_VERT_COEF2 — PrP Channel-1 Vertical Resize Coefficient 2 — 0x10026464

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME		VC19			VC18			VC17			VC16			VC15		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME		VC14			VC13			VC12			VC11			VC10		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register selects Channel-1 Vertical resize coefficient values. The coefficient is 3-bits in width which represent values from 0 to 8. If Vex = 7 then it is treated as 8.

- For I-linear mode (w0 × a+ w1 × b)/8. Coefficient w0 is programmed.
- For man resize ratio number of coefficients will be m.

Table 28-65. PrP Channel-1 Vertical Resize Coefficient-2

Name	Description	Settings
Reserved Bit31	Reserved—This bit is reserved and should read 0.	
VC10 Bits 30–28	**Vertical Resize Coefficient 10**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC11 Bits 27–25	**Vertical Resize Coefficient 11**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC12 Bits 24–22	**Vertical Resize Coefficient 12**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC13 Bits 21–19	**Vertical Resize Coefficient 13**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC14 Bits 18–16	**Vertical Resize Coefficient 14**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
Reserved Bit15	Reserved—This bit is reserved and should read 0.	
VC15 Bits 14–12	**Vertical Resize Coefficient 15**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC16 Bits 11–9	**Vertical Resize Coefficient 16**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC17 Bits 8–6	**Vertical Resize Coefficient 17**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC18 Bits 5–3	**Vertical Resize Coefficient 18**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC19 Bits 2–0	**Vertical Resize Coefficient 19**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.

28.5.17.10 PrP Channel-1 Vertical Resize Valid

PrP_CH1_RZ_VERT_VALID — PrP Channel1 Vertical Resize Valid — 0x10026468

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	AVG_BIL			VERT_TBL_LEN									VOV			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	VOV															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 28-66. PrP Channel-1 Vertical Resize Valid

Name	Description	Settings
AVG_BIL Bit 31	**Averaging or Bilinear**—Mode select.	0 = Averaging 1 = Bi-linear
Reserved Bits 30–29	Reserved—These bits are reserved and should read 0.	
VERT_TBL_LEN Bits 28–24	**Vertical Resize Table Length**—Selects vertical resize table length.	For M:N resize ratio for both bilinear and averaging mode table length should be set to “M”. Range from 1 to 20.
Reserved Bits 23–20	Reserved—These bits are reserved and should read 0.	
VOV Bits 19–0	**Vertical Output Valid**—Bit vector that selects when to output pixel. Skips output pixels when averaging and input pixels when bi-linear.	0 = Pixel is not output 1 = Pixel is output

28.5.17.11 PrP Channel-2 Horizontal Resize Coefficient-1

PrP_CH2_RZ_HORI_COEF1 PrP Channel-2 Horizontal Resize Coefficient 1 0x1002646C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME		HC9			HC8			HC7			HC6			HC5		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME		HC4			HC3			HC2			HC1			HC0		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

This register selects Channel-2 Horizontal resize coefficient values. The coefficient is 3-bits in width which represent value from 0 to 8. If HCx = 7 then it will treat as 8.

- For bi-linear mode (w0 × a+ w1 × b)/8. Coefficient w0 is programmed.
- For m:n resize ratio number of coefficients will be m.

Table 28-67. PrP Channel-2 Horizontal Resize Coefficient-1

Name	Description	Settings
Reserved Bit 31	Reserved—This bit is reserved and should read 0.	
HC9 Bits 30–28	**Horizontal Resize Coefficient 9**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC8 Bits 27–25	**Horizontal Resize Coefficient 8**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC7 Bits 24–22	**Horizontal Resize Coefficient 7**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC6 Bits 21–19	**Horizontal Resize Coefficient 6**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC5 Bits 18–16	**Horizontal Resize Coefficient 5**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
Reserved Bit15	Reserved—This bit is reserved and should read 0.	
HC4 Bits 14–12	**Horizontal Resize Coefficient 4**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC3 Bits 11–9	**Horizontal Resize Coefficient 3**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC2 Bits 8–6	**Horizontal Resize Coefficient 2**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC1 Bits 5–3	**Horizontal Resize Coefficient 1**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC0 Bits 2–0	**Horizontal Resize Coefficient 0**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.

28.5.17.12 PrP Channel-2 Horizontal Resize Coefficient-2

PrP_CH2_RZ_HORI_COEF2 — PrP Channel2 Horizontal Resize Coefficient 2 — 0x10026470

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	RSV	HC19			HC18			HC17			HC16			HC15		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	RSV	HC14			HC13			HC12			HC11			HC10		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register selects Channel-2 Horizontal resize coefficient values. The coefficient is 3-bits in width which represent value from 0 to 8. If HCx = 7 then it will be treated as 8.

- For bi-linear mode (w0 × a+ w1 × b)/8. Coefficient w0 is programmed.
- For m:n resize ratio number of coefficients will be m.

Table 28-68. PrP Channel-2 Horizontal Resize Coefficient-2

Name	Description	Settings
Reserved Bit31	Reserved—This bit is reserved and should read 0.	
HC10 Bits 30–28	**Horizontal Resize Coefficient 10**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC11 Bits 27–25	**Horizontal Resize Coefficient 11**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC12 Bits 24–22	**Horizontal Resize Coefficient 12**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC13 Bits 21–19	**Horizontal Resize Coefficient 13**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC14 Bits 18–16	**Horizontal Resize Coefficient 14**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
Reserved Bit15	Reserved—This bit is reserved and should read 0.	
HC15 Bits 14–12	**Horizontal Resize Coefficient 15**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC16 Bits 11–9	**Horizontal Resize Coefficient 16**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC17 Bits 8–6	**Horizontal Resize Coefficient 17**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC18 Bits 5–3	**Horizontal Resize Coefficient 18**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
HC19 Bits 2–0	**Horizontal Resize Coefficient 19**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.

28.5.17.13 PrP Channel-2 Horizontal Resize Valid

PrP_CH2_RZ_HORI_VALID — PrP Channel-2 Resize Horizontal Valid — 0x10026474

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	AVG_BIL			HORI_TBL_LEN									HOV			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	HOV															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 28-69. PrP Channel-2 Horizontal Resize Valid

Name	Description	Settings
AVG_BIL Bit 31	**Averaging or Bilinear**—Mode select	0 = Averaging 1 = Bi-linear
Reserved Bits 30–29	Reserved—These bits are reserved and should read 0.	
HORI_TBL_LEN Bits 28–24	**Horizontal Resize Table Length**—Selects horizontal resize table length	For M:N resize ratio for both bilinear and averaging mode table length should be set to “M”. Range from 1 to 20.
Reserved Bits 23–20	Reserved—These bits are reserved and should read 0.	
HOV Bits 19–0	**Horizontal Output Valid**—Bit vector that selects when to output pixel. Skips output pixels when averaging and input pixels when bi-linear.	0 = Pixel is not output 1 = Pixel is output

28.5.17.14 PrP Channel-2 Vertical Resize Coefficient-1

PrP_CH2_RZ_VERT_COEF1 PrP Channel2 Vertical Resize Coefficient 1 0x10026478

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME		VC9			VC8			VC7			VC6			VC5		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME		VC4			VC3			VC2			VC1			VC0		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

This register selects Channel-2 vertical resize coefficient values. The coefficient is 3-bits in width which represent values from 0 to 8. If VCx = 7 then it will treat it as 8.

- For bi-linear mode (w0 × a+ w1 × b)/8. Coefficient w0 is programmed.
- For m:n resize ratio number of coefficients will be m.

Table 28-70. PrP Channel-2 Vertical Resize Coefficient-1

Name	Description	Settings
Reserved Bit 31	Reserved—This bit is reserved and should read 0.	
VC9 Bits 30–28	**Vertical Resize Coefficient 9**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC8 Bits 27–25	**Vertical Resize Coefficient 8**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC7 Bits 24–22	**Vertical Resize Coefficient 7**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC6 Bits 21–19	**Vertical Resize Coefficient 6**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC5 Bits 18–16	**Vertical Resize Coefficient 5**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
Reserved Bit15	Reserved—This bit is reserved and should read 0.	
VC4 Bits 14–12	**Vertical Resize Coefficient 4**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC3 Bits 11–9	**Vertical Resize Coefficient 3**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC2 Bits 8–6	**Vertical Resize Coefficient 2**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC1 Bits 5–3	**Vertical Resize Coefficient 1**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC0 Bits 2–0	**Vertical Resize Coefficient 0**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.

28.5.17.15 PrP Channel-2 Vertical Resize Coefficient-2

PrP_CH2_RZ_VERT_COEF2 — PrP Channel-2 Vertical Resize Coefficient 2 — 0x1002647C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME		VC19			VC18			VC17			VC16			VC15		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME		VC14			VC13			VC12			VC11			VC10		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register selects Channel-2 Vertical resize coefficient values. The coefficient is 3-bits in width which represent values from 0 to 8. If VCx = 7 then it is treated as 8.

- For bi-linear mode (w0 × a+ w1*b)/8. Coefficient w0 is programmed.
- For m:n resize ratio number of coefficients will be m.

Table 28-71. PrP Channel-2 Vertical Resize Coefficient-2

Name	Description	Settings
Reserved Bit 31	Reserved—This bit is reserved and should read 0.	
VC10 Bits 30–28	**Vertical Resize Coefficient 10**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC11 Bits 27–25	**Vertical Resize Coefficient 11**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC12 Bits 24–22	**Vertical Resize Coefficient 12**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC13 Bits 21–19	**Vertical Resize Coefficient 13**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC14 Bits 18–16	**Vertical Resize Coefficient 14**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
Reserved Bit15	Reserved—This bit is reserved and should read 0.	
VC15 Bits 14–12	**Vertical Resize Coefficient 15**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC16 Bits 11–9	**Vertical Resize Coefficient 16**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC17 Bits 8–6	**Vertical Resize Coefficient 17**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC18 Bits 5–3	**Vertical Resize Coefficient 18**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.
VC19 Bits 2–0	**Vertical Resize Coefficient 19**	Valid values are 0–6 and 8. To set a value of 8, use 3'b111 instead.

28.5.17.16 PrP Channel-2 Vertical Resize Valid

PrP_CH2_RZ_VERT_VALID — PrP Channel2 Vertical Resize Valid — 0x10026480

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NAME	AVG_BIL			VERT_TBL_LEN									VOV			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME	VOV															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 28-72. PrP Channel-2 Vertical Resize Valid

Name	Description	Settings
AVG_BIL Bit 31	**Averaging or Bilinear**—Mode select.	0 = Averaging 1 = Bi-linear
Reserved Bits 30–29	Reserved—These bits are reserved and should read 0.	
VERT_TBL_LEN Bits 28–24	**Vertical Resize Table Length**—Selects vertical resize table length.	For M:N resize ratio for both bilinear and averaging mode table length should be set to “M”. Range from 1 to 20.
Reserved Bits 23–20	Reserved—These bits are reserved and should read 0.	
VOV Bits 19–0	**Vertical Output Valid**—Bit vector that selects when to output pixel. Skips output pixels when averaging and input pixels when bi-linear.	0 = Pixel is not output 1 = Pixel is output

Chapter 29
MultimediaCard/Secure Digital Host Controller (MMC/SDHC)

The MultiMediaCard (MMC), is a universal low cost data storage and communication media that is designed to cover a wide area of applications such as electronic toys, organizers, PDAs, smart phones, and so on. The MMC communication is based on an advanced 7 pin serial bus designed to operate in a low voltage range, at medium speed (20 Mbps).

The Secure Digital (SD) card, is an evolution of MMC, with an additional 2 pins and the same form factor. It is specifically designed to meet the security, capacity, performance, and environmental requirements inherent in newly emerging audio and video consumer electronic devices. The physical form factor, pin assignment, and data transfer protocol are forward compatible with the MultiMediaCard with some additions. An SD card can be categorized as SD Memory or SD I/O card, commonly known as SDIO. A memory card invokes a copyright protection mechanism that complies with the security of the SDMI standard and is faster and capable of higher memory capacity. The SDIO card provides high-speed data I/O with low-power consumption for mobile electronic devices.

The MiniSD and TransFlash (T-Flash) memory cards are newcomers in the mobile market that are using the same protocol of SD memory cards with a difference of a smaller form factor. The MultimediaCard/Secure Digital Host Controller integrates MMC support with SD memory, MiniSD memory, TransFlash memory and SDIO functions.

Features of the MultimediaCard/Secure Digital Host Controller include the following:

- Fully compatible with the *MMC System Specification Version 3.0*
- Fully compatible with the *SD Memory Card Specification 1.0*, and *SD I/O Specification 1.0* with 1 or 4 channel(s)
- 20–80 Mbps maximum data rate
- Built-in programmable frequency counter for SDHC bus
- Maskable hardware interrupt for SDIO interrupt, internal status and FIFO status
- 32 × 16-bit built-in FIFO
- Plug and play support
- Password protection of cards
- Single or multi block access to the card including erase operation
- Multi-SD function support including multiple I/O and combined I/O and memory
- Up to 7 I/O functions plus one memory supported on single SDIO card
- IRQ supported enable card to interrupt host
- Supports SDIO read wait, interrupt detection during 1/4-bit access

29.1 Host Controller Block Diagram

Figure 29-1 provides a high-level block diagram to show the major components of the MultimediaCard/Secure Digital Host controller (MMC/SD). Figure 29-2 shows the system interconnection between the MMC/SD and the other modules in the i.MX21 processor.

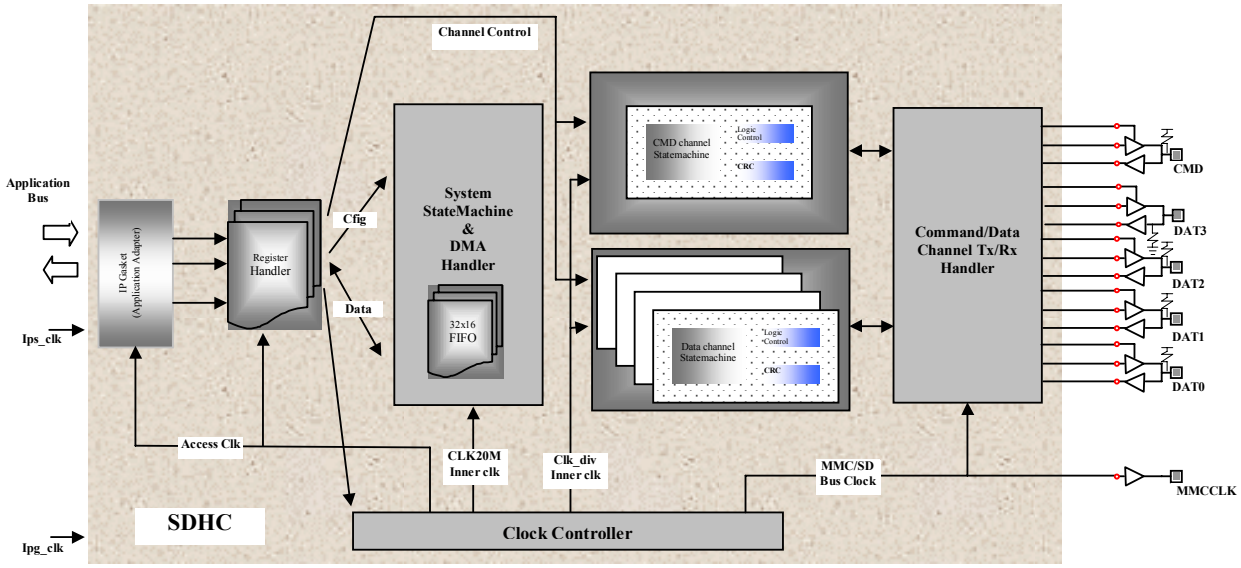


Figure 29-1. Secure Digital Host Controller Block Diagram

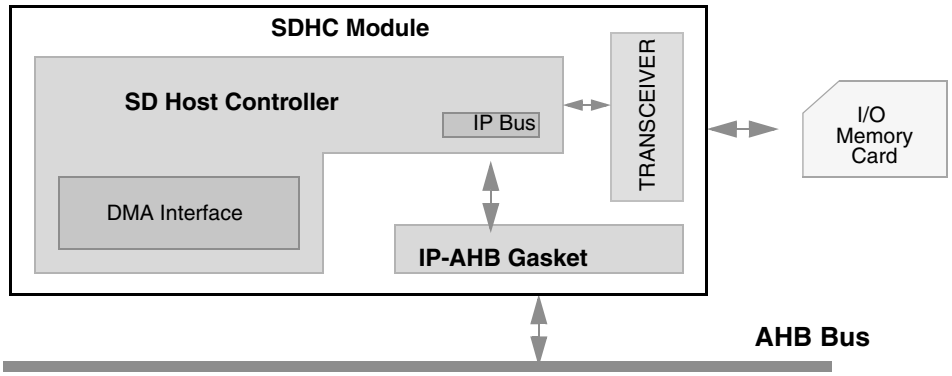


Figure 29-2. System Interconnection with Secure Digital Host Controller

29.2 Host Controller Signal I/O Description

Table 29-1 provides a description of the I/O signals of the MMC/SD host controller.

Table 29-1. MMC/SDHC Signal Description

Signal Name	Signal Type	Description
MMCCLK (Card Bus)	output	Bus clock for external memory card

29.3 Programming Model

The MMC/SD host controller is controlled by a set of registers that the application configures before every operation on the MMC/SD bus. The Table 29-2 describes the registers addresses and reset states. All registers listed are word accessible only—that is, byte or half word accesses to the registers are not supported, even though only the lower 16-bits of the registers are used.

Table 29-2. MMC/SD Host Controller Register Summary

Address [11:0]	Name	Default Value
000	STR_STP_CLK	0000h
004	STATUS (ReadOnly)	0000h
008	CLK_RATE	0008h
00C	CMD_DAT_CONT	0000h
010	RESPONSE_TO	0040h
014	READ_TO	FFFFh
018	BLK_LEN	0000h
01C	NOB	0000h
020	REV_NO	0400h
024	INT_CNTL	0000h
028	CMD	0000h
02C	ARGH	0000h
030	ARGL	0000h
034	RES_FIFO (ReadOnly)	0000h
038	BUFFER_ACCESS	XXXXh

29.3.1 MMC/SD Clock Control Register

The MMC/SD Clock Control register (STR_STP_CLK) allows the user to reset the system and control the MMC/SD clock.

NOTE

The clock to the reset logic in SDHC is off in normal operation to reduce power consumption. When there is one access to this register, there is a clock pulse to the reset logic. For the whole reset period, at least 8 clock pulses are required to finish the reset cycle. Therefore, to reset the SDHC, write to this register with the value 0x0008, then 0x0009, and then 0x001, eight times. The STOP_CLK bit will be 1 after this reset procedure and the card bus MMCCLK is stopped.

STR_STP_CLK — Clock Control Register — Addr 0x10013000 0x10014000

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													MMCSD_RESET		START_CLK	STOP_CLK
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	w	r	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 29-3. MMC/SD Clock Control Register Description

Name	Description	Settings
Reserved Bits 31–4	Reserved—These bits are reserved and should read 0.	
MMCSD_RESET Bit 3	**MMCSD Reset**—Resets the MMC/SD host controller.	0 = No effect 1 = Reset the MMC/SD host controller
Reserved Bit 2	Reserved—This bit is reserved and should read 0.	
START_CLK Bit 1	**Start Clock**—Starts the MMC/SD clock. The MMCSD_ENABLE bit must be set for START_CLK to have any meaning. When START_CLK is changed during the MMC/SD transmission period, the status of the operation determines when the clock is started.Otherwise, the clock begins immediately. Setting a value of 11 on Bits 1:0 is prohibited. **Note:** A transmission period is defined as the time from a card data-access-related command is submitted to the end of the data access operation.	0 = MMC/SD clock inactive 1 = MMC/SD clock active
STOP_CLK Bit 0	**Stop Clock**—Stops the MMC/SD clock. The MMCSD_ENABLE bit must be set for STOP_CLK to have any meaning. When STOP_CLK is changed during the MMC/SD transmission period, the status of the operation determines when the clock is stopped. Otherwise, the clock halts immediately. Setting a value of 11 on Bits 1:0 is prohibited. **Note:** A transmission period is defined as the time from a card data-access-related command is submitted to the end of the data access operation.	0 = MMC/SD clock active 1 = MMC/SD clock stopped

NOTE

There is no module enable bit in SDHC. The module enable bit for SDHC in i.MX21 is in the CRM PCCR0 register.

29.3.2 MMC/SD Status Register

The read-only MMC/SD Status register provides the programmer with information about the status of MMC/SD operations, application FIFO status, and error conditions.

STATUS — Status Register — Addr 0x10013004 0x10014004

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CARD_ PRESENCE	SDIO_ INT_ ACTIVE	END_ CMD_ RESP	WRIT E_OP _ DONE	READ _OP_ DONE	WR_CRC_ ERROR_CO DE		CARD_BU S_CLK_R UN	APPL _BUF F_FF	APPL _BUF F_FE	RESP _CRC _ERR		CRC_ READ _ERR	CRC_ WRIT E_ER R	TIME_ OUT_ RESP	TIME_ OUT_ READ
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 29-4. MMC/SD Status Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
CARD_PRESENCE Bit 15	**Card Presence**—Detects whether an MMC/SD card is present based on SD_DAT [3]. When this bit is set, SDHC generates an interrupt request when the card detection interrupt is enabled in INT_CNTL register (INT_CNTL[15]).	0 = No cards are present 1 = At least 1 card is present
SDIO_INT_ACTIVE Bit 14	**SDIO Interrupt Active**—Indicates whether an interrupt is detected at the SDIO card. A separate acknowledge command to the card is required to clear this interrupt. Write 1 to clear.	0 = No interrupt detected 1 = Interrupt detected via SDIO card bus. Write 1 to clear.
END_CMD_RESP Bit 13	**End Command Response**—Indicates that the command was successfully transmitted to the card. A predefined response package is received or a response time-out after each successful command operation. Write 1 or stop MMCCLK to clear. When this bit is set, SDHC generates an interrupt request when the End_CMD_RESP interrupt is enabled in the INT_CNTL register. **Note:** When this bit is set, the user must verify whether the command send and response receive operation completes without error. For verification use RESP_CRC_ERR (Status[5]) and TIME_OUT_RESP (STATUS[1]).	0 = Command not successful, incomplete, or not applicable (no response) 1 = Command transmitted successfully (response received). Write 1 to clear.

Table 29-4. MMC/SD Status Register Description (continued)

Name	Description	Settings
WRITE_OP_DONE Bit 12	**Write Operation Done**—Indicates when an access operation is complete. The Flash card needs extra idle time to write and requires the MMC/SD host controller to wait until the card buffered data to the inner Flash memory. The MMC/SD host controller automatically detects the status. WRITE_OP_DONE determines the end of the write operation. When this bit is set, SDHC generates an interrupt request if the WRITE_OP_DONE interrupt is enabled in the INT_CNTL register. Write 1 or stop MMCCLK to clear. **Note:** When this bit is set, the user must verify whether the write operation completes without error. For verification use WRITE_CRC_ERR (STATUS[2]).	0 = Write operation in progress or incomplete. 1 = Write operation complete. Write 1 to clear.
READ_OP_DONE Bit 11	**Read Operation Done**—Indicates that a read transfer is complete. READ_OP_DONE determines the end of the read operation. When this bit is set, SDHC generates an interrupt request if the READ_OP_DONE interrupt is enabled. Write 1 or stop MMCCLK to clear. **Note:** When this bit is set, the user must verify whether the read operation completes without error. For verification use READ_CRC_ERR (Status[3]) and TIME_OUT_READ (STATUS[0]).	0 = Data transfer in progress or incomplete 1 = Data transfer complete. Write 1 to clear.
WR_CRC_ERROR_CODE Bits 10–9	**Write CRC Error Code**—CRC code feedback from card in write operation. After receiving a block of data, the card checks the CRC bit and sends feedback upon CRC status. These two bits reflect the CRC status of the recent written data. When the card sends a negative CRC status, the data is not written to the card. Write 1 or stop MMCCLK to clear.	00 = No transmission error, CRC status is 010 (positive) 01 = Transmission error, CRC status is 110 (negative) 10 = No CRC response 11 = Reserved
CARD_BUS_CLK_RUN Bit 8	**MMC/SD Card Clock Running**—Indicates whether the clock is running. The clock rate setting and system configuration can be modified when the clock is turned off by setting the STOP_CLK bit of the STR_STP_CLK register.	0 = MMC/SD clock is not running 1 = MMC/SD clock is running
APPL_BUFF_FF Bit 7	**Application Buffer FIFO Full**—Indicates the status of the 32 × 16-bit inner data FIFO. Usually used in the card read operation.	0 = Buffer is not full 1 = Buffer is full
APPL_BUFF_FE Bit 6	**Application Buffer FIFO Empty**—Indicates the status of the 32 × 16-bit inner data FIFO. Usually used in the card write operation.	0 = Buffer is not empty 1 = Buffer is empty
RESP_CRC_ERR Bit 5	**Response CRC Error**—Indicates that a transmission error occurred on the SD_CMD line during the command and response transfer. This type of error usually results from the electrical environment. The user should identify the cause of the CRC error and re-try the transmission. Write 1 or stop MMCCLK to clear.	0 = No error 1 = Response CRC error occurred. Write 1 to clear.
Reserved Bit 4	Reserved—This bit is reserved and should read 0.	
CRC_READ_ERR Bit 3	**CRC Read Error**—Indicates that a transmission error occurred on the SD_DAT line during a card read. The user should re-try the transmission. Write 1 or stop MMCCLK to clear.	0 = No error 1 = CRC read error occurred. Write 1 to clear.

Table 29-4. MMC/SD Status Register Description (continued)

Name	Description	Settings
CRC_WRITE_ERR Bit 2	**CRC Write Error**—Indicates that a transmission error occurred on the SD_DAT line during a card write. See the WR_CRC_ERROR_CODE field for more information. Write 1 or stop MMCCLK to clear.	0 = No error 1 = CRC write error. Write 1 to clear.
TIME_OUT_RESP Bit 1	**Time Out Response Error**—Indicates that command response is not received in the time specified in the RES_TO register. This can be caused by: • An unsupported command received at the card(s) • Another MMC/SD_OP_COND command submitted after all cards have sent their voltage ranges and the power-up routine is finished • An identification command issued when all cards are in standby state • No card is on the bus Write 1 or stop MMCCLK to clear.	0 = No error 1 = time-out response occurred. Write 1 to clear.
TIME_OUT_READ Bit 0	**Time Out Read Data Error**—Indicates that the expected data from the card is not received in the time specified in the READ_TO register. Write 1 or stop MMCCLK to clear.	0 = No error 1 = Time out read data error occurred. Write 1 to clear.

29.3.3 MMC/SD Clock Rate Register

The MMC/SD Clock Rate register (CLK_RATE) defines the clock divider values for CLK_20M and CLK_DIV. These two signals clock the MMC/SD host controller.

There is one clock divider and one clock prescaler in SDHC to divide the high frequency input clock PERCLK2 to a low frequency clock which can be used by card. The input clock must first go through a 4-bit divider and then a 12-bit prescaler to generate a clock named CLK_20M. This clock is used internally by SDHC and generates the MMCCLK. The MMCCLK to the card has the same clock frequency as CLK_20M. CLK_20M is divided from CLK_DIV by the 12-bit prescaler, and CLK_DIV is divided from the input clock PERCLK2 by the 4-bit divider.

NOTE

When setting the clock rate register, the user must check the maximum clock frequency that the card supports. The user must ensure that the card does not work in an overclocked mode.

CLK_RATE — Clock Rate Register — 0x10013008, 0x10014008

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CLK_PRESCALER												CLK_DIVIDER			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	0x0008															

Table 29-5. MMC/SD Clock Rate Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
CLK_PRESCALER Bits 15–4	**Clock Prescaler**—Specifies the divider value to generate CLK_20M.	000h = CLK_20M is CLK_DIV 001h = CLK_20M is CLK_DIV / 2 002h = CLK_20M is CLK_DIV / 4 004h = CLK_20M is CLK_DIV / 8 008h = CLK_20M is CLK_DIV / 16 010h = CLK_20M is CLK_DIV / 32 020h = CLK_20M is CLK_DIV / 64 040h = CLK_20M is CLK_DIV / 128 080h = CLK_20M is CLK_DIV / 256 100h = CLK_20M is CLK_DIV / 512 200h = CLK_20M is CLK_DIV /1024 400h = CLK_20M is CLK_DIV /2048 800h = CLK_20M is CLK_DIV /4096 Other values: Reserved
CLK_DIVIDER Bits 3–0	**Clock Divider**—Specifies the divider value to generate CLK_DIV. The PERCLK2 feeds into the MMC/SD host controller from the clock controller.	0001 = CLK_DIV is PERCLK / 2 0010 = CLK_DIV is PERCLK / 3 0011 = CLK_DIV is PERCLK / 4 0100 = CLK_DIV is PERCLK / 5 0101 = CLK_DIV is PERCLK / 6 0110 = CLK_DIV is PERCLK / 7 0111 = CLK_DIV is PERCLK / 8 1000 = CLK_DIV is PERCLK / 9 1001 = CLK_DIV is PERCLK / 10 1010 = CLK_DIV is PERCLK / 11 1011 = CLK_DIV is PERCLK / 12 1100 = CLK_DIV is PERCLK / 13 1101 = CLK_DIV is PERCLK / 14 1110 = CLK_DIV is PERCLK / 15 1111 = CLK_DIV is PERCLK / 16 Others values: Reserved

29.3.4 MMC/SD Command and Data Control Register

The MMC/SD Command and Data Control register (CMD_DAT_CONT) allows the user to specify the format of data and response, and to control the ReadWait cycle. Most importantly, the write access to this register with bit[15] triggers SDHC to issue a command to the card. Therefore, the user must first configure the command and argument registers correctly before configuring this register to issue a command to the card.

CMD_DAT_CONT — Command and Data Control Register — Addr 0x1001300C 0x1001400C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMD_ RESUME			CMD_RESP_ LONG_OFF	STOP_ READ_ WAIT	START_ READ_ WAIT	BUS_ WIDTH		INIT			WRITE_ READ	DATA_ ENABLE	FORMAT_OF_ RESPONSE		
TYPE	r	r	r	rw	rw	rw	rw	rw	rw	r	r	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 29-6. MMC/SD Command and Data Control Register Description

Name	Description	Settings
Reserved Bits 31–13	Reserved—These bits are reserved and should read 0.	
CMD_RESUME Bit 15	**Command Resume**—When writing 0 to this bit, the write access triggers the SDHC to issue a command to the card. When this bit is set, SDHC will not issue a command to the card. This bit is used to end the READWAIT cycle and resume the data read operation. In this case, the host must restore the command and data control register for the read operation without issuing the read command to card again.	0= Triggers SDHC to issue command to card 1= Resume the previous command and does not send command to card
CMD_RESP_LONG_OFF Bit 12	**Command Response Long Off**—Allows bit clearance when status is read. Utilized in ReadWait cycle.	0 = Bit not cleared when read 1 = Allows bit clearance
STOP_READWAIT Bit 11	**Stop ReadWait**—Ends the ReadWait cycle.	0 = No effect 1 = Ends cycle
START_READWAIT Bit 10	**Start ReadWait**—Starts the ReadWait cycle.	0 = No effect 1 = Starts cycle

Table 29-6. MMC/SD Command and Data Control Register Description (continued)

Name	Description	Settings
BUS_WIDTH Bits 9–8	**Bus Width**—Specifies the width of the data bus.	00 = 1-bit 01 = Reserved 10 = 4-bit 11 = Reserved
INIT Bit 7	**Initialize**—Specifies whether the optional 80 clock cycle prefix (to initialize the card) will occur before every command. INIT enables/disables the 80 clock initialization time.	0 = Disable 80 clocks 1 = Enable 80 clocks
Reserved Bits 6–5	Reserved—These bits are reserved and should read 0.	
WRITE_READ Bit 4	**Write or Read**—Specifies whether the data transfer of the current command is a write or read operation.	0 = Read 1 =Write
DATA_ENABLE Bit 3	**Data Transfer**—Specifies whether the current command includes a data transfer.	0 = No data transfer included 1 = Data transfer included
FORMAT_OF_RESPONSE Bits 2–0	**Format of Response**—Sets the response format.	000 = No Response 001 = Format R1/R1b/R5/R6 (48-bit response with CRC7 bits) 010 = Format R2 (136-bit response, CSD/CID register read response) 011 = Format R3/R4 (48-bit response without CRC bits) Other values = Reserved

29.3.5 MMC/SD Response Time Out Register

The MMC/SD Response Time Out register (RES_TO) defines the time-out error for a received response.

RES_TO — Response Time Out Register — Addr 0x10013010 0x10014010

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									Response Time Out							
TYPE	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	0x0040															

Table 29-7. MMC/SD Response Time Out Register Description

Name	Description	Settings
Reserved Bits 31–8	Reserved—These bits are reserved and should read 0.	
RESPONSE TIME OUT Bits 7–0	**Response Time Out**—Specifies the number of clock counts between the command and when the MMC/SDHC turns on the time-out error for the received response. the clock counts unit is CLK_20M.	0x01 = 1clock counts … 0xFF = 255 clock counts

29.3.6 MMC/SD Read Time Out Register

The MMC/SD Read Time Out register (READ_TO) defines the time out error for received data.

READ_TO — Read Time Out Register — Addr 0x10013014 0x10014014

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA READ TIME OUT															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0xFFFF															

Table 29-8. MMC/SD Read Time Out Register Description

Name	Description
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.
DATA READ TIME OUT Bits 15–0	**Received Data Time Out**—Specifies the number of clocks between the command and when the MMC/SD host controller turns on the time-out error for the received data. The unit is CLK_20M / 256. A value of 0x2DB4 is recommended.

29.3.7 MMC/SD Block Length Register

The MMC/SD Block Length register (BLK_LEN) defines the number of bytes in a block. For SDIO, the block length must be less than the Maximum Block Size defined in the card's CCCR. For MMC/SD, the block length must be less than the maximum block size defined in the card's CSD register. The maximum block length supported by MMC/SD can be 1–2048 bytes.

NOTE

For multi-block data transfers, multiply the block length by the data buffer size for efficient buffer utilization. There are two options to transfer the data:

Option 1—Split the transaction. Use a single block command to transfer the remaining blocks of data.

Option 2—Add dummy data. Use dummy data in the last block to fill the block size as large as the buffer size.

The data buffer size is 64 bytes in 4-bit mode and 16 bytes in 1-bit mode.

BLK_LEN — Block Length Register — Addr 0x10013018 0x10014018

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							BLOCK LENGTH									
TYPE	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 29-9. MMC/SD Block Length Register Description

Name	Description	Settings
Reserved Bits 31–10	Reserved—These bits are reserved and should read 0.	
BLOCK LENGTH Bits 9–0	**Block Length**—Specifies the number of bytes in a block, and is normally set to 0x200 for MMC/SD data transactions. The User must confirm the value specified in the card's CSD or CCCR register to set the proper block length for the data transfer.	0x000 = 0 byte 0x001 = 1 bytes 0x002 = 2 bytes ... 0x7FE = 2046 bytes 0x7FF = 2047 bytes 0x800 = 2048 bytes Others values = Reserved

29.3.8 MMC/SD Number of Block Register

The MMC/SD Number of Block register (NOB) defines the number of blocks in a data transfer. The NOB register and the BLK_LEN register (see Section 29.3.7 on page -12) define the maximum data to be transferred in a single data transfer command.

Maximum data size to be transferred in bytes = Block Length x Number of Blocks

NOB — Number of Block Register — Addr 0x1001301C 0x1001401C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NOB															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 29-10. Number of Block Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
NOB Bits 15–0	**Block Length**—Specifies the number of blocks in a data transfer. One block is a possibility.	0x0000 = 0 block 0x0001 = 1 blocks … 0xFFFF = 65535 blocks

29.3.9 MMC/SD Revision Number Register

The read-only MMC/SD Revision Number register (REV_NO) is a read-only register that displays the revision number of the module.

REV_NO — Revision Number Register — Addr 0x10013020 0x10014020

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	REVISION NUMBER															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0400															

Table 29-11. MMC/SD Revision Number Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
REVISION NUMBER Bits 15–0	**Module Revision Number**—Specifies the revision number of the MMC/SD module.	Fixed at 0x0400

29.3.10 MMC/SD Interrupt Control Register

When certain conditions exist in the module, the MMC/SD has the ability to set an interrupt, The MMC/SD interrupt Control Register (INT_CNTR) allows the user to control whether these interrupts occur.

Rewriting this register clears the interrupt MMC_IRQ. When the interrupt source is from the SDIO, MMC/SD module continues to interrupt the system. In this case, the user must write to the internal registers on the SDIO card to acknowledge the SDIO interrupt.

When an interrupt is generated, it is possible that Error bits in the Status register were set as well as the Interrupt Status bit. The user must verify the Error status field to ensure that there are no errors in the SDHC operation. For example, when READ_OP_DONE status (STATUS[11]) is set or READ_OP_DONE interrupt is detected, verify STATUS[3] and STATUS[0] to ensure that the read operation completed without a CRC error or a time-out error. Another example during a write operation is when both WRITE_OP_DONE (STATUS[12]) and CRC_WRITE_ERR (STATUS[2]) are set, the write operation will end with a CRC error. Table 29-13 summarizes the relationship of interrupts, the Interrupt Control register, and Status register in the SDHC.

NOTE

For software compatibility with previous i.MX versions, some interrupts are enabled by setting the corresponding bit to 1, others are enabled by setting the corresponding bit to 0.

INT_CNTR — Interrupt Control Register — Addr 0x10013024 0x10014024

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CARD_ DET_ IRG_EN	SDIO_ WAKEUP _EN									DAT0_EN	SDIO	BUF_ READY	END_ CMD_RES	WRITE_ OP_DONE	DATA_TRAN
TYPE	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 29-12. MMC/SD Interrupt Mask Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
CARD_DET_IRQ_EN Bit 15	**Card Detect Interrupt Enable**—Enables the card detection interrupt when the card detection status bit is set. Because card detection is determined through the value of the DAT3 data line, when this card is in 4-bit mode, the data transfer in the DAT3 line might generate a false card insertion interrupt. The card insertion interrupt should be disabled the first time the card insertion is detected.	0 = Card detection interrupt disabled 1 = Card detection interrupt enabled
SDIO_WAKEUP_EN Bit 14	**SDIO Wakeup Enable**—Enable the card interrupt to wakeup system or SDHC in low-power mode. When this bit is set, SDHC can capture the SDIO card interrupt even though all clocks to SDHC are stopped. Setting this bit allows the SDIO interrupt to wakeup the system or SDHC from low-power mode. The user must set this bit before the system enters low-power mode and after the system wakes up, the user must set this bit to 0.	0 = SDIO interrupt wakeup disabled 1 = SDIO interrupt wakeup enabled
Reserved Bits 13–6	Reserved—These bits are reserved and should read 0.	

Table 29-12. MMC/SD Interrupt Mask Register Description (continued)

Name	Description	Settings
DAT0_EN Bit 5	**SD_DAT0 Enable**—Identifies how the SDIO interrupt is detected. An interrupt is determined by SD_DAT [1] = 0, but this bit is an option setting for the SDIO bit.	0 = SDIO Interrupt detection based on SD_DAT [3:1] = 110 1 = SDIO Interrupt detection based on SD_DAT [3:0] = 1101
SDIO Bit 4	**MMC/SDIO**—Masks the interrupt from the SDIO card to the MMC/SD host controller I/O interrupt mask.	0 = Not masked 1 = Masked
BUF_READY Bit 3	**Buffer Ready**—Masks the application buffer FIFO ready (full or empty) interrupt.	0 = Not masked 1 = Masked
END_CMD_RES Bit 2	**End Command Response**—Masks the end command response interrupt.	0 = Not masked 1 = Masked
WRITE_OP_DONE Bit 1	**Write Operation Done**—Masks the write operation done interrupt.	0 = Not masked 1 = Masked
DATA_TRAN Bit 0	**Data Transfer Done**—Masks the data transfer done interrupt.	0 = Not masked 1 = Masked

Table 29-13. Summary of Interrupt Mechanisms in the MMC/SD

Int #	Status Register Source Bit Name (Number)	Generate Interrupt?	INT_CNTL Register Bit Name (Number)	Interrupt/Status Clear Method
1	TIME_OUT_READ (0)	No, alert via the DATA_TRANS_DONE bit in the MMC/SD Status register	DATA_TRAN (0)	Clear status by setting the STOP_CLK bit in the STR_STP_CLK register, or by writing 1 to this bit.
2	TIME_OUT_RESP (1)	No, alert via the END_CMD_RESP bit in the MMC/SD Status register	END_CMD_RES (2)	Clear status by setting the STOP_CLK bit in the STR_STP_CLK register, or by writing 1 to this bit.
3	CRC_WRITE_ERR (2)	No, alert via the DATA_TRANS_DONE bit in the MMC/SD Status register	DATA_TRAN (0)	Clear status by setting the STOP_CLK bit in the STR_STP_CLK register, or by writing 1 to this bit.
4	CRC_READ_ERR (3)	No, alert via the DATA_TRANS_DONE bit in the MMC/SD Status register	DATA_TRAN (0)	Clear status by setting the STOP_CLK bit in the STR_STP_CLK register, or by writing 1 to this bit.
5	RESP_CRC_ERR (5)	No, alert via the END_CMD_RESP bit in the MMC/SD Status register	END_CMD_RES (2)	Clear status by setting the STOP_CLK bit in the STR_STP_CLK register, or by writing 1 to this bit.
6	APPL_BUFF_FE (6)	Yes	BUF_READY (3)	Clear interrupt by writing to INT_CNTL bit. Clear status by writing to the BUFFER_ACCESS register.

Table 29-13. Summary of Interrupt Mechanisms in the MMC/SD (continued)

Int #	Status Register Source Bit Name (Number)	Generate Interrupt?	INT_CNTL Register Bit Name (Number)	Interrupt/Status Clear Method
7	APPL_BUFF_FF (7)	Yes	BUF_READY (3)	Clear interrupt by writing to INT_CNTL bit. Clear status by reading the BUFFER_ACCESS register.
8	DATA_TRANS_DONE (11)	Yes	DATA_TRAN (0)	Clear interrupt by writing to INT_CNTL bit. Clear status by setting the STOP_CLK bit in the STR_STP_CLK register.
9	WRITE_OP_DONE (12)	Yes	WRITE_OP_DONE (1)	Clear interrupt by writing to INT_CNTL bit. Clear status by setting the STOP_CLK bit in the STR_STP_CLK register, or by writing 1 to this bit.
10	END_CMD_RESP (13)	Yes	END_CMD_RES (2)	Clear interrupt by writing to INT_CNTL bit. Clear status by setting the STOP_CLK bit in the STR_STP_CLK register, or by writing 1 to this bit.
11	SDIO_INT_ACTIVE (14)	Yes	SDIO (4)	Clear interrupt by writing to INT_CNTL bit. Clear status by resolving interrupt source on SDIO card (requires separate acknowledge command to SDIO card) and writing a 1 to this bit.

29.3.11 Commands and Arguments

The MMC/SD host controller communicates with MMC/SD cards by sending commands and arguments. The command to the SD card is always 48 bits. It contains: 1 start bit, 1 direction bit, 6 command number bits, 32 argument bit, 7 CRC bit, and 1 end bit. For detail information about the format of commands refer to the *SD Specification*.

The command to send is set in the MMC/SD Command Number register (CMD), and the argument is defined in two registers, the MMC/SD Higher Argument register (ARGH) and the MMC/SD Lower Argument register (ARGL). The full list of commands is shown in Table 29-30.

In SDHC, the command start bit, direction bit, CRC7 bit, and end bit are automatically generated through hardware. The user only needs to configure the SDHC Command register and the SDHC Argument registers to issue a command to the card.

29.3.11.1 MMC/SD Command Number Register

CMD — Command Number Register — Addr 0x10013028 0x10014028

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											COMMAND NUMBER					
TYPE	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 29-14. MMC/SD Command Number Register Description

Name	Description	Settings
Reserved Bits 31–6	Reserved—These bits are reserved and should read 0.	
COMMAND NUMBER Bits 5–0	**Command Number**—Specifies the command number to be executed.	0x00 = CMD0 0x01 = CMD1 … 0x3F = CMD63

29.3.11.2 MMC/SD Higher Argument Register

ARGH — Higher Argument Register — Addr 0x1001302C 0x1001402C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ARGUMENT HIGH															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 29-15. MMC/SD High Argument Register Description

Name	Description
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.
ARGUMENT HIGH Bits 15–0	**Higher Argument**—Specifies the higher word of the argument for the current command.

29.3.11.3 MMC/SD Lower Argument Register

ARGL — Lower Argument Register — Addr 0x10013030 0x10014030

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ARGUMENT LOW															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 29-16. MMC/SD Lower Argument Register Description

Name	Description
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.
ARGUMENT LOW Bits 15–0	**Lower Argument**—Specifies the lower word of the argument for the current command.

29.3.12 MMC/SD Response FIFO Register

The read-only MMC/SD Response FIFO register (RES_FIFO) holds the response sent back to the MMC/SD host controller after every command. The size of this FIFO is 8 × 32-bits, but only bits 15–0 are valid data.

NOTE

Whether the response is 48-bit or 128-bit, the 7-bit CRC and the 1-bit STOP-bit at the end of the response are always checked and cut by the controller hardware. The last byte of the response received from this Response FIFO register is always 0x00.

RES_FIFO — Response FIFO Register — Addr 0x10013034 0x10014034

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RESPONSE CONTENT															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 29-17. MMC/SD Response FIFO Register Description

Name	Description
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.
RESPONSE CONTENT Bits 15–0	**Response Content**—Contains the responses to every command that is sent by the MMC/SD host controller. This size of this FIFO register is 8 × 16-bit.

29.3.13 MMC/SD Buffer Access Register

The MMC/SD Buffer Access register reads or writes data to the data buffer FIFO inside the SDHC.

During read operations, the SDHC stores data received from the card into the buffer. The user must move the data out of the buffer when the buffer is full.

During write operations, the SDHC fetches data from the buffer and transfers it to the card. The user can access the data buffer through the SDHC Data Buffer Access register. The user must move data into the buffer when the buffer is empty.

The size of the FIFO is 8 × 16 bit (16 bytes) for SD 1-bit mode and 32 × 16-bit (64 bytes) for SD 4-bit mode. For reception, the SDHC generates a DMA request when the FIFO is full. When this request is received, the DMA begins to transfer data from the SDHC FIFO to the system memory by reading the Data Buffer Access register for a number of pre-defined bytes. For transmission, the SDHC generates a DMA request when the FIFO is empty. When the request is received, the DMA begins to move data from the system memory to the SDHC FIFO by writing to the Data Buffer Access register for a number of pre-defined bytes.

When using DMA to access this data buffer, the user must configure the FIFO size as 16-bit.

BUFFER_ACCESS — Buffer Access Register — Addr 0x10013038 0x10014038

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FIFO CONTENT															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 29-18. MMC/SD Buffer Access Register Description

Name	Description
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.
FIFO CONTENT Bits 15–0	**FIFO Content**—Holds transfer or receive data between system memory and card memory The size of the FIFO buffer is 8 × 16-bit in 1-bit mode, and 16 × 16-bit in 4-bit mode, FIFO full or empty active MMC_IRQ and MMC_DREQ. Only a completed FIFO read or write can clear the MMC_DREQ request.

29.4 Host Controller Functional Block Description

In this section, a brief introduction will be described on each important block, they include DMA interface, Memory Controller, Logic/Command Interpreter, and System Clock controller.

29.4.1 DMA Interface

The key function of this block is to control all data routing between External Data bus (DMA access), Internal Host data bus usage and Internal System FIFO access. This is done by a dedicated State machine, to monitor the status of Empty-FIFO (EF), Full-FIFO (FF), FIFO address, Byte/Block Counter for both the Host (Inner system) and the Application (User programming).

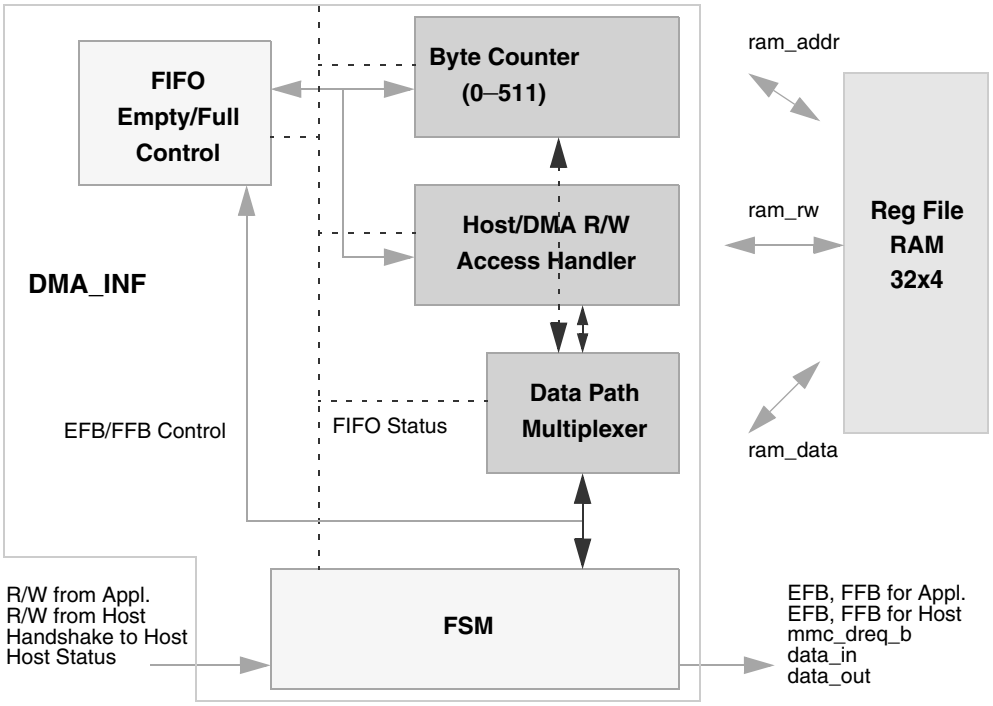


Figure 29-3. DMA Interface Block

This block also handles the burst request to the external DMA controller, internal register write-error detection, SDIO's ReadWait handling and all IP related output response.

DMA Burst request:

In SDHC, there is a 32 × 16-bit FIFO. It is used to speed up the latency during the Data transfer on the MMC/SD bus. The FIFO will be configured differently at 1-bit and 4-bit access mode. The FIFO is operated as Four 8x 16-bit FIFO at 1-bit access, while at 4-bit access, 32 × 16-bit FIFO is configured.

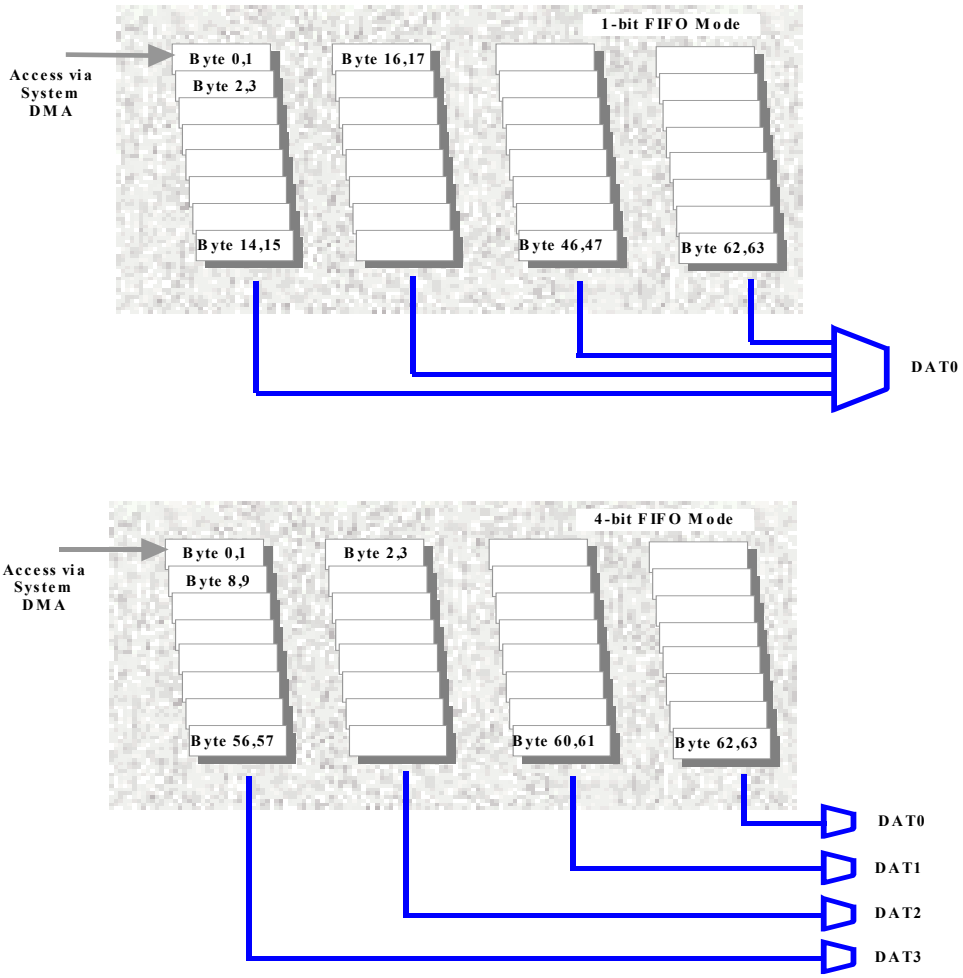


Figure 29-4. FIFO Use at 1-Bit/4-Bit

Write-Error Detection:

To avoid incorrect register access and provide error indication, the block also monitors the transfer type, Access Size and generates a Bus Error and ignores the current invalid configuration. See Table 29-2 for valid register access.

29.4.2 Memory Controller

The SDHC provides SDIO-IRQ and ReadWait service handling, Card detection, Command Response handling, all SDHC Interrupt handling and there is the sub-module to place register table.

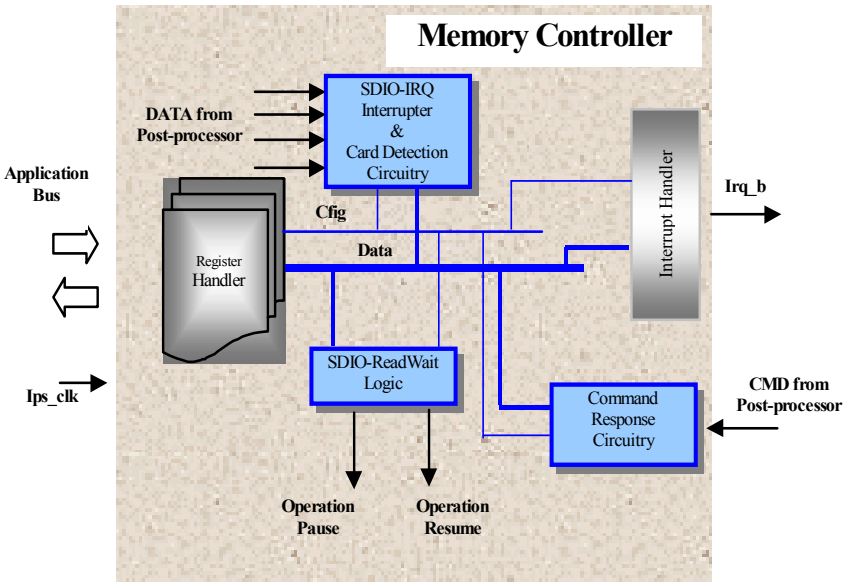


Figure 29-5. Memory Controller Block Diagram

SDIO-IRQ, ReadWait service Handling:

In SDIO, there is a 1- or 4-bit interrupt response from the SDIO peripheral card. In 1-bit data access mode, the interrupt response is active-low at line DAT[1]. DAT[1] is not used as data in 1-bit access mode. The expected changes to the controller happen when the MMC/SD clock is active. An interrupt is generated according to this low, and a system interrupt to the processor continues until the source is removed (DAT[1] goes high). In 4-bit access mode, the interrupt response is more complex. The interrupt only triggers at a particular period called the interrupt period. During the data access, the period only happens at the boundary of each block (512 byte). The controller must sample DAT[1] at this short period to identify the IRQ status of the attached card. Refer to Figure 29-6 and Figure 29-7.

SDHC can capture the SDIO interrupt during an interrupt period. When there is no activity in SDHC, the user can stop the MMCCLK to save power. Even when the MMCCLK is stopped, SDHC can capture the SDIO interrupt.

SDHC also provides a power save operation. When there is no activity in SDHC, the user can stop the clock to SDHC in the CRM to save power. When there is a need to use SDHC, the user can enable SDHC in CRM and continue operation.

INT_CNTL[14] is used to control the SDHC when the SDIO interrupt continues to be captured in low-power mode. When all clocks to SDHC are stopped, SDHC continues to capture SDIO interrupts asynchronously when this bit is set. In this case, SDHC issues interrupts to the CPU. The user must enable the SDHC clock in the IRQ service routine to handle this interrupt. After the clock to SDHC is enabled, software must disable the SDIO interrupt wakeup as the system will be expecting a synchronous interrupt. The user can enable the wakeup before the SDHC enters low-power mode again, when all operations are complete.

This feature can also be used to wakeup the system from low-power mode.

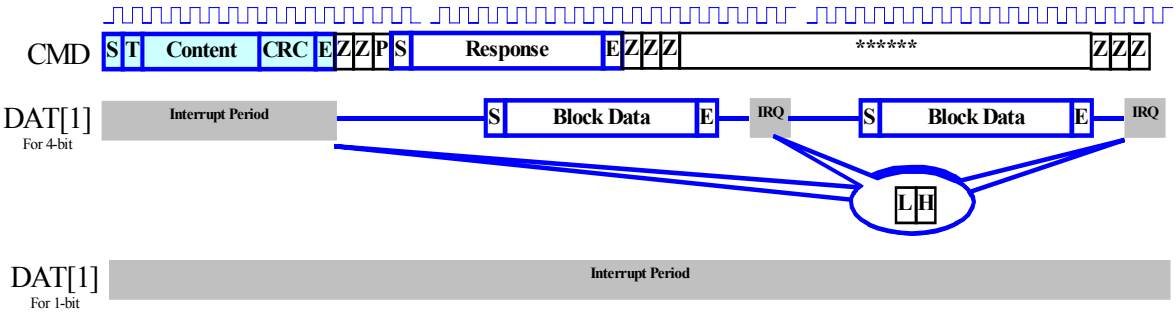
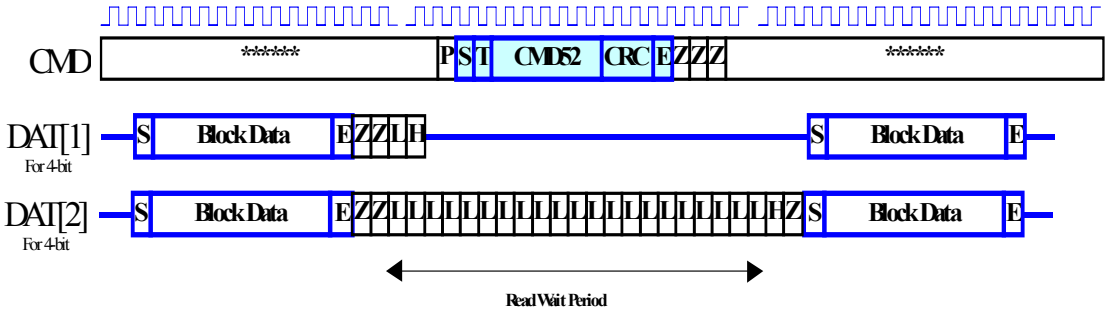


Figure 29-6. SDIO IRQ Timing Diagram

ReadWait is another feature of SDIO. ReadWait allows users to submit commands during data transfer. Figure 29-7 illustrates the ReadWait operation showing the temporary pauses of the data transfer operation counter and status. As shown, the clock continues to run and allows the user to submit commands as normal. After all commands are submitted, the user returns to the data transfer operation, and all counter and status values resume and access continues. To restore the command status, the user must restore the Command and Argument registers as well as the Command and Data Control register. The user should set the CMD_RESUME bit to restore the Command and Data Control register, otherwise the READ command is sent to the card.


CMD
****** P S T CMD52 CRC E Z Z Z ******
DAT[1]
For 4-bit
S Block Data E Z Z Z L H
S Block Data E
DAT[2]
For 4-bit
S Block Data E Z Z Z L L L

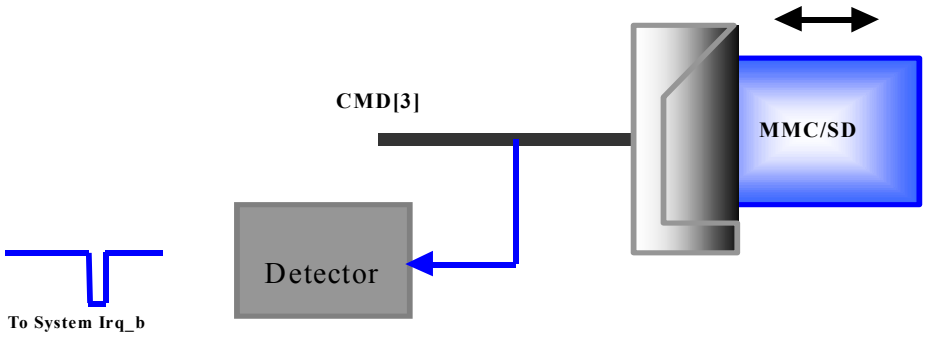


Figure 29-8. Card Detection Mechanism

SDHC Interrupt Handling:

Interrupt generated from SDHC can come from two source type, Error generation, Status Indication. In SDHC, we do check on Response, Data's CRC and few of inner watchdog timer, Error is generated once any of those result incorrectly. Status indication includes Response done, transfer done, FIFO status etc. Interrupts masking and generation is handled here.

29.4.3 Command and Data Interpreter

The command and data interpreters are very similar. They consist of three parts, the inner state machine, the sub-module controller, and the CRC accelerator.

The command interpreter controls everything related to command line (CMD), this includes command data sequence generation, command response extraction, CRC generation and checking, response time-out, and as on. To achieve these functions, a state machine, logic control, and CRC accelerator are used. The block diagram for the command and data interpreters is shown in Figure 29-9.

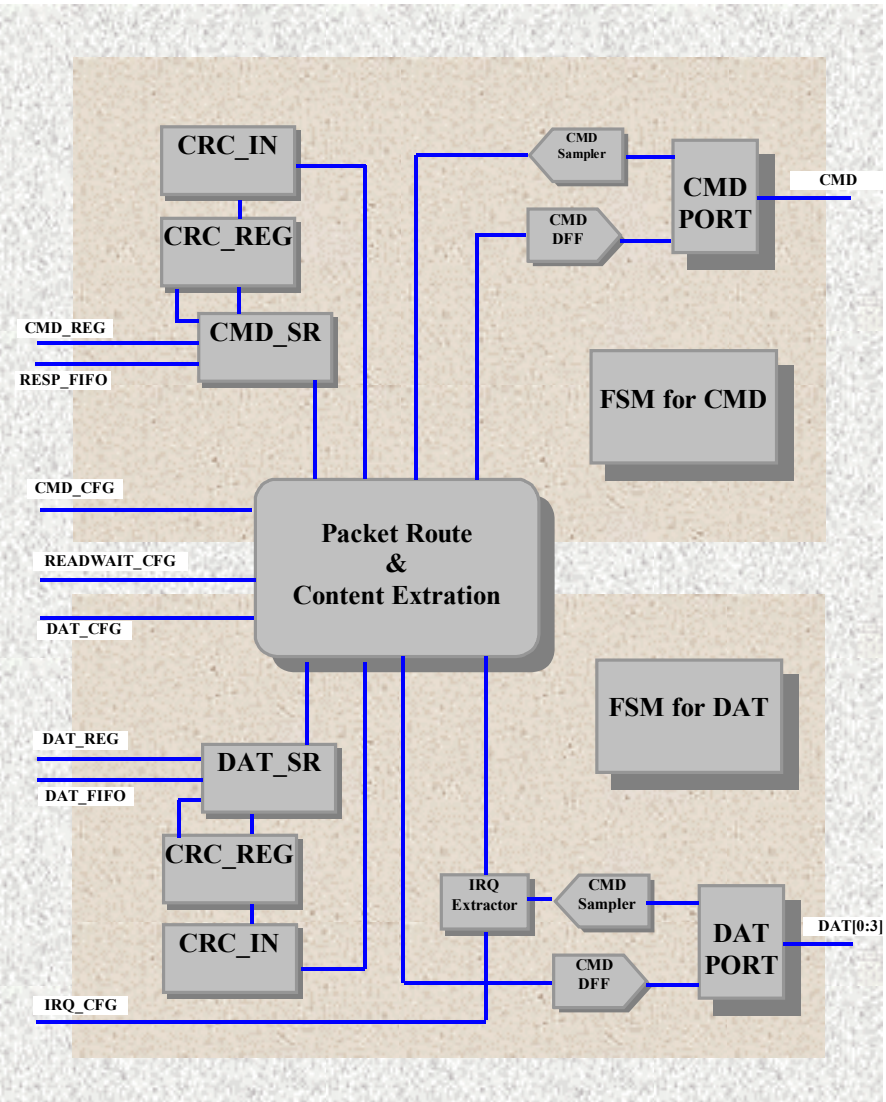


Figure 29-9. Command Interpreter Block Diagram

As stated, the CRC's generation and detection are one of the core hardware designs in the SDHC module. To minimize the gate count, the internal command shift register is re-used as the CRC shift register. Figure 29-10 illustrates the CRC command shift register. The corresponding polynomials for CMD and DAT follow.

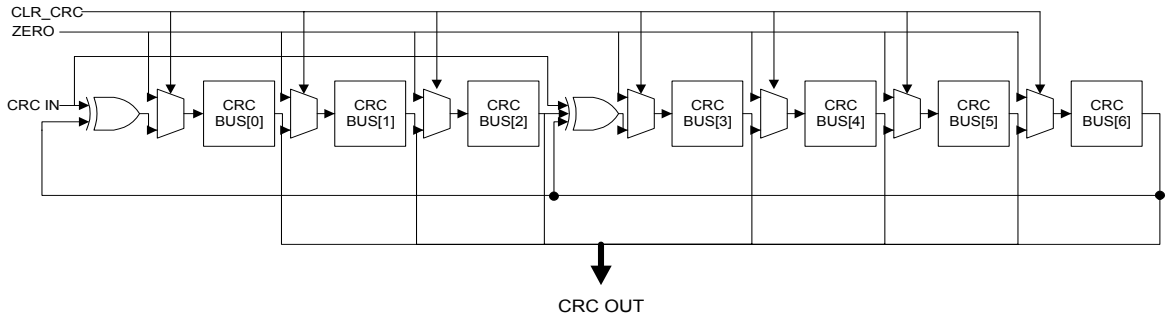


Figure 29-10. Command CRC Shift Register (DAT has Similar Structure)

For CMD:

Generator polynomial: G(x) = x7 + x3 + 1.

M(x) = (first bit) × xn + (second bit) × xn-1 +…+ (last bit) × x0.

CRC[6:0] = Remainder [(M(x) × x7) / G(x)].

For DAT:

Generator polynomial: G(x) = x16 + x12 + x5 +1.

M(x) = (first bit) × xn + (second bit) × xn-1 +…+ (last bit) × x0.

CRC[15:0] = Remainder [(M(x) × x16) / G(x)].

29.4.4 System Clock Controller

To maximize the power-saving capability of the i.MX21 processor, several stages of clocks are used within the module. The input clock (ipg_clk), operating around 20 MHz–100 MHz, first passes through a prescaler to adjust and maintain the inner clock under 20 MHz, as restricted by the maximum operation frequency of MMC/SD card. Only a few of the synchronization registers use this high frequency clock.

The down-graded clock, CLK_20M, then passes to a clock divider. The divider is used to program the final MMC/SD bus clock for different card applications. The CLK_20M accounts for about 10 percent of the total circuitry of the SDHC module.

After the division, the resulting clock, CLK_DIV, is around 0 MHz–20 MHz, and will be used throughout the module. To maximize power-saving during the operation, the MMC/SD bus clock will pause and resume according to the status—for example, when the FIFO is full during the Card Read operation, the bus clock stops when there are no further Read operations to the FIFO, and it resumes when the FIFO is cleared by the user (DMA). This is similar to other situations when the module stops the clock when it is unnecessary.

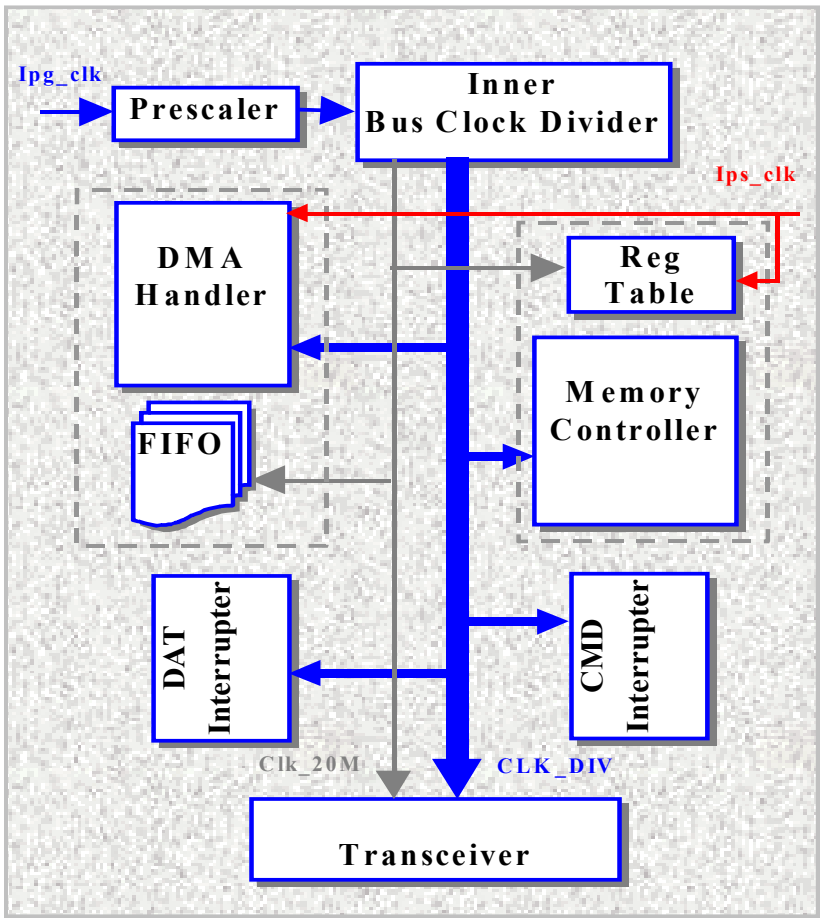


Figure 29-11. Clock Tree Used in SDHC

The controller controls the rate of the host main clock and checks whether it is on or off. The clock is turned off by setting the LSB of the STR_STP_CLK register and is turned on by setting the MSB of the STR_STP_CLK (writing operations to the host can be performed only when the clock is turned off). To change the clock rate, the application must write a new value in the CLK_RATE register.

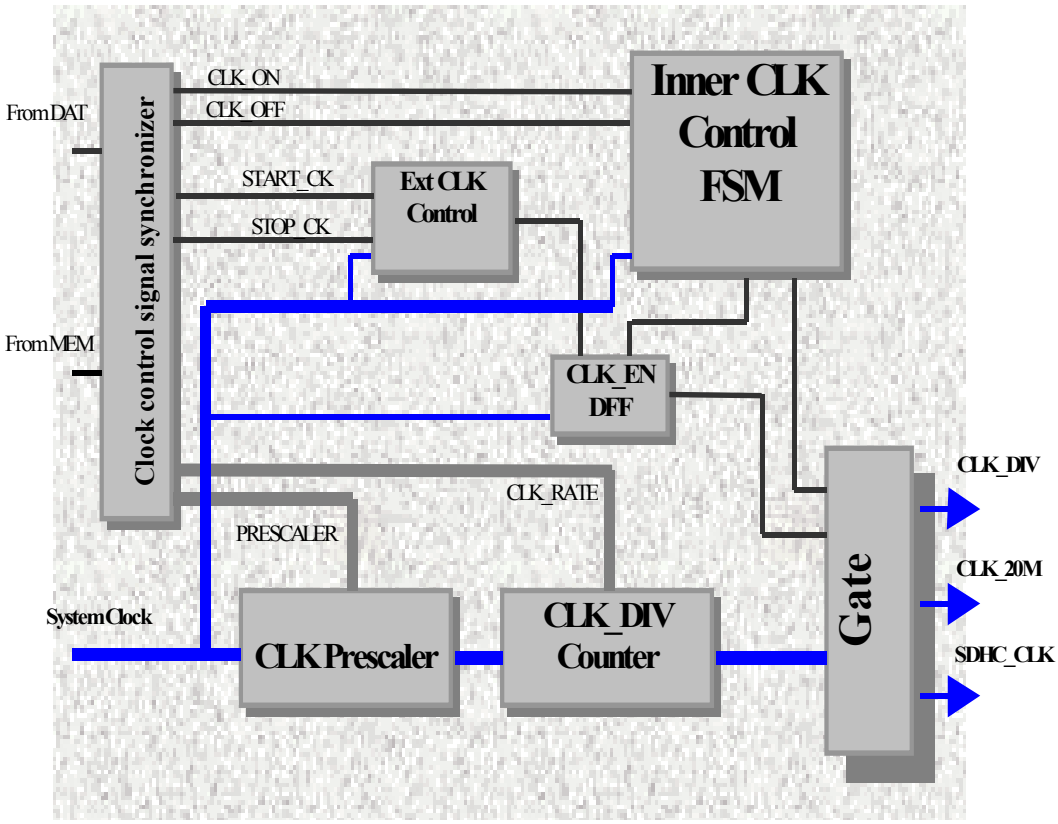


Figure 29-12. System Clock Control Unit

Another purpose of the clock controller is to enable the application to divide the clock, and stop and start the clock without generating a glitch in the process, This is done by insuring that every time the clock stops, it stops when it is low. This is done by insuring that all the clocks from the CLK_DIV counter and the master clock is low, the stop clock data unit enables the data state machine to stop the clock when the application is too slow and cannot update the data FIFO on time in a multiple block or stream write and read.

29.4.5 Command Response Timing on SDHC Bus

The card identification and card operation conditions timing are processed in the open-drain mode. The card response to the host command starts after exactly N_{ID} clock cycles. For the Card address assignment, SET_RCA is also processed in the open-drain mode. The minimum delay between the host command and card response is NCR clock cycles.

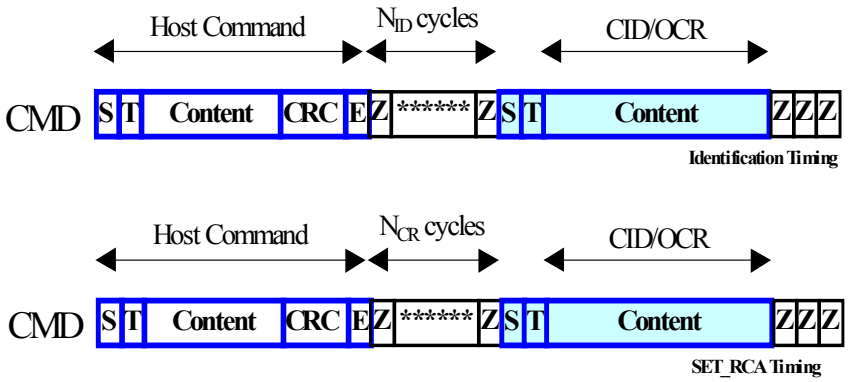


Figure 29-13. Identification Mode Timing Diagrams

After a card receives its RCA it will switch to data transfer mode. As shown on the first wave-train, CMD lines at this mode is driven with push-pull drivers. The command is followed by a period of two Z bits (allowing time for direction switching on the bus) and then by P bits pushed up by the responding card. At the rest of two wave-trains show the separating period N_{RC} and N_{CC}.

