MC68SZ328 Integrated Processor

Reference Manual

MC68SZ328RM/D
Rev. 1.3, 3/2004

MOTOROLA intelligence everywhere™

digital dna™

Freescale Semiconductor, Inc.

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

HOW TO REACH US:

USA/EUROPE/LOCATIONS NOT LISTED:

Motorola Literature Distribution
P.O. Box 5405, Denver, Colorado 80217
1-800-521-6274 or 480-768-2130

JAPAN:

Motorola Japan Ltd.; SPS, Technical Information Center,
3-20-1, Minami-Azabu, Minato-ku, Tokyo 106-8573, Japan
81-3-3440-3569

ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.; Silicon Harbour Centre,
2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T.,
Hong Kong
852-26668334

HOME PAGE:

http://motorola.com/semiconductors

Contents

Chapter 6
System Control

Chapter 7
Chip-Select Module

Chapter 8
DRAM Controller

Chapter 9
DMA Controller

Chapter 10
LCD Controller

Chapter 11
Analog Signal Processor

Chapter 12
General-Purpose Timers

Chapter 13
Real-Time Clock

Chapter 14
Pulse-Width Modulator 1 and 2

Chapter 15
Interrupt Controller

Chapter 16
General Purpose I/O Module

Chapter 17
Multimedia Card/Secure Digital Host Controller

Chapter 18
Memory Stick Host Controller

Chapter 19
Universal Asynchronous Receiver/Transmitter 1 and 2

Chapter 20
Configurable Serial Peripheral Interface

Chapter 21
USB Device Module

Chapter 22
I^2C

Chapter 23
Bootstrap Mode

Chapter 24
In-Circuit Emulation

Chapter 25
Reset Module

Chapter 26
Electrical Characteristics

Chapter 27
Mechanical Data and Ordering Information

List of Figures

List of Tables

About This Book

This reference manual describes the features and operation of the MC68SZ328 (DragonBall™ Super VZ) microprocessor, the fourth generation of the DragonBall family of products. It provides the details of how to initialize, configure, and program the MC68SZ328. The manual presumes basic knowledge of 68000 architecture.

Audience

The MC68SZ328 reference manual is intended to provide a design engineer with the necessary data to successfully integrate the MC68SZ328 into a wide variety of applications. It is assumed that the reader has a good working knowledge of the 68000 core. For programming information about the 68000, see the documents listed in the Suggested Reading section of this preface.

Revision History

The following table summarizes revisions to this document since the previous release (Rev. 1.2).

Revision History

Location	Revision
Chapter 2, “Signal Descriptions.”	Table 2-1, “MC68SZ328 Signal Name Descriptions,” on page 2-1: This table was updated.
Chapter 5, “Clock Generation Module and Power Control Module.”	Section 5.1, “Clock Generation Module Overview,” on page 5-1: This section was updated.
Chapter 6, “System Control.”	• Section 6.2.3, “ID Register,” on page 6-4: This register was updated. • Table 6-3, “ID Register Description,” on page 6-4 and Table 6-4, “I/O Drive Control Register Description,” on page 6-5: These tables were updated.
Chapter 11, “Analog Signal Processor.”	• Section 11.2.3, “Input Select,” on page 11-2: The note in this section was updated. • Section 11.9.1, “ASP Control Register,” on page 11-12: This section was updated.
Chapter 16, “General Purpose I/O Module.”	• Section 16.4.7.9, “Port J Interrupt Polarity Register,” on page 16-36: This section was updated. • Table 16-61, “Port K Dedicated I/O Function Assignments,” on page 16-38: This table was updated.

Revision History (Continued)

Location	Revision
Chapter 26, “Electrical Characteristics.”	• Table 26-3, “Maximum and Minimum DC Characteristics,” on page 26-2: Footnote #2 was updated. • Section 26.4.2.1.2, “68K Write Cycle,” on page 26-5: This section was updated. • Table 26-28, “Pen ADC Characteristics,” on page 26-39: This table was updated.

Organization

The MC68SZ328 reference manual is organized into 27 chapters that cover the operation and programming of the DragonBall Super VZ device. The following list summarizes the chapters.

Chapter 1 **Introduction** — Contains a device feature list, overview of system modules, system block diagrams, and a feature comparison of the different generations of the DragonBall devices.

Chapter 2 **Signal Descriptions —** Contains listings of the MC68SZ328 input and output signals, organized into functional groups.

Chapter 3 **FLX68000 Core —** Provides a high-level overview of the FLX68000 core including its instruction set.

Chapter 4 **Memory Map —** Summarizes the memory organization, programming information, and registers’ addresses and reset values.

Chapter 5 **Clock Generation Module and Power Control Module** — 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 6 **System Control —** Describes the operation of and programming models for the system control, peripheral control, ID register, and I/O drive control registers.

Chapter 7 **Chip-Select Module —** Describes the operation and programming of the chip-select logic. It includes information related to the operation of the DRAM controller and other memory-related applications.

Chapter 8 **DRAM Controller** — Details operation and programming of the DRAM controller. This module provides a glueless interface to 8-bit or 16-bit DRAM supporting EDO RAM, Fast Page Mode, and synchronous DRAM.

Chapter 9 **DMA Controller —** Provides operational details of the DMA controller contained in the MC68SZ328. The DMA controller provides two memory channels and four I/O channels to support a wide variety of DMA operations.

Chapter 10 **LCD Controller —** Describes the operation and programming of the LCD controller, which provides display data for external LCD drivers or for an LCD panel.

Chapter 11 **Analog Signal Processor** — Describes the analog signal processing (ASP) moduleMC68SZ328 which provides support and conversion capabilities for a variety of analog devices, including analog-to-digital controllers (ADC) for voice

processing and pen input. The ASP also includes embedded circuity to support a touch panel.

Chapter 12 **General-Purpose Timers** — Details the two 16-bit timers that can be used as both watchdogs and alarms. It also describes how the timers can be combined into a single 32-bit timer.

Chapter 13 **Real-Time Clock —** Describes the operation of the real-time clock (RTC) 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 **Pulse-Width Modulator 1 and 2 —** Describes both pulse-width modulators. Programming information is also provided.

Chapter 15 **Interrupt Controller —** Provides a description and operational considerations for interrupt controller operation. It includes a description of the vector generator and pen and keyboard interrupts.

Chapter 16 **GPIO Module —** Covers all 76 GPIO lines found in the MC68SZ328. Because each pin is individually configurable, a detailed description of the operation of and programming information for each pin is provided.

Chapter 17 **Multimedia Card and Secure Digital Host Controller —** Describes the Multimedia Card (MMC) host controller which is a Flash-based mass storage product. The MMC bus uses a multi-drop master salve configuration requiring a host device to control the multimedia card. This chapter also describes the Secure Digital feature of the MMC, its operation and programming information.

Chapter 18 **Memory Stick Controller —** Describes how data is transferred to a Memory Stick device. It also discusses how to configure and program the Memory Stick Host Controller module.

Chapter 19 **Universal Asynchronous Receiver/Transmitter 1 and 2 —** Details the two universal asynchronous receiver/transmitter (UART) ports that allow the incorporation of serial communication in existing and new designs. Also described are how data is transported in character blocks using the standard “start-stop” format and details how to configure and program the UART modules.

Chapter 20 **Configurable Serial Peripheral Interface** — Describes the features of the MC68SZ328’s configurable serial peripheral interfaces and how they are used to communicate with external devices.

Chapter 21 **USB Device Module —** Describes the features and programming model of the MC68SZ328’s USB device module.

Chapter 22 **I^2C Module** — Describes the MC68SZ328 I^2C module, including I^2C protocol, clock synchronization, and the registers in the I^2C programing mode.

Chapter 23 **Bootstrap Operation —** Details the operation of bootstrap models and describes 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 24 **In-Circuit Emulation —** Describes the in-circuit emulation (ICE) module and how it is used to support low-cost emulator designs for the MC68SZ328 microprocessor.

Chapter 25 **Reset Module —** Provides a detailed description of the reset module and associated timing and signals. The reset module processes of all of the system reset signals required by the MC68SZ328.

Chapter 26 **Electrical Characteristics —** Contains information about the electrical characteristics of the MC68SZ328 integrated processor.

Chapter 27 **Mechanical Data and Ordering Information —** Provides mechanical data, including illustrations, and ordering information.

Suggested Reading

The following documents are required for a complete description of the MC68SZ328 and are necessary to design properly with the device. Especially for those not familiar with the earlier DragonBall products or the 68000 core, the following documents will be helpful when used in conjunction with this manual.

M68000 Family Programmer's Reference Manual (order number M68000PM/AD)

M68000 User's Manual (order number M68000UM/D)

M68000 User's Manual Addendum (order number M68000UMAD/AD)

MC68EZ328 User's Manual (order number MC68EZ328UM/D)

MC68EZ328 User's Manual Addendum (order number MC68EZ328UMAD/D)

MC68VZ328 Product Brief (order number MC68VZ328P/D)

MC68VZ328 User's Manual (order number MC68VZ328UM/D)

MC68VZ328 User's Manual Addendum (order number MC68VZ328UMAD/D)

MC68SZ328 Product Brief (order number MC68SZ328P/D)

The manuals may be found at the Motorola Web site at http://www.Motorola.com/DragonBall. These documents may be downloaded from the Web site, or a printed version may be obtained from a local sales office. The Web site also may 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 dollar sign ($) or *0x* are hexadecimal.

Definitions, Acronyms, and Abbreviations

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

BCD	binary coded decimal
CGM	clock generation module
DRAM	dynamic RAM
FIFO	first in first out
ICE	in-circuit emulation
MAP	mold array process
MAPBGA	mold array process ball grid array
MIPS	million instructions per second
PWM	pulse-width modulator
RTC	real-time clock
SIM	system integration module
SPI	serial peripheral interface
SRAM	static RAM
TQFP	thin quad flat pack
UART	universal asynchronous receiver/transmitter
XTAL	crystal

Chapter 1
Introduction

The MC68SZ328 (DragonBall Super VZ) microprocessor, the fourth generation of the 68K-based DragonBall family of products, is designed to save system designers time, power, and cost. Requiring less board space, it allows for reduced pin count and fewer programming steps when designing products. The major differences between previous versions of DragonBall processors and the new MC68SZ328 are an improvement in system speed, TFT color LCD support, an A/D converter (with touch panel control), an MMC/SD host controller, a DMA controller, embedded SRAM, a USB device controller, and an I^2C interface.

All these features combine to make the MC68SZ328 the microprocessor of choice among system designers. Its functionality and glue logic are all optimally connected, timed with the same clock, fully tested, and uniformly documented. The MC68SZ328's primary package is a 196-pin MAP BGA designed to occupy the smallest possible footprint on your board.

A summary of the MC68SZ328's benefits includes the following:

- Fourth generation of industry-leading DragonBall family for the personal, portable product market
- High level of system-function integration
- Low-power design combined with increased system performance
- Wide variety of applications including the most popular PDA designs, smart phones, and next-generation wireless communicators

1.1 Block Diagram

Figure 1-1 is a simplified functional block diagram of the MC68SZ328.

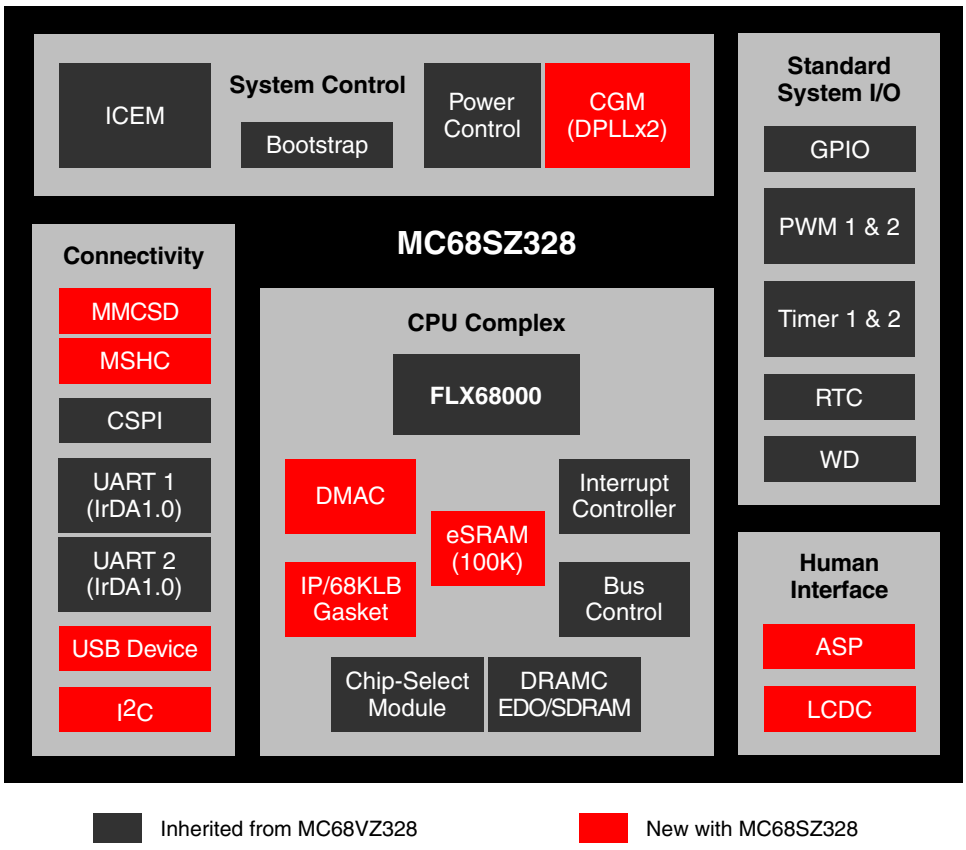


Figure 1-1. MC68SZ328 Functional Block Diagram

1.2 Features

The MC68SZ328 boasts a robust array of features that support a wide variety of applications. The features of the MC68SZ328 include the following:

- Static FLX68000 core—identical to MC68EC000 microprocessor
  - — Full compatibility with MC68000 and MC68EC000
  - — 32-bit internal address bus and 16-bit data bus
  - — Static design allows processor clock to be stopped for power saving
  - — 66.32 MHz processor clock
  - — External M68000 bus interface with selectable bus sizing for 8-bit and 16-bit data ports
- Clock generation module (CGM) and power control module
  - — Digital PLLs (phase-locked loops) for all internal clock generation
  - — Dual crystal inputs:
    - – 32.768 kHz for DMA clock, CPU clock, system clock, and LCD clock generation
    - – Separate crystal input for USB clock generation

— Support for three power modes for different power consumption needs: normal, doze, and sleep
— Programmable processor clock frequency from DC to 66.32 MHz for different bandwidth requirements

- Chip-select module
  — 12 chip-select signals: 10 general-purpose, programmable chip-select signals for external devices with programmable wait state and base address; 1 signal to select eSRAM; 1 signal for the emulation module.
  — Up to 32 Mbyte per chip-select (with a total of 4 Gbyte programmable space for address mapping).
  — Each chip-select can be configured to address an 8-bit or 16-bit space. Both 8-bit and 16-bit contiguous address memory devices can be mixed on a 16-bit data bus system.
  — Glueless interface to SRAM, EDO DRAM, SDRAM, EPROM, and flash memory.
- DRAM controller (DRAMC)
  — Two chip-select signals for supporting either SDRAM or EDO DRAM (self-refresh type only).
  — Each chip-select supports four banks of single data rate 64 Mbit, 128 Mbit, and 256 Mbit SDRAM with up to four banks active simultaneously.
  — Support for 4 Mbit, 16 Mbit, and 64 Mbit self-refresh-type EDO DRAM with 8, 9, 10, and 11 column addresses.
  — Configurable row cycle delay (tRC), row precharge delay (tRP), row-to-column delay (tRCD), and column-to-data delay (CAS latency).
  — Programmable refresh rate.
  — Data retention capability during normal system reset.
  — Support for power-down mode and self-refresh mode for SDRAM power saving.
  — Support for self-refresh mode for EDO DRAM power saving.
- Embedded SRAM (eSRAM)
  — 100 kbyte general-purpose internal SRAM
- Direct memory access controller (DMAC)
  — Provides two memory-to-memory channels and four I/O-to-memory or memory-to-I/O channels.
  — Supports 8-bit and 16-bit FIFO and memory port size for data transfer.
  — Supports modules shown in Table 1-1 on page 1-4.
  — Provides data transfer complete and error (burst time-out or request-out) interrupts to interrupt controller. DMA burst length is configurable for each channel.
  — Provides bus utilization control for memory channels.
  — Generates DMA burst time-out error for both memory and I/O channels to terminate DMA cycle when the burst cannot be completed in a programmed timing period.
  — Generates DMA request time-out error for I/O channels to interrupt 68K core when a DMA request is not expected during a programmed timing period.
  — Supports repeat data transfer function.
  — Supports block transfer function to speed up display functions such as image block movement in LCD display, retrieval of pre-store image pattern, and window effect.
  — Provides $\overline{\text{DMA\_REQ0}}$ and $\overline{\text{DMA\_REQ1}}$, two external DMA request pins for external devices

to initiate data transfer in memory channels 0 and 1, respectively.

Table 1-1. Modules with DMA Support

Module	DMA Capability
CSPI	Yes
UART 1 and 2	Yes
USB	Yes
MMC/SD	Yes
MSHC	Yes
ASP	Yes (enhanced ADC)
I^2C	No
PWM 1 and 2	No
Timer 1 and 2	No
RTC	No
Bootstrap	No

- LCD controller (LCDC)
  - — Support for single-screen (non-split), color or monochrome LCD panels and self-refresh-type LCD panels
  - — Panel sizes and summary of color and monochrome support are shown in Table 1-2 on page 1-5 and Table 1-3 on page 1-5
  - — Support for 4 bpp (bits per pixel) and 8 bpp for passive color panels and support for 4 bpp, 8 bpp, 12 bpp, and 16 bpp for TFT panels
    - – Up to 256 colors from a palette of 4096 colors for 8 bpp displays
    - – Up to 4096 colors for 12 bpp displays
    - – True 64K colors for 16 bpp displays
  - — 16 simultaneous grayscale levels from a palette of 16 for monochrome displays
  - — Standard panel interface for common LCD drivers; panel interface of 16-bit, 8-bit, 4-bit, 2-bit, and 1-bit wide LCD panel data bus for monochrome or color panels
  - — Glueless interface to passive and active color panel (TFT); direct interface to Sharp® 320 × 240 HR-TFT panel
  - — Use of system memory and embedded SRAM for display memory
  - — Hardware blinking cursor programmable at a maximum 61 × 61 pixels; support for logical operation between color hardware cursor and background
  - — Hardware panning (soft horizontal scrolling)
  - — 8-bit pulse-width modulator for software contrast control

Table 1-2. External Memory Used for Display Memory

Panel Type	Bits/ Pixel	Panel Interface (Bits)	Number of Grayscale Levels or Colors	Max. Standard Panel Size Supported (Pixels)[1]
Monochrome	1	1, 2, 4, 8	Black and white	640 x 480
	2	1, 2, 4, 8	4	640 x 480
	4	1, 2, 4, 8	16	640 x 480
CSTN	4, 8	8	16, 256	640 x 480
TFT	4, 8	16	16, 256	640 x 480
	12, 16	12,16	4096, 64K	640 x 480

1. The actual maximum panel size is constrained by system bandwidth use.

Table 1-3. eSRAM Used for Display Memory

Panel Type	Bits/ Pixel	Panel Interface (Bits)	Number of Grayscale Levels or Colors	Max. Standard Panel Size Supported (Pixels)[1]
Monochrome	1	1, 2, 4, 8	Black and white	640 x 480
	2	1, 2, 4, 8	4	640 x 480
	4	1, 2, 4, 8	16	320 x 240 or 240 x 320
CSTN	4, 8	8	16, 256	320 x 240 or 240 x 320
TFT	4, 8	16	16, 256	320 x 240 or 240 x 320
	12, 16	12, 16	4096, 64K	240 x 160 or 160 x 240

1. The actual maximum panel size is constrained by the size of eSRAM.

- Analog signal processing (ASP) module
  - ADC (16-bit resolution, 8-bit accuracy) with 3 inputs for touch panel and low voltage detect
  - Enhanced ADC (16-bit resolution, 8-bit accuracy) for use as second ADC with DMA support
  - Embedded touch panel circuitry
  - Pen ADC that supports auto data sampling at a configured sample rate
  - Interrupt-based operation with 12 × 16 FIFO for pen ADC sample data and two 8 × 16 FIFOs for data for enhanced ADC
  - Significantly reduced software overhead for pen input applications
  - Low power management
- Timers
  - Two identical 16-bit general-purpose timers/counters
  - Capability to cascade timers together to operate as a single 32-bit timer
  - Input capture capability with programmable trigger edge for interval measurement
  - 15 ns resolution at 66.32 MHz system clock
  - Timer input/output pin for event notification