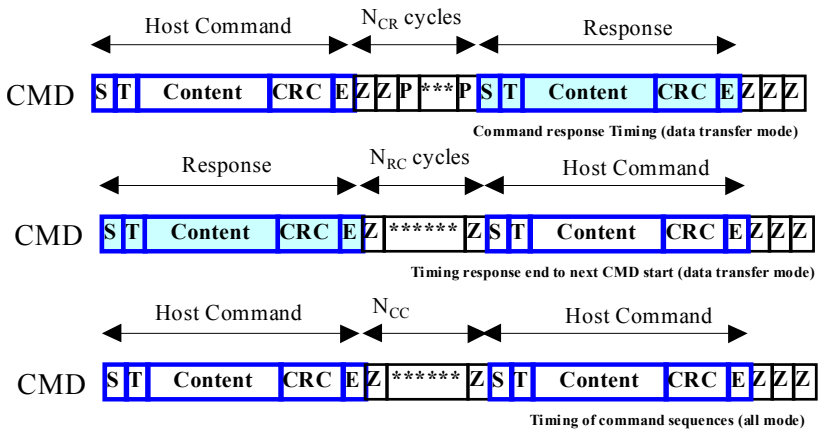


Figure 29-14. Data Transfer Mode Timing Diagram

In a read operation, the sequence starts with a single block read command which specifies the start address in the argument field. The response is sent on the CMD lines as usual. Data transmission from the card starts after the access time delay N_{AC} beginning from the end bit of the read command. If in multiple block read mode, the card sends continuous flow of data blocks with distance N_{AC}, and is terminated by a stop transmission command. The data stops two clock cycles after the end bit of the stop command.

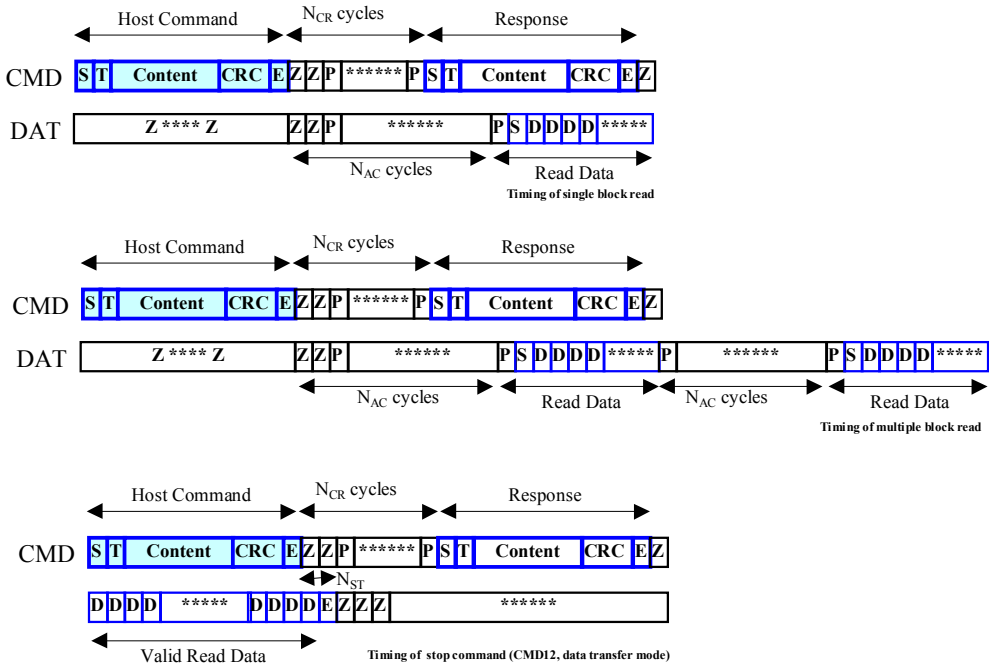


Figure 29-15. Data Read Timing Diagram

Figure 29-16 shows the basic write operation timing. Similar to read, after the card response the data transfer starts N_{WR} cycles. The data is suffixed with CRC check bits to allow the card to check it for transmission errors. The card sends back the CRC check result as a CC status token on the data line. In the case of transmission error the card sends a negative CRC status (101), and a positive CRC status (010) if non errorneous transmission. Card expects continuous flow of data blocks if in Multiple block mode, and its flow is terminated by a stop transmission command.

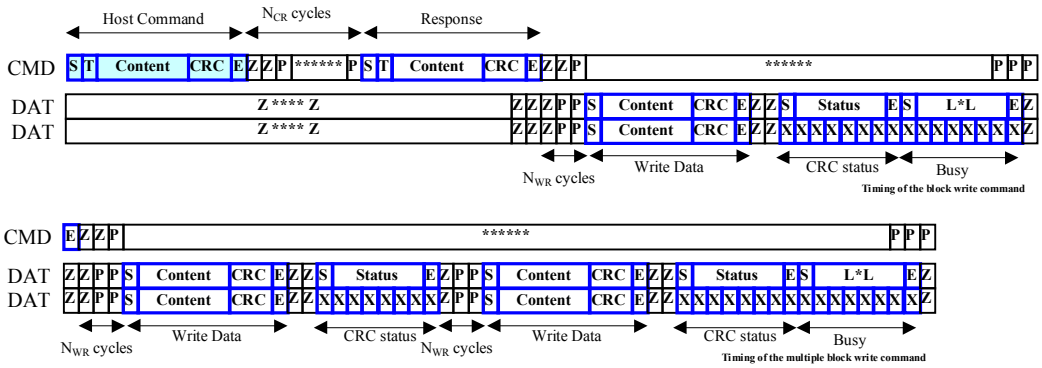


Figure 29-16. Data Write Timing Diagram

The Stop command might occur during different card states, Figure 29-17 shows different scenarios on the bus.

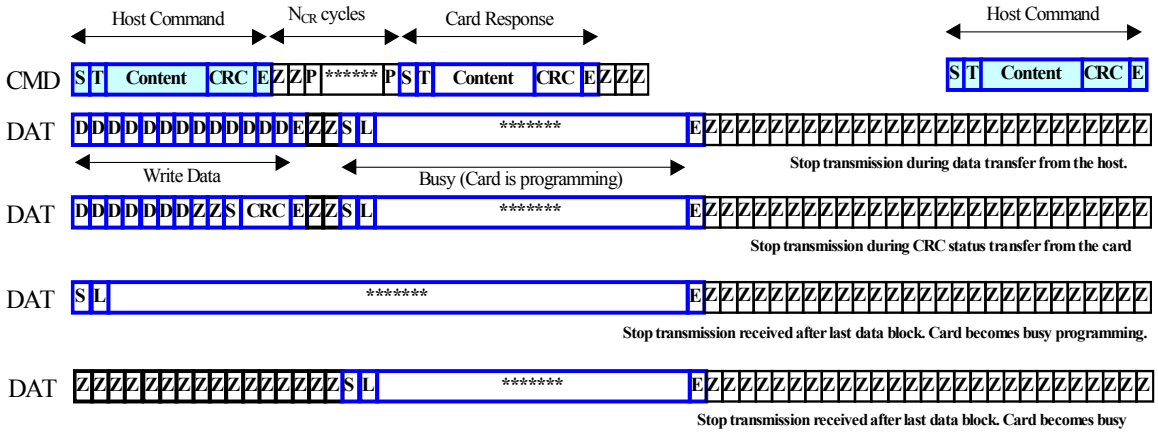


Figure 29-17. Stop Transmission During Different Scenarios

Table 29-19. SDHC Bus Clock Timing Parameters

Parameter	Symbol	Min	Max	Unit	Remarks
SDHC Bus Clock, CLK (All values are referred to min. (V_{IH}) and max (V_{IL})					
Command Response Cycle	N_{CR}	2	64	clock cycles	–
Identification Response Cycle	N_{ID}	5	5	clock cycles	–
Access Time Delay Cycle	N_{AC}	2	TAAC + NSAC	clock cycles	–
Command Read Cycle	N_{RC}	8	–	clock cycles	–
Command-Command Cycle	N_{CC}	8	–	clock cycles	–
Command Write Cycle	N_{WR}	2	–	clock cycles	–
Stop Transmission Cycle	N_{ST}	2	2	clock cycles	–
TAAC: Data read access time -1 defined in CSD register bit[119:112] NSAC: Data read access time -2 in CLK cycles (NSAC*100) defined in CSD register bit[111:104]					
–	–	Z	High impedance state (≥ = 1)	S	Start bit (0)
–	–	D	Data bits	T	Transmitter bit (Host = 1, Card = 0)
–	–	*	Repetition	P	One-cycle pull-up (1)
–	–	CRC	Cyclic redundancy check bits (7 bits)	E	End bit (1)
–	–		Card active		Host active

29.4.6 Response Type

All responses are sent via the command line CMD. The response transmission always starts with the left bit of the bitstring corresponding to the response code word. The code length depends on the response type. A response always starts with a start bit, followed by the bit indicating the direction of transmission. A

value denoted by *x* in the tables below indicates a variable entry. All responses except for the type R3 are protected by a CRC. Every command code word is terminated by the end bit.

R1 (normal response command)—Code Length 48-Bit

The bits 45:40 indicate the index of the command to be response to, this value being interpreted as a binary coded number. The status of the card is coded in 32 bits. Note that in case that data transfer to the card is involved then a busy signal may appear on the data line after the transmission of each block of data. The host shell check for busy after data block transmission.

Table 29-20. R1 Normal Response Command

Bit Position	47	46	[45:40]	[39:8]	[7:1]	0
Value	0	0	x	x	x	1
Description	start bit	transmission bit	command index	card status	CRC7	end bit

R1b

Identical to R1 with an optional busy signal transmitted on the data line. The card may become busy after receiving these commands based on its state prior to the command reception. The host shell check for busy at the response.

R2 (CID, CSD register)—Code Length 136-Bit

The contents of the CID register are sent as a response to the commands CMD2 and CMD10. The contents of the CSD register are sent as a response to CMD9. Only the bits [127...1] of the CID and CSD are transferred, the reserved bit [0] of these registers is replaced by the end bit of the response.

Table 29-21. R2—CID, CSD Register

Bit Position	135	134	[133:128]	[127:1]	0
Value	0	0	111111	x	1
Description	Start bit	Transmission bit	Reserved	CID/CSD including CRC7	End bit

R3 (OCR register)—Code Length 48-Bits

The contents of the OCR register is sent as a response to CMD1 (MMC) or ACMD41 (SD).

Table 29-22. R3—OCR Register

Bit Position	47	46	[45:40]	[39:8]	[7:1]	0
Value	0	0	x	x	x	1
Description	start bit	transmission bit	command index	card status	CRC7	end bit

R4 (Fast IO for MMC Only)—Code Length 48-Bit

The argument field contains the RCA of the addressed card, the register address to be read out or written to and its contents.

Table 29-23. R4—Fast IO for MMC Only

Bit Position	47	46	[45:40]	[39:8]	[7:1]	0
Value	0	0	100111	x	x	1
Description	start bit	transmission bit	CMD39	RCA[31:16], Register addr[15:8], Read register content[7:0]	CRC7	end bit

R4b (SDIO Only)

Table 29-24. R4b—SDIO Only

Bit Position	47	46	[45:40]	39	[38:36]	35	[34:32]	[31:8]	[7:1]	0
Value	0	0	x	x	x	x	x	x	x	1
Description	start bit	dir bit	Resv.	Card is ready	No. of IO functions	Mem Present	Stuff bits	IO ORC	Resv.	end bit

R5 (Interrupt request for MMC only)—Code Length 48-Bit

The argument field contains the RCA of the addressed card, the register address to be read out or written to, and its contents.

Table 29-25. R5—Interrupt Request for MMC Only

Bit Position	47	46	[45:40]	[39:8]	[7:1]	0
Value	0	0	101000	x	x	1
Description	start bit	transmission bit	CMD40	RCA[31:16], Reserved[15:0]	CRC7	end bit

R6 (For SDIO Only)

The card status bits are changed when CMD3 is sent to an IO only card. In this case, the 16 bits of response will be the SDIO-only values:

bit[15] – COM_CRC_ERROR

bit[14] – ILLEGAL_COMMAND

bit[13] – ERROR

bit[12:0] – Reserved

Table 29-26. R6—For SDIO Only

Bit Position	47	46	[45:40]	[39:8]	[7:1]	0
Value	0	0	000011	x	x	1
Description	start bit	direction bit	CMD3	RCA[31:16], card status[15:0]	CRC7	end bit

29.5 Functional Example on MMC/SD

All communication between system and cards is controlled by the host. The host sends commands of two types: broadcast and addressed (point-to-point) commands.

Broadcast commands are intended for all cards, command like “Go_Idle_State”, “Send_Op_Cond”, “All_send_CID” and “Set_relative_Addr” are using way of broadcasting. During Broadcast mode, all cards are in open-drain mode, to avoid bus contention.

After Broadcast commands “Set_relative_Addr” issue, cards are enter standby mode, and Addressed command will be used from now on, in this mode, CMD/DAT will return to push-pull mode, to have maximum driving for maximum operation frequency.

As mentioned in the above section, the MMC and the SD are similar product. Besides the 4x bandwidth and the built-in encryption, they are being programmed similarly, the following example will show how to “initizatize”, “content access” and “content protection” on the Cards.

29.5.1 Command Submit—Response Receive Basic Operation

Code Example 29-1 on page -36 is the program flow used to submit a command to card(s), <command_no> is the targeted command, <argh_no,argl_no> are the corresponding argument, <cmd_dat_cont> is the command configuration required, and finally <int_mask_value> is the interrupt mask used in the user program.

Example 29-1. Submit Command Code

```
send_cmd_wait_resp(command_no, argh_no, argl_no, cmd_dat_cont, int_mask_value)
{
    write_reg(COMMAND, <command_no>);
    write_reg(ARGH, <argh_no>);
    write_reg(ARGL, <argl_no>);
    write_reg(CMD_DAT_CONT, <cmd_dat_cont>);
    write_reg(STR_STP_CLK, 0x010);
    read_reg(STATUS);
    while(!STATUS[8]) Read_reg(STATUS); // Wait till clock is start,
    command submit now.
    while(irq_status); // Wait interrupt generated (End Command Response)
    Write_reg(INT_MASK, <int_mask_value>);
    read_reg(STATUS); // Check whether it is End Command Response or Time out.
    write_reg(STR_STP_CLK, 0x001);
    read_reg(STATUS);
    while(STATUS[8]) Read_reg(STATUS); // Wait till clock is stop,
    command-response end.
}
```

29.5.2 Card Identification Mode

While in card identification mode the host resets all the cards that are in card identification mode, validates operation voltage range, identifies cards and asks them to publish Relative Card Address (RCA). This operation is done to each card separately on its own CMD line. All data communication in the Card Identification Mode uses the command line (CMD) only.

29.5.2.1 Card Detect

Code Example 29-2 is example code to detect card via the host controller.

Example 29-2. Card Detect Example Code

```
card_detect()
{
      while(irq_status);// Wait interrupt generated (Card Presence)
      while(!STATUS[15]) Read_reg(STATUS); // Wait till card presence is detect.
      Write_reg(INT_MASK, 0x40);
}
```

29.5.2.2 Reset

The host consists of 3 type of Reset, hardware reset (Card and Host) which is driven by POR; software reset (Host Only) is proceed by the write operation on THE STR_STP_CLK register, follow the recommended sequence as mentioned in the previous section; finally is the Card reset (Card Only), the command Go_Idle_State, CMD0 is the software reset command for MMC and SD Memory Card, and sets each card into Idle State regardless of the current card state; while in SDIO card, CMD52 is used to write IO reset in CCCR. The cards are initialized with a default relative card address (RCA=0x0000) and with a default driver stage register setting (lowest speed, highest driving current capability).

Example 29-3.

```
software_reset()
{
        write_reg(STR_STP_CLK, 0x8);
        write_reg(STR_STP_CLK, 0x9);
        write_reg(STR_STP_CLK, 0x1);
        write_reg(STR_STP_CLK, 0x1);
        write_reg(STR_STP_CLK, 0x1);
        write_reg(STR_STP_CLK, 0x1);
        write_reg(STR_STP_CLK, 0x1);
        write_reg(STR_STP_CLK, 0x1);
        write_reg(STR_STP_CLK, 0x1);
        write_reg(STR_STP_CLK, 0x1);
        write_reg(CLK_RATE, 0x3F);// Set the lowest clock for initization
        write_reg(READ_TO, 0x2DB4);
        send_cmd_wait_resp(CMD_GO_IDLE_STATE, 0x0, 0x0, 0x80, 0x40);
}
```

29.5.2.3 Voltage Validation

All cards shall be able to establish communication with the host using any operation voltage in the maximal allowed voltage range specified in this standard. However, the supported minimum and maximum values for Vdd are defined in Operation Conditions register (OCR) and may not cover the whole range. Cards that store the CID and CSD data in the payload memory would be able to communicate these information only under data transfer Vdd conditions. That means if host and card have non compatible Vdd ranges, the card will not be able to complete the identification cycle, nor to send CSD data.

Therefore, a special command Send_Op_Cont (CMD1 for MMC), SD_Send_Op_Cont (CMD41 for SD Memory) and IO_Send_Op_Cont (CMD5 for SDIO) are designed to provide a mechanism to identify and reject cards which do not match the Vdd range desired by the host. This is accomplished by the host sending the required Vdd voltage window as the operand of this command. Cards which can not perform data transfer in the specified range must discard themselves from further bus operations and go into Inactive State. By omitting the voltage range in the command, the host can query each card and determine the common voltage range before sending out-of-range cards into the Inactive State. This query should be used if the host is able to select a common voltage range or if a notification to the application of non usable cards in the stack is desired.

Example 29-4.

```
        voltage_validation(voltage_range_h, voltage_range_l)
        {
                send_cmd_wait_resp(IO_RW_DIRECT, 0x8800, 0x0608, 0x05, 0x40); // Reset IO card
                send_cmd_wait_resp(IO_SEND_OP_COND, 0x0, 0x0, 0x04, 0x40); // Send SDIO
operation voltage command
                if(End Command Response true & No. of IO functions> 0 )// it is SDIO
                {       IORDY = 0;
                        while(!(IORDY in I/O ORC)) {// set voltage range
                        send_cmd_wait_resp(IO_SEND_OP_COND, voltage_range_h,
voltage_range_l, 0x04, 0x40);}
                        if(Memory Present flag true)
                                Card = combo; // i.e. SDIO + Memory
                        else
                                Card = sdio;
                }
                else    // SD or MMC
                {       send_cmd_wait_resp(GO_IDLE_STATE, 0x0, 0x0, 0x80, 0x40);// MMC, SD
reset
                        send_cmd_wait_resp(APP_CMD, 0x0, 0x0, 0x01, 0x40);// Application
Command follows
                        if(End Command Response true)
                        {       send_cmd_wait_resp(SEND_OP_COND, 0x0, 0x0, 0x01, 0x40); //
SD card found
                                while(!(card init finished)) {
                                send_cmd_wait_resp(APP_CMD, 0x0, 0x0, 0x01, 0x40);
                                send_cmd_wait_resp(SEND_OP_COND, voltage_range_h,
voltage_range_l, 0x01, 0x40);}
                                Card = sd;
                        {
                        else
                        {       send_cmd_wait_resp(SEND_OP_COND, 0x0, 0x0, 0x01, 0x40); //
MMC card found
                                if(End Command Response true)
                                {Card = mmc;
                                while(!(card init finished)) {
                                send_cmd_wait_resp(SEND_OP_COND, voltage_range_h,
voltage_range_l, 0x01, 0x40); }}
                                else     { Card = No card or failed contact; }
                        }
                }
}
```

29.5.2.4 Card Registry

Card registry on MMC and SD card are different.

For SD Card, Identification process start at clock rate F_{od}, while CMD line output drives are push-pull drivers instead of open-drain. After the bus is activated the host will request the cards to send their valid operation conditions. The response to ACMD41 is the operation condition register of the card. The same command shall be send to all of the new cards in the system. Incompatible cards are sent into Inactive State. The host then issue the command All_Send_CID (CMD2) to each card and get its unique card identification (CID) number. Card that is unidentified—that is, which is in Ready State, sends its CID number as the response. After the CID was sent by the card it goes into Identification State. Thereafter, the host issues Send_Relative_Addr (CMD3) asks the card to publish a new relative card address (RCA), which is shorter that CID and which will be used to address the card in the future data transfer mode. Once the RCA is received the card state changes to the Stand-by State. At this point, if the host wants that the card will have another RCA number, it may ask the card to publish a new number by sending another Send_Relative_Addr command to the card. The last published RCA is the actual RCA of the card.

The host repeats the identification process—that is, the cycles with CMD2 and CMD3 for each card in the system.

In MMC, the host starts the card identification process in open-drain mode with the identification clock rate F_{od}. The open drain driver stages on the CMD line allow parallel card operation during card identification. After the bus is activated the host will request the cards to send their valid operation conditions (CMD1). The response to CMD1 is the ‘wired or’ operation on the condition restrictions of all cards in the system. Incompatible cards are sent into Inactive State. The host then issues the broadcast command All_Send_CID (CMD2), asking all cards for their unique card identification (CID) number. All unidentified cards—that is, those which are in Ready State, simultaneously start sending their CID numbers serially, while bit-wise monitoring their outgoing bitstream. Those cards, whose outgoing CID bits do not match the corresponding bits on the command line in any one of the bit periods stop sending their CID immediately and must wait for the next identification cycle. Since CID is unique for each card, only one card can be successfully send its full CID to the host. This card then goes into Identification State. Thereafter, the host issues Set_Relative_Addr (CMD3) to assign to this card a relative card address (RCA). Once the RCA is received the card state changes to the Stand-by State, and the card does not react to further identification cycles, and its output switches from open-drain to push-pull. The host repeat the process—that is, CMD2 and CMD3, until the host receive time-out condition to recognize completion of the identification process.

Example 29-5.

```
card_registary()
{
        while(ResponseTO from STATUS){
                if(card==combo or sdio)
                {
                        send_cmd_wait_resp(SET_RELATIVE_ADDR, 0x00, 0x00,
0x01, 0x40);
                        rca = SDIO_RCA = address from response FIFO;
                }
                else if(card==sd)
                {
```

```
                                        send_cmd_wait_resp(ALL_SEND_CID, 0x00, 0x00,
0x02, 0x40);
                                        send_cmd_wait_resp(SET_RELATIVE_ADDR, 0x00, 0x00,
0x01, 0x40);
                                        rca = SD_RCA = address from response FIFO;
                                }
                                else if(card==mmc)
                                {
                                        send_cmd_wait_resp(ALL_SEND_CID, 0x00, 0x00,
0x02, 0x40);
                                        rca = MMC_RCA = 0x1;
                                        send_cmd_wait_resp(SET_RELATIVE_ADDR, MMC_RCA,
0x00, 0x01, 0x40);
                                }
                                else
                                        exit due to card not identified;
                        }
                        send_cmd_wait_resp(SELECT_CARD, rca, 0x00, 0x41, 0x40);
                }
```

29.5.3 Card Access

29.5.3.1 Block Access—Block Write and Block Read

Block Write

During block write (CMD24–27) one or more blocks of data are transferred from the host to the card with a CRC appended to the end of each block by the host. A card supporting block write shall always be able to accept a block of data defined by WRITE_BL_LEN. If the CRC fails, the card shall indicate the failure on the DAT line (see below); the transferred data will be discarded and not written, and all further transmitted blocks (in multiple block write mode) will be ignored.

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed (CSD parameter WRITE_BLK_MISALIGN is not set), the card shall detect the block misalignment error and abort programming before the beginning of the first misaligned block. The card shall set the ADDRESS_ERROR error bit in the status register, and while ignoring all further data transfer, wait in the Receive-data-State for a stop command. The write operation shall also be aborted if the host tries to write over a write protected area. In this case, however, the card shall set the WP_VIOLATION bit.

Programming of the CID and CSD registers does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer. If this match fails, then the card will report an error and not change any register contents. Some cards may require long and unpredictable times to write a block of data. After receiving a block of data and completing the CRC check, the card will begin writing and hold the DAT line low if its write buffer is full and unable to accept new data from a new WRITE_BLOCK command. The host may poll the status of the card with a SEND_STATUS command (CMD13) at any time, and the card will respond with its status. The status bit READY_FOR_DATA indicates whether the card can accept new data or whether the write process is still in progress). The host may deselect the card by issuing CMD7 (to select a different card) which will displace the card into the

Disconnect State and release the DAT line without interrupting the write operation. When reselecting the card, it will reactivate busy indication by pulling DAT to low if programming is still in progress and the write buffer is unavailable.

With DMA:

Example 29-6.

```
                block_write(rca, nob, addr_h, addr_l, buswidth)
                {       send_cmd_wait_resp(SEND_STATUS, rca, 0x00, 0x01, 0x40);
                        while(!Ready for data in card status is true){
                                send_cmd_wait_resp(SEND_STATUS, rca, 0x00, 0x01, 0x40);}
                        write_reg(NOB, <nob>);
                        send_cmd_wait_resp(SET_BLOCKLEN, 0x00, 0x0200, 0x01, 0x40);
                        if(buswidth==4-bit mode)
                        {       send_cmd_wait_resp(APP_CMD, rca, 0x0, 0x01, 0x40);
                                send_cmd_wait_resp(SET_BUS_WIDTH, 0x00, 0x02, 0x01, 0x40);
                        }
                        Configure the DMA for FIFO write operation (BUFFER_ACCESS,
SDRAM_ADDR, nob);
                        // Set DMA source address = SDRAM_ADDR
                        // Set DMA target address = BUFFER_ACCESS
                        // Set DMA total byte transfer = nob
                        // Set DMA burst depth = 8 if 1-bit mode, = 32 if 4-bit mode
                        if(nob==1)
                                send_cmd_wait_resp(WRITE_SINGLE_BLOCK, addr_h, addr_l,
0x19, 0x40);
                        else
                                send_cmd_wait_resp(WRITE_MULTIPLE_BLOCK, addr_h, addr_l,
0x19, 0x40);
                        while(!FIFO empty in STATUS is true);
                        Enable DMA operation;
                        while(!Access Operation Done in STATUS true);
                        while(!card bus is stop);
                        if(nob > 1) {
                                send_cmd_wait_resp(STOP_TRANS, 0x00, 0x00, 0x41, 0x40);
                        }
                }
```

Block Read

Block read is similar to stream read, except the basic unit of data transfer is a block whose maximizes is defined in the CSD (READ_BL_LEN). If READ_BL_PARTIAL is set, smaller blocks whose starting and ending address are entirely contained within one physical block (as defined by READ_BL_LEN) may also be transmitted. Unlike stream read, a CRC is appended to the end of each block ensuring data transfer integrity. CMD17 (READ_SINGLE_BLOCK) initiates a block read and after completing the transfer, the card returns to the Transfer State. CMD18 (READ_MULTIPLE_BLOCK) starts a transfer of several consecutive blocks. Blocks will be continuously transferred until a stop command is issued. If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed, the card shall detect a block misalignment at the beginning of the first mis-aligned block, set the ADDRESS_ERROR error bit in the status register, abort transmission and wait in the Data State for a stop command.

Example 29-7. :With DMA

```
                block_read(rca, nob, addr_h, addr_l, buswidth)
                {       send_cmd_wait_resp(SEND_STATUS, rca, 0x00, 0x01, 0x40);
                        while(!Ready for data in card status is true){
                                send_cmd_wait_resp(SEND_STATUS, rca, 0x00, 0x01, 0x40);}
                        write_reg(NOB, <nob>);
                        send_cmd_wait_resp(SET_BLOCKLEN, 0x00, 0x0200, 0x01, 0x40);
                        if(buswidth==4-bit mode)
                        {       send_cmd_wait_resp(APP_CMD, rca, 0x0, 0x01, 0x40);
                                send_cmd_wait_resp(SET_BUS_WIDTH, 0x00, 0x02, 0x01, 0x40);
                        }
                        Configure the DMA for FIFO read operation (BUFFER_ACCESS, SDRAM_ADDR,
nob);
                        // Set DMA source address = BUFFER_ACCESS
                        // Set DMA target address = SDRAM_ADDR
                        // Set DMA total byte transfer = nob
                        // Set DMA burst depth = 8 if 1-bit mode, = 32 if 4-bit mode
                        if(nob==1)
                                send_cmd_wait_resp(READ_SINGLE_BLOCK, addr_h, addr_l,
0x09, 0x40);
                        else
                                send_cmd_wait_resp(READ_MULTIPLE_BLOCK, addr_h, addr_l,
0x09, 0x40);
                        Enable DMA operation;
                        while(!Data Transfer Done in STATUS true);
                        while(!card bus is stop);
                        if(nob > 1) {
                                send_cmd_wait_resp(STOP_TRANS, 0x00, 0x00, 0x41, 0x40);
                        }
                }
```

29.5.3.2 Stream Access—Stream Write and Stream Read (MMC Only)

Stream Write

Stream write (CMD20) starts the data transfer from the host to the card beginning from the starting address until the host issues a stop command. If partial blocks are allowed (if CSD parameter WRITE_BL_PARTIAL is set) the data stream can start and stop at any address within the card address space, otherwise it shall start and stop only at block boundaries. Since the amount of data to be transferred is not determined in advance, CRC can not be used. If the end of the memory range is reached while sending data and no stop command has been sent by the host, all further transferred data is discarded.

The maximum clock frequency for stream write operation is given by the following formula:

$$\text{max. speed} = \text{min. (TRAN\_SPEED, (8} \times \text{2 WRITE\_BL\_LEN - NSAC) / (TAAC} \times \text{R2W\_FACTOR))},$$

these parameters being defined in Chapter 5. If the host attempts to use a higher frequency, the card may not be able to process the data and will stop programming, set the OVERRUN error bit in the status register, and while ignoring all further data transfer, wait (in the Receive-data-State) for a stop command. The write operation shall also be aborted if the host tries to write over a write protected area. In this case, however, the card shall set the WP_VIOLATION bit.

Example 29-8.

```
stream_write(rca, nob, addr_h, addr_l, buswidth)
{       send_cmd_wait_resp(SEND_STATUS, rca, 0x00, 0x01, 0x40);
        while(!Ready for data in card status is true){
                send_cmd_wait_resp(SEND_STATUS, rca, 0x00, 0x01, 0x40);}
        write_reg(NOB, 0xffff);
        send_cmd_wait_resp(SET_BLOCKLEN, 0x00, 0x0200, 0x01, 0x40);
        if(buswidth==4-bit mode)
        {       send_cmd_wait_resp(APP_CMD, rca, 0x0, 0x01, 0x40);
                send_cmd_wait_resp(SET_BUS_WIDTH, 0x00, 0x02, 0x01, 0x40);
        }
        send_cmd_wait_resp(WRITE_DAT_UNTIL_STOP, addr_h, addr_l, 0x19,
0x40);
        while(!FIFO empty in STATUS is true);
        if(buswidth==4-bit mode)
        {       for(i=0;i<(nob*8);i++)
                {       while(!FIFO full in STATUS);// polling instead of
irq or dma req
                        for(j=0;j<32;j++)
                        {SDRAM_ADDR[i*32+j] = BUFFER_ACCESS;}
                        send_cmd_wait_resp(IO_RW_DIRECT, arg_h, arg_l,
0x5, 0x40);
                }
        }
        else    // 1-bit mode
        {       for(i=0;i<(nob*32);i++)
                {       while(!FIFO full in STATUS);// polling instead of
irq or dma req
                        for(j=0;j<8;j++)
                        {SDRAM_ADDR[i*8+j] = BUFFER_ACCESS;}
                        send_cmd_wait_resp(IO_RW_DIRECT, arg_h, arg_l,
0x5, 0x40);
                }
        }
        send_cmd_wait_resp(STOP_TRANS, 0x00, 0x00, 0x41, 0x40);
}
```

Stream Read

There is a stream oriented data transfer controlled by READ_DAT_UNTIL_STOP (CMD11). This command instructs the card to send its payload, starting at a specified address, until the host sends a STOP_TRANSMISSION command (CMD12). The stop command has an execution delay due to the serial command transmission. The data transfer stops after the end bit of the stop command. If the end of the memory range is reached while sending data and no stop command has been sent yet by the host, the contents of the further transferred payload is undefined.

The maximum clock frequency for stream read operation is given by the following formula:

max. speed = min. (TRAN_SPEED, (8 * 2 READ_BL_LEN - NSAC) / TAAC),

these parameters being defined in Chapter 5. If the host attempts to use a higher frequency, the card may not be able to sustain data transfer. If this happens, the card will set the UNDERRUN error bit in the status register, abort the transmission and wait in the data state for a stop command.

Example 29-9.

```
                stream_read(rca, nob, addr_h, addr_l, buswidth)
                {       send_cmd_wait_resp(SEND_STATUS, rca, 0x00, 0x01, 0x40);
                        while(!Ready for data in card status is true){
                                send_cmd_wait_resp(SEND_STATUS, rca, 0x00, 0x01, 0x40);}
                        write_reg(NOB, 0xffff);
                        send_cmd_wait_resp(SET_BLOCKLEN, 0x00, 0x0200, 0x01, 0x40);
                        if(buswidth==4-bit mode)
                        {       send_cmd_wait_resp(APP_CMD, rca, 0x0, 0x01, 0x40);
                                send_cmd_wait_resp(SET_BUS_WIDTH, 0x00, 0x02, 0x01, 0x40);
                        }
                        send_cmd_wait_resp(READ_DAT_UNTIL_STOP, addr_h, addr_l, 0x09, 0x40);
                        if(buswidth==4-bit mode)
                        {       for(i=0;i<(nob*8);i++)
                                {       while(!FIFO full in STATUS);// polling instead of
irq or dma req
                                        for(j=0;j<32;j++)
                                        {SDRAM_ADDR[i*32+j] = BUFFER_ACCESS;}
                                        send_cmd_wait_resp(IO_RW_DIRECT, arg_h, arg_l,
0x5, 0x40);
                                }
                        }
                        else    // 1-bit mode
                        {       for(i=0;i<(nob*32);i++)
                                {       while(!FIFO full in STATUS);// polling instead of
irq or dma req
                                        for(j=0;j<8;j++)
                                        {SDRAM_ADDR[i*8+j] = BUFFER_ACCESS;}
                                        send_cmd_wait_resp(IO_RW_DIRECT, arg_h, arg_l,
0x5, 0x40);
                                }
                        }
                        send_cmd_wait_resp(STOP_TRANS, 0x00, 0x00, 0x41, 0x40);
                }
```

29.5.3.3 Erase—Group Erase and Sector Erase (MMC Only)

It is desirable to erase many sectors simultaneously in order to enhance the data throughput. Identification of these sectors is accomplished with the TAG_* commands. Either an arbitrary set of sectors within a single erase group, or an arbitrary selection of erase groups may be erased at one time, but not both together. That is, the unit of measure for determining an erase is either a sector or an erase group. If a set of sectors must be erased, all selected sectors must lie within the same erase group. To facilitate selection, a first command with the starting address is followed by a second command with the final address, and all sectors (or groups) within this range will be selected for erase. After a range is selected, an individual sector (or group) within that range can be removed using the UNTAG command.

The host must adhere to the following command sequence: TAG_SECTOR_START, TAG_SECTOR_END, UNTAG_SECTOR, and ERASE.

A similar sequence can be applied for groups (TAG_ERASE_GROUP_START, TAG_ERASE_GROUP_END, UNTAG_ERASE_GROUP and ERASE). Up to 16 untag sector or group commands can be sent within one erase cycle. If an erase or tag/untag command is received out of sequence, the card shall set the ERASE_SEQ_ERROR bit in the status register and reset the whole

sequence. If an out of sequence command (except SEND_STATUS) is received, the card shall set the ERASE_RESET status bit in the status register, reset the erase sequence and execute the last command. If the erase range includes write protected sectors, they shall be left intact and only the non protected sectors shall be erased. The WP_ERASE_SKIP status bit in the status register shall be set. The address field in the tag commands is a sector or a group address in byte units. The card will ignore all LSB's below the group or sector size, respectively.

The number of untag commands (CMD34 and CMD37) which are used in a sequence is limited up to 16. As described above for block write, the card will indicate that an erase is in progress by holding DAT low. The actual erase time may be quite long, and the host may issue CMD7 to deselect the card.

29.5.3.4 Wide Bus Selection/Deselection

Wide Bus (4 bit bus width) operation mode may be selected/deselected using ACMD6. The default bus width after power up or GO_IDLE (CMD0) is 1 bit bus width. ACMD6 command is valid in 'trans state' only. That means the bus width may be changed only after a card was selected 9 (CMD7).

29.5.4 Protection Management

Three write protect methods are supported in the host for Cards, Card internal write protect (Cards's responsibility), Mechanical write protect switch (Host responsibility only) and Password protection card lock operation.

29.5.4.1 Card Internal Write Protection

Card data may be protected against either erase or write. The entire card may be permanently write protected by the manufacturer or content provider by setting the permanent or temporary write protect bits in the CSD. For cards which support write protection of groups of sectors by setting the WP_GRP_ENABLE bit in the CSD, portions of the data may be protected (in units of WP_GRP_SIZE sectors as specified in the CSD), and the write protection may be changed by the application. The SET_WRITE_PROT command sets the write protection of the addressed write-protect group, and the CLR_WRITE_PROT command clears the write protection of the addressed write-protect group.

The SEND_WRITE_PROT command is similar to a single block read command. The card shall send a data block containing 32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits. The address field in the write protect commands is a group address in byte units. The card will ignore all LSB's below the group size.

29.5.4.2 Mechanical Write Protect Switch

A mechanical sliding tablet on the side of the card will be used by the user to indicate that a given card is write protected or not. If the sliding tablet is positioned in such a way that the window is open that means the card is write protected. If the window is close the card is not write protected.

A proper, matched, switch on the socket side will indicate to the host that the card is write protected or not. It is the responsibility of the host to protect the card. The position of the write protect switch is un-known to the internal circuitry of the card.

29.5.4.3 Password Protect

The password protection feature enables the host to lock a card while providing a password, which later will be used for unlocking the card. The password and its size is kept in an 128-bit PWD and 8-bit PWD_LEN registers, respectively. These registers are non-volatile so that a power cycle will not erase them.

Locked cards respond to (and execute) all commands in the basic command class (class 0) and "lock card" command class. Thus the host is allowed to reset, initialize, select, query for status, etc., but not to access data on the card. If the password was previously set (the value of PWD_LEN is not 0) will be locked automatically after power on. Similar to the existing CSD and CID register write commands the lock/unlock command is available in "trans_state" only. This means that it does not include an address argument and the card must be selected before using it. The card lock/unlock command has the structure and bus transaction type of a regular single block write command. The transferred data block includes all the required information of the command (password setting mode, PWD itself, card lock/unlock etc.). The following table describes the structure of the command data block.

Table 29-27. Command Data Block Structure

Byte#	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Rsv	Rsv	Rsv	Rsv	ERASE	LOCK_ UNLOCK	CLR_PWD	SET_PWD
1	PWDS_LEN							
2	Password Data							
...								
PWDS_LEN +1								

Command Descriptions:

- **ERASE—**1 Defines Forced Erase Operation (all other bits shall be 0) and only the command byte is sent.
- **LOCK/UNLOCK—**1 = Locks the card. 0 = Unlock the card (note that it is valid to set this bit together with SET_PWD but it is not allowed to set it together with CLR_PWD).
- **CLR_PWD—**1 = Clears PWD.
- **SET_PWD—**1 = Set new password to PWD.
- **PWD_LEN—**Defines the following password length (in bytes).
- **PWD—**The password (new or currently used depending on the command).

The data block size shall be defined by the host before it sends the card lock/unlock command. This will allow different password sizes.

The following paragraphs define the various lock/unlock command sequences:

- Setting the Password:
  - Select a card (CMD7), if not previously selected already

— Define the block length (CMD16), given by the 8bit card lock/unlock mode, the 8 bits pass-word size (in bytes), and the number of bytes of the new password. In case that a password replacement is done, then the block size shall consider that both passwords, the old and the new one, are sent with the command.
  – Send Card Lock/Unlock command with the appropriate data block size on the data line including 16-bit CRC. The data block shall indicate the mode (SET_PWD), the length (PWD_LEN) and the password itself. In case that a password replacement is done, then the length value (PWD_LEN) shall include both passwords, the old and the new one, and the PWD field shall include the old password (currently used) followed by the new pass-word.
  – In case that the sent old password is not correct (not equal in size and content) then LOCK_UNLOCK_FAILED error bit will be set in the status register and the old password does not change. In case that PWD matches the sent old password then the given new password and its size will be saved in the PWD and PWD_LEN fields, respectively.

Note that the password length register (PWD_LEN) indicates if a password is currently set. When it equals 0 there is no password set. If the value of PWD_LEN is not equal to zero the card will lock itself after power up. It is possible to lock the card immediately in the current power session by setting the LOCK/UNLOCK bit (while setting the password) or sending additional command for card lock.

- Reset the Password:
  — Select a card (CMD7), if not previously selected already
  — Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit pass-word size (in bytes), and the number of bytes of the currently used password.
  — Send the card lock/unlock command with the appropriate data block size on the data line including 16-bit CRC. The data block shall indicate the mode CLR_PWD, the length (PWD_LEN) and the password (PWD) itself (LOCK/UNLOCK bit is don't care). If the PWD and PWD_LEN content match the sent password and its size, then the content of the PWD register is cleared and PWD_LEN is set to 0. If the password is not correct then the LOCK_UNLOCK_FAILED error bit will be set in the status register.
- Locking a card:
  — Select a card (CMD7), if not previously selected already
  — Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit pass-word size (in bytes), and the number of bytes of the currently used password.
  — Send the card lock/unlock command with the appropriate data block size on the data line including 16-bit CRC. The data block shall indicate the mode LOCK, the length (PWD_LEN) and the password (PWD) itself.

If the PWD content equals to the sent password then the card will be locked and the card-locked status bit will be set in the status register. If the password is not correct then LOCK_UNLOCK_FAILED error bit will be set in the status register.

Note that it is possible to set the password and to lock the card in the same sequence. In such case the host shall perform all the required steps for setting the password (as described above) including the bit LOCK set while the new password command is sent. If the password was previously set (PWD_LEN is not 0), then the card will be locked automatically after power on reset. An attempt to lock a locked card or to lock

a card that does not have a password will fail and the LOCK_UNLOCK_FAILED error bit will be set in the status register.

- Unlocking the card
  - Select a card (CMD7), if not previously selected already.
  - Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit pass-word size (in bytes), and the number of bytes of the currently used password.
  - Send the card lock/unlock command with the appropriate data block size on the data line including 16-bit CRC. The data block shall indicate the mode UNLOCK, the length (PWD_LEN) and the password (PWD) itself.

If the PWD content equals to the sent password then the card will be unlocked and the card-locked status bit will be cleared in the status register. If the password is not correct then the LOCK_UNLOCK_FAILED error bit will be set in the status register.

Note that the unlocking is done only for the current power session. As long as the PWD is not cleared the card will be locked automatically on the next power up. The only way to unlock the card is by clearing the password. An attempt to unlock an unlocked card will fail and LOCK_UNLOCK_FAILED error bit will be set in the status register.

- Forcing Erase:

  In case that the user forgot the password (the PWD content) it is possible to erase all the card data content along with the PWD content. This operation is called Forced Erase.
  - Select a card (CMD7), if not previously selected already.
  - Define the block length (CMD16) to 1 byte (8bit card lock/unlock command). Send the card lock/unlock command with the appropriate data block of one byte on the data line including 16-bit CRC. The data block shall indicate the mode ERASE (the ERASE bit shall be the only bit set).

If the ERASE bit is not the only bit in the data field then the LOCK_UNLOCK_FAILED error bit will be set in the status register and the erase request is rejected. If the command was accepted then ALL THE CARD CONTENT WILL BE ERASED including the PWD and PWD_LEN register content and the locked card will get unlocked.

An attempt to force erase on an unlocked card will fail and LOCK_UNLOCK_FAILED error bit will be set in the status register.

29.5.5 Card Status

The response format R1 contains a 32-bit field named card status. This field is intended to transmit the card's status information (which may be stored in a local status register) to the host. If not specified otherwise, the status entries are always related to the previous issued command. The semantics of this register is according to the CSD entry SPEC_VERS (see Chapter 5.3), indicating the version of the response formats (possibly used for later extensions).

Table below defines the different entries of the status. The type and clear condition fields in the table are abbreviated as follows:

Type:

- E: Error bit.
- S: Status bit.
- R: Detected and set for the actual command response.
- X: Detected and set during command execution. The host must poll the card by issuing the status command in order to read these bits.

Clear Condition:

- A: According to the card current state.
- B: Always related to the previous command. Reception of a valid command will clear it (with a delay of one command).
- C: Clear by read.

Table 29-28. Card Status Description

Bits	Identifier	Type	Value	Description	Clear Condition
31	OUT_OF_RAGE	E R	1 = Error	The command's argument was out of the allowed range for this card.	C
30	ADDRESS_ERROR	E R X	1 = Error	A misaligned address which did not match the block length was used in the command.	C
29	BLOCK_LEN_ERROR	E R	1 = Error	The transferred block length is not allowed for this, or the number of transferred bytes does not match the block length.	C
28	ERASE_SEQ_ERROR	E R	1 = Error	An error in the sequence of erase commands occurred.	C
27	ERASE_PARAM	E X	1 = Error	An invalid selection of sectors or groups for erase occurred.	C
26	WP_VIOLATION	E R X	1 = Protected	Attempt to program a write protected block.	C
25	CARD_IS_LOCKED	S X	1 = Card locked	When set, signals that the card is locked by the host.	A
24	LOCK_UNLOCK_FAILED	E R X	1 = Error	Set when a sequence or password error has been detected in lock/unlock card command or if there was an attempt to access a locked card.	C
23	COM_CRC_ERROR	E R	1 = Error	The CRC check of the previous command failed.	B
22	ILLEGAL_COMMAND	E R	1 = Error	Command not legal for the card state.	B
21	CARD_ECC_FAILED	E X	1 = failure	Card internal ECC was applied but failed to correct the data.	C
20	CC_ERROR	E R X	1 = Error	Internal card controller error.	C
19	ERROR	E R X	1 = Error	A general or an unknown error occurred during the operation.	C

Table 29-28. Card Status Description (continued)

Bits	Identifier	Type	Value	Description	Clear Condition
18	UNDERRUN	E X	1 = Error	The card could not sustain data transfer in stream read mode.	C
17	OVERRUN	E X	1 = Error	The card could not sustain data programming in stream write mode.	C
16	CID/CSD_OVERWRITE	E R X	1 = Error	Can be either one of the following errors:	C
15	WP_ERASE_SKIP	S X	1 = Protected	Only partial address space was erased due to existing write protected blocks.	C
14	CARD_ECC_DISABLED	S X	1 = disabled	The command has been executed without using the internal ECC.	A
13	ERASE_RESET	S R	1 = set	An erase sequence was cleared before executing because and out of erase sequence command was received.	C
12:9	CURRENT_STATE	S X	0 = idle 1 = ready 2 = ident 3 = stby 4 = tran 5 = data 6 = rcv 7 = prg 8 = dis 9–15 = rsv	The state of the card when receiving the command. If the command execution causes a state change, it will be visisble to the host in the response to the next command. The four bits are interpreted as binary coded number between 0 and 15.	B
8	READY_FOR_DATA	S X	1 = Ready	corresponds to buffer empty signalling on the bus	A
7:6	Reserved	–	–	–	–
5	APP_CMD	S R	1 = Enable	The card will expect ACMD, or indication that the command has been interpreted as ACMD	C
4:0	Reserved	–	–	–	–

29.5.6 SD Status

The SD status contains status bits that are related to the SD Card proprietary features and may be used for future application specific usage. The size of the SD status is one data block of 512bit. The content of this register is transmitted to the Host over the DAT bus along with 16-bit CRC. The SD Status is sent to the host over the DAT bus if ACMD13 is sent (CMD55 followed with CMD13). ACMD13 can be sent to a card only in ‘tran_state’ (card is selected). SD Status structure is described in below.

The same abbreviation for *type* and *clear condition* were used as for the Card Status above.

Table 29-29. SD Status Structure

Bits	Identifier	Type	Value	Description	Clear Condition
511:510	DAT_BUS_WIDTH	S R	00= 1(default) 01= Reserved 10= 4 bit width 11= Reserved	Shows the currently defined data bus width that was defined by SET_BUS_WIDTH command.	A
509	SECURED_MODE	S R	0 = Not in the mode 1 = In the mode	Card is in Secured Mode of operation (refer to “SD Security Specification”).	A
508:496	Reserved				
495:480	SD_CARD_TYPE	S R	All 0 = SD Memory Cards (as defined in Physical Spec Ver.1)	An error in the sequence of erase commands occurred.	A
479:448	SIZE_OF_PROTECTED_AREA	S R	Size of protected area (in units of MULT*BLOCK_LEN refer to CSD register)	An invalid selection of sectors or groups for erase occurred.	A
447:312	Reserved				
311:0	Reserved for manufacturer				

29.5.7 SDIO

I/O access differs from memory in that the registers can be written and read individually and directly without a FAT file structure or the concept of blocks (although block access is supported). These registers allow access to the IO data, control of the IO function, and report on status or transfer I/O data to and from the host.

Each SDIO card may have from 1 to 7 functions plus one memory function built into it. A function is a self contained I/O device. I/O functions may be identical or completely different from each other. All I/O functions are organized as a collection of registers, and there is a maximum of 131,072 registers possible for each I/O function.

29.5.7.1 SDIO Interrupts

In order to allow the SDIO card to interrupt the host, and interrupt function is added to a pin on the SD interface. Pin number 8 which is used as DAT[1] when operating in the 4 bit SD mode is used to signal the card’s interrupt to the host. The use of interrupt is optional for each card or function within a card. The SDIO interrupt is “level sensitive”, that is, the interrupt line must be held active (low) until it is either recognized and acted upon by the host or de-asserted due to the end of the Interrupt Period. Once the host has serviced the interrupt, it is cleared via an IO write to the appropriate bit in the CCCR. The interrupt output of all SDIO cards is active low. This host controller provides pull-up resistors on all data lines DAT[3:0].

As Pin 8 of the card is shared between the IRQ and DAT[1] use in the 4 bit SD mode, and interrupt shall only be sent by the card and recognized by the host during a specific time. The time that a low on Pin 8 will be recognized as an interrupt is defined as the Interrupt Period.

The host here will only sample the level of Pin 8 (DAT[1]/IRQ) into the interrupt detector during the Interrupt Period. At all other times, the host will ignore the level on Pin 8. Note that the INterrupt Period is applicable for both memory and IO operations. The definition of the Interrupt Period is different for operations with single block and multiple block data transfer.

29.5.7.2 SDIO Suspend/Resume

Within a multi-function SDIO or a Combo (Mix IO and Memory) card, there are multiple devices (I/O and memory) that must share access to the SD bus. In order to allow the sharing of access to the host among multiple devices. SDIO and combo cards can implement the optional concept of suspend/resume. I a card supports suspend/resume, the host may temporarily halt a data transfer operation to one function or memory (suspend) in order to free the bus for a higher priority transfer to a different function of memory. Once this higher-priority transfer is complete, the original transfer is re-started where it left off (resume). The host controller here is supported by all IO functions except zero, and the memory of a combo card, and can suspend multiple transactions and resume them in any order desired. IO function zero does not support suspend/resume.

The procedure used to perform the Suspend/Resume operation on the SD bus is:

1. The host determines which function currently used the DAT[] line(s).
2. The host requests the lower priority or slower transaction to suspend.
3. The host checks for the transaction suspension to complete.
4. The host begins the higher priority transaction.
5. The host waits for the completion of the higher priority transaction.
6. The host restores the suspended transaction.

29.5.7.3 SDIO Read Wait

The optional Read Wait (RW) operation is defined only for the SD 1-bit and 4-bit modes. The read wait operation allows a host to signal a card that it is doing a read multiple (CMD53) operation to temporarily stall the data transfer while allowing the host to send commands to any function within the SDIO device. To determine if a card supports the Read Wait protocol, the host must test capability bits in CCCR. THe timing for Read Wait is based on the Interrupt Period.

Example 29-10.

```
block_read_with_read_wait_without_DMA(rca, nob, addr_h, addr_l, buswidth)
{       send_cmd_wait_resp(SEND_STATUS, rca, 0x00, 0x01, 0x40);
        while(!Ready for data in card status is true){
                send_cmd_wait_resp(SEND_STATUS, rca, 0x00, 0x01, 0x40);}
        write_reg(NOB, <nob>);
        send_cmd_wait_resp(SET_BLOCKLEN, 0x00, 0x0200, 0x01, 0x40);
        if(buswidth==4-bit mode)
        {       send_cmd_wait_resp(APP_CMD, rca, 0x0, 0x01, 0x40);
                send_cmd_wait_resp(SET_BUS_WIDTH, 0x00, 0x02, 0x01, 0x40);
```

```
                        }
                        write_reg(CMD_DAT_CONT, set bit 10); // Enable Read Wait at the
following block boundary
                        if(nob==1)
                                send_cmd_wait_resp(READ_SINGLE_BLOCK, addr_h, addr_l,
0x09, 0x40);
                        else
                                send_cmd_wait_resp(READ_MULTIPLE_BLOCK, addr_h, addr_l,
0x09, 0x40);
                        if(buswidth==4-bit mode)
                        {       for(i=0;i<(nob*8);i++)
                                {       while(!FIFO full in STATUS);// polling instead of
irq or dma req
                                        for(j=0;j<32;j++)
                                        {SDRAM_ADDR[i*32+j] = BUFFER_ACCESS;}
                                        // Enable CmdResLongOff when start
                                        send_cmd_wait_resp(IO_RW_DIRECT, arg_h, arg_l,
0x105, 0x40);
                                        // Disable CmdResLongOff when finish
                                        write_reg(CMD_DAT_CONT, set bit 11 to stop Read
Wait);
                                }
                        }
                        else    // 1-bit mode
                        {       for(i=0;i<(nob*32);i++)
                                {       while(!FIFO full in STATUS);// polling instead of
irq or dma req
                                        for(j=0;j<8;j++)
                                        {SDRAM_ADDR[i*8+j] = BUFFER_ACCESS;}
                                        // Enable CmdResLongOff when start
                                        send_cmd_wait_resp(IO_RW_DIRECT, arg_h, arg_l,
0x105, 0x40);
                                        // Disable CmdResLongOff when finish
                                        write_reg(CMD_DAT_CONT, set bit 11 to stop Read
Wait);
                                }
                        }
                        while(!Data Transfer Done in STATUS true);
                        while(!card bus is stop);
                        if(nob > 1) {
                                send_cmd_wait_resp(STOP_TRANS, 0x00, 0x00, 0x41, 0x40);
                        }
                }
```

29.5.8 Application Specified Command Handling

The MultiMediaCard/SD system is designed to provide a standard interface for a variety applications types. In this environment it is anticipated that there will be a need for specific customers/applications features. To enable a common way of implementing these features, two types of generic commands are defined in the standard: Application Specific Command, ACMD, and General Command, GEN_CMD.

GEN_CMD, this command, when received by the card, will cause the card to interpret the following command as an application specific command, ACMD. The ACMD has the same structure as of regular

MultiMediaCard standard commands and it may have the same CMD number. The card will recognize it as ACMD by the fact that it appears after APP_CMD.

The only effect of the APP_CMD is that if the command index of the, immediately, following command has an ACMD overloading, the non standard version will used. If, as an example, a card has a definition for ACMD13 but not for ACMD7 then, if received immediately after APP_CMD command, Command 13 will be interpreted as the non standard ACMD13 but, command 7 as the standard CMD7.

In order to use one of the manufacturer specific ACMD's the host will:

- Send APP_CMD. The response will have the APP_CMD bit (new status bit) set signaling to the host that ACMD is now expected.
- Send the required ACMD. The response will have the APP_CMD bit set, indicating that the accepted command was interpreted as ACMD. If a non-ACMD is sent then it will be respected by the card as normal MultiMediaCard command and the APP_CMD bit in the Card Status stays clear.

If a non valid command is sent (neither ACMD nor CMD) then it will be handled as a standard MultiMediaCard illegal command error.

The bus transaction of the GEN_CMD is the same as the single block read or write commands (CMD24 or CMD17). The difference is that the argument denotes the direction of the data transfer (rather than the address) and the data block is not a memory payload data but has a vendor specific format and meaning.

The card shall be selected ('tran_state') before sending CMD56. The data block size is the BLOCK_LEN that was defined with CMD16. The response to CMD56 will be R1b (card status + busy indication).

29.6 Commands for MMC/SD

Table 29-30. MMC/SD Command Table

Command Index	Type	Argument	Resp	Abbreviation	Description
CMD0	bc	[31:0] stuff bits	–	GO_IDLE_STATE	Resets all MMC and SD memory cards to idle state.
CMD1	bcr	[31:0] OCR without busy	R3	SEND_OP_COND	Asks all MMC and SD Memory cards in idle state to send their operation conditions register contents in the response on the CMD line.
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	Asks all cards to send their CID numbers on the CMD line
CMD3	ac	[31:6] RCA [15:0] stuff bits	R1 R6(SDIO)	SET_RELATIVE_ADDR	Assigns relative address to the card.
CMD4	bc	[31:0] DSR [15:0] stuff bits	–	SET_DSR	Programs the DSR of all cards.
CMD5	bc	[31:0] OCR without busy	R4	IO_SEND_OP_COND	Asks all SDIO cards in idle state to send their operation conditions register contents in the response on the CMD line.
CMD6	**Reserved**				

Table 29-30. MMC/SD Command Table (continued)

Command Index	Type	Argument	Resp	Abbreviation	Description
CMD7	ac	[31:6] RCA [15:0] stuff bits	R1b	SELECT/DESELECT_CARD	Command toggles a card between the stand-by and transfer states or between the programming and disconnect states. In both cases the card is selected by its own relative address and gets deselected by any other address; address 0 deselects all.
CMD8	**Reserved**				
CMD9	ac	[31:6] RCA [15:0] stuff bits	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:6] RCA [15:0] stuff bits	R2	SEND_CID	Addressed card sends its card-identification (CID) on the CMD line.
CMD11	adtc	[31:0] data address	R1	READ_DAT_UNTIL_STOP	Reads data stream from the card, starting at the given address, until a STOP_TRANSMISSION follows.
CMD12	ac	[31:0] stuff bits	R1b	STOP_TRANSMISSION	Forces the card to stop transmission.
CMD13	ac	[31:6] RCA [15:0] stuff bits	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	**Reserved**				
CMD15	ac	[31:6] RCA [15:0] stuff bits	–	GO_INACTIVE_STATE	Sets the card to inactive state in order to protect the card stack against communication breakdowns.
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	Sets the block length (in bytes) for all following block commands (read and write). Default block length is specified in the CSD.
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously transfers data blocks from card to host until interrupted by a stop command.
CMD19	**Reserved**				
CMD20	adtc	[31:0] data address	R1	WRITE_DAT_UNTIL_STOP	Writes data stream from the host, starting at the given address, until a STOP_TRANSMISION follows.
CMD21–23	**Reserved**				
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.
CMD25	adtc	[31:0] data address	R1	WRITE_MULTIPLE_BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows.

Table 29-30. MMC/SD Command Table (continued)

Command Index	Type	Argument	Resp	Abbreviation	Description
CMD26	adtc	[31:0] stuff bits	R1	PROGRAM_CID	Programming of the card identification register. This command shall be issued only once per card. The card contains hardware to prevent this operation after the first programming. Normally this command is reserved for the manufacturer.
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.
CMD28	ac	[31:0] data address	R1b	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29	ac	[31:0] data address	R1b	CLR_WRITE_PROT	If the card provides write protection features, this command clears the write protection bit of the addressed group.
CMD30	adtc	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card provides write protection features, this command asks the card to send the status of the write protection bits.
CMD31	**Reserved**				
CMD32	ac	[31:0] data address	R1	TAG_SECTOR_START	Sets the address of the first sector of the erase group.
CMD33	ac	[31:0] data address	R1	TAG_SECTOR_END	Sets the address of the last sector in a continuous range within the selected of a single sector to be selected for erase.
CMD34	ac	[31:0] data address	R1	UNTAG_SECTOR	Removes one previously selected sector from the erase selection.
CMD35	ac	[31:0] data address	R1	TAG_ERASE_GROUP_START	Sets the address of the first erase group within a range to be selected for erase.
CMD36	ac	[31:0] data address	R1	TAG_ERASE_GROUP_END	Sets the address of the last erase group within a continuous range to be selected for erase.
CMD37	ac	[31:0] data address	R1	UNTAG_ERASE_GROUP	Removes one previously selected erase group from the erase selection.
CMD38	ac	[31:0] stuff bits	R1b	ERASE	Erase all previously selected sectors.
CMD39	ac	[31:0] RCA [15] register write flag [14:8] register address [7:0] register data	R4	FAST_IO	Used to write and read 8-bit (register) data fields. The command address a card and a register and provides the data for writing if the write flag is set. The R4 response contains data read from the address register. This command accesses application dependent registers which are not defined in MMC standard.
CMD40	bcr	[31:0] stuff bits	R5	GO_IRQ_STATE	Sets the system into interrupt mode.
CMD41	**Reserved**				

Table 29-30. MMC/SD Command Table (continued)

Command Index	Type	Argument	Resp	Abbreviation	Description
CDM42	adtc	[31:0] stuff bits	R1b	LOCK_UNLOCK	Used to set/reset the password or lock/unlock the card. The size of the data block is set by the SET_BLOCK_LEN command.
CMD43–51	**Reserved**				
CMD52	–	[31:0] stuff bits	R5	IO_RW_DIRECT	Access a single register within the total 128k of register space in any IO function.
CMD53	–	[31:0] stuff bits	R5	IO_RW_EXTENDED	Access a multiple I/O register with a single command, it allows the reading or writing of a large number of IO registers.
CMD54	**Reserved**				
CMD55	ac	[31:16] RCA [15:0] stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather that a standard command.
CMD56	adtc	[31:1] stuff bits [0]: RD/WR	R1b	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose / application specific commands. The size of the data block shall be set by the SET_BLOCK_LEN command.
CMD57–63	**Reserved**				
ACMDs shall be preceded with APP_CMD command. (Command listed below are used for SD only, other SD commands not listed below are not supported in this module)					
ACMD6	ac	[31:2] stuff bits [1:0] bus width	R1	SET_BUS_WIDTH	Defines the data bus width (00=1bit or 10=4bit bus) to be used for data transfer. The allowed data bus widths are given in SCR register.
ACMD13	adtc	[31:0] stuff bits	R1	SD_STATUS	Send the SD Memory Card status.
ACMD22	adtc	[31:0] stuff bits	R1	SEND_NUM_WR_SECTORS	Send the number of the written (without errors) sectors. Responds with 32bit+CRC data block.
ACMD23	–		–	SET_WR_BLK_ERASE_COUNT	–
ACMD41	bcr	[31:0] OCR	R3	SD_APP_OP_COND	Asks the accessed card to send its operating condition register (OCR) content in the response on the CMD line.
ACMD42	–	–	–	SET_CLR_CARD_DETECT	–
ACMD51	adtc	[31:0] stuff bits	R1	SEND_SCR	Reads the SD Configuration register (SCR)

Chapter 30
Digital Audio Mux (AUDMUX)

The Digital Audio Mux (AUDMUX) provides a programmable interconnect fabric for voice, audio, and synchronous data routing between i.MX21 SSI modules and external SSI, audio and voice codecs. With the AUDMUX, resources do not need to be hard-wired and can be effectively shared in different configurations. The AUDMUX interconnections allow multiple simultaneous separate audio/voice/data flows between the ports in a point-to-point or point-to-multipoint configuration.

The Digital Audio Mux offers the following features:

- 3 host interfaces (2 internal and 1 external)
- 3 peripheral interfaces (All external)
- Full 6-wire SSI interfaces for asynchronous receive and transmit
- Configurable 4-wire, synchronous or 6-wire asynchronous Rx and Tx external host and peripheral interfaces
- Independent Frame sync and clock direction selection for host or peripheral.Clock direction selection to function as master of the flow
- Each host interface can be connected to any other host or peripheral interface in a point-to-point or point-to-multipoint (network mode)
- Transmit and Receive Data switching to support external network mode

Figure 30-1 shows the block diagram of the AUDMUX. In the illustration, on the left are the internal interfaces and on the right, the external interfaces. Port1 and Port2 are internally connected to SSI-1 and SSI-2 respectively and Port3 has special muxes to connect to an external SSI such as a synchronous audio port (SAP) found on a baseband modem. Ports 4–6 are identical and can be connected to any 4-wire or 6-wire SSI, voice, I^2S or AC97 codec. Ports 1–3 are also known as host ports and Ports 4–6 as peripheral ports.

Port1–Port6 have configurable four or six wire interface. When configured as a six-wire interface, the additional RFS and RCLK signals of the interface enable the SSIs to be used in asynchronous mode with separate receive and transmit clocks. In this mode, a device at one port can be connected to two ports (internal or external) configured as input only (simplex) and output only (simplex). Ports 1–3 have muxing arrangements to support internal network mode. Ports 3–6 have a Tx/Rx switch to support external network mode. The Tx/Rx switch enables the Da and Db lines to be swapped so that more than one master connected to any of Ports 1–3 can communicate to more than one slave externally attached at the external ports.

Bit Clock selection direction enables each port to be configured as a master or slave in the flow.

Possible scenarios are:

1. SSI1 (Internal host port) drives voice codec and BT (on External Peripheral port 6) and the Bottom Connector (on External Peripheral port 5) simultaneously using network mode. SSI1 is the master.
2. SAP (External audio port from Baseband) drives voice codec and BT (on port 6) and the Bottom Connector (on port 5) simultaneously using network mode. SAP is the master.

NOTE

The first scenario supports External Network mode when SSI1 provides the corresponding Output Enables for TxData, and the Slave devices receiving the data must be configured to only receive in their corresponding time slot.

In the second case, any slave devices attached locally to the SAP in network mode must be disabled to access slave devices on the other ports (for example, SSI, voice codec, and/or BT). Frame Sync and Bit Clock selections enable each port to be configured as a master or slave in the flow.

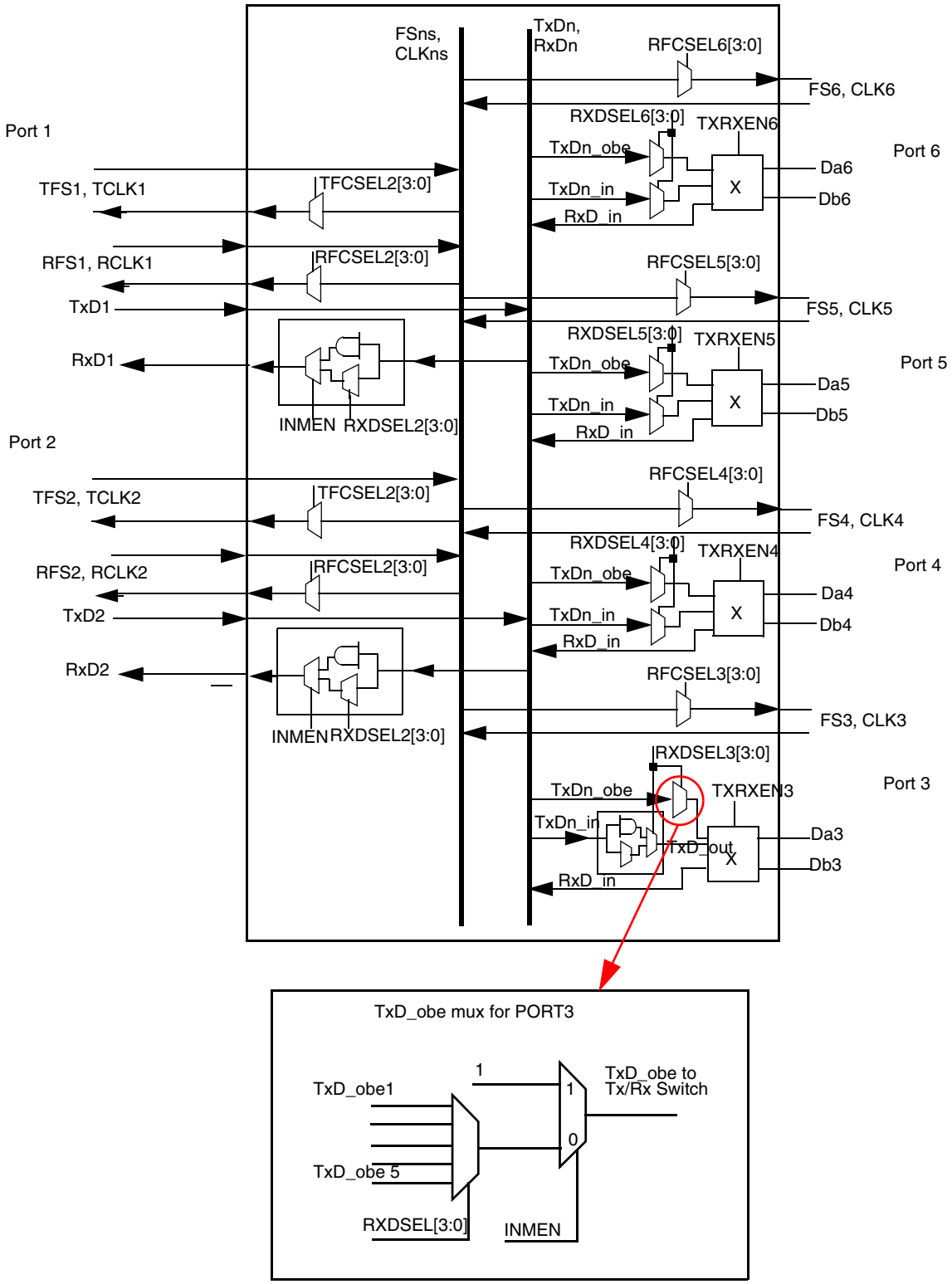


Figure 30-1. AUDMUX Block Diagram

30.1 Internal Network Mode

Figure 30-2 shows the internal network mode selection logic. Network mode is where a master SSI is connected to more than one slave SSI device and communication occurs on a timeslotted frame. Though network mode allows communication between master-slave and slave-slave, the internal network mode supports only master-slave network mode.

In internal network mode (INMEN=1), the output of the AND gate is routed to the output of the port and to the RxD signal of the SSI. The INMMASK bit vector selects the transmit signals of the ports that are to be connected in network mode. The transmit signals (TxD_in from SSI and RxD_in from external ports) are ANDed together to form the output. In network mode, only one device can be transmitting in its predesignated timeslot and all other transmit signals will remain high (tristated and pulled-up), hence non-active signals in the selection will be high and do not influence the output of the AND gate.

In normal mode (INMEN=0), the SSI is connected in point to point (as a master or slave) and the RXDSEL[2:0] settings select the transmit signal from the other ports. Internal network mode can be used with external network as long as slave-only devices are attached in external network mode at a port or in. Internal network mode can also be used with external network mode if all slave devices connected to a master in external network mode are disabled.

Figure 30-2 shows the connections for Port1.

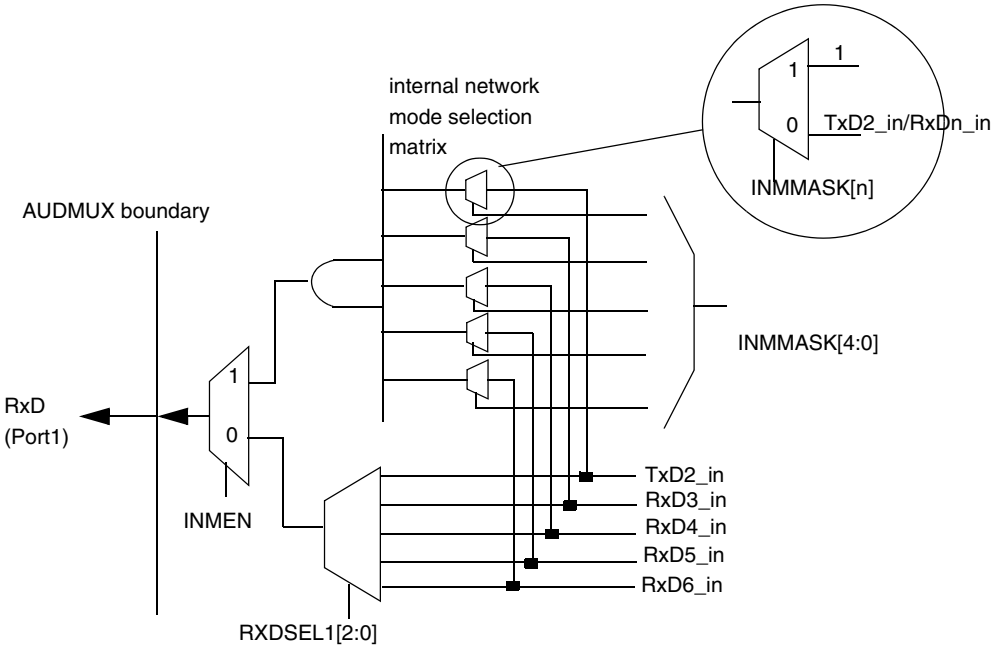


Figure 30-2. Internal Network Mode

30.2 Tx/Rx Switch and External Network Mode

External network mode is the traditional network mode connection. It is called external network mode to differentiate from the internal network mode. In external network mode, devices are connected to the external ports in a star or multidrop configuration.

In network mode, there can be only one master (frame sync and clock source) and the other devices are configured in normal slave mode or network slave mode. Unlike internal network mode, in external network mode both master-slave and slave-slave communication can take place. Codec devices transmit on a single timeslot and SSIs in network master (for example SAP) or slave mode (SSI-2) can process more than one timeslot of data.

Figure 30-3 shows the Tx/Rx switch.TxD_obe is the output buffer enable signal and TxD_out is the data transmit signal from the internal SSI. The RxD_in signal is the receive data signal going towards the RXDSEL and TXDSEL muxes of the SSI ports and external ports.

In normal mode and network slave mode, TXRXEN is disabled (TXRXEN=0) and TxD_out is routed to Da (Da_out) and Db (Db_in) is routed to RxD_in. In normal mode, the output buffer enable, Da_obe is always enabled (asserted) and TxD_out is routed to Da_out. In network mode, the TxD_obe signal is enabled during the SSI's timeslot(s) and the Da output is tristated in other timeslots.

In network mode (SSIx as master), the Tx/Rx switch is enabled (TXRXEN=1) and TxD_out is routed to Db_out and Da_in is routed to RxD_in. The TxD_obe signal is enabled during the SSI's timeslot(s) and the Db output is tristated in other timeslots.

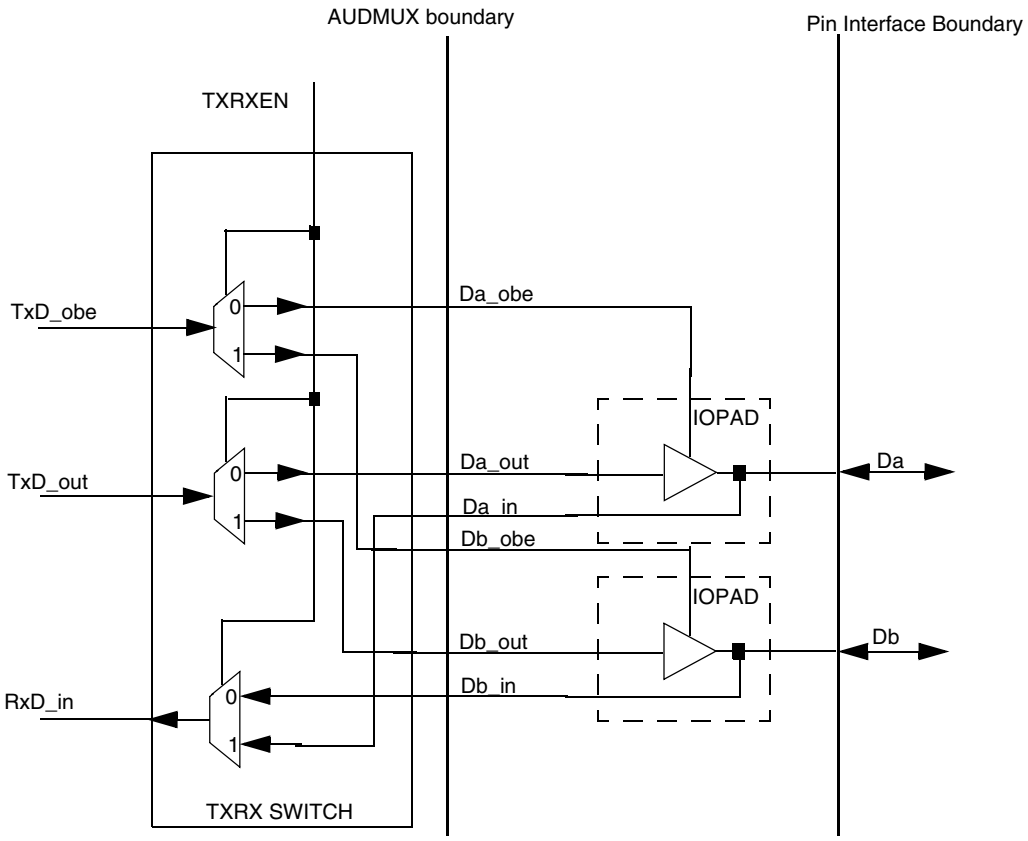


Figure 30-3. Tx/Rx Switch

30.3 Frame Sync and Clocks

The routing of frame syncs and interface clocks are shown in Figure 30-4.

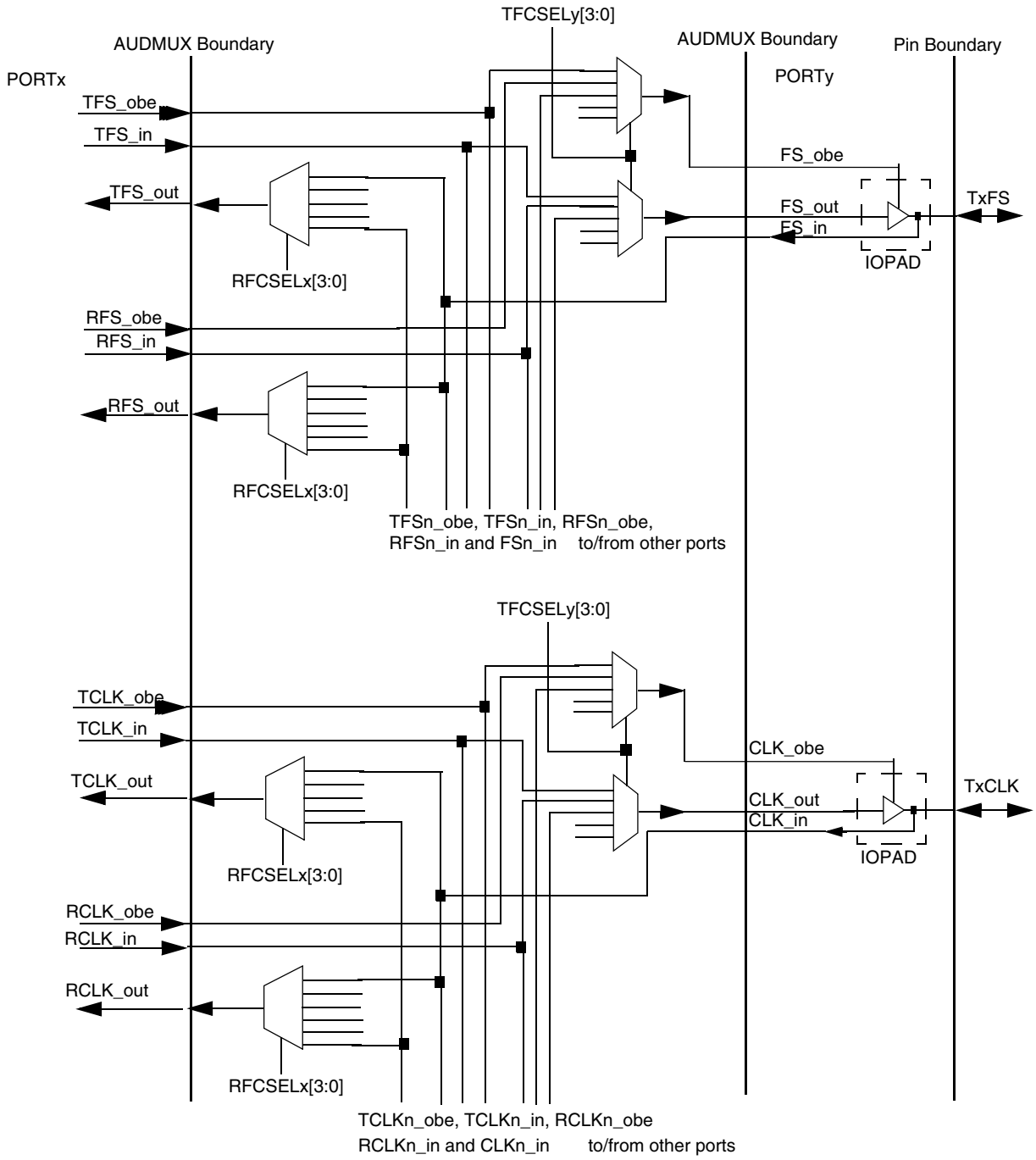


Figure 30-4. Frame Sync and Clock Routing when Peripheral Port is 4-Wire

30.4 Synchronous Mode (4-Wire Interface)

In Synchronous mode the port will have a 4-wire interface—that is, RXD,TXD,TxCLK,TxFS.The Receive clock and the receive frame sync will be the same as Transmit clock (TxCLK) and Transmit frame sync (TxFS), respectively.

As shown in Figure 30-4, PORTx signals can be routed to PORTy, showing a 6-wire to 4-wire port connectivity.

TFS_in, RFS_in, TCLK_in and RCLK_in are input Frame Sync and Bit Clocks from the SSI with their corresponding output buffer enable signals (_obe). TFS_out, RFS_out, TCLK_out and RCLK_out are the Frame Sync and Bit Clocks that are transmitted to the SSI from the other ports.

TFS_out and TCLK_out are selected by the TFCSEL mux settings and RFS_out and RCLK_out are selected by the RFCSEL mux settings. Similarly, in the external direction, the TFCSEL selects the FS_obe and FS_out signals. In this mode RFCSEL is not used.

30.5 Asynchronous Mode (6-Wire Interface)

In Asynchronous mode the port will have a 6-wire interface—that is, RXD, TXD, TxCLK, TxFS, RxCLK, RxFS. There will be additional Receive clock (RxCLK) and the frame sync (RxFS) pins as compared to the Synchronous or 4-wire interface.

Refer to Figure 30-5 and Figure 30-6, PORTx signals can be routed to PORTy, depicting a 6-wire to 6-wire port connectivity.

TFS_in, RFS_in, TCLK_in and RCLK_in are input Frame Sync and Bit Clocks from the SSI (PORTx) with their corresponding output buffer enable signals (_obe). TFS_out, RFS_out, TCLK_out and RCLK_out are the Frame Sync and Bit Clocks that are transmitted to the SSI from the other ports.

TFS_out and TCLK_out are selected by the TFCSEL mux settings and RFS_out and RCLK_out are selected by the RFCSEL mux settings. Similarly, in the external direction, the TFCSEL selects the TxFS_obe,TxFS_out and TxCLK_obe,TxClk_out signals. The RFCSEL selects the RxFS_obe and RxFS_out and RxCLK_obe, RxCLK_out.

NOTE

Noticed that because FS_in and CLK_in from external interfaces are also routed to the TFCSEL muxes of the external ports, these signals do not have corresponding buffer enable signals. Consequently, their corresponding inputs to the TFCSEL mux of the external ports must be tied high.

30.6 SSI to Peripheral Connection

The Figure 30-7 shows the data path interconnections between an internal SSI port and a peripheral port. TxD_obe is the buffer enable signal from the SSI, TxD_in, the input transmit data from the SSI and RxD_out, the receive data output from the AUDMXUX to the SSI.

TXDSEL[2:0] of the peripheral port, selects the buffer enable signal (TxD_obe) and transmit data output (TxD_out) signal from the TxD_obe and TxD_in and RxD_in signals. TXDSEL[2:0] is a common signal to both selection muxes.

NOTE

Because RxD_in signals from external interfaces do not have their buffer enable signals, their corresponding buffer enable signals into the selection mux should be tied to high. This will ensure that selection of RxD_in as TxD_out will also drive the TxD_obe output high.

Transmit Data from the SSI goes into the TXDSEL data mux and comes out as TxD_out and is routed to Da_out when TXRXEN is disabled and to Db_out when TXRXEN is enabled. Similarly, Db_in is routed to RxD_in when TXRXEN is disabled and Da_in is routed to RxD_in when TXRXEN is enabled. If The routing of frame syncs are shown in Figure 30-5 and the routing of interface clocks are shown in Figure 30-6.

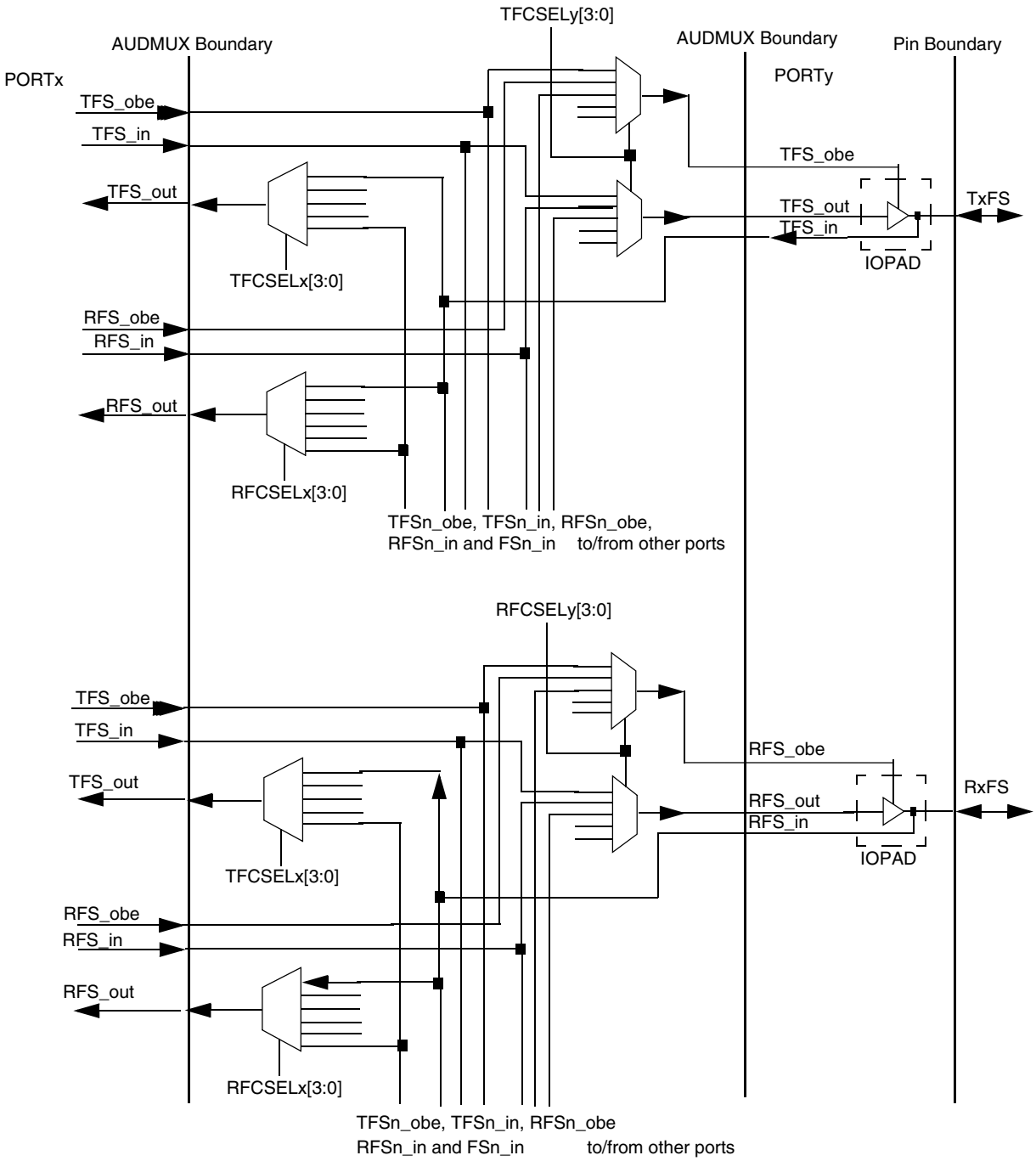


Figure 30-5. Frame Sync Routing when Peripheral Port is 6-Wired

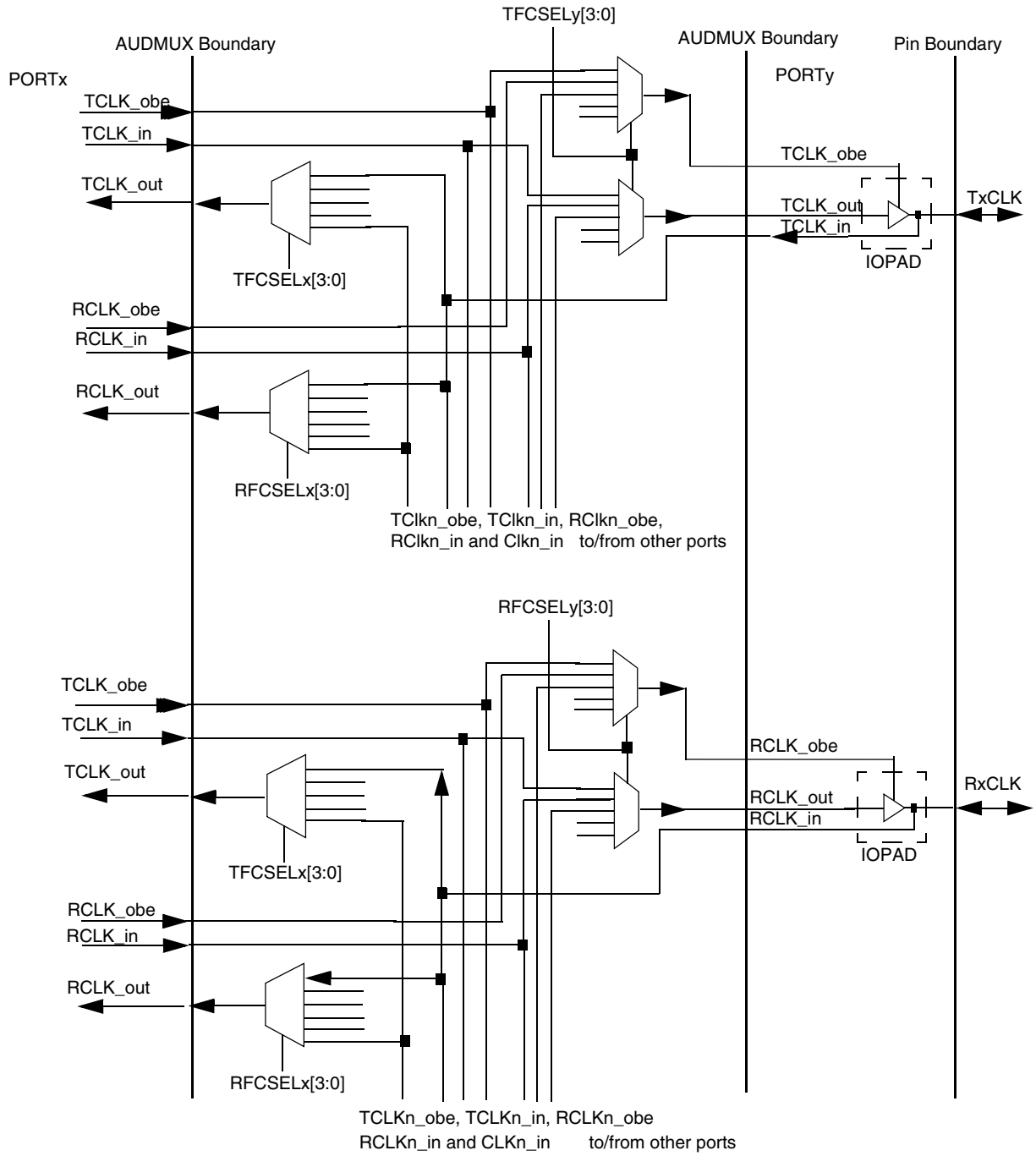


Figure 30-6. Clock Routing when Peripheral Port is 6-Wire

If internal network mode is disabled, then RXDSEL selects the RxD_in which is then output from the AUDMUX to the SSI. When internal network mode is selected, the RxD_in is ANDED with other TxD_in and RxD_in signals from other ports before output as RxD_out to the SSI.

If there are more than one device attached to the external port at Da and Db interfaces and one of the devices is a network master, then two conditions have to be noted:

a) When the external master is enabled in network mode, then the SSI must be configured as slave (normal or network mode). No Tx/Rx switching is required.

b) When the external master is disabled and the SSI and other slave devices require to communicate, then the SSI must be configured as network mode master and the Tx/Rx switch must be enabled (TXRXEN=1). This ensures that the transmit and receive paths are connected appropriately.

To communicate with more than one port, internal network mode must be enabled at the SSI port. In internal network mode, it is possible to communicate with any device attached to the other ports. Internal network mode must be enabled at the port that is the SSI network mode master.

30.7 SSI to SAP

The Figure 30-8 shows the detailed interconnection of a SSI port to a SAP port. The SSI and SAP port can act as masters or slaves and be configured in normal or network mode. The SAP port can communicate with more than one external device attached to its own interface in external network mode and additionally with one device attached to another AUDMUX port.

If the SAP must communicate with more than one port, then the internal network mode must be enabled.

30.8 Peripheral Port to Peripheral Port

Peripherals attached to the AUDMUX can communicate with each other in 2 ways:

a) One peripheral acts is configured as master and which sources the clock and/or the frame sync and the other peripheral is configured as slave.

b) Both peripherals are configured as slaves but with data routing established from external port to external port with one additional port configured as master (SSI or SAP) which sources the frame sync and clock to the two ports.

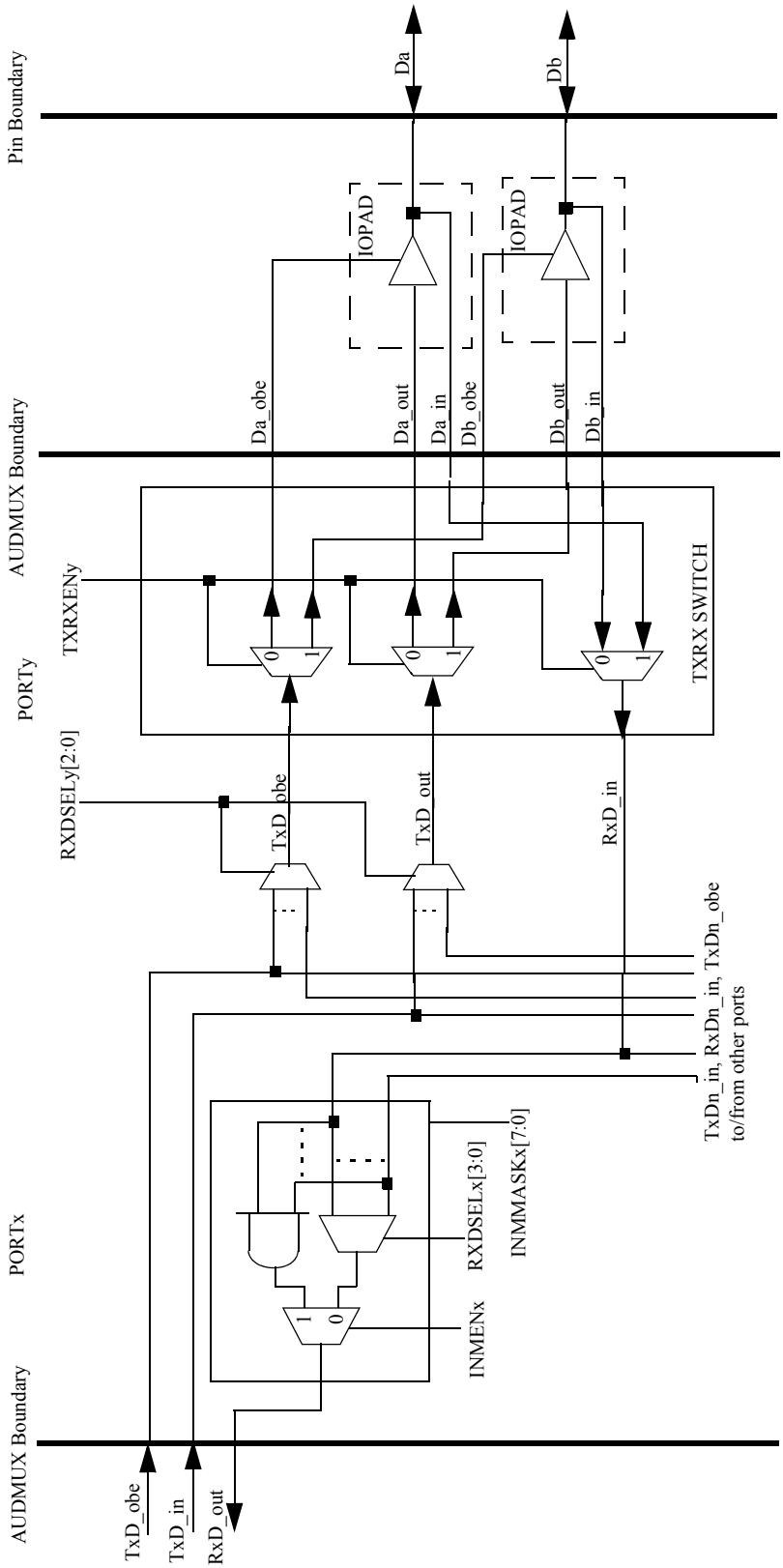


Figure 30-7. SSI to Peripheral Port Interconnection

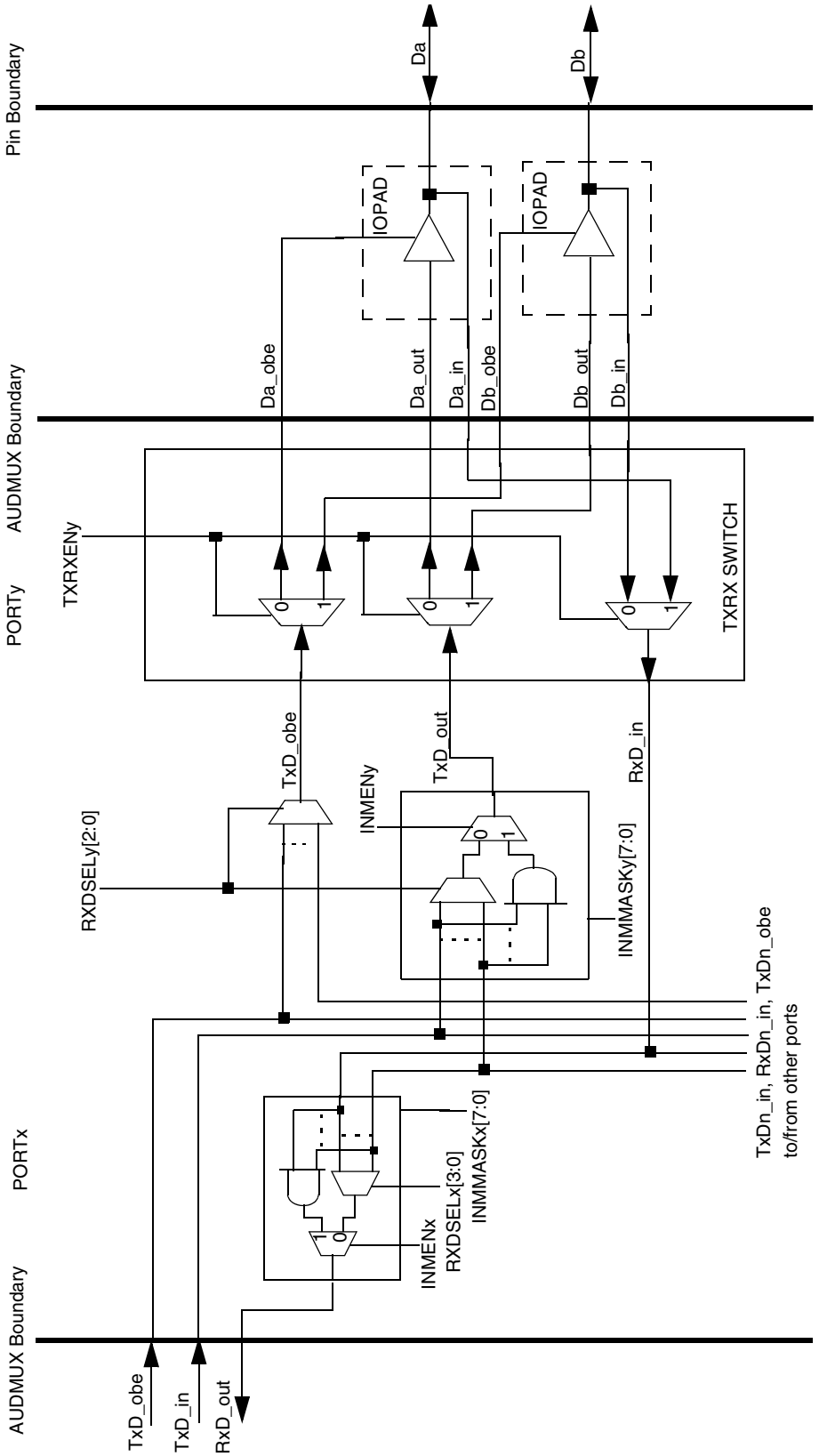


Figure 30-8. SSI to SAP Interconnection

30.9 Programming Model

There is one configuration register per host port and peripheral ports and there are a total of 6 registers. Table 30-1 shows the control register and address mapping for AUDMUX. The base address is as follows:

- 0x1001 6000

Table 30-1. AUDMUX Register Summary

Name		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
HPCR1 $BASE_ADDR	R	TFS DIR	TCLKDIR	TFCSEL[3:0]				RFS DIR	RCLKDIR	RFCSEL[3:0]				0	0	0	0
	W																
	R	RXDSEL[2:0]			SYN	0	0	0	INMEN	INMMASK[7:0]							
	W																
HPCR2 $BASE_ADDR+$04	R	TFSDIR	TCLKDIR	TFCSEL[3:0]				RFSDIR	RCLKDIR	RFCSEL[3:0]				0	0	0	0
	W																
	R	RXDSEL[2:0]			SYN	0	0	0	INMEN	INMMASK[7:0]							
	W																
HPCR3 $BASE_ADDR+$08	R	TFSDIR	TCLKDIR	TFCSEL[3:0]				RFSDIR	RCLKDIR	RFCSEL[3:0]				0	0	0	0
	W																
	R	RXDSEL[2:0]			SYN	0	TXRXEN	0	INMEN	INMMASK[7:0]							
	W																
PPCR1 $BASE_ADDR+$10	R	TFSDIR	TCLKDIR	TFCSEL[3:0]				RFSDIR	RCLKDIR	RFCSEL[3:0]				0	0	0	0
	W																
	R	RXDSEL[2:0]			SYN	0	TXRXEN	0	0	0	0	0	0	0	0	0	0
	W																
PPCR2 $BASE_ADDR+$14	R	TFSDIR	TCLKDIR	TFCSEL[3:0]				RFSDIR	RCLKDIR	RFCSEL[3:0]				0	0	0	0
	W																
	R	RXDSEL[2:0]			SYN	0	TXRXEN	0	0	0	0	0	0	0	0	0	0
	W																
PPCR3 $BASE_ADDR+$1C	R	TFSDIR	TCLKDIR	TFCSEL[3:0]				RFSDIR	RCLKDIR	RFCSEL[3:0]				0	0	0	0
	W																
	R	RXDSEL[2:0]			SYN	0	TXRXEN	0	0	0	0	0	0	0	0	0	0
	W																

30.9.1 Host Port Configuration Register (HPCR)

There is one Host Port Configuration Register (HPCR) for each host port.

HPCR1 Host Port Configuration Register $BASE
HPCR2 $BASE+$04
HPCR3 $BASE+$08

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TFS DIR	TCLK DIR	TFCSEL				RFS DIR	RCLKDIR	RFCSEL							
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXDSEL			SYN		TXRXEN		INMEN	INMMASK							
TYPE	rw	rw	rw	rw	r	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Table 30-2. Host Port Configuration Register Description

Name	Description	Settings
TFSDIR Bit 31	**Transmit Frame Sync Direction Control**—This bit sets the direction of the TxFS pin of the interface as output or input.When set as input, the TFCSEL settings are ignored.When set as output, the TFCSEL settings determine the source port of the Frame Sync.	0 = TxFS is input pin 1 = TxFS is output
TCLKDIR Bit 30	**Transmit Clock Direction Control**—This bit sets the direction of the TxClk pin of the interface as output or input.When set as input, the TFCSEL settings are ignored.When set as output, the TFCSEL settings determine the source port of the Clock.	0 = TxClk is input pin 1 = TxClk is output
TFCSEL Bits 29–26	**Transmit Frame Sync and Clock Select**—Selects the source port from which TxFS and TxClk are sourced.	0xxx = selects TxFS and TxClk from port 1xxx = selects RxFS and RxClk from port 000—101 = Port 1–Port 6 110 = reserved 111 = reserved
RFSDIR Bit 25	**Receive Frame Sync Direction Control**—This bit sets the direction of the RxFS pin of the interface as output or input.When set as input, the RFCSEL settings are ignored.When set as output, the RFCSEL settings determine the source port of the Frame Sync.	0 = RxFS is input pin 1 = RxFS is output
RCLKDIR Bit 24	**Receive Clock Direction Control**—This bit sets the direction of the RxClk pin of the interface as output or input.When set as input, the RFCSEL settings are ignored.When set as output, the RFCSEL settings determine the source port of the Clock.	0 = RxClk is input pin 1 = RxClk is output

Table 30-2. Host Port Configuration Register Description (continued)

Name	Description	Settings
RFCSEL Bits 23–20	**Receive Frame Sync and Clock Select**—Selects the source port from which RxFS and RxClk are sourced. RxFS and RxClk can be sourced from TxFS and TxClk respectively from other ports.	0xxx = selects TxFS and TxClk from port 1xxx = selects RxFS and RxClk from port 000–101 = Port 1–Port 6 110 = reserved 111 = reserved
Reserved Bits 19–16	Reserved—These bits are reserved and should read 0.	
RXDSEL Bits 15–13	**Receive Data Select**—Selects the source port for the RxD data. RXDSEL is ignored if INMEN is enabled	xxx = port number for RxD, ignored if equal to self port number 000–101 = Port 1–Port 6 110 = reserved 111 = reserved
SYN Bit 12	**Synchronous/Asynchronous Select**—When SYN is set, synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals i.e the port is a 4-wire interface.When SYN is cleared, asynchronous mode is chosen and separate clock and frame sync signals are used for the transmit and receive sections i.e the port is a 6-wire interface	0= Asynchronous mode 1= Synchronous mode. (default).
Reserved Bit 11	Reserved—This bit is reserved and should read 0.	
TXRXEN Bit 10	**Transmit/Receive Switch Enable**—Swaps the transmit and receive signals from (Da-TxD, Db-RxD) to (Da-RxD, Db-TxD) **Note:** Present only in Port 3	0 = No switch 1 = switch
Reserved Bit 9	Reserved—This bit is reserved and should be read as 0.	
INMEN Bit 8	**Internal Network Mode Enable**—RxD from ports in internal network mode are ANDED together. RXDSEL is ignored. INMMASK determines which RxD signals are ANDed together. When internal network mode is enabled at Port3, then RXDSEL3[3:0] for TxDn_obe selection is ignored and TxD_obe is always driven high i.e asserted for all timeslots. This places a restriction on slave devices connected in external network mode such that these slave devices have all to be disabled. Please see Figure 30-12 on page 30-22, *Internal and External Network Mode Restriction*.	0 = Disable 1 = Enable internal network mode
INMMASK Bits 7–0	**Internal Network Mode Mask**—Bit mask that selects which of the RxD signals from ports are to be anded together for internal network mode. Bit 7 represents RxD from port8 and bit0 represents RxD from Port1. **Note:** Bit in self port position should be set as 1.	0 = include RxDn for ANDing. 1 = excludes the RxDn from ANDing.

30.9.2 Peripheral Port Configuration Register (PPCR)

There is one Peripheral Port Configuration Register (PPCR) for each peripheral port.

PPCR1 Peripheral Port Configuration Register $BASE+$10
PPCR2 $BASE+$14
PPCR3 $BASE+$1C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TFSDIR	TCLKDIR	TFCSEL				RFS DIR	RCLKDIR	RFCSEL							
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXDSEL			SYN		TXRXEN										
TYPE	rw	rw	rw	rw	r	rw	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Table 30-3. Peripheral Port Configuration Register Description

Name	Description	Settings
TFSDIR Bit 31	**Transmit Frame Sync Direction Control**—This bit sets the direction of the TxFS pin of the interface as output or input.When set as input, the TFCSEL settings are ignored.When set as output, the TFCSEL settings determine the source port of the Frame Sync.	0 = TxFS is input pin 1 = TxFS is output
TCLKDIR Bit 30	**Transmit Clock Direction Control**—This bit sets the direction of the TxClk pin of the interface as output or input.When set as input, the TFCSEL settings are ignored.When set as output, the TFCSEL settings determine the source port of the Clock.	0 = TxClk is input pin 1 = TxClk is output
TFCSEL Bits 29–26	**Transmit Frame Sync and Clock Select**—Selects the source port from which FS_obe, FS_out and CLK_obe and CLK_out are sourced.	0xxx = selects TxFS and TxClk from port 1xxx = selects RxFS and RxClk from port xxx = selection ignored if self-port number 110 = reserved 111 = reserved
RFSDIR Bit 25	**Receive Frame Sync Direction Control**—This bit sets the direction of the RxFS pin of the interface as output or input.When set as input, the RFCSEL settings are ignored.When set as output, the RFCSEL settings determine the source port of the Frame Sync.	0 = RxFS is input pin 1 = RxFS is output
RCLKDIR Bit 24	**Receive Clock Direction Control**—This bit sets the direction of the Waxlike pin of the interface as output or input.When set as input, the RFCSEL settings are ignored.When set as output, the RFCSEL settings determine the source port of the Clock.	0 = RxClk is input pin 1 = RxClk is output

Table 30-3. Peripheral Port Configuration Register Description (continued)

Name	Description	Settings
RFCSEL Bits 23–20	**Receive Frame Sync and Clock Select**—Selects the source port from which RxFS and RxClk are sourced. RxFS and RxClk can be sourced from TxFS and TxClk respectively from other ports.	0xxx = selects TxFS and TxClk from port 1xxx = selects RxFS and RxClk from port 000–101 = Port 1–Port 6 110 = reserved 111 = reserved
Reserved Bit 19–16	Reserved	N/A
RXDSEL Bits 15–13	**Receive Data Select**—selects the source port for the RxD data (TxD_in or RxD_in).	xxx = port number for TxD_in or RxD_in, ignored if equal to self port number 110 = reserved 111 = reserved
SYN Bit 12	**Synchronous/Asynchronous Select**—SYN controls whether the receive and transmit functions of the port occur synchronously or asynchronously with respect to each other. When SYN is set, synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals i.e the port is a 4-wire interface.When SYN is cleared, asynchronous mode is chosen and separate clock and frame sync signals are used for the transmit and receive sections i.e the port is a 6-wire interface.	0= Asynchronous mode 1= Synchronous mode (default).
Reserved Bits 11	Reserved—These bits are reserved and should read 0.	
TXRXEN Bit 10	**Transmit/Receive Switch Enable**—Swaps the transmit and receive signals from (Da-TxD, Db-RxD) to (Da-RxD, Db-TxD).	0 = No switch 1 = Switch
Reserved Bit 9–0	Reserved—These bits are reserved and should read 0.	

30.10 Peripheral Connectivity through AUDMUX Configuration

This section describes some of the peripheral connectivity scenarios through AUDMUX configuration and some limitations.

30.10.1 Generic Configuration

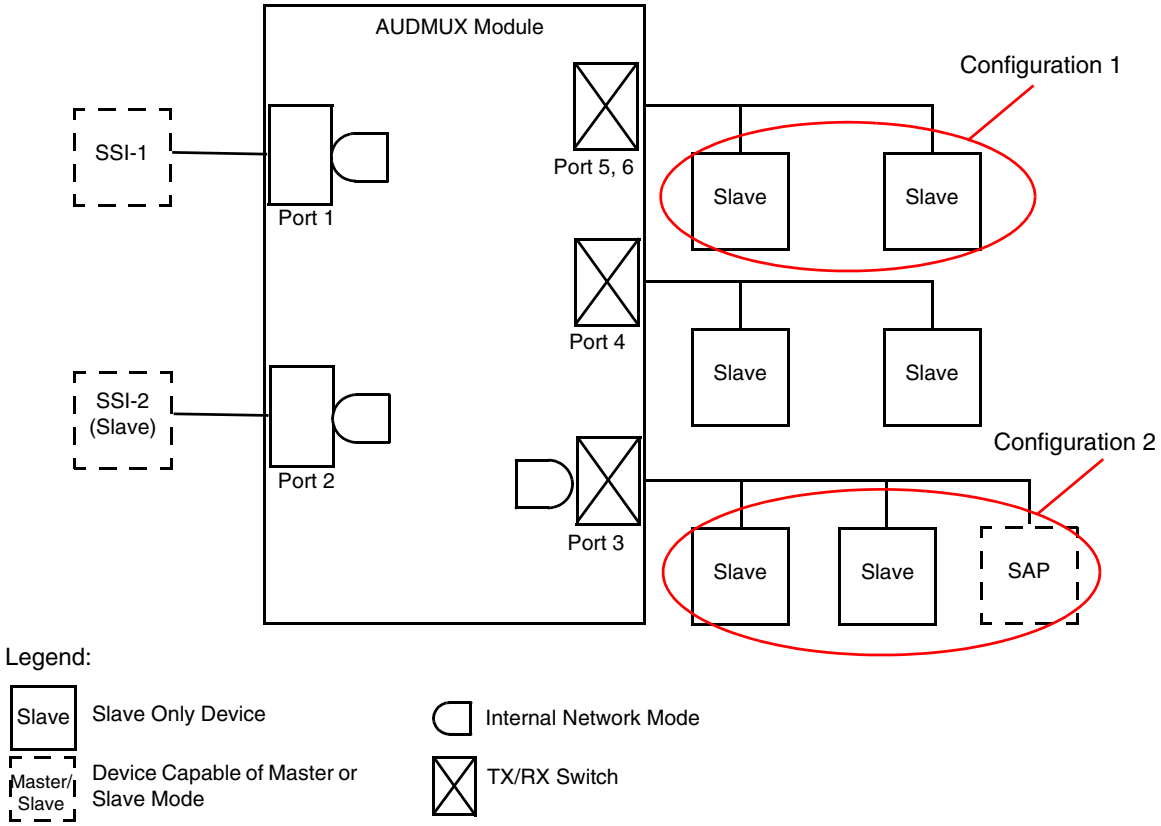


Figure 30-9. SAP as Master to SSI2 as Slave Interconnection

- Figure 30-9 shows a general configuration of the audmux
- It does not show what paths are enabled or possible
- Only Ports 1, 2, 3 are capable of internal network mode
- Only Ports 3, 4, 5, 6 are capable of Tx/Rx switch

Where internal network mode is enabled, for example at Port 1, then Port 1 is referred to as the egress port.

Hence, Port 1, 2, 3 become egress ports when internal network mode is enabled at that port.

— ***Config1:*** only slave devices are connected in network mode.

— ***Config2:*** one master/slave capable device with slave only devices.

30.10.2 AUDMUX Configuration with SSI1 and SAP as Master

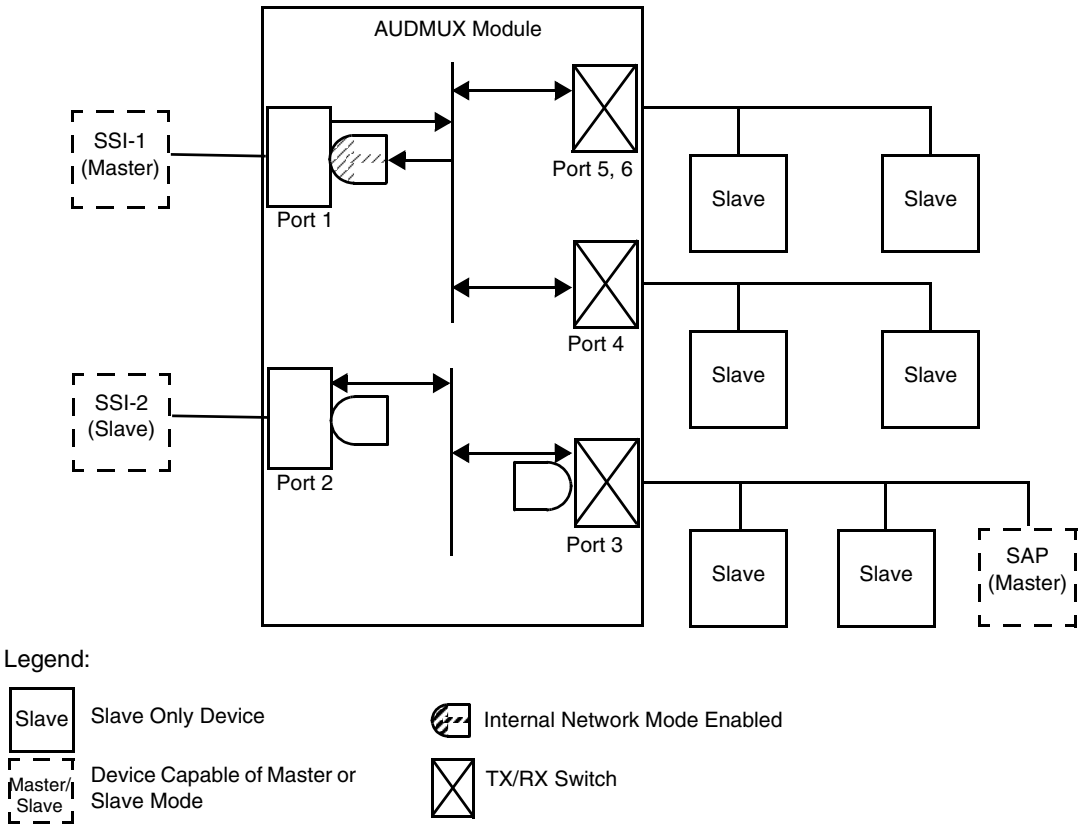


Figure 30-10. SSI1 as Master in Internal Network Mode

Figure 30-10 shows two possible audio paths which can be configured simultaneously.

Audio Path 1:

- Figure 30-10 shows, SS1-1 configured as master communicating to slaves on Ports 4, 5, and 6.
- Port 1 internal network mode is enabled hence it is the egress port.

Audio Path 2:

- The SAP (Audio port from the Baseband) is connected to slave ports and SSI-2. Tx/Rx switch is not enabled, neither is internal network mode.
- SSI-2 (Internal to i.MX21) is connected as slave.

30.10.3 Tx-Rx Switch Enabled

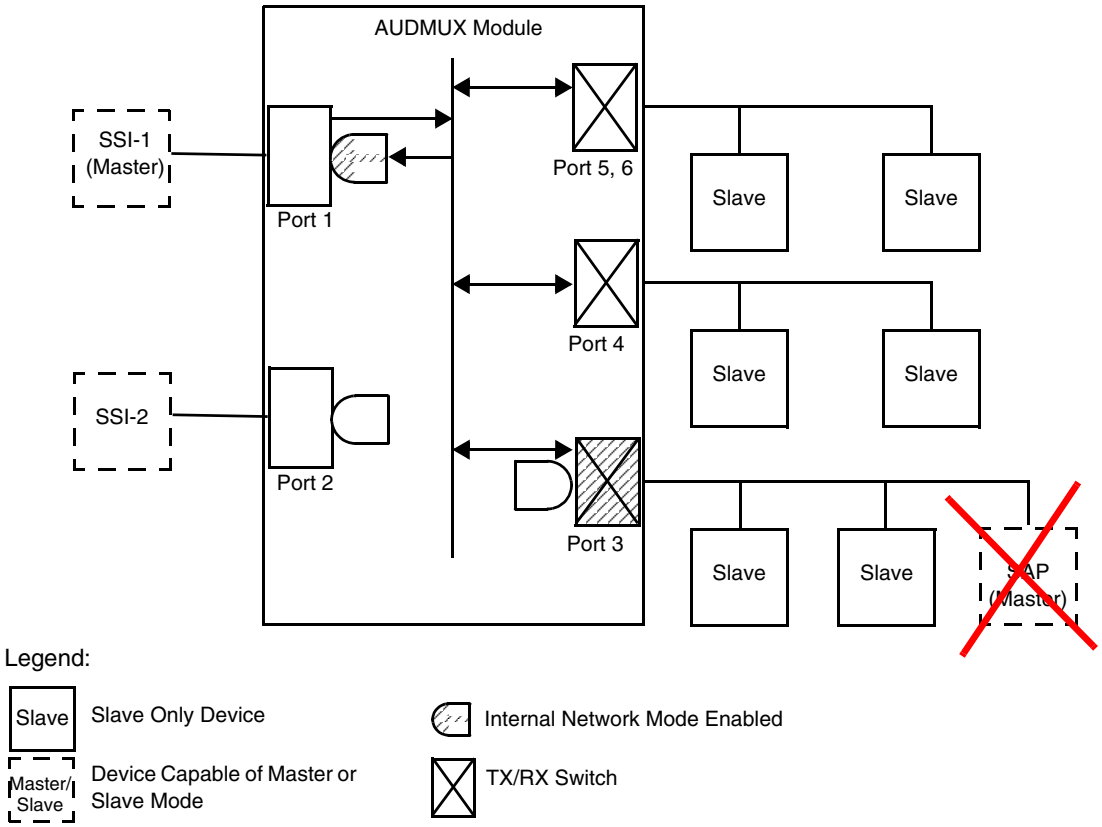


Figure 30-11. Tx-Rx Switch Restriction

Figure 30-11 AUDMUX configuration with SSI-1 as the master of the flow.

Flow:

- SSI-1 is the master which is connected to all peripheral ports.
- Internal network mode is enabled at Port 1 to receive data from Ports 3, 4, 5, 6.
- Tx/Rx Switch is enabled only at Port 3 to maintain signal directions to be consistent for slaves on Port 3.
- The SAP (Audio port from the Baseband) master must be disabled because its Tx and Rx signals will not be consistent.
- Though disabling the SAP master, apparently, makes Config2 into Config1, signal directions are not the same as config1, hence the requirement for the Tx/Rx switch.

30.10.4 Internal/External Network Mode

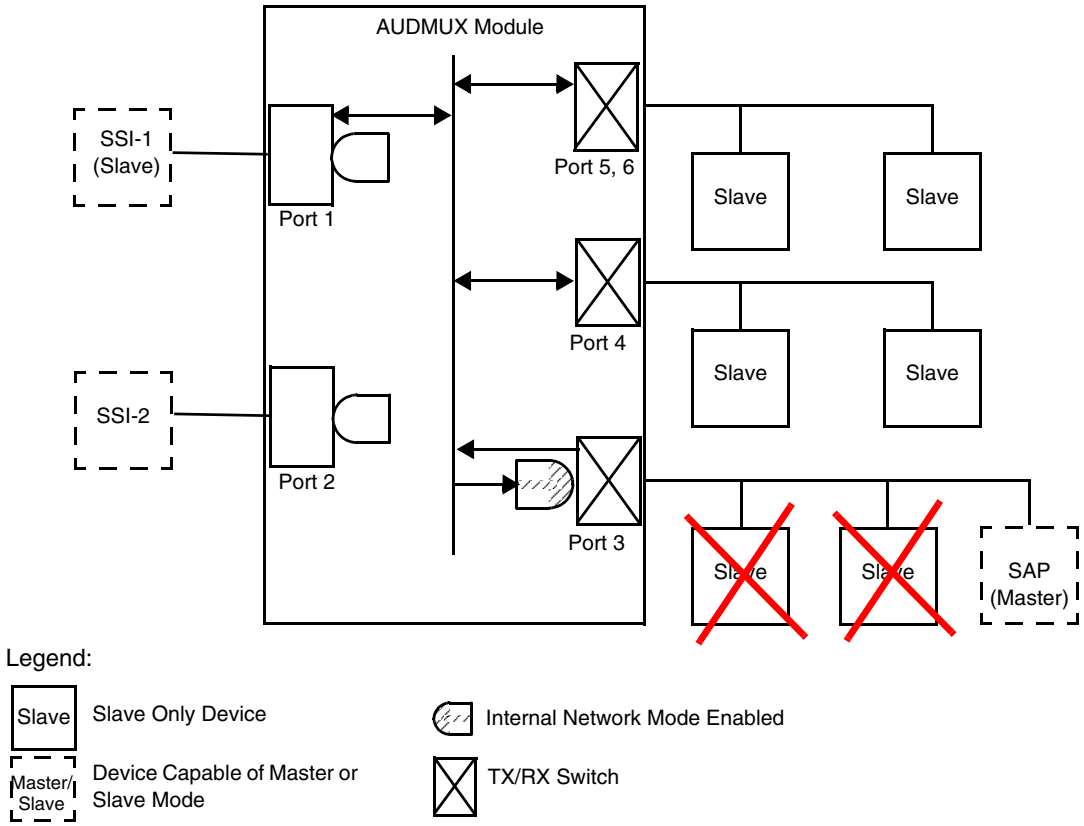


Figure 30-12. Internal and External Mode Restriction

Figure 30-12 depicts the limitation posed by Internal network mode.

- In this flow, the SAP (Audio port from the Baseband) is now the master and internal network mode is enabled at Port 3.
- Port 3 is now an egress port for receive data coming from the other ports.
- The locally attached slave devices must be disabled because the internal network mode will ALWAYS drive the output at the egress port regardless of timeslots which will cause a multiple driver conflict with slave transmission at Port3, otherwise.
- Disabling of slave devices in config2 effectively removes external network mode at Port 3. This is what is meant by internal network mode cannot be used with external network mode.

Part 7
Connectivity and Expansion

Chapter 31
Universal Asynchronous Receiver/Transmitters (UART) Modules

This chapter describes the four universal asynchronous receiver/transmitter (UART) modules in i.MX21. The UART modules are capable of standard RS-232 non-return-to-zero (NRZ) encoding format and IrDA-compatible infrared modes. Each UART provides serial communication capability with external devices through an RS-232 cable or through use of external circuitry that converts infrared signals to electrical signals (for reception) or transforms electrical signals to signals that drive an infrared LED (for transmission) to provide low speed IrDA compatibility to the i.MX21.

All UARTs transmit and receive characters that are either 7 or 8 bits in length (program selectable). To transmit, data is written from the peripheral data bus to a 32-byte transmitter FIFO (TxFIFO). This data is passed to the shift register and shifted serially out on the transmitter pin (TXD). To receive, data is received serially from the receiver pin (RXD) and stored in a 32-half-words-deep receiver FIFO (RxFIFO). The received data is retrieved from the RxFIFO on the peripheral data bus. The RxFIFO and TxFIFO generate maskable interrupts as well as DMA Requests when the data level in each of the FIFO reaches a programmed threshold level.

The UARTs generate baud rates based on a configurable divisor and input clock. The UARTs also contain configurable auto baud detection circuitry to receive 1 or 2 stop bits as well as odd, even, or no parity. The receiver detects framing errors, idle conditions, BREAK characters, parity errors, and overrun errors.

The UART modules use a software interface for control of modem operations and have a serial infrared (IR) module that decodes and encodes IrDA-compatible serial IR data.

The following list contains the key features of the i.MX21 UARTs:

- High speed TIA/EIA-232-F compatible
- 7 or 8 data bits
- 1 or 2 stop bits
- Programmable parity (even, odd, and no parity)
- Hardware flow control support for request to send (RTS) and clear to send (CTS) signals
- Edge selectable RTS and edge detect interrupts
- Status flags for various flow control and FIFO states
- Serial IR interface low speed, IrDA-compatible (up to 115.2 kbit/s)
- Voting logic for improved noise immunity (16x oversampling)
- Transmitter FIFO empty interrupt suppression
- Auto baud rate detection
- Receiver and transmitter enable/disable

- RTS, IrDA asynchronous wake (AIRINT), and receive asynchronous wake (AWAKE) interrupts wake the ARM926EJ-S processor from STOP mode
- Maskable interrupts
- Two DMA Requests (TxFIFO DMA Request and RxFIFO DMA Request)
- Escape character sequence detection
- Software reset ($\overline{\text{SRST}}$), in configuration register 2

31.1 Module Interface

The serial and modem control signals used by the UART module are identified and described in Table 31-1.

Table 31-1. UART Module Interface Signals

Signal Names	I/O	Active	UART	Comments
Serial Signals				
UART1_TXD UART2_TXD UART3_TXD UART4_TXD	OUT	HIGH	1 2 3 4	Transmitter serial output (multiplexed IR or NRZ encoding format)
UART1_RXD UART2_RXD UART3_RXD UART4_RXD	IN	HIGH	1 2 3 4	Receiver serial input (multiplexed IR or NRZ encoding format)
Modem Control Signals				
$\overline{\text{UART1\_RTS}}$ $\overline{\text{UART2\_RTSU}}$ $\overline{\text{ART3\_RTSUA}}$ $\overline{\text{RT4\_RTS}}$	IN	LOW	1 2 3 4	Controls the transmitter. By asserting RTS, the modem signals ready to receive to the UART. Normally, the transmitter waits until $\overline{\text{UARTx\_RTS}}$ is active (low) before transmitting a character, however when the ignore **$\overline{\text{UARTx\_RTS}}$ pin** (IRTS) bit is set, the transmitter sends each character as it is ready to transmit. When this pin serves as a general purpose input, its status is read in the RTSS bit. This pin can post an interrupt on any transition of this pin and wake the ARM9 core from STOP mode on its assertion. When $\overline{\text{RTS}}$ is negated during a transmission, the UART transmitter finishes transmitting the current character and shuts off. The contents of the TxFIFO (characters to be transmitted) remain undisturbed.
$\overline{\text{UART1\_CTSU}}$ $\overline{\text{ART2\_CTSUA}}$ $\overline{\text{RT3\_CTSUAR}}$ $\overline{\text{T4\_CTS}}$	OUT	LOW	1 2 3 4	This output pin serves two purposes. Normally, the receiver indicates that it is ready to receive data by asserting this pin (low). When the receiver detects a pending overrun, it negates this pin. For other applications, this pin functions as a general purpose output controlled by the CTS bit in UART Control Register 2.

31.2 Pin Configuration for UART1, UART2, UART3, and UART4

Table 31-2 lists the pins used for the UART1, UART2, UART3, and UART4 modules. These pins are multiplexed with other functions on the device, and must be configured for UART operation.

NOTE

The user must ensure that the data direction bits in the GPIO are set to the correct direction for proper operation. See GPIO chapter “Data Direction Register” for details.

Table 31-2. Pin Configuration

Pin	Setting	Configuration Procedure
UART1_RXD	Primary function of GPIO Port E [13]	1.Clear bit 13 of Port E GPIO In Use Register (GIUS_E) 2.Clear bit 13 of Port E General Purpose Register (GPR_E)
UART1_TXD	Primary function of GPIO Port E [12]	1.Clear bit 12 of Port E GPIO In Use Register (GIUS_E) 2.Clear bit 12 of Port E General Purpose Register (GPR_E)
$\overline{\text{UART1\_RTS}}$	Primary function of GPIO Port E [15]	1.Clear bit 15 of Port E GPIO In Use Register (GIUS_E) 2.Clear bit 15 of Port E General Purpose Register (GPR_E)
$\overline{\text{UART1\_CTS}}$	Primary function of GPIO Port E [14]	1.Clear bit 14 of Port E GPIO In Use Register (GIUS_E) 2.Clear bit 14 of Port E General Purpose Register (GPR_E)
UART2_RXD	Primary function of GPIO Port E [7]	1.Clear bit 7 of Port E GPIO In Use Register (GIUS_E) 2.Clear bit 7 of Port E General Purpose Register (GPR_E)
$\overline{\text{UART2\_TXD}}$	Primary function of GPIO Port E [6]	1.Clear bit 6 of Port E GPIO In Use Register (GIUS_E) 2.Clear bit 6 of Port E General Purpose Register (GPR_E)
$\overline{\text{UART2\_RTS}}$	Primary function of GPIO Port E [4]	1.Clear bit 4 of Port E GPIO In Use Register (GIUS_E) 2.Clear bit 4 of Port E General Purpose Register (GPR_E)
$\overline{\text{UART2\_CTS}}$	Primary function of GPIO Port E [3]	1.Clear bit 3 of Port E GPIO In Use Register (GIUS_E) 2.Clear bit 3 of Port E General Purpose Register (GPR_E)
UART3_RXD	Primary function of GPIO Port E [9]	1.Clear bit 9 of Port E GPIO In Use Register (GIUS_E) 2.Clear bit 9 of Port E General Purpose Register (GPR_E)
UART3_TXD	Primary function of GPIO Port E [8]	1.Clear bit 8 of Port E GPIO In Use Register (GIUS_E) 2.Clear bit 8 of Port E General Purpose Register (GPR_E)
$\overline{\text{UART3\_RTS}}$	Primary function of GPIO Port E [11]	1.Clear bit 11 of Port E GPIO In Use Register (GIUS_E) 2.Clear bit 11 of Port E General Purpose Register (GPR_E)
$\overline{\text{UART3\_CTS}}$	Primary function of GPIO Port E [10]	1.Clear bit 10 of Port E GPIO In Use Register (GIUS_E) 2.Clear bit 10 of Port E General Purpose Register (GPR_E)
UART4_RXD	Alternate function of GPIO Port B [31]	1.Clear bit 31of Port B GPIO In Use Register (GIUS_B) 2.Set bit 31of Port B General Purpose Register (GPR_B)
UART4_TXD	Alternate function of GPIO Port B [28]	1.Clear bit 28 of Port B GPIO In Use Register (GIUS_B) 2.Set bit 28 of Port B General Purpose Register (GPR_B)
$\overline{\text{UART4\_RTS}}$	Alternate function of GPIO Port B [26]	1.Clear bit 26 of Port B GPIO In Use Register (GIUS_B) 1.Set bit 26 of Port B General Purpose Register (GPR_B)
$\overline{\text{UART4\_CTS}}$	Alternate function of GPIO Port B [29]	1.Clear bit 29 of Port B GPIO In Use Register (GIUS_B) 2.Set bit 29 of Port B General Purpose Register (GPR_B)

31.3 General UART Definitions

Definitions of terms that occur the following discussions are given in this section.

- **Bit Time**—The period of time required to serially transmit or receive 1-bit of data (1 cycle of the baud rate frequency)
- **Start bit**—The bit time of a logic 0 that indicates the beginning of a data frame. A start bit begins with a 1-to-0 transition, and is preceded by at least 1-bit time of logic 1.
- **Stop bit**—1-bit time of logic 1 that indicates the end of a data frame.
- **BREAK**—A frame in which all of the data bits, including the stop bit, are logic 0. This type of frame is usually sent to signal the end of a message or the beginning of a new message.
- **Frame**—A start bit followed by a specified number of data or information bits and terminated by a stop bit. The number of data or information bits depends on the format specified and must be the same for the transmitting device and the receiving device. The most common frame format is 1 start bit followed by 8 data bits (least significant bit first) and terminated by 1 stop bit. An additional stop bit and a parity bit also can be included.
- **Framing Error**—An error condition that occurs when the stop bit of a received frame is missing, usually when the frame boundaries in the received bit stream are not synchronized with the receiver bit counter. Framing errors can go undetected if a data bit in the expected stop bit time happens to be a logic 1. A framing error is always present on the receiver side when the transmitter is sending BREAKs. However, when the UART is programmed to expect 2 stop bits and only the first stop bit is received, this is not a framing error by definition.
- **Parity Error**—An error condition that occurs when the calculated parity of the received data bits in a frame does not match the parity bit received on the RXD input. Parity error is calculated only after an entire frame is received.
- **Idle**—One in NRZ encoding format and selectable polarity in IrDA mode.
- **Overrun Error**—An error condition that occurs when the latest character received is ignored to prevent overwriting a character already present in the UART receive buffer (RxFIFO). An overrun error indicates that the software reading the buffer (RxFIFO) is not keeping up with the actual reception of characters on the RXD input.

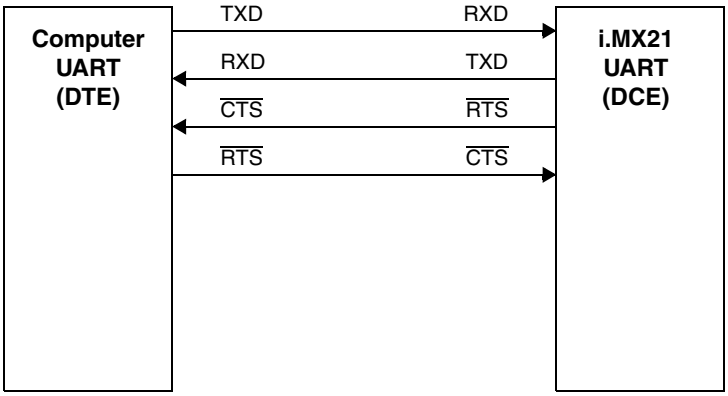


Figure 31-1. General Connections for a UART with a Modem

31.3.1 RTS—UART Request To Send

The UART request to send input controls the transmitter. The modem or other terminal equipment signals the UART when it is ready to receive by asserting RTS on the UARTx_RTS pin. Normally, the transmitter waits until this signal is active (low) before transmitting a character, however when the ignore UARTx_RTS pin (IRTS) bit is set, the transmitter sends a character as soon as it is ready to transmit. This pin can post an interrupt on any transition of this pin and wakes the ARM926EJ-S processor from STOP mode on its assertion. When RTS is negated during a transmission, the UART transmitter finishes transmitting the current character and shuts off. The contents of the TxFIFO (characters to be transmitted) remain undisturbed.

31.3.2 RTS Edge Triggered Interrupt

The input to the UARTx_RTS pin can be programmed to generate an interrupt on a selectable edge. The operation of the RTS edge triggered interrupt (RTSF) is summarized in Table 31-3.

To enable the UARTx_RTS pin to generate an interrupt, set the request to send interrupt enable (RTSEN) bit in the UART Control Register 2 to 1. Writing 1 to the RTS edge triggered interrupt flag (RTSF) bit in UCR2_x clears the interrupt flag.The interrupt can occur on the rising edge, falling edge, or either edge of the RTS input. The request to send edge control (RTEC) field in UCR2_x programs the edge that generates an interrupt. When RTEC is set to 0x00 and RTSEN = 1, the interrupt occurs on the rising edge (default). When RTEC is set to 0x01 and RTSEN = 1, the interrupt occurs on the falling edge. When RTEC is set to 0x1X and RTSEN = 1, the interrupt occurs on either edge.This is a synchronous interrupt. The RTSF bit is cleared by writing 1 to it. Writing 0 to RTSF has no effect.

Table 31-3. RTS Edge Triggered Interrupt Truth Table

RTS	RTSEN	RTEC [1]	RTEC [0]	RTSF	Interrupt Occurs On...	IPI_UART_MINT
X	0	X	X	0	Interrupt disabled	1
1≥0	1	0	0	0	Rising edge	1
0≥1	1	0	0	1	Rising edge	0
1≥0	1	0	1	1	Falling edge	0
0≥1	1	0	1	0	Falling edge	1
1≥0	1	1	X	1	Either edge	0
0≥1	1	1	X	1	Either edge	0

There is another RTS interrupt that is not programmable, however it asserts the RTS Delta (RTSD) bit when the RTS pin changes state. The status bit RTSD asserts the IPI_UART_MINT interrupt when the RTS delta interrupt enable = 1. This is an asynchronous interrupt. The RTSD bit is cleared by writing 1 to it. Writing 0 to the RTSD bit has no effect.

31.3.3 $\overline{\text{CTS}}$—Clear To Send

Normally, the receiver indicates that it is ready to receive data by asserting this pin (low). When the CTS trigger level is programmed to trigger at 32 characters received and the receiver detects the valid start bit of the 33rd character, it deasserts this pin.

31.3.4 Programmable CTS Deassertion

The CTS output can also be programmed to de-assert when the RxFIFO reaches a certain level. Setting the CTS trigger level at any value less than 32 deasserts the CTS pin on detection of the valid start bit of the N + 1 character (where N is the trigger level setting). The receiver continues to receive characters until the RxFIFO is full.

31.3.5 TXD—UART Transmit

This is the transmitter serial output. When operating in normal mode, NRZ encoded data is output. When operating in infrared mode, a 3/16 bit-period pulse is output for each 0 bit transmitted, and no pulse is output for each 1-bit transmitted. For RS-232 applications, this pin must be connected to an RS-232 transmitter.

31.3.6 RXD—UART Receive

This is the receiver serial input. When operating in normal mode, NRZ encoded data is expected. When operating in infrared mode, a narrow pulse is expected for each 0 bit received and no pulse is expected for each 1-bit received. External circuitry must convert the IR signal to an electrical signal. RS-232 applications require an external RS-232 receiver to convert voltage levels.

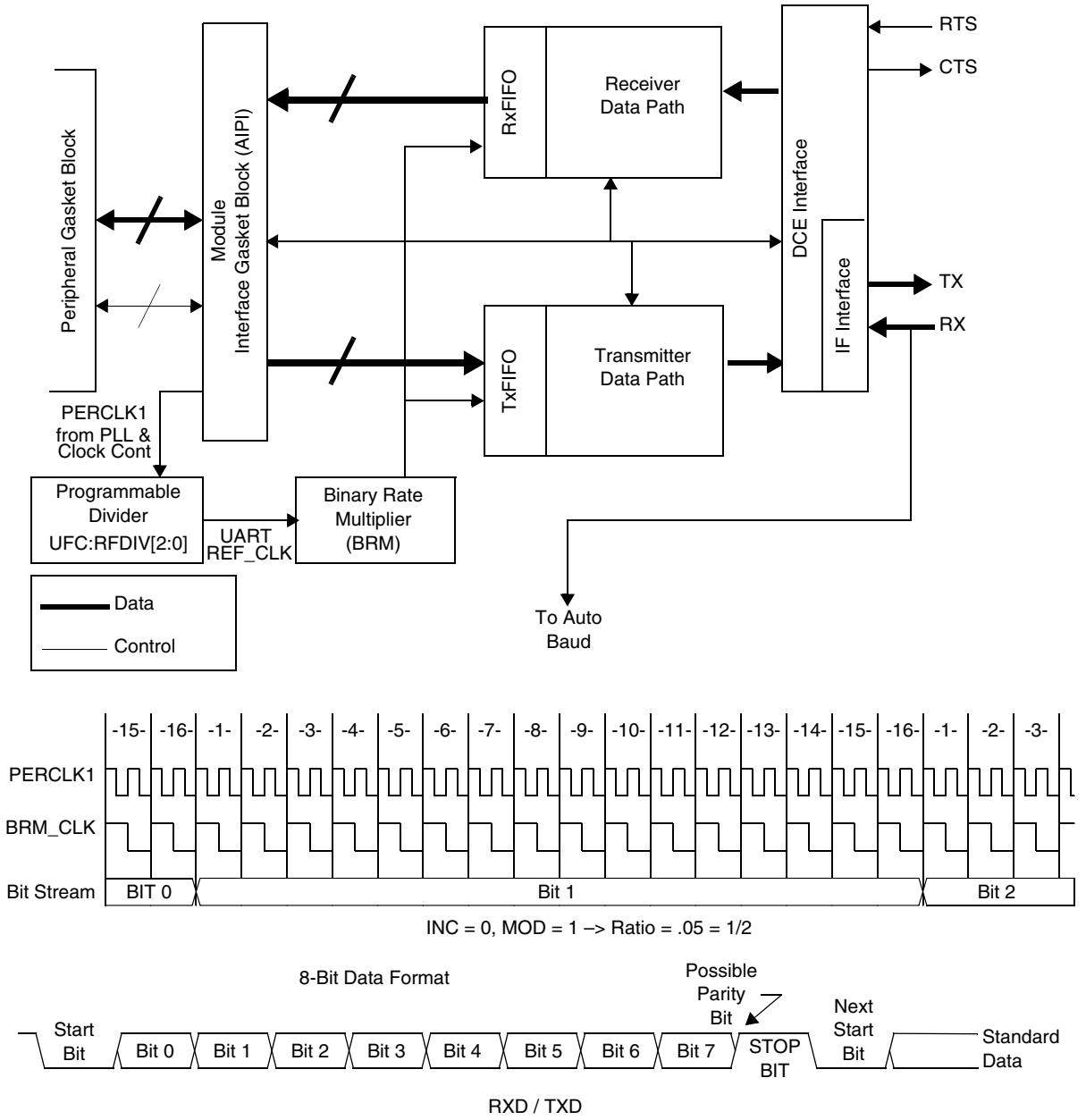


Figure 31-2. UART Block Diagram and Clock Generation Diagram

31.4 Sub-Block Description

The UART module is easy to use from both hardware and software perspectives. Six working registers provide all status (USR1_x and USR2_x) and control (UCR1_x, UCR2_x, UCR3_x, and UCR4_x) functions. A separate test register is provided for applications that require it. The binary rate multiplier registers (UBIR_x, UBMR_x) control the UART bit rate.

There is also a transmitter register (UTXD_x) and a receiver register (URXD_x). The registers are optimized for a 16-bit bus. All status bits associated with the received data are accessible along with the data in a single read. Except for the transmit data (TX_DATA) field in the UART Transmitter Registers,

all register bits are readable and most are read/write. The UART Baud Rate Count Register (UBRC_x) performs automatic baud rate detection. There are also three registers for the escape sequence detection, the UART Escape Character Register (UESC_x) and the UART Escape Timer Register (UTIM_x) and the UART One Millisecond Register (ONEMS_x). The following sections describe the basic functionality of the major blocks in UART module.

The reference frequency is defined as the input peripheral clock, PERCLK1, divided by the value of the RFDIV [2:0] bits in the UFCR. Also note that **the frequency of the system clock (ipg_clk) must be greater than the UART reference frequency**. For example, if the UART reference frequency is 16 MHz, ipg_clk must be greater than 16 MHz. If the ipg_clk frequency is not greater, some of the UART features may not operate properly. For more details on ipg_clk please refer to Chapter “Phase-Locked Loop and Clock Controller”.

31.4.1 Transmitter

The transmitter accepts a parallel character from the ARM926EJ-S processor and transmits it serially. The start, stop, and parity (when enabled) bits are added to the character. When the ignore RTS bit (IRTS) is set, the transmitter sends a character as soon as it is ready to transmit. RTS can be used to provide flow-control of the serial data. When RTS is negated (high), the transmitter finishes sending the character in progress (if any), stops, and waits for $\overline{\text{RTS}}$ to be asserted (low) again. Generation of BREAK characters and parity errors (for debugging purposes) is supported. The transmitter operates from the 1x clock provided by the BRM. Normal NRZ encoded data is transmitted when the IR interface is disabled.

The transmitter FIFO (TxFIFO) contains 32 bytes. The data is written to TxFIFO by writing to the UTXD_x register with the byte data to the [7:0] bits. The data is written consecutively if the TxFIFO is not full or is read consecutively if the TxFIFO is not empty. If the TxFIFO is full and data is again attempted to be written to the FIFO, the data cannot be written unless a read is first performed.

31.4.2 Transmitter FIFO Empty Interrupt Suppression

The transmitter FIFO empty interrupt suppression logic suppresses the interrupt between writes to the TxFIFO. When a character is written to the TxFIFO, it is immediately transferred to the transmitter shift register (PISO_OUT) on the next transmit baud rate clock (when the transmitter is enabled). The suppression logic allows the software to write another character to the TxFIFO before the interrupt is asserted. When the transmitter shift register empties before another character is written to the TxFIFO, the interrupt is asserted. Writing data (even a single character) to the TxFIFO releases the interrupt.

The interrupt is asserted on the following conditions:

- System Reset.
- UART module reset.
- When a single character has been written to Transmitter FIFO and then the Transmitter FIFO and the Transmitter Shift Register become empty until another character is written to the Transmitter FIFO.
- The last character in the TxFIFO is transferred to the shift register, when TxFIFO contains two or more characters. See Figure 31-3.

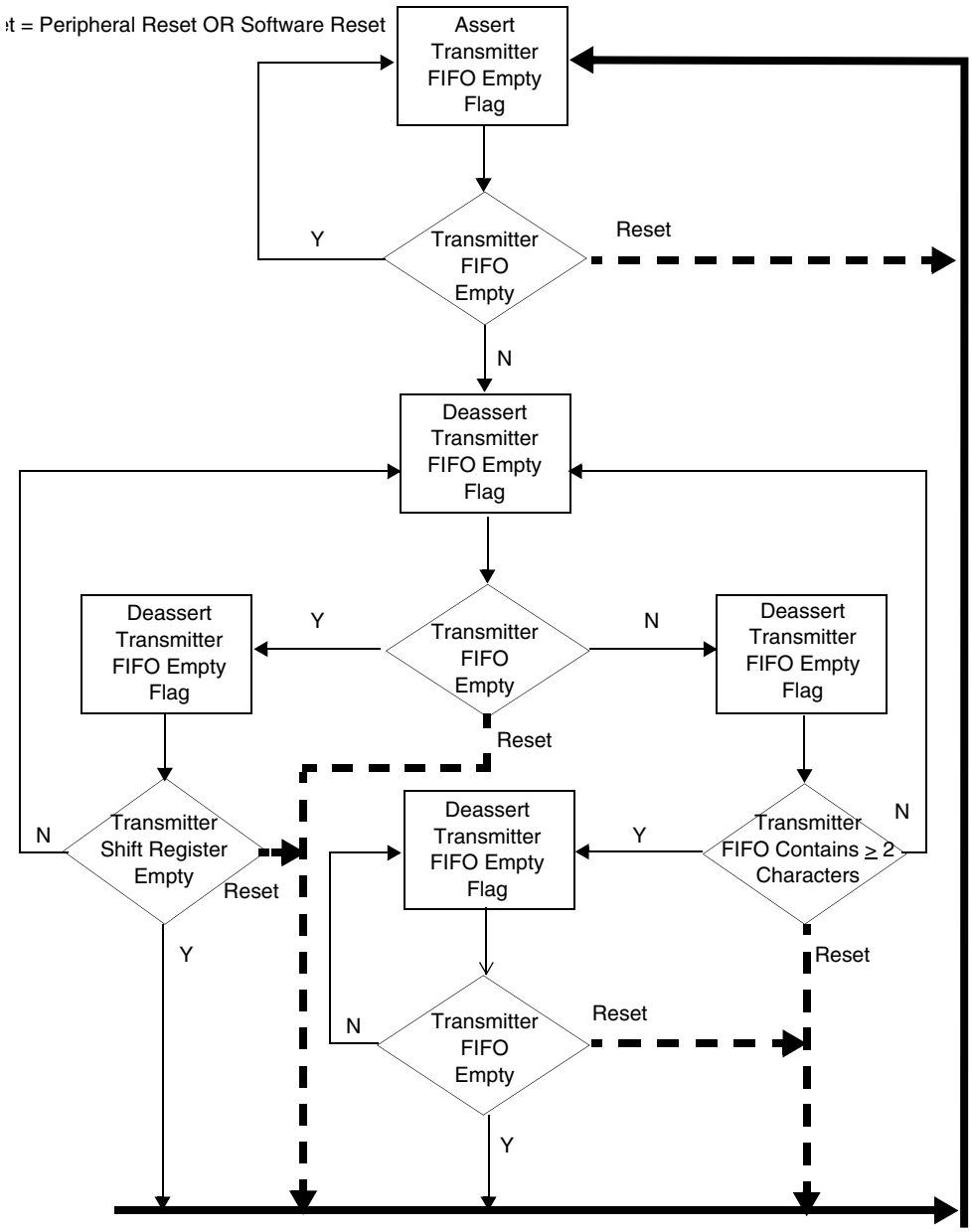


Figure 31-3. Transmitter FIFO Empty Interrupt Suppression Flow Chart

31.4.3 Receiver

The receiver accepts a serial data stream and converts it into parallel characters. When enabled, it searches for a start bit, qualifies it, and samples the following data bits at the bit-center. Jitter tolerance and noise immunity are provided by sampling at a 16x rate and using voting techniques to clean up the samples. Once the start bit is found, the data bits, parity bit (if enabled), and stop bits (either 1 or 2 depending on user selection) are shifted in. Parity is checked and its status reported in the URXD_x register when parity is enabled. Frame errors and BREAKs are also checked and reported. When a new character is ready to be received by the RxFIFO, the receive data ready (RDR) bit in the UART Status Register 2 (USR2_x) is asserted and an interrupt is posted (if DREN = 1). If the receiver trigger level is set to 0, the receiver ready

interrupt flag (RRDY) is asserted and an interrupt is posted if the receiver ready interrupt enable bit is set (RRDYEN = 1). If the UART Receiver Register (URXD_x) is read as a word and there is only 1 character in the RxFIFO, the interrupt generated by the RRDY bit is automatically cleared and the data along with 4 status bits are read by the ARM926EJ-S processor (see the bit descriptions in Section 31.6.8, “UART Status Register 1,” on page -35 and Section 31.6.9, “UART Status Register 2,” on page -37). The RRDY bit is cleared when the data in the RxFIFO falls below the programmed trigger level.