- Real-time clock and sampling timer (RTC)
  - 32.768 kHz clock input
  - Full clock function: second, minute, hour, and day (up to 512 days)
  - One programmable alarm with interrupt capability
  - Sampling timer with selectable frequency (4 Hz, 8 Hz, 16 Hz, 32 Hz, 64 Hz, 256 Hz, 512 Hz) and interrupt capability that can be used for digitizer sampling or keyboard debouncing
- Watchdog timer (WD)
  - Programmable watchdog interrupt or reset
- Pulse-width modulation (PWM) modules
  - 8-bit (PWM 1) and 16-bit (PWM 2) resolution
  - 5-byte FIFO in 8-bit PWM 1 provides more flexibility on performance
  - Capability to generate high quality sound, tone, or melody
- Interrupt controller
  - All interrupts are maskable.
  - Interrupt level of all internal modules is configurable.
  - Four dedicated external interrupt IRQ signals (defined interrupt level; programmable edge, level, and polarity).
- General-purpose I/O (GPIO) ports
  - 93 total I/O ports in 12 groups with interrupt capability, all multiplexed with peripheral functions.
  - All ports are specially designed with dedicated internal pull-up and pull-down resistors.
- Multimedia card and secure digital (MMC/SD) host controller
  - Compatible with the MMC system specification version 2.2
  - Compatible with the SD Memory Card specification 1.0
  - Support for up to ten cards, one of which may be SD
  - Password protection of cards
  - Multi-SD function support
  - Auto error detection for access CRC, response CRC, and time out
  - Built-in 7 and 16 CRC generation and checking for command and data
  - Built-in programmable frequency counter for MMC/SD host controller bus
  - Maskable hardware interrupt for internal status and FIFO status indicator
  - 16-bit internal data operation, 8 × 16-bit FIFO, and DMA interface
  - Automatic operation pause when internal FIFO full, allowing flexible packet transaction for dynamic DMA transfer
  - Built-in 3-bit prescaler and 3-bit bus clock divider to maximize the performance of data transaction between memory card and host
- Memory Stick® host controller (MSHC)
  - Built-in 8-byte (4-word) FIFO buffer for transmit and receive
  - Built-in CRC circuit

  - Support for internal or external serial clock source
  - Built-in Serial Clock Divider: maximum 33.16 MHz serial data transfer rate
  - DMA supported; DMA request condition is selectable based upon FIFO status
  - Automatic command execution when an interrupt from the Memory Stick is detected (can be turned on/off)
  - RDY time-out period can be set by the number of serial clock cycles
  - Interrupt can also be output to the core when a time out occurs
  - Two built-in general-purpose input pins for detecting Memory Stick insertion/extraction
  - 16-bit host bus access (byte access not supported)
- Universal asynchronous receiver/transmitters (UARTs)
  - Two identical UARTs with interrupt-based operation
  - Support for serial data transmit/receive operation: 7 or 8 data bits, 1 or 2 stop bits, programmable parity (even, odd, or none)
  - Programmable standard baud rates up to 460.8 kbps
  - Receive FIFO size is 32 bytes; transmit FIFO size is 32 bytes
  - Both UARTs IrDA 1.0 ready
  - Maximum non-standard baud rate of 4.14 Mbps
  - Flexible DMA burst access to both UART 1 and UART 2 FIFO architectures
- Configurable serial peripheral interface (CSPI)
  - Master/slave configurable
  - 8 × 16 data-in FIFO and 8 × 16 data-out FIFO
  - Flow control signals incorporated to enable fast data communication
- Reset module
  - Provides stable system power-on reset and normal reset
- Bootstrap mode function
  - Allows user to initialize system and download programs or data to system memory through either UART
  - Accepts execution command to run program stored in system memory
  - Provides a 32-byte-long instruction buffer for 68000 instruction storage and execution
- Universal Serial Bus (USB) device
  - Compliant with Universal Serial Bus Specification revision 1.1.
  - Endpoint configurations are as shown in Table 1-4 on page 1-8. Five pipes are available for mapping.
    - Endpoint 0 is required by the USB specification.
    - Endpoints 1, 2, 3, and 4 may be configured as bulk or interrupt pipes (IN or OUT).
  - A frame match interrupt feature is supported to notify the user when a specific USB frame occurs. For DMA access, the maximum packet size for each endpoint is restricted by the FIFO size of the endpoint.
  - Four bulk/interrupt pipes are supported for 12 Mbps data transfer. The packet sizes are limited to 8, 16, 32, or 64 bytes, and the maximum packet size depends on the FIFO size of endpoint.

— No power drawn from the USB bus.
— Remote wake-up feature is supported via a register bit.
— Complete FIFO interrupts are provided (full, empty, error, high, low).
— End-of-frame and start-of-frame interrupt support.
— Full-speed (12 Mbps) operation.
— Intelligence related to packet retries and data framing is built into the FIFO controller.

Table 1-4. Endpoint Configurations

Endpoint	Direction	Physical FIFO Size (Bytes)	Endpoint Configuration	Maximum Packet Size
0	IN and OUT	32	Control	16
1	IN or OUT	16	Bulk or interrupt	16
2	IN or OUT	16	Bulk or interrupt	16
3	IN or OUT	128	Bulk or interrupt	64
4	IN or OUT	128	Bulk or interrupt	64

- Inter-IC (I^2C) bus module
  - Compliant with Philips I^2C-bus standard (support for Standard-mode and Fast-mode)
  - Support for 7-bit address
  - Support for 3.0 V devices
  - Multiple-master operation
  - Software-programmable for 1 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 and detection
  - Repeated START signal generation
  - Acknowledge bit generation and detection
  - Bus-busy detection
- In-circuit emulation (ICE) module
  - Dedicated memory space for emulator debug monitor with chip-select
  - Dedicated interrupt (interrupt level 7) for in-circuit emulation
  - One address-signal comparator and one control-signal comparator, with masking to support single or multiple hardware execution breakpoints
  - One breakpoint instruction insertion unit
- Operating system frequency
  - Up to 66.32 MHz

- Operating voltage
  - Internal: 1.8 V
  - I/O: 2.7 V to 3.3 V
- Package type
  - 196-pin MAP BGA
  - Size: 12 mm × 12 mm
  - Pitch size: 0.8 mm

1.3 DragonBall Series Comparison

The DragonBall Super VZ marks the fourth generation in the DragonBall series. Table 1-5 compares the new processor with previous generations. The gray cells indicate the features that are available for the first time on the MC68SZ328.

Table 1-5. DragonBall Series Comparison

Feature	DragonBall	DragonBall EZ	DragonBall VZ	DragonBall Super VZ
Core	68EC000	68EC000	FLX68000	FLX68000
LCD controller	Up to 4 gray (1024 x 512)	Up to 16 gray (320 x 240), 2 gray (640 x 512)	Up to 16 gray (640 x 512)	Up to 16 gray for monochrome Up to 256 colors for passive Up to 64K (16-bit) colors for active TFT
Chip-selects	16	8 (external), 1 ($\overline{\text{EMUCS}}$)	8 (external), 1 ($\overline{\text{EMUCS}}$)	10 (external), 1 ($\overline{\text{EMUCS}}$), 1 (internal SRAM)
DRAM controller	Not provided	Provided	Provided (supports SDRAM)	Provided (supports EDO DRAM and SDRAM)
PLL and power control	Provided	Provided	Provided	Provided (digital PLL)
Interrupt controller	Provided	Provided	Provided	Provided
Timers	2	1	2	2
RTC	1	Enhanced (with sampling timer)	Enhanced (with sampling timer)	Enhanced (with sampling timer)
SPI 1	Master (SPIM)	Master	Master/slave	Master/slave
SPI 2	Slave (SPIS)	Not provided	Master	Not provided
UART (with infrared interface)	1	1	2	2
16-bit PWM	Provided	Not provided	Provided	Provided
8-bit PWM	Not provided	Provided	Provided	Provided
ICE module	Not provided	Provided	Provided	Provided
Bootstrap	Not provided	Provided	Provided	Provided
GPIO	77	54	76	93

Table 1-5. DragonBall Series Comparison (Continued)

Feature	DragonBall	DragonBall EZ	DragonBall VZ	DragonBall Super VZ
ASP	Not provided	Not provided	Not provided	Provided (with touch panel control circuitry)
MMC/SD host controller	Not provided	Not provided	Not provided	Provided
Memory Stick host controller	Not provided	Not provided	Not provided	Provided
DMA	Not provided	Not provided	Not provided	Provided
Embedded SRAM	Not provided	Not provided	Not provided	100 kbyte
USB	Not provided	Not provided	Not provided	Provided
I^2C	Not provided	Not provided	Not provided	Provided
Speed	Up to 16.58 MHz	Up to 16.58 MHz and 20 MHz	Up to 33.16 MHz	Up to 66.32 MHz
Voltage	3.0 V to 3.6 V 4.5 V to 5.5 V	3.0 V to 3.6 V	2.7 V to 3.3 V	2.7 V to 3.3 V
Packages	144-pin TQFP	100-pin TQFP, 144-pin BGA	144-pin TQFP, 144-pin MAP BGA	196-pin MAP BGA

Chapter 2
Signal Descriptions

2.1 Signal Descriptions by Function

All of the interface signals, pin names, and descriptions used in the MC68SZ328 are listed in Table 2-1.

Table 2-1. MC68SZ328 Signal Name Descriptions

Signal Name	Pin Name	Description
Address Bus Signals		
A0 **A[9:1]/MA[9:1]**	PG1/A0/MA0 A[9:1]/MA[9:1]	Address bits 9–0, mux with multiplexed DRAM row/column address MA[9:0]
A[15:10]	A[15:10]	Address bits 15–10.
A16	A16/SDBIA9	Address bit 16, mux with multiplexed SDRAM interleaved mode A9.
A[24:17]	A[19:17] PF[6:3]/A[23:20] PE2/A24	Address bits 24–19
Data Bus Signals		
D[15:0]	D[15:0]	Data bits 15–0
Bus Control Signals		
BUSW	PG0/BUSW/$\overline{\text{DTACK}}$	Bus width selection of boot chip-select signal.
DTACK	PG0/BUSW/$\overline{\text{DTACK}}$	Data transfer acknowledge
$\overline{\text{LWE}}$	$\overline{\text{LWE}}$/LB	Lower write enable
$\overline{\text{UWE}}$	$\overline{\text{UWE}}$/UB	Upper write enable
$\overline{\text{OE}}$	$\overline{\text{OE}}$	Output enable
R/$\overline{\text{W}}$	PK1/R/$\overline{\text{W}}$	Read/write signal from internal FLX68000 core
$\overline{\text{UDS}}$	$\overline{\text{PB2}}$/CSC0/$\overline{\text{UDS}}$	Upper data strobe signal from internal FLX68000 core
$\overline{\text{LDS}}$	$\overline{\text{PB3}}$/CSC1/$\overline{\text{LDS}}$	Lower data strobe signal from internal FLX68000 core

Table 2-1. MC68SZ328 Signal Name Descriptions (Continued)

Signal Name	Pin Name	Description
Clock Generation Module and Power Controller Module (CGM)		
CLK16I	CLK16I	16 MHz crystal input, or 16 MHz oscillator input when internal oscillator circuit is bypassed.
CLK16O	CLK16O	16 MHz crystal output
CLK32I	CLK32I	32.768 kHz low frequency crystal input pin. Equivalent to EXTAL pin in MC68VZ328.
CLK32O	CLK32O	32.768 kHz low frequency crystal output pin.Equivalent to XTAL pin in MC68VZ328.
CLKO	PF2/CLKO	Clock output signal from SYSCLK, CPUCLK, DMACLK, USB-CLK, USBPLL, MCUPLL or PREMUL clock.
Chip-Select Module		
$\overline{\text{CSA0}}$	$\overline{\text{CSA0}}$	Chip-select A bit 0
$\overline{\text{CSA1}}$	$\overline{\text{PF7/CSA1}}$	Chip-select A bit 1
$\overline{\text{CSB0}}$	$\overline{\text{PB0/CSB0}}$	Chip-select B bit 0
$\overline{\text{CSB1}}$	$\overline{\text{PB1/CSB1}}$	Chip-select B bit 1
$\overline{\text{CSC0}}$	$\overline{\text{PB2/CSC0/UDS}}$	Chip-select C bit 0
$\overline{\text{CSC1}}$	$\overline{\text{PB3/CSC1/LDS}}$	Chip-select C bit 1
$\overline{\text{CSD0}}$	$\overline{\text{PB4/CSD0/DMA\_REQ0}}$	Chip-select D bit 0
$\overline{\text{CSD1}}$	$\overline{\text{PB5/CSD1/DMA\_REQ1}}$	Chip-select D bit 1
$\overline{\text{CSE/SDCS0/RAS0}}$	$\overline{\text{PG6/CSE/SDCS0/RAS0}}$	Chip-select E, mux with DRAM signals
$\overline{\text{CSF/SDCS1/RAS1}}$	$\overline{\text{PG7/CSF/SDCS1/RAS1}}$	Chip-select F, mux with DRAM signals
DRAM Controller		
$\overline{\text{SDCS0/RAS0}}$	$\overline{\text{PG6/CSE/SDCS0/RAS0}}$	SDRAM chip-select or EDO DRAM RAS signal
$\overline{\text{SDCS1/RAS1}}$	$\overline{\text{PG7/CSF/SDCS1/RAS1}}$	SDRAM chip-select or EDO DRAM RAS signal
$\overline{\text{SDRAS/CAS0}}$	$\overline{\text{PK2/SDRAS/CAS0}}$	SDRAM CAS signal or EDO DRAM CAS0 signal
$\overline{\text{SDCAS/CAS1}}$	$\overline{\text{PK3/SDCAS/CAS1}}$	SDRAM CAS signal or EDO DRAM CAS1 signal
$\overline{\text{SDWE/DWE}}$	$\overline{\text{PM4/SDWE/DWE}}$	SDRAM WE signal or EDO DRAM WE signal
SDCLKE0/$\overline{\text{DOE}}$	$\overline{\text{PM1}}$/SDCLKE0/$\overline{\text{DOE}}$	SDRAM clock enable 0 or EDO DRAM OE signal
SDCLKE1	$\overline{\text{PM5}}$/SDCLKE1	SDRAM clock enable 1 only
SDCLK	SDCLK	SDRAM clock
DQM[1:0]	$\overline{\text{PM[3:2]}}$/DQM[1:0]	SDRAM output enable and write mask

Table 2-1. MC68SZ328 Signal Name Descriptions (Continued)

Signal Name	Pin Name	Description
A[9:0]/MA[9:0]	PG1/A0/MA0 A[9:1]/MA[9:1]	Multiplexed DRAM row/column address MA[9:1], mux with address bits 9–1.
MA[11:10]	PM[7:6]/MA[11:10]	Multiplexed ROW/COL address, not mux with any system address line.
A[15:10]	A[15:10]	Address bits 15–10.
A16	A16/SDBIA9	Address bit 16, mux with multiplexed SDRAM interleaved mode A9.
A[24:17]	A[19:17] PF[6:3]/A[23:20] PE2/A24	Address bits 24–19
D[15:0]	D[15:0]	SDRAM data bus, using system data bus.
DMA Controller (DMAC)		
DMA_REQ0	$\overline{\text{PB4/CSD0/DMA\_REQ0}}$	DMA channel 0 request pin.
DMA_REQ1	$\overline{\text{PB5/CSD1/DMA\_REQ1}}$	DMA channel 1 request pin.
LCD Controller (LCDC)		
LD[15:0]	PC[3:0]/LD[3:0] PK[7:4]/LD[7:4] PP[7:0]/LD[15:8]	LCD data bus least significant bit (support only up to 5x6x5). All LCD signals are driving low after reset and when LCD is off.
FLM/VSYNC	PC4/FLM/VSYNC	Frame sync or Vsync
LP/HSYNC	PC5/LP/HSYNC	Line pulse or Hsync
SCLK	PC6/SCLK	Shift clock
ACD/OE	PC7/ACD/OE	Alternate crystal direction or LCD OE signal
LCONTRAST	PF0/LCONTRAST	This signal is used to control the LCD bias voltage as contrast control
SPL/SPR	PD0/SPL/SPR	Program horizontal scan direction (sharp panel)
PS	PD1/PS	Control signal output for source driver (sharp panel)
CLS	PD2/CLS	Start signal output for gate driver. This signal is inverted version of PS. (sharp)
REV	PD3/REV	Signal for common electrode driving signal preparation. (sharp)
Analog Signal Processor (ASP)		
UIP	Uip	Analog U input (for low voltage, temperature measurement, etc.)
UIN	Uin	Analog U reference

Table 2-1. MC68SZ328 Signal Name Descriptions (Continued)

Signal Name	Pin Name	Description
PX1	Px1	Touch panel x-axis analog input
PX2	Px2	Touch panel x-axis analog reference
PY1	Py1	Touch panel y-axis analog input
PY2	Py2	Touch panel y-axis analog reference
R1A	R1a	Reference resistor Rp1 terminal
R1B	R1b	Reference resistor Rp1 terminal
R2A	R2a	Reference resistor Rp2 terminal
R2B	R2b	Reference resistor Rp2 terminal
RVM	RVM	Reference resistor Rref1 terminal
RVP	RVP	Reference resistor Rref1 terminal
EADCP	EADCP	Positive enhanced ADC analog input
EADCN	EADCN	Negative enhanced ADC analog input
REF1	Ref1	Positive reference
REF2	Ref2	Negative reference
RBGP	RbgP	Positive bandgap reference
RBGN	RbgN	Negative bandgap reference
General-Purpose Timers 1 and 2		
TIN/TOUT	PB6/TIN/TOUT	Input capture or output compare of Timers 1 and 2
TIN2/TOUT2	PF1/TIN2/TOUT2	Input capture or output compare of Timer 2
Pulse-Width Modulators 1 and 2 (PWM 1 and 2)		
PWMO1	PB7/PWMO1	PWM OUTPUT1
PWMO2	PK0/$\overline{\text{DATA\_READY}}$/PWMO2	PWM OUTPUT2
Interrupt Controller		
IRQ1	PD4/IRQ1	Dedicated external interrupt pins (level 1)
IRQ2	PD5/IRQ2	Dedicated external interrupt pins (level 2)
IRQ3	PD6/IRQ3	Dedicated external interrupt pins (level 3)
IRQ6	PD7/IRQ6	Dedicated external interrupt pins (level 6)
$\overline{\text{EMUIRQ}}$	PG2/$\overline{\text{EMUIRQ}}$	Level 7 interrupt pin or EMUIRQ pin for ICEM

Table 2-1. MC68SZ328 Signal Name Descriptions (Continued)

Signal Name	Pin Name	Description
I/O Ports		
PB[7:0]	PB[7:0]/...	Bits 7–0 of PORT B
PC[7:0]	PC[7:0]/...	Bits 7–0 of PORT C
PD[7:0]	PD[7:0]/...	Bits 7–0 of PORT D
PE[7:0]	PE[7:0]/...	Bits 7–0 of PORT E
PF[7:0]	PF[7:0]/...	Bits 7–0 of PORT F
PG[7:0]	PG[7:0]/...	Bits 7–0 of PORT G
PJ[7:0]	PJ[7:0]/...	Bits 7–0 of PORT J
PK[7:0]	PK[7:0]/...	Bits 7–0 of PORT K
PM[7:1]	PM[7:1]/...	Bits 7–1 of PORT M
PN[7:0]	PN[7:0]/...	Bits 7–0 of PORT N
PP[7:0]	PP[7:0]/...	Bits 7–0 of PORT P
PR[5:0]	PR[5:0]/...	Bits 5–0 of PORT R
Multimedia Card/Secure Digital Host Controller (MMC/SD)		
MMCSD_CLK	PR0/MMCSD_CLK/ MS_SDIO	MMCSD Output Clock
MMCSD_CMD	PR1/MMCSD_CMD/MS_PI0	MMCSD Command
MMCSD_DAT[0]	PR2/MMCSD_DAT[0]/MS_BS	MMCSD Data0
MMCSD_DAT[1]	PR3/MMCSD_DAT[1]/MS_PI1	MMCSD Data1
MMCSD_DAT[2]	PR4/MMCSD_DAT[2]/MS_SCLKI	MMCSD Data2
MMCSD_DAT[3]	PR5/MMCSD_DAT[3]/MS_SCLKO	MMCSD Data3
Universal Asynchronous Receiver/Transmitter 1 (UART1)		
RXD1	PE4/RXD1	UART 1 receive data
TXD1	PE5/TXD1	UART 1 transmit data
$\overline{\text{RTS1}}$	PE6/$\overline{\text{RTS1}}$	UART 1 request to send
$\overline{\text{CTS1}}$	PE7/$\overline{\text{CTS1}}$	UART 1 clear to send
UCLK	PE3/UCLK	UART 1 and 2 baud clock input and output