Normal NRZ encoded data is expected when the IR interface is disabled.

The RxFIFO contains 32 half-words. The data is read from the RxFIFO by reading the half-word data in the [15:0] bits in the URXD_x register. The data is written consecutively if the RxFIFO is not full, or is read consecutively if the RxFIFO is not empty. When additional data is written to the RxFIFO while it is full, the write operation cannot complete unless a read is performed. If a write is performed on the RxFIFO when it is full, the ORE bit in the USR2_x register is set. The ORE bit is cleared by writing 1 to it.

31.4.4 Idle Line Detect

The receiver logic block includes the ability to detect an idle line. Idle lines indicate the end or the beginning of a message. For an idle condition to occur, RxFIFO must be empty and the RXD pin must be idle for more than a configured number of frames.

When the idle condition detected interrupt enable (IDEN) bit in the UART Control Register 1 (UCR1_x) is set and the line is idle for 4 (default), 8, 16, or 32 (maximum) frames, the detection of an idle condition flags an interrupt. When an idle condition is detected, the IDLE bit in the USR2_x register is set. Clear the IDLE bit by writing 1 to it. Writing 0 to the IDLE bit has no effect.

Similarly, the RXDS in the USR1_x indicates that the receiver state machine is in an IDLE state, the next state is IDLE, and the receive pin is high. RXDS is automatically cleared when a character is received. The RXDSEN in the UCR3_x register controls the receive status interrupt. When this bit is enabled and RXDS status bit is set, the interrupt will be generated.

31.4.4.1 Idle Condition Detect Configuration

The idle condition detect (ICD [1:0]) field is located in the UART Control Register 1. If the bits are set to 00b, RXD must be idle for more than 4 frames before the IDLE bit is asserted. If the bits are set to 01b, RXD must be idle for more than 8 frames before the IDLE bit is asserted. If the bits are set to 10b, RXD must be idle for more than 16 frames before the IDLE bit is asserted. If the bits are set to 11b, RXD must be idle for more than 32 frames before the IDLE bit is asserted (see Table 31-4).

Table 31-4. IDLE Detection Truth Table

IDEN	ICD [1]	ICD [0]	IDLE	UART_INT
0	X	X	0	1
1	0	0	Asserted after 4 idle frames	Asserted after 4 idle frames
1	0	1	Asserted after 8 idle frames	Asserted after 8 idle frames
1	1	0	Asserted after 16 idle frames	Asserted after 16 idle frames

Table 31-4. IDLE Detection Truth Table (continued)

IDEN	ICD [1]	ICD [0]	IDLE	UART_INT
1	1	1	Asserted after 32 idle frames	Asserted after 32 idle frames
Note: This table assumes that no other interrupt is set at the same time this interrupt is set for theUART_INTsignal. This table shows how this interrupt affects the $\overline{\text{UART\_INT}}$ signal.				

During a normal message there is no idle time between frames. When all of the information bits in a frame are logic 1s, the start bit ensures that at least one logic 0 bit time occurs for each frame so that the IDLE bit is not asserted.

31.4.5 Ageing Character Detect

The receiver block also includes the possibility to detect when at least character has been sitting into the RxFIFO for a time corresponding to 8 characters. This ageing character capability allows the UART to inform the MCU that there is less character into the RxFIFO than the Rx trigger and, no new character has been detected on the RXD line. The ageing capability is a timer which starts to count as soon as there is a character in RxFIFO. This counter is reset when either a RxFIFO read is performed or another character has been received in RxFIFO. If none of those two events occurs, the bit AGTIM (USR1[8]) is set when the counter has measured a time corresponding to 8 characters. AGTIM bit is cleared by writing a 1 to it. AGTIM can flag an interrupt to MCU on UART_INT if ATEN (UCR2[3]) has been set.

To summarize, AGTIM is set when:

- There is at least one character into RxFIFO
- No Read has occurred on RxFIFO and RXD line has stayed high for a time corresponding to 8 characters
- The RxFIFO trigger is not reached (RRDY=0)

31.4.6 Receiver Wake

The receiver logic block includes the WAKE bit in the USR2_x register that is set when the receiver detects the start bit. The WAKE bit is not set until the start bit is qualified. When the wake interrupt enable (WKEN) bit is enabled, the receiver flags an interrupt if the WAKE status bit is set. The WAKE bit is cleared by writing 1 to it. Writing 0 to the WAKE bit has no effect.

When the asynchronous wake interrupt (AWAKE) is enabled (AWAKEN = 1) and the ARM926EJ-S core is in STOP mode, a falling edge detected on the receive pin asserts the AWAKE bit and the interrupt to wake the ARM926EJ-S processor from STOP mode. Clear the AWAKE bit by writing 1 to it. Writing 0 to the AWAKE bit has no effect.

If the asynchronous IR WAKE interrupt is enabled (AIRINTEN = 1), the UART is configured for IR mode, the ARM926EJ-S processor is in STOP mode, and a falling edge is detected on the receive pin, this asserts the AIRINT bit and the interrupt and wakes the ARM926EJ-S processor from STOP mode. Clear the AIRINT bit by writing 1 to it. Writing 0 to the AIRINT bit has no effect.

Recommended procedure for programming the asynchronous interrupts is to first clear them by writing 1 to the appropriate bit in the UART Status Register 1 (USR1_x). Poll or enable the interrupt for the Receiver

IDLE Interrupt Flag (RXDS) in the USR1_x. When asserted, the RXDS bit indicates to the software that the receiver state machine is in the idle state, the next state is idle, and the RXD pin is idle (high). After following this procedure, enable the asynchronous interrupt and enter STOP mode.

31.4.7 Receiving a BREAK Condition

A BREAK condition is received when the receiver detects all 0s (including a 0 during the bit time of the stop bit) in a frame. The BREAK condition asserts the BRCD bit in the USR2_x register and writes only the first BREAK character to the RxFIFO. Clear the BRCD bit by writing 1 to it. Writing 0 to the BRCD bit has no effect. The BRCD bit remains asserted as long as a BREAK condition exists. The BREAK condition ends when the receiver detects a 0-to-1 transition.

31.4.8 Vote Logic

The vote logic block provides jitter tolerance and noise immunity by sampling with respect to VOTE_CLK and using voting techniques to clean up the samples. The voting is implemented by sampling the incoming signal constantly on the rising edge of VOTE_CLK. The receiver is provided with the majority vote value, which is 2 out of the 3 samples. Examples of the majority vote results of the vote logic are shown in Table 31-5.

Table 31-5. Majority Vote Results

Samples	Vote
000	0
101	1
001	0
111	1

The vote logic captures a sample on every rising edge of BRM_CLK, however the receiver uses 16x oversampling to take its value in the middle of the sample frame. The idle character may be longer or shorter than 16 counts, however the receiver looks for a 1-to-0 transition. The receiver starts to count when the Start bit is set however it does not capture the contents of the RxFIFO at the time the Start bit is set. The start bit is validated when 0s are received for 7 consecutive bit times following the 1-to-0 transition. Once the counter reaches 0xF, it starts counting on the next bit and captures it in the middle of the sampling frame (see Figure 31-3). All data bits are captured in the same manner. Once the stop bit is detected, the receiver shift register (SIPO_OUT) data is parallel shifted to the RxFIFO.

There is a special case when the BRM_CLK period is longer than the period of the IR 0 pulse. In this case, the software sets the IRSC bit so that the i.MX21 reference clock is the clock for the voting logic. The pulse is validated by counting the length of the pulse.

When setting IRSC = 1, either:

- Ignore the first character received, clear the status flags after the RDR bit is set, and proceed,
- OR enables the software reset of the UART module after the initial configuration of UART, however before setting IRSC bit to high. Using this method, insures the first character received is correct.

31.4.9 Binary Rate Multiplier (BRM)

The BRM submodule generates all baud rates that required integer and non-integer division. The input and output frequency ratio is programmed in the UART BRM Incremental Register (UBIR_x) and UART BRM MOD Register (UBMR_x). The output frequency is divided by the input frequency to produce this ratio. For integer division, set the UBIR_x = 0x000F and write the divisor to the UBMR_x register. All values written to these registers must be one less than the actual value to eliminate division by 0 (undefined), and to increase the maximum range of the registers.

Updating the BRM registers requires writing to both registers. The UBIR_x register must be written before writing to the UBMR_x register. If only one register is written to by the software, the BRM continues to use the previous values.

The following examples show how to determine what values are to be programmed into UBIR and UBMR for a given reference frequency and desired baud rate. The following equation can be used to help determine these values:

[(Desired Baud Rate)*16]/(reference frequency)=NUM/DENOM ***Eqn. 31-1***

UBIR = NUM - 1 ***Eqn. 31-2***

UBMR = DENOM - 1 ***Eqn. 31-3***

reference frequency = PERCLK1 / RFDIV [2:0] ***Eqn. 31-4***

Example: Integer Division ÷ 2

Reference Frequency = 19.44 MHz
NUM = 0x0000
DENOM = 0x0001
UBIR = NUM - 1
UBMR = DENOM - 1

NOTE

Notice each value written to the registers is one less than the actual value.

Example: Non-integer Division

Reference Frequency = 16 MHz
Output Frequency = 920 kbps × 16 (sampling) = 14.72 MHz
Ratio --> 14.72 ÷ 16 = 0.92
NUM = 92 (decimal) = 0x5C
DENOM = 100 (decimal) = 0x64
UBIR = NUM - 1
UBMR = DENOM - 1

NOTE

The ratio is derived directly from the division with no factoring (easiest). To derive the exact ratio, some factoring must be performed. Factoring the above ratio produces:

Factored Ratio: 23 ÷ 25
NUM = 22 (decimal) = 0x0016
DENOM = 25 (decimal) = 0x0019

Example: Non-integer Division

Reference Frequency = 25 MHz
Output Frequency = 920 kbps × 16 (sampling) = 14.72 MHz
Ratio --> 14.72 ÷ 25 = 0.5888
NUM = 5888 (decimal) = 0x1700
DENOM = 10000 (decimal) = 0x2710
UBIR = NUM - 1
UBMR = DENOM - 1

NOTE

The ratio is derived directly from the division with no factoring (easiest). To derive the exact ratio, some factoring must be performed. Factoring the above ratio produces:

Factored Ratio = 184 ÷ 313
NUM = 183 (decimal) = 0x00B7
DENOM = 313 (decimal) = 0x0139

Example: Non-integer Division:

Reference Frequency: 30 MHz
Output Frequency: 920 kbps × 16 (sampling) = 14.72 MHz
Ratio --> 14.72 ÷ 30 = 0.4907
NUM = 4907 (decimal) = 0x132B
DENOM = 10000 (decimal) = 0x2710

NOTE

The ratio is derived directly from the division with no factoring (easiest). To derive the exact ratio, some factoring must be perform. Factoring the above ratio produces:

Factored Ratio: 184 ÷ 325
NUM = 183 (decimal) = 0x00B7
DENOM = 325 (decimal) = 0x0144

Baud Rate = (PERCLK1) ÷ (16 × Divisor)
or

Divisor = (PERCLK1) ÷ (16 × Baud Rate)
Transmitter Clock = 16 × Baud Rate
Receiver Clock = PERCLK1 ÷ Divisor = 16 × Baud Rate

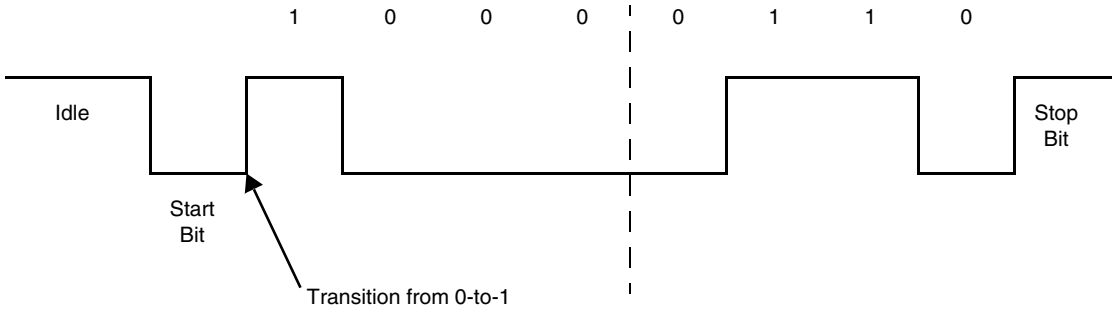
31.4.10 Baud Rate Automatic Detection Logic

When the baud rate automatic detection logic is enabled, the UART locks onto the incoming baud rate. To enable this feature, set the automatic detection of baud rate bit (ADBR = 1) in the UART Control Register 1 and write 1 to the ADET bit in the UART Status Register 2 to clear it. When ADET=0 and ADBR =1, the UART automatically sets the INC=0x0 (UBIR_x) and MOD =0x0(UBMR_x). it waits for the start bit (transition from 1-to-0) and tries to lock onto the incoming baud rate. Once the start bit is detected, the length of the start bit is calculated by counting until the 0-to-1 transition (see Figure 31-4 and Section 31.4.10.1, “Baud Rate Automatic Detection Protocol.”). The new baud rate is determined using this equation:

$$\text{BaudRate} = \langle \frac{\text{Count}}{16} \rangle \qquad \textit{Eqn. 31-5}$$

Table 31-6. Baud Rate Automatic Detection

ADBR	ADET	Baud Rate Detection	UART_INT
0	X	Manual Configuration	1
1	0	Auto Detection	1
1	1	Auto Detection Complete	0
Note: This table assumes that no other interrupt is set at the same time this interrupt is set for the UART_INT signal.			



Note: LSB transmitted first.

Figure 31-4. Baud Rate Detection Protocol Diagram

If any of the UART BRM registers are written to simultaneously by the baud rate automatic detection logic and peripheral data bus, the peripheral data bus has priority.

31.4.10.1 Baud Rate Automatic Detection Protocol

The receiver must receive an ASCII character “A” or “a” to verify proper detection of the incoming baud rate. When an ASCII character “A” or “a” is received and no error occurs, the Automatic Detect baud rate bit is set (ADET=1) and if the interrupt is enabled (ADEN=1),an interrupt is generated.

When an ASCII character “A” or “a” is not received (because of a bit error or the transmission of another character), the value written into the UBIR_x and UBMR_x registers may not be accurate. When the ADET bit is not asserted after the required time-out value (based on baud rate, word size, parity, and number of stop bits), look out for the parity/frame error interrupt if enabled. After the interrupt is asserted, re-send the character “A” or “a” and repeat the above procedure until the ADET bit is set.

As long as ADET = 0 and ADBR = 1, the UART continues to try to lock onto the incoming baud rate. Once the ASCII character “A” or “a” is detected and the ADET bit is set, the receiver ignores the ADBR bit and continues normal operation with the calculated baud rate divisor.

The UART interrupt is active as long as ADET = 1 and ADBR = 1. This can be disabled by clearing the automatic baud rate detection interrupt enable bit (ADEN = 0). Before starting an automatic baud rate detection sequence, set ADET = 0 and ADBR = 1.

The RxFIFO must contain the ASCII character “A” or “a” following the automatic baud rate detection interrupt, which can be cleared by reading it.

The 16-bit UART Baud Rate Count Register (UBRC_x) is reset to 4 and stays at 0xFFFF when an overflow occurs. The UBRC_x register counts the start bit of the incoming baud rate. When the start bit is detected and counted, the UART Baud Rate Count Register retains its value until the next automatic baud rate detection sequence is initiated.

The read only Baud Rate Count Register counts only when auto detection is enabled.

31.4.10.2 Baud Rate Automatic Detection Protocol Improved

Several issues have been reported for ICs using the autobaud protocol like it is described above, especially for 57.6 kbit/s and 115.2 kbit/s. This led to improve this protocol. The old one is still available in the current UART IP, but several modifications can also be used in order to make this autobaud detection more reliable. If the user wants to keep with the old method, it must set the bit ADNIMP (UCR3[7]) to 1. If this bit is not set (default), the autobaud improvements will be used. Those improvements are mainly grouped in two categories: the new baudrate measurement and the new ACST bit (and interrupt).

31.4.10.2.1 New Baudrate Determination

To fight against the problems caused by the distortion and the noise on the RXD line, the duration of the baudrate measurement has been extended. Previously, as described above, this determination was based on the measurement of the START bit duration. Now, this measurement is based on the duration of START bit + bit0. Bit0 is the first bit following the START bit. In fact, the counter which is started at the falling edge of START bit is no longer stopped at next rising edge (end of START bit), but it is stopped at the next falling edge (end of bit0). As the character sent is always a “A” (41h) or a “a” (61h), this second falling edge will be always be present and it will signal the end of bit0. Once this counter is stopped, the result is divided by 2 and used by the BRM to determine the incoming baud rate.

NOTE

UBRC register contains the result of this division by two, in consequence it reflects the measurement of one bit.

31.4.10.2.2 New Autobaud Counter Stopped Bit and Interrupt

A new bit has been added in thUSR2 register: ACST (USR2[11]). This bit is set immediately after the determination of the baud rate.

Therefore:

- if ADNIMP is not set (default), ACST is set to 1 after the end of bit0
- if ADNIMP is set to 1, ACST is set to 1 at the end of START bit

If ACIEN (UCR3[0]) is set to 1, ACST will flag an interrupt on signal. This interrupt informs the MCU the BRM has just been set with the result of the bit length measurement. If needed, the MCU can perform a read of UBMR (or UBRC) register and determine by itself the baudrate measured. Then the MCU has the possibility to correct the BRM registers with the nearest standardized baudrate.

NOTE

- ACST is set only if ADBR is set to 1—that is, the UART is autobauding.
- Clear the ACST bit by writing 1 to it. Writing 0 to the ACST bit has no effect.

31.4.11 Escape Sequence Detection

An escape sequence typically consists of 3 characters entered in rapid succession (such as +++). Because these are valid characters by themselves, the time between characters determines if it is a valid escape sequence. Too much time between two of the + characters is interpreted as two + characters, and not part of an escape sequence.

The software chooses the escape character and writes its value to the UART Escape Character Register (UESC_x). The hardware compares this value to incoming characters in the RxFIFO. When an escape character is detected, the internal escape timer starts to count. The software specifies a time-out value for the maximum allowable time between escape characters. The escape timer is programmable in intervals of 2 msec to a maximum interval of 8.192 seconds.

Table 31-7. Escape Timer Scaling

UTIM_x Register	Maximum Time Between Specified Escape Characters
0x000	2 msec
0x001	4 msec
0x002	6 msec
0x003	8 msec
0x004	10 msec
…	…
0F8	498 msec
0F9	500 msec
…	…
9C3	5 sec

Table 31-7. Escape Timer Scaling (continued)

UTIM_x Register	Maximum Time Between Specified Escape Characters
…	…
FFD	8.188 sec
FFE	8.190 sec
FFF	8.192 sec
Note: To calculate the time interval: (UTIM_Value + 1) × 0.002 = Time_Interval Example: (09C3 + 1) × 0.002 = 5 sec.	

The escape sequence detection feature is available for all the reference frequencies. Before using Escape Sequence Detection, the user must fill the ONEMS register. This 16-bit register must contain the value of the UART internal frequency divided by 1000. The internal frequency is obtained after the UART internal divider.

Example:

- If the UART input clock frequency is 66.5 MHz.,
  And if the UART input clock is divided by 2 with the internal divider: UFCR[9:7] = 3’b100

$$\text{ONEMS} = \frac{66.5\times10^{6}}{2\times1000} = 33250 = 81\text{E2h} \qquad \textit{Eqn. 31-6}$$

The escape sequence detection feature asserts the escape sequence interrupt flag (ESCF) bit when the escape sequence interrupt enable (ESCI) bit is set and an escape sequence is detected. Clear the ESCF bit by writing 1 to it. Writing 0 to the ESCF bit has no effect.

31.5 Infrared Interface

31.5.1 Generalities

The Infrared interface is selected when IREN (UCR1[7]) is set to 1.

The Infrared Interface converts data to be transmitted or received as specified in the IRDA Serial Infrared Physical Layer Specification. In this specification, a 0 is represented by a positive pulse, and a 1 is represented by no pulse (line remains low).

In the UART:

In TX: For each 0 to be transmitted, a narrow positive pulse which is 3/16 of a bit time is generated. For each 1 to be transmitted no pulse is generated (output is low). External circuitry must be provided to drive an Infrared LED.

In RX: When receiving, a narrow negative pulse is expected for each 0 transmitted while no pulse is expected for each 1 transmitted (input is high). Note that Rx part of IR block expects to receive an inverted

signal compared to IRDA specification. Circuitry external to the IC transforms the Infrared signal to an electrical signal.

The IR interface has an edge triggered interrupt (IRINT). This interrupt validates a zero bit being received. This interrupt is enabled by writing a 1 to ENIRI bit.

The behavior of Infrared Interface is determined by 3 bits INVT (UCR3[1]), INVR (UCR4[9]) and IRSC (UCR4[5]).

31.5.2 Inverted Transmission and Reception Bits (INVT and INVR)

The values of INVT and INVR depend of the IRDA transceiver connected on the TXD_IR and RXD_IR pins of the UART. If this transceiver is not inverting on both paths Tx and Rx (like in IRDA specification)—that is, a Zero is represented as a positive pulse and a One is represented by no pulse (line remains low) for TX and RX, the bit INVT must be set to 0 and the bit INVR must be set to 1 (because Rx IR block expects an inverted signal). On the contrary we must have INVT=1 and INVR=0 if both paths of the transceiver are inverting (a Zero is represented as a negative pulse and a One is represented by no pulse (line remains high). The transceiver can also be inverting on only one path (Tx or Rx), in this case INVT and INVR must be together equal to 1 or to 0 (depending on which path is inverted).

31.5.3 InfraRed Special Case (IRSC) Bit

The value to apply to IRSC bit depends essentially of 2 parameters: the baud rate and the Minimum Pulse Duration (MPD) of the of the transceiver. As already written, in IRDA a Zero is represented by a positive pulse. The IRDA specification says that for SIR (Serial IR) baudrates (from 2.4 kbit/s to 115.2 kbit/s) this nominal pulse duration is equal to 3/16 of a bit duration (at the baudrate selected). But, for all the baud rates a Minimum Pulse Duration is also specified. For SIR, the MPD is constant and equal to 1.41 us.

In order to understand the meaning of bit IRSC, me must have an idea of how works the Rx path in IRDA.

In the UART module when in IRDA, a Zero is not only detected by the state of the RXD_IR line, but also with the duration of the pulse. This pulse duration can be measured with 2 different clocks. The choice of the clock is done with IRSC bit.

If IRSC = 0, the clock used is the BRM clock.

If IRSC = 1, the clock used is the UART internal clock (UART clock after the divider (RFDIV)).

In normal operation, IRSC=0. This means at any time, the user must be sure the frequency of BRM_clock is high enough to measure the pulse. In the UART module and for IRSC=0, the pulse must last **at least 2 BRM clock cycles**.

If this condition is not fulfilled, IRSC must be set to 1.

Let’s take 2 examples, with the Minimum Pulse Duration equals to 2us.

Example #1: The user wants to receive IRDA data at 115.2 kbit/s. The UBIR and UBMR registers are set in order to create the BRM_clock with a frequency of 16*baudrate = 16 * 115.2 = 1.8432 MHz. But at the same time, in order to correctly detect the pulse, the user must be sure that 2* BRM_clock period is lower than 2us. Lets check:

BRM_clock period = 1/1843200 = 542 ns

So 2*BRM_clock period = 1.09 us < 2 us. It is fine.

Example #2: This time the user wants to receive data at 19.2 kbit/s. So, the BRM_clock is set to 16*19200 = 307.2 kHz. Let's check if 2* BRM_clock period < 2 us:

BRM_clock period =1/307200 = 3.255 us

So 2*BRM_clock period = 6.51 us >> 2 us. It does not work.

Therefore in this situation, the BRM clock cannot be used to measure the pulse duration, and the user must select the UART internal clock by setting IRSC =1.

NOTE

Before using IR Special Case (IRSC=1), the user must fill the ONEMS register.

When IRSC is set to 1, the nominal MPD of the UART is 1us.

31.5.4 IRDA Interrupt

Serial infrared mode (SIR) uses an edge triggered interrupt flag (the serial infrared interrupt flag, IRINT, in the USR2_x register) that validates 0 bits as they are received. When INVR =0, detection of a falling edge on the UART_RXD pin assets the IRINT bit. When INVR=1, detection of a rising edge on the UART_RXD pin assets the IRINT bit. When both the IRINT and ENIRI bits are asserted, the interrupt is asserted. Clear the IRINT bit by writing 1 to it. Writing 0 to the IRINT bit has no effect.

31.5.5 Conclusion about IRDA

Before using the UART in IRDA, the baud rate limit must be calculated. This baud rate limit will inform the user if IRSC bit must be set or not.

Let's determine this limit:

As already written, if IRSC = 0 the following condition must always be fulfilled:

$$2 \times \mathrm{BRMClockPeriod} < \mathrm{MinPulseDuration}$$

So,

$$\mathrm{BRMClockFrequency} > \frac{2}{\mathrm{MPD}}$$

So, knowing BRM_clock frequency = 16 * Baudrate, we get:

$$\mathrm{BaudRate} > \frac{1}{8 \times \mathrm{MinPulseDuration}}$$

So, the user needs to set IRSC = 0 when:

If Minimum Pulse Duration = 2.5 us and Baudrate > 50 kbit/s.

If Minimum Pulse Duration = 2.0 us and Baudrate > 62.5 kbit/s.

If Minimum Pulse Duration = 1.41 us and Baudrate > 88.6 kbit/s.

For baud rates lower than the limit, IRSC must be set to 1.

31.6 Programming Model

The UARTx module includes 16 user-accessible 32-bit registers. All registers use word, halfword, or byte access. Table 31-8 summarizes these registers and their addresses.

Table 31-8. UART Module Register Summary

Description	Name	Address
UART1 Receiver Register UART2 Receiver Register UART3 Receiver Register UART4 Receiver Register	URXD_1 URXD_2 URXD_3 URXD_4	0x1000_A000 0x1000_B000 0x1000_C000 0x1000_D000
UART1 Transmitter Register UART2 Transmitter Register UART3 Transmitter Register UART4 Transmitter Register	UTXD_1 UTXD_2 UTXD_3 UTXD_4	0x1000_A040 0x1000_B040 0x1000_C040 0x1000_D040
UART1 Control Register 1 UART2 Control Register 1 UART3 Control Register 1 UART4 Control Register 1	UCR1_1 UCR1_2 UCR1_3 UCR1_4	0x1000_A080 0x1000_B080 0x1000_C080 0x1000_D080
UART1 Control Register 2 UART2 Control Register 2 UART3 Control Register 2 UART4 Control Register 2	UCR2_1 UCR2_2 UCR2_3 UCR2_4	0x1000_A084 0x1000_B084 0x1000_C084 0x1000_D084
UART1 Control Register 3 UART2 Control Register 3 UART3 Control Register 3 UART4 Control Register 3	UCR3_1 UCR3_2 UCR3_3 UCR3_4	0x1000_A088 0x1000_B088 0x1000_C088 0x1000_D088
UART1 Control Register 4 UART2 Control Register 4 UART3 Control Register 4 UART4 Control Register 4	UCR4_1 UCR4_2 UCR4_3 UCR4_4	0x1000_A08C 0x1000_B08C 0x1000_C08C 0x1000_D08C
UART1 FIFO Control Register UART2 FIFO Control Register UART3 FIFO Control Register UART4 FIFO Control Register	UFCR_1 UFCR_2 UFCR_3 UFCR_4	0x1000_A090 0x1000_B090 0x1000_C090 0x1000_D090
UART1 Status Register 1 UART2 Status Register 1 UART3 Status Register 1 UART4 Status Register 1	USR1_1 USR1_2 USR1_3 USR1_4	0x1000_A094 0x1000_B094 0x1000_C094 0x1000_D094
UART1 Status Register 2 UART2 Status Register 2 UART3 Status Register 2 UART4 Status Register 2	USR2_1 USR2_2 USR2_3 USR2_4	0x1000_A098 0x1000_B098 0x1000_C098 0x1000_D098

Table 31-8. UART Module Register Summary (continued)

Description	Name	Address
UART1 Escape Character Register UART2 Escape Character Register UART3 Escape Character Register UART4 Escape Character Register	UESC_1 UESC_2 UESC_3 UESC_4	0x1000_A09C 0x1000_B09C 0x1000_C09C 0x1000_D09C
UART1 Escape Timer Register UART2 Escape Timer Register UART3 Escape Timer Register UART4 Escape Timer Register	UTIM_1 UTIM_2 UTIM_3 UTIM_4	0x1000_A0A0 0x1000_B0A0 0x1000_C0A0 0x1000_D0A0
UART1 BRM Incremental Register UART2 BRM Incremental Register UART3 BRM Incremental Register UART4 BRM Incremental Register	UBIR_1 UBIR_2 UBIR_3 UBIR_4	0x1000_A0A4 0x1000_B0A4 0x1000_C0A4 0x1000_D0A4
UART1 BRM Modulator Register UART2 BRM Modulator Register UART3 BRM Modulator Register UART4 BRM Modulator Register	UBMR_1 UBMR_2 UBMR_3 UBMR_4	0x1000_A0A8 0x1000_B0A8 0x1000_C0A8 0x1000_D0A8
UART1 Baud Rate Count Register UART2 Baud Rate Count Register UART3 Baud Rate Count Register UART4 Baud Rate Count Register	UBRC_1 UBRC_2 UBRC_3 UBRC_4	0x1000_A0AC 0x1000_B0AC 0x1000_C0AC 0x1000_D0AC
UART1 One Millisecond Register UART2 One Millisecond Register UART3 One Millisecond Register UART4 One Millisecond Register	ONEMS_1 ONEMS_2 ONEMS_3 ONEMS_4	0x1000_A0B0 0x1000_B0B0 0x1000_C0B0 0x1000_D0B0
UART1 Test Register 1 UART2 Test Register 1 UART3 Test Register 1 UART4 Test Register 1	UTS_1 UTS_2 UTS_2 UTS_4	0x1000_A0B4 0x1000_B0B4 0x1000_C0B4 0x1000_D0B4

31.6.1 UART Receiver Registers

The read-only UART Receiver Registers contain the received character and its status. These registers allow 8-bit data read for DMA transfer. When DMA is configured to 8-bit port reading from URXD_x, the status bits are ignored and only RX_DATA are transferred. When DMA is configured to 16-bit port reading from URXD_x, the status bits and RX_DATA are all obtained. After reset, when the receiver enable bit is set (RXEN = 1), these registers contain random data and the CHARRDY bit is 0 until the first character is received.

		Addr
URXD_1	UART1 Receiver Register	0x1000_A000
URXD_2	UART2 Receiver Register	0x1000_B000 0x1000_C000
URXD_3	UART3 Receiver Register	0x1000_D000
URXD_4	UART4 Receiver Register	

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CHARRDY	ERR	OVRRUN	FRMERR	BRK	PRERR			RX_DATA							
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?
	0x0000															

Table 31-9. UART1 through UART4 Receiver Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
CHARRDY Bit 15	**Character Ready**—Indicates an invalid read when the RxFIFO is empty and the software attempts to read the previously read data.	0 = The character in the RX_DATA field and its associated flags are invalid. 1 = The character in the RX_DATA field and its associated flags are valid and ready to read
ERR Bit 14	**Error Detect**—Indicates whether the character present in the RX_DATA field has an error (OVRRUN, FRMERR, BRK or PRERR) status. The ERR bit is updated and valid for each received character.	0 = An error status was detected 1 = No error status was detected
OVRRUN Bit 13	**Receiver Overrun**—Indicates whether the receiver ignored data to prevent overwriting the data in the RxFIFO. This error indicates that the software is not keeping up with the incoming data rate. OVRRUN is set for the last (32nd) character written to the RxFIFO to indicate that all characters following this character will be ignored if a read is not performed by the software. OVRRUN is updated and valid for each received character. Under normal circumstances, OVRRUN is never set.	0 = No RxFIFO overrun was detected 1 = A RxFIFO overrun was detected
FRMERR Bit 12	**Frame Error**—Indicates whether the current character had a framing error (a missing stop bit) and is possibly corrupted. FRMERR is updated for each character read from the RxFIFO.	0 = The current character has no framing error 1 = The current character has a framing error

Table 31-9. UART1 through UART4 Receiver Register Description (continued)

Name	Description	Settings
BRK Bit 11	**BREAK Detect**—Indicates whether the current character was detected as a BREAK character. The data bits and the stop bit are all 0. The FRMERR bit is set when BRK is set. When odd parity is selected, PRERR is also set when BRK is set. BRK is valid for each character read from the RxFIFO.	0 = The current character is not a BREAK character 1 = The current character is a BREAK character
PRERR Bit 10	**Parity Error**—Indicates whether the current character was detected with a parity error and is possibly corrupted. PRERR is updated for each character read from the RxFIFO. When parity is disabled, PRERR always reads as 0.	0 = No parity error was detected for data in the RX_DATA field 1 = A parity error was detected for data in the RX_DATA field
Reserved Bits 9–8	Reserved—These bits are reserved and should read 0.	
RX_DATA Bits 7–0	**Received Data**—Holds the received character. In 7-bit mode, the most significant bit (MSB) is forced to 0. In 8-bit mode, all bits are active. Support 8-bit data read for DMA transfer. When DMA is configured to 8-bit port reading from RxFIFO, the status bits are ignored and only RX_DATA are transferred.	

31.6.2 UART Transmitter Registers

The write-only UART Transmitter Registers are where the ARM926EJ-S processor writes the data to be transmitted. When using DMA to transfer data to txfifo, these registers should be configured as 8-bit ports. When these registers are read, the TX_DATA bits are always read as 0.

		Addr
UTXD_1	UART1 Transmitter Register	0x1000_A040
UTXD_2	UART2 Transmitter Register	0x1000_B040 0x1000_C040
UTXD_3	UART3 Transmitter Register	0x1000_D040
UTXD_4	UART4 Transmitter Register	

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									TX_DATA							
TYPE	r	r	r	r	r	r	r	r	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?
	0x0000															

Table 31-10. UART1 through UART4 Transmitter Register Description

Name	Description
Reserved Bits 31–8	Reserved—These bits are reserved and should read 0.
TX_DATA Bits 7–0	**Transmit Data**—Holds the parallel transmit data inputs. In 7-bit mode, D7 is ignored. In 8-bit mode, all bits are used. Data is transmitted least significant bit (LSB) first. A new character is transmitted when the TX_DATA field is written. The TX_DATA field must be written only when the TRDY bit is high to ensure that corrupted data is not sent.

31.6.3 UART Control Register 1

The UART Control Register 1s enable the UART and the transmit and receive blocks. They controls the TxFIFO and RxFIFO levels and enables the TRDY and RRDY interrupts.

		Addr
UCR1_1	UART1 Control Register 1	0x1000_A080
UCR1_2	UART2 Control Register 1	0x1000_B080
UCR1_3	UART3 Control Register 1	0x1000_C080
UCR1_4	UART4 Control Register 1	0x1000_D080

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADEN	ADBR	TRDYEN	IDEN	ICD		RRDYEN	RXDMAEN	IREN	TXMPTYEN	RTSDEN	SNDBRK	TXDMAEN		DOZE	UARTEN
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 31-11. UART1 through UART4 Control Register 1 Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
ADEN Bit 15	**Automatic Baud Rate Detection Interrupt Enable**—Enables or disables the automatic baud rate detect complete (ADET) bit to generate an interrupt.	0 = Disable the automatic baud rate detection interrupt 1 = Enable the automatic baud rate detection interrupt
ADBR Bit 14	**Automatic Detection of Baud Rate**—Enables/Disables automatic baud rate detection. When the ADBR bit is set and the ADET bit is cleared, the receiver detects the incoming baud rate automatically. The ADET flag is set when the receiver verifies that the incoming baud rate is detected properly by detecting an ASCII character “A” or “a” (0x61 or 0x41).	0 = Disable automatic detection of baud rate 1 = Enable automatic detection of baud rate

Table 31-11. UART1 through UART4 Control Register 1 Description (continued)

Name	Description	Settings
TRDYEN Bit 13	**Transmitter Ready Interrupt Enable**—Enables/Disables the transmitter Ready Interrupt (TRDY) when the transmitter has one or more slots available in the TxFIFO. The fill level in the TXFIFO at which an interrupt is generated is controlled by TxTL bits. When TRDYEN is negated, the transmitter ready interrupt is disabled.	0 = Disable the transmitter ready interrupt 1 = Enable the transmitter ready interrupt
IDEN Bit 12	**Idle Condition Detected Interrupt Enable**—Enables/Disables the IDLE bit to generate an interrupt.	0 = Disable the IDLE bit 1 = Enable the IDLE bit
ICD Bits 11–10	**Idle Condition Detect**—Controls the number of frames RXD is allowed to be idle before an idle condition is reported.	00 = Idle for more than 4 frames 01 = Idle for more than 8 frames 10 = Idle for more than 16 frames 11 = Idle for more than 32 frames
RRDYEN Bit 9	**Receiver Ready Interrupt Enable**—Enables/Disables the RRDY interrupt when the RxFIFO contains data. The fill level in the RXFIFO at which an interrupt is generated is controlled by the RXTL bits. When RRDYEN is negated, the receiver ready interrupt is disabled.	0 = Disables the RRDY interrupt 1 = Enables the RRDY interrupt
RXDMAEN Bit 8	**Receive Ready DMA Enable**—Enables/Disables the receive DMA request $\overline{\text{IPD\_UART\_RX\_DMAREQ}}$ when the receiver has data in the RxFIFO. The fill level in the RxFIFO at which a DMA request is generated is controlled by the RXTL bits. When negated, the receive DMA request is disabled.	0 = Disable $\overline{\text{IPD\_UART\_RX\_DMAREQ}}$ DMA request 1 = Enable $\overline{\text{IPD\_UART\_RX\_DMAREQ}}$ DMA request
IREN Bit 7	**Infrared Interface Enable**—Enables/Disables the IR interface. See the IR interface description in Section 31.5 for more information.	0 = Disable the IR interface 1 = Enable the IR interface
TXMPTYEN Bit 6	**Transmitter Empty Interrupt Enable**—Enables/Disables the transmitter FIFO empty (TXFE) interrupt. When negated, the TXFE interrupt is disabled.	0 = Disable the transmitter FIFO empty interrupt 1 = Enable the transmitter FIFO empty interrupt
RTSDEN Bit 5	**RTS Delta Interrupt Enable**—Enables/Disables the RTSD interrupt. The current status of the $\overline{\text{UARTx\_RTS}}$ pin is read in the RTSS bit.	0 = Disable RTSD interrupt 1 = Enable RTSD interrupt
SNDBRK Bit 4	**Send Break**—Forces the transmitter to send a BREAK character. The transmitter finishes sending the character in progress (if any) and sends BREAK characters until SNDBRK is reset. Because the transmitter samples SNDBRK after every bit is transmitted, it is important that SNDBRK is asserted high for a sufficient period of time to generate a valid BREAK. After the BREAK transmission completes, the UART transmits 2 mark bits. The user can continue to fill the TxFIFO and any characters remaining are transmitted when the BREAK is terminated.	0 = Do not send a BREAK character 1 = Send a BREAK character (continuous 0s)
TXDMAEN Bit 3	**Transmitter Ready DMA Enable**—Enables/Disables the transmit DMA request $\overline{\text{IPD\_UART\_TX\_DMAREQ}}$ when the transmitter has one or more slots available in the TxFIFO. The fill level in the TxFIFO that generates the $\overline{\text{IPD\_UART\_TX\_DMAREQ}}$ is controlled by the TXTL bits.	0 = Disable transmit DMA request 1 = Enable transmit DMA request
Reserved Bit 2	Reserved—Read as zero and should be written with zero for future compatibility.	

Table 31-11. UART1 through UART4 Control Register 1 Description (continued)

Name	Description	Settings
DOZE Bit 1	**DOZE**—Determines the UART enable condition in the DOZE state. Not used in i.MX21. should be write and read as 0.	0 = The UART is enabled when in DOZE state 1 = The UART is disabled when in DOZE state
UARTEN Bit 0	**UART Enable**—Enables/Disables the UART. If UARTEN is negated in the middle of a transmission, the transmitter stops and pulls the TXD line to a logic 1. UARTEN must be set to 1 before any access to UTXD and URXD registers.	0 = Disable the UART 1 = Enable the UART

31.6.4 UART Control Register 2

The UART Control Register 2s control the overall operation of the UART. The bits in these registers set the BITSEL and its source, specify the number of bits per character, enable or disable parity generation and checking, and control the $\overline{RTS}$ and $\overline{CTS}$ pins.

		Addr
UCR2_1	UART1 Control Register 2	0x1000_A084
UCR2_2	UART2 Control Register 2	0x1000_B084
UCR2_3	UART3 Control Register 2	0x1000_C084
UCR2_4	UART4 Control Register 2	0x1000_D084

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ESCI	IRTS	CTSC	CTS	ESC EN	RTEC		PREN	PROE	STPB	WS	RTS EN	ATE N	TXEN	RXEN	$\overline{SRST}$
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	0x0001															

Table 31-12. UART1 through UART4 Control Register 2 Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
ESCI Bit 15	**Escape Sequence Interrupt Enable**—Enables/Disables the ESCF bit to generate an interrupt.	0 = Disable the escape sequence interrupt 1 = Enable the escape sequence interrupt

Table 31-12. UART1 through UART4 Control Register 2 Description (continued)

Name	Description	Settings
IRTS Bit 14	**Ignore UARTx_RTS Pin**—Forces the UARTx_RTS input signal presented to the transmitter to always be asserted, effectively ignoring the external pin. When in this mode, the UARTx_RTS pin serves as a general purpose input.	0 = Transmit only when the UARTx_RTS pin is asserted 1 = Ignore the UARTx_RTS pin
CTSC Bit 13	**UARTx_CTS Pin Control**—Controls the operation of the UARTx_CTS output pin. When CTSC is asserted, the UARTx_CTS output pin is controlled by the receiver. When the RxFIFO is filled to the level of the programmed trigger level and the start bit of the overflowing character (TRIGGER LEVEL + 1) is validated, the UARTx_CTS output pin is negated to indicate to the far-end transmitter to stop transmitting. When the trigger level is programmed for less than 32, the receiver continues to receive data until the RxFIFO is full. When the CTSC bit is negated, the UARTx_CTS output pin is controlled by the CTS bit. On reset, because CTSC is cleared to 0, the UARTx_CTS pin is controlled by the CTS bit, which again is cleared to 0 on reset. This means that on reset the UARTx_CTS signal is negated.	0 = The UARTx_CTS pin is controlled by the CTS bit 1 = The UARTx_CTS pin is controlled by the receiver
CTS Bit 12	**Clear to Send**—Controls the UARTx_CTS pin when the CTSC bit is negated. CTS has no function when CTSC is asserted.	0 = The UARTx_CTS pin is high (inactive) 1 = The UARTx_CTS pin is low (active)
ESCEN Bit 11	**Escape Enable**—Enables/Disables the escape sequence detection logic.	0 = Disable escape sequence detection 1 = Enable escape sequence detection
RTEC Bits 10–9	**Request to Send Edge Control**—Selects the edge that triggers the RTS interrupt. This has no effect on the RTS delta interrupt. RTEC has an effect only when RTSEN = 1 (see Table 31-3).	00 = Trigger interrupt on a rising edge 01 = Trigger interrupt on a falling edge 1X = Trigger interrupt on any edge
PREN Bit 8	**Parity Enable**—Enables/Disables the parity generator in the transmitter and parity checker in the receiver. When PREN is asserted, the parity generator and checker are enabled, and disabled when PREN is negated.	0 = Disable parity generator and checker 1 = Enable parity generator and checker
PROE Bit 7	**Parity Odd/Even**—Controls the sense of the parity generator and checker. When PROE is high, odd parity is generated and expected. When PROE is low, even parity is generated and expected. PROE has no function if PREN is low.	0 = Even parity 1 = Odd parity
STPB Bit 6	**Stop**—Controls the number of stop bits transmitted after a character. When STPB is high, 2 stop bits are sent. When STPB is low, 1 stop bit is sent. STPB has no effect on the receiver, which expects 1 or more stop bits.	0 = 1 stop bit transmitted 1 = 2 stop bits transmitted
WS Bit 5	**Word Size**—Controls the character length. When WS is high, the transmitter and receiver are in 8-bit mode. When WS is low, they are in 7-bit mode. The transmitter ignores bit 7 and the receiver sets bit 7 to 0. WS can be changed in-between transmission (reception) of characters, however not when a transmission (reception) is in progress, in which case the length of the current character being transmitted (received) is unpredictable.	0 = 7-bit transmit and receive character length (not including START, STOP or PARITY bits) 1 = 8-bit transmit and receive character length (not including START, STOP or PARITY bits)
RTSEN Bit 4	**Request to Send Interrupt Enable**—Controls the RTS edge sensitive interrupt. When RTSEN is asserted and the programmed edge is detected on the UARTx_RTS pin, the RTSF bit is asserted (see Table 31-3).	0 = Disable request to send interrupt 1 = Enable request to send interrupt

Table 31-12. UART1 through UART4 Control Register 2 Description (continued)

Name	Description	Settings
ATEN Bit 3	**Aging Timer Enable**—This bit is used to mask the aging timer interrupt (triggered with AGTIM).	0 = AGTIM interrupt disabled 1 = AGTIM interrupt enabled
TXEN Bit 2	**Transmitter Enable**—Enables/Disables the transmitter. When TXEN is negated the transmitter is disabled and idle. When the UARTEN and TXEN bits are set the transmitter is enabled. If TXEN is negated in the middle of a transmission, the UART disables the transmitter immediately.	0 = Disable the transmitter 1 = Enable the transmitter
RXEN Bit 1	**Receiver Enable**—Enables/Disables the receiver. When the receiver is enabled, if the RXD input is already low, the receiver does not recognize BREAK characters, because it requires a valid 1-to-0 transition before it can accept any character.	0 = Disable the receiver 1 = Enable the receiver
$\overline{\text{SRST}}$ Bit 0	**Software Reset**—Resets the transmitter and receiver state machines, all FIFOs, and all status registers. Once the software writes 0 to $\overline{\text{SRST}}$, the software reset remains active for 4 clock cycles of ipg_clk before the hardware deasserts $\overline{\text{SRST}}$. The software can only write 0 to $\overline{\text{SRST}}$. Writing 1 to $\overline{\text{SRST}}$ is ignored.	0 = Reset the transmit and receive state machines, all FIFOs and all status registers 1 = No reset

31.6.5 UART Control Register 3

The UART Control Register 3s control the overall operation of the UART.

		Addr
UCR3_1	UART1 Control Register 3	0x1000_A088
UCR3_2	UART2 Control Register 3	0x1000_B088
UCR3_3	UART3 Control Register 3	0x1000_C088
UCR3_4	UART4 Control Register 3	0x1000_D088

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000														

BIT	15 14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DPEC	DTREN	PARERR EN	FRAERR EN	DSR	DCD	RI	ADNIMP	RXDS EN	AIR INTEN	AWAK EN		RXDMUXSEL	INVT	ACIEN
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000														

Table 31-13. UART1 through UART4 Control Register 3 Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
DPEC Bit 15–14	**DTR Interrupt Edge Control**—These bits control the edge on which an interrupt will be generated. An interrupt will only be generated if the DTREN bit is set. Not used in i.MX21. should be write and read as 0.	00 = interrupt generated on rising edge 01 = interrupt generated on falling edge 1X = interrupt generated on either edge
DTREN Bit 13	**Data Terminal Ready Interrupt Enable**—When this bit is set, it will enable the status bit DTRF (USR2 [13]) (DTR edge sensitive interrupt) to cause an interrupt. Not used in i.MX21. should be write and read as 0.	0 = Data Terminal Ready Interrupt Disabled 1 = Data Terminal Ready Interrupt Enabled
PARERREN Bit 12	**Parity Error Interrupt Enable**—Enables/Disables the interrupt. When asserted, PARERREN causes the PARITYERR bit to generate an interrupt.	0 = Disable the parity error interrupt 1 = Enable the parity error interrupt
FRAERREN Bit 11	**Frame Error Interrupt Enable**—Enables/Disables the interrupt. When asserted, FRAERREN causes the FRAMERR bit to generate an interrupt.	0 = Disable the frame error interrupt 1 = Enable the frame error interrupt
DSR Bit 10	**Data Set Ready**—This bit is used by software to control the $\overline{\text{DSR}}$/$\overline{\text{DTR}}$ output pin for the modem interface.	0 = $\overline{\text{DSR}}$/ $\overline{\text{DTR}}$ pin is logic zero 1 = $\overline{\text{DSR}}$/ $\overline{\text{DTR}}$ pin is logic one

Table 31-13. UART1 through UART4 Control Register 3 Description (continued)

Name	Description	Settings
DCD Bit 9	**Data Carrier Detect**—In DCE mode this bit is used by software to control the $\overline{DCD}$ output pin for the modem interface.	0 = $\overline{DCD}$ pin is logic zero (DCE mode) 1 = $\overline{DCD}$ pin is logic one (DCE mode)
RI Bit 8	**Ring Indicator**—In DCE mode this bit is used by software to control the $\overline{RI}$ output pin for the modem interface.	0 = $\overline{RI}$ pin is logic zero (DCE mode) 1 = $\overline{RI}$ pin is logic one (DCE mode)
ADNIMP Bit 7	**Autobaud Detection Not Improved**—Disables new features of autobaud detection (see Section 31.4.10.2, “Baud Rate Automatic Detection Protocol Improved,” on page -16 for more details).	0 = Autobaud detection new features selected 1 = Keep old autobaud detection mechanism
RXDSEN Bit 6	**Receive Status Interrupt Enable**—Controls the receive status interrupt. When this bit is enabled and RXDS status bit is set, the interrupt will be generated.	0 = Disable the RXDS interrupt 1 = Enable the RXDS interrupt
AIRINTEN Bit 5	**Asynchronous IR WAKE Interrupt Enable**—Controls the asynchronous IR WAKE interrupt. An interrupt is generated when AIRINTEN is asserted and a pulse is detected on the UART_RX pin.	0 = Disable the AIRINT interrupt 1 = Enable the AIRINT interrupt
AWAKEN Bit 4	**Asynchronous WAKE Interrupt Enable**—Controls the asynchronous WAKE interrupt. An interrupt is generated when AWAKEN is asserted and a falling edge is detected on the RXD pin.	0 = Disable the AWAKE interrupt 1 = Enable the AWAKE interrupt
Reserved Bit 3	Reserved—Read as zero and should be written with zero for future compatibility.	
RXDMUXSEL Bit 2	**RXD Muxed Input Selected**—Selects the IPP_UART_RXD_MUX input pin for serial and Infrared input signal In i.MX21. this bit should be set 1’b1.	0 = Serial input pin is IPP_UART_RXD and IR input pin is IPP_UART_RXD_IR 1 = Input pin is IPP_UART_RXD_MUX for serial and IR interfaces
INVT Bit 1	**Inverted Infrared Transmission**—Sets the active level for the transmission. When INVT is cleared, the infrared logic block transmits a positive IR 3/16 pulse for all 0s and 0s are transmitted for 1s. When INVT is set (INVT = 1), the infrared logic block transmits an active low or negative infrared 3/16 pulse for all 0s and 1s are transmitted for 1s.	0 = Active low transmission 1 = Active high transmission
ACIEN Bit 0	**Autobaud Counter Interrupt Enable**—This bit is used to mask the autobaud counter stopped interrupt.	0 = ACST interrupt disabled 1 = ACST interrupt enabled

31.6.6 UART Control Register 4

The UART Control Register 4s control the operation of some of the UART interrupts.

		Addr
UCR4_1	UART1 Control Register 4	0x1000_A08C
UCR4_2	UART2 Control Register 4	0x1000_B08C
UCR4_3	UART3 Control Register 4	0x1000_C08C
UCR4_4	UART4 Control Register 4	0x1000_D08C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CTSTL						INVR	ENIRI	WKEN		IRSC	LPBYP	TCEN	BKEN	OREN	DREN
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	rw	rw	rw	rw	rw	rw
RESET	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x8000															

Table 31-14. UART1 through UART4 Control Register 4 Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
CTSTL Bits 15–10	**CTS Trigger Level**—Controls the threshold at which the CTS pin is deasserted by the RxFIFO. After the trigger level is reached and the CTS pin is deasserted, the RxFIFO continues to receive data until it is full. The CTSTL bits are encoded as shown in the Settings column.	000000 = 0 characters received 000001 = 1 characters in the RxFIFO ... 100000 = 32 characters in the RxFIFO (maximum) All Other Settings Reserved
INVR Bit 9	**Inverted Infrared Reception**—Determines the logic level for the detection. When cleared, the infrared logic block expects an active low or negative IR 3/16 pulse for 0s and 1s are expected for 1s. When INVR is set (INVR = 1), the infrared logic block expects an active high or positive IR 3/16 pulse for 0s and 0s are expected for 1s.	0 = Active low detection 1 = Active high detection
ENIRI Bit 8	**Serial Infrared Interrupt Enable**—Enables/Disables the serial infrared interrupt.	0 = Serial infrared Interrupt disabled 1 = Serial infrared Interrupt enabled
WKEN Bit 7	**WAKE Interrupt Enable**—Enables/Disables the WAKE bit to generate an interrupt. The WAKE bit is set at the detection of a start bit by the receiver.	0 = Disable the WAKE interrupt 1 = Enable the WAKE interrupt
Reserved Bit 6	Reserved— Read as zero and should be written with zero for future compatibility.	N/A

Table 31-14. UART1 through UART4 Control Register 4 Description (continued)

Name	Description	Settings
IRSC Bit 5	**IR Special Case**—Selects the clock for the vote logic. When set, IRSC switches the vote logic clock from the sampling clock to the UART reference clock. The IR pulses are counted a predetermined amount of time depending on the reference frequency.	0 = The vote logic uses the sampling clock (16x baud rate) for normal operation 1 = The vote logic uses the UART reference clock
LPBYP Bit 4	**Low Power Bypass**—Allows to bypass the low power new features in UART for i.MX21. To use during debug phase.	0 = Low power features enabled 1 = Low power features disabled
TCEN Bit 3	**Transmit Complete Interrupt Enable**—Enables/Disables the TXDC bit to generate an interrupt.	0 = Disable TXDC interrupt 1 = Enable TXDC interrupt
BKEN Bit 2	**BREAK Condition Detected Interrupt Enable**—Enables/Disables the BRCD bit to generate an interrupt.	0 = Disable the BRCD interrupt 1 = Enable the BRCD interrupt
OREN Bit 1	**Receiver Overrun Interrupt Enable**—Enables/Disables the ORE bit to generate an interrupt.	0 = Disable ORE interrupt 1 = Enable ORE interrupt
DREN Bit 0	**Receive Data Ready Interrupt Enable**—Enables/Disables the RDR bit to generate an interrupt.	0 = Disable RDR interrupt 1 = Enable RDR interrupt

31.6.7 UART FIFO Control Registers

The UART FIFO Control Registers control the operation of the UART FIFO trigger levels and interrupts.

		Addr
UFCR_1	UART1 FIFO Control Register	0x1000_A090
UFCR_2	UART2 FIFO Control Register	0x1000_B090
UFCR_3	UART3 FIFO Control Register	0x1000_C090
UFCR_4	UART4 FIFO Control Register	0x1000_D090

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXTL						RFDIV			DCEDTE	RXTL					
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
	0x0801															

Table 31-15. UART1 through UART4 FIFO Control Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
TXTL Bits 15–10	**Transmitter Trigger Level**—Controls the threshold at which a maskable interrupt is generated by the TxFIFO. A maskable interrupt is generated whenever the data level in the TxFIFO falls below the selected threshold. The bits are encoded as shown in the Settings column.	000000 = Reserved 000001 = Reserved 000010 = TxFIFO has 2 or fewer characters ... 011111 = TxFIFO has 31 or fewer characters 100000 = TxFIFO has 32 characters (maximum) All Other Settings Reserved
RFDIV Bits 9–7	**Reference Frequency Divider**—Controls the divide ratio for the reference clock. The input clock is IPG_CLK. The output from the divider must be synchronous with the bus clock.	000 = Divide input clock by 6 001 = Divide input clock by 5 010 = Divide input clock by 4 011 = Divide input clock by 3 100 = Divide input clock by 2 101 = Divide input clock by 1 110 = Divide input clock by 7
DCEDTE Bit 6	**DCE/DTE Mode Select**—Selects DCE (Data Communication Equipment) or DTE (Data terminal Equipment) mode.	0 = DCE mode selected 1 = DTE mode selected
RXTL Bits 5–0	**Receiver Trigger Level**—Controls the threshold at which a maskable interrupt is generated by the RxFIFO. A maskable interrupt is generated whenever the data level in the RxFIFO reaches the selected threshold. The RXTL bits are encoded as shown in the Settings column.	000000 = 0 characters received 000001 = RxFIFO has 1 character ... 011111 = RxFIFO has 31 characters 100000 = RxFIFO has 32 characters (maximum) All Other Settings Reserved

31.6.8 UART Status Register 1

The UART Status Register 1s report errors and status flags.

		Addr
USR1_1	UART1 Status Register 1	0x1000_A094
USR1_2	UART2 Status Register 1	0x1000_B094
USR1_3	UART3 Status Register 1	0x1000_C094
USR1_4	UART4 Status Register 1	0x1000_D094

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PARITY ERR	RTSS	TRDY	RTSD	ESCF	FRAM ERR	RRDY	AGTIM		RXDS	AIR INT	AWAKE				
TYPE	rw	r	r	rw	rw	rw	r	rw	r	r	rw	rw	r	r	r	r
RESET	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
	0x2040															

Table 31-16. UART1 through UART4 Status Register 1 Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
PARITYERR Bit 15	**Parity Error Interrupt Flag**—Indicates a parity error is detected. PARITYERR is cleared by writing 1 to it. Writing 0 to PARITYERR has no effect. When parity is disabled, PARITYERR always reads 0. At reset, PARITYERR is set to 0.	0 = No parity error detected 1 = Parity error detected
RTSS Bit 14	**$\overline{\text{RTS}}$ Pin Status**—Indicates the current status of the $\overline{\text{UARTx\_RTS}}$ pin. A “snapshot” of the pin is taken immediately before RTSS is presented to the data bus. RTSS cannot be cleared because all writes to RTSS are ignored. At reset, RTSS is set to 0.	0 = The $\overline{\text{UARTx\_RTS}}$ pin is high (inactive) 1 = The $\overline{\text{UARTx\_RTS}}$ pin is low (active)
TRDY Bit 13	**Transmitter Ready Interrupt / DMA Flag**—Indicates that the TxFIFO emptied below its target threshold and requires data. TRDY is automatically cleared when the data level in the TxFIFO goes beyond above the set threshold level by TXFL bits. At reset, TRDY is set to 1.	0 = The transmitter does not require data 1 = The transmitter requires data (interrupt posted)
RTSD Bit 12	**RTS Delta**—Indicates whether the $\overline{\text{RTS}}$ pin changed state. It (RTSD) generates a maskable interrupt. When in STOP mode, only RTS assertion sets RTSD and wakes the ARM9 core. The current state of the $\overline{\text{UARTx\_RTS}}$ pin is available on the RTSS bit. Clear RTSD by writing 1 to it. Writing 0 to RTSD has no effect. At reset, RTSD is set to 0.	0 = $\overline{\text{UARTx\_RTS}}$ pin did not change state since last cleared 1 = $\overline{\text{UARTx\_RTS}}$ pin changed state (write 1 to clear)

Table 31-16. UART1 through UART4 Status Register 1 Description (continued)

Name	Description	Settings
ESCF Bit 11	**Escape Sequence Interrupt Flag**—Indicates if an escape sequence was detected. ESCF is asserted when the ESCEN bit is set and an escape sequence is detected in the RxFIFO. Clear ESCF by writing 1 to it. Writing 0 to ESCF has no effect.	0 = No escape sequence detected 1 = Escape sequence detected (write 1 to clear)
FRAMERR Bit 10	**Frame Error Interrupt Flag**—Indicates that a frame error is detected. The interrupt generated by this. Clear FRAMERR by writing 1 to it. Writing 0 to FRAMERR has no effect.	0 = No frame error detected 1 = Frame error detected
RRDY Bit 9	**Receiver Ready Interrupt / DMA Flag**—Indicates that the RxFIFO data level is above the threshold set by the RXTL bits. (See the RXFL bits description for setting the interrupt threshold.) When asserted, RRDY generates a maskable interrupt or DMA request. In conjunction with the CHARRDY bit in the URXDn_1 or URXDn_2 register, the software can continue to read the RxFIFO in an interrupt service routine until the RxFIFO is empty. RRDY is automatically cleared when data level in the RxFIFO goes below the set threshold level. At reset, RRDY is set to 0.	0 = No character ready 1 = Character(s) ready (interrupt posted)
AGTIM Bit 8	**Ageing Timer Interrupt Flag**—Indicates that data in the RxFIFO has been idle for a time of 8 character lengths (where a character length consists of 7 or 8 bits, depending on the setting of the WS bit in UCR2, with the bit time corresponding to the baud rate setting) and FIFO data level is less than RxFIFO threshold level (RxTL in the UFCR). Clear by writing a 1 to it.	0 = AGTIM is not active 1 = AGTIM is active
Reserved Bits 7	Reserved—This bit is reserved and should read 0.	
RXDS Bit 6	**Receiver IDLE Interrupt Flag**—Indicates that the receiver state machine is in an IDLE state, the next state is IDLE, and the receive pin is high. RXDS is automatically cleared when a character is received. RXDS is reset to high.	0 = Receive in progress 1 = Receiver is IDLE
AIRINT Bit 5	**Asynchronous IR WAKE Interrupt Flag**—Indicates that the IR WAKE pulse was detected (on the RXD pin). Clear AIRINT by writing 1 to it. Writing 0 to AIRINT has no effect.	0 = No pulse was detected on the RXD pin 1 = A pulse was detected on the RXD pin
AWAKE Bit 4	**Asynchronous WAKE Interrupt Flag**—Indicates that a falling edge was detected on the RXD pin. Clear AWAKE by writing 1 to it. Writing 0 to AWAKE has no effect.	0 = No falling edge was detected on the RXD pin 1 = A falling edge was detected on the RXD pin
Reserved Bits 3–0	Reserved—These bits are reserved and should read 0.	

31.6.9 UART Status Register 2

The UART Status Register 2s are registers that represent reports of errors and status flags.

		Addr
USR2_1	UART1 Status Register 2	0x1000_A098
USR2_2 USR2_3	UART2 Status Register 2	0x1000_B098
USR2_4	UART3 Status Register 2	0x1000_C098
	UART4 Status Register 2	0x1000_D098

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADET	TXFE	DTRF	IDLE	ACST	RIDELT	RIIN	IRINT	WAKE	DCDDELT	DCDIN	RTSF	TXDC	BRCD	ORE	RDR
TYPE	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	rw	rw	r
RESET	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	0x4008															

Table 31-17. UART1 through UART4 Status Register 2 Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
ADET Bit 15	**Automatic Baud Rate Detect Complete**—Indicates that an “A” or “a” was received and that the receiver detected and verified the incoming baud rate. Clear ADET by writing 1 to it. Writing 0 to ADET has no effect.	0 = ASCII “A” or “a” was not received 1 = ASCII “A” or “a” was received (write 1 to clear)
TXFE Bit 14	**Transmit Buffer FIFO Empty**—Indicates that the transmit buffer (TxFIFO) is empty. TXFE is cleared automatically when data is written to the TxFIFO. Even though TXFE is high, the transmission might still be in progress.	0 = The transmit buffer (TxFIFO) is not empty 1 = The transmit buffer (TxFIFO) is empty
DTRF Bit 13	**DTR edge triggered interrupt Flag**—This bit is asserted, when the programmed edge is detected on the $\overline{\text{DTR}}$ pin.	0 = Programmed edge not detected on $\overline{\text{DTR}}$/$\overline{\text{DSR}}$ 1 = Programmed edge detected on $\overline{\text{DTR}}$/$\overline{\text{DSR}}$ (write a 1 to clear)
IDLE Bit 12	**Idle Condition**—Indicates that an idle condition has existed for more than a programmed amount frame (see Section 31.4.4.1). The interrupt generated by this IDLE bit is cleared by writing 1 to IDLE. Writing 0 to IDLE has no effect.	0 = No idle condition detected 1 = Idle condition detected (write 1 to clear)

Table 31-17. UART1 through UART4 Status Register 2 Description (continued)

Name	Description	Settings
ACST Bit 11	**Autobaud Counter Stopped**—In autobaud detection (ADBR=1), indicates the counter which determines the baudrate was running and is now stopped. This means either START bit is finished (if ADNIMP=1), or Bit0 is finished (if ADNIMP=0). See 36.5.10.2 for more details. An interrupt can be flagged if ACIEN=1.	0 = Measurement of bit length not finished (in autobaud) 1 = Measurement of bit length finished (in autobaud). (write a 1 to clear)
RIDELT Bit 10	**Ring Indicator Delta**—This bit is used in DTE mode to indicate that the Ring Indicator input has changed state.	0 = Ring Indicator input has not changed state 1 = Ring Indicator input has changed state (write a 1 to clear)
RIIN Bit 9	**Ring Indicator Input**—This bit is used in DTE mode reflect the status if the Ring Indicator input.	0 = Ring Detected 1 = No Ring Detected
IRINT Bit 8	**Serial Infrared Interrupt Flag**—When an edge is detected on the RX pin during SIR Mode, this flag will be asserted. This flag can cause an interrupt which can be masked using the control bit ENIRI: UCR4 [8].	0 = no edge detected 1 = valid edge detected (write a 1 to clear)
WAKE Bit 7	**Wake**—Indicates the start bit is detected. WAKE can generate an interrupt that can be masked using the WKEN bit. Clear WAKE by writing 1 to it. Writing 0 to WAKE has no effect.	0 = start bit not detected 1 = start bit detected (write 1 to clear)
DCDDELT Bit 6	**Data Carrier Detect Delta**—This bit is used in DTE mode to indicate that the Data Carrier Detect input has changed state.	0 = Data Carrier Detect input has not changed state 1 = Data Carrier Detect input has changed state (write a 1 to clear)
DCDIN Bit 5	**Data Carrier Detect Input**—This bit is used in DTE mode reflect the status if the Data Carrier Detect input.	0 = Carrier signal Detected 1 = No Carrier signal Detected
RTSF Bit 4	**RTS Edge Triggered Interrupt Flag**—Indicates if a programmed edge is detected on the $\overline{\text{RTS}}$ pin. The RTEC bits select the edge that generates an interrupt (see Table 31-3). RTSF can generate an interrupt that can be masked using the RTSEN bit. Clear RTSF by writing 1 to it. Writing 0 to RTSF has no effect.	0 = Programmed edge not detected on RTS 1 = Programmed edge detected on RTS (write 1 to clear)
TXDC Bit 3	**Transmitter Complete**—Indicates that the transmit buffer (TxFIFO) and Shift Register is empty; therefore the transmission is complete. TXDC is cleared automatically when data is written to the TxFIFO.	0 = Transmit is incomplete 1 = Transmit is complete
BRCD Bit 2	**BREAK Condition Detected**—Indicates that a BREAK condition was detected by the receiver. Clear BRCD by writing 1 to it. Writing 0 to BRCD has no effect.	0 = No BREAK condition was detected 1 = A BREAK condition was detected (write 1 to clear)
ORE Bit 1	**Overrun Error**—Indicates that the receive buffer (RxFIFO) was full when data was being received. Clear ORE by writing 1 to it. Writing 0 to ORE has no effect.	0 = No overrun error 1 = Overrun error (write 1 to clear)
RDR Bit 0	**Receive Data Ready**—Indicates that at least 1 character is received and written to the RxFIFO. If the URXD_x register is read and there is only 1 character in the RxFIFO, RDR is automatically cleared.	0 = No receive data ready 1 = Receive data ready

31.6.10 UART Escape Character Registers

The UART Escape Character Registers establishes the software-selected escape character.

		Addr
UESC_1	UART1 Escape Character Register	0x1000_A09C
UESC_2	UART2 Escape Character Register	0x1000_B09C
UESC_3	UART3 Escape Character Register	0x1000_C09C
UESC_4	UART4 Escape Character Register	0x1000_D09C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									ESC_CHAR							
TYPE	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1
	0x002B															

Table 31-18. UART1 through UART4 Escape Character Register Description

Name	Description
Reserved Bits 31–8	Reserved—These bits are reserved and should read 0.
ESC_CHAR Bits 7–0	**UART Escape Character**—Holds the selected escape character that all received characters are compared against to detect an escape sequence.

31.6.11 UART Escape Timer Registers

The UART Escape Timer Registers establish the software-selected maximum interval between escape characters. These registers are scalable in intervals of 2 msec to a maximum of 8.192 sec. Note that ONEMS register must also be programmed for Escape detection.

		Addr
UTIM_1	UART1 Escape Timer Register	0x1000_A0A0
UTIM_2	UART2 Escape Timer Register	0x1000_B0A0
UTIM_3	UART3 Escape Timer Register	0x1000_C0A0
UTIM_4	UART4 Escape Timer Register	0x1000_D0A0

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					TIM											
TYPE	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 31-19. UART1 through UART4 Escape Timer Register Description

Name	Description
Reserved Bits 31–12	Reserved—These bits are reserved and should read 0.
TIM Bits 11–0	**UART Escape Timer**—Holds the maximum interval allowed between escape characters.

31.6.12 UART BRM Incremental Registers

The BRM (Binary Rate Multiplier) Incremental registers hold the numerator value (minus one) of the BRM ratio. These registers can be written to at any time. Hardware updates the value in the UART BRM Modulator Registers (UBMR_x) at the appropriate time to avoid glitches on the BRM_CLK output. (sampling clock). The BRM is not updated until both the modulator (UBMR_x) and the Incremental (UBIR_x) registers are written to by software. If only one register is written to by the software, the BRM ignores this data until the other register is also written.

		Addr
UBIR_1	UART1 BRM Incremental Register	0x1000_A0A4
UBIR_2	UART2 BRM Incremental Register	0x1000_B0A4
UBIR_3	UART3 BRM Incremental Register	0x1000_C0A4
UBIR_4	UART4 BRM Incremental Register	0x1000_D0A4

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INC															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 31-20. UART1 through UART4 BRM Incremental Register Description

Name	Description
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.
INC Bits 15–0	**Incremental Numerator**—Holds the numerator value minus one of the BRM ratio (see Section 31.4.9). The UBIR register MUST be updated before the UBMR register for the baud rate to be updated correctly. If only one register is written to by software, the BRM will ignore this data until the other register is written to by software. Updating this field using byte accesses is not recommended and is undefined. UARTx_EN (PCCR0[3:0]) MUST be enabled before accessing this register.

31.6.13 UART BRM Modulator Registers

The UART BRM (Binary Rate Multiplier) Modulator Registers hold the denominator value (minus one) of the BRM ratio. These registers can be written to at any time. Hardware updates the BRM Modulator value at the appropriate time to avoid glitches on the BRM_CLK output (sampling clock).

The BRM is not updated until both the Modulator (UBMR_x) and the Incremental (UBIR_x) registers are written to by software. If only one register is written to by the software, the BRM ignores this data until the other register is also written.

		Addr
UBMR_1	UART1 BRM Modulator Register	0x1000_A0A8
UBMR_2	UART2 BRM Modulator Register	0x1000_B0A8
UBMR_3	UART3 BRM Modulator Register	0x1000_C0A8
UBMR_4	UART4 BRM Modulator Register	0x1000_D0A8

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MOD															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 31-21. UART1 through UART4 BRM Modulator Register Description

Name	Description
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.
MOD Bits 15–0	**Modulator Denominator**—Holds the value of the denominator minus one of the BRM ratio (see Section 31.4.9). The UBIR register MUST be updated before the UBMR register for the baud rate to be updated correctly. If only one register is written to by software, the BRM will ignore this data until the other register is written to by software. Updating this register using byte accesses is not recommended and undefined. UARTx_EN (PCCR0[3:0]) MUST be enabled before accessing this register.

31.6.14 UART Baud Rate Count Registers

The read-only UART Baud Rate Count Registers count the start bit of the incoming baud rate. When the start bit is detected and counted, these registers retain their value until the next automatic baud rate detection sequence is initiated. The registers are reset to 0x00000004, however they stay at 0xFFFF in the case of an overflow.

		Addr
UBRC_1	UART1 Baud Rate Count Register	0x1000_A0AC
UBRC_2	UART2 Baud Rate Count Register	0x1000_B0AC
UBRC_3	UART3 Baud Rate Count Register	0x1000_C0AC
UBRC_4	UART4 Baud Rate Count Register	0x1000_D0AC

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BCNT															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	0x0004															

Table 31-22. UART1 through UART4 Baud Rate Count Register Description

Name	Description
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.
BCNT Bits 15–0	**Baud Rate Count Register**—This read only register is used to count the start bit of the incoming baud rate (if ADNIMP=1), or start bit + bit0 (if ADNIMP=0). When the measurement is done, the Baud Rate Count Register contains the number of UART internal clock cycles (clock after divider) present in an incoming bit. BCNT retains its value until the next Automatic Baud Rate Detection sequence has been initiated. The 16-bit Baud Rate Count register is reset to 4 and stays at hex FFFF in the case of an overflow.

31.6.15 One Millisecond Registers

This 16-bit read/write register must contain the value of the UART internal frequency divided by 1000. The internal frequency is obtained after the UART internal divider. In fact this register contains the value corresponding to the number of UART internal clock cycles that are present in one millisecond. This register must be filled when Escape timer and Infrared Special Case features are used.

		Addr
ONEMS_1	UART1 One Millisecond Register	0x1000_A0B0
ONEMS_2	UART2 One Millisecond Register	0x1000_B0B0
ONEMS_3	UART3 One Millisecond Register	0x1000_C0B0
ONEMS_4	UART4 One Millisecond Register	0x1000_D0B0

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONEMS															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 31-23. UART1 through UART4 BRM Modulator Register Description

Name	Description
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.
ONEMS Bits 15–0	**One Millisecond Register**—This 16-bit register must contain the value of the UART internal frequency divided by 1000. The internal frequency is obtained after the UART internal divider. In fact this register contains the value corresponding to the number of UART internal clock cycles are present in one millisecond.

31.6.16 UART Test Register

The UART Test Register controls the various test features of the UART module.

		Addr
UTS_1	UART1 Test Register	0x1000_A0B4
UTS_2	UART2 Test Register	0x1000_B0B4
UTS_3	UART3 Test Register	0x1000_C0B4
UTS_4	UART4 Test Register	0x1000_D0B4

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			FRCP ERR	LOOP	DBGEN	LOOPIR	RXDB G			TX EMPTY	RX EMPTY	TX FULL	RX FULL			SOFT RST
TYPE	r	r	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	0x0060															

Table 31-24. UART1 through UART4 Test Register Description

Name	Description	Settings
Reserved Bits 31–14	Reserved—These bits are reserved and should read 0.	
FRCPERR Bit 13	**Force Parity Error**—Forces the transmitter to generate a parity error if parity is enabled. FRCPERR is provided for system debugging.	0 = Generate normal parity 1 = Generate inverted parity (error)
LOOP Bit 12	**Loop TX and RX for Test**—Controls loopback for test purposes. When LOOP is high, the receiver input is internally connected to the transmitter and ignores the RXD pin. The transmitter is unaffected by LOOP.	0 = Normal receiver operation 1 = Internally connect the transmitter output to the receiver input
DBGEN Bit 11	**$\overline{\text{Debug Enable}}$**—This bit controls whether to respond to the debug_b input signal.	0 = UART will go into debug mode when debug_b is LOW 1 = UART will not go into debug mode even if debug_b is LOW
LOOPIR Bit 10	**Loop TX and RX for IR Test (LOOPIR)**—This bit controls loopback from transmitter to receiver in the InfraRed interface.	0 = No IR loop 1 = Connect IR transmitter to IR receiver
RXDBG Bit 9	**RXFIFO Debug Mode**—This bit controls the operation of the RX fifo read counter when in debug mode.	0 = Rx FIFO read pointer does not increment 1 = Rx FIFO read pointer increments as normal

Table 31-24. UART1 through UART4 Test Register Description (continued)

Name	Description	Settings
Reserved Bits 8–7	Reserved—These bits are reserved and should read 0.	
TXEMPTY Bit 6	**Tx FIFO Empty**—Indicates that the TxFIFO is empty.	0 = The Tx FIFO is not empty 1 = The Tx FIFO is empty
RXEMPTY Bit 5	**Rx FIFO Empty**—Indicates the RxFIFO is empty.	0 = The Rx FIFO is not empty 1 = The Rx FIFO is empty
TXFULL Bit 4	**Tx FIFO Full**—Indicates the TxFIFO is full.	0 = The Tx FIFO is not full 1 = The Tx FIFO is full
RXFULL Bit 3	**Rx FIFO Full**—Indicates the RxFIFO is full.	0 = The Rx FIFO is not full 1 = The Rx FIFO is full
Reserved Bits 2–1	Reserved—These bits are reserved and should read 0.	
SOFTRST Bit 0	**Software Reset**—Indicates the status of the software reset ($\overline{\text{SRST}}$).	0 = No software reset 1 = Generate software reset

31.7 UART Operation in Low-Power System States

The UART's serial interface will operate as long as the 16x bit clock generator is provided with the PERCLK1. The RXEN, TXEN and UARTEN bits are set by the user and provide software control of low-power modes.

The following UART interrupts wake the ARM926EJ-S processor from STOP mode:

- RTS
- IrDA Asynchronous WAKE (AIRINT)
- Asynchronous WAKE (AWAKE)

The UART_EN bit (PCCR0[3:0]) enables the clock inputs to the UART module. If the control bit is set to 0, it abruptly disables the clock inputs to the UART module. To get maximum power conservation when the module is shut off, make sure that the above clock control bit is set to 0. Setting the UARTEN (UCR1_x) bit to 0 only shuts off the receiver and transmitter logic and the associated clocks.

If the UART is used only in transmit mode, UARTEN and TXEN must be set to 1. If the UART is used only in receive mode, UARTEN and RXEN must be set to 1. Setting TXEN or RXEN to 0 allows to save a lot of power.

When an asynchronous WAKE interrupt exits the ARM926EJ-S processor from STOP mode, make sure that a dummy character is sent first because the first character may not be received correctly. After the settling time of the USB_PLL, actual characters can be sent to the UART. Even though the dummy character is written to RxFIFO, just ignore it.

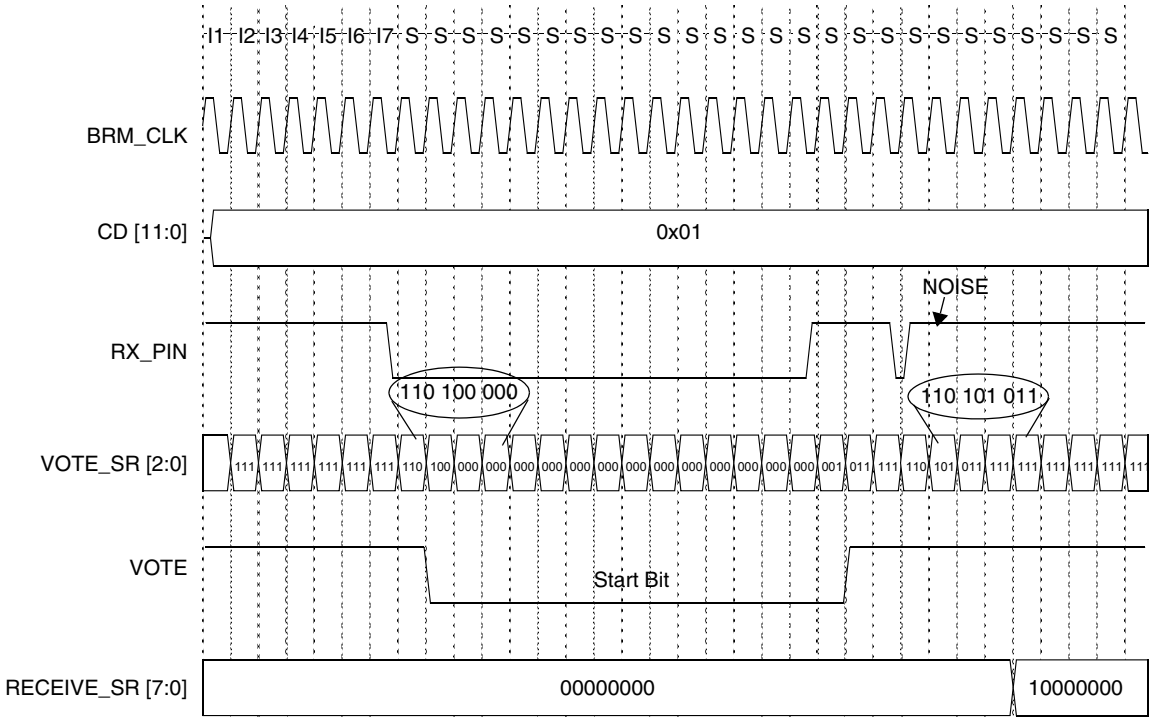


Figure 31-5. Majority Vote Results

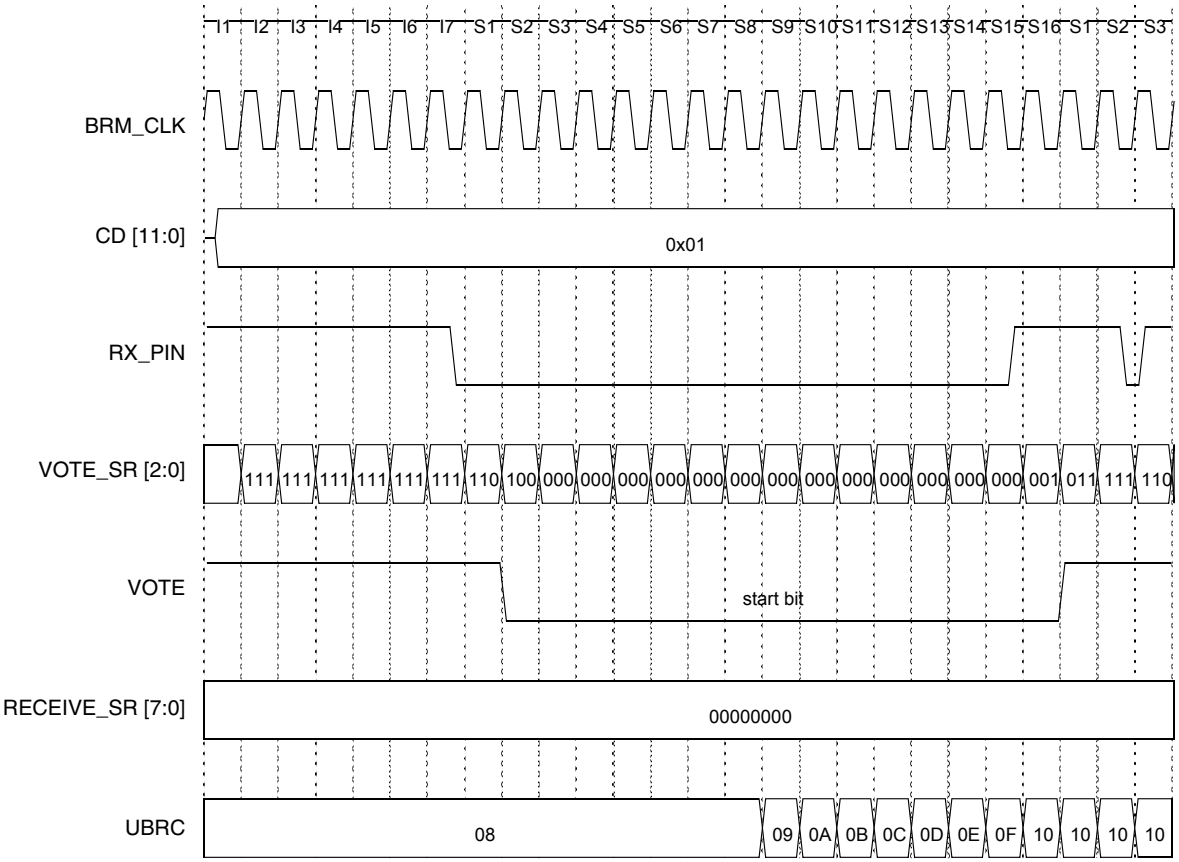


Figure 31-6. Baud Rate Detection of Divisor = 1

Chapter 32
Universal Serial Bus On-The-Go (USB OTG)

The USB OTG module is designed to provide high performance full speed USB On-The-Go (OTG) compliant functionality to the i.MX21. USB OTG, is an extension of the USB 2.0 specification for connecting peripheral devices to each other. USB OTG products can communicate with each other without the need to be connected to a host.

The i.MX21 USB OTG module is designed to minimize power consumption, host processor overhead and system resource utilization while providing maximum bulk data throughput and reliable isochronous transfers for superior and reliable OTG performance.

The USB OTG module features:

- USB Host and OTG/Function controllers, which can operate concurrently.
- Minimum number of interrupts to the host processor as a result of bundled and delayed interrupt control, USB transfer level hardware.
- Bus mastering design to off-load host processing.
- Hardware assisted protocol negotiation for minimal software intervention resulting in increased power savings.
- Transceiver interface compliant to OTG Transceiver Interface Specification with seamless interface to industry standard OTG transceivers.
- Special bypass modes to allow USB hosts to access USB devices downstream of the i.MX21.

The USB OTG module is specifically designed and optimized for low power, mobile and embedded applications.

32.1 USB OTG Architecture

The USB OTG Host/Function Controller module is composed of the following major blocks:

- **USB Host Controller logic—**Together with the host Serial Interface Engine (SIE) and root hub, manages transfer and low level protocols, connectivity and power management.
- **USB Function Controller logic—**Together with the function SIE manages all the low-level protocol and the USB function protocol.
- **Memory-mapped application interface.**
- **Memory block—**The Host Controller logic, the Function Controller logic, and the application interface all have access to the memory block for reading and writing data and control information. A RAM arbiter handles RAM access-related issues.
- **Host Negotiation Protocol (HNP)—**To be implemented in software. The application can disable this negotiation process and force the OTG part of the USB OTG controller to operate as a host or function.

- **Register Sets—**The USB OTG registers provide the control and coordination of the host, function, and USB-OTG controllers as well as the top-level features such as clock control.

  The USB OTG registers can also configure the entire core for the desired mode of operation. There are some instances when only USB host or device operation is desired. In this case, the configuration can be set in the Hardware Mode Register. The default mode of operation is with software HNP enabled which allows host, device, and OTG functionality. There are three sets of registers in the USB OTG module:

  — USB OTG Module-level registers: Contain all registers for USB OTG module-level configuration, control, and status.

  — Host Control registers: These are registers specific to the Host Controller functions.

  — Function Controller registers: These registers are specific to the Function Controller settings, control, and status.
- DMA controller engine for direct memory transfer to/from the system memory.
- I^2C module for external OTG-transceiver interface.

32.1.1 Host Controller Features

- Fully compliant with the USB 2.0 Specification and the USB On-The-Go 1.0 Specification provided as a supplement to the USB 2.0 Specification. All USB ports are compliant with these specifications for full speed (12Mbs) and low speed (1.5Mbs) operation
- Hardware based Endpoint Transfer Descriptors (ETD) data structure management and scheduling to support simultaneous operation of up to 32 active USB pipes.
- True transfer level operation with transaction scheduling and handling (data sequence toggle, error re-try, etc.) carried out in hardware.
- Reliable isochronous transfer with loose timing requirements on the microprocessor.
- Up to 97% USB bandwidth utilization.

32.1.2 Function Controller Features

- USB Specification Revision 2.0 compliant full speed function controller.
- Software configurable as a standard bus-powered, self-powered, or USB OTG function.
- 32 physical endpoints (pipes), 16 logical user configurable endpoints, each of which can be programmed in terms of endpoint type, maximum packet size, data buffer association.

32.1.3 USB On-The-Go Features

- Compliant with USB On-The-Go Supplement Specification Revision 1.0.
- HNP/SRP is implemented in software.

32.2 USB OTG Support

The USB OTG standard (*USB On-The-Go Supplement to the USB 2.0 Specification*) is proposed in response to market demands on connectivity among desktop and portable devices. It eliminates the PC as

the “middle-man” supporting peer-to-peer data exchange between devices under the USB protocol. Devices that will support the USB OTG standard include cell phones, digital static cameras, personal digital assistants (PDAs), set top boxes (STBs), and MP3 players.

According to the USB OTG specification, any device supporting the protocol must be able to work as a *USB Function*. That is, it must be able to be connected directly to the PC, operating as an ordinary USB device. An OTG device with USB host functionality is called a Dual Role Device (DRD), which must support the following:

- Ability to determine whether the DRD is assuming the role of an A-Device, or that of a B-Device.
- As the A-Device, it must provide VBUS power of 5V with a current loading capacity of at least 8 mA.
- It must support the *Host Negotiation Protocol* (HNP), which allows two connected DRDs to exchange their roles in the USB host-function relationship in the middle of an OTG session.
- It must support the *Session Request Protocol* (SRP), which allows one of the two connected devices to initiate a new OTG session after the VBUS has been turned off (as a power saving measure).

32.3 OTG-IP CORE Operational Configurations

The USB OTG module is a high-performance compound OTG and host controller that has three active ports; one OTG port and two host ports that can run simultaneously. The two host ports are provided so that these ports can be used for either internal or external connections. For example, a web tablet with a touch-pad may want to have an external OTG port, but wish to use one of the two host ports as an internal interface to a touch pad. The OTG can be programmed using the CRECFG field in the Hardware Mode Register to operate in any one of four modes by programming the component to operate as an OTG, host or function device. These modes are briefly described in the following sections. A device with OTG capability to switch roles between Host and Function modes is referred to as a Dual Role Device (DRD).

32.3.1 Host Only Mode

All the ports will be USB Hosts. The HNP and Function Controller are disabled.

32.3.2 Function Host Mode

The shared USB port operates as a USB Function. All other ports will operate as USB Hosts. The HNP is disabled in this mode. This mode allows simultaneous host and device operation.

32.3.3 Software HNP

The USB-OTG module allows software to control the operation of the shared USB port. Software is responsible for the timing, tracking the HNP states, and performing the HNP protocol. The shared OTG port can work either as a USB Host or USB Function depending on the outcome of the OTG handshaking.

32.4 HNP and SRP

The software HNP/SRP will make use of the control and status registers provided in the OTG hardware.

When at any time the ID pin changes, an ID Change Interrupt is asserted. Following the interrupt, software must read the ISADEV and ISBDEV bits of the HNP control status register. In software HNP, software must write to the Master or Slave bits respectively, to enable the appropriate device and state machine. Software must always set both Band Gap Enable and Comparator Enable bits before using the HNP logic

32.5 USB Host Architectural Overview

The USB OTG modules allows a large number of host endpoints since it is typical for a host controller to connect to more than one device. The number of endpoints on the device side is not as critical a device controller can only connect to one host.

All transfers are described using transfer descriptors. These descriptors are based on the OHCI recommended data structures and completely describe the transfer. In order to configure a transfer, software only needs to write the correct descriptor information. The host scheduler and controller will handle all the processing from there. The endpoint transfer descriptor plays a central role in how the host processing works and so the following section will examine the ETD in more detail.

32.5.1 Endpoint Transfer Descriptor

The endpoint transfer descriptor is used to describe a transfer to the Host controller. From the data in the descriptor, the host can extract the data location in memory, direction of transfer, type of transfer, etc. The following is a list of some of the fields described by the ETD:

- Device address
- Device speed
- Endpoint number
- Data transfer type/direction (IN, OUT, SETUP)
- Endpoint type (control, bulk, interrupt or isochronous)
- Maximum packet size

The basic communication structure between the Software Host Control Driver and the Host Controller is the Endpoint Transfer Descriptors. An ETD is composed of two parts: the Endpoint Descriptor (ED) and the Transfer Descriptor (TD).

The Host Controller Driver has full read/write access to all portions of the structures. The fields in the structures that are modified by the Host Controller hardware are noted in the field descriptions. An ETD can only be executed only when it is enabled. Software is discouraged from modifying an ETD without disabling it first. The enabling and disabling of ETDs is done through writes to the ETD Enables Register.

32.5.2 Data Buffer Specification

The hardware uses a double buffering mechanism to keep the flow of data smooth. This means that two buffers are used simultaneously. In the case of an OUT transfer (away from the Host), the one buffer contains data that is being transferred to the serial bus, and the other buffer is being loaded by either the MCU or the DMA master. The buffer addresses for each endpoint are described in the ETD.

32.6 Software Considerations

32.6.1 Creating a Transfer

At the time a transfer is created, software must:

- Create an ETD data structure in the ETD memory of the hardware.
- Initialize transfer related properties in the ETD.
- Allocate data buffer(s) in the Data Memory for the pipe. Set data buffer related information (number of buffers, their addresses and sizes, etc.) into the data buffer specification part of the ETD.
- Initialize the Toggle value to the desired value in the ETD (usually 0).
- Set the ETD Enable bit to 1 to start the transfer or place it in the scheduling queue.

32.6.2 Post Transfer Processing

At the time when a transfer is completed (whether successful or not), user software shall:

- Process the transfer based on the final status reported by the hardware. These are referred to as Completion Codes.
- If the transfer was not successful, determine whether the transfer is to be kept open. If the transfer is not to be retried, software should discard the data and the descriptor contents. Otherwise, software must reinitialize the descriptor (most fields should contain the correct data already) and restart it.

32.6.3 Endpoint Descriptor Description

The USB OTG module has three separate memory spaces: ETD memory, EP memory and Data memory. Only the Host Controller uses the ETD memory and only the Function Controller uses the EP memory. Both Host Controller and Function Controller share the Data memory.

The ETD memory can contain up to 32 ETDs, each of them comprised by 4 DWORDs. Therefore an ETD memory with 32 entries is 512 bytes in size. Software has full Read and Write access to the ETD memory. All the unused fields in the ETD structure should be initialized to 0.

As the Host Controller scans the enabled ETDs for transmission during a frame, the Interrupt ETDs are processed first, the Isochronous ETDs second, and then the Control and Bulk ETDs are processed last. Control endpoints are given equal or more access to the bus in comparison with Bulk Endpoints as described in the OHCI specification. If the ratio needs to be changed, software has control over this via the ControlBulkServiceRatio field of the Host Control Register. The range of possible Control Bulk Service Ratios is from 1:1 to 4:1.

For low speed transactions, regardless of the data size, the Host Controller compares the current number of USB bit cycles remaining in the frame with the value of the LSThreshold field of the LowSpeedThreshold Register. If the frame remaining time is less than LSThreshold, the low speed transaction is not started on the bus.

A non-ISO transfer is completed when the Host Controller successfully transfers, to or from an endpoint TotalTransferCount number of bytes. An Isochronous TD is completed when all data packets have been transferred. On successful completion, the Host Controller sets CompletionCode to NoError, and retires the TD to the ETDDoneStatus Register. The complete list of error codes is found in the Table 32-2.

Table 32-1. Transaction Types

Transaction Type	Description
Isochronous	• Specifically associated with a defined frame number • Has only one packet per frame • Each packet can have different lengths ranging from 0 to 1023 Bytes
Interrupt	• Used to take into account the fact that interrupt endpoints need to be served at a pre-defined time interval, at a given relative polling position, at that interval • If “NAKed” the transfer will be retried at the next interval
Bulk/Control	• Takes advantage of DMA like mechanism • Retried on errors or “NAKed” • Quality guaranteed • Timing is not guaranteed • Each stage of a Control transfer is similar to a Bulk transfer.

32.6.4 NAK Handshake

For non-ISO transfers, in normal working mode, the StopOnNak bit is set to 0. When an endpoint returns a NAK handshake, all TD fields remain the same after the transaction, as they were when the transaction began. The Host Controller will retry the packet until a non-NAK response is received. For Interrupt type packets, the transaction will be retried after PollingInterval number of frames. For Control/Bulk packets, the RetryDelay field is used to determine how long, if at all, the Host Controller should wait before retrying. When the StopOnNak bit is SET and an endpoint returns a NAK handshake, the ETD is retired without retry.

If an Isochronous endpoint returns NAK during the data phase of an IN, the Host Controller writes the CompletionCode of the frame’s PacketStatusWord. The Isochronous TD is not retired early and the endpoint is not halted.

32.6.5 STALL Handshake

If a non-Isochronous endpoint returns a STALL PID, the Host Controller retires the TD with the CompletionCode set to STALL and halts the endpoint. The ErrorCount, and DataToggle fields retain the values that they had at the start of the transaction. In this case, the ETD is retired without retry.

If an isochronous endpoint returns STALL during the data phase of an IN, the Host Controller writes the CompletionCode to STALL and sets the data size to 0. The Isochronous TD is not retired early and the endpoint is not halted.

Table 32-2. ETD Completion Code Description

Code	Meaning	Description
0000b	NoError	TD data packet processing completed with no detected errors
0001b	CRC	Last data packet from endpoint contained a CRC error.
0010b	BitStuffing	Last data packet from endpoint contained a bit stuffing violation
0011b	DataToggleMismatch	Last packet from endpoint had data toggle PID that did not match the expected value.
0100b	Stall	TD was moved to the Done Queue because the endpoint returned a STALL PID
0101b	DeviceNotResponding	Device did not respond to token (IN) or did not provide a handshake (OUT)
0110b	PIDFailure	Check bits on PID from endpoint failed on data PID (IN) or handshake (OUT); Receive PID was not valid when encountered or PID value is not defined
0111b	Reserved	–
1000b	DataOverrun	The amount of data returned by the endpoint exceeded either the size of the maximum data packet allowed from the endpoint (found in MaximumPacketSize field of ED) or the remaining buffer size.
1001b	DataUnderrun	The endpoint returned less than MaximumPacketSize and that amount was not sufficient to fill the specified buffer
1010b	ACK	An ACK handshake is received. For debug purpose.
1011b	NAK	A NAK handshake is received. Used in coordination with StopOnNak bit.
1100b	BufferOverrun	During an IN, Host Controller received data from endpoint faster than its buffer is freed by the system
1101b	BufferUnderrun	During an OUT, the system could not transfer data fast enough from the system memory to local memory to keep application with USB data rate. Used for isochronous ETDs.
1110b	SCHEDULEOVERRUN	Host Controller has detected a schedule-overrun condition.
1111b	Not Accessed	This code is set by software before the TD or the corresponding data packet is downloaded into the Host Controller's local memory.

32.6.6 USB Mux Overview

The USB Mux allows connection of an external USB Host to a USB Function device attached to the i.MX21 in what is called the Bypass mode. The i.MX21 can directly interface, without transceivers, to an on-board USB Device through one of the USB Host (for example, USB Host1) ports while simultaneously connected to an external transceiver based connection via the USB OTG port.

The USB Mux provides a means to allow an externally attached USB Host device (connected via the OTG pin interface) to access the locally attached USB Function device. When the input pin, $\overline{\text{USB\_Bypass}}$ is pulled low, the signals from the external transceiver are routed through the USB Mux to the external USB Function (which are connected to the USB Host1 pins). When $\overline{\text{USB\_Bypass}}$ is high, the bypass mode is off and the locally attached USB Function is directly connected to the USB Host1 and the OTG port is connected to the external OTG or USB Host. Figure 32-1 shows the Bypass mode operation scenarios.

The USB_Bypass input is an active low signal. USB_Overcurrent is an input signal. These signals can only be used for USB function, not for GPIO function.

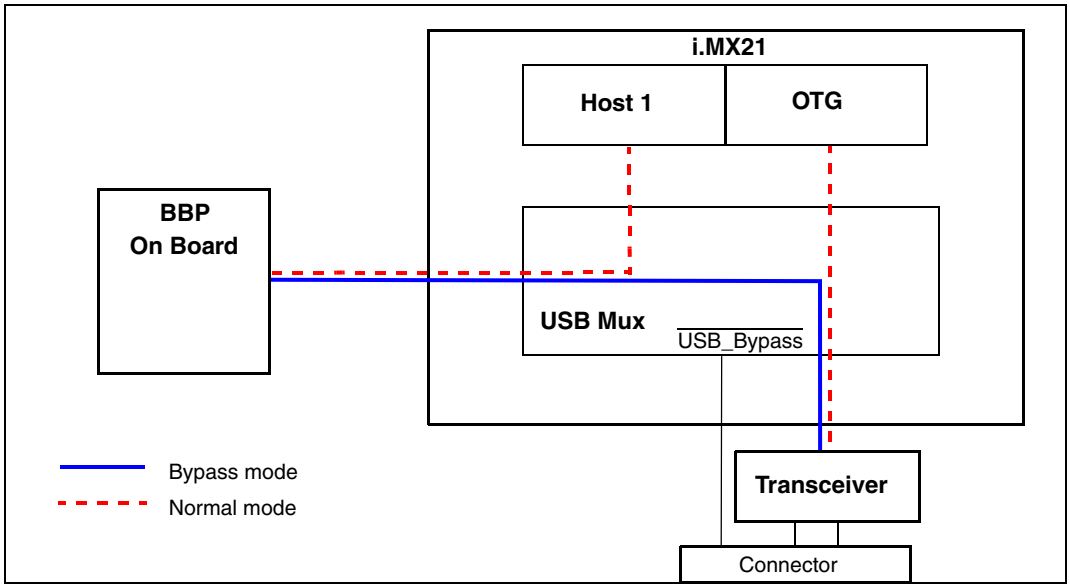


Figure 32-1. USB On-Board Diagram

32.7 Programming Model

The USB OTG module level registers are described and listed in order of their address in Table 32-3.

The host registers are described in Table 32-27.

The function register summery is located in Table 32-55.

The DMA register summary is located in Table 32-72.

Table 32-3. USB OTG Module Level Register Summary

Description	Name	Address
System register	Hardware Mode Register	0x1002 4000
USB OTG module interrupt source	USBOTG Module Interrupt Status Register	0x1002 4004
USB OTG module interrupt mask	USBOTG Module Interrupt Status Enable Register	0x1002 4008
Allows for disabling and enabling the clocks of the Host and Function	Clock Control Register	0x1002 400C
Allows for independent resets of each of the blocks in the Host and Function	Reset Control Register	0x1002 4010
Sets the width of a USB Frame in full-speed bit times	Frame Interval Register	0x1002 4014
Contains the number of full-speed bit times remaining in the current frame and sets the relationship between *mClk* width and full speed bit time	Frame Remaining Register	0x1002 4018
Gives the current state of the HNP	Hnp Control Status Register	0x1002 401C
HNP interrupt sources	Hnp Interrupt Status Register	0x1002 402C

Table 32-3. USB OTG Module Level Register Summary (continued)

Description	Name	Address
HNP interrupt mask	Hnp Interrupt Enable Status Register	0x1002 4030
Controls the wake-up interrupts and MUX logic	USB Control Register	0x1002 4600

32.7.1 Hardware Mode Register

In certain applications the user can force the USB OTG module to work directly as a host or as a function. This register is reset by a system reset or a write to the Hardware Mode Register with the ResetControlLogic bit set.

HWMODE — USB OTG Hardware Mode Register — Addr 0x10024000

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FUNCREV								HSTREV							
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
	0x2020															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		ANASDBEN							OTGXCVR		HOSTXCVR				CRECFG	
TYPE	r	rw	r	r	r	r	r	r	rw	rw	rw	rw	r	r	rw	rw
RESET	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1
	0x00A3															

Table 32-4. Hardware Mode Register Description

Name	Description	Settings
FUNCREV Bits 31–24	**Function Revision**—Indicates the revision of the Function Controller code and features.	–
HSTREV Bits 23–16	**Host Revision**—Indicates the revision of the Host Controller code and features.	–
Reserved Bit 15	Reserved—This bit is reserved and should read 0.	
ANASDBEN Bit 14	**OTG Analog Signal Short Debounce Enable**—	0 = Enable the OTG VBUS and ID signal long debounce logic. 1 = Disable the OTG VBUS and ID signal long debounce logic.
Reserved Bits 13–8	Reserved—These bits are reserved and should read 0.	

Table 32-4. Hardware Mode Register Description (continued)

Name	Description	Settings
OTGXCVR Bits 7–6	**OTG Transceiver Properties**—This field defines various transceiver configurations.	Transmit/Receive 00 = Differential (TxDp, TxDm)/Differential (RxDp, RxDm, RxD) 10 = Single-Ended (Dat, SE0/Differential (RxDp, RxDm, RxD) 01 = Differential (TxDp, TxDm)/Single-Ended (Dat, SE0) 11 = Single-Ended (Dat, SE0)/Single-Ended (Dat, SE0)
HOSTXCVR Bits 5–4	**Host Transceiver Properties**—This field defines different transceiver configurations. **Note:** All Host transceivers must share the same properties.	**Transmit/Receive** 00 = Differential (TxDp, TxDm)/Differential (RxDp, RxDm, RxD) 10 = Single-Ended (Dat, SE0)/Differential (RxDp, RxDm, RxD) 01 = Differential (TxDp, TxDm)/Single-Ended (Dat, SE0) 11 = Single-Ended (Dat, SE0)/Single-Ended (Dat, SE0)
Reserved Bits 3–2	Reserved—These bits are reserved and should read 0.	
CRECFG Bits 1–0	**USB OTG Module Configuration**— This field is used to configure the USB OTG operation between Host and Function operation.	00 = Reserved. 01 = Host Only Operation. Disable the HNP procedure. Force the OTG to work as a Host. 10 = Function Host Operation. Disable the HNP procedure. Force the OTG to work as a Function, however other ports are in Host mode. 11 = Software HNP. The software is responsible for doing HNP.

32.7.2 USB OTG Module Interrupt Status Register

This register provides status on various USB OTG module level events that cause hardware interrupts. When an USB OTG module level event occurs, hardware sets the corresponding bit in this register. When a bit becomes set, a hardware interrupt is generated if the interrupt is enabled in the USB OTG Module Interrupt Status Register. These interrupts will not be de-asserted until the source has been cleared or disabled.

CINT_STAT — USB OTG Interrupt Status Register — Addr 0x10024004

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											ASHNPINT	ASFCINT	ASHCINT	HNPINT	FCINT	HCINT
TYPE	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-5. USB OTG Interrupt Status Register Description

Name	Description	Setting
Reserved Bits 31–6	Reserved—These bits are reserved and should read 0.	
ASHNPINT Bit 5	**Asynchronous HNP Interrupt**—When asserted indicates that the asynchronous HNP event has triggered an interrupt. To clear this bit, the MainClockEnable must be set.	0 = Async HNP event has not occurred 1 = Async HNP event has occured
ASFCINT Bit 4	**Asynchronous Function Interrupt**—When asserted indicates that the asynchronous Function Controller event has triggered an interrupt. To clear this bit, the FunctionClockEnable must be set.	0 = Async function controller event has not occured 1 = Async function controller event has occured
ASHCINT Bit 3	**Asynchronous Host Interrupt**—When asserted indicates that the asynchronous Host Controller event has triggered an interrupt. To clear this bit, the HostClockEnable must be set.	0 = Async host controller event has not occured 1 = Async host controller event has occured
HNPINT Bit 2	**HNP Interrupt**—When asserted indicates that the HNP has triggered an interrupt. To clear this bit all source flags must be cleared or disabled.	0 = HNP event has not occured 1 = HNP event has occured
FCINT Bit 1	**Function Interrupt**—When asserted indicates that the Function Controller has triggered an interrupt. To clear this bit all source flags must be cleared or disabled.	0 = Function controller event has not occured 1 = Function controller event has occured
HCINT Bit 0	**Host Interrupt**—When asserted indicates that the Host Controller has triggered an interrupt. To clear this bit all source flags must be cleared or disabled.	0 = Host controller event has not occured 1 = Host controller event has occured

32.7.3 USB OTG Module Interrupt Enable Register

This register contains the corresponding enable bits of the USB OTG Module Interrupt Status register. An interrupt is only asserted if its enable bit is active. This register is reset by a system reset or a write to the Reset Control Register Reset Control Logic bit.

CINT_STEN — USB OTG Interrupt Status Enable Register — Addr 0x10024008

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											ASHNP INTEN	ASFC INTEN	ASHC INTEN	HNP INTEN	FC INTEN	HC INTEN
TYPE	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-6. OTG Interrupt Enable Register Description

Name	Description	Settings
Reserved Bits 31–6	Reserved—These bits are reserved and should read 0.	
ASHNPINTEN Bit 5	**Asynchronous HNP Interrupt Enable**—When set it allows asynchronous HNP event to trigger an interrupt.	0 = Disable Async HNP event interrupt 1 = Enable Async HNP event interrupt
ASFCINTEN Bit 4	**Asynchronous Function Interrupt Enable**—When set it allows asynchronous Function Controller event to trigger an interrupt.	0 = Disable Async Function Controller interrupt 1 = Enable Async Function Controller interrupt
ASHCINTEN Bit 3	**Asynchronous Host Interrupt Enable**—When set it allows asynchronous Host Controller event to trigger an interrupt.	0 = Disable Async Host interrupt 1 = Enable Async Host interrupt
HNPINTEN Bit 2	**HNP Interrupt Enable**—When set it allows HNP interrupts to be asserted.	0 = Disable HNP event interrupt 1 = Enable HNP event interrupt
FCINTEN Bit 1	**Function Interrupt Enable**—When set it allows Function interrupts to be asserted.	0 = Disable Function Controller interrupt 1 = Enable Function Controller interrupt
HCINTEN Bit 0	**Host Interrupt Enable**—When set it allows Host interrupts to be asserted.	0 = Disable Host interrupt 1 = Async Host interrupt

32.7.4 USB OTG Module Clock Control Register

This register controls the clocks of the USB OTG module to reduce power and turn off sections of logic when they are not needed. For example, when the USB OTG module is working in Host Only Operation, the application can decide to switch off the clock input to the Function Controller USB OTG module logic. This register is reset by a system reset or a write to the Reset Control Register Reset Control Logic bit. Upon reset these clocks are disabled and must be enabled before use.

CLK_CTRL — USB OTG Clock Control Register — Addr 0x1002400C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														FUNCCLK	HSTCLK	MAINCLK
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-7. Clock Control Register Description

Name	Description	Settings
Reserved Bits 31–3	Reserved—These bits are reserved and should read 0.	
FUNCCLK Bit 2	**Function Clock Enable**—When set the clocks to the Function controller are enabled. Setting this bit clears the Asynchronous Function Interrupt bit in the USB OTG Interrupt Status Register.	0 = Disable clocks to Function controller 1 = Enable clocks to Function controller
HSTCLK Bit 1	**Host Clock Enable**—When set the clocks to the Host controller are enabled. Setting this bit clears the Asynchronous Host Interrupt bit in the USB OTG Interrupt Status Register.	0 = Disable clocks to Host controller 1 = Enable clocks to Host controller
MAINCLK Bit 0	**Main Clock Enable**—When set the main clock to the USB OTG module is enabled. Setting this bit clears the Asynchronous HNP Interrupt bit in the USB OTG Interrupt Status Register. **Note:** This bit cannot be cleared if either the Function Clock Enable or Host Clock Enable is set.	0 = Disable main clock 1 = Enable main clock

32.7.5 USB OTG Module Reset Control Register

This register allows for independent resets of the major blocks of the USB OTG module. Each bit is self-clearing after the reset is complete.

RST_CTRL — USB OTG Reset Control Register — Addr 0x10024010

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSTI2C										RSTCTRL	RSTFC	RSTFSKE	RSTRH	RSTHSIE	RSTHC
TYPE	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-8. USB OTG Reset Control Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
RSTI2C Bit 15	**Reset I^2C Controller**—When set, all of the registers in the I^2C interface are reset to their default condition.	0 = Clears RSTI2C bit 1 = Resets I^2C controller
Reserved Bits 14–6	Reserved—These bits are reserved and should read 0.	
RSTCTRL Bit 5	**Reset Control Logic**—When set this resets all of the registers in the CTRL interface to their default condition. This bit is self-clearing.	0 = Clears RSTCTL bit 1 = Resets all CTRL interface registers
RSTFC Bit 4	**Reset Function Controller**—When set this resets all of the registers in the Function Controller interface to their default condition. This bit is self-clearing.	0 = Clears RSTFC bit 1 = Resets all CTRL interface registers
RSTFSIE Bit 3	**Reset Function SIE**—When set this resets all of the registers in the Function SIE interface to their default condition. This bit is self-clearing.	0 = Clears the RSTFSIE bit 1 = Resets all Function SIE I/F registers
RSTRH Bit 2	**Reset Root Hub**—When set this resets all of the registers in the Root Hub interface to their default condition. This bit is self-clearing.	0 = Clears RSTRH bit 1 = Resets all Root Hub interface registers.
RSTHSIE Bit 1	**Reset Host SIE**—When set this resets all of the registers in the Host SIE interface to their default condition. This bit is self-clearing.	0 = Clears RSTHSIE bit 1 = Resets all Host SIE interface registers.
RSTHC Bit 0	**Reset Host Controller**—When set this resets all of the registers in the Host Controller interface to their default condition. This bit is self-clearing.	0 = Clears RSTHC bit 1 = Resets all Host controller interface registers.

32.7.6 Frame Interval Register

The Frame Interval is the number of bit times between two consecutive Start of Frame (SOF) packets. Software can be used to perform minor adjustments to the Frame Interval by writing a new value over the current value at each SOF. If the RESETFRAME bit is not used the new frame time is updated at the next SOF. This allows the Host Controller to be synchronized with an external clocking resource and adjusted for unknown local clock offsets. The FRAMEINTERVALPERIODIC field contains the number of bit times in a frame when periodic packets like Isochronous and Interrupt can be sent. Depending on the Software application being used this value can be adjusted. This register is reset by a system reset or a write to the RESETCONTROLLOGIC bit in the USB OTG Reset Control Register.

NOTE

This register is an advanced register and changing the values without understanding the implications may cause the USB OTG module to become out of compliance with the USB specification.

FRM_INVTL — Frame Interval Register — Addr 0x10024014

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			FRMINTPER													
TYPE	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	1	0	1	0	1	0	0	0	1	0	1	1	1	1
	0x2A2F															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSTFRM		FRMINT													
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1
	0x2EDF															

Table 32-9. Frame Interval Register Description

Name	Description
Reserved Bits 31–30	Reserved—These bits are reserved and should read 0.
FRMINPER Bits 29–16	**Frame Interval Periodic**—The value written here sets the amount of time in a frame that Periodic packets can be sent. This value is in the number of full-speed bit times.
RSTFRM Bit 15	**Reset Frame**—When set the Frame Remaining Register is cleared and set to the value in the FRMINT field of this register.
Reserved Bit 14	Reserved—These bits are reserved and should read 0.
FRMINT Bits 13–0	**Frame Interval**—This field contains the width of a frame. This value is in the number of full-speed bit times. This specifies the interval between two consecutive SOFs in bit times. The nominal value is set to be 11,999.

32.7.7 Frame Remaining Register

The Frame Remaining Register contains a 14-bit down counter showing the bit time remaining in the current Frame.

FRM_REMAIN — Frame Remaining Register — Addr 0x10024018

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			FRMREMN													
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	1	0	1	0	1	0	0	0	1	0	1	1	1	1
	0x2A2F															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1
	0x2EDF															

Table 32-10. Frame Remaining Register Description

Name	Description
Reserved Bits 31–30	Reserved—These bits are reserved and should read 0.
FRMREMN Bits 29–16	**Frame Remaining**—This number represents the number of full-speed bit times still remaining in the current frame. This counter is decremented at each bit time. When it reaches zero, loading the Frame Interval value specified in Frame Remaining at the next bit time boundary resets it. When entering the USB Operational state, the Host Controller re-loads the content with the Frame Interval and uses the updated value from the next SOF.
Reserved Bits 15–0	Reserved—These bits are reserved and should read 0.

32.7.8 USB OTG HNP Control Status Register

This register contains the status and control bits used to operate and configure the HNP controller. This register is reset by a system reset or a write to the Reset Control Register Reset Control Logic bit.

HNP_CTRL — HNP Control Register — Addr 0x1002401C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		HNPDAT	VBUSGTBSE	VBUSABSV	VBUSGTAVV					SLAVE	MASTER	BGEN	CMPEN	ISBDEV	ISADEV	
TYPE	r	rw	rw	rw	rw	r	rw	rw	r	rw	rw	rw	rw	rw	rw	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	0x0004															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SWVBUSPUL			SWAUTORST	SWPUDP		SWPDDM						CLRERROR	ADROPBUS	ABBUSREQ	
TYPE	rw	r	r	rw	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	r
RESET	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	0x0200															

Table 32-11. HNP Control Register Description

Name	Description
Reserved Bit 31	Reserved—These bits are reserved and should read 0.
HNPDAT Bit 30	**HNP Data Toggle**—When asserted this bit indicates that either the DP or the DM line is asserted. SOFTWARE can use this line to detect Data line pulsing SRP.
VBUSBSE Bit 29	**V Bus Greater Than B Session End**—When asserted, this bit indicates that the bus has been sampled greater than or equal to b_session_end (0.8V). When not asserted a B-DEVICE can consider the session ended.
VBUSABSV Bit 28	**V Bus A B Session Valid**—When asserted this indicates that the bus has been sampled greater than or equal to 0.8V, however, between 0.8V and 2.0V. This is in the a_session_valid and the b_session_valid range.
VBUSGTAVV Bit 27	**V Bus Greater Than A V Bus Valid**—When asserted this indicates that the bus has been sampled greater than or equal to 4.4V. This indicates a_session_valid.
Reserved Bit 26–23	Reserved—These bits are reserved and should read 0.
SLAVE Bit 22	**HNP Slave State**—If Software writes when this bit is SET, it will cause the USB OTG module to work as a Function.

Table 32-11. HNP Control Register Description (continued)

Name	Description
MASTER Bit 21	**HNP Master State**—If Software writes when this bit is SET, it will cause the USB OTG module to work as a Host.
BGEN Bit 20	**Band Gap Enable**—When SET enables the charge pump band gap.
CMPEN Bit 19	**Comparator Enable**—When SET enables the charge pump comparator.
ISBDEV Bit 18	**Is B Device**—The hardware determines if this device is a B-DEVICE by checking the state of the ID pin. The default state of any OTG device is a B-DEVICE.
ISADEV Bit 17	**Is A Device**—The hardware determines if this device is an A-DEVICE by checking the state of the ID pin. A device will be considered an A-DEVICE if the ID pin is low.
Reserved Bit 16	Reserved—These bits are reserved and should read 0.
SWVBUSPUL Bit 15	**Software V Bus Pulse**—If Software writes with this bit SET it will cause the VBUS to be pulsed.
Reserved Bits 14–13	Reserved—These bits are reserved and should read 0.
SWAUTORST Bit 12	**Software Automatic Reset**—This bit is only valid while in Software HNP. Software can write to this bit so that the hardware will automatically generate a downstream reset upon a connection to the port. This bit is only valid while the Hnp Master State bit is set. When the B-DEVICE receives a SET_FEATURE HNP_ENABLE, and the B-DEVICE removes its Pull-up resistor, software should set this bit to automatically drive reset upon a detection of connection. This bit is self-clearing upon the generation of the USB reset.
SWPUDP Bit 11	**Software Pull-Up DP**—Software can write to this bit to turn on or off the pull-up resistor. When SET the pull-up is enabled. When CLEAR the pull-down is enabled. This register is only used when in both **Function Host Mode** and **Software HNP**.
Reserved Bit 10	Reserved—These bits are reserved and should read 0.
SWPDDM Bit 9	**Software Pull-Down DM**—When SET the pull-down on the DM line is enabled. This register is writable only in Software HNP.
Reserved Bits 8–4	Reserved—These bits are reserved and should read 0.
CLRERROR Bit 3	**HNP Clear Error State**—A-DEVICE: When SET this bit when it has recovered from an over-current condition. Software needs to issue Hnp Clear Error State to clear the error when the A-DEVICE HNP is in A_VBUS_ERROR state and does not want to release the bus. As a B-DEVICE this bit is used to force the transition from B_SLAVE to B_IDLE.
ADROPBUS Bit 2	**A Drop V Bus**—When SET indicates that the Software of the A-DEVICE wants to power down the VBUS.
ABBUSREQ Bit 1	**A B Bus Request**—When SET indicates that the Software is requesting to be A_MASTER or B_MASTER. This bit remains true when this device wants to use the bus, and is false when it no longer wants to use the bus.
Reserved Bit 0	Reserved—These bits are reserved and should read 0.

32.7.9 HNP Interrupt Status Register

This register provides status on various HNP level events that cause hardware interrupts. When an HNP event occurs, hardware sets the corresponding bit in this register. When a bit becomes set, an interrupt is generated if the interrupt is enabled in the HnpInterruptEnableStatus Register. This register is reset by a system reset or a write to the Reset Control Register Reset Control Logic bit.

HNP_INT_STAT — HNP Interrupt Status Register — Addr 0x1002402C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	I2COTGINT										SRPINT		ABSESVAILD	AVUBSVAILD		IDCHANGE
TYPE	rw	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-12. HNP Interrupt Status Register Description

Name	Description
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.
I^2COTGINT Bit 15	**I^2C OTG Transceiver Controller Interrupt**—This bit is used to signal an interrupt from the OTG Transceiver. The interrupt source on the transceiver will have to be gathered through the I^2C OTG Transceiver Controller and the I^2C OTG Transceiver. This interrupt will not be deasserted until the source has been cleared or disabled.
Reserved Bits 14–6	Reserved—These bits are reserved and should read 0.
SRPINT Bit 5	**Session Request Detect Interrupt**—After receiving this interrupt software must decide if it wants to respond to the Session Request. This interrupt is asserted when either a VBUS pulse is detected and/or a Data pulse is detected.
Reserved Bit 4	Reserved—This bit is reserved and should read 0.
ABSESVAILD Bit 3	**A B Session Valid Change Interrupt**—This interrupt is only valid in Software HNP and Function Host Mode. When asserted this means that the VBUS has crossed the b_session_vld threshold. Software should read the HnpControlStatus Register to determine the current status of the VBUS.
AVBUSVAILD Bit 2	**AV Bus Valid Change Interrupt**—This interrupt is only valid in Software HNP and Function Host Mode. When asserted this means that the VBUS has crossed the a_vbus_vld level. Software should read the HnpControlStatus Register to determine the current status of the VBUS.

Table 32-12. HNP Interrupt Status Register Description (continued)

Name	Description
Reserved Bit 1	Reserved—This bit is reserved and should read 0.
IDCHANGE Bit 0	**ID Change Interrupt**—Indicates the state of the ID pin has changed. Software should read the HnpControlStatus Register to determine the current status of the ID pin.

32.7.10 HNP Interrupt Enable Register

This register is reset by a system reset or a write to the Reset Control Register Reset Control Logic bit.

HNP_INT_EN — HNP Interrupt Enable Status Register — Addr 0x10024030

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	I2COTGINTEN										SRPINTEN		ABSESVAILDEN	AVUBSVAILDEN		IDCHANGEEN
TYPE	rw	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-13. HNP Interrupt Enable Status Register Description

Name	Description
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.
I^2COTGINTEN Bit 15	**I^2C OTG Transceiver Controller Interrupt Enable**—Enable bit of I^2C OTG Transceiver Controller Interrupt Enable.
Reserved Bits 14–6	Reserved—These bits are reserved and should read 0.
SRPINTEN Bit 5	**Session Request Detect Interrupt Enable**—Enable bit of Session Request Detect Interrupt.
Reserved Bit 4	Reserved—This bit is reserved and should read 0.

Table 32-13. HNP Interrupt Enable Status Register Description (continued)

Name	Description
ABSESVAILDEN Bit 3	**AB Session Valid Interrupt Enable**—Enable bit of AB Session Valid Interrupt.
AVBUSVAILDEN Bit 2	**AV Bus Valid Interrupt Enable**—Enable bit of AV Bus Valid Interrupt.
Reserved Bit 1	Reserved—This bit is reserved and should read 0.
IDCHANGEEN Bit 0	**ID Change Interrupt Enable**—Enable bit of ID Change Interrupt.

32.7.11 USB Control Register

This register controls the wake-up interrupts and MUX logic.

USBCTRL — USB Control Register — Addr 0x10024600

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					I2C WU INT STAT	OTG WU INT STAT	HOST WU INT STAT	FNT WU INT STAT					I2C WU INT EN	OTG WU INT EN	HOST WU INT EN	FNT WU INT EN
TYPE					r	r	r	r					rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			$\overline{OTG}$ $\overline{RCV}$ $\overline{RXDP}$	HOST 1 BYP TLL	OTG BYP VAL		HOST1 BYP VAL			OTG PWR MASK	HOST1 PWR MASK	HOST2 PWR MASK		USB BYP	HOST1 TXEN OE	
TYPE					rw	rw	rw	rw		rw	rw	rw		r	rw	
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 32-14. USB Control Register Description

Name	Description	Settings
Reserved Bits 31–28	Reserved—These bits are reserved and should read 0.	
12C WU INT STAT Bit 27	**I^2C Wake-up Interrupt Status**—Indicates that I^2C wake-up interrupt is asserted. Status depends on the external hardware condition.	0 = Not asserted 1 = Asserted

Table 32-14. USB Control Register Description (continued)

Name	Description	Settings
OTG WU INT STAT Bits 26	**OTG Wake-Up Interrupt Status**—Indicates that the OTG Wake-up interrupt is asserted. Status dependant on external hardware condition.	0 = Not asserted 1 = Asserted
HOST WU INT STAT Bit 25	**HOST Wake-Up Interrupt Status—**Indicate that HOST Wake-up Interrupt is asserted. Status dependant on external hardware condition.	0 = Not asserted 1 = Asserted
FNT WU INT STAT Bit 24	**FUNCTION Wake-Up Interrupt Status—**Indicate that FUNCTION Wake-up Interrupt is asserted. Status dependant on external hardware condition.	0 = Not asserted 1 = Asserted
Reserved Bits 23–20	Reserved—These bits are reserved and should read 0.	
I2C WU INT EN Bit 19	**I^2C Wake-Up Interrupt Enable**—Enable I^2C Wake-up Interrupt.	0 = Disable 1 = Enable
OTG WU INT EN Bit 18	**OTG Wake-Up Interrupt Enable**—Enable OTG Wake-up Interrupt.	0 = Disable 1 = Enable
HOST WU INT EN Bit 17	**HOST Wake-Up Interrupt Enable—**Enable HOST Wake-up Interrupt.	0 = Disable 1 = Enable
FNT WU INT EN Bit 16	**FUNCTION Wake-Up Interrupt Enable—**Enable FUNCTION Wake-up Interrupt.	0 = Disable 1 = Enable
Reserved Bits 15–14	Reserved—These bits are reserved and should read 0.	
$\overline{\text{OTG RCV RXDP}}$ Bit 13	**OTG Differential Receiver**—When enabled, the design connects the OTG port's RxDP to the OTG's receiver pin internally to the core. When disabled, the receiver pin is left to the module port connection, for an external transceiver connection.	0 = Enable 1 = Disable Default value = 0
HOST1 BYP TLL Bit 12	**Host 1 Bypass Transceiver-Less Logic**—When active, the transceiver emulation logic in the transceiver-less link logic is bypassed. The interface signal propagates as driven. When inactive, the transceiver emulation logic is employed in the transceiver-less link logic block, to support transceiver-less interface.	0 = Disable 1 = Enable (active) Default value = 1
OTG BYP VAL Bits 11–10	**OTG Bypass Value**—Mux Input Value for DP and DM. Default value should be used.	
HOST1 BYP VAL Bits 9–8	**Host 1 Bypass Value**—Mux Input Value for DP and DM. Default value should be used.	
Reserved Bit 7	Reserved—These bits are reserved and should read 0.	
OTG PWR MASK Bit 6	**OTG Power Output Pin Mask—**Controls whether the USB_PWR pin for USBOTG is Masked.	0 = Unmasked 1 = Masked
HOST1 PWR MASK Bit 5	**Host 1 Power Output Pin Mask—**Controls whether the USB_PWR pin for USBH1 is masked.	0 = Unmasked 1 = Masked
HOST2 PWR MASK Bit 4	**Host 2 Power Output Pin Mask—**Controls whether the USB_PWR pin for USBH2 is masked.	0 = Unmasked 1 = Masked
Reserved Bit 3	Reserved—These bits are reserved and should read 0.	

Table 32-14. USB Control Register Description (continued)

Name	Description	Settings
USB BYP Bit 2	**USB Bypass Enable**—Status of USB Mux Bypass pin.	0 = USB Mux bypass disable 1 = USB Mux bypass enable
HOST1 TXEN OE Bit 1	**Host 1 Transmit_Enable Output Enable—**Controls whether the pin for Host 1 Transmit Enable is output enabled.	0 = Output Disable 1 = Output Enable
Reserved Bit 0	Reserved—These bits are reserved and should read 0.	

Table 32-15 describes the ETD memory map.

Table 32-15. ETD Memory Map

ETD Address	Description
0x10024200	ETD 0 DWORD0
0x10024204	ETD 0 DWORD1
0x10024208	ETD 0 DWORD2
0x1002420C	ETD 0 DWORD3
0x10024210	ETD 1 DWORD0
…	…

32.7.12 Host Endpoint Transfer Descriptor WORD0 Format

Host Endpoint Descriptor Word0 Format

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		SNDNAK		TOGCRY	HALTED		MAXPKTSIZ									
TYPE	r	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
	0x															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FORMAT		SPEED	DIRECT		ENDPNT				ADDRESS						
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
	0x															

Table 32-16. Host Endpoint Transfer Descriptor DWORD0 Description

Name	Description	Setting
Reserved Bit 31	This bit is reserved and should read zero.	
SNDNAK Bit 30	**Stop On NAK**—When CLEARED, NAK is a good handshake and no special action is taken (normal USB working mode); When SET, upon a NAK handshake, the ETD is retired and put onEndpointDoneStatus. An interrupt will be generated immediately no matter what the software set in the DelayInterrupt bit and the ImmediateInterrupt bit.	0 = NAK (normal) 1 = NAK interrupt generated
Reserved Bit 29	This bit is reserved and should read zero.	
TOGCRY Bit 28	**Toggle Carry**—This bit is the data toggle carry bit. Whenever a TD is retired, this bit is written back by Host Controller to contain the last data toggle value (LSB of Data Toggle field from the retired TD). This field is not used for Isochronous Endpoints.	See description
HALTED Bit 27	Endpoint Halted—This bit is set by the Host Controller to indicate that processing of the TD on the endpoint is halted, usually due to an error in processing a TD.	0 = TD processing OK 1 = TD Processing halted
Reserved Bit 26	This bit is reserved and should read zero.	
MAXPKTSIZ Bits 25–16	**Maximum Packet Size**—This field indicates the maximum number of bytes that can be sent to or received from the endpoint in a single data packet. Table 32-17 shows max size allowed by USB 2.0 spec for low-speed and full-speed devices.	
FORMAT Bits 15–14	**Format**—This bit indicates the format of the TD linked to this ED.	00 = Control 01 = ISO 10 = Bulk 11 = Interrupt
SPEED Bit 13	**Speed**—Indicates the speed of the endpoint.	0 = Full speed 1 = Low speed
DIRECT Bits 12–11	**Direction**—This field indicates the direction of data flow (IN or OUT.) If neither IN nor OUT is specified, then the direction is determined from the Direction PID field of the TD.	00 = Get direction from TD 01 = OUT to endpoint 10 = IN from endpoint 11 = Get direction from TD
ENDPNT Bits 10–7	**Endpoint Number**—This is the USB address of the endpoint within the function that is either the source or destination of this transfer.	See description
ADDRESS Bits 6–0	**Address**—This is the USB address of the function containing the endpoint that this ED controls.	See description

Table 32-17. ETD Word 0 Maximum Packet Size Limits

Packet Type	Interrupt	Control	Bulk	Isochronous
Maximum number of bytes (full speed)	64	64	64	1023
Maximum number of bytes (low speed)	8	8	Not allowed	Not allowed

32.7.12.1 Control/Bulk Transfer Descriptor Format

Control/Bulk Transfer Descriptor Format WORD1

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	YBUFSRTAD															
TYPE	r	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	XBUFSRTAD															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Table 32-18. Control/Bulk Transfer Descriptor DWORD1 Description

Name	Description
YBUFSRTAD Bits 31–16	**Y Buffer Start Address**—Contains the physical start address of the mega-buffer Y allocated for this TD that will be accessed for transfer to/from the endpoint. This address is the byte address location in the memory. Since the IP USB OTG module is a 32 bits, the last two bits of the address will be dropped.
XBUFSRTAD Bits 15–0	**X Buffer Start Address**—Contains the physical start address of the mega-buffer X allocated for this TD that will be accessed for transfer to/from the endpoint. This address is the byte address location in the memory. Since the IP USB OTG module is a 32 bits, the last two bits of the address will be dropped.

Control/Bulk Transfer Descriptor Format WORD2

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	COMPCODE				ERRORCNT				DATATOG		DELAYINT			BUFROUND	DIRPID	
TYPE	r	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									RTRYDELAY							
TYPE	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw

Table 32-19. Control/Bulk Transfer Descriptor DWORD2 Description

Name	Description	Setting
COMPCODE Bits 31–28	**Completion Code**—This field contains the completion status of the last attempted transaction. A complete list of completion codes is found in Table 32-2.	See description
ERRORCNT Bits 27–24	**Error Count**—For each transmission error, this value is incremented. The MSB of the Error Count is used to define two levels of error tolerance. MSB=0: standard OHCI tolerance: If **ErrorCount** is 3 and another error occurs (4 consecutive errors), the error type is recorded in the **CompletionCode** field and the ETD is placed on the ETDDoneStatus register. MSB=1: enhanced tolerance: If **ErrorCount** is 7 and another error occurs (8 consecutive errors), the error type is recorded in the **CompletionCode** field and the ETD is placed on the ETDDoneStatus register. **ErrorCount** is reset to 0 after each error-free transaction. When a transaction completes without error, **ErrorCount** is reset to 0.	See description
DATATOGL Bits 23–22	**Data Toggle**—This field is used to generate/compare the data PID value (DATA0 or DATA1). The Host Controller updates this DataToggle field after each successful transmission/reception of a data packet. The MSB of this field is 0 when the data toggle value is acquired from the Toggle Carry field in the ED and 1 when the data toggle value is taken from the LSB of this field.	See description
DELAYINT Bits 21–19	**Delay Interrupt**—This field contains the interrupt delay count for this TD. When a TD is completed, the Host Controller may wait for Delay Interrupt frames before generating an interrupt.	See description
BUFROUND Bit 18	**Buffer Rounding**—If this bit is 0, then the data packet to a TD from an endpoint must exactly equal to the software defined data packet size. If the bit is 1, then the data packet may be smaller than the software defined data packet without causing an error condition on the completion of this transfer.	0 = Data packet size equals defined size 1 = Data packet size does not equal defined size
DIRPID Bits 17–16	**Direction PID**—This field indicates the direction of data flow and the PID to be used for the token. This field is only relevant to the Host Controller if the Direction field in the ED was set to 00b or 11b indicating that the PID determination is deferred to the TD. The encoding of the bits within the byte for this field is:	PID Type/Data Direction 00 = SETUP/OUT to endpoint 01 = OUT/OUT to endpoint 10 = IN/IN from endpoint 11 = Reserved
Reserved Bits 15–8	These bits are reserved and should read zero	
RTRYDELAY Bits 7–0	**Retry Delay**—Used when a non-periodic endpoint is to be executed consecutively, without any other endpoints in-between. The unit is ms (frame). When 0, immediate retry can be done. When not 0, retry will be done only after Retry Delay frames.	see description

Control/Bulk Transfer Descriptor Format WORD3

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BUFSIZE											TOTBYECNT				
TYPE	r	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TOTBYECNT															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Table 32-20. Control/Bulk Transfer Descriptor DWORD3 Description

Name	Description
BUFSIZE Bits 31–21	**Buffer Size**—Size of X/Y Buffer. This value is given in bytes and is a value between 0–2047, corresponding to buffer sizes between 1 and 2 Kbytes. For example: If the buffer size will be 0x40 (64 bytes), then software should write 0x3F in this field.
TOTBYECNT Bits 20–0	**Total Byte Count**—This is the total number of bytes to be transferred by the ETD. The ETD can be set to transfer 0 - (2M-1) bytes.

32.8 Interrupt Transfer Descriptor

Interrupt Transfer Descriptor DWORD1

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	YBUFSRTAD															
TYPE	r	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	XBUFSRTAD															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Table 32-21. Interrupt Transfer Descriptor DWORD1 Description

Name	Description
YBUFSRTAD Bits 31–16	**Y Buffer Start Address**—Contains the physical start address of the mega-buffer Y allocated for this TD that will be accessed for transfer to/from the endpoint. This address is the byte address location in the memory. Since the IP USB OTG module is 32 bits, the last two bits of the address are dropped.
XBUFSRTAD Bits 15–0	**X Buffer Start Address**—Contains the physical start address of the mega-buffer X allocated for this TD that will be accessed for transfer to/from the endpoint. This address is the byte address location in the memory. Since the IP USB OTG module is a 32 bits, the last two bits of the address will be dropped.

32.8.1 Interrupt Transfer Descriptor DWORD2

Interrupt Transfer Descriptor DWORD2

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	COMPCODE				ERRORCNT				DATATOG		DELAYINT			BUFROUND	DIRPID	
TYPE	r	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RELPOLPOS								POLINTERV							
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Table 32-22. Interrupt Transfer Descriptor DWORD2 Description

Name	Description	Setting
COMPCODE Bits 31–28	**Completion Code**—This field contains the completion status of the last attempted transaction. A complete list of completion codes is found in Table 32-2.	See description
ERRORCNT Bits 27–24	**Error Count**—For each transmission error, this value is incremented. The MSB of the Error Count is used to define two levels of error tolerance. MSB=0: standard OHCI tolerance: If Error Count is 3 and another error occurs (4 consecutive errors), the error type is recorded in the Completion Code field and the ETD is placed on the ETDDoneStatus register. MSB=1: enhanced tolerance: If Error Count is 7 and another error occurs (8 consecutive errors), the error type is recorded in the Completion Code field and the ETD is placed on the ETDDoneStatus register. Error Count is reset to 0 after each error-free transaction. When a transaction completes without error, Error Count is reset to 0.	See description
DATATOGL Bits 23–22	**Data Toggle**—This field is used to generate/compare the data PID value (DATA0 or DATA1). The Host Controller updates this DataToggle field after each successful transmission/reception of a data packet. The MSB of this field is 0 when the data toggle value is acquired from the ToggleCarry field in the ED and 1 when the data toggle value is taken from the LSB of this field.	See description
DELAYINT Bits 21–19	**Delay Interrupt**—This field contains the interrupt delay count for this TD. When a TD is completed, the Host Controller may wait for DelayInterrupt frames before generating an interrupt.	See description
BUFROUND Bit 18	**Buffer Rounding**—If this bit is 0, then the data packet to a TD from an endpoint must exactly equal to the software defined data packet size. If the bit is 1, then the data packet may be smaller than the software defined data packet without causing an error condition on the completion of this transfer.	0 = Data packet size equals defined size 1 = Data packet size does not equal defined size
DIRPID Bits 17–16	**Direction PID**—This field indicates the direction of data flow and the PID to be used for the token. This field is only relevant to the Host Controller if the *Direction* field in the ED was set to 00b or 11b indicating that the PID determination is deferred to the TD.	PID Type/Data Direction 00 = SETUP/OUT to endpoint 01 = OUT/OUT to endpoint 10 = IN/IN from endpoint 11 = Reserved

Table 32-22. Interrupt Transfer Descriptor DWORD2 Description (continued)

Name	Description	Setting
RELPOLPOS Bits 15–8	**Relative Polling Position**—This field indicates in which relative frame within the polling interval the transaction needs to be attempted for the interrupt endpoint. The value is between 0 and 255. Once the ETD is written into the ETD table and is enabled, and the 8 LSB of the Frame Number matches this number, first interrupt transaction is attempted, then every Polling Interval afterward.	See description
POLINTERV Bits 7–0	**Polling Interval**—A value of 0x00 means in each frame, Host Controller would process one Interrupt transaction for this TD. Acceptable values are between 0 and 255ms.	See description

32.8.2 Interrupt Transfer Descriptor DWORD3

Interrupt Transfer Descriptor DWORD3

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BUFSIZE											TOTBYECNT				
TYPE	r	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TOTBYECNT															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Table 32-23. Interrupt Transfer Descriptor DWORD3 Description

Name	Description
BUFSIZE Bits 31–21	**Buffer Size**—Size of X/Y Buffer. This value is given in bytes and is a value between 0–2047, corresponding to buffer sizes between 1 and 2 Kbytes. For example: If the buffer size will be 0x40 (64 bytes), then software should write 0x3F in this field.
TOTBYECNT Bits 20–0	**Total Byte Count**—This is the total number of bytes to be transferred by the ETD. The ETD can be set to transfer 0 - (2M-1) bytes.

32.9 Isochronous Transfer Descriptor

Isochronous Transfer Descriptor DWORD1

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	YBUFSRTAD															
TYPE	r	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	XBUFSRTAD															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Table 32-24. Isochronous Transfer Descriptor DWORD1 Description

Name	Description
YBUFSRTAD Bits 31–16	**Y Buffer Start Address**—Contains the physical start address of the mega-buffer Y allocated for this TD that will be accessed for transfer to/from the endpoint. This address is the byte address location in the memory. Since the IP USB OTG module is a 32 bits, the last two bits of the address will be dropped.
XBUFSRTAD Bits 15–0	**X Buffer Start Address**—Contains the physical start address of the mega-buffer X allocated for this TD that will be accessed for transfer to/from the endpoint. This address is the byte address location in the memory. Since the IP USB OTG module is a 32 bits, the last two bits of the address will be dropped.

Isochronous Transfer Descriptor DWORD2

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	COMPCODE							FRAMECNT			DELAYIN					
TYPE	r	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STARTFRM															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Table 32-25. Isochronous Transfer Descriptor DWORD2 Description

Name	SW	HW	Description
COMPCODE Bits 31–28	R	W	**Completion Code**—This field contains the completion code when the Isochronous TD is moved to the ETD Done Status register. A complete list of completion codes is found in Table 32-2.
Reserved Bits 27–25			
FRAMECNT Bit 24	RW	R	**Frame Count**—Number of data packets described by this Isochronous TD. 0: There is 1 data packet 1: There are 2 data packets
Reserved Bits 23–22			
DELAYIN Bits 21–19	RW	R	**Delay Interrupt**—This field contains the interrupt delay count for this TD. When a TD is completed, the Host Controller may wait for Delay Interrupt frames before generating an interrupt.
Reserved Bits 18–16			
STARTFRM Bits 15–0	RW	R	**Starting Frame**—This field contains the bottom 16 bits of the Frame Number in which the first data packet of the Isochronous TD is to be sent.

Isochronous Transfer Descriptor DWORD3

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	COMPCODE1						PKTLEN1									
TYPE	r	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COMPCODE0						PKTLEN0									
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Table 32-26. Isochronous Transfer Descriptor DWORD3 Description

Name	Description
COMPCODE1 Bits 31–28	**Completion Code 1**—For packet 1. This CompetionCode1 is set to Not Accessed by software before the TD or the corresponding data packet is downloaded into the Host Controller's local memory. A complete list of completion codes is found in Table 32-2.
Reserved Bits 27–26	Reserved
PKTLEN1 Bits 25–16	**Packet Length 1**—This field defines the length of the buffer. It defines the size of the second isochronous data packet. The Packet Length must be less than or equal to the Max Packet Size.
COMPCODE0 Bits 15–12	**Completion Code 0**—Used for packet 0. This CompetionCode0 is set to Not Accessed by software before the TD or the corresponding data packet is downloaded into the Host Controller's local memory. A complete list of completion codes is found in Table 32-2.
Reserved Bits 11–10	Reserved
PKTLEN0 Bits 9–0	**Packet Length 0** – This field defines the length of buffer which is the size of the 1st isochronous data packet. The Packet Length must be less than or equal to the Max Packet Size.

32.10 Host Registers

This section describes the registers used to control the Host Controller of the USB OTG module. A summary of these registers is shown in Table 32-27.

Table 32-27. Host Register Summary

Address	Register Bit Name	Description
0x1002 4080	HostControl Register	Host controller configuration.
0x1002 4084	RESERVED	—
0x1002 4088	System Interrupt Status Register	Interrupt status.
0x1002 408C	System Interrupt Enables Register	Interrupt mask.
0x1002 4090–4094	RESERVED	—
0x1002 4098	XBufferInterruptStatus Register	X buffer Interrupt status.
0x1002 409C	YBufferInterruptStatus Register	Y buffer Interrupt status.

Table 32-27. Host Register Summary (continued)

Address	Register Bit Name	Description
0x1002 40A0	XYInterruptEnables Register	X/Y Interrupt enables.
0x1002 40A4	RESERVED	—
0x1002 40A8	XFilledStatus Register	X buffer filled status.
0x1002 40AC	YFilledStatus Register	Y buffer filled status.
0x1002 40B0–40BC	RESERVED	—
0x1002 40C0	ETDEnables Register	Informs the host controller which ETDs are enabled.
0x1002 40C4–40C8	RESERVED	—
0x1002 40CC	ImmediateInterrupt Register	Informs the host controller that all ETDs that are set here should generate an interrupt immediately upon completion, instead of waiting for an SOF.
0x1002 40D0	EndpointDoneStatus Register	Asserts which ETDs are done processing.
0x1002 40D4	ETDDoneEnable Register	ETD done mask.
0x1002 40D8–40DC	RESERVED	—
0x1002 40E0	FrameNumber Register	The current Frame Number.
0x1002 40E4	LowSpeedThreshold Register	Defines the Low Speed Threshold value
0x1002 40E8	RootHubDescriptorA Register	Root Hub Characteristics Register 1
0x1002 40EC	RootHubDescriptorB Register	Root Hub Characteristics Register 2
0x1002 40F0	RootHubStatus Register	Root Hub Status and Change Bits
0x1002 40F4	PortStatus[1:3] Register	Port 1 Status and Change Bits
0x1002 40F8	PortStatus[1:3] Register	Port 2 Status and Change Bits
0x1002 40FC	PortStatus[1:3] Register	Port 3 Status and Change Bits

32.10.1 Effect of Resets on Host Controller Registers

There are several reset sources for the USB Host Controller: system master reset, a transition to Host Controller USB Reset, state and software-initiated controller resets. The software-initiated controller resets have the same effect as the system reset except they can selectively reset particular blocks. These resets are accomplished by writing to the ResetControl Register with ResetHostSIE, ResetRootHub, and/or ResetHostController bits SET. The USB bus reset is also generated by software. Writing to the HostControl Register and setting the HostControllerUSBState to USBReset, will reset the Host Controller and the Root Hub. After the reset has completed, the Host Controller will automatically enter the USBSuspend state.

32.10.2 Host Control Register

HOST_CTRL — Host Control Register — Addr 0x10024080

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	HCRESET														SCHEDOVR	
TYPE	rw	r	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												RMTWUEN	HCUSBSTE		CTLBLKSR	
TYPE	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-28. Host Control Register

Name	Description	Settings
HCRESET Bit 31	**Host Controller Reset**—Writing a 1 will generate a software-initiated hardware reset. Writing a '0' leaves this bit unchanged.	0 = No effect 1 = S/W initiated H/W reset
Reserved Bits 30–18	Reserved, bits should read zero	
SCHEDOVR Bits 17–16	**Scheduler Overrun Count**—This count is incremented each time a Scheduler Overrun event is detected in the Host Controller.	see description
Reserved Bits 15–5	Reserved, bits should read zero	
RMTWUEN Bit 4	**Remote Wake-Up Enable**—Setting this bit allows the Host to Wake-up when a resume event is detected downstream.	0 = Host does not respond to resume event 1 = Host wakes up when resume event detected
HCUSBSTE Bits 3–2	**Host Controller USB State**—Software uses these bits to control the state of the USB state machine. Writing these states controls the Root Hub.	00 = USB reset 01 = USB resume 10 = USB operational 11 = USB suspend
CTLBLKSR Bits 1–0	**Control Bulk Service Ratio**—This setting dictates the number of CONTROL packets sent out for each BULK packet.	00 = 1:1 01 = 2:1 10 = 3:1 11 = 4:1

32.10.3 System Interrupt Status Register

SYSISR — System Interrupt Status Register — Addr 0x10024088

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	rw	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										PSCINT	FMOFINT	HERRINT	RESDETINT	SOFINT	DONEINT	SORINT
TYPE	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-29. System Interrupt Status Register Description

Name	SW	HW	Description	Setting
Reserved Bits 31–7			Reserved	
PSCINT Bit 6	RC	S	**Port Status Change Interrupt**—When asserted, indicates that a Port Status has changed.	0 = Port status has not changed 1 = Port status has changed
FMOFINT Bit 5	RC	S	**Frame Number Overflow Interrupt**—When asserted, indicates that the Frame Number register has over flowed.	0 = Frame register has not overflowed 1 = Frame register overflow
HERRINT Bit 4	RC	S	**Host Error Interrupt**—When asserted, indicates that the Host has encountered a major scheduling error.	0 = No scheduling error 1 = Scheduling error
RESDETINT Bit 3	RC	S	**Resume Detected Interrupt**—When asserted, this bit indicates that the Host has detected a resume transition on the USB bus.	0 = No resume transition detected 1 = Resume transition detected
SOFINT Bit 2	RC	S	**Start Of Frame Interrupt**—When asserted, this indicates that an SOF has been transmitted.	0 = No SOF xmitted 1 = SOF xmitted
DONEINT Bit 1	RC	S	**Done Register Interrupt**—When asserted, this indicates that there are completed ETDs on the ETDDoneStatus Register.	0 = No completed ETDs 1 = Completed ETDs
SORINT Bit 0	RC	S	**Scheduler Overrun Interrupt**—When asserted, indicates that the Host has experienced a Scheduling Overrun condition.	0 = No scheduler overrun 1 = Scheduler overrun

32.10.4 System Interrupt Enable Register

SYSIEN — System Interrupt Enable Register — Addr 0x1002408C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	rw	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										PSCINT	FMOFINT	HERRINT	RESDETINT	SOFINT	DONEINT	SORINT
TYPE	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-30. System Interrupt Status Register Description

Name	Description	Setting
Reserved Bits 31–7	Reserved	
PSCINT Bit 6	**Port Status Change Interrupt Enable**—When SET, this will allow the *PortStatusChangeInterrupt* to be asserted.	0 = Port status interrupt not enabled 1 = Port status interrupt enabled
FMOFINT Bit 5	**Frame Number Overflow Interrupt Enable**—When SET, this will allow the *FrameNumberOverflowInterrupt* to be asserted.	0 = Frame number interrupt not enabled 1 = Frame number interrupt enabled
HERRINT Bit 4	**Host Error Interrupt Enable**—When SET, this will allow the *HostErrorInterrupt* to be asserted.	0 = Host Error interrupt not enabled 1 = Host Error interrupt enabled
RESDETINT Bit 3	**Resume Detected Interrupt Enable**—When SET, this will allow the *ResumeDetectedInterrupt* to be asserted.	0 = Resume Detect interrupt not enabled 1 = Resume Detect interrupt enabled
SOFINT Bit 2	**Start Of Frame Interrupt Enable**—When SET, this will allow the *StartOfFrameInterrupt* to be asserted.	0 = SOF interrupt not enabled 1 = SOF interrupt enabled
DONEINT Bit 1	**Done Register Interrupt Enable**—When SET, this will allow the *DoneRegisterInterrupt* to be asserted.	0 = Done register interrupt not enabled 1 = Done register interrupt enabled
SORINT Bit 0	**Scheduler Overrun Interrupt Enable**—When SET, this will allow the *SchedulerOverrunInterrupt* to be asserted.	0 = Scheduler overrun interrupt not enabled 1 = Scheduler overrun interrupt enabled

32.10.5 X Buffer Interrupt Status Register

XBUFSTAT — X Buffer Interrupt Status Register — Addr 0x10024098

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	XBUF31INT	XBUF30INT	XBUF29INT	XBUF28INT	XBUF27INT	XBUF26INT	XBUF25INT	XBUF24INT	XBUF23INT	XBUF22INT	XBUF21INT	XBUF20INT	XBUF19INT	XBUF18INT	XBUF17INT	XBUF16INT
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	XBUF15INT	XBUF14INT	XBUF13INT	XBUF12INT	XBUF11INT	XBUF10INT	XBUF09INT	XBUF08INT	XBUF07INT	XBUF06INT	XBUF05INT	XBUF04INT	XBUF03INT	XBUF02INT	XBUF01INT	XBUF00INT
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-31. X Buffer Interrupt Status Register Description

Name	SW	HW	Description
XBUFnINT Bits 31–0	RC	S	**X Buffer <n> Interrupt**—When asserted indicates that the X buffer of ETD <n> has been emptied (for OUTs) or filled (for INs) by the host. Writing the asserted bit back to the register clears this bit.