Table 2-1. MC68SZ328 Signal Name Descriptions (Continued)

Signal Name	Pin Name	Description
Universal Asynchronous Receiver/Transmitter 2 (UART2)		
RXD2	PJ4/RXD2	UART 2 receive data
TXD2	PJ5/TXD2	UART 2 transmit data
$\overline{\text{RTS2}}$	PJ6/$\overline{\text{RTS2}}$	UART 2 request to send
$\overline{\text{CTS2}}$	PJ7/$\overline{\text{CTS2}}$	UART 2 clear to send
Configurable Serial Peripheral Interface (CSPI)		
MOSI	PJ0/MOSI	Master out/slave in.
MISO	PJ1/MISO	Slave in/master out.
SPICLK	PJ2/SPICLK	Serial clock
$\overline{\text{SS}}$	PJ3/$\overline{\text{SS}}$	CSPI slave select
$\overline{\text{DATA\_READY}}$	PK0/$\overline{\text{DATA\_READY}}$/PWMO2	CSPI data ready
USB Device Controller		
USBD_AFE	PN0/USBD_AFE	Analogue front-end enable.
$\overline{\text{USBD\_ROE}}$	PN1/$\overline{\text{USBD\_ROE}}$	Reverse output enable.
USBD_VMO	PN2/USBD_VMO	USB module data output.
USBD_VPO	PN3/USBD_VPO	USB module data output.
USBD_SUSPND	PN4/USBD_SUSPND	Transceiver suspend enable.
USBD_RCV	PN5/USBD_RCV	USB module receive data.
USBD_VP	PN6/USBD_VP	Input D+ signal connected directly to the D+.
USBD_VM	PN7/USBD_VM	Input D- signal connected directly to the D-.
I^2C		
SCL	PE0/SDA	Serial clock
SDA	PE1/SCL	Serial data
Reset and Boot Mode Select		
$\overline{\text{PWR\_RST}}$	$\overline{\text{PWR\_RST}}$	Power up reset pin (cold reset)
$\overline{\text{RST}}$	$\overline{\text{RST}}$	Normal reset pin (hot reset)
BST0	BST0	Boot mode select 0
BST1	BST1	Boot mode select 1

Table 2-1. MC68SZ328 Signal Name Descriptions (Continued)

Signal Name	Pin Name	Description
BST2	BST2	Boot mode select 2
In-Circuit Emulation Module		
EMUIRQ	PG2/$\overline{\text{EMUIRQ}}$	Emulator interrupt request (level 7)
P/$\overline{\text{D}}$	PG3/ P/$\overline{\text{D}}$	Program/data
EMUCS	PG4/$\overline{\text{EMUCS}}$	Emulator chip-select (8-bit data bus width)
EMUBRK	PG5/$\overline{\text{EMUBRK}}$	Emulator breakpoint
Voltage Regulator		
QVDD	QVDD	This is the output of the voltage regulator which should have an external capacitor tied to ground.
Power and Ground Signals		
NVDD1, NVDD2, NVDD3	NVDD1, NVDD2, NVDD3	External power supply to drive all I/O pins
NVSS1, NVSS2, NVSS3	NVSS1, NVSS2, NVSS3	Signal return pin for digital circuits
AVDD	AVDD	Quiet analog supply for internal voltage regulator.
AVSS	AVSS	Quiet system ground
Test Pins		
TP0	TP0	Test pin0 (disable Internal Voltage Regulator)
TP1	TP1	Test pin1 (bypass MCUPLL)
Memory Stick Host Controller (MSHC)		
MS_SDIO	PR0/MMCSD_CLK/ MS_SDIO	Memory Stick Serial Data
MS_PI0	PR1/MMCSD_CMD/MS_PI0	Memory Stick General Purpose Input 0.
MS_BS	PR2/MMCSD_DAT[0]/MS_BS	Memory Stick Bus State.
MS_PI1	PR3/MMCSD_DAT[1]/MS_PI1	Memory Stick General Purpose Input 1.
MS_SCLKI	PR4/MMCSD_DAT[2]/MS_SCLKI	Memory Stick External Clock (Input).
MS_SCLKO	PR5/MMCSD_DAT[3]/MS_SCLKO	Memory Stick Serial Clock (Output).

Freescale Semiconductor, Inc.

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Chapter 3
FLX68000 Core

The FLX68000 core in the MC68SZ328 is an updated implementation of the 68000 32-bit microprocessor architecture. The features of the FLX68000 core are:

- Low-power, fully static HCMOS implementation
- 32-bit address bus and 16-bit data bus
- Sixteen 32-bit data and address registers
- 56 powerful instruction types that support high-level development languages
- 14 addressing modes and 5 main data types
- Seven priority levels for interrupt control

The FLX68000 core is completely code compatible with other members of the M68000 family, enabling access to a broad base of established real-time kernels, operating systems, languages, applications, and development tools.

3.1 Core Programming Model

The FLX68000 core has 32-bit registers and a 32-bit program counter, as shown in Figure 3-1. The first eight registers (D7–D0) are data registers that are used for byte (8-bit), word (16-bit), and long-word (32-bit) operations. When used to manipulate data, the data registers affect the status register (SR). The next seven registers (A6–A0) and the user stack pointer (USP) can function as software stack pointers and base address registers. These registers can be used for word and long-word operations, but they do not affect the status register. The D7–D0 and A6–A0 registers can be used as index registers.

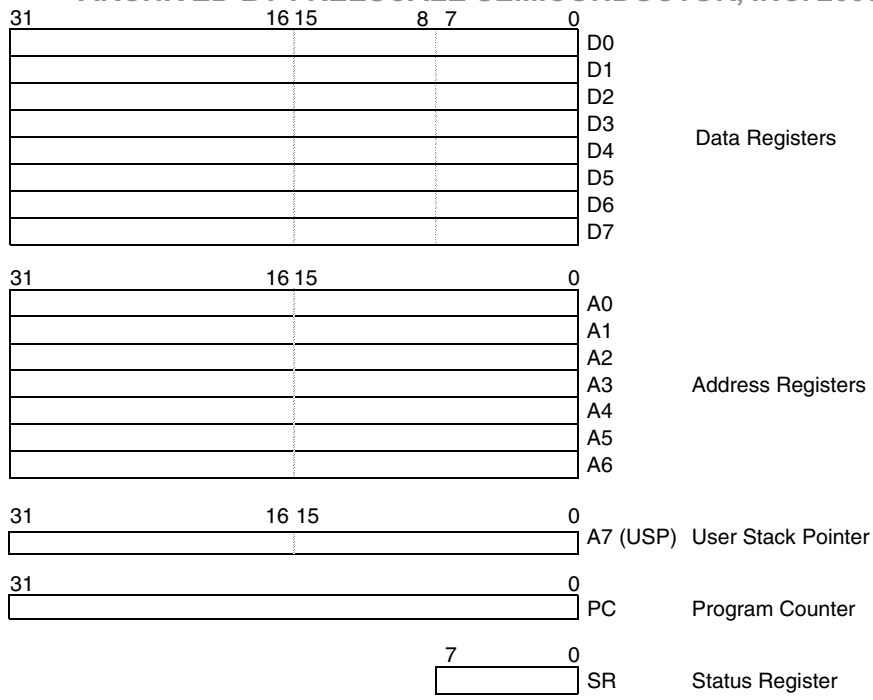


Figure 3-1. User Programming Model

In supervisor mode, the upper byte of the status register and the supervisor stack pointer (SSP) can also be programmed, as shown in Figure 3-2.

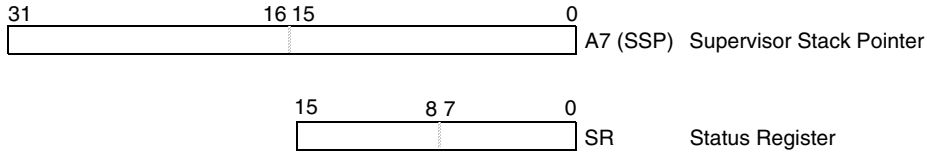


Figure 3-2. Supervisor Programming Model Supplement

The status register contains the interrupt mask with seven available levels, as well as extend (X), negative (N), zero (Z), overflow (V), and carry (C) condition codes. The T bit indicates when the processor is in trace mode, and the S bit indicates when it is in supervisor or user mode.

3.2 Data and Address Modes

The FLX68000 core supports five types of data and six main types of address modes. The five types of data are bits, binary-coded decimal (BCD) digits, bytes, words, and long words. The six types of address modes are shown in Table 3-1.

Table 3-1. Address Modes

Address Mode	Syntax
Register direct address	
• Data register direct	Dn
• Address register direct	An

Table 3-1. Address Modes (Continued)

Address Mode	Syntax
Absolute data address • Absolute short • Absolute long	 xxx.W xxx.L
Program counter relative address • Relative with offset • Relative with index offset	 d_{16}(PC) d_8(PC, Xn)
Register indirect address • Register indirect • Postincrement register indirect • Predecrement register indirect • Register indirect with offset • Indexed register indirect with offset	 (An) (An)+ –(An) d_{16}(An) d_8(An, Xn)
Immediate data address • Immediate • Quick immediate	 #xxx #1–#8
Implied address • Implied register	 SR/USP/SP/PC
Notes: Dn = Data register An = Address register Xn = Address or data register used as index register SR = Status register PC = Program counter SP = Stack pointer USP = User stack pointer ( ) = Effective address d_8 = 8-bit offset (displacement) d_{16} = 16-bit offset (displacement) #xxx = Immediate data	

3.3 FLX68000 Instruction Set

The FLX68000 core instruction set supports high-level languages that facilitate programming. Almost every instruction operates on bytes, words, and long words, and most of them can use any of the 14 address modes. Combining instruction types, data types, and address modes provides access to over 1,000 possible instructions. These instructions, shown in Table 3-2 on page 3-3, include signed and unsigned multiply and divide, quick arithmetic operations, binary-coded decimal (BCD) arithmetic, and expanded operations (through traps).

Table 3-2. Instruction Set

Mnemonic	Description	Mnemonic	Description
ABCD	Add decimal with extend	MOVEM	Move multiple registers
ADD	Add	MOVEP	Move peripheral data

Table 3-2. Instruction Set (Continued)

Mnemonic	Description	Mnemonic	Description
ADDA	Add address	MOVEQ	Move quick
ADDQ	Add quick	MOVE from SR	Move from status register
ADDI	Add immediate	MOVE to SR	Move to status register
ADDX	Add with extend	MOVE to CCR	Move to condition codes
AND	Logical AND	MOVE USP	Move user stack pointer
ANDI	AND immediate	MULS	Signed multiply
ANDI to CCR	AND immediate to condition codes	MULU	Unsigned multiply
ANDI to SR	AND immediate to status register	NBCD	Negate decimal with extend
ASL	Arithmetic shift left	NEG	Negate
ASR	Arithmetic shift right	NEGX	Negate with extend
Bcc	Branch conditionally	NOP	No operation
BCHG	Bit test and change	NOT	One's-complement
BCLR	Bit test and clear	OR	Logical OR
BRA	Branch always	ORI	OR immediate
BSET	Bit test and set	ORI to CCR	OR immediate to condition codes
BSR	Branch to subroutine	ORI to SR	OR immediate to status register
BTST	Bit test	PEA	Push effective address
CHK	Check register against bounds	RESET	Reset external devices
CLR	Clear operand	ROL	Rotate left without extend
CMP	Compare	ROR	Rotate right without extend
CMPA	Compare address	ROXL	Rotate left with extend
CMPM	Compare memory	ROXR	Rotate right with extend
CMPI	Compare immediate	RTE	Return from exception
DBcc	Test conditionally, decrement, and branch	RTR	Return and restore
DIVS	Signed divide	RTS	Return from subroutine
DIVU	Unsigned divide	SBCD	Subtract decimal with extend
EOR	Exclusive OR	Scc	Set conditional
EORI	Exclusive OR immediate	STOP	Stop
EORI to CCR	Exclusive OR immediate to condition codes	SUB	Subtract

Table 3-2. Instruction Set (Continued)

Mnemonic	Description	Mnemonic	Description
EORI to SR	Exclusive OR immediate to status register	SUBA	Subtract address
EXG	Exchange registers	SUBI	Subtract immediate
EXT	Sign extend	SUBQ	Subtract quick
JMP	Jump	SUBX	Subtract with extend
JSR	Jump to subroutine	SWAP	Swap data register halves
LEA	Load effective address	TAS	Test and set operand
LINK	Link stack	TRAP	Trap
LSL	Logical shift left	TRAPV	Trap on overflow
LSR	Logical shift right	TST	Test
MOVE	Move	UNLK	Unlink
MOVEA	Move address		

Chapter 4
Memory Map

This chapter describes the MC68SZ328 memory map, which serves as a guide to all on-chip resources. When configuring the MC68SZ328, refer to Figure 4-1 and Table 4-1 on page 4-2, which is sorted by address.

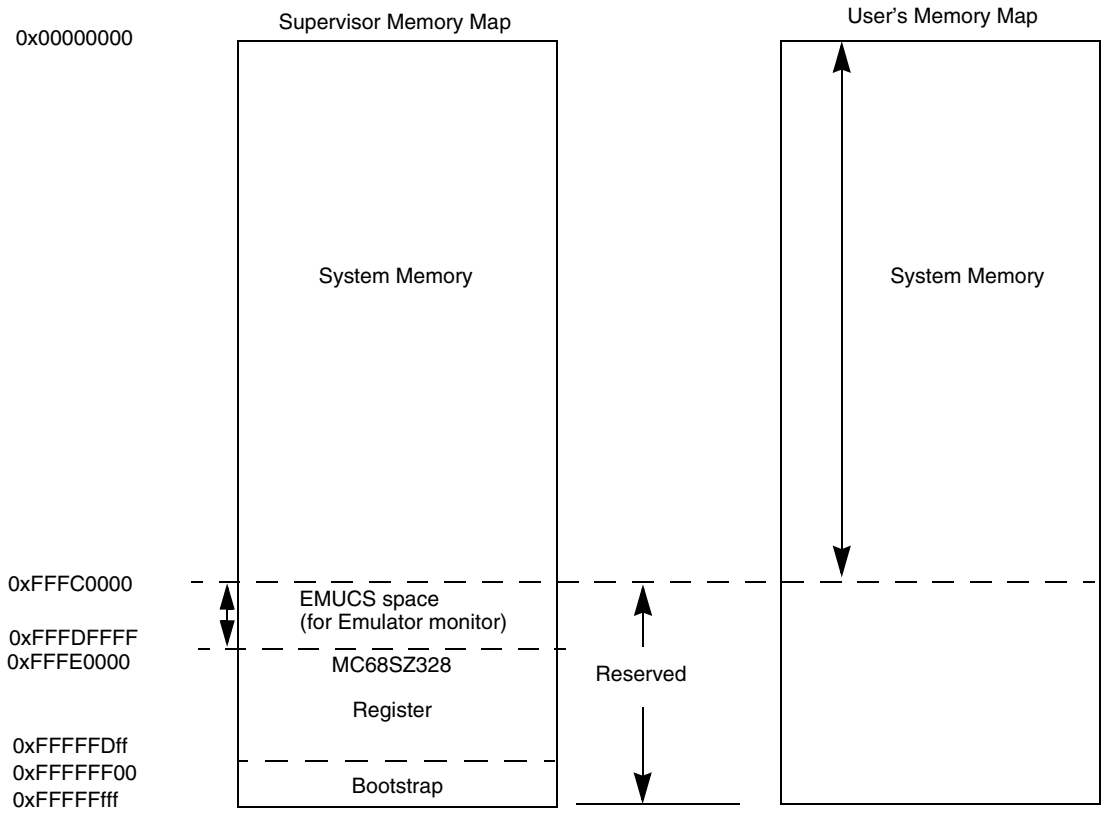


Figure 4-1. MC68SZ328 System Memory Map

4.1 Programmer's Memory Map

On reset (double mapped bit set) the base address used in the table is 0xFFFFF000 (or 0xXXFFF000, where XX is "don't care"). If the double-mapped bit is cleared in the System Control register, then the base address is 0xFFFFF000 only. Registers that can be reset by Normal Resets means that they can be reset by either a Power Up Reset or a Normal Reset. A Power Up Reset means that they can be reset by a Power Up Reset only.

NOTE:

Unpredictable results may occur if users write to any register space not documented in Table 4-1.

Table 4-1. Programmer's Memory Map (Sorted by Address)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFE0000	DCR	16	DMA Control Register	0x0000	9-6	Normal Reset
0xFFFE0002	DTSR	16	DMA Transfer Status Register	0x0000	9-6	Normal Reset
0xFFFE0004	DIMR	16	DMA Interrupt Mask Register	0x003F	9-7	Normal Reset
0xFFFE0006	DBTOSR	16	DMA Burst Time-Out Status Register	0x0000	9-8	Normal Reset
0xFFFE0008	DRTOSR	16	DMA Request Time-Out Status Register	0x0000	9-9	Normal Reset
0xFFFE000A		-	Reserved	-		
0xFFFE000C		-	Reserved	-		
0xFFFE000E	DBTOCR	16	DMA Burst Time-Out Control Register	0x0000	9-9	Normal Reset
0xFFFE0010-03F		-	Reserved	-		
0xFFFE0040	MSAR0	32	Memory Channel 0 Source Address Register	0x00000000	9-11	Normal Reset
0xFFFE0044	MDAR0	32	Memory Channel 0 Destination Address Register	0x00000000	9-12	Normal Reset
0xFFFE0048	MCNTR0	32	Memory Channel 0 Count Register	0x00000000	9-13	Normal Reset
0xFFFE004C	MCR0	16	Memory Channel 0 Control Register	0x0000	9-14	Normal Reset
0xFFFE004E	MBLR0	16	Memory Channel 0 Burst Length Register	0x0000	9-15	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFE0050	MBUCR0	16	Memory Channel 0 Bus Utilization Control Register	0x0000	9-16	Normal Reset
0xFFFE0052	BLR0	16	Memory Channel 0 Block Length Register	0x0000	9-17	Normal Reset
0xFFFE0054	SBSDR0	16	Memory Channel 0 Source Block Separation Distance Register	0x0000	9-17	Normal Reset
0xFFFE0056	MRTOR0	16	Memory Channel 0 DMA Request Time-Out Register	0x0000	9-18	Normal Reset
0xFFFE0058	DBSDR0	16	Memory Channel 0 Destination Block Separation Distance Register	0x0000	9-19	Normal Reset
0xFFFE005A-07F		-	Reserved	-		
0xFFFE0080	MSAR1	32	Memory Channel 1 Source Address Register	0x00000000	9-20	Normal Reset
0xFFFE0084	MDAR1	32	Memory Channel 1 Destination Address Register	0x00000000	9-21	Normal Reset
0xFFFE0088	MCNTR1	32	Memory Channel 1 Count Register	0x00000000	9-22	Normal Reset
0xFFFE008C	MCR1	16	Memory Channel 1 Control Register	0x0000	9-23	Normal Reset
0xFFFE008E	MBLR1	16	Memory Channel 1 Burst Length Register	0x0000	9-25	Normal Reset
0xFFFE0090	MBUCR1	16	Memory Channel 1 Bus Utilization Control Register	0x0000	9-16	Normal Reset
0xFFFE0092	BLR1	16	Memory Channel 1 Block Length Register	0x0000	9-17	Normal Reset
0xFFFE0094	SBSDR1	16	Memory Channel 1 Source Block Separation Distance Register	0x0000	9-17	Normal Reset
0xFFFE0096	MRTOR1	16	Memory Channel 1 DMA Request Time-Out Register	0x0000	9-27	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFE0098	DBSDR1	16	Memory Channel 1 Destination Block Separation Distance Register	0x0000	9-19	Normal Reset
0xFFFE009A-0BF		-	Reserved	-		t
0xFFFE00C0	IMAR2	32	I/O Channel 2 Memory Address Register	0x00000000	9-20	Normal Reset
0xFFFE00C4	IPAR2	32	I/O Channel 2 Peripheral Address Register	0x00000000	9-21	Normal Reset
0xFFFE00C8	ICNTR2	32	I/O Channel 2 Count Register	0x00000000	9-22	Normal Reset
0xFFFE00CC	ICR2	16	I/O Channel 2 Control Register	0x0000	9-14	Normal Reset
0xFFFE00CE	IRSSR2	16	I/O Channel 2 Request Source Select Register	0x0000	9-25	Normal Reset
0xFFFE00D0	IBLR2	16	I/O Channel 2 Burst Length Register	0x0000	9-25	Normal Reset
0xFFFE00D2	IRTOR2	16	I/O Channel 2 DMA Request Time-Out Register	0x0000	9-27	Normal Reset
0xFFFE00D4-0FF		-	Reserved	0X0		
0xFFFE0100	IMAR3	32	I/O Channel 3 Memory Address Register	0x00000000	9-20	Normal Reset
0xFFFE0104	IPAR3	32	I/O Channel 3 Peripheral Address Register	0x00000000	9-21	Normal Reset
0xFFFE0108	ICNTR3	32	I/O Channel 3 Count Register	0x00000000	9-22	Normal Reset
0xFFFE010C	ICR3	16	I/O Channel 3 Control Register	0x0000	9-23	Normal Reset
0xFFFE010E	IRSSR3	16	I/O Channel 3 Request Source Select Register	0x0000	9-25	Normal Reset
0xFFFE0110	IBLR3	16	I/O Channel 3 Burst Length Register	0x0000	9-25	Normal Reset
0xFFFE0112	IRTOR3	16	I/O Channel 3 DMA Request Time-Out Register	0x0000	9-27	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFE0114-13F		-	Reserved	0X0		
0xFFFE0140	IMAR4	32	I/O Channel 4 Memory Address Register	0x00000000	9-20	Normal Reset
0xFFFE0144	IPAR4	32	I/O Channel 4 Peripheral Address Register	0x00000000	9-21	Normal Reset
0xFFFE0148	ICNTR4	32	I/O Channel 4 Count Register	0x00000000	9-22	Normal Reset
0xFFFE014C	ICR4	16	I/O Channel 4 Control Register	0x0000	9-23	Normal Reset
0xFFFE014E	IRSSR4	16	I/O Channel 4 Request Source Select Register	0x0000	9-25	Normal Reset
0xFFFE0150	IBLR4	16	I/O Channel 4 Burst Length Register	0x0000	9-25	Normal Reset
0xFFFE0152	IRTOR4	16	I/O Channel 4 DMA Request Time-Out Register	0x0000	9-27	Normal Reset
0xFFFE0154-17F		-	Reserved	0X0		
0xFFFE0180	IMAR5	32	I/O Channel 5 Memory Address Register	0x00000000	9-20	Normal Reset
0xFFFE0184	IPAR5	32	I/O Channel 5 Peripheral Address Register	0x00000000	9-21	Normal Reset
0xFFFE0188	ICNTR5	32	I/O Channel 5 Count Register	0x00000000	9-22	Normal Reset
0xFFFE018C	ICR5	16	I/O Channel 5 Control Register	0x0000	9-23	Normal Reset
0xFFFE018E	IRSSR5	16	I/O Channel 5 Request Source Select Register	0x0000	9-25	Normal Reset
0xFFFE0190	IBLR5	16	I/O Channel 5 Burst Length Register	0x0000	9-25	Normal Reset
0xFFFE0192	IRTOR5	16	I/O Channel 5 DMA Request Time-Out Register	0x0000	9-27	Normal Reset
0xFFFE0194-1FF		-	Reserved	0X0		
0xFFFE0200	ASP_PADFIFO	32	Pen Sample FIFO Register	0x00000000	11-19	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFE0204	ASP_EADFIFO	32	Enhanced ADC Register	0x00000000	11-20	Normal Reset
0xFFFE020C	ASP_EADCOEF	32	Enhanced ADC FIR Coefficients RAM Register	0x00000000	11-20	Normal Reset
0xFFFE0210	ASP_ACNTLCR	32	ASP Control Register	0x03040000	11-12	Normal Reset
0xFFFE0214	ASP_PSMPLRG	32	Pen ADC Sample Rate Control Register	0x00000000	11-14	Normal Reset
0xFFFE0218	ASP_ICNTLR	32	Interrupt Control Register	0x00000000	11-16	Normal Reset
0xFFFE021C	ASP_ISTATR	32	Interrupt/Error Status Register	0x00000000	11-17	Normal Reset
0xFFFE0220	ASP_EADGAIN	32	Enhanced ADC Control Register	0x00000000	11-20	Normal Reset
0xFFFE0226	-	-	Reserved (Do not write to this register.)	-		
0xFFFE022C	ASP_CLKDIVO	32	Clock Divide Register	0x00000000	11-21	Normal Reset
0xFFFE0230	ASPCMPCNTL	32	Compare Control register	0x00000000	11-15	Normal Reset
0xFFFE0300	STR_STP_CLK	16	MMC/SD Clock Control Register	0x0000	17-25	Normal Reset
0xFFFE0304	STATUS	16	MMC/SD Status Register	0x0000	17-26	Normal Reset
0xFFFE0308	CLK_RATE	16	MMC/SD Clock Rate Register	0x0036	17-28	Normal Reset
0xFFFE030C	-	-	Reserved	-		
0xFFFE0310	CMD_DAT_CONT	16	MMC/SD Command and Data Control Register	0x0000	17-29	Normal Reset
0xFFFE0314	RES_TO	16	MMC/SD Response Time Out Register	0x0040	17-30	Normal Reset
0xFFFE0318	READ_TO	16	MMC/SD Read Time Out Register	0xFFFF	17-31	Normal Reset
0xFFFE031C	BLK_LEN	16	MMC/SD Block Length Register	0x0000	17-31	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFE0320	NOB	16	MMC/SD Number of Blocks Register	0x0000	17-32	Normal Reset
0xFFFE0324	REV_NO	16	MMC/SD Revision Number Register	0x0380	17-32	Normal Reset
0xFFFE0326	-	-	Reserved	-		
0xFFFE0328	INT_MASK	16	MMC/SD Interrupt Mask Register	0x0000	17-33	Normal Reset
0xFFFE032C	CMD	16	MMC/SD Command Number Register	0x0000	17-33	Normal Reset
0xFFFE0330	ARGUMENTH	16	MMC/SD Higher Argument Register	0x0000	17-34	Normal Reset
0xFFFE0332	ARGUMENTL	16	MMC/SD Lower Argument Register	0x0000	17-34	Normal Reset
0xFFFE0334	RES_FIFO	16	MMC/SD Response FIFO Register	0x0000	17-34	Normal Reset
0xFFFE0338	BUFFER_ACCESS	16	MMC/SD Buffer Access Register	0xXXXX	17-35	Normal Reset
0xFFFE033C	BUF_PART_FULL	16	MMC/SD Buffer Part Full Register	0x0000	17-35	Normal Reset
0xFFFE0400	USB_FRAME	32	USB Frame Number and Match Register	0x00000000	21-12	Normal Reset
0xFFFE0404	USB_SPEC	32	USB Specification/Release Number Register	0x00001010	21-13	Normal Reset
0xFFFE0408	USB_STAT	32	USB Status Register	0x00000000	21-13	Normal Reset
0xFFFE040C	USB_CTRL	32	USB Control Register	0x00000010	21-14	Normal Reset
0xFFFE0410	USB_CFGSTAT	32	USB Configuration Status Register	0x80000000	21-16	Normal Reset
0xFFFE0414	USB_DDAT	32	USB Endpoint Buffer Data Register	0x000000XX	21-17	Normal Reset
0xFFFE0418	USB_GEN_ISR	32	USB General Interrupt Status Register	0x00000000	21-18	Normal Reset
0xFFFE041C	USB_MASK	32	USB General Interrupt Mask Register	0x800000FF	21-20	Normal Reset
0xFFFE0420	RESERVED	32	Reserved	-	-	