32.10.6 Y Buffer Interrupt Status Register

YBUFSTAT — Y Buffer Interrupt Status Register — Addr 0x1002409C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	YBUF31INT	YBUF30INT	YBUF29INT	YBUF28INT	YBUF27INT	YBUF26INT	YBUF25INT	YBUF24INT	YBUF23INT	YBUF22INT	YBUF21INT	YBUF20INT	YBUF19INT	YBUF18INT	YBUF17INT	YBUF16INT
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	YBUF15INT	YBUF14INT	YBUF13INT	YBUF12INT	YBUF11INT	YBUF10INT	YBUF09INT	YBUF08INT	YBUF07INT	YBUF06INT	YBUF05INT	YBUF04INT	YBUF03INT	YBUF02INT	YBUF01INT	YBUF00INT
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-32. Y Buffer Interrupt Status Register Description

Name	SW	HW	Description
YBUFnINT Bits 31–0	RC	S	**Y Buffer <n> Interrupt**—When asserted indicates that the Y buffer of ETD <n> has been emptied (for OUTs) or filled (for INs) by the host. Writing the asserted bit back to the register clears this bit.

32.10.7 XY Interrupt Enable Register

XYINTEN — XY Interrupt Enable Register — Addr 0x100240A0

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	XY31IEN	XY30IEN	XY29IEN	XY28IEN	XY27IEN	XY26IEN	XY25IEN	XY24IEN	XY23IEN	XY22IEN	XY21IEN	XY20IEN	XY19IEN	XY18IEN	XY17IEN	XY16IEN
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	XY15IEN	XY14IEN	XY13IEN	XY12IEN	XY11IEN	XY10IEN	XY09IEN	XY08IEN	XY07IEN	XY06IEN	XY05IEN	XY04IEN	XY03IEN	XY02IEN	XY01IEN	XY00IEN
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-33. XY Interrupt Enable Register Description

Name	Description
XYnIEN Bits 31–0	**XY <n> Interrupt Enable**—When set allows for an interrupt for both X and Y buffers on ETD <n> to be asserted.

32.10.8 X Filled Status Register

Software toggles the bits in this register by writing to it with the appropriate bits SET.

For an OUT ETD, the application indicates that the X Buffer has been filled and is ready for transfer by setting the XFILLED<N>STATUS bit. Once the buffer has been completely emptied by the Host Controller this bit is cleared.

For an IN ETD, the host indicates that it has completely filled the X Buffer by setting the appropriate XFILLED<N>STATUS bit. Once the buffer has been completely emptied by software this bit should be cleared.

XFILLSTAT — X Filled Status Register — Addr 0x100240A8

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	XFILL31I	XFILL30I	XFILL29I	XFILL28I	XFILL27I	XFILL26I	XFILL25I	XFILL24I	XFILL23I	XFILL22I	XFILL21I	XFILL20I	XFILL19I	XFILL18I	XFILL17I	XFILL16I
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	XFILL15I	XFILL14I	XFILL13I	XFILL12I	XFILL11I	XFILL10I	XFILL09I	XFILL08I	XFILL07I	XFILL06I	XFILL05I	XFILL04I	XFILL03I	XFILL02I	XFILL01I	XFILL00I
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-34. X Filled Status Register Description

Name	Description
XFILLn Bits 31–0	**X Filled <n> Status**—When SET, this indicates that the corresponding X Buffer is Full. In the case of a ZLP (Zero Length Packet), there will be no XFilled assertion/de-assertion and instead the Done Flag will be raised. Write 1 to clear bit.

32.10.9 Y Filled Status Register

Same operation as the X Filled Status Register except the action is for the Y Buffer.

YFILLSTAT — Y Filled Status Register — Addr 0x100240AC

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	YFILL31I	YFILL30I	YFILL29I	YFILL28I	YFILL27I	YFILL26I	YFILL25I	YFILL24I	YFILL23I	YFILL22I	YFILL21I	YFILL20I	YFILL19I	YFILL18I	YFILL17I	YFILL16I
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	YFILL15I	YFILL14I	YFILL13I	YFILL12I	YFILL11I	YFILL10I	YFILL09I	YFILL08I	YFILL07I	YFILL06I	YFILL05I	YFILL04I	YFILL03I	YFILL02I	YFILL01I	YFILL00I
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-35. Y Filled Status Register Description

Name	Description
YFILLn Bits 31–0	**Y Filled <n> Status**—When SET, this indicates that the corresponding Y Buffer is Full. In the case of a ZLP (Zero Length Packet), there will be no YFilled assertion/de-assertion and instead the Done Flag will be raised. Write 1 to clear bit.

32.10.10 ETD Enable Set Register

The Host Controller supports a maximum of 32 ETDs. This register is used by software to tell the Host Controller that an ETD is ready to be processed. When the descriptor is properly configured and the transfer is ready to commence, the corresponding bit should be set in this register. The bit is cleared by the corresponding bit in the ETD Clear Register.

ETDENSET — ETD Enable Set Register — Addr 0x100240C0

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ETD31SET	ETD30SET	ETD29SET	ETD28SET	ETD27SET	ETD26SET	ETD25SET	ETD24SET	ETD23SET	ETD22SET	ETD21SET	ETD20SET	ETD19SET	ETD18SET	ETD17SET	ETD16SET
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETD15SET	ETD14SET	ETD13SET	ETD12SET	ETD11SET	ETD10SET	ETD09SET	ETD08SET	ETD07SET	ETD06SET	ETD05SET	ETD04SET	ETD03SET	ETD02SET	ETD01SET	ETD00SET
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-36. ETD Enable Set Register Description

Name	Description
ETDnSET Bits 31–0	**ETD<n>SET**—When bit is set it indicates that an ETD is enabled and ready to be processed by the host.

32.10.11 ETD Enable Clear Register

When this bit is set, the ETDSet Register will be cleared to 0. The ETD register cannot be cleared by writing a 0 to the ETDSet Register.

ETDENCLR — ETD Enable Clear Register — Addr 0x100240C4

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ETD31CLR	ETD30CLR	ETD29CLR	ETD28CLR	ETD27CLR	ETD26CLR	ETD25CLR	ETD24CLR	ETD23CLR	ETD22CLR	ETD21CLR	ETD20CLR	ETD19CLR	ETD18CLR	ETD17CLR	ETD16CLR
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETD15CLR	ETD14CLR	ETD13CLR	ETD12CLR	ETD11CLR	ETD10CLR	ETD09CLR	ETD08CLR	ETD07CLR	ETD06CLR	ETD05CLR	ETD04CLR	ETD03CLR	ETD02CLR	ETD01CLR	ETD00CLR
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-37. ETD Enable Clear Register Description

Name	Description
ETDnCLR Bits 31–0	**ETD<n>CLear**—When SET indicates that ETD <n> is disabled. This register cannot be read.

32.10.12 Immediate Interrupt Register

In order to reduce the number of interrupts occurring in a frame, all ETD<n>DoneStatus are flagged on the SOF following the retirement of the ETD. However, if Software wishes to see the done status flagged immediately when the ETD is retired it can by setting the correct bit in this register.

IMMEDINT — Immediate Interrupt Register — Addr 0x100240CC

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IMMINT31	IMMINT30	IMMINT29	IMMINT28	IMMINT27	IMMINT26	IMMINT25	IMMINT24	IMMINT23	IMMINT22	IMMINT21	IMMINT20	IMMINT19	IMMINT18	IMMINT17	IMMINT16
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IMMINT15	IMMINT14	IMMINT13	IMMINT12	IMMINT11	IMMINT10	IMMINT09	IMMINT08	IMMINT07	IMMINT06	IMMINT05	IMMINT04	IMMINT03	IMMINT02	IMMINT01	IMMINT00
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-38. Immediate Interrupt Register Description

Name	Description
IMMINTn Bits 31–0	**Immediate Interrupt <n>**—When SET the Host Controller will assert an interrupt immediately upon retirement of an ETD instead of waiting until the SOF.

32.10.13 ETD Done Status Register

This register indicates which ETD transfers have been completed and are waiting for further Software instructions. The ETDDoneStatus Register is updated immediately upon retirement of an ETD, however, it will not assert an interrupt until the SOF marker unless the corresponding ImmediateInterrupt bit is SET. Software should read this register to determine which ETDs are in need of service.

ETDDONESTAT — ETD Done Status Register — Addr 0x100240D0

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ETD31DONE	ETD30DONE	ETD29DONE	ETD28DONE	ETD27DONE	ETD26DONE	ETD25DONE	ETD24DONE	ETD23DONE	ETD22DONE	ETD21DONE	ETD20DONE	ETD19DONE	ETD18DONE	ETD17DONE	ETD16DONE
TYPE	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETD15DONE	ETD14DONE	ETD13DONE	ETD12DONE	ETD11DONE	ETD10DONE	ETD09DONE	ETD08DONE	ETD07DONE	ETD06DONE	ETD05DONE	ETD04DONE	ETD03DONE	ETD02DONE	ETD01DONE	ETD00DONE
TYPE	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-39. ETD Done Status Register Description

Name	Description
ETDnDONE Bits 31–0	**ETD <n> Done Status**—When asserted indicates that *ETD<n>DoneStatus* has been retired, either through normal operation or through an error condition. The *CompletionCode* field of the ETD should be read to determine the reason for retirement. A complete list of completion codes is found in ETD completion code descriptions inTable 32-2.

32.10.14 ETD Done Enable Register

This register indicates which ETD<n>DoneStatus will generate an interrupt when set. This applies to interrupts generated either immediately or on the SOF marker.

ETDDONEN — ETD Done Enable Register — Addr 0x100240D4

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ETD31DNEN	ETD30DNEN	ETD29DNEN	ETD28DNEN	ETD27DNEN	ETD26DNEN	ETD25DNEN	ETD24DNEN	ETD23DNEN	ETD22DNEN	ETD21DNEN	ETD20DNEN	ETD19DNEN	ETD18DNEN	ETD17DNEN	ETD16DNEN
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETD15DNEN	ETD14DNEN	ETD13DNEN	ETD12DNEN	ETD11DNEN	ETD10DNEN	ETD09DNEN	ETD08DNEN	ETD07DNEN	ETD06DNEN	ETD05DNEN	ETD04DNEN	ETD03DNEN	ETD02DNEN	ETD01DNEN	ETD00DNEN
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-40. ETD Done Enable Register Description

Name	Description
ETDnDNEN Bits 31–0	**ETD<n>DoneEnable**—When SET allows ETD <n> to generate done interrupts. The *ETD<n>DoneEnable* must be SET for both immediate and SOF marker interrupts.

32.10.15 Frame Number Register

FRMNUB — Frame Number Register — Addr 0x100240E0

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FRMNUMB															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-41. Frame Number Register Description

Name	Description
Reserved Bits 31–16	Reserved
FRMNUMB Bits 15–0	**Frame Number**—This contains the current frame number for the Host Controller. This number is placed in the SOF packet used to signal the beginning of the new frame.

32.10.16 Low Speed Threshold Register

LSTHRESH — Low Speed Threshold Register — Addr 0x100240E4

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						LSTHRESH										
TYPE	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0
	0x0628															

Table 32-42. Low Speed Threshold Register Description

Name	Description
Reserved Bits 31–11	Reserved
LSTHRESH Bits 10–0	**Low Speed Threshold**—This is the number in USB full speed bit times that are required to be remaining in a frame to allow a low speed packet to be transmitted. All low speed packets have a *MaxPacketSize* maximum of 8 bytes, so this should be set to the number of bit times needed to transmit an 8 byte packet. The default value of 628h should not be changed for normal operation.

32.10.17 Root Hub Descriptor A Register

ROOTHUBA — Root Hub Descriptor A Register — Addr 0x100240E8

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PWRTOGOOD															
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	0x0100															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				NOOVRCURP	OVRCURPM	DEVTYPE	PWRSWTMD	NPWRSWT	NDNSTMPRT							
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1
	0x0103															

Table 32-43. Root Hub Descriptor A Register Description

Name	Description
PWRTOGOOD Bits 31–24	**Power On To Power Good Time**—This byte specifies the duration that software must wait before accessing a powered-on port of the Root Hub.
Reserved Bits 23–13	Reserved
NOOVRCURP Bit 12	**No Over Current Protection**—This bit describes how the over current status for the Root Hub is reported. In this implementation, the over current status is reported collectively for all downstream ports.
OVRCURPM Bit 11	**Over Current Protection Mode**—This bit describes how the over current status for the Root Hub ports are reported. This implementation uses gang-power and gang-over current reporting.
DEVTYPE Bit 10	**Device Type**—This bit specifies that the Root Hub is not a compound device. The Root Hub is not permitted to be a compound device. This field should always read as 0.
PWRSWTMD Bit 9	**Power Switching Mode**—This bit is used to specify how the power switching of the Root Hub ports is controlled. In this implementation, each port is powered individually. This allows for power to be controlled on a per-port basis. The port responds only to port power commands (**Set/ClearPortPower**).
NOPWRSWT Bit 8	**No Power Switching**—All ports are power switched. This bit will always read as SET.
NDNSTMPRT Bits 7–0	**Number Downstream Ports**—These bits specify the number of downstream ports supported by the Root Hub.

32.10.18 Root Hub Descriptor B Register

ROTHUBB — Root Hub Descriptor B Register — Addr 0x100240EC

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									PRTPWRCM							
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
									0x0007							

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									DEVREMOVE							
TYPE	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
									0x0000							

Table 32-44. Root Hub Descriptor B Register Description

Name	Description
Reserved Bits 31–24	Reserved
PRTPWRCM Bits 23–16	**Port Power Control Mask**—Each bit indicates if a port is affected by a global power control command when **PowerSwitchingMode** is set. When set, the port's power state is only affected by per-port power control (**Set/ClearPortPower**). When cleared, the port is controlled by the global power switch (**Set/ClearGlobalPower**). bit 0: this port can be OTG/ #1 port (during HNP negotiation) bit 1: Port #2 bit 2: Port #3
Reserved Bits 15–8	Reserved
DEVREMOVE Bits 7–0	**Device Removable**—Each bit is dedicated to a port of the Root Hub. When cleared, the attached device is removable. When set, the attached device is not removable. bit 0: this port can be OTG/ #1 port (during HNP negotiation) bit 1: Device attached to Port #2 bit 2: Device attached to Port #3

32.10.19 Root Hub Status Register

ROOTSTAT — Root Hub Status Register — Addr 0x100240F0

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CLRRMTWUE														OVRCURCHG	
TYPE	w	r	r	r	r	r	r	r	r	r	r	r	r	r	rc	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	0x0007															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DEVCONWUE														OVRCURI	LOCPWRS
TYPE	rw	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-45. Root Hub Status Register Description

Name	Description
CLRMTWUE Bit 31	**Clear Remote Wake-Up Enable**—Setting this bit clears *DeviceRemoteWakeupEnable.*
Reserved Bits 30–18	Reserved
OVRCURCHG Bit 17	**Over Current Indicator Change**—This bit is set by hardware when a change has occurred to the OCI field of this register. The HCD clears this bit by writing a 1. Writing a 0 has no effect.
Reserved Bit 16	Reserved
DEVCONWUE Bit 15	**Device Connect Wake-up Enable**—This bit enables a *ConnectStatusChange* bit as a resume event, causing a USBSUSPEND to USBRESUME state transition and setting the *ResumeDetected* interrupt. 0 = *ConnectStatusChange* is not a remote wakeup event. 1 = *ConnectStatusChange* is a remote wakeup event. Writing a 1 sets *DeviceRemoveWakeupEnable*. Writing a 0 has no effect.
Reserved Bits 14–2	Reserved
OVRCURI Bit 1	**Over Current Indicator**—This bit reports over current conditions when the global reporting is implemented. When set, an over current condition exists. When cleared, all power operations are normal. If per-port over current protection is implemented this bit is always 0
LOCPWRS Bit 0	**Local Power Status**—The Root Hub does not support the local power status feature; thus, this bit is always read as 0.

32.10.20 Port Status 1–3 Register

PORTSTAT1 PORTSTAT2 PORTSTAT3	Port Status 1–3 Register	Addr 0x100240F4 0x100240F8 0x100240FC

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												PRTRSTSC	OVRCURIC	PRTSTATSC	PRTEBLSC	CONNECTSC
TYPE	r	r	r	r	r	r	r	r	r	r	r	rc	rc	rc	rc	rc
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							LSDEVCON	PRTPWRST				PRTRSTST	PRTOVRCURI	PRTSUSPST	PRTENABST	CURCONST
TYPE	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-46. Port Status 1–3 Register Description

Name	Description
Reserved Bits 31–21	Reserved
PRTRSTSC Bit 20	**Port Reset Status Change**—This bit is SET at the end of the 50-ms port reset signal. Only a write back with this bit SET will clear this bit.
OVRCURIC Bit 19	**Port Over Current Indicator Change**—This bit is SET when an *PortOverCurrentIndicator* has changed on this port and is only valid if over current conditions are reported on a per-port basis. Only a write back with this bit SET will clear this bit.
PRTSTATSC Bit 18	**Port Suspend Status Change**—This bit is set when the resume sequence has been fully completed. A write back with this bit SET or when *ResetStatusChange* is SET when will clear this bit.
PRTENBLSC Bit 17	**Port Enable Status Change**—This bit is set when hardware events cause the *PortEnableStatus* bit to be cleared. Only a write back with this bit SET will clear this bit.
CONNECTSC Bit 16	**Connect Status Change**—When a connect or disconnect event occurs this bit will be SET. A write with this bit SET will clear this bit. Writing a 0 has no effect. If *CurrentConnectStatus* is cleared when a *SetPortReset*, *SetPortEnable*, or **SetPortSuspend** write occurs, this bit is SET to force the driver to re-evaluate the connection status since these writes should not occur if the port is disconnected. Only a write back with this bit SET will clear this bit.
Reserved Bits 15–10	Reserved

Table 32-46. Port Status 1–3 Register Description (continued)

Name	Description
LSDEVCON Bit 9	**(Read) Low Speed Device Attached**—This bit indicates the speed of the device attached to this port. When SET, a low speed device is attached to this port. When CLEARED, a full speed device is attached to this port. This field is valid only when the *CurrentConnectStatus* is set. **(Write) Clear Port Power**—The HCD clears the *PortPowerStatus* bit by a write with this bit SET. Writing with this bit CLEARED has no effect.
PRTPWRST Bit 8	**(Read) PortPowerStatus**—This bit reflects the port's power status. This bit is cleared if an over current condition is detected. HCD sets this bit by writing *SetPortPower*. HCD clears this bit by writing *ClearPortPower*. When port power is disabled, *CurrentConnectStatus*, *PortEnableStatus*, *PortSuspendStatus*, and *PortResetStatus* should be reset. **(Write) SetPortPower**—The HCD writes a 1 to set the *PortPowerStatus* bit. Writing a 0 has no effect.
Reserved Bits 7–5	
PRTRSTST Bit 4	**(Read) Port Reset Status**—When this bit is SET by a write to *SetPortReset*, and will remain asserted until the port reset signaling is completed. When reset is completed, this bit is cleared and *PortResetStatusChange* is SET. This bit cannot be set if *CurrentConnectStatus* is cleared. **(Write) Set Port Reset**—The HCD sets the port reset signaling by writing a 1 to this bit. Writing a 0 has no effect. If *CurrentConnectStatus* is cleared, this write does not set *PortResetStatus*, but instead sets *ConnectStatusChange*. This informs the driver that it attempted to reset a disconnected port.
PRTOVRCURI Bit 3	**(Read) Port Over Current Indicator**—This bit is only valid when the Root Hub is configured so that over current conditions are reported on a per-port basis. If per-port over current reporting is not supported, this bit is always read as 0. If cleared, all power operations are normal for this port. If SET, an overcurrent condition exists on this port. **(Write) Clear Suspend Status**—The HCD writes a 1 to initiate a resume. Writing a 0 has no effect. A resume is initiated only if PortSuspendStatus is SET.
PRTSUSPST Bit 2	**(Read) PortSuspendStatus**—When SET bit indicates the port is suspended or in the resume sequence. It is SET by a *SetSuspendState* write and at the end of the resume interval. This bit cannot be set if *CurrentConnectStatus* is cleared. This bit is also cleared when *PortResetStatusChange* is set at the end of the port reset or when the Host Controller is placed in the USBRESUME state. **(Write) SetPortSuspend**—The HCD sets the *PortSuspendStatus* bit by writing a 1 to this bit. Writing a 0 has no effect. If *CurrentConnectStatus* is cleared, this write does not set *PortSuspendStatus*; instead it sets *ConnectStatusChange*. This informs the driver that it attempted to suspend a disconnected port.

Table 32-46. Port Status 1–3 Register Description (continued)

Name	Description
PRTENABST Bit 1	**(Read) PortEnableStatus**—When SET this bit indicates whether the port is enabled. The Root Hub will clear this bit when an over current condition, disconnect event, the power is switched-off, or operational bus error has occurred. A change in this bit causes *PortEnabledStatusChange* to be SET. HCD sets this bit by writing *SetPortEnable* and clears it by writing *ClearPortEnable*. This bit cannot be set when *CurrentConnectStatus* is cleared. This bit is also set, if not already, at the completion of a port reset when *ResetStatusChange* is set or port suspend when *SuspendStatusChange* is set. 0 = port is disabled 1 = port is enabled **(Write) SetPortEnable**—The HCD sets PortEnableStatus by writing a 1. Writing a 0 has no effect. If *CurrentConnectStatus* is cleared, this write does not set *PortEnableStatus*, but instead sets *ConnectStatusChange*. This informs the driver that it attempted to enable a disconnected port.
CURCONST Bit 0	**(Read) Current Connect Status**—This bit reflects the current state of the downstream port. A read with this bit SET indicates a device is connected. **(Write) ClearPortEnable**—The HCD writes a 1 to this bit to clear the *PortEnableStatus* bit. Writing a 0 has no effect. The *CurrentConnectStatus* is not affected by any write.

32.11 USB Function

The USB OTG module has a built-in full speed function controller supporting up to 32 unidirectional endpoints or 16 bi-directional endpoints. Much like the host, each endpoint is described by an Endpoint Descriptor and the function hardware contains enough data structure memory for a maximum of 16 IN endpoints and 16 OUT endpoints. The type and speed can be assigned and programmed by the user. Please note that EP0 is a dedicated control endpoint and must remain that way. This EP has special significance to the USB protocol.

To save power, the system may be placed in USB Suspend mode as defined by the USB specification. There are various ways to awake from this power savings mode and all ways described by the USB specification are supported.

It is mandatory for all USB Devices to allocate endpoint 0 (EP0) for device identification and setup (enumeration). The EP0 is a dedicated endpoint for control transfers. Moreover, it also serves as a communication channel for the host to send commands and data to the device, and to retrieve data from the device. Interactions through EP0 are referred to as “requests” in Chapter 9 of the USB Specification. A request can be system, class, or vendor specific. After reset software must configure endpoint 0 to be a control endpoint.

The 16 logical general-purpose endpoints are identical data structures. Except for endpoint 0, which should be configured as a control endpoint, each of the 15 endpoints can be programmed to work as a bi-directional bulk, interrupt, control or isochronous endpoint with an X-Buffer and a Y-Buffer allocated in the data memory.

32.11.1 OTG-IPFC Operation: Software

After system reset or a Software Reset, the Endpoint Descriptors are reset and disabled. The control endpoint needs to be configured and enabled before the host is able to start communication with the device. It is important that Software finishes the configuration of all necessary endpoints or software must delay the status stage of the USB request, such as SET_ADDRESS, by “NAKing” the IN token sent from the host in the status stage, until the endpoint programming process has finished. By then an empty packet is returned to the host in response to the IN token of the status stage.

In addition to filling out endpoint related properties, an X-Buffer and a Y-Buffer must be allocated in Data Memory. Their size, which must be a multiple of the maximum packet size of the pipe, is application dependent.

32.11.2 Transfer Anticipation

To anticipate a transfer, the user software shall:

- Prepare the data buffer in the microprocessor space. For OUT (host to device) transfers, this is the destination of the USB transfer. For IN (device to host) transfers, this is the source.
- The size and address of this data buffer in microprocessor space must be registered with the associated endpoint.
- Make sure the endpoint is ready to undertake a new transfer.
- Enable the endpoint.

32.11.3 Transfer Processing

The transfer is considered ended if:

- For an IN (device to host) endpoint, all anticipated data has been retrieved by the host.
- For an OUT (host to device) endpoint, all anticipated data has been received; or a transaction with a data packet of less than the maximum packet size is received.

An interrupt is generated upon completion of a transfer. It should be noted that interrupts will occur before, during, and after the transfer, not just on the completion of the transfer.

Table 32-47 describes the EP memory map.

Table 32-47. EP Memory Map

EP Address	Description
0x100244200	EP0 OUT DWORD0
0x10024404	EP0 OUT DWORD1
0x10024408	EP0 OUT DWORD2
0x1002440C	EP0 OUT DWORD3
0x10024410	EP0 IN DWORD0
0x10024414	EP0 IN DWORD1
0x10024418	EP0 IN DWORD2

Table 32-47. EP Memory Map

EP Address	Description
0x1002441C	EP0 IN DWORD3
…	…

32.11.4 Function Endpoint Type Format

Function Endpoint Type DWORD0 Format

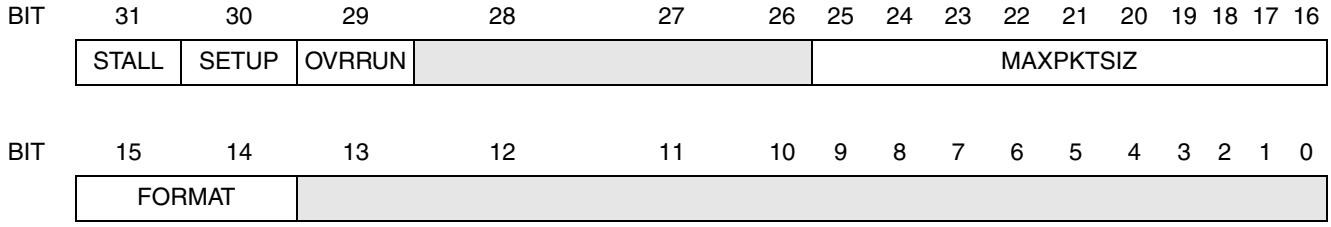


Table 32-48. Function Endpoint Type DWORD0 Description

Name	Description	Setting
STALL Bit 31	**Stall Received**—When SET this bit indicates to the function controller to stall all incoming packets to this endpoint.	0 = normal operation 1 = stall incoming packets
SETUP Bit 30	**Setup Received**—This bit indicates that the last received packet was a SETUP type packet.	0 = Last packet not SETUP type 1 = Last packet was SETUP type
OVERRUN Bit 29	**Data Overrun Detected**—The function received more data than **MaxPacketSize** resulting in a Data Overrun Detected on the last packet	0 = No data overun 1 = Data overrun has occured
Reserved Bits 28–26	Reserved	
MAXPKTSIZ Bits 25–16	**Maximum Packet Size**—This field contains the maximum allowable packet size for the EP.	see description
FORMAT Bits 15–14	**Endpoint Format**—Defines the packet format for the EP.	00 = Control 01 = Isochronous 10 = Bulk 11 = Interrupt
Reserved Bits 13–0	Reserved	

32.11.5 Control/Bulk/Interrupt Endpoint Format

Control/Bulk/Interrupt Endpoint Format Word 1

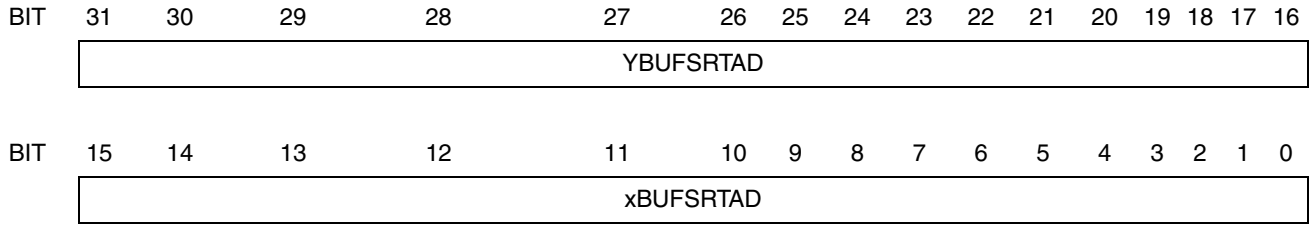


Table 32-49. Control/Bulk/Interrupt Endpoint WORD1

Name	SW	HW	Description
YBUFSRTAD Bits 31–16	RW	R	**Y Buffer Start Address**—Contains the physical start address of the mega-buffer Y allocated for this EP that will be accessed for transfer to/from the endpoint. This address is the byte address location in the memory. Since the IP USB OTG module is a 32 bits, the last two bits of the address will be dropped.
XBUFSRTAD Bits 15–0	RW	R	**X Buffer Start Address**—Contains the physical start address of the mega-buffer X allocated for this EP that will be accessed for transfer to/from the endpoint. This address is the byte address location in the memory. Since the IP USB OTG module is a 32 bits, the last two bits of the address will be dropped.

Table 32-50. Control/Bulk/Interrupt Endpoint WORD2

Name	Description
RESERVED Bits 31–0	Reserved

Control/Bulk/Interrupt Endpoint Format Word 3

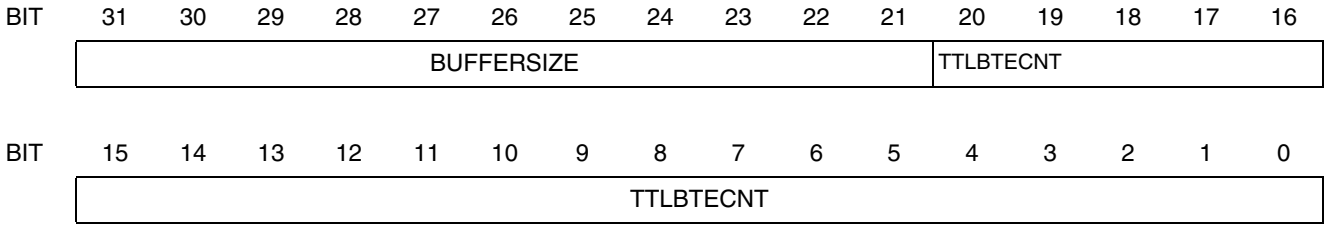


Table 32-51. Control/Bulk/Interrupt Endpoint WORD3

Name	Description
BUFFERSIZE Bits 31–21	**Buffer Size**—Size of X/Y Buffer. This value is given in bytes and is a value between 0–2047, corresponding to buffer sizes between 1 and 2 Kbytes. **Example:** If the BUFFERSIZE is to be set to 0x40 (64) bytes, software should write 0x3F in this field.
TTLBTECNT Bits 20–0	**Total Byte Count To Transfer**—This is the total number of bytes to be transferred by the EP. The EP can be set to transfer 0 - (2M-1) bytes.

32.11.5.1 Isochronous Endpoint

Isochronous Endpoint Word 1

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	YBUFSRTAD															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	xBUFSRTAD															

Table 32-52. Isochronous Endpoint WORD1

Name	SW	HW	Description
YBUFSRTAD Bits 31–16	RW	R	**Y Buffer Start Address**—Contains the physical start address of the mega-buffer Y allocated for this EP that will be accessed for transfer to/from the endpoint. This address is the byte address location in the memory. Since the IP USB OTG module is a 32 bits, the last two bits of the address will be dropped.
XBUFSRTAD Bits 15–0	RW	R	**X Buffer Start Address**—Contains the physical start address of the mega-buffer X allocated for this EP that will be accessed for transfer to/from the endpoint. This address is the byte address location in the memory. Since the IP USB OTG module is a 32 bits, the last two bits of the address will be dropped.

Table 32-53. Isochronous Endpoint WORD2 Description

Name	Description
RESERVED Bits 31–0	Reserved

Isochronous Endpoint Format Word 3

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FRAMECNT						PKTLEN1									

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							PKTLEN0									

Table 32-54. Isochronous Endpoint WORD3 Description

Name	Description	Setting
FRAMECNT Bit 31	**Frame Count**—Number of data packets described by this Isochronous EP.	0: There is 1 data packet 1: There is 2 data packets
Reserved Bits 30–26	Reserved	

Table 32-54. Isochronous Endpoint WORD3 Description (continued)

Name	Description	Setting
PKTLEN1 Bits 25–16	**Packet Length 1**—Length in bytes of the second isochronous packet.	
Reserved Bits 15–10	Reserved	
PKTLEN0 Bits 9–0	**Packet Length 0**—Length in bytes of the first isochronous packet.	

32.11.6 Function Registers

This section describes the registers used to control the USB functions. A summary of the registers used is shown on Table 32-55.

Table 32-55. Function Register Summary

Address	Register Bit Name	Description
0x1002 4040	FunctionCommandStatus Register	Current status of the function controller.
0x1002 4044	Device Address Register	Assigned address for the function controller from last enumeration.
0x1002 4048	SystemInterruptStatus Register	Interrupt vector.
0x1002 404C	SystemInterruptEnables Register	Interrupt mask.
0x1002 4050	XBufferInterruptStatus Register	X buffer Interrupt status.
0x1002 4054	YBufferInterruptStatus Register	Y buffer Interrupt status.
0x1002 4058	XYInterruptEnables	X/Y Interrupt enables.
0x1002 405C	XFilledStatus Register	X buffer filled status.
0x1002 4060	YFilledStatus Register	Y buffer filled status.
0x1002 4064	EndpointEnables Register	Informs the function controller which endpoints are enabled.
0x1002 4068	EndpointReady Register	If set and EP is enabled, then the transfer will begin. If not set and EP is enabled, then device returns NAKs.
0x1002 406C	ImmediateInterrupt Register	Informs the function controller that all endpoints that are set here should generate an interrupt immediately upon completion.
0x1002 4070	EndpointDoneStatus Register	Asserts which endpoints are done processing.
0x1002 4074	EndpointDoneEnable Register	Endpoints done mask.
0x1002 4078	EndpointToggleBits Register	The current toggle bit for the endpoint.
0x1002 407C	FrameNumber Register	The number of the last Start Of Frame.

32.12 Effect of Different Resets on Registers

There are several reset resources for the USB Function Controller: system reset and a software initiated reset. The software initiated reset has the same effect as the system reset, however it can be localized to a particular block. The different reset options are described in the ResetControl Register (0x010) in section

ResetControl Register (0x010). All of the reset bits are self-clearing (SC). The USB bus reset will trigger an interrupt if enabled to alert Software that a bus reset has been detected. This interrupt is triggered on both the entering and exiting of the USB bus reset state. Software can use this interrupt to then set the soft reset bit to reset the Function Controller, if desired.

32.12.1 Function Command Status Register

FUNCOMSTAT — Function Command Status Register — Addr 0x10024040

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									SOFT RESET				BAD ISOAP	SUSP DET	RSMIN PROG	RESET DET
TYPE	r	r	r	r	r	r	r	r	w	r	r	r	rw	rw	rw	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	0x0001															

Table 32-56. Function Command Status Register

Name	Description
Reserved Bits 31–8	Reserved
SOFTRESET Bit 7	**Soft Reset Function Controller**—Writing with this bit SET will generate a software-initiated hardware reset. Writing a '0' leaves this bit unchanged.
Reserved Bits 6–4	Reserved
BADISOAP Bit 3	**Bad ISO Accepted**—Writing with this bit SET will cause the Function Controller to accept an ISO packet with corrupted data, like data CRC error. When this bit is cleared, or by default, bad ISO packets are automatically dropped.
SUSPDET Bit 2	**Suspend Detected**—When Software reads this bit to be 1, it indicates that the USB bus is in the SUSPEND state. But it does not mean the function controller is in the SUSPEND state. A 0 indicates that the USB bus is not in the suspend state. But it does not mean the function controller is not in the suspend state. Writing a 0 leaves the hardware unchanged. Writing a 1 sets the Function port in the SUSPEND state and sets the transceiver in the power-savings mode. It will not affect the value of the physical register bit 2 which merely reflects the status. **Note:** If SuspendDetected=1, it does not mean the Function controller is in the SUSPEND state. It just means that the USB bus is in the suspend state. Software needs to write 1 to set the Function Controller in the suspend state. But the Function controller does not keep this state in a register.

Table 32-56. Function Command Status Register

Name	Description
RSMINPROG Bit 1	**Resume In Progress**—When Software reads this bit to be 1, it indicates that the USB bus and the Function controller are in the RESUME state. A 0 indicates that the USB bus and the Function controller are not in the RESUME state. Writing a 0 leaves the hardware unchanged. Writing a 1 generates a RESUME signal to the upstream port and lets the xcvr exit the power-saving mode. Writing a 1 is a RESUME command. It will not affect the value of the physical register bit which merely displays the current status.
RESETDET Bit 0	**USB Bus Reset Detected**—A read with this bit SET indicates that the USB device has detected bus reset.

32.12.2 Device Address Register

The value contained in this register is the Host assigned address. This is the ONLY address to which the function controller will respond. When there is a system reset or the ResetFunctionController bit is SET, this value will be cleared to the register's default value. In the status stage of the standard SET_ADDRESS request, software should write received device address into this register immediately after sending an empty packet to the Host.

DEVADDR — Device Address Register — Addr 0x10024044

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										DEVADDR						
TYPE	r	r	r	r	r	r	r	r	w	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-57. Device Address Register

Name	Description
Reserved Bits 31–7	Reserved
DEVADDR Bits 6–0	**Device Address**—This field contains the device address.

32.12.3 System Interrupt Status Register

SYSINTSTAT — System Interrupt Status Register — Addr 0x10024048

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												SOFDETINT	DONEREGINT	SUSPDETINT	RSMFINNT	RESETINT
TYPE	r	r	r	r	r	r	r	r	r	r	r	rc	rc	rc	rc	rc
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-58. System Interrupt Status Register Description

Name	Description
Reserved Bits 31–5	Reserved
SOFDETINT Bit 4	**SOF Detected Interrupt**—When asserted indicates that the USB device has received a Start Of Frame.
DONEREGINT Bit 3	**Done Register Interrupt**—When asserted this bit indicates one or more bits in EndpointDoneStatus Register (0x070) is set.
SUSPDETINT Bit 2	**Suspend Detected Interrupt**—When asserted indicates that the USB device has detected an active to suspend state on the bus.
RSMFININT Bit 1	**Resume Finished Interrupt**—When asserted indicates that the USB device has detected a suspend to active state change.
RESETINT Bit 0	**Reset Detected Interrupt**—The interrupt will be asserted on the rising edge and falling edge of the bus reset detection.

32.12.4 System Interrupt Enables Register

SYSINTEN — System Interrupt Enables Register — Addr 0x1002404C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												SOFDETIEN	DONEREGIEN	SUSPDETIEN	RSMFINIEN	RESETIEN
TYPE	r	r	r	r	r	r	r	r	r	r	r	rc	rc	rc	rc	rc
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-59. System Interrupt Enable Register Description

Name	Description
Reserved Bits 31–5	Reserved
SOFDETIEN Bit 4	**SOF Detected Interrupt Enable**—When SET this will allow the *SOFDetectedInterrupt* to be asserted.
DONEREGIEN Bit 3	**Done Register Interrupt Enable**—When SET this will allow the *DoneRegisterInterrupt* to be asserted.
SUSPDETIEN Bit 2	**Suspend Detected Interrupt Enable**—When SET this will allow the *SuspendDetectedInterrupt* to be asserted.
RSMFINIEN Bit 1	**Resume Finished Interrupt Enable**—When SET this will allow the *ResumeFinishedInterrupt* to be asserted.
RESETIEN Bit 0	**USB Bus Reset Detected Enable**—When SET this will allow the **USBBusResetDetected** to be asserted.

32.12.5 X Buffer Interrupt Status Register

XBUFINTSTAT — X Buffer Interrupt Status Register — Addr 0x10024050

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	XBUFFININT	XBUFFOUTINT	XBUFFININT	XBUFFOUTINT	XBUFFININT	XBUFFOUTINT	XBUFFININT	XBUFFOUTINT	XBUFFININT	XBUFFOUTINT	XBUFFININT	XBUFFOUTINT	XBUFFININT	XBUFFOUTINT	XBUFFININT	XBUFFOUTINT
TYPE	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	XBUFFININT	XBUFFOUTINT	XBUFFININT	XBUFFOUTINT	XBUFFININT	XBUFFOUTINT	XBUFFININT	XBUFFOUTINT	XBUFFININT	XBUFFOUTINT	XBUFFININT	XBUFFOUTINT	XBUFFININT	XBUFFOUTINT	XBUFFININT	XBUFFOUTINT
TYPE	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-60. X Buffer Interrupt Status Register

Name	Description
XBUFnININT All odd bits	**XBuffer<n>InInterrupt**—When asserted indicates that the XIN buffer of endpoint <n> has been emptied by the host. Writing the asserted bit back to the register clears this bit.
XBUFnOUTINT All even bits	**XBuffer<n>OutInterrupt**—When asserted indicates that the XOUT buffer of endpoint <n> has been filled by the host. Writing the asserted bit back to the register clears this bit.

32.12.6 Y Buffer Interrupt Status Register

YBUFINTSTAT — Y Buffer Interrupt Status Register — Addr 0x10024054

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	YBUFFININT	YBUFFOUTINT	YBUFFININT	YBUFFOUTINT	YBUFFININT	YBUFFOUTINT	YBUFFININT	YBUFFOUTINT	YBUFFININT	YBUFFOUTINT	YBUFFININT	YBUFFOUTINT	YBUFFININT	YBUFFOUTINT	YBUFFININT	YBUFFOUTINT
TYPE	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	YBUFFININT	YBUFFOUTINT	YBUFFININT	YBUFFOUTINT	YBUFFININT	YBUFFOUTINT	YBUFFININT	YBUFFOUTINT	YBUFFININT	YBUFFOUTINT	YBUFFININT	YBUFFOUTINT	YBUFFININT	YBUFFOUTINT	YBUFFININT	YBUFFOUTINT
TYPE	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-61. Y Buffer Interrupt Status Register

Name	Description
YBUFnININT All odd bits	**Y Buffer<n>InInterrupt**—When asserted indicates that the YIN buffer of endpoint <n> has been emptied by the host. Writing the asserted bit back to the register clears this bit.
YBUFnOUTINT All even bits	**Y Buffer<n>OutInterrupt**—When asserted indicates that the YOUT buffer of endpoint <n> has been filled by the host. Writing the asserted bit back to the register clears this bit.

32.12.7 XY Interrupt Enable Register

XYINT_INT_EN — XY Interrupt Enable Register — Addr 0x10024058

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	XY FINEN	XY FOUTEN	XY EINEN	XY EOUTEN	XY DINEN	XY DOUTEN	XY CINEN	XY COUTEN	XY BINEN	XY BOUTEN	XY AINEN	XY AOUTEN	XY 9INEN	XY 9OUTEN	XY 8INEN	XY 8OUTEN
TYPE	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	XY 7INEN	XY 7OUTEN	XY 6INEN	XY 6OUTEN	XY 5INEN	XY 5OUTEN	XY 4INEN	XY 4OUTEN	XY 3INEN	XY 3OUTEN	XY 2INEN	XY 2OUTEN	XY 1INEN	XY 1OUTEN	XY 0INEN	XY 0OUTEN
TYPE	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-62. XY Buffer Interrupt Status Register

Name	Description
XYBUFnININT All odd bits	**XY Buffer<n>InInterrupt**—When asserted indicates that the XYIN buffer of endpoint <n> has been emptied by the host. Writing the asserted bit back to the register clears this bit.
XYBUFnOUTINT All even bits	**XY Buffer<n>OutInterrupt**—When asserted indicates that the XYOUT buffer of endpoint <n> has been filled by the host. Writing the asserted bit back to the register clears this bit.

32.12.8 X Filled Status Register

Software toggles the bits in this register by writing to it with the appropriate bits SET.

For an IN endpoint, the application indicates that the X Buffer has been filled and is ready for transfer by setting the XFilled<n>InStatus bit. Once the buffer has been completely emptied by the Hardware, this bit is cleared.

For an OUT endpoint, the host indicates that it has completely filled the Xbuffer by setting the appropriate XFilled<n>OutStatus. Once the buffer has been completely emptied by Software, this bit should be cleared.

XFILLSTAT — X Filled Status Register — Addr 0x1002405C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	XFILLFIN	XFILLFOUT	XFILLEIN	XFILLEOUT	XFILLDIN	XFILLDOUT	XFILLCIN	XFILLCOUT	XFILLBIN	XFILLBOUT	XFILLAIN	XFILLAOUT	XFILL9IN	XFILL9OUT	XFILL8IN	XFILL8OUT
TYPE	s	rc	s	rc	s	rc	s	rc	s	rc	s	rc	s	rc	s	rc
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	XFILL7IN	XFILL7OUT	XFILL6IN	XFILL6OUT	XFILL5IN	XFILL5OUT	XFILL4IN	XFILL4OUT	XFILL3IN	XFILL3OUT	XFILL2IN	XFILL2OUT	XFILL1IN	XFILL1OUT	XFILL0IN	XFILL0OUT
TYPE	s	rc	s	rc	s	rc	s	rc	s	rc	s	rc	s	rc	s	rc
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-63. X Filled Status Register

Name	Description
XFILLnIN All odd bits	**XFilled<n>InStatus**—When a bit is asserted, it indicates to the Hardware that the <n> X Buffer has been filled and the function controller can begin sending the data on the next IN. In the case of a ZLP (Zero Length Packet), there will be no XFilled assertion/de-assertion and instead the Done Flag will be raised. This register is a toggle register. If it is set, writing a 1 will clear it. If it is cleared, writing a 1 will set it. Writing a 0 has no effect.
XFILLnOUT All even bits	**XFilled<n>OutStatus**—When a bit is cleared, it indicates to the Hardware that the <n> X Buffer has been drained and the function controller can begin receiving data from the next OUT. This register is a toggle register. If it is set, writing a 1 will clear it. If it is cleared, writing a 1 will set it. Writing a 0 has no effect.

32.12.9 Y Filled Status Register

This register has the same operation principles as the except for the Y Buffer.

YFILLSTAT — Y Filled Status Register — Addr 0x10024060

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	YFILLFIN	YFILLFOUT	YFILLEIN	YFILLEOUT	YFILLDIN	YFILLDOUT	YFILLCIN	YFILLCOUT	YFILLBIN	YFILLBOUT	YFILLAIN	YFILLAOUT	YFILL9IN	YFILL9OUT	YFILL8IN	YFILL8OUT
TYPE	s	rc	s	rc	s	rc	s	rc	s	rc	s	rc	s	rc	s	rc
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	YFILL7IN	YFILL7OUT	YFILL6IN	YFILL6OUT	YFILL5IN	YFILL5OUT	YFILL4IN	YFILL4OUT	YFILL3IN	YFILL3OUT	YFILL2IN	YFILL2OUT	YFILL1IN	YFILL1OUT	YFILL0IN	YFILL0OUT
TYPE	s	rc	s	rc	s	rc	s	rc	s	rc	s	rc	s	rc	s	rc
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-64. Y Filled Status Register

Name	Description
YFILLnIN All odd bits	**YFilled<n>InStatus**—When a bit is asserted, it indicates to the Hardware that the <n> Y Buffer has been filled and the function controller can begin sending the data on the next IN. In the case of a ZLP (Zero Length Packet), there will be no YFilled assertion/de-assertion and instead the Done Flag will be raised. This register is a toggle register. If it is set, writing a 1 will clear it. If it is cleared, writing a 1 will set it. Writing a 0 has no effect.
YFILLnOUT All even bits	**YFilled<n>OutStatus**—When a bit is cleared, it indicates to the Hardware that the <n> YY Buffer has been drained and the function controller can begin receiving data from the next OUT. This register is a toggle register. If it is set, writing a 1 will clear it. If it is cleared, writing a 1 will set it. Writing a 0 has no effect.

32.12.10 Endpoint Enables Register

The Function Controller can support up to 32 physical endpoints or 16 bi-directional endpoints. The controller uses the values contained in this register to indicate which requests from the host to respond to. Software can choose to enable only the IN, the OUT or both. After either a system reset or a soft reset all of the endpoints are disabled.

ENDPNTEN — Endpoint Enables Register — Addr 0x10024064

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EPFINEN	EPFOUTEN	EPEINEN	EPEOUTEN	EPDINEN	EPDOUTEN	EPCINEN	EPCOUTEN	EPBINEN	EPBOUTEN	EPAINEN	EPAOUTEN	EP9INEN	EP9OUTEN	EP8INEN	EP8OUTEN
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EP7INEN	EP7OUTEN	EP6INEN	EP6OUTEN	EP5INEN	EP5OUTEN	EP4INEN	EP4OUTEN	EP3INEN	EP3OUTEN	EP2INEN	EP2OUTEN	EP1INEN	EP1OUTEN	EP0INEN	EP0OUTEN
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-65. Endpoint Enables Register Description

Name	Description
EPnINEN All odd bits	**Endpoint<n>InEnabled**—When SET indicates that endpoint <n> is ready to respond to the host on the next IN token. If there is not any data to be sent, the Function Controller will respond with a NAK.
EPnOUTEN All even bits	**Endpoint<n>OutEnabled**—When SET indicates that endpoint <n> is ready to respond to the host on the next OUT token. If the Function Controller is unable to receive any data because the X and/or Y Buffers are full, it will respond with a NAK.

32.12.11 Endpoint Ready Set Register

ENDPNRDY — Endpoint Ready Set Register — Addr 0x10024068

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EPFINRDY	EPFOUTRDY	EPEINRDY	EPEOUTRDY	EPDINRDY	EPDOUTRDY	EPCINRDY	EPCOUTRDY	EPBINRDY	EPBOUTRDY	EPAINRDY	EPAOUTRDY	EP9INRDY	EP9OUTRDY	EP8INRDY	EP8OUTRDY
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EP7INRDY	EP7OUTRDY	EP6INRDY	EP6OUTRDY	EP5INRDY	EP5OUTRDY	EP4INRDY	EP4OUTRDY	EP3INRDY	EP3OUTRDY	EP2INRDY	EP2OUTRDY	EP1INRDY	EP1OUTRDY	EP0INRDY	EP0OUTRDY
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-66. Endpoint Ready Set Register

Name	Description
EPnINREADY All odd bits	**Endpoint<n>In Ready**—This register is directly written. Writing a 1 to it will set it and a 0 will not affect it. In order to clear this register, a 1 must be written to the corresponding bit in the Frame Number Register. Software must write to this bit to tell the Core whether or not the Endpoint is ready to be transferred. If this bit is 0, and the EP is enabled, then the function controller will return NAKs. If this bit is 1, then the transfer will commence. This bit is ignored if the EP is not enabled.
EPnOUTREADY All even bits	**Endpoint<n>Out Read**y—This register is directly written. Writing a 1 to it will set it and a 0 will not affect it. In order to clear this register, a 1 must be written to the corresponding bit in the Frame Number Register. Software must write to this bit to tell the Core whether or not the Endpoint is ready to be transferred. If this bit is 0, and the EP is enabled, then the function controller will return NAKs. If this bit is 1, then the transfer will commence. This bit is ignored if the EP is not enabled.

32.12.12 Immediate Interrupt Register

In order to reduce the number of interrupts occurring in a frame, all **Endpoint<n>InDoneStatus and Endpoint<n>OutDoneStatus** are flagged on the SOF. However, if software wishes to see the done status flagged immediately it can by setting the correct bit in this register.

IMEDINT — Immediate Interrupt Register — Addr 0x1002406C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IMFININT	IMFOUTINT	IMEININT	IMEOUTINT	IMDININT	IMDOUTINT	IMCININT	IMCOUTINT	IMBININT	IMBOUTINT	IMAININT	IMAOUTINT	IM9ININT	IM9OUTINT	IM8ININT	IM8OUTINT
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IM7ININT	IM7OUTINT	IM6ININT	IM6OUTINT	IM5ININT	IM5OUTINT	IM4ININT	IM4OUTINT	IM3ININT	IM3OUTINT	IM2ININT	IM2OUTINT	IM1ININT	IM1OUTINT	IM0ININT	IM0OUTINT
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-67. Immediate Interrupt Register Description

Name	Description
IMnININT All odd bits	**Immediate<n>InInterrupt**—When SET the Function controller will assert an interrupt immediately instead of waiting until the SOF.
IMnOUTINT All even bits	**Immediate<n>OutInterrupt**—When SET the Function controller will assert an interrupt immediately instead of waiting until the SOF.

32.12.13 Endpoint Done Status Register

This register indicates which endpoint transfers have been completed and are waiting for further software instructions. The EndpointDoneStatus Register is updated immediately, however it will not assert an interrupt until the SOF marker occurs, unless the corresponding ImmediateInterrupt bit is SET. Software should read this register to determine which endpoints are in need of service. In the case of a host OUT transfer overflow condition the endpoint is placed on the Endpoint Done Status Register.

EPNTDONESTAT — Endpoint Done Status Register — Addr 0x10024070

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EPFINDONE	EPFOUTDONE	EPEINDONE	EPEOUTDONE	EPDINDONE	EPDOUTDONE	EPCINDONE	EPCOUTDONE	EPBINDONE	EPBOUTDONE	EPAINDONE	EPAOUTDONE	EP9INDONE	EP9OUTDONE	EP8INDONE	EP8OUTINT
TYPE	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EP7INDONE	EP7OUTDONE	EP6INDONE	EP6OUTDONE	EP5INDONE	EP5OUTDONE	EP4INDONE	EP4OUTDONE	EP3INDONE	EP3OUTDONE	EP2INDONE	EP2OUTDONE	EP1INDONE	EP1OUTDONE	EP0INDONE	EP0OUTINT
TYPE	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-68. Endpoint Done Status Register Description

Name	Description
EPnINDONE All odd bits	**Endpoint<n>InDoneStatus**—When asserted indicates that endpoint <n> IN has transferred all of the data currently set in the Function Endpoint Descriptor.
EPnOUTDONE All even bits	**Endpoint<n>OutDoneStatus**—When asserted indicates that endpoint <n> OUT has received all of the data currently set in the Function Endpoint Descriptor or the endpoint has received larger than the *MaxPacketSize* for this endpoint.

32.12.14 Endpoint Done Enable Register

This register indicates which bit in the EndpointDoneStatus Register will generate an interrupt when set. This applies to interrupts generated either immediately or on the SOF marker.

EPNTDONEEN — Endpoint Done Enable Register — Addr 0x10024074

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EPFINDNEN	EPFOUTDNEN	EPEINDNEN	EPEOUTDNEN	EPDINDNEN	EPDOUTDNEN	EPCINDNEN	EPCOUTDNEN	EPBINDNEN	EPBOUTDNEN	EPAINDNEN	EPAOUTDNEN	EP9INDNEN	EP9OUTDNEN	EP8INDNEN	IM8OUTINT
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EP7INDNEN	EP7OUTDNEN	EP6INDNEN	EP6OUTDNEN	EP5INDNEN	EP5OUTDNEN	EP4INDNEN	EP4OUTDNEN	EP3INDNEN	EP3OUTDNEN	EP2INDNEN	EP2OUTDNEN	EP1INDNEN	EP1OUTDNEN	EP0INDNEN	EP0OUTINT
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-69. Endpoint Done Enable Register Description

Name	Description
EPnINDNEN Bits Odd	**Endpoint<n>InDoneEnable**—When SET allows endpoint <n> IN direction to generate done interrupts. The *Endpoint<n>InDoneEnable* must be SET for both immediate and SOF marker interrupts.
EPnOUTDNEN Bits Even	**Endpoint<n>OutDoneEnable**—When SET allows endpoint <n> IN direction to generate done interrupts. The *Endpoint<n>OutDoneEnable* must be SET for both immediate and SOF marker interrupts.

32.12.15 Endpoint Toggle Bits Register

On all transfers the PID of data packets alternate between DATA0 and DATA1 to ensure the ordering of the packets. The data toggle must continue across multiple transfers and are different for each physical endpoint. Each control transfer must begin from the host with DATA0.

EPNTTOGBITS — Endpoint Toggle Bits Register — Addr 0x10024078

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EPFINTOG	EPFOUTTOG	EPEINTOG	EPEOUTTOG	EPDINTOG	EPDOUTTOG	EPCINTOG	EPCOUTTOG	EPBINTOG	EPBOUTTOG	EPAINTOG	EPAOUTTOG	EP9INTOG	EP9OUTTOG	EP8INTOG	IM8OUTOG
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EP7INTOG	EP7OUTTOG	EP6INTOG	EP6OUTTOG	EP5INTOG	EP5OUTTOG	EP4INTOG	EP4OUTTOG	EP3INTOG	EP3OUTTOG	EP2INTOG	EP2OUTTOG	EP1INTOG	EP1OUTTOG	EP0INTOG	EP0OUTOG
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-70. Endpoint Toggle Bits Register Description

Name	Description
EPnINTOG Bits Odd	**Endpoint<n>InToggleBit**—This register is a toggle register. Writing a 1 will toggle the value and writing a 0 will leave it unchanged. The expected value of the current data PID is determined by the hardware from the last PID from the host for that endpoint. Software should change this bit while the EP is currently full or disabled if they wish to change the next data toggle bit.
EPnOUTTOG Bits Even	**Endpoint<n>OutToggleBit**—This register is a toggle register. Writing a 1 will toggle the value and writing a 0 will leave it unchanged. The value of the next data PID to be sent by hardware is determined by this bit. This bit is updated by Hardware after each successful transmission. Software should change this bit while the EP is currently empty or disabled if they wish to change the next data toggle bit.

32.12.16 Frame Number and Endpoint Ready Clear Register

Reading this returns the frame number of the last successfully received SOF. The frame number is returned with the least significant byte (LSB) first. Writing this register will clear the corresponding Endpoint Ready bit.

FNEPRDYCLR — Frame Number and Endpoint Ready Clear Register — Addr 0x1002407C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
	EPRDYCLEAR (write)															
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						FRMNUMB (read)										
TYPE	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
	EPRDYCLEAR (write)															
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-71. Frame Number and Endpoint Ready Clear Register Description

Name	Description
Reserved Bits 31–11	Reserved
FRMNUMB Bits 10–0 (Read)	**Last Frame Number**—Hardware will set this value based upon the *FrameNumber* contained in the last received SOF packet.
EPRDYCLEAR Bits 31–0 (Write)	**Endpoint Ready Clear**—Writing a 1 to this register will clear the corresponding Endpoint Ready bit but will not affect the Frame Number. This is a workaround that had to be implemented due to a register contention condition that exists in the Core. Since address space was not available, the Frame Number register was multiplexed with this functionality.

32.13 AHB/DMA IP Core

The OTG-IP CORE supports an AHB interface as well as a general bus interface. This interface goes on top of the original bus interface and conforms to the AHB protocol as defined in the AMBA specification. The AHB/DMA IP Core contains both an AHB Master and Slave, a DMA Controller, and Core, ETD, EP, and data memory interfaces that connect directly to the OTG-IP CORE.

32.14 AHB HCLK Variation

The AHB/DMA IP core allows the slave AHB and the master AHB clocks to be asynchronous to each other. As in some low power applications, it is possible that the AHB clock will not be a fixed frequency clock, but instead will vary in frequency. Moreover, the clock may drop completely while the system is in the sleep mode. This situation can occur in designs that have different levels of power saving modes. However, since many signals in the Core interface are dependent on timing, the results would be unpredictable if the interface clock were to stop or slow drastically. For this reason, we introduced an optional partition between the fixed frequency interface clock (mClk) and the variable frequency interface clock (HCLK).

32.14.1 DMA Theory of Operation

The ARM9 core contains a DMA controller integrated with an AHB master. Instead of using the normal DMA Request and Acknowledge handshaking which requires an external DMA handler, the USB OTG module DMA controller emulates an AHB master.

Software sets the address in system memory where the data should be retrieved or sent and the number of bytes that will be transferred. Once the DMA is enabled, the DMA acts as an AHB master and will request to the arbiter for access to the bus. From there, the master will automatically send and retrieve files to/from the system memory using standard AHB protocol. This method removes the need for logic external to the core to handle the DMA handshaking normally associated with DMA transfers.

32.14.2 DMA Registers

The default location of the DMA registers is the address space: 0x0800–0x09FF. A table summary of the registers used is shown in Table 32-72.

Table 32-72. USB OTG DMA Register Summary

Address	Register Name	Description
0x1002 4800	DmaRevision Register	Revision code for DMA core.
0x1002 4804	DmaInterruptStatus Register	EP and ETD interrupt for erroneous transfers.
0x1002 4808	DmaInterruptEnable Register	Enable or mask DMA Interrupts
0x1002 480C	EtdDmaErrorStatus Register	Identifies which ETD client DMA aborted on error.
0x1002 4810	EpDmaErrorStatus Register	Identifies which EP client DMA aborted on error.
0x1002 4814	DmaInterruptEnableClr Register	Interrupt Enable Clear to mask DMA Interrupts
0x1002 4818–481C	Reserved	—

Table 32-72. USB OTG DMA Register Summary (continued)

Address	Register Name	Description
0x1002 4820	EtdDmaEnable Register	Enables ETD for DMA transfer.
0x1002 4824	EpDmaEnable Register	Enables EP for DMA transfer.
0x1002 4828	EtdDmaEnableXTriggerRequest Register	Enables ETD for DMA and causes an XTrigger request.
0x1002 482c	EpDmaEnableXTriggerRequest Register	Enables EP for DMA and causes an XTrigger request.
0x1002 4830	EtdDmaEnableXYTriggerRequest Register	Enables ETD for DMA and causes an XTrigger and a YTrigger request.
0x1002 4834	EpDmaEnableXYTriggerRequest Register	Enables EP for DMA and causes an XTrigger and a Ytrigger request.
0x1002 4838	EtdDmaBurst4Enable Register	Enable Burst4 mode for ETD transfers (transfers data 4 Dwords at a time)
0x1002 483C	EpDmaBurst4Enable Register	Enable Burst4 mode for EP transfers (transfers data 4 Dwords at a time)
0x1002 4840	MiscControl Register	Miscellaneous control bits
0x1002 4844	Reserved	—
0x1002 4848	EtdDmaChannelClear Register	Clear ETD enabled DMA channels
0x1002 484C	EpDmaChannelClear Register	Clear EP enabled DMA channels
0x1002 4850–48FC	Reserved	—
0x1002 4900–497C	ETD 0–31: Etd<n>SystemMemoryStartAddress Registers	Starting system memory address location where data will be put/fetched for ETD 0–31.
0x1002 4980 + (8*N) where N=EP Number	EP 0–15 OUT: Ep<n>SystemMemoryStartAddress Registers	Starting system memory address location where data will be put/fetched for EP 0–15 OUT
0x1002 4984 + (8*N) where N=EP Number	EP 0–15 IN: Ep<n>SystemMemoryStartAddress Registers	Starting system memory address location where data will be put/fetched for EP 0–15 IN
0x1002 4A00–4a7c	ETD 0–31: Etd<n>DmaBufferXferPtr Registers	Stores index of current buffer for ETD 0–31
0x1002 4A80 + (8*N)	EP 0–15 OUT: Ep<n>DmaBufferXferPtr Registers	Stores index of current buffer for EP 0–15 OUT
0x1002 4A84 + (8*N)	EP 0–15: IN: Ep<n>DmaBufferXferPtr Registers	Stores index of current buffer for EP 0–15 IN
0x1002 4B00–4FFF	Reserved	—

32.14.3 DMA Revision Register

The DmaRevision Register contains the revision code for the block.

DMAREV — DMA Revision Register — Addr 0x10024800

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									REVCODE							
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	0x0010															

Table 32-73. DMA Revision Register Description

Name	Description
Reserved Bits 31–8	Reserved
REVCODE Bits 7–0	**RevisionCode**—Revision code for DMAIP block design.

32.14.4 DMA Interrupt Status Register

The DMA Interrupt Status reports the interrupt status for the DMAIP.

DMAINTSTAT — DMA Interrupt Status Register — Addr 0x10024804

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															EPERR	ETDERR
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-74. DMA Interrupt Status Register Description

Name	Description
Reserved Bits 31–2	Reserved
EPERR Bit 1	**Endpoint Error Interrupt**—EP DMA transfer aborted on error (read 0x0810 to find out which EP client). The error status is cleared by hardware.
ETDERR Bit 0	**ETD Error Interrupt**—ETD DMA transfer aborted on error (read 0x080c to find out which ETD client). The error status is cleared by hardware.

32.14.5 DMA Interrupt Enable Register

The DMA Interrupt Enable Register is used to enable the corresponding status bits in rIntStat to assert the DMAIP Interrupt.

DMAINTEN — DMA Interrupt Enable Register — Addr 0x10024808

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															EPERRINTEN	ETDERRINTEN
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-75. DMA Interrupt Enable Register Description

Name	Description	Setting
Reserved Bits 31–2	Reserved	
EPERRINTEN Bit 1	**Endpoint Error Interrupt Enable**—	0 = Mask interrupt 1 = Enable interrupt
ETDERRINTEN Bit 0	**ETD Error Interrupt Enable**—	0 = Mask interrupt 1 = Enable interrupt

32.14.6 ETD DMA Error Status Register

The ETD DMA Error Status register identifies which ETD client DMA transfer was aborted on an Error Response.

ETDDMAERSTAT — ETD DMA Error Status Register — Addr 0x1002480C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ETDDMAERST[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETDDMAERST[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-76. ETD DMA Error Status Register Description

Name	Description
ETDDMAERST Bits 31–0	**ETD DMA Error Status**—Indicates the Error status from each client. Writing 1 clears the corresponding status bit.

32.14.7 EP DMA Error Status Register

The EpDmaErrorStatus Register identifies which EP client DMA transfer was aborted on an Error Response.

EPDMAERSTAT — EP DMA Error Status Register — Addr 0x10024810

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EPDMAERST[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EPDMAERST[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-77. Endpoint DMA Error Status Register Description

Name	Description
EPDMAERRST Bits 31–0	**Endpoint DMA Error Status**—Indicates the Error status from each client. Writing 1 clears the corresponding status bit.

32.14.8 ETD DMA Enable Register

The ETD DMA Enable Register enables the DMA for the corresponding ETD client.

ETDDMAEN — ETD DMA Enable Register — Addr 0x10024820

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ETDDMAENB[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETDDMAENB[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-78. ETD DMA Enable Register Description

Name	Description
ETDDMAENB Bits 31–0	**ETD DMA Enable**—Setting the ETDDmaEnable bit enables DMA transfers on the corresponding ETD. The transfer is auto-disabled on the completion of the transfer. Software can clear this bit using EtdDmaChannelClear Register.

32.14.9 EP DMA Enable Register

The EP DMA Enable Register enables the DMA for the corresponding EP client.

EPDMAEN — EP DMA Enable Register — Addr 0x10024824

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EPDMAENB[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EPDMAENB[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-79. EP DMA Enable Register Description

Name	Description
EPDMAENB Bits 31–0	**Endpoint DMA Enable**—Setting the EndpointDMAEnble bit will enable DMA transfer on the corresponding EP. Will be auto-disabled at the end of the transfer. Software can clear this bit by using EpDmaChannelClear Register.

32.14.10 ETD DMA Enable X Trigger Request Register

Writing to the EtdDmaEnableXTriggerRequest Register results in two things: it sets the ETD DMA enable bit AND causes an XTrigger Request from the ETD client.

ETDDMAXTEN — ETD DMA Enable X Trigger Request Register — Addr 0x10024828

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ETDDMAXTEN[31:0]															
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETDDMAXTEN[31:0]															
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-80. ETD DMA XTrigger Enable Register Description

Name	Description
ETDDMAXTEN Bits 31–0	**ETD DMA XTrigger Enable**—Writing a 1 sets the corresponding ETDDMAEnable bit and causes an external XTrigger for the client.

32.14.11 EP DMA Enable X Trigger Request Register

Writing to the EpDmaEnableXTriggerRequest Register address results in two things: it sets the EP DMA enable bit AND fakes an XTrigger Request from the EP client.

EPDMAXTEN — EP DMA Enable X Trigger Request Register — Addr 0x1002482C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EPDMAXTEN[31:0]															
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EPDMAXTEN[31:0]															
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-81. EP DMA Enable X Trigger Request Register Description

Name	Description
EPDMAXTEN Bits 31–0	**Endpoint DMA X Trigger Enable**—Writing a 1 sets the corresponding Endpoint DMA Enable bit and causes an external X Trigger for the client.

32.14.12 ETD DMA Enable XY Trigger Request Register

The EtdDmaEnbXYTrigRequest Register functionality is the same as the EtdDmaEnbXTrigRequest Register, except that it causes both XTrigger and YTrigger requests from the ETD client.

ETDDMAENXYT — ETD DMA Enable XY Trigger Request Register — Addr 0x10024830

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ETDDMAENXYT[31:0]															
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETDDMAENXYT[31:0]															
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-82. ETD DMA Enable XY Trigger Request Register Description

Name	Description
ETDDMAENXYT Bits 31–0	**ETD DMA X And Y Trigger Enable**—Writing 1 sets the corresponding ETDDMAEnable and causes both X Trigger and Y Trigger for the client

32.14.13 EP DMA Enable XY Trigger Request Register

The EpDmaEnableXYTrigerRequest Register functionality is the same as the EpDmaEnableXTrigerRequest Register, except that it causes both XTrigger and YTrigger Requests from the EP client.

EPDMAENXYT — EP DMA Enable XY Trigger Request Register — Addr 0x10024834

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EPDMAENXYT[31:0]															
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EPDMAENXYT[31:0]															
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-83. EP DMA Enable XY Trigger Request Register Description

Name	Description
EPDMAENXYT Bits 31–0	**Endpoint DMA X and Y Trigger Enable**—Writing a 1 sets the corresponding EndpointDMAEnable bit and causes both X Trigger and Y Trigger for the client

32.14.14 ETD DMA Burst4 Enable Register

The EtdDmaBurst4Enable Register is set by default for all clients. The DMA will transfer words on the AHB using the INCR4 transfer size (burst of 4 DWORDS). If for some reason a source/destination of the DMA transfer cannot handle bursts of 4 DWORDS, the corresponding bit should be cleared to transfer one word at a time.

ETCDMABST4EN — ETD DMA Burst4 Enable Register — Addr 0x10024838

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ETDDMABST4EN[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETDDMABST4EN[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0x0000															

Table 32-84. ETD DMA Burst4 Enable Register Description

Name	Description
ETDDMABST4EN Bits 31–0	**ETD DMA Burst4 Enable**—Writing a 1 sets the corresponding ETD DMA Enable and causes both X Trigger and Y Trigger for the client.

32.14.15 EP DMA Burst4 Enable Register

The EP DMA Burst4 Enable Register is set by default for all clients. The DMA will transfer words on the AHB using the INCR4 transfer size (burst of 4 DWORDS). If for some reason a source/destination of the DMA transfer cannot handle bursts of 4 DWORDS, the corresponding bit should be cleared to transfer one word at a time.

EPDMABST4EN — EP DMA Burst4 Enable Register — Addr 0x1002483C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EPDMABST4EN[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EPDMABST4EN[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	0x0000															

Table 32-85. EP DMA Burst4 Enable Register Description

Name	Description
EPDMABST4EN Bits 31–0	**EP DMA Burst4 Enable**—Writing a 1 sets the corresponding EP DMA Enable and causes both X Trigger and Y Trigger for the client.

32.14.16 Misc Control Register

The MiscControl register enables certain modes that may make development easier. Then enabling of the modes is application dependent.

MISCCONTROL — Misc Control Register — Addr 0x10024840

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													ISOPREVFRM	SKPRTRY	ARBMODE	FILTCC
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-86. Misc Register Description

Name	Description
Reserved Bits 31–4	Reserved
IsoPrevFrm Bit 3	**ISO OUT Previous Frame Mode**—If set, the DMA will only service an ISO OUT ETD one frame before the transfer will start. The frame number where the transfer will occur can be found in the StartingFrame field of the Host Isochronous ETD descriptor format.
SkpRtry Bit 2	**Skip On Retry Mode**—If set, will skip current DMA transfer if an AHB RETRY response is encountered, and will start next pending transfer. The transfer will be retried after the rest of the pending transfers have been completed.
ArbMode Bit 1	**Arbiter Mode Select:** 1: Puts the DMA arbiter in Round Robin mode. The arbitration for the DMA servicing will proceed in a round robin fashion. 0: Puts the DMA arbiter in Priority Access mode. The DMA enabled transfers are serviced in a prioritized fashion in order of decreasing ETD/EP number (Zero gets highest priority).
FiltCC Bit 0	**Filter On Completion Code**—If set, the DMA controller will check the completion code of the transfer. The data will only be transferred if the completion code shows no error.

32.14.17 ETD DMA Channel Clear Register

This register will clear the DMA channel corresponding to whichever bit is set and also disable the DMA transfer. It can be used to abort a DMA transfer.

ETDDMACHANLCLR — ETD DMA Channel Clear Register — Addr 0x10024848

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ETDDMACHANLCLR[31:0]															
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETDDMACHANLCLR[31:0]															
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-87. ETD DMA Channel Clear Register Description

Name	Description
ETDDMACHANLCLR Bits 31–0	**ETD DMA Channel Clear**—Will clear the DMA channel corresponding to the bit that is set. Will also clear the DMA enable bit for the ETD. This register is a Write-Clear register. Software will always read 0 on this register.

32.14.18 EP DMA Channel Clear Register

This register will clear the DMA channel corresponding to whichever bit is set and also disable the DMA transfer. It can be used to abort a DMA transfer.

EPDMACHANLCLR — Endpoint DMA Channel Clear Register — Addr 0x1002484C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ETDDMACHANLCLR[31:0]															
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETDDMACHANLCLR[31:0]															
TYPE	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-88. ETD DMA Channel Clear Register Description

Name	Description
EPDMACHANLCLR Bits 31–0	**Endpoint DMA Channel Clear**—Will clear the DMA channel corresponding to the bit that is set. Will also clear the DMA enable bit for the EP. This register is a Write-Clear register. Software will always read 0 on this register.

32.14.19 ETD<n>System Memory Start Address Register

Before Enabling DMA for a client ETD, the corresponding register must be loaded with the ETD System Memory Start Address where the data will be stored. The ETD System Memory Start Address are byte addresses that must be DWORD aligned.

ETDSMSA — ETD<n>System Memory Start Address Register — Addr 0x10024900 to 0x1002497C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ETDSMSA[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETDSMSA[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-89. ETD<n>System Memory Start Address Register Description

Name	Description
ETDSMSA Bits 31–0	**ETD System Memory Start Address**—Starting address in system memory where DMA will put/fetch data to for ETD<n>.

32.14.20 EP<n>System Memory Start Address Register

Before enabling the DMA for a client EP, the corresponding register must be loaded with the system memory address where the data will be stored. The EndpointSystemMemoryStartAddress are byte addresses that must be DWORD aligned.

EPSMSA — EP<n>System Memory Start Address Register — Addr 0x10024980 to 0x100249FC

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EPSMSA[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EPSMSA[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-90. EP<n>System Memory Start Address Register Description

Name	Description
EPSMSA Bits 31–0	**Endpoint System Memory Start Address**—Starting address in system memory where DMA will put/fetch data to for EP<n>.

32.14.21 ETD<n>DMA Buffer Xfer Ptr Registers

If the DMA is paused due to the encounter of an AHB RETRY response and the SKIPONRETRY feature is enabled, the current index into the buffer is stored in these registers so that the DMA knows where to re-start the transfer. The buffer transfer pointers are accessible only for debug purposes.

ETDDMABUFPTR — ETD<n>DMA Buffer Xfer Ptr Registers — Addr 0x10024A00 to 0x10024A7C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ETDDMABUFPTR[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETDDMABUFPTR[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-91. ETD<n>DMA Buffer Xfer Ptr Register Description

Name	Description
EPSMSA Bits 31–0	**ETD DMA Buffer Transfer Pointer**—Stores the current index of the buffer when AHB RETRY response is asserted.

32.14.22 EP<n>DMA Buffer Xfer Ptr Registers

If the DMA is paused due to the encounter of an AHB RETRY response and the SKIPONRETRY feature is enabled, the current index into the buffer is stored in these registers so that the DMA knows where to re-start the transfer. The buffer transfer pointers are accessible only for debug purposes.

EPDMABUFPTR — EP<n>DMA Buffer Xfer Ptr Registers — Addr 0x10024A80 to 0x10024AFC

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EPDMABUFPTR[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EPDMABUFPTR[31:0]															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 32-92. EP<n>DMA Buffer Xfer Ptr Register Description

Name	Description
EPDMABUFPTR Bits 31–0	**EP DMA Buffer Transfer Pointer**—Stores the current index of the buffer when AHB RETRY response is asserted.

32.15 OTG-I^{2}C Transceiver Controller

The I^{2}C OTG Transceiver Controller is designed to support external OTG transceivers via the I^{2}C interface. The OTG transceiver controller consists of a set of control registers with associated logic. The OTG transceiver controller registers are fully accessible from software. The registers in the OTG transceiver are mapped directly to the controller so that software does not have to write I^{2}C routines in order to access them.

NOTE

The OTG-I2C transceiver controller is designed to support the addresses found in Philips ISP1301 transceiver. If using a different OTG transceiver, it is recommended to verify the registers between the OTG-I2C transceiver controller and OTG transceiver match. If there are any differences found, it is highly recommended to use the i.MX21 I2C module to communicate to the OTG transceiver to ensure proper transceiver programming.

32.15.1 Interrupt Handling

The USB OTG module contains an I^2C Interrupt Register with 4 lower bits to register the interrupt occurrences. The upper four bits are the interrupt enable bits. The interrupt register records three types of interrupts, and each of these three interrupts is associated with three major functions of the external OTG transceiver operations. The OtgXcvrINT indicates the status changes in the external OTG transceiver operations conditions such as VBus status, ID, and A and B device connections. The R/W_ready indicates the readiness of the I^2C register read/write operations. This bit remains cleared, while the USB OTG module is communicating with the external transceiver via the I^2C interface. The I2C_NoACK bit represents the error condition on the I^2C interface. When the USB OTG module receives a NACK while it is expecting a ACK response, it sets this bit to alert the system indicating a failure occurrence of the current I^2C operation.

The USB OTG module reacts differently to each of these interrupt conditions:

- OtgXcvrINT interrupt
  - This interrupt causes the USB OTG module to abandon the current and pending I^2C operations, and automatically start a read operation from the Interrupt Source Register in the external OTG transceiver. The read data is copied into the Interrupt Source Register (shadow register) in the I^2C interface control block in the USB OTG module, and thus in turn reports the USB line status.
- R/W_ready interrupt
  - The USB OTG module does not take any action when the R/W_ready interrupt occurs. This interrupt is merely indicating that the USB OTG module is free to do the next I^2C operation.
- I2c_NoACK interrupt
  - When the I2c_NoACK interrupt occurs, the USB OTG module aborts the current operation, purges all the pending operations, and stores the failed external OTG transceiver register address into the ‘Sequential Read Start Address’ register. The USB OTG module expects the software to read the ‘Sequential Read Start Address’, and clear the I2c_NoACK interrupt bit before it initiates another operation.

32.15.2 Software Control Mode

In normal software control mode, the Core allows the Software driver to set (or clear) any register bits except the Interrupt Source register bits. The Core monitors USB bus activities such as ‘connect’, ‘disconnect’, ‘resume’, ‘suspend’, and ‘reset’. All registers are functional and when interrupt events occur, the Core passes the interrupt information from the OTG transceiver to the MCU via the interrupt pin.

32.15.3 Hardware Control Mode

When the USB OTG module is configured to operate in hardware mode, the I^2C controller generates all the appropriate control signals based on the OTG transceiver interface pin states and the contents of the control and interrupt registers. If Control Register 1 is reconfigured while the USB bus is active, the I^2C controller does not update the control registers in the OTG transceivers until the USB bus becomes idle.

32.15.4 Accessing the OTG Transceiver

Accessing the OTG Transceiver through the Transceiver Controller is easier on software than going through a dedicated I^2C driver however, there are required order of operations to access the transceiver. This order must be followed to ensure that the data is sent to the OTG Transceiver correctly.

The USB OTG module contains a set of I^2C control and status registers that are identical to the registers in the external OTG transceiver. The Transceiver Controller's interrupt registers behave as shadow (duplicate) registers of the external OTG transceiver interrupt registers. The contents of the control registers in the external OTG transceiver are also mapped to the Transceiver Controller. All of these registers are directly accessible by the MCU Interface of the Core and, when written to, will automatically send the updated values to the external OTG transceivers through the I^2C interface.

32.15.5 Product and Vendor ID Register

Product and Vendor ID Register

	31–24				23–16			
	31 30 29 28 27 26 25 24				23 22 21 20 19 18 17 16			
Address	0x103				0x102			
Name	PRODUCT ID							
Reset	0001 0011				0000 0001			

	15–8				7–0			
	15 14 13 12 11 10 9 8				7 6 5 4 3 2 1 0			
Address	0x101				0x100			
Name	VENDOR ID							
Reset	0000 0100				1100 1100			

Table 32-93. Product and Vendor ID Register Description

Name	Description
PRODUCT ID Bits 31–16	Product ID supplied by the USB-IF to each xcvr mfr.
VENDOR ID Bits 15–0	Vendor ID supplied by the USB-IF to each xcvr mfr.

32.15.6 OTG Transceiver Control Register I²C-OTG Transceiver Controller Registers

The USB OTG module contains a set of I^2C control and status registers that are identical to the registers in the external OTG transceiver. The Transceiver Controller's interrupt registers behave as shadow (duplicate) registers of the external OTG transceiver interrupt registers. The contents of the control registers in the external OTG transceiver are also mapped to the Transceiver Controller. All of these registers are directly accessible by the MCU Interface of the USB OTG module and, when written to, will automatically send the updated values to the external OTG transceivers through the I^2C interface.

All the I^2C control registers are directly readable and/or writable with access timing similar to the other registers in the OTGIP USB OTG module from a software point of view. Please note that internally, the access is not similar due to the latency of accessing the I^2C interface on the transceiver. Depending on when the I^2C status register read access is initiated, the read status may not be exactly the same as the current status of the OTG transceiver. This is due to the inherent cycle delays associated with moving the I^2C register contents to and from the registers in the external OTG transceiver via I^2C interface.

The I^2C control logic in the USB OTG module immediately triggers one or more write operations to the OTG transceiver whenever the contents of the control register in the USB OTG module are updated by either hardware or by the Software. Since each of the registers in the OTG Transceiver is a byte wide register, we cannot make 32-bit accesses to them. Instead, you must address each register directly on a byte-by-byte basis. For this reason, the individual byte addresses have been included in the register tables and descriptions. It takes a minimum of 20 SCL (I^2C clock) cycles before the OTG transceiver recognizes this new control information.

NOTE

All the OTG-I^2C Registers are BYTE accessible and support Little Endian only.

Each time you access a register, the OTG Transceiver Controller decodes the access and routes the access request command to the OTG Transceiver. While this occurs, the busy bit in the I^2C Operations Register will be asserted. When it is de-asserted, that means the operation was completed successfully.

32.15.7 OTG Transceiver Control Register

OTG Transceiver Control Register

	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
Address	0x107								0x106								0x105								0x104							
Name	OTG Control (Clear)								OTG Control (Set)								Mode Control (Clear)								Mode Control (Set)							
	VBUSCHGCLR	VBUSDISCLR	VBUSDRVCLR	IDGNDCLR	DMPULLDNCLR	DPPULLDNCLR	DMPULLUPCLR	DPPULLUPCLR	VBUSCHGSET	VBUSDISSET	VBUSDRVSET	IDGNDSET	DMPULLDNSET	DPPULLDNSET	DMPULLUPSET	DPPULLUPSET	RESERVED	OEINTCLR	BDISACONCLR	TRANSPCLR	DATSE0CLR	SSPNDCLR	SPEEDCLR	RESERVED	OEINTSET	BDISACONSET	TRANSPSET	DATSE0SET	SSPNDSET	SPEEDSET		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0		0	0	0	0	0	0		0	0	0	0	0	0		

Table 32-94. OTG Transceiver Control Register Description

Name	SW	HW	Description
VBUSCHGCLR Bit 31	RC	R	Disable Charging Vbus through a resistor
VBUSDISCLR Bit 30	RC	R	Disable Discharging Vbus through a resistor to ground
VBUSDRVCLR Bit 29	RC	R	Disable power to VBUS

Table 32-94. OTG Transceiver Control Register Description (continued)

Name	SW	HW	Description
IDGNDCLR Bit 28	RC	R	Disable Connecting ID pin to ground
DMPULLDNCLR Bit 27	RC	R	Disable D- pull down
DPPULLDNCLR Bit 26	RC	R	Disable D+ pull down
DMPULLUPCLR Bit 25	RC	R	Disable D- pull up
DPPULLUPCLR Bit 24	RC	R	Disable D+ pull up
VBUSCHGSET Bit 23	RS	R	Charge Vbus through a resistor
VBUSDISSET Bit 22	RS	R	Discharge Vbus through a resistor to ground
VBUSDRVSET Bit 21	RS	R	Provide power to VBUS
IDGNDSET Bit 20	RS	R	Connect ID pin to ground
DMPULLDNSET Bit 19	RS	R	Connect D- pull down
DPPULLDNSET Bit 18	RS	R	Connect D+ pull down
DMPULLUPSET Bit 17	RS	R	Connect D- pull up
DPPULLUPSET Bit 16	RS	R	Connect D+ pull up
Reserved Bits 15–14			
OEINTCLR Bit 13	RC	R	Clears the OE_TP_INT/ control bit.
BDISACONCLR Bit 12	RC	R	Enable A-device to connect if B-device disconnect detected
TRANSPCLR Bit 11	RC	R	Clear Transparent mode.
DATSE0CLR Bit 10	RC	R	Clear DAT SE0 Mode.
SSPNDCLR Bit 9	RC	R	Clear suspend mode for transceiver.
SPEEDCLR Bit 8	RC	R	Clear rise and fall times for transmit driver.

Table 32-94. OTG Transceiver Control Register Description (continued)

Name	SW	HW	Description
Reserved Bits 7–6			
OEINTSET Bit 5	RS	R	When asserted, and when suspend bit set, then OE_TP_INT/ becomes an output, and is asserted low when an interrupt occurs
BDISACONSET Bit 4	RS	R	Enable A-device to connect if B-device disconnect detected
TRANSPSET Bit 3	RS	R	0 = direct I^2C mode 1 = transparent I^2C mode enabled
DATSE0SET Bit 2	RS	R	0 = VP_VM USB mode 1 = DAT_SE0 USB mode
SSPNDSET Bit 1	RS	R	Put transceiver in low power mode
SPEEDSET Bit 0	RS	R	Set rise and fall times of transmit driver.

32.15.8 Interrupt Source and Latch Register

	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
Address	0x10B								0x10A								0x109								0x108							
Name	Interrupt Latch (Clear)								Interrupt Latch (Set)								Reserved								Interrupt Source							
	BSESSENDCLR	BDISACONCLR	IDFLOATCLR	IDRESISTCLR	IDGNDCLR	DPSRPCLR	ASESSVLDCLR	VBUSVLDCLR	BSESSENDSET	BDISACONSET	IDFLOATSET	IDRESISTSET	IDGNDSET	DPSRPSET	ASESSVLDSET	VBUSVLDSET									BSESSEND	BDISACON	IDFLOAT	IDRESIST	IDGND	DPSRP	ASESSVLD	VBUSVLD
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0

Table 32-95. Interrupt Source and Latch Register Description

Name	SW	HW	Description
BSESSENDCLR Bit 31	RC	R	Clear BSessEnd Interrupt
BDISACONCLR Bit 30	RC	R	Clear BDisACon Interrupt
IDFLOATCLR Bit 29	RC	R	Clear IDFloat Interrupt

Table 32-95. Interrupt Source and Latch Register Description (continued)

Name	SW	HW	Description
IDRESISTCLR Bit 28	RC	R	Clear IDResist Interrupt
IDGNDCLR Bit 27	RC	R	Clear IDGnd Interrupt
DPSRPCLR Bit 26	RC	R	Clear D+SRP Interrupt
ASESSVLDCLR Bit 25	RC	R	Clear AsessVld Interrupt
VBUSVLDCLR Bit 24	RC	R	Clear VbusVld Interrupt
BSESSENDSET Bit 23	RS	R	Set BSessEnd Interrupt
BDISACONSET Bit 22	RS	R	Set BDisACon Interrupt
IDFLOATSET Bit 21	RS	R	Set IDFloat Interrupt
IDGNDSET Bit 20	RS	R	Set IDResist Interrupt
IDRESISTSET Bit 19	RS	R	Set IDGnd Interrupt
DPSRPSET Bit 18	RS	R	Set D+SRP Interrupt
ASESSVLDSET Bit 17	RS	R	Set AsessVld Interrupt
VBUSVLDSET Bit 16	RS	R	Set VbusVld Interrupt
Reserved Bits 15:8			Reserved
BSESSEND Bit 7	R	W	Session end comparator, threshold < 0.8V
BDISACON Bit 6	R	W	Set when bdis_acon_en is set, and transceiver asserts dp_pullup after detecting B-device disconnect
IDFLOAT Bit 5	R	W	ID pin floating
IDGND Bit 4	R	W	ID pin resistively connected to ground
IDRESIST Bit 3	R	W	ID pin ground
DPSRP Bit 2	R	W	D+ asserted during SRP

Table 32-95. Interrupt Source and Latch Register Description (continued)

Name	SW	HW	Description
ASESSVLD Bit 1	R	W	Session valid comparator, threshold = 0.8V – 2.0V
VBUSVLD Bit 0	R	W	A-device Vbus valid comparator, threshold > 4.4V

32.15.9 Interrupt Mask True and False Register

This register allows interrupts to be generated on either the assertion or de-assertion of a condition. In HNP, there are many times when it is desired to know when a session either becomes valid (at initiation of SRP) or when a session becomes invalid (when a session is ended). In this case, it may be desirable to set both the True and False mask for this condition so that it can be known when each case occurs.

Interrupt Mask True and False Register

	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
Address	0x10F								0x10E								0x10D								0x10C							
Name	Int Mask True (Clear)								Int Mask True (Set)								Int Mask False (Clear)								Int Mask False (Set)							
	BSESSENDTCLR	BDISACONTCLR	IDFLOATTCLR	IDRESISTTCLR	IDGNDTCLR	DPSRPTCLR	ASESSVLDTCLR	VBUSVLDTCLR	BSESSENDTSET	BDISACONTSET	IDFLOATTSET	IDRESISTTSET	IDGNDTSET	DPSRPTSET	ASESSVLDTSET	VBUSVLDTSET	BSESSENDFCLR	BDISACONFCLR	IDFLOATFCLR	IDRESISTFCLR	IDGNDFCLR	DPSRPFCLR	ASESSVLDFCLR	VBUSVLDFCLR	BSESSENDFSET	BDISACONFSET	IDFLOATFSET	IDRESISTFSET	IDGNDFSET	DPSRPFSET	ASESSVLDFSET	VBUSVLDFSET
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 32-96. Interrupt Mask True and False Register Description

Name	SW	HW	Description
Int Mask True (Clear) Bits 31–24	RC	R	The True Mask generates an interrupt when the condition is asserted or true. This clears the True Mask.
Int Mask True (Set) Bits 23–16	RS	R	The True Mask generates an interrupt when the condition is asserted or true. This sets the True Mask.
Int Mask False (Clear) Bits 15–8	RC	R	The False Mask generates an interrupt when the condition is de-asserted or false. This clears the False Mask.
Int Mask False (Set) Bits 7–0	RS	R	The False Mask generates an interrupt when the condition is de-asserted or false. This clears the False Mask.

32.15.10 OTG Control Register

OTG Control Register

	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
Address	0x113								0x112								0x111								0x110							
Name	Interrupt Latch (Clear)								Interrupt Latch (Set)								Reserved								Reserved							
	Reserved			TRANSPDIRCLR		BIDIRCLR	SPDSUSPCLR	GLBPWRDWNCLR	Reserved			TRANSPDIRSET		BIDIRSET	SPDSUSPSET	GLBPWRDWNSET	Reserved								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																

Table 32-97. OTG Control Register

Name	SW	HW	Description
Reserved Bits 31–29			Reserved
TRANSPDIRCLR Bits 28–27	RC	R	Clears the Transparent Direction control bit.
BIDIRCLR Bit 26	RC	R	Clears the OE_TP_INT control bit.
SPDSUSPCLR Bit 25	RC	R	Clears the Speed/Suspend control bit.
GLBPWRDWNCLR Bit 24	RC	R	Clears the Global Power Down control bit.
Reserved Bits 23–21			Reserved
TRANSPDIRSET Bits 20–19	RS	R	Controls the direction of data transfer in the transparent mode· **00**: DAT_VP ≥ DP, SE0_VM ≥ DM **01**: DAT_VP ≥ DP, SE0_VM ≤ DM **10**: DAT_VP ≤ DP, SE0_VM ≥ DM **11**: DAT_VP ≤ DP, SE0_VM ≤ DM
BIDIRSET Bit 18	RS	R	Enables OE_TP_INT\ control **0**: DAT_VP and SE0_VM transmit only **1**: OE_TP_INT\ controls the direction of DAT_VP and SE0_VM
SPDSUSPSET Bit 17	RS	R	Controls the speed and suspend in USB modes **0**: control with Speed and Suspend pins **1**: control with the Mode register [1:0]

Table 32-97. OTG Control Register (continued)

Name	SW	HW	Description
GLBPWRDWNSET Bit 16	RS	R	Set to power down the chip. **Note:** Activities on the I²C or OTG transceiver wakes up the OTG transceiver
Reserved Bits 15–0			

32.15.11 Device Address and I²C Operations Register

	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
Address	0x11B								0x11A								0x119								0x118							
Name	I²C Operation Ctrl								Seq Rd Start Addr								Sequential Ops								Device Address							
	I2CBUSY	Reserved						HWSWMODE	I2COE	SEQREADSTRT							NUMSEQOPS							RD/WRN	OTGDEVADDR						
Reset	0	0000						0	0	0000 1000							0000 0000							1	010 1100						

Table 32-98. Device Address and I²C Operations Register Description

Name	SW	HW	Description
I²C BUSY Bit 31	R	W	This bit is set while the Core is transmitting /receiving data from the external OTG Transceiver.
Reserved Bits 30–26			Reserved
HWSWMODE Bit 25	RW	R	OTG_TDI bypass mode. This bit allows the CPU to control the OTG Transceiver via the I²C interface. The mode needs to be set to manual anytime the MCU wants to access the I²C transceivers manually. Otherwise, it should be in Automatic mode to handle normal operation. 1: Hardware control mode 0: Software control mode
I2COE Bit 24	RW	R	Tristates all I²C Output pin: allows a Wired-OR operation (external device has direct control of the OTGXcvr) 1: Output Enable 0: Output Tristate
SEQREADSTRT Bits 23–16	RW	R	Sequential Read Start Address
NUMSEQOPS Bits 15–8	RW	R	Number of sequential operations to be performed on the I²C interface.

Table 32-98. Device Address and I²C Operations Register Description (continued)

Name	SW	HW	Description
RD/WRN Bit 7	RW	R	Read/Write the OTG Transceiver Registers. 0: Write 1: Read
OTGDEVADDR Bits 6–0	RW	R	These bits are the OTG Transceiver's I²C device address. USB_OTG specification requires Add[6:2] = b01011, least significant 2 bits are for chip select (needs to match with ADR input pin to the OTG Transceiver)

32.15.12 I²C Interrupt and Control Register

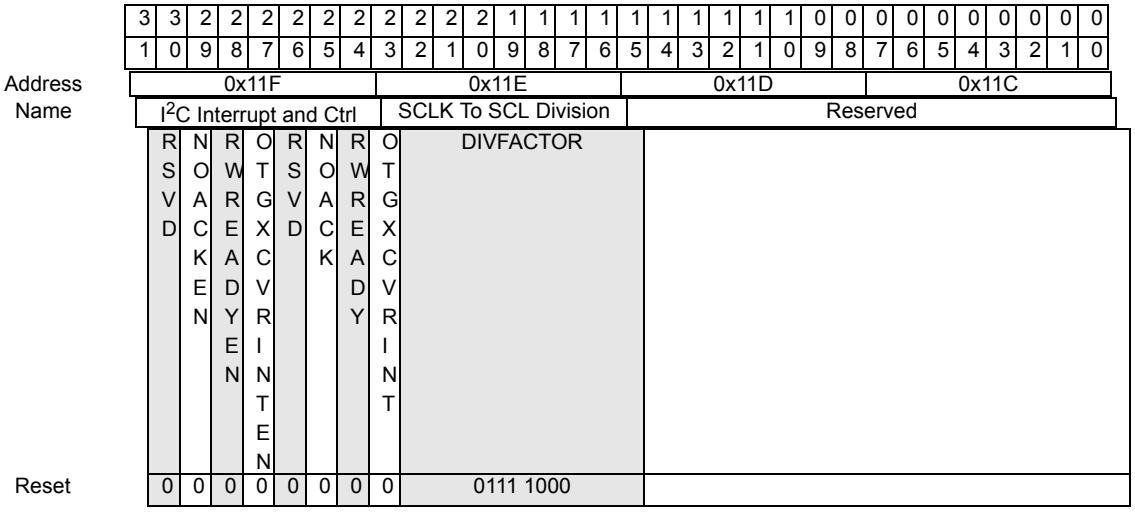


Table 32-99. I2C Interrupt and Control Register

Name	SW	HW	Description
Reserved Bit 31			
NOACKEN Bit 30	RW	R	I²C No ACK Interrupt enable
RWREADYEN Bit 29	RW	R	R/W_ready interrupt enable
OTGXCVRINTEN Bit 28	RW	R	OtgXcvr interrupt enable
Reserved Bit 27			Reserved
NOACK Bit 26	RC	W	Indicates that external OTG Xcvr did not return ACK when it was expected. The Hardware goes into the IDLE state when this error occurs
RWREADY Bit 25	RC	W	I²C Read Write operation is completed or there is no active I²C operation.

Table 32-99. I2C Interrupt and Control Register (continued)

Name	SW	HW	Description
OTGXCVRINT Bit 24	R	W	Interrupt from external OTG Transceiver
DIVFACTOR Bits 23–16	RW	R	This value indicates the division factor between Sclk and the half period of Scl. The Initial value is h'78 (d'120, for Clk = 48MHz, Scl = 200KHz).

32.16 Wake-Up Events and Power Management

The USB OTG module offers a power management feature through clock gating. By clearing the clock enable bits in the ClockControl Register, the clocks are gated in the low position to either the Host Controller and/or the Function Controller. While in this state it is possible that events can occur that require the Host or Function Controllers to respond. These events are called "wake up events" and are described in the following tables.

Table 32-100. Host Wake-Up Events

Description
One or more devices have been connected on a previously unconnected port. This is only detected with the DeviceConnectWakeupEnable bit set in the RootHubStatus Register.
One or more devices have been disconnected on a suspended port.
One or more suspended downstream device initiates RESUME. This is only detected with the DeviceConnectWakeupEnable bit set in the RootHubStatus Register. The port will immediately reflect the RESUME back downstream.
One or more suspended downstream device is disconnected.
One or more downstream devices have caused a potential over current event.

Table 32-101. Function Wake-Up Events

Description
The function has detected a host initiated resume while suspended.
The function has detected a host initiated resume while suspended.
The function receives a reset while suspended
The VBUS power is lost or raised. Please note that this wake up will appear as an HNP wake up event.

HNP wake up events can only occur if both the Host and Function clocks have been turned off.

Table 32-102. HNP Wake-Up Events

Description
A mini-A plug is inserted into or removed from the receptacle, the ID pin value is changed.
A VBUS pulse with the UseVBusPulseSrpDetection SET. This detection will occur in Software HNP.
A Data pulse with the UseDataPulseSrpDetection SET. This detection will occur in Software HNP.

Table 32-102. HNP Wake-Up Events (continued)

Description
As a B-DEVICE detects that the VBUS has transitioned to VBusABSessionValid level or drops below the VBusABSessionValid level.
If the external I^2C-OTG Transceiver is used, the HNP Wake UP event will be the external I^2C-OTG Transceiver interrupt, which is caused by the ID and VBUS status changes.

It is suggested that the application should have enables for these signals in the event that the application does not want to support the asynchronous wake up event. If the application wants to disable the clock to the USB OTG module level it is suggested to disable the clocks of the Host and Function Controller first.

32.16.1 Software Requirements for Wake-Up Events

32.16.1.1 Host Controller

The Host Controller clock is controlled by the ClockControl Register (0x00C). In order for the Host Controller to respond to a wake-up event, software must set at least one of the bits from Table 32-103.

Table 32-103. External Function Wake-Up Events

Action	Effect
SET RemoteWakeUpEnable HostControl Register (0x080) Bit 4	This will allow the host controller to wake up from a resume event on a suspended port.
SET DeviceConnectWakeUpEnable RootHubStatus Register (0x0F0) Bit 15	By setting this bit the detection of a connect event on a powered port, a detection of a disconnect event on a suspended port, or the detection of an over current event will cause a wakeup event.
SET AsyncHostInterruptEnable USB OTG module InterruptStatus Register (0x004) Bit 4	This will allow the wake up signal from the host controller to generate an asynchronous or synchronous interrupt depending if the main clock is enabled

Before Software disables the host clock, it should ensure that all of the following conditions have been met:

- All the enabled and connected Host ports should be in the suspended state. Software needs to wait for at least 5ms before it disables the clock.
- The host controller in the USBSuspend state.
- There are no pending or enabled ETD's.
- There are no pending port changes.

Software should not shut down the clock while the Host Controller is in USBResume state.

The integrator must allow Software to know the source of the wake-up event so that it can re-enable the host clock. Once the clock has been started there may not be any indication immediately what the source of the wakeup is. Software must wait 10ms before disabling the host clock again.Software is responsible to ensure that the clock to the USB OTG module is fully operational before making accesses to the host requests.

32.16.1.2 Function Controller

The Function Controller clock is controlled by the ClockControl Register (0x00C). In order for Software to disable the clock, Software must have received the SuspendDetectedInterrupt. After Software receives the SuspendDetectedInterrupt it cannot start a resume for at least 2 ms, and it must reduce power consumption within 6ms. In order to receive a wake up from the Function Controller the AsyncFunctionWakeUp bit must be set in the USB OTG module InterruptStatus Register (0x004). For an HNP enabled device, Software should ensure that the OTG port is in B_SLAVE or B_IDLE before disabling the clocks.

Chapter 33
PCMCIA/CF Interface

33.1 Overview

PCMCIA is an acronym for the Personal Computer Memory Card International Association. PCMCIA 2.1 is a standard for using memory and I/O devices as insertable, exchangeable peripherals for personal computers or PDAs like memory expansion devices, modems, wireless and local area network adapters.

The PCMCIA/CF controller module provides the control logic for a PCMCIA socket interface and requires only additional external analog power switching logic and buffering. The controller supports one PCMCIA socket.

Figure 33-1 illustrates the block diagram of the PCMCIA/CF interface.

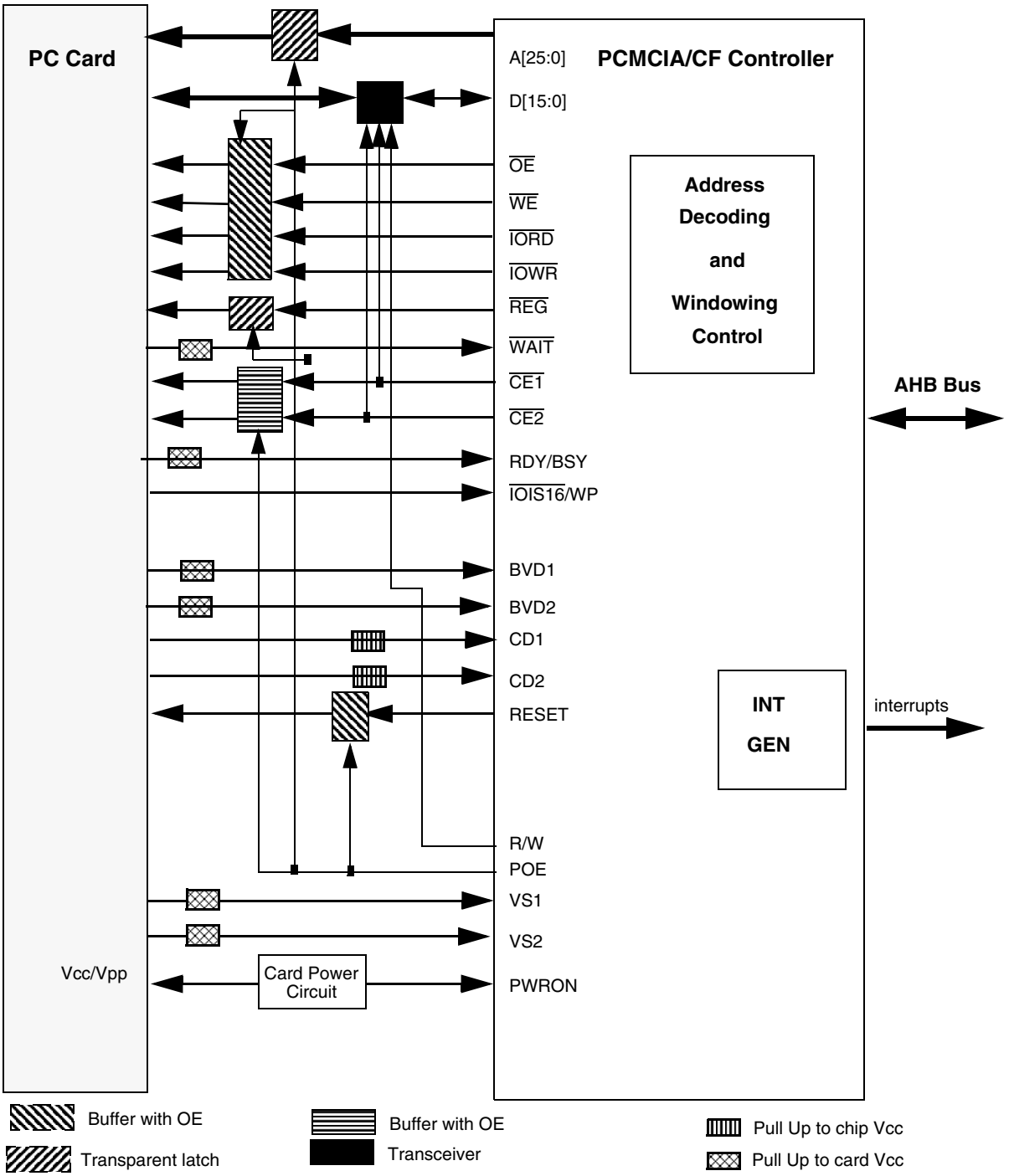


Figure 33-1. PCMCIA/CF Controller Block Diagram

33.1.1 Terminology

The following list identifies some of the acronyms used throughout this chapter.

AMS—Attribute Mode Select

ATA—Advanced Technology (AT) Attachment (IDE interface for PC)

CIS—Card Information Structure

EMI—External Memory Interface

EIM—External Interface to Memory

IDE—Integrated Drive Electronics (hard disk)

PCMCIA—Personal Computer Memory Card International Association

33.1.2 Features

The PCMCIA/CF controller includes these distinctive features:

- A host adapter interface fully compliant with the PCMCIA standard release 2.1 (PC Card -16)
- Supports one PCMCIA socket
- Supports hot-insertion
- Supports card detection
- Mapping to common memory space, attribute memory space and I/O space. Each space is up to 32 kbyte in size
- Supports 5 memory or I/O windows
- Generates a single interrupt to the CPU
- Supports programmable PC card access timing
- Handles card interrupts
- Provides buffer enable and data direction control signals for capacitive load isolation
- Supports memory, I/O, PC-ATA (memory, I/O) and True IDE modes

33.1.3 Modes of Operation

The PCMCIA/CF controller module supports the following modes of operation:

- Memory-only card mode
- I/O card mode
- True IDE mode

33.2 External Signal Description

Table 33-1. PCMCIA/CF External Signal Descriptions

Signal Name	Direction	Description
A[10:0]	O	**Address Bus Signals**—Output signals that allow direct addressing of up to 64Mbytes of memory on the PCMCIA card.
D[15:0]	I/O	**Data Bus Signals**—These are bi-directional signals to the PCMCIA/CF socket data I/O pins.

Table 33-1. PCMCIA/CF External Signal Descriptions (continued)

Signal Name	Direction	Description
$\overline{CE1}$ $\overline{CE2}$	O	**Card Enable Signals**—When a PCMCIA access is performed, CE1 enables even bytes and CE2 enables odd bytes. In True IDE mode, these become CS0 and CS1, respectively. See also Section 33.4.7 on page -22 and Table 33-21.
$\overline{OE}$	O	**Output Enable**—During PCMCIA accesses, this output signal is used to drive memory read data from a PC card in a PCMCIA socket.
$\overline{WE}$	O	**Write Enable/Program**—During PCMCIA access, this output signal is used to latch memory write data to the PC card in a PCMCIA socket. The signal can also be used as the programming strobe for PC cards that have programmable memory capability.
$\overline{REG}$	O	**Register Accesses**—Output signal for Attribute Memory Select. When REG is asserted during PCMCIA access, card access is limited to attribute memory when a memory access occurs (WE or OE are asserted) and to I/O ports when I/O access occurs (IORD or IOWR is asserted). If REG is asserted, accesses to common memory are blocked.
$\overline{IORD}$	O	**I/O Read**—This output goes active (low) for I/O reads from the socket. This signal is asserted together with REG and it is used to read data from the PC card I/O space. IORD is valid only when REG and either CE1 or CE2 signals are also asserted.
$\overline{IOWR}$	O	**I/O Write**—This output goes active (low) for I/O write to the socket. Asserted with REG during PCMCIA accesses, used to latch data into the PC card I/O space. IOWR is valid only when REG and either CE1 and CE2 signals are also asserted.
$\overline{WAIT}$	I	**Extend Bus Cycle**—This input signal is asserted by the PC card to delay completion of the pending memory or I/O cycle.
$\overline{IOIS16}$/WP	I	**I/O Port**—This is a 16-bit input signal. When the card and its socket are programmed for I/O interface operation, this signal is used as IOIS16 and must be asserted by the PC card when the address on the bus corresponds to an address on the PC card and the I/O port being addressed supports 16-bit accesses. If the I/O region in which the address resides is programmed as 8-bit wide IOIS16 is ignored. **Write Protect (WP)**—This input is used when the card and socket are programmed for memory interface operation. It reflects the state of the write protect of the PC card. The PC card must assert WP when the card switch is enabled. It must be negated when the switch is disabled. For a PC card that is writable without a switch, WP must be connected to ground. If the PC card is permanently write-protected, WP must be connected to Vcc.
$\overline{VS1}$ $\overline{VS2}$	I	**Voltage Sense 1 and 2**—These input signals are generated by the PC card to notify the socket of the card's CIS Vcc requirements.
$\overline{CD1}$ $\overline{CD2}$	I	**Card Detect**—This signal provides proper detection of card insertion. The signals must be connected to ground internally on the PC card, thus, these signals are forced low when a card is placed on the socket. These signals must be pulled up to system Vcc to allow card detection to function when the card socket is powered down.
BVD1/$\overline{STSCHG}$ BVD2/$\overline{SPKR}$	I	**Battery Voltage Detect**—When the card and its socket are programmed for memory interface operation, these signals are generated by the PC card with on board battery to report the battery condition. See Table 33-2.
	I	**STSCHG**—When the card is in I/O interface operation, BVD1 is used as Status Change and is generated by the I/O PC card. Status Change must be held negated when the "signal on change" bit and the "change" bit in the card status register are either or both zero. STSCHG must be asserted when both bits = 1.
	I	**SPKR**—When the card is in I/O interface operation BVD2 is used as Audio Digital Waveform. A card that does not have this capability should drive SPKR high.

Table 33-1. PCMCIA/CF External Signal Descriptions (continued)

Signal Name	Direction	Description
READY/$\overline{\text{IREQ}}$	I	**Ready and Interrupt Request**—When the card and its socket are programmed for memory interface operation, this signal is used as RDY/$\overline{\text{BSY}}$ and must be asserted by a PC card to indicate that a PC card is busy processing a previous write command. When the card and its socket are programmed for I/O interface operation, this signal is used as $\overline{\text{IREQ}}$ and must be asserted by a PC card to indicate that a device on the PC card requires service by host software. Must be held negated when no interrupt is requested.
RESET	O	**Card Reset**—This output signal is provided to clear a card's Configuration Option Register, thus placing a card in its default (memory only interface) state and beginning an additional card initialization. RESET signal has inverted polarity when in True IDE mode.
The following pins are not PCMCIA standard pins but are part of the PCMCIA/CF controller module.		
PWRON	I	**Power is On**—The PC card supply circuitry can use this signal as an interrupt to notify when the card's power supply reaches the full required voltage.
R/$\overline{\text{W}}$	O	**External Transceiver Direction**—This output signal is negated during read cycles and asserted during write accesses.
$\overline{\text{POE}}$	O	**PCMCIA Buffers Output Enable**—An output line reflecting the value of PGCR[POE] bit. Used to tri-state control signals and to latch the address. See Figure 33-2.
SPKROUT	O	**Speaker Output**—This signal provides a digital audio waveform to be driven to the system's speaker. This signal is connected directly to the SPKR input. SPKROUT can be connected to the system speaker through a low-pass filter to play the audio waveform received at the SPKR input.

Table 33-2. BVD1 and BVD2 Signal Description

BVD1	BVD2	Description
1	1	Battery is in good condition.
1	0	Battery is in warning condition and should be replaced, although data integrity on the card is assured.
0	X	Battery is in no longer serviceable and data is lost.

33.3 Programming Model

Table 33-3 is the register summary of the PCMCIA/CF controller. The PCMCIA/CF controller maps the PC-Card attribute memory, common memory, and I/O accesses to the 64 Mbyte region defined by 0xD400_0000 to 0xD7FF_FFFF.

Table 33-3. PCMCIA/CF Controller Summary

Address	Description/Name	Access
0xDF00 2000	PCMCIA input pins register (PIPR)	Read Only
0xDF00 2004	PCMCIA status changed register (PSCR)	Read - W1C
0xDF00 2008	PCMCIA enable register (PER)	Read-Write
0xDF00 200C	PCMCIA Base Register 0 (PBR0)	Read-Write
0xDF00 2010	PCMCIA Base Register 1 (PBR1)	Read-Write
0xDF00 2014	PCMCIA Base Register 2 (PBR2)	Read-Write
0xDF00 2018	PCMCIA Base Register 3 (PBR3)	Read-Write
0xDF00 201C	PCMCIA Base Register 4 (PBR4)	Read-Write
0xDF00 2028	PCMCIA Option Register 0 (POR0)	Read-Write
0xDF00 202C	PCMCIA Option Register 1 (POR1)	Read-Write
0xDF00 2030	PCMCIA Option Register 2 (POR2)	Read-Write
0xDF00 2034	PCMCIA Option Register 3 (POR3)	Read-Write
0xDF00 2038	PCMCIA Option Register 4 (POR4)	Read-Write
0xDF00 2044	PCMCIA Offset Register 0 (POFR0)	Read-Write
0xDF00 2048	PCMCIA Offset Register 1 (POFR1)	Read-Write
0xDF00 204C	PCMCIA Offset Register 2 (POFR2)	Read-Write
0xDF00 2050	PCMCIA Offset Register 3 (POFR3)	Read-Write
0xDF00 2054	PCMCIA Offset Register 4 (POFR4)	Read-Write
0xDF00 2060	PCMCIA General Control Register (PGCR)	Read-Write
0xDF00 2064	PCMCIA General Status Register (PGSR)	Read Only

33.3.1 PCMCIA Input Pins Register (PIPR)

Status of inputs from PCMCIA card to the host (BVD, CD, RDY, VS) is reported to the PIPR, shown in Figure 33-1. PIPR is a read only register; write operations are ignored.

When accessing this register 2-wait state will be added by the PCMCIA/CF controller.The reset values mentioned in Figure 33.3.1 are the reset values of the PIPR, however, a read to the register will return the values seen on the external pins.

PIPR — PCMCIA Input Pins Register — Addr 0xDF00 2000

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								POWERON	RDY	BVD2	BVD1	CD		WP	VS	
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0

Table 33-4. PCMCIA Input Pins Register Description

Name	Description	Settings
Reserved Bits 31–9	Reserved—These bits are reserved and should read 0.	
POWERON Bit 8	**Power On**—This bit indicates the status of the power is on signal from the card.	1 = Card indicates power is on. 0 = Card indicates that it did not reach its power supply requirements.
RDY Bit 7	**RDY/$\overline{\text{BSY}}$/$\overline{\text{IREQ}}$**—When the card and its socket are in memory interface operation, this bit functions as RDY/BSY indicating that the card is busy processing a previous write command. When the card and its socket are in I/O interface operation, this bit functions as IREQ indicating that a device on the PC card requires service by host software.	*Memory Interface mode:* 1 = PC card is ready to accept a new data-transfer command. 0 = PC card is busy processing a previous command or performing initialization. *I/O interface mode:* 1 = No host software service required (default). 0 = A card device requires host software service.
BVD2 Bit 6	**Battery Voltage Detect 1/SPKR In**—When the card and its socket are in memory interface mode, this bit reflects the BVD2 signal. When the card and its socket are in I/O mode, this bit is used as SPKR IN.	N/A

Table 33-4. PCMCIA Input Pins Register Description (continued)

Name	Description	Settings
BVD1 Bit 5	**Battery Voltage Detect 1/STSCHG In**—When the card and its socket are in memory interface mode, this bit reflects the BVD1 signal. When the card and its socket are in I/O mode, this bit is used as STSCHG indicator.	N/A
CD Bits 4–3	**Card Detect 1 and Card Detect 2**—Card detect 1 and card detect 2 bits indicate of a proper detection of card insertion. Only when both bits are 0, a card is properly inserted. To save power these bits will be asynchronous.	N/A
WP Bit 2	**Write Protect**—This bit reflects the state of the write protect switch on the PC card.	1 = Write Protect switch is enabled 0 = Write Protect Switch is disabled
VS Bits 1–0	**Voltage Sense**—The voltage sense bits notify the host of the card's CIS Vcc requirements. This data can be used by the host to control external voltage transceiver.	N/A

33.3.2 PCMCIA Status Change Register (PSCR)

The bits in the PSCR register are set when there is any change in the corresponding signal. Writing zeroes has no affect; writing ones clears the corresponding interrupt state. The content of PSCR, shown in Figure 33-1, are logically ANDed with the PER to generate a PCMCIA/CF controller interrupt.

When accessing this register a 2-wait state will be added by the PCMCIA/CF controller.

PSCR — PCMCIA Status Change Register — Addr 0xDF00 2004

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					POWC	RDYR	RDYF	RDYH	RDYL	BVDC2	BVDC1	CDC2	CDC1	WPC	VSC2	VSC1
TYPE	r	r	r	r	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 33-5. PCMCIA Status Change Register Description

Name	Description
Reserved Bits 31–12	Reserved—These bits are reserved and should read 0.

Table 33-5. PCMCIA Status Change Register Description (continued)

Name	Description
POWC Bit 11	**Power On Signal Changed** 0 = No change has occurred in the POWERON signal since system reset. 1 = A change has occurred in the POWERON signal since system reset.
RDYR Bit 10	**Ready / Interrupt Request Rising Edge**—Device and socket positive edge interrupt. 0 = No rising edge has occurred in RDY since system reset. 1 = A rising edge occurred in RDY since system reset.
RDYF Bit 9	**Ready / Interrupt Request Falling Edge**—Device and socket negative edge interrupt. 0 = No falling edge has occurred in RDY since system reset. 1 = A falling edge occurred in RDY since system reset.
RDYH Bit 8	**Ready / Interrupt Request High**—This bit reflects value of RDY signal. RDY/$\overline{\text{IREQ}}$ pin is high indicating a device and socket high level interrupt.
RDYL Bit 7	**Ready / Interrupt Request Low**—Device and socket low-level interrupt. This bit is the inverted value of RDY signal.
BVDC2 Bit 6	**Battery Voltage Detect 2/SPKR In Changed** 0 = No change has occurred in Battery Voltage 2 or SPKR since system reset. 1 = A change has occurred in Battery Voltage 2 or SPKR since system reset.
BVDC1 Bit 5	**Battery Voltage Detect 1/STSCHG Changed** 0 = No change has occurred in Battery Voltage #1 since system reset. 1 = A change has occurred in Battery Voltage #1 since system reset.
CDC2 Bit 4	**Card Detect 2 Changed** 0 = No change has occurred in card detect 2 since system reset. 1 = A change has occurred in card detect 2 since system reset.
CDC1 Bit 3	**Card Detect 1 Changed** 0 = No change has occurred in card detect 1 since system reset. 1 = A change has occurred in card detect 1 since system reset.
WPC Bit 2	**Write Protect Changed** 0 = No change has occurred in the write protect status since system reset. 1 = A change has occurred in the write protect status since system reset.
VSC2 Bit 1	**Voltage Sense 2 Changed** 0 = No change has occurred in voltage sensor #2 since system reset. 1 = A change has occurred in voltage sensor #2 since system reset.
VSC1 Bit 0	**Voltage Sense 1 Changed** 0 = No change has occurred in voltage sensor #1 since system reset. 1 = A change has occurred in voltage sensor #1 since system reset.

33.3.3 PCMCIA Enable Register (PER)

Setting a bit in PER, shown in Figure 33-3, enables the corresponding interrupt. When accessing this register 1-wait state will be added by the PCMCIA/CF controller.

PER — PCMCIA Enable Register — 0xDF00 2008

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				ERRINTEN	POWERONEN	RDYRE	RDYFE	RDYHE	RDYLE	BVDE2	BVDE1	CDE2	CDE1	WPE	VSE2	VSE1
TYPE	r	r	r	r/w	r/w	r/w	r/w	r/w	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	0	0	0	1	1	0	0	0

Table 33-6. PCMCIA Enable Register Description

Name	Description
Reserved Bits 31–13	Reserved—These bits are reserved and should read 0.
ERRINTEN Bit 12	**Error Interrupt Enable**—Setting this bit enables the interrupt as a result of an error signal. See 33.4.2.1 for more information. 0 = Interrupt disabled. 1 = Interrupt enabled. Note that the default is to enable the interrupt.
POWERONEN Bit 11	**Power On Interrupt Enable**—Setting this bit enables the interrupt as a result of any Power On signal change. 0 = Interrupt disabled. 1 = Interrupt enabled.
RDYRE Bit 10	**RDY/$\overline{\text{IREQ}}$ Rising Edge Interrupt Enable** 0 = Interrupt disabled. 1 = Interrupt enabled.
RDYFE Bit 9	**RDY/$\overline{\text{IREQ}}$ Falling Edge Interrupt Enable** 0 = Interrupt disabled. 1 = Interrupt enabled.
RDYHE Bit 8	**RDY/$\overline{\text{IREQ}}$ High Interrupt Enable** 0 = Interrupt disabled. 1 = Interrupt enabled.
RDYLE Bit 7	**RDY/$\overline{\text{IREQ}}$ Low Interrupt Enable** 0 = Interrupt disabled. 1 = Interrupt enabled.
BVDE2 Bit 6	**Battery Voltage 2/SPKR In Interrupt Enable**—Setting this bit enables the interrupt as a result of any signal change from battery voltage sensor 2, or from the SPKR IN signal. 0 = Interrupt disabled. 1 = Interrupt enabled.

Table 33-6. PCMCIA Enable Register Description (continued)

Name	Description
BVDE1 Bit 5	**Battery Voltage 1/STSCHG Interrupt Enable**—Setting this bit enables the interrupt as a result of any signal change from the battery voltage sensor 1. 0 = Interrupt disabled. 1 = Interrupt enabled.
CDE2 Bit 4	**Card Detect 2 Interrupt Enable**—Setting this bit enables the interrupt as a result of any signal change from card detect #2. **Note:** the default setting enables the interrupt 0 = Interrupt disabled. 1 = Interrupt enabled.
CDE1 Bit 3	**Card Detect 1 Interrupt Enable**—Setting this bit enables the interrupt as a result of any signal change from card detect #1 **Note:** the default setting enables the interrupt 0 = Interrupt disabled. 1 = Interrupt enabled.
WPE Bit 2	**Write Protect Interrupt Enable**—Setting this bit enables the interrupt as a result of any signal change in the write protect status 0 = Interrupt disabled. 1 = Interrupt enabled.
VSE2 Bit 1	**Voltage Sense 2 Interrupt Enable**—Setting this bit enables the interrupt as a result of any signal change from voltage sensor #2. 0 = Interrupt disabled. 1 = Interrupt enabled.
VSE1 Bit 0	**Voltage Sense 1 Interrupt Enable**—Setting this bit enables the interrupt as a result of any signal change from voltage sensor #1. 0 = Interrupt disabled. 1 = Interrupt enabled.

33.3.4 PCMCIA Base Registers 0–4 (PBR0–PBR4)

PBA [10:0]—PCMCIA base address. Compared to the address bus to determine if a PCMCIA window is being accessed by an internal bus master. PBA is used in conjunction with POR[BSIZE]. When accessing this register 1-wait state is added by the PCMCIA/CF controller.

PBR0–PBR4 PCMCIA Base Registers 0–4 Addr 0xDF00 200C ... 0xDF00 201C

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		PBA														
TYPE	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 33-7. PCMCIA Base Register Description

Name	Description
Reserved Bits 31–15, 3–0	Reserved—these fields are reserved and should be set to 0.
PBA Bit 14–4	**Base Register Address**—This field contains the base register address used by the PCMCIA device.

33.3.5 PCMCIA Option Registers 0–4 (POR0–POR4)

The POR registers handle time manipulation, provide the address mask for the bank size, and define the region, write protection, and validation. When accessing this register 1-wait state is added by the PCMCIA/CF controller.

POR0–POR4 PCMCIA Option Registers 0–4 Addr 0xDF00 2028 ... 0xDF00 2038

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			PV	WPEN	WP	PRS		PPS	PSL							PSST
TYPE	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSST					PSHT							BSIZE			
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 33-8. PCMCIA Options Register Description

Name	Description	Settings
Reserved Bits 31–30	Reserved—These bits are reserved and should read 0.	
PV Bit 29	**PCMCIA Valid**—Defines whether the contents of the PBR and POR pair are valid (window enable).	1 = This bank is valid 0 = This bank is invalid
WPEN Bit 28	**PCMCIA Write Protect Input Enable**—This bit is the write protect input signal enable bit, controlled by software. When this bit is cleared the WP input to the PCMCIA/CF controller module is ignored. For the relationship between this bit, the WP bit, and the WP signal coming from the card refer to Section 33.4.5, “Write Protect (WP),” on page -21. **Note:** Setting PSST=00001 is not allowed when WPEN bit is set since the synchronization of WP signal takes 2 clocks.	1 = Write Protect input (WP) signal is enabled 0 = Write Protect (WP) input signal is ignored
WP Bit 27	**PCMCIA Write Protect Enable**—This bit is the write protect enable bit controlled by software. To the relation between this bit, the WPEN bit, and the WP signal coming from the card refer to Section 33.4.5, “Write Protect (WP),” on page -21.	1 = Write Protected. Attempting to write to this window causes an interrupt 0 = Not write protected
PRS Bits 26–25	**PCMCIA Region Select**—	See Table 33-14
PPS Bit 24	**PCMCIA Port Size**—Specifies the port size of the PCMCIA window. Refer to Section 33.4.6, “16-bit/8-bit Support,” on page -21 for more details.	1 = 8-bit port size 0 = 16-bit port size

Table 33-8. PCMCIA Options Register Description (continued)

Name	Description	Settings
PSL Bits 23–17	**PPCMCIA Strobe Length**—Determines the number of cycles the strobe is asserted during a PCMCIA access for this window and, thus, it is the main parameter for determining cycle length. The cycle may lengthened by asserting $\overline{\text{WAIT}}$. **Note:** To sample $\overline{\text{WAIT}}$ signal the PSL should be calculated as the maximum valid time on WAIT plus two. For example suppose a 100MHz system bus clock and the card's specification requires that the time from $\overline{\text{WE}}$/$\overline{\text{OE}}$ low to $\overline{\text{WAIT}}$ valid is 70ns, then PSL should be at least: 70ns/10ns+2=9.	See Table 33-13
PSST Bits 16–11	**PCMCIA Strobe Setup Time (address to strobe assertion)**—Specifies when $\overline{\text{IOWR}}$ or $\overline{\text{WE}}$ are asserted during a PCMCIA write access or when $\overline{\text{IORD}}$ or $\overline{\text{OE}}$ are asserted during a PCMCIA read access handled by the PCMCIA/CF controller. This helps meet address/setup time requirements for slow memories and peripherals.	See Table 33-12
PSHT Bits 10–5	**PCMCIA Strobe Hold Time (strobe negation to address negation)**—Specifies when $\overline{\text{IOWR}}$ or $\overline{\text{WE}}$ are negated during a PCMCIA write or when $\overline{\text{IORD}}$ or $\overline{\text{OE}}$ are negated during a PCMCIA read. Used to meet address/data hold time requirements for slow memories and peripherals.	See Table 33-11
Reserved Bit 4	This bit is reserved and should read 0.	
BSIZE Bits 3–0	**PCMCIA Bank Size**—Determines the address mask field of each POR and provides masking for any of the corresponding bits in the associated PBR. The bank size is calculated as banksize = 2, where BSIZE represents the gray code shown on Table 33-8.	See Table 33-9

Table 33-9 provides the values for the Bank Size field. Refer to the MC9328MX21RM Addendum if your silicon ID is the older, *n*L45X silicon version for Bank Size values. See Chapter 8 Section 8.1.1, "Silicon ID Register." to identify your specific silicon version.

Table 33-9. BSIZE Values

Value	Meaning	Value	Meaning
0000	16 byte	0101	1 kbyte
0001	32 byte	0100	2 kbyte
0011	64 byte	1100	4 kbyte
0010	128 byte	1101	8 kbyte
0110	256 byte	1111	16 kbyte
0111	512 byte	1110	32 kbyte

BSIZE determines not only the bank size, but also how the address is compared with PBR[PBA]. If virtual field, MASK is defined as shown on Table 33-10. Refer to the MC9328MX21RM Addendum if your silicon ID is the older, *n*L45X silicon version for Bank Size values. See Chapter 8 Section 8.1.1, "Silicon ID Register." to identify your specific silicon version.

Table 33-10. BSIZE Mask

BSIZE	A14						Mask							A0
0000	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0001	1	1	1	1	1	1	1	1	1	0	0	0	0	0
0011	1	1	1	1	1	1	1	1	0	0	0	0	0	0
0010	1	1	1	1	1	1	1	0	0	0	0	0	0	0
0110	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0111	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0101	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0100	1	1	1	0	0	0	0	0	0	0	0	0	0	0
1100	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1101	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1111	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1110	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Addr and MASK = PBA and MASK for a valid PCMCIA access; otherwise, it is not a valid PCMCIA access.

Table 33-11. PSHT Bits

PSHT[5:0]	Description
000000	Strobe negation to address change 0 clock
000001	Strobe negation to address change 1 clock
…	–
111111	Strobe negation to address change 63 clock

Table 33-12. PSST Bits

PSST[5:0]	Description
000000	Reserved
000001	Address to strobe assertion 1 clock
000010	Address to strobe assertion 2 clock
…	–
111111	Address to strobe assertion 63 clock

NOTE

Using PSST=00001 is not allowed when WPEN bit is set since the synchronization of WP signal takes 2 clocks.

Table 33-13. PSL Bits

PSL[6:0]	Description
0000000	Strobe asserted 128 clocks cycles
0000001	Strobe asserted 1 clocks cycles
0000010	Strobe asserted 2 clocks cycles
…	–
1111111	Strobe asserted 127 clocks cycles

Table 33-14. PRS Bits

PRS	Description
00	Common memory space
01	TrueIDE mode
10	Attribute memory space
11	I/O space

33.3.6 PCMCIA Offset Registers 0–4 (POFR0–POFR4)

The offset address of the window. PBA is used in conjunction with POR[BSIZE]. The external address is ext_addr = POFA + haddr and MASK.

For example, if:

```
haddr = 0x263, MASK = 0x7C0, POFA = 0x161
```

then:

```
ext_addr = 0x161 + 0x263 and (0x7C0) = 0x161 + 0x23 = 0x184
```

When accessing this register 1-wait state is added by the PCMCIA/CF controller.

POFR0–POFR4 — PCMCIA Offset Registers 0–4 — Addr 0xDF00 2044 ... 0xDF00 2054

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		POFA														
TYPE	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 33-15. PCMCIA Offset Register Description

Name	Description
Reserved Bits 31–15, 3–0	Reserved—These bits are reserved and should read 0.
POFA Bits 14–4	**PCMCIA Offset Address**

33.3.7 PCMCIA General Control Register (PGCR)

This is a general control register. When accessing this register 1-wait state is added by the PCMCIA/CF controller.

PGCR — PCMCIA General Control Register — Addr 0xDF00 2060

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													LPMEN	SPKREN	POE	RESET
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Table 33-16. PCMCIA General Control Register Description

Name	Description	Settings
LPMEN Bit 3	**LPMEN**—Low Power Mode Enable This bit enters the module into low power mode, external memory accesses are disabled. The reset value is 1 (Low power mode)	1 = Low power mode 0 = Normal mode
SPKREN Bit 2	**SPKREN**—SPKROUT routing Enable This bit enables the routing of SPKRIN to SPKROUT.	1 = Routing enable 0 = Routing disabled
POE Bit 1	**POE**—Controller Output Enable This bit is enables the POE signal, used to three-state the external buffers. The POE signal will be: POE (signal) = PGCR[POE] and (PCMCIA_ACCESS (signal) | ATAMODE).	N/A
RESET Bit 0	**RESET**—Card Reset. This bit provides a software reset to the card. This bit is not self cleared, software should take care of changing the state of this bit in order to take the card out of reset.	1 = Provide a reset to the card 0 = Clear the reset bit

33.3.8 PCMCIA General Status Register (PGSR)

This is a general status register. If an error interrupt was generated to the host, the host can access this register in order to find out what caused the error interrupt. All the bits in this register are cleared by writing 1 to the appropriate bit. Writing 0 has no effect. When accessing this register 1-wait state is added by the PCMCIA/CF controller.

PGSR — PCMCIA General Status Register — Addr 0xDF00 2064

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												NWINE	LPE	SE	CDE	WPE
TYPE	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 33-17. PCMCIA General Status Register Description

Name	Description
NWINE Bit 4	**No Window Error**—Attempt to access a card address to an address that is not mapped by any window.
LPE Bit 3	**Low Power Error**—Attempt to access a card when in low power mode.
SE Bit 2	**Size Error**—A 16-bit access to an 8-bit card. See Section 33.4.2.1 on page -20.
CDE Bit 1	**Card Detect Error**—Attempt to access a card when the card is not inserted. Once set, no access is allowed until cleared (even if a card is inserted).
WPE Bit 0	**Write Protect Error**—Attempt to write to a write protected address.

33.4 Functional Description

This section describes the operation of memory and I/O cards, interrupt detection and handling, power control and reset.

33.4.1 Windowing Capabilities

The PCMCIA/CF controller provides 5 memory windows. The user can define each memory window as a common memory space, I/O space, True IDE or attribute memory space. This is done by programming the PRS bits in each POR register for each memory window.

Configuring a window is done by programming the window's base address (PBA bits in the corresponding PBR register), and by programming the bank size (BSIZE bits in the corresponding POR register).

33.4.1.1 Window Overlapping

Window overlapping is not allowed. The PCMCIA/CF controller does not notify the CPU about window overlapping. It is the software's responsibility to ensure this does not occur. Having overlapping windows will have unexpected results.

33.4.2 $\overline{\text{WAIT}}$ Signal

WAIT signal is asserted by the PC card to delay completion of the pending cycle. The access must terminate before the bus time out monitor generates a bus time-out error (1024 cycles in i.MX21).

33.4.2.1 Error Interrupt

Any of the following error conditions will cause an Error Interrupt:

- Attempt to access a card when the card is in Low Power Mode.
- Attempt to write to a write protected area.
- Attempt to access a card when the card is not inserted.
- Attempt to do a 16-bit or 32-bit access to an 8-bit card violating the PPS settings. See Section 33.4.6, "16-bit/8-bit Support," on page -21.
- Attempt to access card with an address that does not match any memory window.

An access to the card which yields an error will take 0–3 wait states to complete according to the following rules:

- Card Detect error (when PGSR[CDE] is cleared) and Write Protect (WP) error, which result from the WP signal from the card, will take 2 wait states. In case these signals (WP and CD) change during access, the access will end 2 cycles after (see note).
- Size error, no window error, and write protect error, which comes from POR[WP], will take 1 wait state.
- Low power mode error and card access with PGSR[CDE] set takes 0 wait states.

NOTE

Due to the synchronization mechanism on CD1 and CD2 signals and the WP signal, a change in these signals during access less than two cycles before it finishes will not yield an error. This should not be a problem since these signals are not expected to change much.

33.4.3 Power Control

When LPMEN is set in the PGCR the PCMCIA/CF controller internal clocks should be gated to the off value. The module is in "listening mode"; that is, it waits for an indication that a card has been inserted. All the static signals are synchronous in both cases and status change interrupts are generated if necessary (in order to wake up the core from stop on card detect, for example). In the first case (LPMEN) read from

PIPR and read/write from/to PSCR should take 2 wait states to complete because of the synchronization of the static signals to the core's clock.

33.4.4 Reset and Tri-state Control

The card can be reset by software by writing to the RESET bit in the PGCR register. Output of external latches can be disabled by writing to the POE bit in PGCR register.

33.4.5 Write Protect (WP)

Write protect is handled in 2 ways:

- Card write protect—The WP signal coming from the card.
- Window write protect—The WP bit in the POR register.

When the WPEN is cleared and the card is in memory interface mode, the WP signal coming from the card is ignored. This way it is possible to enable write protection on selected memory regions, even if the card's WP pin is asserted.

When the card is in IO mode (POR.PRS = 11) WPEN bit is ignored.

Table 33-18. Write Protect

POR.PRS	POR.WP Bit	WP Signal	POR.WPEN	Protect Mode
X0 (memory mode)	X	0	1	Write enabled
		1	1	Write protected
	0	X	0	Write enabled
	1	X	0	Write protected
11(I/O mode)	0	X	X	Write enabled
	1			Write protected

An interrupt is generated by the PCMCIA/CF controller at any attempt to access a write protected area.

33.4.6 16-bit/8-bit Support

The PCMCIA/CF controller supports 16-bit and 8-bit accesses. The access size is defined according to $\overline{\text{IOIS16}}$ and the PPS bit in the POR register. See Table 33-19.

Table 33-19. $\overline{\text{IOIS16}}$ and PPS Relationships

IOIS16	PPS	Access size
0	0	16-bit access to a 16-bit card. The host can generate 8-bit accesses and 16-bit access.
0	1	8-bit access although the card is 16-bit. The host should generate 8-bit accesses only. If the host tries to do a 16-bit access an interrupt is generated.

Table 33-19. $\overline{\text{IOIS16}}$ and PPS Relationships (continued)

IOIS16	PPS	Access size
1	0	8-bit access although 16-bit access is selected by PPS. If the host attempts to do a 16-bit access the PCMCIA/CF controller writes the lower part of the data to the card.
1	1	8-bit access to 8-bit card. The host should generate 8-bit accesses only. If the host tries to do a 16-bit access an interrupt is generated.

33.4.7 Data and Control Signals Relationships

Table 33-20 describes the relationship between Data and Control signals for the different access modes.

Table 33-20. Data and Control Relationships

Function Mode	REG	CE2	CE1	A0	OE	WE	IORD	IOWR	D[15:8]	D[7:0]
Standby mode	x	1	1	x	x	x	x	x	High-Z	High-Z
8-bit read from common memory	1 1 1	1 1 0	0 0 1	0 1 x	0 0 0	1 1 1	1 1 1	1 1 1	High-Z High-Z Odd-Byte*	Even-Byte Odd-Byte High-Z*
16-bit read from common memory	1	0	0	x	0	1	1	1	Odd-Byte	Even-Byte
8-bit write to common memory	1 1 1	1 1 0	0 0 1	0 1 x	1 1 1	0 0 0	1 1 1	1 1 1	xxx xxx Odd-Byte*	Even-Byte Odd-Byte xxx*
16-bit write to common memory	1	0	0	x	1	0	1	1	Odd-Byte	Even-Byte
8-bit read from attribute memory	0 0 0	1 1 0	0 0 0	0 1 x	0 0 0	1 1 1	1 1 1	1 1 1	High-Z High-Z Not-valid	Even-Byte Not-valid Even-Byte
16-bit read from attribute memory	0	0	0	x	0	1	1	1	Not-valid	Even-Byte
8-bit write to attribute memory	0 0	1 0	0 0	0 x	1 1	0 0	1 1	1 1	xxx xxx	Even-Byte Even-Byte
8-bit read from I/O	0 0 0	1 1 0	0 0 1	0 1 x	1 1 1	1 1 1	0 0 0	1 1 1	High-Z High-Z Odd-Byte*	Even-Byte Odd-Byte High-Z*
16-bit read from I/O	0	0	0	x	1	1	0	1	Odd-Byte	Even-Byte
8-bit write to I/O	0 0 0	1 1 0	0 0 1	0 1 x	1 1 1	1 1 1	1 1 1	0 0 0	xxx xxx Odd-Byte	Even-Byte Odd-Byte xxx
16-bit write to I/O	0	0	0	x	1	1	1	0	Odd-Byte	Even-Byte
I/O inhibit	1	x	x	x	x	x	0	1	High-Z	High-Z

NOTE

These are all the access modes which are supported by the standard. In PCMCIA/CF controller the 8-bit access to odd byte is done by CE1 and A0 only; That is, the data will be always driven on D[7:0].

33.4.8 True IDE Mode

33.4.8.1 Pin Assignment

The ATA standard specifies the AT attachment interface between host systems and storage devices. The PCMCIA/CF controller supports the PC-Card ATA Protocol via memory and IO mode. In addition to the PC-Card ATA Protocol, the controller supports True IDE mode. True IDE mode is supported by Compact Flash cards and in this mode, no CIS support is required. In True IDE mode, the card and therefore the controller, too, require only to support 3 address lines, A2-A0.

The True IDE signal assignments on the PCMCIA/CF connector is described in Table 33-21.

Table 33-21. True IDE Signal Names and Assignments

PC Card Signal	True IDE Signal	Comments
D[15:0]	D[15:0]	–
CE1	CS0	Task file register select in TrueIDE mode.
CE2	CS1	Alternate status register select in TrueIDE mode.
OE	ATASEL	The card samples this bit on power-on sequence. if low the card will enter TrueIDE mode else it will enter PC CARD mode.
A[2:0]	A[2:0]	Address
A[10:3]	not used	These bit should be connected to 0 on TrueIDE
WE	not used	this pin should be pulled high at the host
READY/$\overline{\text{IREQ}}$	INTRQ	Interrupt request—INTRQ is the True IDE notation and is asserted HIGH.
WP/$\overline{\text{IOIS16}}$	IOCS16	–
CD1	CD1	–
CD2	CD2	–
VS1	VS1	–
VS2	VS2	–
RESET	RESET	This signal is asserted LOW in True IDE mode, HIGH in other modes.
WAIT	IORDY	I/O Ready—asserted HIGH - polarity inversion of $\overline{\text{WAIT}}$.
REG	not used	this pin should be pulled high at the host
BVD1/$\overline{\text{STSCHG}}$	PDIAG	Diagnostics complete signal.
BVD2/$\overline{\text{SPKR}}$	DASP	Disk Active

Table 33-21. True IDE Signal Names and Assignments (continued)

PC Card Signal	True IDE Signal	Comments
IORD	IORD	–
IOWR	IOWR	–

NOTE

In True IDE mode, #CS0 and #CS1 (task file chip select) behave differently from #CE1 and #CE2.

33.4.8.2 Data Access

True IDE devices define a primary set and alternate set of registers. The primary set of registers are referred to as the Task File (8 registers) and the alternate set is referred to as Alternate Status and Device Control Registers (Alternate Registers). These 2 sets of registers overlap each other and access to the Task File or the Alternate Registers is determined by the $\overline{CS0}$ and $\overline{CS1}$ signals. When $\overline{CS0}$ is low and $\overline{CS1}$ is high, the Task File is accessed and when $\overline{CS0}$ is high and $\overline{CS1}$ is low, the Alternate Registers are accessed.

When a window is configured in True IDE mode, then the controller decides which set of registers to access based on the address presented to it. If address bit, A3 is 0, then the Task File is accessed and when A3 is 1, then the Alternate Registers are accessed.

haddr[3] = 0 – Access to Task File

haddr[3] = 1 – Access to Alternate Registers

Table 33-22. Data, Control, and Address Relations in TrueIDE Mode

Function Mode	CE2	CE1	A0–3	IORD	IOWR	D[15:8]	D[7:0]
Standby mode	1	1	xx	x	x	High-z	High-z
Task file Write	1	0	1–7h	1	0	xxx	Data In
Task file Read	1	0	1–7h	0	1	High-z	Data Out
Data Register Write	1	0	0	1	0	Odd-Byte	Even-Byte
Data Register Read	1	0	0	0	1	Odd-Byte	Even-Byte
Control Register Write	0	1	6h	1	0	xxx	Data In
Alt Status Read	0	1	6h	0	1	High-z	Data out
Invalid Mode	0	0	x	x	x	High-z	High-z

33.4.9 Card Extraction

When a card is extracted, the PCMCIA/CF controller's registers are not reset. Instead, the register settings remain the same as before the card's extraction. This allows the host software to quickly activate a card once the CIS indicates that it's the same card.