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFE0424	USB_ENAB	32	USB Enable Register	0x00000000	21-21	Normal Reset
0xFFFE0428	USB_ISR	32	USB Interrupt Status Register	0x00000000	21-22	Normal Reset
0xFFFE042C	RESERVED	32	Reserved	-	-	
0xFFFE0430	USB_EP0_STATCR	32	USB Endpoint 0 Status/Control Register	0x00000000	21-23	Normal Reset
0xFFFE0434	USB_EP0_ISR	32	USB Endpoint 0 Interrupt Status Register	0x00000080	21-24	Normal Reset
0xFFFE0438	USB_EP0_MASK	32	USB Endpoint 0 Interrupt Mask Register	0x000001FF	21-26	Normal Reset
0xFFFE043C	USB_EP0_FDAT	16	USB Endpoint 0 FIFO Data Register	0x0000	21-27	Normal Reset
0xFFFE0440	USB_EP0_FSTAT	32	USB Endpoint 0 FIFO Status Register	0x00010000	21-28	Normal Reset
0xFFFE0444	USB_EP0_FCTRL	32	USB Endpoint 0 FIFO Control Register	0x01000000	21-31	Normal Reset
0xFFFE0448	USB_EP0_LRFP	32	USB Endpoint 0 FIFO Last Read Frame Pointer Register	0x00000000	21-32	Normal Reset
0xFFFE044C	USB_EP0_LWFP	32	USB Endpoint 0 FIFO Last Write Frame Pointer	0x00000000	21-33	Normal Reset
0xFFFE0450	USB_EP0_FALRM	32	USB Endpoint 0 FIFO Alarm Register	0x00000000	21-34	Normal Reset
0xFFFE0454	USB_EP0_FRDP	32	USB Endpoint 0 FIFO Read Pointer Register	0x00000000	21-35	Normal Reset
0xFFFE0458	USB_EP0_FWRP	32	USB Endpoint 0 FIFO Write Pointer Register	0x00000000	21-36	Normal Reset
0xFFFE045C	RESERVED	32	Reserved	-	-	
0xFFFE0460	USB_EP1_STATCR	32	USB Endpoint 1 Status/Control Register	0x00000000	21-23	Normal Reset
0xFFFE0464	USB_EP1_ISR	32	USB Endpoint 1 Interrupt Status Register	0x00000080	21-24	Normal Reset
0xFFFE0468	USB_EP1_MASK	32	USB Endpoint 1 Interrupt Mask Register	0x000001FF	21-26	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFE046C	USB_EP1_FDAT	16	USB Endpoint 1 FIFO Data Register	0x0000	21-27	Normal Reset
0xFFFE0470	USB_EP1_FSTAT	32	USB Endpoint 1 FIFO Status Register	0x00010000	21-28	Normal Reset
0xFFFE0474	USB_EP1_FCTRL	32	USB Endpoint 1 FIFO Control Register	0x01000000	21-31	Normal Reset
0xFFFE0478	USB_EP1_LRFP	32	USB Endpoint 1 FIFO Last Read Frame Pointer Register	0x00000000	21-32	Normal Reset
0xFFFE047C	USB_EP1_LWFP	32	USB Endpoint 1 FIFO Last Write Frame Pointer Register	0x00000000	21-33	Normal Reset
0xFFFE0480	USB_EP1_FALRM	32	USB Endpoint 1 FIFO Alarm Register	0x00000000	21-34	Normal Reset
0xFFFE0484	USB_EP1_FRDP	32	USB Endpoint 1 FIFO Read Pointer Register	0x00000000	21-35	Normal Reset
0xFFFE0488	USB_EP1_FWRP	32	USB Endpoint 1 FIFO Write Pointer Register	0x00000000	21-36	Normal Reset
0xFFFE048C	RESERVED	32	Reserved	-		
0xFFFE0490	USB_EP2_STATCR	32	USB Endpoint 2 Status/Control Register	0x00000000	21-23	Normal Reset
0xFFFE0494	USB_EP2_ISR	32	USB Endpoint 2 Interrupt Status Register	0x00000080	21-24	Normal Reset
0xFFFE0498	USB_EP2_MASK	32	USB Endpoint 2 Interrupt Mask Register	0x000001FF	21-26	Normal Reset
0xFFFE049C	USB_EP2_FDAT	16	USB Endpoint 2 FIFO Data Register	0x0000	21-27	Normal Reset
0xFFFE04A0	USB_EP2_FSTAT	32	USB Endpoint 2 FIFO Status Register	0x00010000	21-28	Normal Reset
0xFFFE04A4	USB_EP2_FCTRL	32	USB Endpoint 2 FIFO Control Register	0x01000000	21-31	Normal Reset
0xFFFE04A8	USB_EP2_LRFP	32	USB Endpoint 2 FIFO Last Read Frame Pointer Register	0x00000000	21-32	Normal Reset
0xFFFE04AC	USB_EP2_LWFP	32	USB Endpoint 2 FIFO Last Write Frame Pointer Register	0x00000000	21-33	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFE04B0	USB_EP2_FALRM	32	USB Endpoint 2 FIFO Alarm Register	0x00000000	21-34	Normal Reset
0xFFFE04B4	USB_EP2_FRDP	32	USB Endpoint 2 FIFO Read Pointer Register	0x00000000	21-35	Normal Reset
0xFFFE04B8	USB_EP2_FWRP	32	USB Endpoint 2 FIFO Write Pointer Register	0x00000000	21-36	Normal Reset
0xFFFE04BC	RESERVED	32	Reserved	-		t
0xFFFE04C0	USB_EP3_STATCR	32	USB Endpoint 3 Status/Control Register	0x00000000	21-23	Normal Reset
0xFFFE04C4	USB_EP3_ISR	32	USB Endpoint 3 Interrupt Status Register	0x00000080	21-24	Normal Reset
0xFFFE04C8	USB_EP3_MASK	32	USB Endpoint 3 Interrupt Mask Register	0x000001FF	21-26	Normal Reset
0xFFFE04CC	USB_EP3_FDAT	16	USB Endpoint 3 FIFO Data Register	0x0000	21-27	Normal Reset
0xFFFE04D0	USB_EP3_FSTAT	32	USB Endpoint 3 FIFO Status Register	0x00010000	21-28	Normal Reset
0xFFFE04D4	USB_EP3_FCTRL	32	USB Endpoint 3 FIFO Control Register	0x01000000	21-31	Normal Reset
0xFFFE04D8	USB_EP3_LRFP	32	USB Endpoint 3 FIFO Last Read Frame Pointer Register	0x00000000	21-32	Normal Reset
0xFFFE04DC	USB_EP3_LWFP	32	USB Endpoint 3 FIFO Last Write Frame Pointer Register	0x00000000	21-33	Normal Reset
0xFFFE04E0	USB_EP3_FALRM	32	USB Endpoint 3 FIFO Alarm Register	0x00000000	21-34	Normal Reset
0xFFFE04E4	USB_EP3_FRDP	32	USB Endpoint 3 FIFO Read Pointer Register	0x00000000	21-35	Normal Reset
0xFFFE04E8	USB_EP3_FWRP	32	USB Endpoint 3 FIFO Write Pointer Register	0x00000000	21-36	Normal Reset
0xFFFE04EC	RESERVED	32	Reserved	-		
0xFFFE04F0	USB_EP4_STATCR	32	USB Endpoint 4 Status/Control Register	0x00000000	21-23	Normal Reset
0xFFFE04F4	USB_EP4_ISR	32	USB Endpoint 4 Interrupt Status Register	0x00000080	21-24	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFE04F8	USB_EP4_MASK	32	USB Endpoint 4 Interrupt Mask Register	0x000001FF	21-26	Normal Reset
0xFFFE04FC	USB_EP4_FDAT	16	USB Endpoint 4 FIFO Data Register	0x0000	21-27	Normal Reset
0xFFFE0500	USB_EP4_FSTAT	32	USB Endpoint 4 FIFO Status Register	0x00010000	21-28	Normal Reset
0xFFFE0504	USB_EP4_FCTRL	32	USB Endpoint 4 FIFO Control Register	0x01000000	21-31	Normal Reset
0xFFFE0508	USB_EP4_LRFP	32	USB Endpoint 4 FIFO Last Read Frame Pointer Register	0x00000000	21-32	Normal Reset
0xFFFE050C	USB_EP4_LWFP	32	USB Endpoint 4 FIFO Last Write Frame Pointer Register	0x00000000	21-33	Normal Reset
0xFFFE0510	USB_EP4_FALRM	32	USB Endpoint 4 FIFO Alarm Register	0x00000000	21-34	Normal Reset
0xFFFE0514	USB_EP4_FRDP	32	USB Endpoint 4 FIFO Read Pointer Register	0x00000000	21-35	Normal Reset
0xFFFE0518	USB_EP4_FWRP	32	USB Endpoint 4 FIFO Write Pointer Register	0x00000000	21-36	Normal Reset
0xFFFE051C	RESERVED	32	Reserved	-	-	
0xFFFE0600	MSCMD	16	Memory Stick Command Register	0x0000	18-14	Normal Reset
0xFFFE0602	MSCS	16	Memory Stick Control/Status Register	0x050A	18-15	Normal Reset
0xFFFE0604	MSTDATA	16	Memory Stick Transmit FIFO Data Register	0x0000	18-16	Normal Reset
0xFFFE0604	MSRDATA	16	Memory Stick Receive FIFO Data Register	0x0000	18-17	Normal Reset
0xFFFE0606	MSICS	16	Memory Stick Interrupt Control/Status Register	0x0000	18-18	Normal Reset
0xFFFE0608	MSPPCD	16	Memory Stick Parallel Port Control/Data Register	0x0000	18-21	Normal Reset
0xFFFE060A	MSC2	16	Memory Stick Control 2 Register	0x0000	18-22	Normal Reset
0xFFFE060C	MSACD	16	Memory Stick Auto Command Register	0x7001	18-23	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFE060E	MSFAECS	16	Memory Stick FIFO Access Error Control/Status Register	0x0000	18-24	Normal Reset
0xFFFE0610	MSCLKD	16	Memory Stick SCLK Divider Control Register	0x0002	18-25	Normal Reset
0xFFFE0612	MSDRQC	16	Memory Stick DMA Request Control Register	0x0000	18-25	Normal Reset
0xFFFE0800	LSSA	32	LCD Screen Start Address Register	0x00000000	10-17	Normal Reset
0xFFFE0804	LSS	16	LCD Screen Size Register	0x0000	10-17	Normal Reset
0xFFFE0806	LVPW	16	LCD Virtual Page Width Register	0x0000	10-18	Normal Reset
0xFFFE0808	LCXP	16	LCD Cursor X Position Register	0x0000	10-24	Normal Reset
0xFFFE080A	LCYP	16	LCD Cursor Y Position Register	0x0000	10-25	Normal Reset
0xFFFE080C	LCSR	16	LCD Cursor Size Register	0x0101	10-26	Normal reset
0xFFFE080E	LBLKC	16	LCD Blink Control Register	0x00FF	10-27	Normal Reset
0xFFFE0810	LCUR_COL	16	LCD Color Cursor Mapping Register	0x0000	10-27	Normal reset
0xFFFE0812	LPCON0	16	LCD Panel Configuration Register 0	0x0000	10-19	Normal Reset
0xFFFE0814	LPCON1	16	LCD Panel Configuration Register 1	0x0000	10-19	Normal Reset
0xFFFE0816	LHCON0	16	LCD Horizontal Configuration Register 0	0x0000	10-21	Normal Reset
0xFFFE0818	LHCON1	16	LCD Horizontal Configuration Register 1	0x0400	10-21	Normal Reset
0xFFFE081A	LVCON0	16	LCD Vertical Configuration Register 0	0x0000	10-22	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFE081C	LVCON1	16	LCD Vertical Configuration Register 1	0x0401	10-22	Normal Reset
0xFFFE081E	LPANOFF	16	LCD Panning Offset Register	0x0000	10-24	Normal Reset
0xFFFE0820	LGPMR	16	LCD Gray Palette Mapping Register	0x0073	10-28	Normal Reset
0xFFFE0822	PWMR	16	PWM Contrast Control Register	0x0000	10-28	Normal Reset
0xFFFE0824	LDMACR	16	LCD DMA Control Register	0x0404	10-30	Normal Reset
0xFFFE0826	RMCR	16	Refresh Mode Control Register	0x0000	10-29	Normal Reset
0xFFFE0828	LICFR	16	LCD Interrupt Configuration Register	0x0000	10-31	Normal Reset
0xFFFE082A	LISR	16	LCD Interrupt Status Register	0x0000	10-32	Normal Reset
0xFFFE0A00-0xFFFE0BFF	-	-	LCD Mapping Ram Registers	-	10-33	Normal Reset
0xFFFFF000	SCR	8	System Control Register	0x1C	6-2	Normal Reset
0xFFFFF003	PCR	8	Peripheral Control Register	0x00	6-3	Power-up reset
0xFFFFF004	IDR	32	Silicon Id Register	0x53100000	6-4	Can Not Reset
0xFFFFF008	IODCR	16	I/O Drive Control Register	0x0000	6-5	Power-up reset
0xFFFFF100	CSGBA	16	Chip-Select Group A Base Register	0x0000	7-4	Normal Reset
0xFFFFF102	CSGBB	16	Chip-Select Group B Base Register	0x0000	7-4	Normal Reset
0xFFFFF104	CSGBC	16	Chip-Select Group C Base Register	0x0000	7-4	Normal Reset
0xFFFFF106	CSGBD	16	Chip-Select Group D Base Register	0x0000	7-4	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFFF10C	CSCTRL2	16	Chip-Select Control Register 2	0x0000	7-23	Normal Reset
0xFFFFF110	CSA	16	Chip-Select Register A	0x00B0	7-7	Normal Reset
0xFFFFF112	CSB	16	Chip-Select Register B	0x0000	7-7	Normal Reset
0xFFFFF114	CSC	16	Chip-Select Register C	0x0000	7-7	Normal Reset
0xFFFFF116	CSD	16	Chip-Select Register D	0x0200	7-7	Normal Reset
0xFFFFF118	EMUCS	16	Emulation Chip-Select Register	0x0060	7-20	Normal Reset
0xFFFFF150	CSCTRL3	16	Chip-Select Control Register 3	0x9C00	7-23	Normal Reset
0xFFFFF180	CSGBE	16	Chip-Select Group E Base Register	0x0000	7-4	Normal Reset
0xFFFFF182	CSGBF	16	Chip-Select Group F Base Register	0x0000	7-4	Normal Reset
0xFFFFF184	CSGBG	16	Chip-Select Group G Base Register	0x0000	7-4	Normal Reset
0xFFFFF18A	CSCTRL1	16	Chip-Select Control Register 1	0x0000	7-21	Normal Reset
0xFFFFF190	CSE	16	Chip-Select Register E	0x0000	7-7	Power - up Reset except bit 0 is Normal Reset
0xFFFFF192	CSF	16	Chip-Select Register F	0x0000	7-7	Power - up Reset except bit 0 is Normal Reset
0xFFFFF194	CSG	16	Chip-Select Register G	0x0000	7-7	Normal Reset
0xFFFFF200	PLLCR	16	PLL Control Register	0x2404	5-10	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFFF202	MPFSR0	16	MCU PLL Frequency Select Register 0	0x3CE8	5-11	Normal Reset
0xFFFFF204	MPFSR1	16	MCU PLL Frequency Select Register 1	0x18FF	5-12	Normal Reset
0xFFFFF207	PCTLR	8	CPU Power Control Register	0x1F	5-9	Normal Reset
0xFFFFF208	UPFSR0	16	USBPLL Frequency Select Register 0	0x2C01	5-13	Normal Reset
0xFFFFF20A	UPFSR1	16	USBPLL Frequency Select Register 1	0x0000	5-14	Normal Reset
0xFFFF20C	CSCR	16	Clock Sources Control Register	0x8903	5-15	Normal Reset
0xFFFFF300	IVR	8	Interrupt Vector Register	0x00	15-7	Normal Reset
0xFFFFF302	ICR	16	Interrupt Control Register	0x0000	15-8	Normal Reset
0xFFFFF304	IMR	32	Interrupt Mask Register	0xFFFFFFFF	15-10	Normal Reset
0xFFFFF308	RES	32	Reserved	—	—	
0xFFFFF30C	ISR	32	Interrupt Status Register	0x00000000	15-12	Normal Reset
0xFFFFF310	IPR	32	Interrupt Pending Register	0x00000000	15-17	Normal Reset
0xFFFFF314	ILCR	16	Interrupt Level Control Register 1	0x6533	15-21	Normal Reset
0xFFFFF316	ILCR2	16	Interrupt Level Control Register 2	0x4533	15-21	Normal Reset
0xFFFFF318	ILCR3	16	Interrupt Level Control Register 3	0x4666	15-21	Normal Reset
0xFFFFF31A	ILCR4	16	Interrupt Level Control Register 4	0x4444	15-21	Normal Reset
0xFFFFF31C	ILCR5	16	Interrupt Level Control Register 5	0x4444	15-21	Normal Reset
0xFFFFF31E	ILCR6	16	Interrupt Level Control Register 6	0x4444	15-21	Normal Reset
0xFFFF320	ILCR7	16	Interrupt Level Control Register 7	0x0646	15-21	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFFF403 0xFFFFF407	RES	8	Reserved	—		
0xFFFFF408	PBDIR	8	Port B Direction Register	0x00	16-6	Normal Reset
0xFFFFF409	PBDATA	8	Port B Data Register	0xFF	16-7	Normal Reset
0xFFFFF40A	PBPUEN	8	Port B Pull-up Enable Register	0xFF	16-8	Normal Reset
0xFFFFF40B	PBSEL	8	Port B Select Register	0xFF	16-8	Normal Reset
0xFFFFF410	PCDIR	8	Port C Direction Register	0x00	16-9	Normal Reset
0xFFFFF411	PCDATA	8	Port C Data Register	0x00	16-10	Normal Reset
0xFFFFF412	PCPUEN	8	Port C Pull-Up Enable Register	0xFF	16-11	Normal Reset
0xFFFFF413	PCSEL	8	Port C Select Register	0xFF	16-11	Normal Reset
0xFFFFF418	PDDIR	8	Port D Direction Register	0x00	16-12	Normal Reset
0xFFFFF419	PDDATA	8	Port D Data Register	0xFF	16-13	Normal Reset
0xFFFFF41A	PDPUEN	8	Port D Pull-up Enable Register	0xFF	16-14	Normal Reset
0xFFFFF41B	PDSEL	8	Port D Select Register	0xFF	16-14	Normal Reset
0xFFFFF41C	PDIMR	8	Port D Interrupt Mask Register	0x00	16-15	Normal Reset
0xFFFFF41D	PDISR	8	Port D Interrupt Status Register	0x00	16-15	Normal Reset
0xFFFFF41E	PDIER	8	Port D Interrupt Edge Register	0x00	16-16	Normal Reset
0xFFFFF41F	PDIPR	8	Port D Interrupt Polarity Register	0x00	16-16	Normal Reset
0xFFFFF420	PEDIR	8	Port E Direction Register	0x00	16-17	Normal Reset
0xFFFFF421	PEDATA	8	Port E Data Register	0xFF	16-17	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFFF422	PEPUEN	8	Port E Pull-up Enable Register	0xFF	16-18	Normal Reset
0xFFFFF423	PESEL	8	Port E Select Register	0xFF	16-19	Normal Reset
0xFFFFF424	PEIMR	8	Port E Interrupt Mask Register	0x00	16-19	Normal Reset
0xFFFFF425	PEISR	8	Port E Interrupt Status Register	0x00	16-20	Normal Reset
0xFFFFF426	PEIER	8	Port E Interrupt Edge Register	0x00	16-20	Normal Reset
0xFFFFF427	PEIPR	8	Port E Interrupt Polarity Register	0x00	16-21	Normal Reset
0xFFFFF428	PFDIR	8	Port F Direction Register	0x00	16-22	Normal Reset
0xFFFFF429	PFDATA	8	Port F Data Register	0xFF	16-22	Normal Reset
0xFFFFF42A	PFPUEN	8	Port F Pull-up/Pull-down Enable Register	0xFF	16-24	Normal Reset
0xFFFFF42B	PFSEL	8	Port F Select Register	0x87	16-24	Normal Reset
0xFFFFF42C	PFIMR	8	Port F Interrupt Mask Register	0x00	16-24	Normal Reset
0xFFFFF42D	PFISR	8	Port F Interrupt Status Register	0x00	16-25	Normal Reset
0xFFFFF42E	PFIER	8	Port F Interrupt Edge Register	0x00	16-25	Normal Reset
0xFFFFF42F	PFIPR	8	Port F Interrupt Polarity Register	0x00	16-26	Normal Reset
0xFFFFF430	PGDIR	8	Port G Direction Register	0x00	16-27	Normal Reset
0xFFFFF431	PGDATA	8	Port G Data Register	0x3F	16-27	Normal Reset
0xFFFFF432	PGPUEN	8	Port G Pull-up Enable Register	0x3D	16-29	Normal Reset
0xFFFFF433	PGSEL	8	Port G Select Register	0x08	16-29	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFFF434	PGIMR	8	Port G Interrupt Mask Register	0x00	16-30	Normal Reset
0xFFFFF435	PGISR	8	Port G Interrupt Status Register	0x00	16-30	Normal Reset
0xFFFFF436	PGIER	8	Port G Interrupt Edge Register	0x00	16-31	Normal Reset
0xFFFFF437	PGIPR	8	Port G Interrupt Polarity Register	0x00	16-31	Normal Reset
0xFFFFF438	PJDIR	8	Port J Direction Register	0x00	16-32	Normal Reset
0xFFFFF439	PJDATA	8	Port J Data Register	0xFF	16-33	Power Up Reset
0xFFFFF43A	PJPUEN	8	Port J Pull-up Enable Register	0xFF	16-34	Power Up Reset
0xFFFFF43B	PJSEL	8	Port J Select Register	0xEF	16-34	Power Up Reset
0xFFFFF43C	PJIMR	8	Port J Interrupt Mask Register	0x00	16-35	Power Up Reset
0xFFFFF43D	PJISR	8	Port J Interrupt Status Register	0x00	16-35	Power Up Reset
0xFFFFF43E	PJIER	8	Port J Interrupt Edge Register	0x00	16-36	Power Up Reset
0xFFFFF43F	PJIPR	8	Port J Interrupt Polarity Register	0x00	16-36	Power Up Reset
0xFFFFF440	PKDIR	8	Port K Direction Register	0x00	16-37	Power Up Reset
0xFFFFF441	PKDATA	8	Port K Data Register	0x0F	16-37	Normal Reset
0xFFFFF442	PKPUEN	8	Port K Pull-up/Pull-down Enable Register	0xFF	16-39	Normal Reset
0xFFFFF443	PKSEL	8	Port K Select Register	0x00	16-39	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFFF444	PKIMR	8	Port K Interrupt Mask Register	0x00	16-40	Normal Reset
0xFFFFF445	PKISR	8	Port K Interrupt Status Register	0x00	16-40	Normal Reset
0xFFFFF446	PKIER	8	Port K Interrupt Edge Register	0x00	16-41	Normal Reset
0xFFFFF447	PKIPR	8	Port K Interrupt Polarity Register	0x00	16-41	Normal Reset
0xFFFFF448	PMDIR	8	Port M Direction Register	0x00	16-42	Power -up Reset
0xFFFFF449	PMDATA	8	Port M Data Register	0x20	16-43	Power -up Reset
0xFFFFF44A	PMPUEN	8	Port M Pull-up/Pull-down Enable Register	0x3F	16-44	Power -up Reset
0xFFFFF44B	PMSEL	8	Port M Select Register	0x3F	16-45	Power -up Reset
0xFFFFF44C	PMIMR	8	Port M Interrupt Mask Register	0x00	16-45	Power -up Reset
0xFFFFF44D	PMISR	8	Port M Interrupt Status Register	0x00	16-46	Power -up Reset
0xFFFFF44E	PMIER	8	Port M Interrupt Edge Register	0x00	16-46	Power -up Reset
0xFFFFF44F	PMIPR	8	Port M Interrupt Polarity Register	0x00	16-47	Power -up Reset
0xFFFFF450	PNDIR	8	Port N Direction Register	0x00	16-48	Normal Reset
0xFFFFF451	PNDATA	8	Port N Data Register	0x0F	16-48	Normal Reset
0xFFFFF452	PNPUEN	8	Port N Pull-up/Pull-down Enable Register	0xFF	16-49	Normal Reset
0xFFFFF453	PNSEL	8	Port N Select Register	0xFF	16-50	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFFF454	PNIMR	8	Port N Interrupt Mask Register	0x00	16-50	Normal Reset
0xFFFFF455	PNISR	8	Port N Interrupt Status Register	0x00	16-51	Normal Reset
0xFFFFF456	PNIER	8	Port N Interrupt Edge Register	0x00	16-51	Normal Reset
0xFFFFF457	PNIPR	8	Port N Interrupt Polarity Register	0x00	16-52	Normal Reset
0xFFFFF458	PPDIR	8	Port P Direction Register	0x00	16-53	Normal Reset
0xFFFFF459	PPDATA	8	Port P Data Register	0x0F	16-53	Normal Reset
0xFFFFF45A	PPPUEN	8	Port P Pull-up/Pull-down Enable Register	0xFF	16-54	Normal Reset
0xFFFFF45B	PPSEL	8	Port P Select Register	0xFF	16-55	Normal Reset
0xFFFFF45C	PPIMR	8	Port P Interrupt Mask Register	0x00	16-55	Normal Reset
0xFFFFF45D	PPISR	8	Port P Interrupt Status Register	0x00	16-56	Normal Reset
0xFFFFF45E	PPIER	8	Port P Interrupt Edge Register	0x00	16-56	Normal Reset
0xFFFFF45F	PPIPR	8	Port P Interrupt Polarity Register	0x00	16-57	Normal Reset
0xFFFFF460	PRDIR	8	Port R Direction Register	0x00	16-58	Normal Reset
0xFFFFF461	PRDATA	8	Port R Data Register	0x0F	16-58	Normal Reset
0xFFFFF462	PRPUEN	8	Port R Pull-up/Pull-down Enable Register	0xFF	16-59	Normal Reset
0xFFFFF463	PRSEL	8	Port R Select Register	0xFF	16-60	Normal Reset
0xFFFFF464	PRIMR	8	Port R Interrupt Mask Register	0x00	16-60	Normal Reset
0xFFFFF465	PRISR	8	Port R Interrupt Status Register	0x00	16-61	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFFF466	PRIER	8	Port R Interrupt Edge Register	0x00	16-61	Normal Reset
0xFFFFF467	PRIPR	8	Port R Interrupt Polarity Register	0x00	16-62	Normal Reset
0xFFFFF500	PWMC1	16	PWM Unit 1 Control Register	0x0020	14-4	Normal Reset
0xFFFFF502	PWMS1	16	PWM Unit 1 Sample Register	0xXXXX	14-6	Normal Reset
0xFFFFF504	PWMP1	8	PWM Unit 1 Period Register	0xFE	14-7	Normal Reset
0xFFFFF505	PWMCNT1	8	PWM Unit 1 Counter Register	0x00	14-7	Normal Reset
0xFFFFF506	RES	16	Reserved	—	—	
0xFFFFF510	PWMC2	16	PWM Unit 2 Control Register	0x0000	14-8	Normal Reset
0xFFFFF512	PWMP2	16	PWM Unit 2 Period Register	0x0000	14-9	Normal Reset
0xFFFFF514	PWMW2	16	PWM Unit 2 Width Control Register	0x0000	14-9	Normal Reset
0xFFFFF516	PWMCNT2	16	PWM Unit 2 Counter Register	0x0000	14-10	Normal Reset
0xFFFFF600	TCTL1	16	Timer Unit 1 Control Register	0x0000	12-4	Normal Reset
0xFFFFF602	TPRER1	16	Timer Unit 1 Prescaler Register	0x0000	12-6	Normal Reset
0xFFFFF604	TCMP1	16	Timer Unit 1 Compare Register	0xFFFF	12-7	Normal Reset
0xFFFFF606	TCR1	16	Timer Unit 1 Capture Register	0x0000	12-8	Normal Reset
0xFFFFF608	TCN1	16	Timer Unit 1 Counter Register	0x0000	12-9	Normal Reset
0xFFFFF60A	TSTAT1	16	Timer Unit 1 Status Register	0x0000	12-10	Normal Reset
0xFFFFF610	TCTL2	16	Timer Unit 2 Control Register	0x0000	12-4	Normal Reset
0xFFFFF612	TPRER2	16	Timer Unit 2 Prescaler Register	0x0000	12-6	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFFF614	TCMP2	16	Timer Unit 2 Compare Register	0xFFFF	12-7	Normal Reset
0xFFFFF616	TCR2	16	Timer Unit 2 Capture Register	0x0000	12-8	Normal Reset
0xFFFFF618	TCN2	16	Timer Unit 2 Counter Register	0x0000	12-9	Normal Reset
0xFFFFF61A	TSTAT2	16	Timer Unit 2 Status Register	0x0000	12-10	Normal Reset
0xFFFFF700	SPIRXD	16	CSPI Receive Data Register	0x0000	20-3	Normal Reset
0xFFFFF702	SPITXD	16	CSPI Transmit Data Register	0x0000	20-4	Normal Reset
0xFFFFF704	SPICONT	16	CSPI Control/Status Register	0x0000	20-5	Normal Reset
0xFFFFF706	SPIINTCS	16	CSPI Interrupt Control/Status Register	0x0000	20-7	Normal Reset
0xFFFFF708	SPITEST	16	CSPI Test Register	0x0000	20-9	Normal Reset
0xFFFFF70A	SPISPC	16	CSPI Sample Period Control Register	0x0000	20-10	Normal Reset
0xFFFFF70C	SPIDMA	16	CSPI DMA Register	0x0000	20-11	Normal Reset
0x(FF)FFF800	IADR	8	I^2C Address Register	0x00	22-6	Normal Reset
0x(FF)FFF804	IFDR	8	I^2C Frequency Divider Register	0x00	22-7	Normal Reset
0x(FF)FFF808	I2CR	8	I^2C Control Register	0x00	22-8	Normal Reset
0x(FF)FFF80C	I2SR	8	I^2C Status Register	0x81	22-9	Normal Reset
0x(FF)FFF810	I2DR	8	I^2C Data I/O Register	0x00	22-11	Normal Reset
0x(FF)FFF814	IBCR	8	I^2C Byte Counter Register	0x00	22-11	Normal Reset
0xFFFFF900	USTCNT1	16	UART Unit 1 Status/Control Register	0x0000	19-11	Normal Reset
0xFFFFF902	UBAUD1	16	UART Unit 1 Baud Control Register	0x0002	19-12	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFFF904	URX1	16	UART Unit 1 Receiver Register	0x0000	19-13	Normal Reset
0xFFFFF906	UTX1	16	UART Unit 1 Transmitter Register	0xE800	19-16	Normal Reset
0xFFFFF908	UMISC1	16	UART Unit 1 Miscellaneous Register	0x0000	19-17	Normal Reset
0xFFFFF90A	NIPR1	16	UART Unit 1 Non-integer Prescaler Register	0x0000	19-19	Normal Reset
0xFFFFF90C	HMARK1	16	UART Unit 1 FIFO Level Marker Interrupt Register	0x0102	19-20	Normal Reset
0xFFFFF90E	-	-	Reserved	-		
0xFFFFF910	USTCNT2	16	UART Unit 2 Status/Control Register	0x0000	19-21	Normal Reset
0xFFFFF912	UBAUD2	16	UART Unit 2 Baud Control Register	0x0002	19-23	Normal Reset
0xFFFFF914	URX2	16	UART Unit 2 Receiver Register	0x0000	19-24	Normal Reset
0xFFFFF916	UTX2	16	UART Unit 2 Transmitter Register	0xE800	19-26	Normal Reset
0xFFFFF918	UMISC2	16	Uart Unit 2 Miscellaneous Register	0x0000	19-27	Normal Reset
0xFFFFF91A	NIPR2	16	UART Unit 2 Non-Integer Prescaler Register	0x0000	19-29	Normal Reset
0xFFFFF91C	HMARK2	16	UART Unit 2 FIFO Level Marker Interrupt Register	0x0102	19-30	Normal Reset
0xFFFFF91E	-	-	Reserved	-		
0xFFFFFB00	RTCTIME	32	RTC Time Of Day Register	0xXXXX00XX	13-4	Normal Reset
0xFFFFFB04	RTCALRM	32	RTC Alarm Register	0x00000000	13-5	Normal Reset
0xFFFFFB0A	WATCHDOG	16	Watchdog Timer Register	0x0001	13-3	Normal Reset
0xFFFFFB0C	RTCCTL	16	RTC Control Register	0x0080	13-8	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFFFB0E	RTCISR	16	RTC Interrupt Status Register	0x0000	13-8	Normal Reset
0xFFFFFB10	RTCIENR	16	RTC Interrupt Enable Register	0x0000	13-11	Normal Reset
0xFFFFFB12	STPWCH	16	Stopwatch Minutes Register	0x003F	13-13	Normal Reset
0xFFFFFB1A	DAYR	16	RTC Day Count Register	0x0xxx	13-5	Normal Reset
0xFFFFFB1C	DAYALRM	16	RTC Day Alarm Register	0x0000	13-6	Normal Reset
0xFFFFFC00	SDCTLe_H	16	SDRAM Control Register (High Word) for Chip-Select E	0x0100	8-7	Power up Reset
0xFFFFFC02	SDCTLe_L	16	SDRAM Control Register (Low Word) For Chip-Select E	0x0300	8-7	Power up Reset
0xFFFFFC04	SDCTLf_H	16	SDRAM Control Register (High Word) For Chip-Select F	0x0100	8-7	Power up Reset
0xFFFFFC06	SDCTLf_L	16	SDRAM Control Register (Low Word) For Chip-Select F	0x0300	8-7	Power up Reset
0xFFFFFC08	EDOCTLe_H	16	EDO Control Register (High Word) For Chip-Select E	0x0000	8-12	Power up Reset
0xFFFFFC0A	EDOCTLe_L	16	EDO Control Register (Low Word) For Chip-Select E	0x0000	8-12	Power up Reset
0xFFFFFC0C	EDOCTLf_H	16	EDO Control Register (High Word) For Chip-Select F	0x0000	8-12	Power up Reset
0xFFFFFC0E	EDOCTLf_L	16	EDO Control Register (Low Word) For Chip-Select F	0x0000	8-12	Power up Reset
0xFFFFFC10	SECTL	16	Secondary Control Register	0x0000	8-6	Power up Reset
0xFFFFFC80	RES	—	Reserved	—		
0xFFFFFD00	ICEMACR	32	ICE Module Address Compare Register	0x00000000	24-5	Normal Reset