33.5 Timing Diagrams

Timing diagrams showing typical PCMCIA/CF controller accesses are shown in Figure 33-2 and Figure 33-3

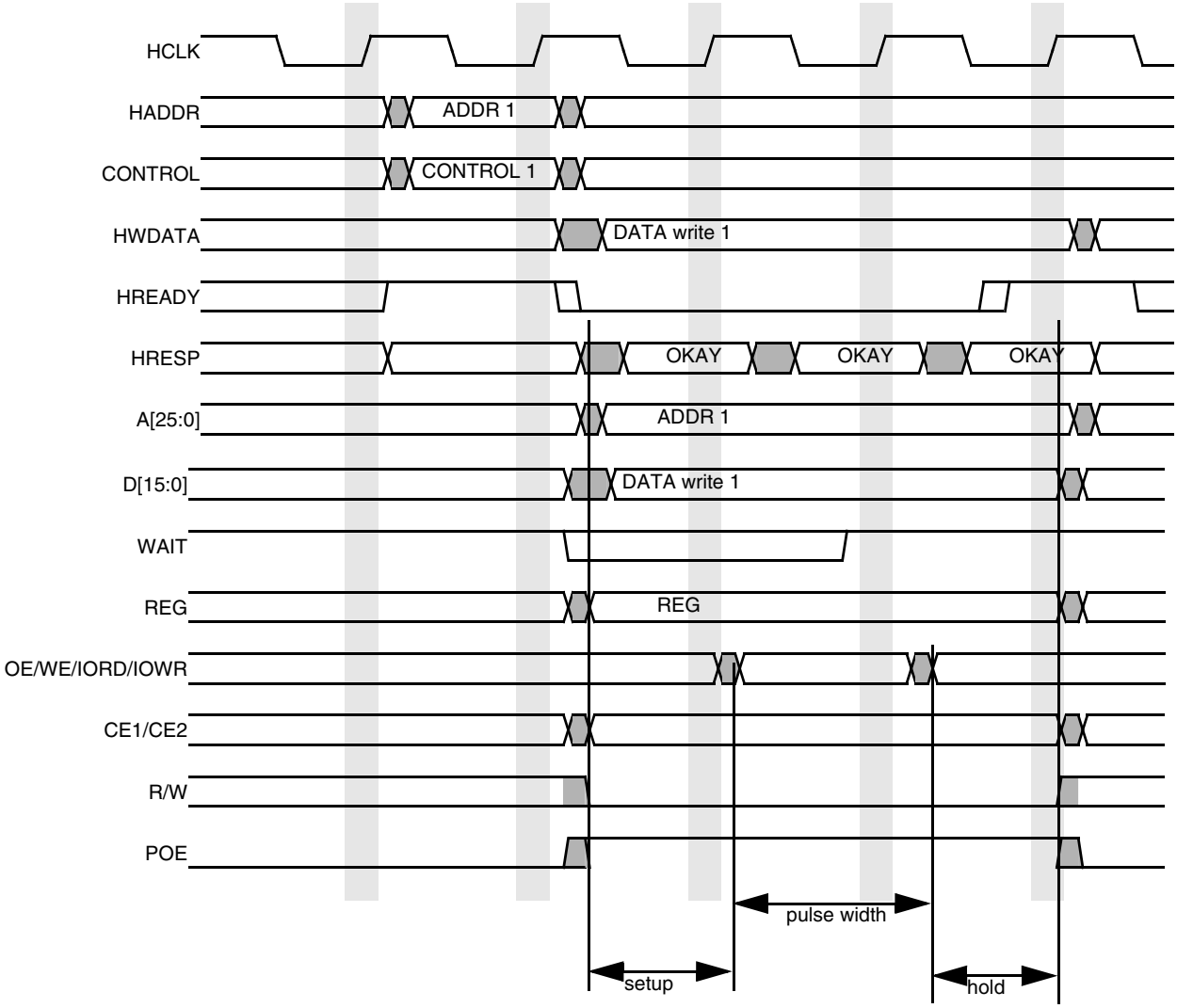


Figure 33-2. Write Accesses PSHT=1, PSST =1

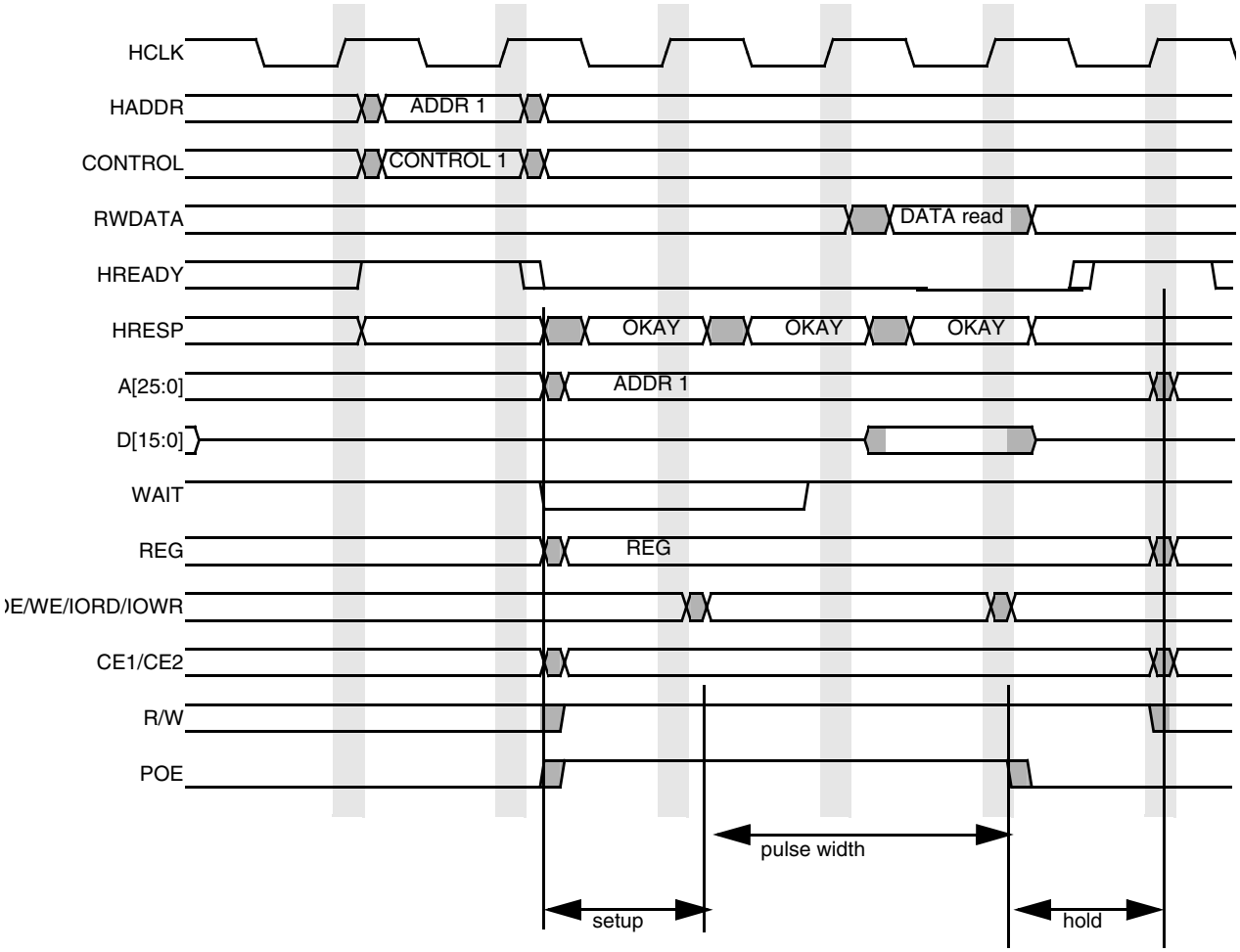


Figure 33-3. Read Cycle PSHT=1, PSST =1

Chapter 34
Keypad Port (KPP)

The Keypad Port is a 16-bit peripheral which can be used either for keypad matrix scanning or as general purpose I/O. The block diagram of the KPP is shown in Figure 34-1.

The Keypad Port features include the following:

- Supports up to 8 × 8 external key pad matrix
- Port pins can be used as general purpose I/O
- Open drain design
- Glitch suppression circuit design
- Multiple keys detection
- Long key press detection
- Standby key press detection
- Synchronizer chain clear

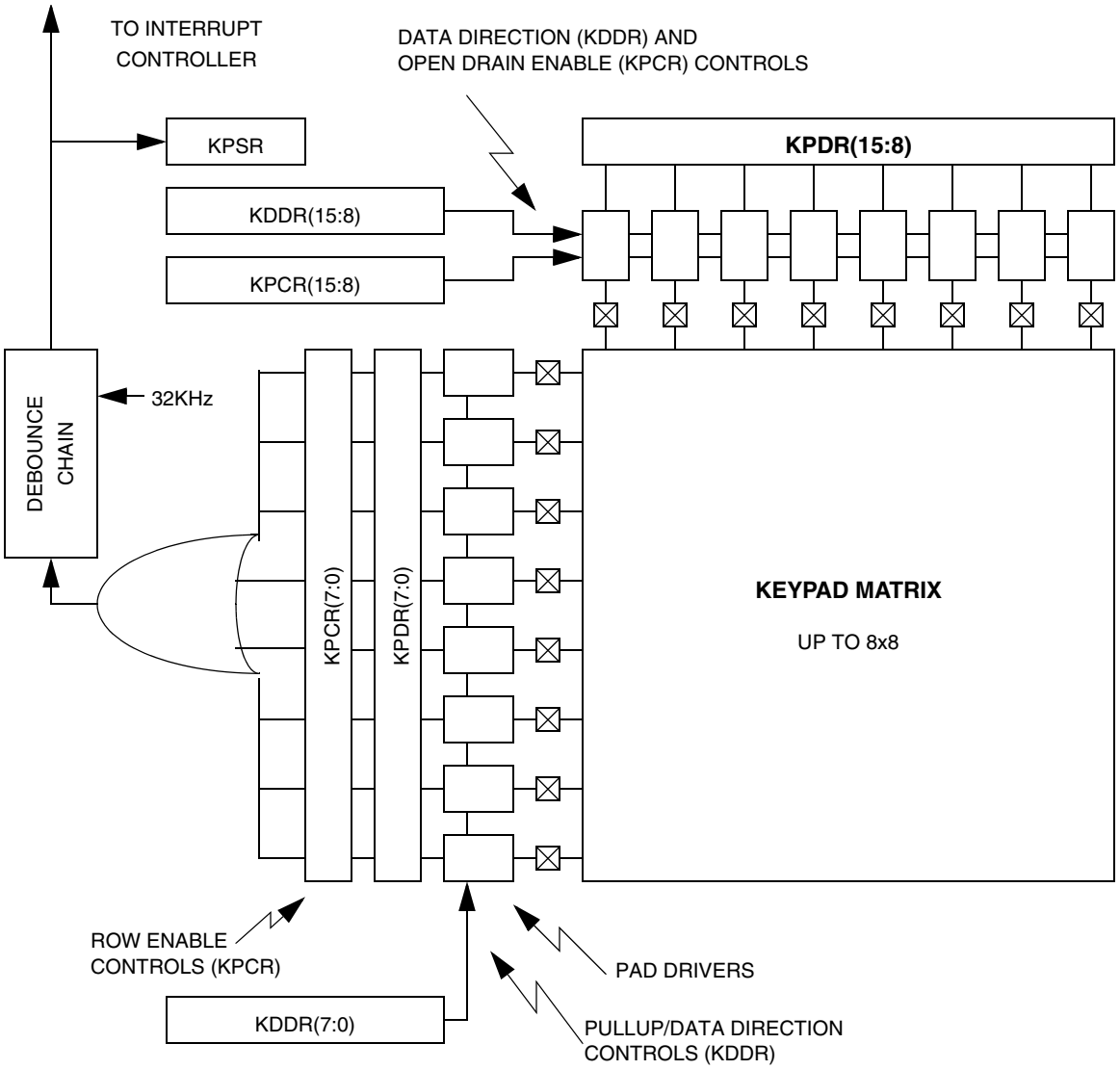


Figure 34-1. KPP Peripheral Block Diagram

34.1 KPP Peripheral Pin Description

Sixteen pins are dedicated to the KPP. Keypads of any configuration up to 8 rows and 8 columns are supported through software configuration of the peripheral pins. Any pins not used for the keypad are available as general purpose input/output. The registers are configured such that the pins can be treated as an I/O port up to 16 bits wide.

34.1.1 Input Pins

Any of the 16 pins associated with the KPP can be configured as inputs by writing a 0 to the appropriate bits in the KDDR. Additionally, the least significant 8 bits (ROW inputs) corresponding to KDDR7:0 have internal pullups which are enabled when the pin is used as an input.

34.1.2 Output Pins

Any of the 16 pins associated with the KPP can be configured as outputs by writing the appropriate bits in the KDDR to 1. Additionally, the 8 most significant bits (15–8) can be designated as open drain outputs by writing a 1 into the appropriate bits in KPCR. The lower 8 bits (7–0) are always totem pole style drive when configured as outputs. See Table 34-1.

Table 34-1. Keypad Port Column Modes

KDDR (15:8)	KPCR (15:8)	Pin Function
0	x	Input
1	0	Totem-Pole Output
1	1	Open-Drain Output

34.1.3 Generation of Transfer Error Signal on IP bus

The KPP Module asserts a transfer error signal (ips_xfr_err) on IP bus in following case:

“On getting an IP access to an address which is not implemented”

Errors on un-implemented address are generated only if input pin resp_sel is low; otherwise, a transfer error is not generated.

34.2 Programming Model

This section provides the detailed descriptions of the Keypad Port registers. Table 34-2 provides the Keypad Port register summary.

Table 34-2. Keypad Port Register Summary

Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KPCR (0x10008000)	R	KCO7	KCO6	KCO5	KCO4	KCO3	KCO2	KCO1	KCO0	KRE7	KRE6	KRE5	KRE4	KRE3	KRE2	KRE1	KRE0
	W																
KPSR (0x100080002	R	0	0	0	0	0		KRIE	KDIE	0	0	0	0	0	0	KPKR	KPKD
	W						KPP_EN							KRSS	KDSC		
KDDR (0x10008004)	R	KCDD 7	KCDD 6	KCDD 5	KCDD 4	KCDD 3	KCDD 2	KCDD 1	KCDD 0	KRDD 7	KRDD 6	KRDD 5	KRDD 4	KRDD 3	KRDD 2	KRDD 1	KRDD 0
	W																
KPDR (0x100080006)	R	KCD7	KCD6	KCD5	KCD4	KCD3	KCD2	KCD1	KCD0	KRD7	KRD6	KRD5	KRD4	KRD3	KRD2	KRD1	KRD0
	W																

34.2.1 Keypad Control Register (KPCR)

The Keypad Control Register determines which of the eight possible column strobes are to be open drain when configured as outputs and which of the eight row sense lines are considered in generating an interrupt to the core.

It is up to the programmer to ensure that pins being used for functions other than the keypad are properly disabled. The KPCR register is byte or half word addressable.

KPCR — Keypad Control Register — Addr 0x10008000

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCO7	KCO6	KCO5	KCO4	KCO3	KCO2	KCO1	KCO0	KRE7	KRE6	KRE5	KRE4	KRE3	KRE2	KRE1	KRE0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 34-3. KPCR Register Description

Name	Description	Settings
KCO Bits 15–8	**Keypad Column Strobe Open-Drain Enable**— Setting a column open-drain enable bit (KCO7–KCO0) disables the Pullup driver on that pin. Clearing the bit allows the pin to drive to the high state. This bit has no effect when the pin is configured as an input.	0 = Column strobe output is totem-pole drive (P-Channel enabled). 1 = Column strobe output is open drain (P-Channel disabled).
KRE Bits 7–0	**Keypad Row Enable**—Setting a row enable control bit in this register enables the corresponding row line to participate in interrupt generation. Likewise, clearing a bit disables that row from being used to generate an interrupt. This register is cleared by reset, disabling all rows. The row enable logic is independent of the programmed direction of the pin. Writing a 0 to the data register of pins configured as outputs will cause a keypad interrupt to be generated if the row enable associated with that bit is set.	0 = Row is not included in keypad key press detect. 1 = Row is included in keypad key press detect.

34.2.2 Keypad Status Register (KPSR)

The Keypad Status Register reflects the state of the keypress detect circuit. The KPSR register is byte or half word addressable.

KPSR — Keypad Status Register — Addr 0x10008002

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						KPP_EN	KRIE	KDIE					KRSS	KDSC	KPKR	KPKD
TYPE	r	r	r	r	r	rw	rw	rw	r	r	r	r	w	w	w1c	w1c
RESET						1	0	0							0	0

Table 34-4. KPSR Register Description

Name	Description	Settings
Reserved Bits 15–11	Reserved—These bits are reserved and should read 0.	
KPP_EN Bit 10	**Keypad Clock Gating Enable**—The keypad clock gating enable bit is set when the “ipg_clk” to the module is to be enabled. Software decides to turn off module clock bit. However, the functionality of clock gating in i.MX21 is controlled by CRM, and not by KPP_EN bit in this module.	0 = Disable clock to keypad module 1 = Enable clock to keypad module
KRIE Bit 9	**Keypad Release Interrupt Enable**—Software should ensure that the interrupt for a Key Release event is masked until it has entered the Key Pressed state (and vice-versa) unless this activity is desired, as might be the case when a repeated interrupt is to be generated. The synchronizer chains are capable of being initialized to detect repeated key presses or releases. If they are not initialized when the corresponding event flag is cleared, false interrupts may be generated for depress (or release) events shorter than the length of the corresponding chain.	0 = No interrupt request is generated when KPKR is set 1 = An interrupt request is generated when KPKRis set

Table 34-4. KPSR Register Description (continued)

Name	Description	Settings
KDIE Bit 8	**Keypad Key Depress Interrupt Enable**—Software should ensure that the interrupt for a Key Release event is masked until it has entered the Key Pressed state (and vice-versa) unless this activity is desired, as might be the case when a repeated interrupt is to be generated. The synchronizer chains are capable of being initialized to detect repeated key presses or releases. If they are not initialized when the corresponding event flag is cleared, false interrupts may be generated for depress (or release) events shorter than the length of the corresponding chain.	0 = No interrupt request is generated when KPKD is set 1 = An interrupt request is generated when KPKD is set
Reserved Bits 7–4	Reserved—These bits are reserved and should read 0.	
KRSS Bit 3	**Key Depress Synchronizer Set**—Self Clear bit. The Key release synchronizer is set by writing a logic one into this bit. Reads return a value of 0.	0 = No effect 1 = Set bits which sets keypad release synchronizer chain
KDSC Bit 2	**Key Depress Synchronizer Clear**—Self Clear bit. The Key depress synchronizer is cleared by writing a logic one into this bit. Reads return a value of 0.	0 = No effect 1 = Set bits which clear the keypad depress synchronizer chain
KPKR Bit 1	**Keypad Key Release**—The keypad key release (KPKR) status bit is set when all enabled rows are detected high after synchronization. (The KPKR status bit will be set when cleared by reset).The bit will typically not be set again until the detect circuit senses a key depressed followed by all keys released. The KPKR bit may be used to generate a maskable key release interrupt. The key release synchronizer may be set high by software after scanning the keypad to ensure a known state. Due to the logic function of the release and depress synchronizer chains, it is possible to see the re-assertion of a status flag (KPKD or KPKR) if it is cleared by software prior to the system exiting the state it represents. Reset value of register is “0” as long as $\overline{\text{HARD\_ASYNC\_RESET}}$ is asserted. However when $\overline{\text{HARD\_ASYNC\_RESET}}$ is deasserted, value of register depends upon the external row pins and can become 1.	0 = No key release detected 1 = All keys have been released
KPKD Bit 0	**Keypad Key Depress**—The keypad key depress (KPKD) status bit is set when one or more enabled rows are detected low after synchronization. The KPKD status bit remains set until cleared by software. The KPKD bit may be used to generate a maskable key depress interrupt. If desired, software may clear the key press synchronizer chain to allow a repeated interrupt to be generated while a key remains pressed. In this case, a new interrupt will be generated after the synchronizer delay elapses if a key remains pressed. Due to the logic function of the release and depress synchronizer chains, it is possible to see the re-assertion of a status flag (KPKD or KPKR) if it is cleared by software prior to the system exiting the state it represents.	0 = No key presses detected 1 = A key has been depressed

34.2.3 Keypad Data Direction Register (KDDR)

The bits in the KDDR control the direction of the keypad port pins. The upper eight bits in the register affect the pins designated as column strobes, while the lower eight bits affect the row sense pins. Setting any bit in this register configures the corresponding pin as an output. Clearing any bit in this register configures the corresponding port pin as an input. For bits 7:0, an internal pullup is enabled if the corresponding bit is clear. This register is cleared by reset, configuring all pins as inputs. The KDDR register is byte or halfword addressable.

KDDR	Keypad Data Direction Register												Addr 0x10008004			
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCDD7	KCDD6	KCDD5	KCDD4	KCDD3	KCDD2	KCDD1	KCDD0	KRDD7	KRDD6	KRDD5	KRDD4	KRDD3	KRDD2	KRDD1	KRDD0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 34-5. KDDR Register Description

Name	Description	Settings
KCDD Bits 15–8	**Keypad Column Data Direction**—Setting any bit configures the corresponding pin as an output.	0 = COLn[1] pin is configured as input 1 = COLn pin is configured as output
KRDD Bits 7–0	**Keypad Row Data Direction**—Setting any bit configures the corresponding pin as an output.	0 = ROWn pin is configured as input 1 = ROWn pin is configured as output

[1] n = 7-0

34.2.4 Keypad Data Register (KPDR)

This 16-bit register is used to access the column and row data. Data written to this register is stored in an internal latch, and for each pin configured as an output, the stored data is driven onto the pin. A read of this register returns the value on the pin for those bits configured as inputs. Otherwise, the value read is the value stored in the register.

The KPDR register is byte or halfword addressable. This register is not initialized by reset. Valid data should be written to this register before any bits are configured as outputs.

KPDR	Keypad Data Register												Addr 0x10008006			
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCD7	KCD6	KCD5	KCD4	KCD3	KCD2	KCD1	KCD0	KRD7	KRD6	KRD5	KRD4	KRD3	KRD2	KRD1	KRD0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Table 34-6. KPDR Register Description

Name	Description
KCD Bits 15–8	**Keypad Column Data**—A read of these bits returns the value on the pin for those bits configured as inputs. Otherwise, the value read is the value stored in the register.
KRD Bits 7–0	**Keypad Row Data**—A read of these bits returns the value on the pin for those bits configured as inputs. Otherwise, the value read is the value stored in the register.

34.3 Keypad Operation

The Keypad Port is designed to simplify the software task of scanning a keypad matrix. With appropriate software support, the KPP is capable of detecting, debouncing and decoding one or two keys pressed simultaneously in the keypad.

Logic in the KPP is capable of detecting a key press even while the processor is in one of the low power standby modes—that is, even when there is no ipg_clk. The KPP may generate an interrupt to the ARM9 platform any time a key press or key release is detected. This interrupt is capable of forcing the processor out of a low power mode.

34.3.1 Keypad Matrix Construction

The Keypad Port is designed to interface to a keypad matrix which shorts the intersecting row and column lines together whenever a key is depressed. The interface is not optimized for any other switch configuration.

34.3.2 Keypad Port Configuration

Software must initialize the Keypad Port for the size of the keypad matrix. Pins connected to the keypad columns should be configured as open-drain outputs. Pins connected to the keypad rows should be configured as inputs. On-chip pullup resistors are implemented for active keypad rows. Besides enabling of row inputs in Keypad control register, corresponding interrupt (depress or/and release) must also be enable to generate an interrupt. Discrete switches which are not part of the matrix may be connected to any unused row inputs. The second terminal of the discrete switch is connected to ground. The hardware will detect closures of these switches without the need for software polling.

34.3.3 Keypad Matrix Scanning

Keypad scanning is performed by a software loop which walks a zero across each of the keypad columns, reading the value on the rows at each step. The process is repeated several times in succession, with the results of each pass optionally compared with those from the previous pass. When several (3 or 4) consecutive scans yield the same key closures, a valid key press has been detected. Software then can decode exactly which switch was depressed and pass the value up to the next higher software layer.

The basic debouncing period, which must be defined in the software routine, may be controlled with an internal timer. The basic period is the period between the scan of two consecutive columns, so the debounce time between two consecutive scans of the whole matrix shall be the number of columns multiplied by the basic period.

34.3.4 Keypad Standby

There is no need for the ARM9 platform to continually scan the keypad. Between key presses, the keypad can be left in a state which requires no software intervention until the next key press is detected. To place the keypad in a standby state, software should write all column outputs low. Row inputs are left enabled. At this point the ARM9 platform can attend to other tasks or revert to a low power standby mode. The Keypad Port will interrupt the ARM9 platform if any key is pressed.

Upon receiving a keypad interrupt, the ARM9 platform should set all the column strobes high, and begin a normal keypad scanning routine to determine which key was pressed. It is important that open-drain drivers be used when scanning to prevent a possible DC path between power and ground through two or more switches.

34.3.5 Glitch Suppression on Keypad Inputs

A glitch suppression circuit qualifies the keypad inputs to prevent noise from inadvertently interrupting the ARM9 platform. The circuit is a 4 state synchronizer clocked from a 32KHz clock source. This clock must continue to run in any low power mode where the keypad is a wake-up source, as the ARM9 platform interrupt is generated from the synchronized input. An interrupt is not generated until all 4 synchronizer stages have latched a valid key assertion. This guarantees of filtering out any noise less than 3 clock periods (for 32KHz clock: 93.75 uS) in duration. Noise filtering of duration between 3 to 4 clock periods (for 32KHz clock: between 93.75us and 125us) can not be guaranteed. The interrupt output is latched in an S-R latch and remains asserted until cleared by software. The Set input of the latch is rising-edge clocked.

34.3.6 Multiple Key Closures

Using the Key press and Key release interrupts the software can detect multiple keys or achieve N key rollover. The key scanning routine can be programmed accordingly.

The KPP module only detects if one or multiple keys are pressed or released. When a simple keypad matrix is used, there is a chance of *ghost* key detection when three or more keys are pressed. As shown in Figure 34-2, three keys pressed simultaneously can short between the column currently *scanned* by software and another column. Depending on the location of the third key pressed, a ghost key press may be detected. However, this can be corrected by using a keypad matrix which provides ghost key protection. Such a matrix implements a one way *diode* at all keypad points between rows and columns. This way the multiple pressing of 3 keys will not cause a short at a 4th key (see Figure 34-3).

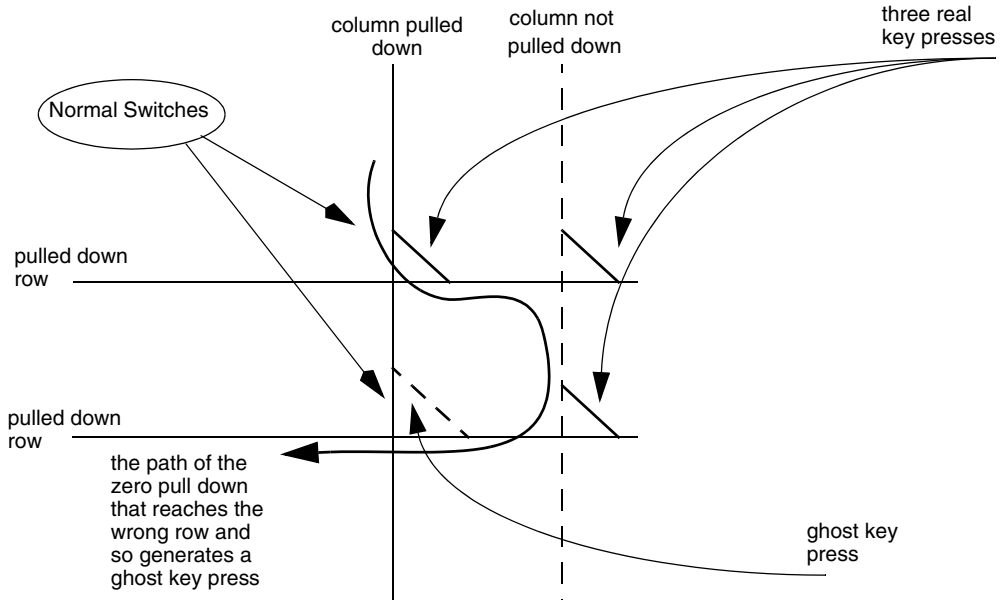


Figure 34-2. Decoding Wrong Three- Key-Presses

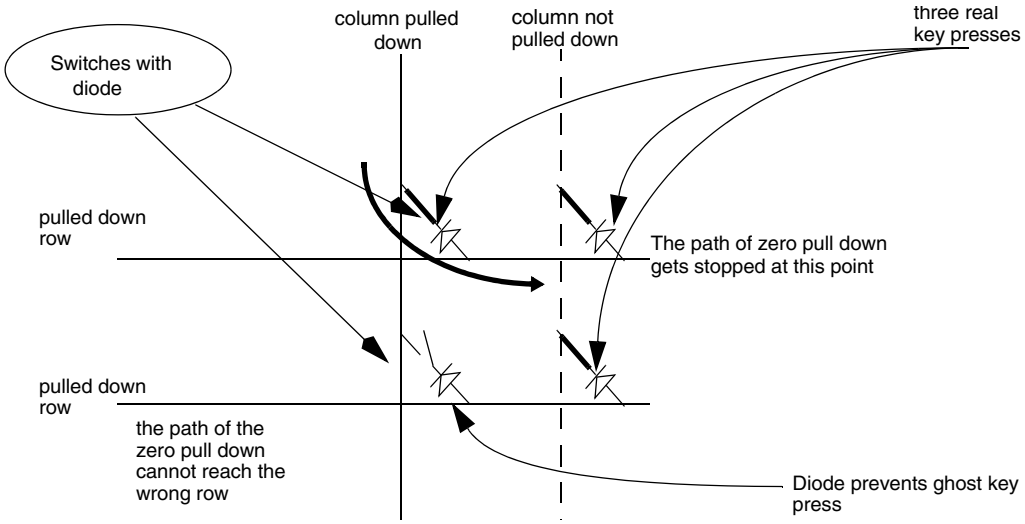


Figure 34-3. Matrix with "Ghost" Key Protections

34.3.7 Typical Keypad Configuration and Scanning Sequence

Configure Keypad:

1. Enable number of rows in keypad (KPCR[7:0])
2. Write 0's to KPDR[15:8]
3. Configure keypad columns as open-drain (KPCR[15:8])
4. Configure columns as output, rows as input (KDDR[15:0])
5. Clear the KPKD Status Flag and Synchronizer chain
6. Set the KDIE control bit, clear the KRIE control bit (avoid false release events) (now in standby mode, awaiting a key press…)

Key press Interrupt Detected

Begin Keypad Scanning Routine:

1. Disable both (depress and release) keypad interrupts
2. Write 1’s to KPDR[15:8] setting column data to 1’s
3. Configure columns as totem pole outputs (for quick discharging of keypad capacitance)
4. Configure columns as open-drain
5. Write a single column to 0, others to 1
6. Sample row inputs and save data. Multiple key presses can be detected on a single column.
7. Repeat steps 2–6 for remaining columns
8. Return all columns to 0 in preparation for standby mode
9. Clear KPKD and KPKR status bit(s) by writing to a 1 set the KPKR synchronizer chain by writing 1 to KRSS register, clear the KPKD synchronizer chain by writing 1 to KDSC register
10. Set the KPKR synchronizer chain
11. Clear the KPKD synchronizer chain
12. Re-enable the appropriate keypad interrupt(s)
13. KDIE to detect a key hold condition
14. or KRIE to detect a key release event

Chapter 35
Fast InfraRed Interface (FIRI) Module

This section describes Fast InfraRed Interface module (FIRI), which is integrated in the i.MX21 It is capable of establishing a 0.576 Mbit/sec, 1.152 Mbit/sec or 4 Mbit/sec half duplex link via a LED and IR detector. It supports 0.576 Mbit/sec, 1.152 Mbit/sec Medium InfraRed (MIR) physical layer protocol and 4Mbit/sec Fast InfraRed (FIR) physical layer protocol defined by IrDA, version 1.4. In addition, i.MX21 supports the Serial InfraRed (SIR) protocol via the UART modules. The LED, IR detector and UART pins are shared for both modules and can be controlled via GPIO, as shown in Figure 35-1.

The FIRI includes these distinctive features:

- Supports 0.576 Mbit/sec, MIR protocol
- Supports 1.152 Mbit/sec MIR protocol
- Supports 4 Mbit/sec, 4PPM, FIR protocol
- Device Destination Detection Hardware Support
- Interrupt generation
- DMA capability
- SIP generation for collision avoidance

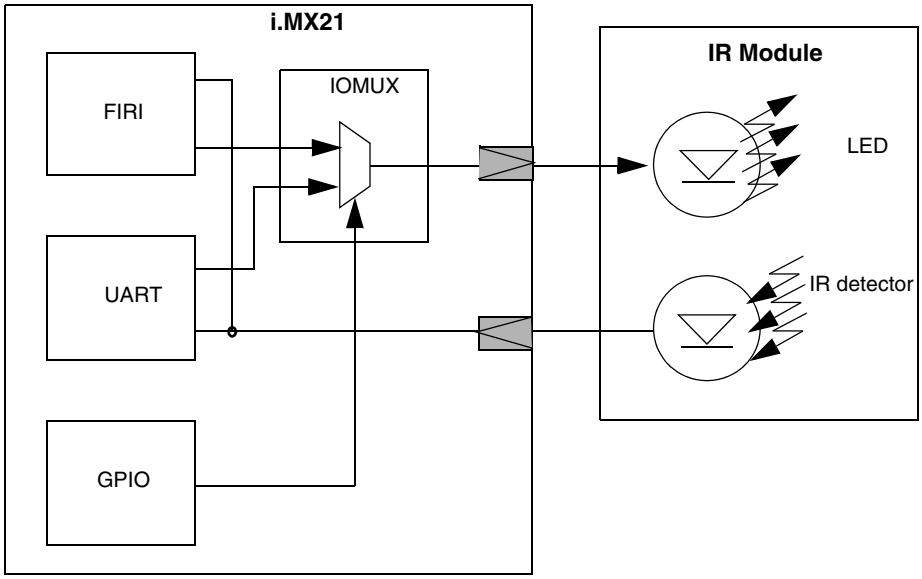


Figure 35-1. FIRI Integration Diagram

35.1 Modes of Operation

The FIRI supports the following modes:

- Hardware packet assembly
  - FIR mode
  - 0.576 Mbps MIR mode
  - 1.152 Mbps MIR mode
  - Serial Infrared Interaction Pulse (SIP) generation
- Hardware packet search
  - FIR mode
  - 0.576 Mbps MIR mode
  - 1.152 Mbps MIR mode
- Software packet assembling
- Software packet search

35.2 Overview

The FIRI is divided to the following functional parts:

- Packet Assembler
- Searcher
- MIR Modulator
- Demodulator (supports MIR and FIR)
- CRC Encoders (supports MIR and FIR)
- FIFO and FIFO Controller
- IP Interface and IP Registers

The FIRI block diagram is shown in Figure 35-2.

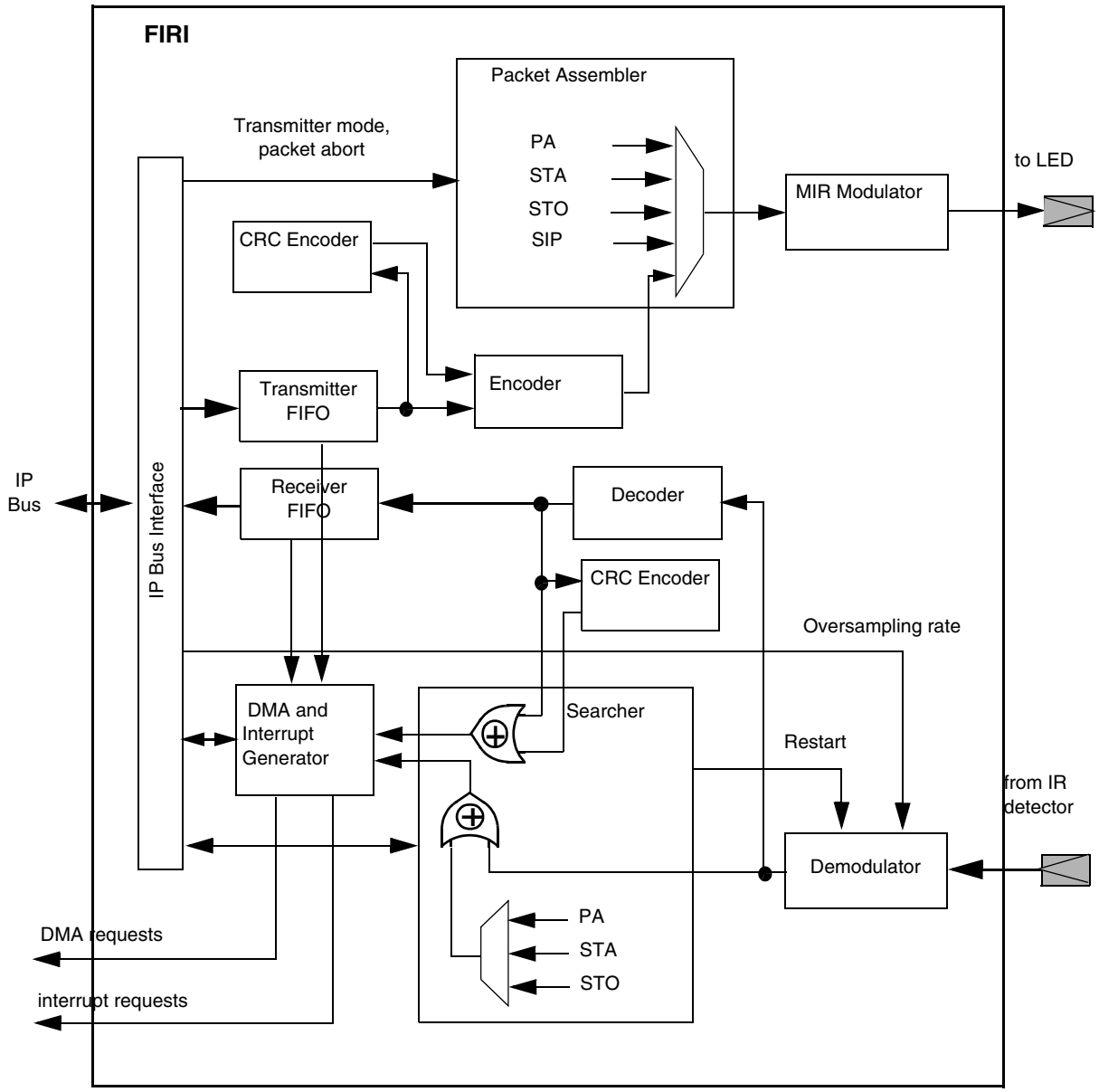


Figure 35-2. FIRI Block Diagram

35.3 IrDA Standards Overview

The following sections provide an overview of the IrDA standards and the operation of the FIRI module.

35.3.1 IrDA Medium InfraRed and Fast InfraRed Standards

The FIRI supports MIR (0.576 Mbit/sec, 1.152 Mbit.sec) and FIR (4 Mbit/sec) physical layer protocols. This section provides a brief overview of the MIR and FIR standards.

35.3.1.1 MIR Packet Structure

The data of MIR physical layer organized into packets:

Table 35-1. MIR Packet Structure

STA	STA	DD (Address & Control & Data)	CRC	STO

where STA and STO are equal predefined sequence (left symbol transmitted/received first):

STA, STO: b'0111,1110

DD: can be up to 2048 bytes long, address (8 bits) and control are optional

CRC: generated according to CRC-CCITT algorithm

The DD and CRC fields are not transmitted as-is, they are first converted according to the MIR standard, as described in the next sections.

35.3.1.2 FIR Packet Structure

The data of FIR physical layer organized into packets:

Table 35-2. FIR Packet Structure

PA	STA	DD (Address & Control & Data)	CRC32	STO

where PA, STA and STO are predefined sequence (left symbol transmitted/received first):

PA: b'1000,0000,1010,1000 repeated 16 times

STA: b'0000,1100,0000,1100,0110,0000,0110,0000

STO: b'0000,1100,0000,1100,0000,0110,0000,0110

DD: can be up to 2048 bytes long, address (8 bits) and control are optional

CRC32: generated according to IEEE 802 CRC32 algorithm

The DD and CRC32 fields are not transmitted as-is, they are first modulated with the 4 Pulse Position Modulation, described in next sections.

35.3.1.3 MIR CRC

The CRC field is 16 bits long and generated according to CRC-CCITT algorithm. The CRC polynomial is defined as follows:

$$CRC(x) = x^{16} + x^{12} + x^5 + 1 \qquad \textit{Eqn. 35-1}$$

The CRC(x) value is inverted prior transmission.

35.3.1.4 FIR CRC

The CRC32 field is 32 bits long and generated according to IEEE 802 CRC32 algorithm. The CRC32 polynomial is defined as follows:

Eqn. 35-2

$$CRC32(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

The CRC(x) value is inverted prior transmission.

35.3.1.5 MIR Modulation

The transmitter sends out 1/4-bit-length pulse when “zero” must be transmitted. When “one” must be transmitted the LED is OFF. Tolerance of bit rate according to the standard must not exceed ± 0.1%.

35.3.1.6 FIR Modulation

The 4PPM encoding is achieved by mapping data stream into data symbols as shown in Table 35-3.

Table 35-3. 4PPM Mapping

Data Bit Pair	4PPM Data Symbol (DD)
00	1000
01	0100
10	0010
11	0001

Each bit inside a symbol denotes a “chip”. When zero must be transmitted the LED is OFF, when one must be transmitted the LED is ON. Tolerance of chip rate according to the standard should not exceed ±0.01%. Pulse width tolerance of electrical signal driven by IR detector (Photo Diode) can be greater than the tolerance of the optical signal defined by IrDA and can depend from LED and IR devices used with the FIRI.

35.3.2 Transmitter Overview

The transmitter has three modes of operation.

35.3.2.1 MIR Mode

In MIR mode the Packet Assembler block send the predefined STA sequence (2 or more times) to MIR Modulator block. Then the Encoder block is involved. The Encoder block accesses the FIFO, aligns the data, and sends DD (Address, Control and Data) to the Packet Assembler block. The CRC field calculated by the CRC block follows the DD field. The Encoder block inserts a zero bit after five consecutive 1’s for DD and CRC fields. After DD and CRC fields the STO bits are sent to the Modulator block by Packet Assembler block. The MIR Modulator block converts the bit stream to 1/4-bit-length pulses and sends them to the LED.

35.3.2.2 FIR Mode

In FIR mode, the MIR Modulator block is disabled. The Packet Assembler block sends the predefined PA (16 or more times) and STA sequences to the LED. Then the Encoder block accesses the FIFO, encodes

the data (4PPM), and sends DD (Address, Control and Data) to the Packet Assembler block. The CRC32 field calculated by CRC block follows the DD field after which the STO bits are sent to the Modulator.

If the DMA request is not served during MIR or FIR mode and there is no valid data to transmit, the Assembler Module terminates the packet using the packet abort symbol or with CRC/CRC32 and STO fields (see also PCF bit description) and TFUI interrupt is generated. The packet can also be aborted by the packet length counter or by software (see description of PC bit).

35.3.2.3 Serial Infrared Interaction Pulse

At least each 500 ms the transmitter must send the “Serial Infrared Interaction Pulse” (SIP) in order to guarantee that low speed IR devices will not interfere. The pulse must be 8.7 usec wide, positive phase must be 1.6 usec wide.

35.3.2.4 Software Packet Assembly Mode

In Software packet assembly mode, the entire packet is read from the FIFO, including PA, STA and STO bits. The MIR modulator and Encoder blocks are disabled. This mode is used for debug purposes and can be used for testing compatibility with future IrDA protocols, if appropriate.

35.3.3 Receiver Overview

The receiver has three modes of operation.

In MIR and FIR modes each incoming pulse is oversampled by the programable factor in the Demodulator block. Next, the Demodulator block detects the edges of the pulses and synchronizes to the phase of the transmitted pulse. Phase correction is done until the end of the packet because the bit rate tolerance accumulates to very large values for long packets.

35.3.3.1 MIR Mode

In MIR mode the Searcher block searches for the STA field. If the sequence is matched, the data stream following the STA field is sent to the Decoder block. The Decoder block searches for five consecutive 1’s and skips next the “zero” bit inserted by transceiver for phase synchronization. The data is then written to the FIFO. In parallel the Searcher Module searches for the STO field and when the STO field is detected a PEI interrupt is generated together with DMA request and the corresponding flag is set in the status register. The searcher module continuously searches for illegal symbols (seven or more consecutive “ones”). If an illegal symbol is detected, the PAI interrupt is generated and the corresponding flag is set in the status register.

35.3.3.2 FIR Mode

In FIR mode, the Searcher Module searches for the PA field and then for the STA field. If found, the “chip” stream is converted to a data stream by the Decoder block, as it shown in Table 35-3, and then it is written to the FIFO. In a parallel operation the Searcher Module searches for the STO field. When the STO field is detected a PEI interrupt can be generated together with DMA request. While receiving the searcher module continuously searches the PA, STA, DD, CRC32, STO fields for illegal symbols. In addition it

searches for a packet abort symbol. On detection of an illegal symbol, the PAI interrupt is generated and the corresponding flag is set in status register.

In MIR and FIR modes the “Address” bits can be compared to a predefined value or/and to broadcast value 0xFF (see description of RAM bits). If CRC field is not match an expected value calculated by CRC Encoder block a PAI interrupt is generated. The CRC field will be send to the FIFO with no respect to comparison result.

35.3.3.3 Software Packet Disassembly Mode

In “Software packet disassembly” mode the entire packet is send to the FIFO, even if illegal symbols have been detected.

35.3.4 FIFO

In the FIRI module 128-byte depth FIFO are used for transmitter and receiver. The FIFOs operate independently. There are pre-programmed FIFO fill levels which trigger DMA requests. The DMA requests deasserted by the FIFO full and FIFO empty events for transmitter and receiver respectively. If the DMA request was not serviced and, as a result, there is no valid data to transmit or there is no room in the FIFO for received data, the corresponding flags is set in FIRISR. The RFOI and TFUI interrupts are generated (if enabled). The FIFO pointers can only be read software. The content of transmitter or receiver FIFO is lost if the TE or RE bit is cleared. The corresponding FIFO pointers and status bits are cleared as well.

35.4 External Signal Description

Table 35-4. External Signal Description

Signal Name	Description	GPIO
IR_TXD	Data Transmit signal. Active High	Ain of PE8
IR_RXD	Data Receive signal. Active high	Aout of PE9

35.5 Programming Model

Table 35-5 summarizes the memory and register addresses in the FIRI module using the following Base Address:

- 0x1002 8000

Table 35-5. FIRI Module Register Summary

Address	Description/Name	Access
base + 28	FIRI Control Register (FIRICR)	R/W
base + 0	FIRI Transmit Control Register (FIRITCR)	R/W
base + 4	FIRI Transmit Count Register (FIRITCTR)	R/W
base + 8	FIRI Receive Control Register (FIRIRCR)	R/W

Table 35-5. FIRI Module Register Summary (continued)

Address	Description/Name	Access
base + 12	FIRI Transmit Status Register (FIRITSR)	R/Write one to clear
base + 16	FIRI Receive Status Register (FIRIRSR)	R/Write one to clear
base + 20	Transmitter FIFO	W
base + 24	Receiver FIFO	R

35.5.1 FIRI Control Register

The Control Register is a 32-bit read-write register which is shared by receiver and transmitter.

FIRICR — FIRI Control Register — base + 28

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	BL								OSF			
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 35-6. FIRI Control Register Description

Name	Description	Settings
Reserved Bits 31–12	Reserved—These bits are reserved and should read 0.	
BL Bits 11–5	**Burst Length**—This field contains the number of bytes that are transferred in a DMA burst. Used to prevent FIFO overrun/underrun. DMA request will be released according to this bits. See also description of TDT and RDT bits. For i.MX21 IC the value of BL bits should not exceed 64 due to limitation of FIFO size in DMA module.	1 = 1 byte 2 = 2 bytes 3 = 3 bytes ... 127 = 127 bytes 0 = 128 bytes
Reserved Bit 4	Reserved—This bit should always be written as 0. Writing a ‘1’ can cause incorrect operation of the module.	0 = default
OSF Bits 3–0	**Over Sampling Factor**—This field controls the oversampling factor of the “chip” if RE bit is set and prescaling factor of ipg_clk_48m if TE bit is set. **Note:** Chip rate is 4 and 2 times greater than bit rate for MIR and FIR respectively.	0 = Do not oversample 1 = Oversample by 2 2 = Oversample by 3 ... 15 = Oversample by 16

35.5.2 FIRI Transmitter Control Register

The Transmitter Control Register is a 32-bit read-write register which control transmitter operation.

FIRITCR FIRI Transmitter Control Register base + 0

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	HAG	TPA							
W																
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R		SRF		TDT			TCIE	TPEIE	TFUIE	PCF	PC	0	TPP	TM		TE
W												SIP				
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 35-7. Transmitter Control Register Description

Name	Description	Settings
Reserved Bits 31–25	Reserved—These bits are reserved and should read 0.	
HAG Bit 24	**Hardware Address Generator**—When this bit is set the content of TPA bits is transmitted as packet address. When the bit is cleared the packet address is read from FIFO.	0 = Read packet address from FIFO 1 = Use TPA bits as packet address
TPA Bits 23–16	**Transmit Packet Address**—This field contains the 8-bit Transmit Packet Address. If HAG bit is cleared the TPA bits have no effect.	
Reserved Bit 15	Reserved—This bit is reserved and should read 0.	
SRF Bits 14–13	**Start Field Repeat Factor**—This field contains the number of PA or STA fields transmitted in the beginning of the packet for FIR and MIR mode respectively.	00 = 16 PA or 2 STA fields is transmitted 01 = 32 PA or 4 STA fields is transmitted 10 = 64 PA or 8 STA fields is transmitted 11 = 128 PA or 16 STA fields is transmitted
TDT Bits 12–10	**Transmitter DMA Request Trigger Level**—This field controls the FIFO depth that triggers a DMA request. The Burst Length value (BL) should not exceed the number of vacant bits in the FIFO which is controlled by TDT.	0 = DMA request generated when FIFO is empty 1 = DMA request generated when FIFO contain 16 bytes of data 2 = DMA request generated when FIFO contain 32 bytes of data 3 = DMA request generated when FIFO contain 48 bytes of data ... 7 = DMA request generated when FIFO contain 112 bytes of data
TCIE Bit 9	**Transmit Complete Interrupt Enable**—This bit enables the TC status bits to generate a TCI interrupt.	0 = TCI interrupt is not triggered by TC bit 1 = TCI interrupt is triggered by TC bit

Table 35-7. Transmitter Control Register Description (continued)

Name	Description	Settings
TPEIE Bit 8	**Transmitter Packet End Interrupt Enable**—This bit enables the TPE and SIPE status bits to generate a PEI interrupt.	0 = PEI interrupt is not triggered by TPE or SIPE bits 1 = PEI interrupt is triggered by TPE or SIPE bits
TFUIE Bit 7	**Transmitter FIFO Underrun Interrupt Enable**—This bit enables the TFU status bit to generate a TFU interrupt.	0 = TFU interrupt disabled 1 = TFU interrupt enabled
PCF Bit 6	**Packet Complete by FIFO**—This bit determines how a packet is completed if a FIFO underrun event occurs.	0 = Send CRC and STO fields 1 = Send packet abort symbol
PC Bit 5	**Packet Complete**—This bit determines how a packet is completed when TE bit is cleared or when SIP bit is set in the middle of the transfer.	0 = Send CRC and STO fields 1 = Send packet abort symbol
SIP Bit 4	**SIP Transmit Enable**—Writing 1 to this bit produces a “Serial infrared Interaction Pulse” transmission. Writing zero to this bit is ignored. This bit is always read as zero. If this bit is set while in the middle of the transfer, the packet will be completed according to setting of the PC bit.	
TPP Bit 3	**Transmitter Pulse Polarity**—See settings.	0 = Transmitted pulse is not inverted 1 = Transmitted pulse is inverted
TM Bits 2–1	**Transmitter Mode**—These bits control the transmission mode.	11 = Software Packet Assembling 10 = 1.152 Mbps MIR Mode 01 = 0.576 Mbps MIR Mode 00 = FIR Mode
TE Bit 0	**Transmitter Enable**—This bit controls the FIRI transmitter. If this bit is cleared in the middle of the transfer the packet will be completed according to the setting of the PC bit. When the packet is completed, the transmitter clocks are then gated OFF.	0 = Transmitter Disabled 1 = Transmitter Enabled

35.5.3 FIRI Transmitter Count Register

The Transmitter Count Register is 32-bit read-write register which control packed size.

FIRITCTR — FIRI Transmitter Count Register — base + 4

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R						TPL										
W																
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 35-8. Transmitter Count Register Description

Name	Description	Settings
Reserved Bits 31–11	Reserved—These bits are reserved and should read 0.	
TPL Bits 10–0	**Transmit Packet Length**—The length of DD field.	0 = Send 1 byte 1 = Send 2 bytes ... 2047 = Send 2048 bytes

35.5.4 FIRI Receiver Control Register

The Receiver Control Register is 32-bit read-write register which control receiver operation.

FIRIRCR — FIRI Receiver Control Register — base + 8

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	RAM		RA							
W																
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R		0	0	0	RPEDE	RDT			RPA	RPEIE	PAIE	RFOIE	RPP	RM		RE
W																
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 35-9. Receiver Control Register Description

Name	Description	Settings
Reserved Bits 31–26	Reserved—These bits are reserved and should read 0.	
RAM Bits 25–24	**Address Match**—See settings. The value of this bits can be changed when RE bit is cleared.	00 = Do not match address 01 = Match packet address to RA bits 10 = Match packet address to Broadcast address 11 = Match packet address to RA bits and to broadcast address
RA Bits 23–16	**RA**—Determine Receiver Packet Address. If RAM bits value is 00 or 10 the RA bits have not effect. The value of this bits can be changed when RE bit is cleared.	NA
Reserved Bits 15–12	Reserved—These bits are reserved and should read 0.	

Table 35-9. Receiver Control Register Description (continued)

Name	Description	Settings
RPEDE Bit 11	**Receiver Packet End DMA Request Enable**—This bit enables DMA request generation at end of packet. If this bit is set the BL value should not be grater then DD+CRC size.	1 = DMA request is generated at end of packet 0 = DMA request is not affected by end of packet
RDT Bits 10–8	**Receiver DMA request Trigger Level**—This field sets the minimum FIFO depth which will trigger DMA request. A value of BL bits should not exceed the trigger level selected by RDT bits.	0 = Reserved 1 = DMA request generated when FIFO contain 16 bytes of data 2 = DMA request generated when FIFO contain 32 bytes of data 3 = DMA request generated when FIFO contain 48 bytes of data ... 7 = DMA request generated when FIFO contain 112 bytes of data
RPA Bit 7	**Receiver Packet Abort**—Determines behavior of the FIFO upon detection of an illegal symbol. When an illegal symbol is detected, the DDE or CRCE bit is set and if RPA is set, then the receive FIFO pointers are cleared and the receiver starts to search for PA or STA fields for FIR and MIR mode respectively. If RPA is cleared, the receiver continues to write to the FIFO.	1 = Clear the FIFO upon detection of illegal symbol 0 = Do not clear the FIFO upon detection of illegal symbol
RPEIE Bit 6	**Receiver Packet End Interrupt Enable**—This bit enables the RPE status bits to cause PEI interrupt.	1 = PEI interrupt is triggered by RPE bit 0 = PEI interrupt is not triggered by RPE bit
PAIE Bit 5	**Packet Abort Interrupt Enable**—This bit enables the DDE, CRCE status bits to cause PAI interrupt.	1 = PAI interrupt enabled 0 = PAI interrupt disabled
RFOIE Bit 4	**Receiver FIFO Overrun Interrupt Enable**—This bit enables the RFO status bit to cause RFOI interrupt.	1 = RFOI interrupt enabled 0 = RFOI interrupt disabled
RPP Bit 3	**Receiver Pulse Polarity**—See settings.	1 = Received signal is inverted 0 = Receiver signal is not inverted
RM Bits 2–1	**Receiver Mode**—See settings.	11 = Software Packet Assembling 10 = 1.152 Mbps MIR Mode 01 = 0.576 Mbps MIR Mode 00 = FIR Mode
RE Bit 0	**Receiver Enable**—See settings. When this bit is cleared the receiver clocks are gated OFF.	1 = Receiver Enabled 0 = Receiver Disabled

35.5.5 FIRI Transmit Status Register

The Transmit Status Register is 32-bit register which reflect the status of transmitter and FIFO. It contains read only and write 1 to clear bits.

FIRITSR — FIRI Transmit Status Register — base + 12

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TFP								0	0	0	0	TC	SIPE	TPE	TFU
W													1	1	1	1
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 35-10. Transmit Status Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
TFP Bits 15–8	**Transmitter FIFO Pointer**—This field points to the bottom of transmitter FIFO. Read Only bits.	NA
Reserved Bits 7–4	Reserved—These bits are reserved and should read 0.	
TC Bit 3	**Transmit Complete**—This bit indicates that the transmission is completed (including CRC and STO field) and the transmitter FIFO is empty. This bit is cleared by writing 1.	1 = Transmitting is completed 0 = Transmitting is not completed
SIPE Bit 2	**SIP End**—This bit indicates that “Serial infrared Interaction Pulse” had been transmitted. This bit is cleared by writing 1.	1 = SIP had been transmitted 0 = SIP transmission is not completed
TPE Bit 1	**Transmitter Packet End**—This bit indicates that TPS bytes of data (DD field) had been transmitted. This bit is cleared by writing 1.	1 = DD had been transmitted 0 = DD transmissions is not completed
TFU Bit 0	**Transmitter FIFO Underrun**—This bit indicates that the transmitter attempt to read from FIFO when the FIFO is empty, and there is no valid data to transmit. This bit is cleared by writing 1.	1 = Transmitter FIFO is underrun 0 = No transmitter FIFO underrun

35.5.6 FIRI Receive Status Register

The Receive Status Register is 32-bit register which reflects the status of receiver and FIFO. It contains read only and write 1 to clear bits.

FIRIRSR — FIRI Receive Status Register — base + 16

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RFP								0	0	PAS	RPE	RFO	BAM	CRCE	DDE
W												1	1	1	1	1
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 35-11. Receive Status Register Description

Name	Description	Settings
Reserved Bits 31–16	Reserved—These bits are reserved and should read 0.	
RFP Bits 15–8	**Receiver FIFO Pointer**—This field points to the bottom of receiver FIFO. Read Only bits.	NA
Reserved Bits 7–6	Reserved—These bits are reserved and should read 0.	
PAS Bit 5	**Preamble Search**—This bit indicates that the receiver is searching for PA+STA or STA fields for FIR and MIR modes respectively. Read Only bit.	1 = Receiver searches for PA or STA field 0 = Receiver does not search for PA or STA field
RPE Bit 4	**Receiver Packet End**—This bit indicates that STO field or packet abort symbol (b'0000,000 and b'0000,0000 for MIR and FIR restrictively) is detected. This bit is cleared by writing 1.	1 = STO field is detected 0 = STO was not detected
RFO Bit 3	**Receiver FIFO Overrun**—This bit indicates that the receiver attempt to write to FIFO when the FIFO is full. This bit is cleared by writing 1.	1 = Receiver FIFO is overrun 0 = Receiver FIFO is not overrun
BAM Bit 2	**Broadcast Address Match**—This bit indicates that the address field of the packet match Broadcast Address 0xFF. This bit is cleared by writing 1.	1 = Broadcast packet 0 = No Broadcast
CRCE Bit 1	**CRC Error**—This bit indicates that CRC check failed. This bit is cleared by writing 1.	CRC check failure No CRC check failure
DDE Bit 0	**DD Error**—This bit indicates that an illegal symbol has been detected in DD or CRC32/CRC fields. This bit is cleared by writing 1.	1 = Illegal symbol in DD or CRC field 0 = No illegal symbols in DD or CRC field

35.6 Software Restrictions

1. The value of FIRITCR, FIRITCTR, FIRICR registers, except TDT bits, should not be changed if TE bit is set or if transmission of current packet is not completed.
2. The value of FIRIRCR register, except RDT bits, should not be changed if RE bit is set.
3. The “write one to clear” status bits can be cleared (by software) after a period grater then chip period from a time point when it was set (by hardware). This restriction can be especially actual when the status bit is cleared at the beginning of interrupt routine.
4. The “clear” request of “write one to clear” status bits is accepted after a period grater then 2 chip periods.
5. The “write one to clear” status bits can not be cleared if TE/RE bit is cleared.
6. If TE bit is cleared in the middle of the transfer the transmitter should not be enabled again if the transmission of current packet is not completed.
7. If the RE is cleared by software it can be set after a period grater then 4 chip periods.

Chapter 36
1-Wire Interface (1-Wire®)

The 1-Wire® module is a peripheral device that communicates with the ARM926EJ-S™ core and provides a communication line to a 1 Kbit Add-Only Memory (DS2502). The DS2502 is a 1k-bit 1-Wire® EPROM. The 1-Wire interface is able to send or receive one bit at a time to the DS2502. The required protocol for accessing the DS2502 is defined by Dallas Semiconductor. The DS2502 is used to hold battery characteristic information. This specification describes the 1-Wire function, timing diagrams, port definitions as well as notes on testing the 1-Wire module.

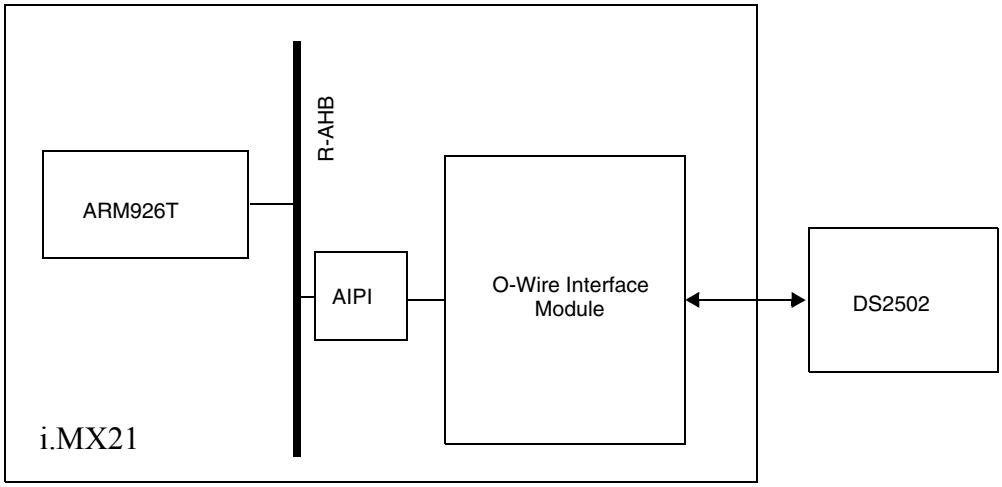


Figure 36-1. 1-Wire Connection

36.1 Peripheral Architecture

A block-level description of the 1-Wire module is contained in Figure 36-1.

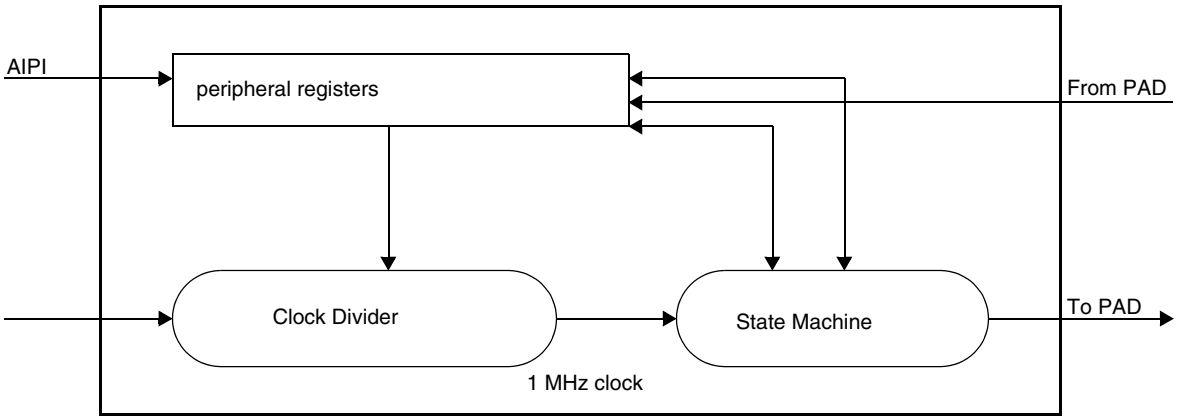


Figure 36-2. 1-Wire Block-Level Description

The clock divider generates a 1 MHz clock used as a time reference by the state machine. Transitions between the states of the state machine as well as actions triggered at precise time deadlines are expressed using this 1 MHz clock. The state machine performs all required actions to dialog with the external device.

36.2 Port Definitions

The inputs and outputs for the 1-Wire are listed in the table below. The DS2502 input and output lines listed in Table 36-1 are the lines that interface with the DS2502.

Table 36-1. 1-Wire Port Definitions: DS2502

Signal	I/O	Comments
BATTERY_LINE_IN	input	1-Wire bus.
BATTERY_LINE_OUT	output	Connected to GND for open drain
OUTPUT_ENABLE	output	Enable for output driver 1-Wire bus. In hdl model, represents DS2502 input

Note: The outputs above have been set for a standard I/O pad.

Note: The DS2502 specifies an external 5K pull-up should be used. The i.MX21 provides a 69K pull-up resistor on the 1-Wire pin. An external pull-up is not required if the 1-Wire module is connected within few inches to the DS2502.

36.3 Pin Configuration

Table 36-2 identifies the pin used for the 1-Wire module. This pin is multiplexed with other functions on the device and must be configured for 1-Wire operation.

Table 36-2. 1-Wire Pin Configuration

Module	Setting	Configuration Procedure
GPIO	Alternate Function of GPIO Port E [16]	1. Clear bit 16 of Port E GPIO In Use Register (GIUS_E) 2. Set bit 16 of Port E General Purpose Register (GPR_E)

36.4 Clock Enable and AIPI Configuration

Table 36-3. CRM and API Register Descriptions

Module	Setting	Configuration Procedure
PLL Clock Controller and Reset Module	CRM_PCCR1	Set bit [31] to enable the clock to 1-Wire
AIPI	AIPI1_PSR0 and AIPI1_PSR1	Set AIPI1_PSR0 bit [9] and Clear AIPI1_PSR1 bit [9] to match 1-Wire bus width 16 bits.

36.5 Functional Description

The 1-Wire interfaces with the 1Kbit Add-only Memory (DS2502) through a simple 1 bit bus. The DS2502 1 Kbit Add-only Memory, manufactured by Dallas Semiconductor, uses the 1-Wire line to program and read a 1024 bit EPROM. The DS2502 also has a 64-bit lasered ROM and status bytes. The DS2502 requires a special protocol to access the EPROM. The protocol involves first issuing one of four ROM

function commands before the EPROM is accessible: read ROM, match ROM, search ROM and skip ROM. Through the 1-Wire bus, the ARM926EJ-S Core interfaces with the DS2502 and allows the required commands to be issued to control the EPROM. The details of the DS2502 required procedures of operation can be found in the referenced document [1]. The ARM926EJ-S (through the 1-Wire interface) is the bus master and the DS2502 device(s) are the slave(s). The 1-Wire peripheral does not trigger interrupts; hence a polling of the 1-Wire module register is necessary to manage a correct operation of the block (refer to the description of Section 36.6).

36.5.1 Low-Power Modes

When the 1-Wire module enters a low power mode it gates off its clock when it is not in use—that is, when the RPP, WR0 and WR1 bits in the Control register are all cleared.

36.5.2 Reset Sequence with Reset Pulse Presence Pulse

To begin any communications with the DS2502, it is required that an initialization procedure be issued. A reset pulse must be generated and then a presence pulse must be detected. The minimum reset pulse length is 480 us. The bus master (1-Wire) will generate this pulse, then after the DS2502 detects a rising edge on the 1-Wire bus, it will wait 15-60 µs before it will transmit back a presence pulse. The presence pulse will exist for 60–240 µs.

The timing diagram for this sequence is shown in Figure 36-3.

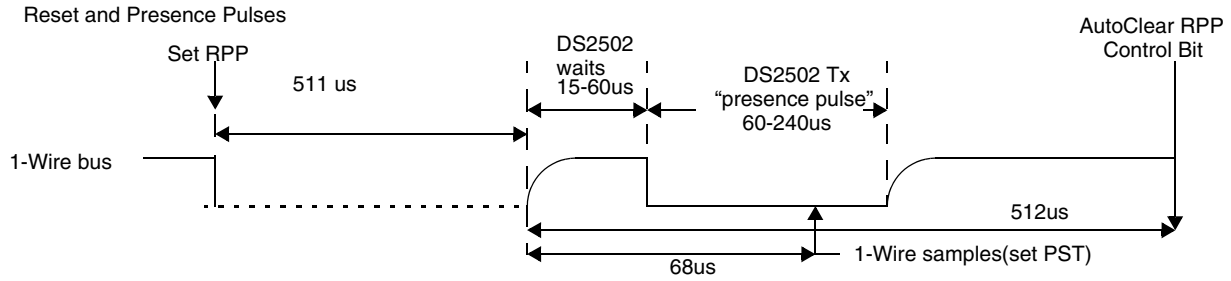


Figure 36-3. 1-Wire Initialization

The reset pulse begins the initialization sequence and it is initiated when the RPP control register bit is set. When the presence pulse is detected, this bit will be cleared. The presence pulse is used by the bus master to determine if at least one DS2502 is connected. Software will determine if more than one DS2502 exists. The 1-Wire module will sample for the DS2502 presence pulse. The presence pulse is latched in the 1-Wire control register PST. When the PST bit is set to a one, it means that a DS2502 is present; if the bit is set to a zero, then no device was found.

36.5.3 Write 0

The Write 0 function simply writes a zero bit to the DS2502. The sequence takes 117 µs. The 1-Wire bus is held low for 100 µs.

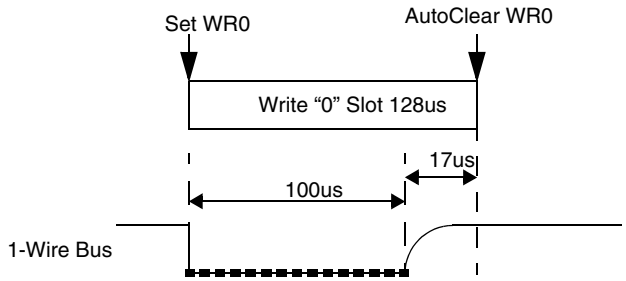


Figure 36-4. Write 0 Timing

The Write 0 pulse sequence is initiated when the WR0 control bit register is set. When the write is complete, the WR0 register will be auto cleared.

36.5.4 Write 1 and Read Data

The Write 1 and Read timing is identical. The time slot is first driven low. According to the DS2502 documentation, the DS2502 has a delay circuit which is used to synchronize the DS2502 with the bus master (1-Wire). This delay circuit is triggered by the falling edge of the data line and is used to decide when the DS2502 will sample the line. In the case of a write 1 or read 1, after a delay, a 1 will be transmitted / received. When a read 0 slot is issued, the delay circuit will hold the data line low to override the 1 generated by the bus master (1-Wire).

For the Write 1 or Read, the control register WR1/RD is set and auto-cleared when the sequence has been completed. After a Read, the control register RDST bit is set to the value of the read.

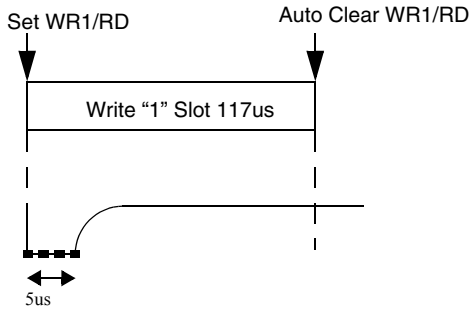


Figure 36-5. Write 1 Timing

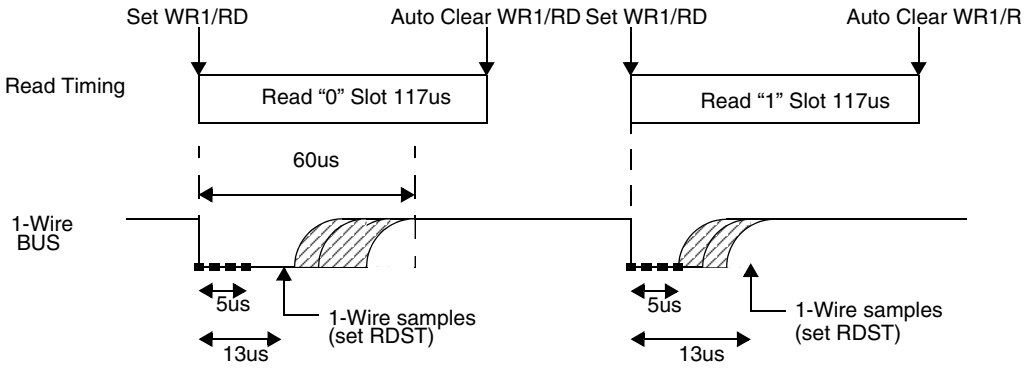


Figure 36-6. Read Timing

36.5.5 Program Pulse

The Program Pulse sequence is described in the DS2502 documentation as one of the functions of the 1-Wire signaling. The 12 volt programming pulse function is not used in the 1-Wire.

36.6 Programming Model

The 1-Wire module includes three user-accessible 16-bit registers. Table 36-4 summarizes these registers and their addresses.

Table 36-4. 1-Wire Module Register Summary

Description	Name	Address
1-Wire Control Register	CONTROL	0x10009000
1-Wire Time Divider Register	TIME_DIVIDER	0x10009002
1-Wire Reset Register	RESET	0x10009004

36.6.1 Control Register

The Control register is used to monitor and control the 1-Wire communications between the i.MX21 and the 1-Wire external device.

CONTROL — Control Register — $1000_9000

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									RPP	PST	WR0	WR1	RDST			
TYPE	r	r	r	r	r	r	r	r	rwm	r	rwm	rwm	r	0	0	0
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 36-5. Control Register Description

Name	Description	Settings
Reserved Bits 15–8	Reserved—these bits are reserved and should read 0.	
RPP Bit 7	**Reset Presence Pulse**—This bit is self clearing and will be cleared after the presence of the Presence pulse from the 1-Wire is detected.	0 = Do nothing / pulse complete. 1 = Generate Reset Pulse and sample for DS2502 presence pulse. This bit is self clearing and will be cleared after the presence is detected.
PST Bit 6	**Presence Status**—This bit is valid after the RPP bit is self-cleared.	0 = Device is not present. 1 = Device is present. This bit is valid after the RPP bit is self cleared.
WR0 Bit 5	**Write 0**—This bit is self-clearing and will be cleared when the write of the bit is complete.	0 = Do nothing / Write sequence complete. 1 = Write a 0 bit to the interface. This bit is self clearing and will be cleared when the write of the bit is complete.

Table 36-5. Control Register Description (continued)

Name	Description	Settings
WR1 Bit 4	**Write 1 / Read**—This bit is self-clearing and is cleared when the writing of the bit is complete. This also reads the bit since the Write 1 and Read timing are identical. The value of the read bit is stored in RDST, and is valid after WR1/RD is self-cleared.	0 = Do nothing / Write sequence complete. 1 = Write a 1 bit to the interface. This bit is self-clearing and will be cleared when the write of the bit is complete. This also reads the bit since the Write 1 and Read timing are identical. The value of the read bit is stored in RDST, and is valid after WR1/RD is self-cleared.
RDST Bits 3	**Read Status**—This bit is valid after the WR1/RD bit is self-cleared.	0 = A 0 was sampled during a read. 1 = A 1 was sampled during a read. This bit is valid after the WR1/RD bit is self-cleared.
Reserved Bits 2–0	Reserved—these bits are reserved and should read 0.	

36.6.2 Time Divider Register

1-Wire Time Divider Register clock divider register used to generate the internal time base within the module. Internal time generation is made up by a clock divider. The purpose of this internal time generation is to make a 1 MHz clock from the main clock.

TIME_DIVIDER — Time Divider Register — $1000_9002

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									dvdr							
TYPE	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 36-6. Time Divider Register Description

Name	Description	Settings
Reserved Bits 15–8	Reserved—these bits are reserved and should read 0.	
dvdr Bits 7–0	**Predivider Factor**—This field is used to select the clock divider setting to control the frequency of the generated clock.	0 = Divisor value is 1 (default) 1 = Divisor value is 2 … FF = Divisor value is 256

It is the user's responsibility to program this register so that the binary rate frequency is as close as possible to 1 MHz (1 MHz = clock / (divider + 1)). If the clock frequency is 30 MHz, then the proper value to write into the divider register is 29.

NOTE

The precision of the generated clock is very important to ensure the proper operation of the 1-Wire module. This module is based on a state machine which undertakes actions at defined times.

Table 36-7. System Timing Requirements

Times	Values (microsec)	Minimum (microsec)	Maximum (microsec)	Absolute Precision	Relative Precision
RSTL	511	480	–	31	0.0645
PST	68	60	75	7	0.1
RSTH	512	480	–	32	0.0645
LOW0	100	60	120	20	0.2
LOWR	5	1	15	4	0.8
READ_sample	13	–	15	2	0.15

The most stringent constraint is 0.0645 as a relative time imprecision.
The time relative precision is directly derived from the frequency of the derivated clock (f):
time relative precision = 1/f -1 = divider/clock(MHz) - 1

The Table 36-8 gathers relative time precision for different main clock frequencies.

Table 36-8. System Clock Requirements

Main Clock Frequency (MHz)	13	16.8	19.44
Clock divide ratio	13	17	19
Generated frequency (MHz)	1	0.9882	1.023
Relative time imprecision	0	0.0117	0.023

This shows that the user should take care of the main clock frequency when using the 1-Wire module. If the main clock is an exact integer multiple of 1 MHz, then the generated frequency will be exactly 1 MHz.

NOTE

A main clock frequency below 10 MHz could cause stability problems and incorrect operation in the 1-Wire module.

36.6.3 Reset Register

The Reset Register is used to reset the 1-Wire module through software. This register is not self-clearing, therefore the programmer must write a 1 to reset the register and then write a 0 to release the reset signal.

RESET — Reset Register — $1000_9004

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																rst
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 36-9. Reset Description

Name	Description	Settings
Reserved Bits 15–1	Reserved—These bits are reserved and should read 0.	
RST Bit 0	**Software Reset**—The reset register is used to reset the module through software.	0 = 1-Wire is not reset 1 = 1-Wire is reset

36.7 Reference Documents

[1] DS2502 data sheet, Dallas Semiconductor

Index

F

G

H

I

J

L

M

P

R

S

T

U

V

W

X

Y