Table 4-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page	Reset by
0xFFFFFD04	ICEMAMR	32	ICE Module Address Mask Register	0x00000000	24-3	Normal Reset
0xFFFFFD08	ICEMCCR	16	ICE Module Control Compare Register	0x0000	24-5	Normal Reset
0xFFFFFD0A	ICEMCMR	16	ICE Module Control Mask Register	0x0000	24-6	Normal Reset
0xFFFFFD0C	ICEMCR	16	ICE Module Control Register	0x0000	24-6	Normal Reset
0xFFFFFD0E	ICEMSR	16	ICE Module Status Register	0x0000	24-8	Normal Reset
0xFFFFFFxx	Bootloader	—	Bootloader Microcode Space	—	—	Normal Reset

Chapter 5
Clock Generation Module and Power Control Module

This chapter describes the clock generation module (CGM) and power control module (PCM). The description of both modules is comprised in a single chapter because their operation is so closely integrated. The programmability of the individual clock signals makes the CGM a flexible clock source for the MC68SZ328 and its associated peripherals.

The CGM uses two oscillators in conjunction with a multiplier/divider chain to produce the clock signals used throughout the MC68SZ328 integrated processor. The frequency of all clock signals (except the low-frequency reference) are individually selectable through software control. The MC68SZ328 has four different power modes to provide optimum power efficiency.

The PCM controls the power consumption of MC68SZ328 by controlling the PLL operation and by applying clock signals to the FLX68000 core at reduced burst widths. For maximum power savings, the MC68SZ328 features a sleep mode in which all clocks (except the low-frequency clock) are disabled.

The features of the clock generation and power control modules are:

- Generate all clocks including the DMA clock, CPU clock, system clock, LCD clock, and the USB clock
- Comply with USB standard in the implementation of the USB clock
- Programmable PLLs to allow generation of up to 200 MHz clock from a 16 MHz clock
- Enable bypass of pre-multiplier and PLLs
- Supports output of internal clock to pin for system use, testing, and debugging
- Provides individual clock dividers to generate different frequencies for various clocks
- Enable control of different levels of power consumption
- Provides dedicated logic to shut down and wake up PLL smoothly

5.1 Clock Generation Module Overview

The clock generation module (CGM) uses two oscillators to provide the clock sources to the FLX68000 core, the USB module, and all other peripherals. The clock source for the FLX68000 core and other peripherals operate at 32.768 kHz. The clock source for the USB module operates at 16 MHz or 32.768 kHz to generate a 48 MHz clock.

The CGM uses one pre-multiplier and one PLL to generate the clocks for the FLX68000 core and most of the peripherals. The first stage pre-multiplier takes an input crystal frequency of 32.768 kHz and multiplies the signal to 16.777 MHz. The second stage is a programmable PLL, the MCUPLL, that produces a maximum output frequency of 200 MHz. A dedicated external crystal or external signal source may be used to provide 16 MHz to the USBPLL. The USBPLL is used to generate a 48 MHz clock for the USB.

Table 5-1. Clock Signal Distribution

Module Name	CPU Interface	Module Application
Reset	~	CLK_32 for qualifiers, counter
CGM	CPU_CLK	DMA_CLK for CPCM, system clock divider and LCD clock divider CLK_32 for shutdown and wakeup logic MCUPLL, USBPLL clock for other logic
Chip Select	DMA_CLK	DMA_CLK for wait state
DRAMC	DMA_CLK	DMA_CLK, feedback SDCLK
DMAC	CPU_CLK	DMA_CLK and CLK_32 for request time-out logic DMA_CLK for other logic
eSRAM	DMA_CLK	DMA_CLK for BIST
CSPI	DMA_CLK	SYS_CLK, CLK_32
LCDC	DMA_CLK	DMA_CLK for data fetch LCD_CLK for LCD panel
MSC	DMA_CLK	DMA_CLK, SCLKI
MMC/SD	DMA_CLK	DMA_CLK
USBD	DMA_CLK	USB_CLK
UART1 and 2	DMA_CLK	SYS_CLK for baud gen
ASP	DMA_CLK	SYS_CLK
GPIO	DMA_CLK	~
INT Controller	N1[1]	~
PWM1	DMA_CLK	SYS_CLK, CLK_32
PWM2	DMA_CLK	SYS_CLK
Timers 1 and 2	N1	SYS_CLK, CLK_32 for counter
RTC	CPU_CLK	CLK_32 for time tick
I2C	N1	SYS_CLK for baud
BOOTSTRAP	N1	~
ICE	N1	~

1. N1 stands for 68k control signals

5.2 Clock Generation Module Operation

Figure 5-1 shows the block diagram of the clock generation module and power control. The overall clocking scheme of the MC68SZ328 design is depicted as well as the important signals and register controls of the module.

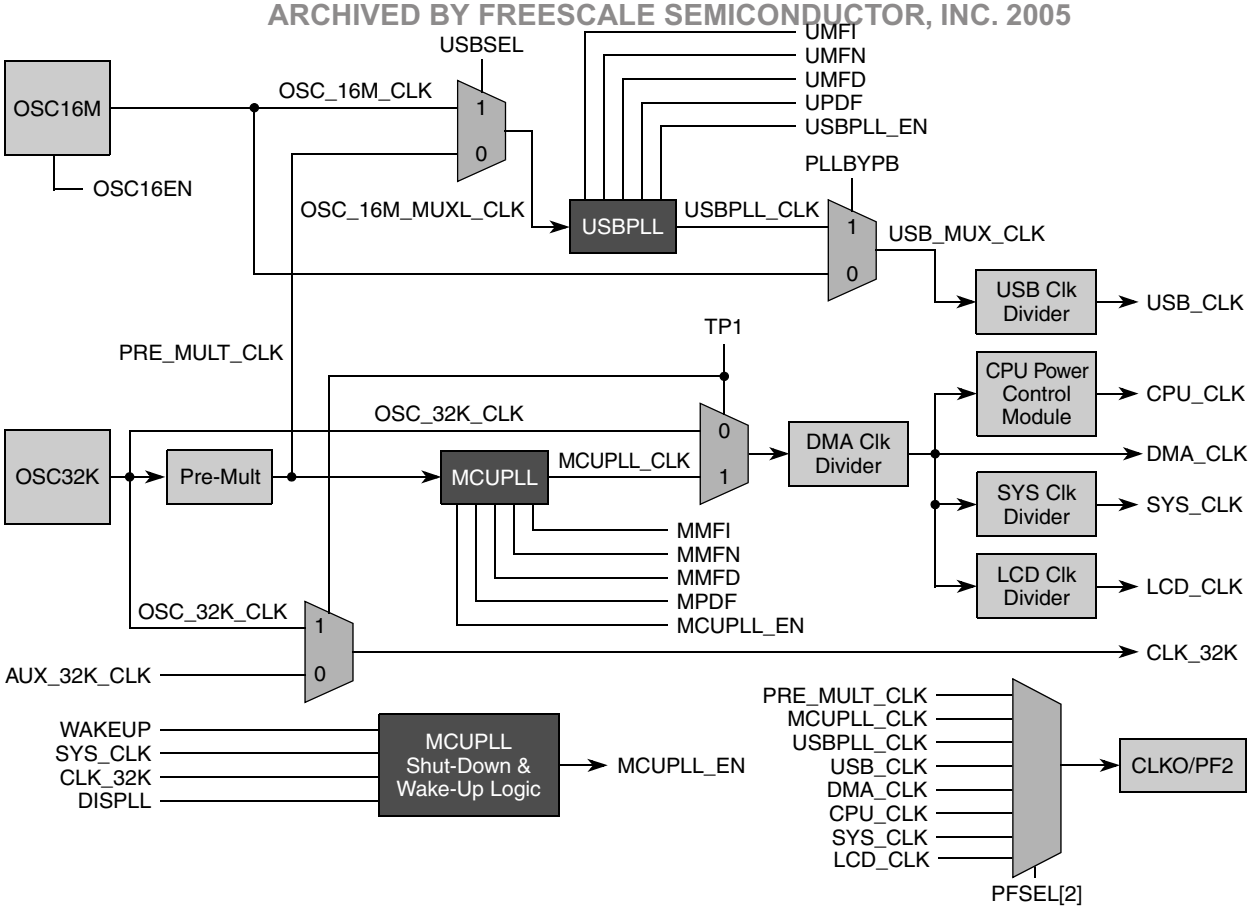


Figure 5-1. Clock Generation Module and Power Control Block Diagram

5.2.1 Clock Generation Circuits

The CGM contains two separate oscillator circuits, one circuit for a 32.768 kHz crystal and the other circuit for a 16 MHz crystal. Section 5.2.1.1, “MCUPLL Clock,” and Section 5.2.1.2, “USB Clock (48 MHz),” describe the clock generation of these oscillator circuits.

5.2.1.1 MCUPLL Clock

The external clock source for the CPU and most of the peripherals is a 32.768 kHz crystal connected to an internal oscillator circuit. The output of the oscillator is the OSC_32K_CLK signal that feeds the pre-multiplier which multiplies the signal by 512 to produce a 16.777 MHz output. This output then becomes the input signal to the MCUPLL. The pre-multiplier and the MCUPLL stages can be bypassed by connecting the TP1 pin to ground. The MCUPLL generates a clock signal, MCUPLL_CLK, the frequency of which is determined by Equation 5-1:

$$F_{mcupll} = 2 \times F_{mcupllin} \times \frac{MMFI + MMFN \div MMFD}{MPDF} \qquad \textit{Eqn. 5-1}$$

where F_{mcupll} = Output frequency of MCUPLL

$F_{mcupllin}$ = Output frequency of pre-multiplier
(normally, it is equal to 32.768 k × 512 = 16.777 MHz)

MMFI = Integer of the MCUPLL multiplication factor

MMFN = Numerator of the MCUPLL multiplication factor

MMFD = Denominator of the MCUPLL multiplication factor

MPDF = Pre-division factor of the MCUPLL. See Table 5-3.

MMFD should be set to be no larger than 0x3FE and the corresponding input values to the MPFSR1 register are in the Table 5-2.

Table 5-2. MMFD Settings in the MPFSR1 Register

MMFD[9:0]	MMFD
0	1
1	2
...	...
0x3FD	0x3FE
0x3FE	0x3FF

The MPDF input values to the MPFSR1 register are provided in Table 5-3.

Table 5-3. MPDF Settings in the MPFSR1 Register

MPDF[3:0]	MPDF
$0	1
$1	2
$2	3
...	...
$e	15
$F	16

The four parameters MMFI, MMFN, MMFD, and MPDF are programmable in the MCUPLL frequency select registers, MPFSR0 and MPFSR1. The new parameters take effect when the MPRS bit in the PLLCR register is set. Setting this bit restarts the MCUPLL at a newly assigned frequency.

NOTE:

A minimum of 100 µs is required to switch to the new frequency after the MPRS bit is set. The old frequency clock is stopped for 3 VCO clock cycles. The new frequency clock will be generated after the MCUPLL is locked to the newly assigned frequency. No clock is generated during the period when the old frequency clock is stopped and before the MCUPLL is locked to the new frequency.

5.2.1.2 USB Clock (48 MHz)

The external clock source for the 48 MHz USB clock is an external 16 MHz crystal connected to an internal oscillator circuit. The output of the oscillator is the OSC_16M_CLK signal which is fed to the USBPLL. The output of the pre-multiplier may also be used instead of using the external 16 MHz crystal and its associated oscillator circuit. The pre-multiplier output is used by writing '0' to the USBSEL bit of the clock source control register (CSCR).When writing '0' to the USBSEL bit, the pre-multiplier and the MCUPLL are used to generate the USB clock.

The USBPLL generates a clock signal, USBPLL_CLK, whose 48 MHz frequency is determined by Equation 5-2:

$$F_{usbpll} = 2 \times F_{usbpllin} \times \frac{UMFI + UMFN \div UMFD}{UPDF} \qquad \textit{Eqn. 5-2}$$

where F_{usbpll} = Output frequency of USBPLL

$F_{usbpllin}$ = Output frequency of OSC16M (USB crystal oscillator)

UMFI = Integer of the USBPLL multiplication factor

UMFN = Numerator of the USBPLL multiplication factor

UMFD = Denominator of the USBPLL multiplication factor. See Table 5-4.

UPDF = Pre-division factor of the USBPLL. See Table 5-5.

The corresponding input values of the UMFD to the UPFSR1 register are in the Table 5-4.

Table 5-4. UMFD Settings in the UPFSR1 Register

UMFD[9:0]	UMFD
0	1
1	2
2	3
...	...
0x3FD	0x3FE
0x3FE	0x3FF

The value to set for the USBPLL pre-division factor (UPDF) is provided in Table 5-5.

Table 5-5. UPDF Settings in the UPFSR1 Register

UPDF[3:0]	UPDF
$0	1
$1	2
$2	3
...	...
$e	15

Table 5-5. UPDF Settings in the UPFSR1 Register (Continued)

UPDF[3:0]	UPDF
$F	16

The 48 MHz frequency generated by USBPLL is accomplished by programming the UMFI, UMFN, UMFD and UPDF bits in the USBPLL frequency select registers, UPFSR0 and UPFSR1. The new parameters take effect when the UPRS bit in PLLCR register is set. Setting this bit restarts the USBPLL at the newly assigned frequency.

NOTE:

A minimum of 100 µs is required to switch to a new frequency after the UPRS bit is set. The old frequency clock is stopped for 3 VCO clock cycles. The new frequency clock is generated after the USBPLL is locked to the newly assigned frequency. No clock signal is generated during the period when the old frequency clock is stopped and before the USBPLL is locked to the new frequency.

5.2.1.3 Bypassing the MCUPLL Clock

The MC68SZ328 allows the internal PLLs to be bypassed and the external 32.768 kHz crystal replaced with an external high-frequency (for example, 66.32 MHz) oscillator. In this configuration, the internal pre-multiplier and the MCUPLL will be bypassed and the high-frequency clock will be used as the MCUPLL_CLK. However, the 32.768 kHz clock is still required and should be provided through the PE3/UCLK pin.

Each of the internal clocks has its own divider to program to the correct frequency. Although the USBPLL can be programmed to 48 MHz for the USB module, a USB clock divider is available. It is used when the pre-multiplier and the PLLs are bypassed and an external high-frequency oscillator is used to generate all the internal clocks. Because the USB clock frequency may be different from those produced by the other clocks, the USB clock divider is used to program its output to the desired frequency.

5.2.2 Test Capability

In addition to the ability to bypass the pre-multiplier or the PLLs, there are a total of eight internal clocks in the CGM which can be made available to external devices through the buffer at the CLKO/PF2 pin for testing and debugging or other application needs. Because the CLKO/PF2 pin is multiplexed, it is only necessary to set PFSEL[2] in the Port F Select Register to 0 to apply the CLKO signal to pin 2 of Port F.

To select the desired clock the CLKOSEL field in the clock source control register (CSCR) must be programmed as shown in Table 5-6.

Table 5-6. Clock Selections

CLKOSEL[2:0]	CLKO
000	PRE_MULT_CLK
001	MCUPLL_CLK
010	USBPLL_CLK
011	SYS_CLK

Table 5-6. Clock Selections (Continued)

CLKOSEL[2:0]	CLKO
100	USB_CLK
101	DMA_CLK
110	CPU_CLK
111	LCD_CLK

5.3 Power Management

The power control logic optimizes the power consumption of the MC68SZ328 by controlling the efficiency of the clocking system. Depending on the application, the user has several methods for power management of the chip. For maximum power savings, the MC68SZ328 can be placed in sleep mode in which all clocks except the 32.768 kHz clock are disabled. To control the power consumption of the FLX68000 core, the logic provides different burst widths of the CPU clock which is programmable as either burst or doze modes. Each of the power modes are described in full detail in Section 5.3.3.1, “Power Modes,” on page 5-8.

The power management modes are summarized in the Table 5-7.

Table 5-7. Summary of Power Control in MC68SZ328

Clock Power Control	Setting to Obtain Shut Down or Low Power	Wake Up Event or Normal Mode
MCUPLL	Set the DISPLL bit in the PLLCR register and then execute a STOP instruction. The MCUPLL will shut down after 30 CPU clock cycles. • Pre-Multiplier is also shut down. • All clocks except the 32.768 kHz clock are stopped.	Receives wake-up event.
USBPLL	Set the DISUPLL bit in the PLLCR register. • The USBPLL cannot be stopped after MCUPLL is disabled. • Only the USB clock is stopped. • Enters sleep mode if the DISPLL bit in the PLLCR register is set—for all clocks.	Reset the DISUPLL bit in the PLLCR register.
CPU	Enable the PCEN bit in the PCTLR register and configure WIDTH bit setting. • Enter doze mode when width = 0. • Enter burst mode when 32 > width > 0	Receives wake-up event or resetting PCEN bit (burst mode only).

5.3.1 MCUPLL Clock Control

The MCUPLL clock (MCUPLL_CLK) can be stopped by disabling the MCUPLL. Setting the DISPLL bit in the PLLCR register disables the MCUPLL clock. The MCUPLL is shut down for 30 clock cycles of the CPU clock after the DISPLL bit is set. This allows sufficient time to execute the STOP instruction and for the peripherals to stop operation smoothly before the clocks are stopped. When the MCUPLL is disabled, the CPU clock, the DMA clock, and the LCD clock are stopped.

Because the PLLCR register is clocked by the CPU clock, the user cannot write '0' to the DISPLL bit to wake-up the MCUPLL. A wake-up event is needed to re-enable the MCUPLL. The wake-up events can be any hardware or software interrupts clocked by the continuously running 32.768 kHz clock. When a wake-up event occurs and the MCUPLL is enabled, the MCUPLL_CLK begins operation after a delay of 300 µs which allows the MCUPLL to get locked and stable. The remainder of the clocks derived from the MCUPLL_CLK start running accordingly.

NOTE:

It is the responsibility of the user to execute the STOP instruction to stop the FLX68000 core from executing further instructions. The clock generation and power control modules cannot stop the FLX68000 core.

5.3.2 USBPLL Clock Control

The USBPLL can be shut down by writing a '1' to the DISUPLL bit in the PLLCR register. When the USBPLL is disabled, the USBPLL clock (USBPLL_CLK) is stopped. To wake-up the USBPLL, the DISUPLL must be reset. Notice that no shut down or wake-up logic is used to control these operations.

WARNING:

The USBPLL cannot be disabled after disabling the MCUPLL because the PLLCR register is clocked by the CPU clock. When the MCUPLL is shut down, the CPU clock will be stopped. There is no way to shut down the USBPLL until the MCUPLL is woken up again.

5.3.3 CPU Clock Control

The CPU power control module (PCM) is inherited from the last generation of DragonBall, the MC68VZ328. The PCM controls the power consumption of the CPU by turning the CPU clock on for a programmed number of clock pulses. The CPU consumes more power than most of the other components in the MC68SZ328, however, it conserves power when the CPU is idle. The CPU clock can be disabled or be applied at different burst lengths. When the MC68SZ328 is in one of these reduced-power modes, it is restored to normal operation by a wake-up event.

5.3.3.1 Power Modes

The MC68SZ328 has four power modes: normal, burst, doze, and sleep. In normal mode, the PCM is off. The CPU enters burst mode when the PCM module is enabled. If the burst width of the CPU clock is reduced to zero, CPU clock is disabled and the CPU is in doze mode. The lowest setting is sleep mode. Sections Section 5.3.3.1.1, "Normal Mode," through Section 5.3.3.1.4, "Sleep Mode," give detailed information about each of the four power modes.

5.3.3.1.1 Normal Mode

The CPU enters normal mode upon power reset. In this mode, the PCM is off. The CPU clock runs continuously and the CPU consumes maximum power.

5.3.3.1.2 Burst Mode

Setting the PCEN bit in the power control register (PCTLR) enables the PCM, which causes the clock burst width of the CPU clock to be the same as the WIDTH bit setting in the PCTLR register in increments of 1/31 32.768 kHz clock cycles. The WIDTH bit settings can be from 0 to 31. To explain the operation, an example is given here and is illustrated in Figure 5-2. When the WIDTH bit setting is five, the system

clock is available for a period of five 32.768 kHz clock cycles (burst width). In the remaining twenty-six 32.768 kHz clock cycles the system clock is gated. This pattern is repeated and effectively produces a CPU clock with a variable burst width and which provides power control.

Resetting the PCEN bit or a wake-up event returns the CPU to normal mode.

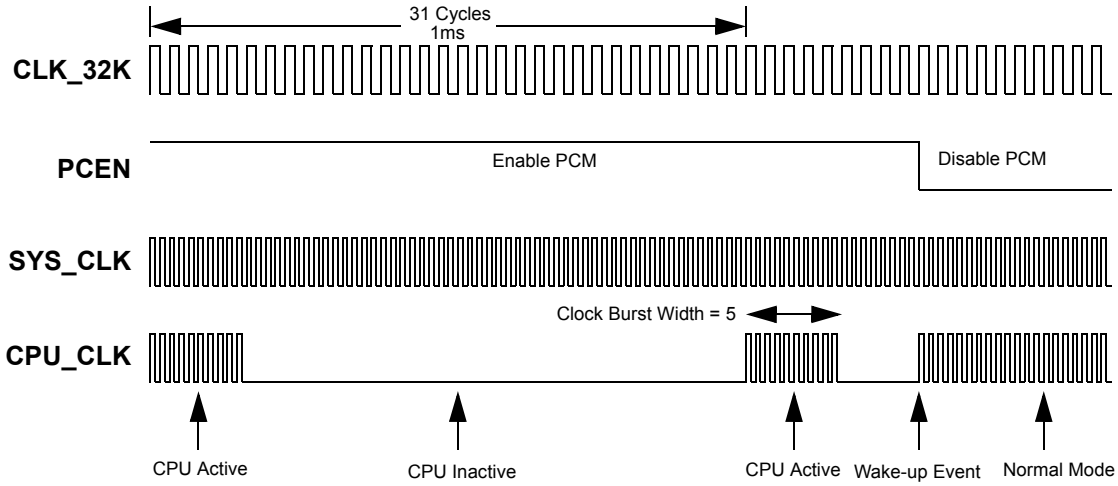


Figure 5-2. CPU Power Control Operation with Burst Width of Five

5.3.3.1.3 Doze Mode

When the PCM is enabled and the WIDTH bit set to 0, the burst width of the CPU clock is reduced to zero, which causes the CPU to enter doze mode. The CPU clock is stopped. To exit the doze mode and return to normal mode, a wake-up event is required.

When a wake-up event is received during burst or doze mode, the PCM is immediately disabled and the continuous CPU clock resumes. It is recommended that the user reenable the PCM using the wake-up service routine, if CPU burst or doze mode is wanted after the wake-up event.

5.3.3.1.4 Sleep Mode

The sleep mode disables all of the clocks in the chip except the 32.768 kHz clock. There are two ways to put the MC68SZ328 in sleep mode: either set the DISUPLL and DISPLL bits *at the same time* or set the DISUPLL bit *and then the* DISPLL bit—not conversely. Only the 32.768 kHz clock works to keep the real-time clock operational. A wake-up event is required to activate the PLL(s). The USB clock resumes operation as soon as the DISUPLL bit is enabled.

5.3.4 CPU Power Control and DMA Controller

The DMA controller is not affected by the PCM. Before the CPU clock is stopped, the DMA controller requests the bus from the CPU. The CPU clock stops only after the bus is granted. When a bus grant to the DMA controller is asserted, the DMA controller is allowed full access to the bus even when the PCM is enabled. This process also directs the LCD screen to stay refreshed.

If a wake-up event occurs while the CPU clock is disabled, the PCM is disabled and the CPU clock is immediately restored so that the CPU can process the event. The DMA controller always has priority over the bus. Therefore, if the DMA access is in progress, the CPU will wait unit the DMA controller has completed access before servicing the wake-up routine.

5.4 Programming Model

This section describes the registers that enable and control the frequency of the CGM clocks. To maintain the software compatibility between the MC68SZ328 and MC68VZ328, the PLL control register (PLLCR) and the power control register (PCTLR) have been maintained. Their memory locations and register bits remain the same as the MC68VZ328. However, the frequency select register (PLLFSR) which controls the multiplication factor of the PLL has been removed because the PLL operations of the two chips are completely different. Nevertheless, the MCUPLL frequency select register 0 (MPFSR0) which controls the frequency of the main PLL, MCUPLL, replaces the PLLFSR register. The MPFSR0 register is at the same memory location as the PLLFSR register in MC68VZ328.

5.4.1 PLL Control Register

The PLL control register is used to control the operation and frequency of the PLLs in the MC68SZ328. The settings for each bit and field in the register are described in Table 5-8.

PLLCR PLL Control Register **0x(FF)FFF200**

BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
MPRS	UPRS	LCDCLK SEL			SYSCLK SEL						SDSEL	DISPLL	DISUPLL		
TYPE	rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw		
RESET	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0
	0x2404															

Table 5-8. PLL Control Register Description

Name	Description	Setting
MPRS Bit 15	**MCUPLL Restart**—This bit restarts the MCUPLL at a newly assigned frequency. This bit is automatically cleared.	0 = MCUPLL keep current frequency. 1 = Restart MCUPLL.
UPRS Bit 14	**USBPLL Restart**—This bit restarts the USBPLL at a newly assigned frequency. This bit is automatically cleared.	0 = USBPLL keep current frequency. 1 = Restart USBPLL.
LCDCLK SEL Bits 13–11	**LCD Clock Select**—This field controls the divide ratio used by the LCD clock divider to convert DMA clock to LCD clock. It can be changed at any time. This field operates as it did in the MC68VZ328.	000 = DMACLK / 2 001 = DMACLK / 4 010 = DMACLK / 8 011 = DMACLK / 16 1xx = DMACLK / 1
SYSCLK SEL Bits 10–8	**System Clock Select**—This field controls the divide ratio used by the system clock divider to convert DMA clock to system clock. It can be changed at any time. This field operates as it did in the MC68VZ328.	000 = DMACLK / 2 001 = DMACLK / 4 010 = DMACLK / 8 011 = DMACLK / 16 1xx = DMACLK / 1
Reserved Bits 7–5	Reserved	These bits are reserved and should be set to 0.

Table 5-8. PLL Control Register Description (Continued)

Name	Description	Setting
SDSEL Bits 4	**Shutdown Select**—This field selects the number of CPU clock cycles before MCUPLL is shutdown, after DISPLL bit is set. **Note:** This feature is only supported by the 0L95J maskset and onward.	0 = 31 clock cycles. 1 = 63 clock cycles.
DISPLL Bit 3	**Disable USBPLL**—This bit, when set, disables the output of MCUPLL to reduce power.	0 = MCUPLL enabled. 1 = MCUPLL disabled. This bit is cleared automatically after wakeup from sleep mode.
DISUPLL Bit 2	**Disable USBPLL**—This bit, when set, disables the output of USBPLL to reduce power.	0 = USBPLL enabled. 1 = USBPLL disabled.
Reserved Bits 1–0	Reserved	These bits are reserved and should be set to 0.

5.4.2 MCUPLL Frequency Select Register 0

The MCUPLL frequency select register 0 (MPFSR0) is one of two registers that contain fields and registers to control the output frequency of the MCUPLL. The settings for each bit and field in the register are described in Table 5-9.

MPFSR0 MCUPLL Frequency Select Register 0 **0x(FF)FFF202**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
		MMFI				MBRMO	MMFN									
TYPE		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	1	1	1	1	0	0	1	1	1	0	1	0	0	0
	0x3CE8															

Table 5-9. MCUPLL Frequency Select Register 0 Description

Name	Description	Setting
Reserved Bit 15	Reserved	This bit is reserved and should be set to 0.
MMFI Bits 14–11	**MCUPLL Multiplication Factor Integer**—This field contains the integer part of the fraction of the multiplication factor of the MCUPLL. This field is used by the MCUPLL to generate the assigned frequency.	This field should be programmed to MMFI and set so that the output frequency is less than 200 MHz. The minimum value for this field is 5. If a value less than 5 is used, the MMFI will operate as if the value was set to 5.

Table 5-9. MCUPLL Frequency Select Register 0 Description (Continued)

Name	Description	Setting
MBRMO Bit 10	**MCUPLL BRM Order**—This bit is used to control the BRM order of the PLLs. When this bit is set, the BRM has the second order, otherwise, it has the first order. The first order BRM has to be used if a MPF fractional part is both more than one-tenth and less than nine-tenths. In other cases, the second order BRM should be used. **Note:** The value of MPF is derived from MMFN/MMFD. When 1/10 < MPF<9/10: MBRMO = 0. When MPF ≤1/10 or MPF ≥ 9/10 MBRMO = 1.	0 = BRM has first order. 1 = BRM has second order.
MMFN Bits 9–0	**MCUPLL Multiplication Factor Numerator**—This field contains the numerator of fractional part of Multiplication Factor of MCUPLL. It is used by MCUPLL to generate the assigned frequency.	This field should be programmed to MMFD and set so that the output frequency is less than 200 MHz.

5.4.3 MCUPLL Frequency Select Register 1

The MCUPLL frequency select register 1 (MPFSR1) is one of two registers that contain fields and registers which control the output frequency of the MCUPLL. The settings for each bit and field in the register are described in Table 5-10.

MPFSR1 MCUPLL Frequency Select Register 1 **0x(FF)FFF204**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
		MPDF[3:0]					MMFD									
TYPE		rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	1	1	0	0	0	1	1	1	1	1	1	1	1
	0x18FF															

Table 5-10. MCUPLL Frequency Select Register 1 Description

Name	Description	Setting
Reserved Bit 15	Reserved	This bit is reserved and should be set to 0.
MPDF[3:0] Bits 14–11	**MCUPLL Pre-Division Factor**—This field contains the pre-divider factor of the MCUPLL. It is used by the MCUPLL to generate the assigned frequency.	This field should be programmed to MPDF-1 and set so that the output frequency is less than 200 MHz.
Reserved Bit 10	Reserved	This bit is reserved and should be set to 0.
MMFD Bits 9–0	**MCUPLL Multiplication Factor Denominator**—This field contains the divider of the fractional part of the multiplication factor of the MCUPLL. It is used by the MCUPLL to generate the assigned frequency.	This field should be programmed to MMFD-1 and set so that the output frequency is less than 200 MHz.

5.4.4 USBPLL Frequency Select Register 0

The USBPLL frequency select register 0 (UPFSR0) is one of two registers that contain the fields and registers which control the output frequency of the USBPLL. The settings for each bit and field in the register are described in Table 5-11.

UPFSR0 USBPLL Frequency Select Register 0 **0x(FF)FFF208**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
		UMFI				UBRMO	UMFN									
TYPE		rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1
	0x2C01															

Table 5-11. USBPLL Frequency Select Register 0 Description

Name	Description	Setting
Reserved Bit 15	Reserved	This bit is reserved and should be set to 0.
UMFI Bits 14–11	**USBPLL Multiplication Factor Integer**—This field contains the integer part of the fraction part of multiplication factor of USBPLL. It is used by USBPLL to generate the assigned frequency.	This field should be programmed to UMFI and set so that the output frequency is less than 200 MHz. The minimum value for this field is 5. If a value less than 5 is used, the UMFI will operate as if the value was set to 5.
UBRMO Bit 10	**USBPLL BRM Order**—This bit is used to control the BRM order of the PLLs. When this bit is set, the BRM has the second order, otherwise, it has the first order. The first order BRM must be used if an MPF fractional part is both more than one-tenth and less than nine-tenths. In other cases, the second order BRM should be used. **Note:** The value of MPF is derived from UMFN/UMFD. When 1/10 < MPF< 9/10: UBRMO = 0. When MPF ≤ 1/10 or MPF ≥ 9/10: UBRMO = 1.	0 = BRM has first order. 1 = BRM has second order.
UMFN Bits 9–0	**USBPLL Multiplication Factor Numerator**—This field contains the numerator of the fractional part of multiplication factor of the USBPLL. It is used by the USBPLL to generate the assigned frequency.	This field should be programmed to UMFD and set so that the output frequency is less than 200 MHz.

5.4.5 USBPLL Frequency Select Register 1

The USBPLL frequency select register 1 (UPFSR1) is one of two registers containing fields and registers that control the output frequency of the USBPLL. The settings for each bit and field in the register are described in Table 5-12.

UPFSR1 USBPLL Frequency Select Register 1 **0x(FF)FFF20A**

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
		UPDF[3:0]					UMFD									
TYPE		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 5-12. USBPLL Frequency Select Register 1 Description

Name	Description	Setting
Reserved Bit 15	Reserved	This bit is reserved and should be set to 0.
UPDF[3:0] Bits 14–11	**USBPLL Pre-Division Factor**—This field contains the pre-divider factor of USBPLL. This field is used by the USBPLL to generate the assigned frequency.	This field should be programmed to UPDF-1 and set so that the output frequency is less than 200 MHz.
Reserved Bit 10	Reserved	This bit is reserved and should be set to 0.
UMFD Bits 9–0	**USBPLL Multiplication Factor Denominator**—This field contains the divider of the fractional part of the multiplication factor of USBPLL. This field is used by the USBPLL to generate the assigned frequency.	This field should be programmed to UMFD-1 and set so that the output frequency is less than 200 MHz.

5.4.6 CPU Power Control Register

The CPU power control register (PCTLR) controls the operation of the CPU power control module and also controls the width of the CPU clock bursts. The settings for each bit and field in the register are described in Table 5-13.

PCTLR CPU Power Control Register **0x(FF)FFF207**

BIT	7	6	5	4	3	2	1	BIT 0
	PCEN			WIDTH				
TYPE	rw			rw	rw	rw	rw	rw
RESET	0	0	0	1	1	1	1	1
	0x1F							

Table 5-13. CPU Power Control Register Description

Name	Description	Setting
PCEN Bit 7	**Power Control Enable**—This bit, derived from MC68VZ328, controls the operation of the CPU power control module. When this bit is low, the CPU clock is on continuously. When this bit is high, the pulse-width comparator presents the clock to the CPU in bursts or disables it. When this bit is high, a masked interrupt can disable the CPU power control module.	0 = CPU power control is disabled. 1 = CPU power control is enabled.
Reserved Bits 6–5	Reserved	These bits are reserved and should set to 0.
WIDTH Bits 4–0	**Width**—This field controls the width of the CPU clock bursts in increments of 1/31 32.768 kHz clock cycles. When this field is set between 1 to 30, the CPU clock is applied in burst widths of 1 to 30, respectively, 32.768 kHz clock cycles. When it is set to 0, the clock is always off. When it is set to $31, the clock is always on. The CPU clock can be woken up without waiting for the MCUPLL to reacquire lock. The contents of this field are not affected by the PCEN bit. When an interrupt disables the CPU power control module, these bits are not changed.	00000 = CPU clock always off. 00001 = 1/31 32.768 kHz clock burst width. 00010 = 2/31 32.768 kHz clock burst width. . . . 11111 = CPU clock always on.

5.4.7 Clock Sources Control Register

The clock sources control register (CSCR) is used to select the clock source and control the divide ratios for the USB and DMA clocks. The settings for each bit and field in the register are described in Table 5-14.

CSCR Clock Sources Control Register **0x(FF)FFF20C**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	USBSEL	PLLBYPB	USBCDIV			DMACDIV							OSC16EN	CLKOSEL		
TYPE		rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	rw
RESET	1	0	0	0	1	0	0	1	0	0	0	0	0	0	1	1
	0x8903															

Table 5-14. Clock Sources Control Register Description

Name	Description	Setting
USBSEL Bit 15	**USB Select**—This bit selects the clock source from the external 16 MHz crystal or the 16 MHz clock from pre-multiplier output.	0 = Pre-multiplier output. 1 = External 16 MHz crystal.
PLLBYPB Bit 14	**USBPLL ByPass**—This bit, when negated, bypasses the USBPLL. The external oscillator is used to provide all the clocks to the chip except for the 32.768 kHz clock.	0 = Bypasses USBPLL. 1 = Use USBPLL.
USBCDIV Bits 13–11	**USB Clock Divider**—This field controls the divide ratio used by the USB clock divider to convert USBPLL_CLK to the USB clock.	000 = USBPLL_CLK / 2 001 = USBPLL_CLK / 4 010 = USBPLL_CLK / 8 011 = USBPLL_CLK / 16 1xx = USBPLL_CLK / 1

Table 5-14. Clock Sources Control Register Description (Continued)

Name	Description	Setting
DMACDIV Bits 10–8	**DMA Clock Divider**—This field controls the divide ratio used by the DMA clock divider to convert MCUPLL_CLK to the DMA clock.	000 = MCUPLL_CLK / 2 001 = MCUPLL_CLK / 4 010 = MCUPLL_CLK / 8 011 = MCUPLL_CLK / 16 1xx = MCUPLL_CLK / 1
Reserved Bits 7–4	Reserved	These bits are reserved and should set to 0.
OSC16EN Bit 3	**16 MHz Oscillator Enable**—This bit enables or disables the internal 16 MHz oscillator.	0 = Disable the internal 16 MHz oscillator. 1 = Enable the internal 16 MHz oscillator.
CLKOSEL Bits 2–0	**Clock Select**—This field selects one of the eight internal signals or clocks brought outside the chip.	000 = PRE_MULT_CLK 001= MCUPLL_CLK 010 = USBPLL_CLK 011 = System clock 100 = USB clock 101 = DMA clock 110 = CPU clock 111 = LCD clock

Chapter 6
System Control

This chapter describes the system control register of the MC68SZ328 microprocessor. The system control register enables system software to control and customize the following functions:

- Access permission from the internal peripheral registers
- Address space of the internal peripheral registers
- Bus time-out control and status (bus error generator)

6.1 System Control Operation

The on-chip resources use a reserved 8,192-byte block of address space for their registers. This block is mapped beginning at location 0xFFFE0000 (32-bit) or 0xXXFE0000 (24-bit, where XX is “don’t care”) on reset. The DMAP bit in the system control register disables double mapping in a 32-bit system. If this bit is cleared, the on-chip peripheral registers appear only at the top of the 4 Gbyte address range starting at 0xFFFE0000.

The system control register provides control of system operation functions such as bus interface and watchdog protection. The system control register contains status bits that allow exception handler code to interrogate the cause of both exceptions and resets. The bus time-out monitor and the watchdog timer provide system protection. The bus time-out monitor generates a bus error when a bus cycle is not terminated by the $\overline{\text{DTACK}}$ signal after 128 clock cycles have elapsed.

6.1.1 Bus Monitors and Watchdog Timers

The bus error time-out logic consists of a bus time-out monitor that, when enabled, begins to count clock cycles as the internal $\overline{\text{AS}}$ pin is asserted for internal or external bus accesses. The deassertion of $\overline{\text{AS}}$ normally terminates the count, but if the count reaches terminal count before $\overline{\text{AS}}$ is deasserted, $\overline{\text{BERR}}$ is asserted until $\overline{\text{AS}}$ is deasserted. The bus error time-out logic consists of 1 control bit and 1 status bit in the system control register. The BETO bit in the system control register is set after a bus time out, which may indicate a write-protect violation or privilege.

The watchdog timer resets the MC68SZ328 if it is enabled and not cleared or disabled before reaching terminal count. The watchdog timer is enabled at reset.

6.2 Programming Model

The following sections provide detailed programming information about the system control register and the other registers associated with its operation.

6.2.1 System Control Register

The 8-bit read/write system control register (SCR) resides at the address 0xFFFE0000 or 0xXXFE0000 (where XX is “don’t care”) after reset. The SCR and all other internal registers cannot be accessed in the 68000’s user mode if the SO bit is set to 1. 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 6-1.

SCR System Control Register **0x(FF)FFF000**

	BIT 7	6	5	4	3	2	1	BIT 0
	BETO	WPV	PRV	BETEN	SO	DMAP		
TYPE	rw	rw	rw	rw	rw	rw		
RESET	0	0	0	1	1	1	0	0
	0x1C							

Table 6-1. System Control Register Description

Name	Description	Setting
BETO Bit 7	**Bus Error Time Out—**This status bit indicates whether or not a bus-error-timer time out has occurred. When a bus cycle is not terminated by the $\overline{\text{DTACK}}$ signal after 128 clock cycles have elapsed, the BETO bit is set. However, the BETEN bit must be set for a bus error time out to occur. This bit is cleared by writing a 1 (writing a 0 has no effect).	0 = A bus-error-timer time out did not occur. 1 = A bus-error-timer time out has occurred because an undecoded address space has been accessed or because a write-protect or privilege violation has occurred.
WPV Bit 6	**Write-Protect Violation—**This status bit indicates that a write-protect violation has occurred. If a write-protect violation occurs and the BETEN bit is not set, the current bus cycle will not terminate. The BETEN bit must be set for a bus error exception to occur during a write-protect violation. This bit is cleared by writing a 1 (writing a 0 has no effect).	0 = A write-protect violation did not occur. 1 = A write-protect violation has occurred.
PRV Bit 5	**Privilege Violation—**This status bit indicates that if a privilege violation occurs and the BETEN bit is not set, the cycle will not terminate. The BETEN bit must be set for a bus error exception to occur during a privilege violation. This bit is cleared by writing a 1 (writing a 0 has no effect).	0 = A privilege violation did not occur. 1 = A privilege violation has occurred.
BETEN Bit 4	**Bus Error Time-Out Enable—**This control bit enables the bus error timer.	0 = Disable the bus error timer. 1 = Enable the bus error timer.
SO Bit 3	**Supervisor Only—**This control bit limits on-chip registers to supervisor accesses only.	0 = User and supervisor mode. 1 = Supervisor-only mode.

Table 6-1. System Control Register Description (Continued)

Name	Description	Setting
DMAP Bit 2	**Double Map—**This control bit controls the double-mapping function.	0 = The on-chip registers are mapped at 0xFFFE0000–0xFFFFFFFF. 1 = The on-chip registers are mapped at 0xXXFE0000–0xXXFFFFFF. This bit is used for manufacturing testing and is enabled by default. The user should set it to 0 for general use.
Reserved Bits 1–0	Reserved	These bits are reserved and read 0.

6.2.2 Peripheral Control Register

This register controls the PWM logical block operation, timer TIN/TOUT signal, and UART UCLK signal. 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 6-2.

PCR Peripheral Control Register **0x(FF)FFF003**

	BIT 7	6	5	4	3	2	1	BIT 0
				UCLK	P[1:0]		T[1:0]	
TYPE				rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 6-2. Peripheral Control Register Description

Name	Description	Setting
Reserved Bits 7–5	Reserved	Do not use these bits.
UCLK Bit 4	**UART Clock Pin Configuration—**When UCLK of UART 1 and UART 2 is configured to output signal, this bit selects UART 1's or UART 2's UCLK for UCLK pin output. When UCLK of UART 1 and UART 2 is configured as input, this bit is "don't care," and UCLK pin is an input signal.	0 = UCLK pin is connected to UART 1. 1 = UCLK pin is connected to UART 2.
P[1:0] Bits 3–2	**PWM Outputs Logic Operation—**These bits select the logical combination for final PWM pin output.	00 = 8-bit PWM out only (default). 01 = 16-bit PWM out only. 10 = Logic OR of both PWM outputs. 11 = Logic AND of both PWM outputs.

Table 6-2. Peripheral Control Register Description

Name	Description	Setting
T[1:0] Bits 1–0	**TIN/TOUT Signal Configuration—**These 2 bits are used to configure the external TIN/TOUT signal when pin PB6/TIN/TOUT is selected as TIN/TOUT function. For detailed information on using this function, see Section 12.2, “Timer Multiplexing,” on page 12-2.	00 = TIN/TOUT is connected to Timer 1. 01 = TIN/TOUT is connected to Timer 2. 10 = Timer 2 OUT -> Timer 1 IN; TIN -> Timer 2 (DIR6 = 0), or TOUT -> Timer 1 (DIR6 = 1). 11 = Timer 1 OUT -> Timer 2 IN; TIN -> Timer 1 (DIR6 = 0), or TOUT -> Timer 2 (DIR6 = 1).

6.2.3 ID Register

This 32-bit read-only register shows the chip identification. 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 6-3.

IDR ID Register **0x(FF)FFF004**

	BIT 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	BIT 16
	CHIPID								MASKID							
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	1	0	1	0	0	1	1	0	0	0	1	0	0	0	0
	0x5310 0x5301															

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	SWID															
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 6-3. ID Register Description

Name	Description	Setting
CHIPID Bits 31–24	**Chip ID Field—**This field contains the chip identification number for the DragonBall series MPU.	See description
MASKID Bits 23–16	**Maskset ID Field—**This field contains the maskset number for the silicon. Note: Silicon 0L95J = ”00010000”, silicon 3L57D = ”00000001.”	See description
SWID Bits 15–0	**Software ID—**This field contains the custom software ID. It is normally “0000.”	See description

6.2.4 I/O Drive Control Register

This register controls the driving strength of all I/O signals. By default, all pins are defaulted to 4 mA driving current. After reset, system software should select 8 mA driving for those signals that do not need high-current driving for power saving. The bit assignments for the register are shown in the following display. The settings for the bits in the register are listed in Table 6-4.

IODCR I/O Drive Control Register **0x(FF)FFF008**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	PR	PP	PN	AB	DB	CB	PM	PK	PJ	PG	PF	PE	PD	PC	PB	
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	0
	0x0000															

Table 6-4. I/O Drive Control Register Description

Name	Description	Setting
PR Bit 15	**Port R I/O Drive Control—**This bit controls the drive current for the lines of the port R.	0 = I/O drive current for each pin is 4 mA. 1 = I/O drive current for each pin is 8 mA.
PP Bit 14	**Port P I/O Drive Control—**This bit controls the drive current for the lines of the port P.	0 = I/O drive current for each pin is 4 mA. 1 = I/O drive current for each pin is 8 mA.
PN Bit 13	**Port N I/O Drive Control—**This bit controls the drive current for the lines of the port N.	0 = I/O drive current for each pin is 4 mA. 1 = I/O drive current for each pin is 8 mA.
AB Bit 12	**Address Bus Signals I/O Drive Control—**It should be noted that A[23:20] are controlled by the PF bit.	0 = I/O drive current for each pin is 4 mA. 1 = I/O drive current for each pin is 8 mA.
DB Bit 11	**Upper Data Bus Signals I/O Drive Control—**The lower data bus is controlled by the PA bit.	0 = I/O drive current for each pin is 4 mA. 1 = I/O drive current for each pin is 8 mA.
CB Bit 10	**Control Bus Signals—**Only those signals or functions not multiplexed with GPIO are controlled by this bit.	0 = I/O drive current for each pin is 4 mA. 1 = I/O drive current for each pin is 8 mA.
PM–PB Bits 9–0	**Port M to Port B Group I/O Drive Control—**Each bit controls the drive current for the lines in the respective port.	0 = I/O drive current for each pin is 4 mA. 1 = I/O drive current for each pin is 8 mA.
Reserved Bit 0	Reserved	This bit is reserved and reads 0.

Freescale Semiconductor, Inc.

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Chapter 7
Chip-Select Module

This chapter describes the chip-select logic's function and operation and provides programming information for controlling its operation.

7.1 Overview

The MC68SZ328 microprocessor contains twelve chip-select signals: ten general-purpose, programmable chip-select signals, which are used to select external devices; one signal to select eSRAM; and one for the emulation module. The signals are arranged in four groups of two chip-selects and four groups of single chip-selects—$\overline{\text{CSA}}$[1:0], $\overline{\text{CSB}}$[1:0], $\overline{\text{CSC}}$[1:0], $\overline{\text{CSD}}$[1:0], $\overline{\text{CSE}}$, $\overline{\text{CSF}}$, $\overline{\text{CSG}}$, and $\overline{\text{EMUCS}}$.

$\overline{\text{CSA0}}$ is a special-purpose chip-select signal, which is the boot device chip-select. After reset in normal mode, all the addresses are mapped to $\overline{\text{CSA0}}$ until such time that the group base address A is programmed and the chip-select enable (EN) bit is set in the appropriate chip-select register. From that point forward, $\overline{\text{CSA0}}$ does not decode globally and is only asserted when decoded from the programming information in the chip-select register.

Group E (CSE) and Group F (CSF) are different from Groups A, B, C, and D. CSE can be programmed as row address strobe 0 (RAS0) for the DRAM interface or chip-select 0 (SD_CS0) for the SDRAM interface. Similarly, CSF can be programmed as row address strobe 1 (RAS1) for the DRAM interface or chip-select 1 (SD_CS1) for the SDRAM interface. Users should note that column address strobes (CAS0/CAS1) are connected to different I/O pins. For details, refer to Section 8.6.2, "SDRAM Control Registers," on page 8-7 and Section 7.3.2, "Chip-Select Registers," in this chapter. Group G (CSG) is unique, in that it is a specific internal signal for eSRAM.

Each memory area can be defined as an internally generated cycle-termination signal, called $\overline{\text{DTACK}}$, with a programmable number of wait states. This feature saves board space that would otherwise be used for cycle-termination logic. Using CDL, the system designer can adopt a flexible memory configuration based on cost and availability. Up to four different classes of devices and memory can be used in a system without the need for external decode or wait-state generation logic. Specifically, 8- or 16-bit combinations of ROM, SRAM, flash memory, and DRAM (EDO RAM, or synchronous) are supported, as shown in Table 7-1.

Table 7-1. Chip-Select and Memory Types

Chip-Select Signal	Memory Supported	Max Size
$\overline{\text{CSA0}}$	ROM, SRAM, flash memory chip	32 Mbyte
$\overline{\text{CSA1}}$	ROM, SRAM, flash memory chip	32 Mbyte
$\overline{\text{CSB0}}$	ROM, SRAM, flash memory chip	32 Mbyte
$\overline{\text{CSB1}}$	ROM, SRAM, flash memory chip	32 Mbyte

Table 7-1. Chip-Select and Memory Types (Continued)

Chip-Select Signal	Memory Supported	Max Size
$\overline{CSC0}$	ROM, SRAM, flash memory chip	32 Mbyte
$\overline{CSC1}$	ROM, SRAM, flash memory chip	32 Mbyte
$\overline{CSD0}$	ROM, SRAM, flash memory chip	32 Mbyte
$\overline{CSD1}$	ROM, SRAM, flash memory chip	32 Mbyte
$\overline{CSE}/\overline{RAS0}/\overline{SDCS0}$	EDO DRAM, SDRAM, ROM, SRAM, flash memory chip-select	32 Mbyte
$\overline{CSF}/\overline{RAS1}/\overline{SDCS1}$	EDO DRAM, SDRAM, ROM, SRAM, flash memory chip-select	32 Mbyte
$\overline{CSG}$	eSRAM only	eSRAM size
$\overline{EMUCS}$	Used with In-circuit emulation module	emulation space

The basic chip-select model allows the chip-select output signal to assert in response to an address match. The signals are asserted externally shortly after the internal Address Strobe ($\overline{AS}$) signal goes low. The address match is described in terms of a group base address register and a chip-select register. The memory size of the chip-select can be selected from a set of predefined ranges (256 kbyte, 512 kbyte, 1 Mbyte, 2 Mbyte, 4 Mbyte, 8 Mbyte, 16 Mbyte, or 32 Mbyte). These memory ranges represent the most popular memory sizes available on the market and apply to the registers CSB, CSC, CSD, CSE, and CSF. The CSA register primarily supports ROM. Using this scheme, it is easy to design software without the necessity of programming a chip-select mask register.

The chip-select can be programmed to allow read-only or read/write accesses. Other parameters that can be programmed include the internal or external DTACK selection, the number of wait states (from 0 to 13) and data bus size selection.

7.2 Chip-Select Operation

A chip-select output signal is asserted when an address is matched and after the $\overline{AS}$ signal goes low. The base address and address mask registers are used in the compare logic to generate an address match. The byte size of the matching block must be a power of two and the base address must be an integer multiple of this size. Therefore, a 256 kbyte block size must begin on a 256 kbyte boundary, and a 512 kbyte block size can only begin on a 512 kbyte boundary. Each chip-select is programmable, and the registers have read/write capability so that the programmed values can be read back.

NOTE:

The chip-select logic does not allow an address match to occur during interrupt acknowledge (Function Code 7) cycles.

7.2.1 Memory Protection

The chip-select range of the four chip-selects can be programmed as read-only or read/write. Chip-selects that control the crucial system data are usually programmed as supervisor-only and read-only so they can be protected from system misuse (for example, a low battery). However, a certain area of this

chip-select–controlled area can be programmed as read/write, which provides optimal memory use, as shown in Figure 7-1. This area can be defined by programming the UPSIZ bits in the CSB, CSC, CSD, CSE, and CSF registers to between 32K and the entire chip-select area.

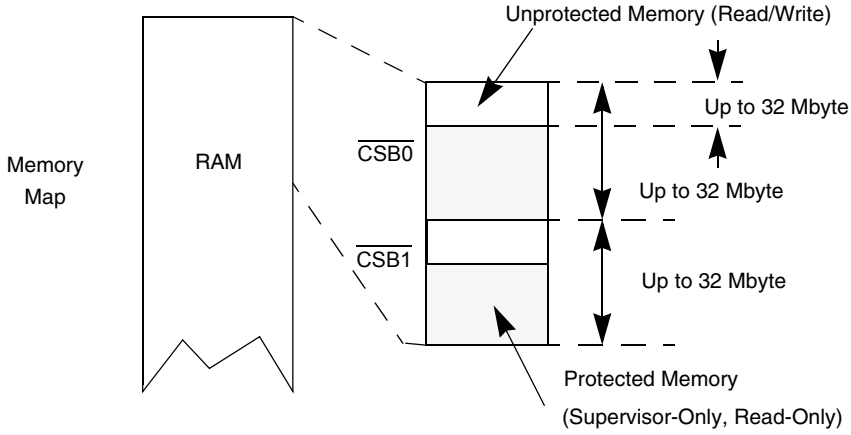


Figure 7-1. $\overline{\text{CSB0}}$ and $\overline{\text{CSB1}}$ Size Selection and Memory Protection

7.2.2 Programmable Data Bus Size

Each chip-select can be configured to address an 8- or 16-bit space. Both 16- and 8-bit contiguous address memory devices can be mixed on a 16-bit data bus system. If the CPU performs a 16-bit data transfer in an 8-bit memory space, then two 8-bit cycles will occur. However, the address and data strobes remain asserted until the end of the second 8-bit cycle. In this case, only the external CPU data bus upper byte (D[15:8]) is used, and the least significant bit of the address (A0) increments automatically from one to the next. A0 should be ignored in 16-bit data bus cycles even if only the upper or lower byte is being read or written. For an external peripheral that only needs an 8-bit data bus interface and does not require contiguous address locations (unused bytes on empty addresses), use a chip-select configured to a 16-bit data bus width and connect to the D[7:0] pins. This balances the load of the two data bus halves in an 8-bit system. The internal data bus is 16 bits wide. All internal registers can be read or written in a zero wait-state cycle.

Except for $\overline{\text{CSA0}}$, all chip-select signals are disabled by default. The data bus width (BSW) field of the chip-select option register enables 16- and 8-bit data bus widths for each of the 16 chip-select ranges. The initial bus width for the boot chip-select can be selected by placing a logic 0 or 1 on the BUSW pin at reset to specify the width of the data bus. This allows a boot EPROM of the data bus width to be used in any given system. All external accesses that do not match one of the chip-select address ranges are assumed to be a 16-bit device. This results in a single access performed for a 16-bit transfer. If it is applied to an 8-bit port, the port is accessed every other byte.

The boot chip-select is initialized from reset to assert in response to any address except the on-chip register space (0xFFFE0000 to 0xFFFFFFFF). This ensures that a chip-select to the boot ROM or EPROM will fetch the reset vector and execute the initialization code, which should set up the chip-select ranges.

A logic 0 on the BUSW pin sets the boot device's data bus to be 8 bits wide, and a logic 1 sets it to be 16 bits wide. At reset, the data bus port size for $\overline{\text{CSA0}}$ and the data width of the boot ROM device are determined by the state of BUSW. The other chip-selects are initialized to be invalid, so they will not assert until they are programmed and the EN bit is set in the chip-select registers.

7.2.3 Overlapping Chip-Select Registers

Do not program group address and chip-select registers to overlap, or the chip-select signals will overlap. Unused chip-selects must be disabled. Map them to an unused space, if possible.

When the CPU tries to write to a read-only location that has already been programmed, the chip-select and $\overline{\text{DTACK}}$ signals will not be generated internally. $\overline{\text{BERR}}$ will be asserted internally if the bus error time-out function is enabled.

NOTE:

The chip-select logic does not allow an address match during interrupt acknowledge cycles.

7.3 Programming Model

The chip-select module contains registers that are programmed to control external devices, such as memory. Chip-selects do not operate until the register in a particular group of devices is initialized and the EN bit is set in the corresponding chip-select register. The only exception is the $\overline{\text{CSA0}}$ signal, which is the boot device chip-select.

7.3.1 Chip-Select Group Base Address Registers

The upper 14 bits of each base address register select the starting address for the chip-select address range. The GBAx field is compared to the address on the address bus to determine if the group is decoded. The chip-select base address must be set according to the size of the corresponding chip-select signals of the group. For example, if $\overline{\text{CSA1}}$ and $\overline{\text{CSA0}}$ are each assigned a 2 Mbyte memory space, the CSGBA register must be set in a 4 Mbyte space boundary, such as system address 0 × 0, 0 × 4 Mbyte, 0 × 8 Mbyte, and so on. It cannot be set at 0 × 1 Mbyte, 0 × 2 Mbyte, 0 × 3 Mbyte, 0 × 5 Mbyte, and so on.

CSGBA Chip-Select Group A Base Address Register **0x(FF)FFF100**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	GBA31	GBA30	GBA29	GBA28	GBA27	GBA26	GBA25	GBA24	GBA23	GBA22	GBA21	GBA20	GBA19	GBA18		
TYPE	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 7-2. Chip-Select Group A Base Address Register Description

Name	Description	Setting
GBAx Bits 15–2	**Group A Base Address—**These bits select the high-order bits (31–18) of the starting address for the chip-select range.	The chip-select base address must be set according to the size of the corresponding chip-select signals of the group.
Reserved Bits 1–0	Reserved	These bits are reserved and should be set to 0.

CSGBB Chip-Select Group B Base Address Register 0x(FF)FFF102

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	GBB31	GBB30	GBB29	GBB28	GBB27	GBB26	GBB25	GBB24	GBB23	GBB22	GBB21	GBB20	GBB19	GBB18		
TYPE	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 7-3. Chip-Select Group B Base Address Register Description

Name	Description	Setting
GBBx Bits 15–2	**Group B Base Address—**These bits select the high-order bits (31–18) of the starting address for the chip-select range.	The chip-select base address must be set according to the size of the corresponding chip-select signals of the group.
Reserved Bits 1–0	Reserved	These bits are reserved and should be set to 0.

CSGBC Chip-Select Group C Base Address Register 0x(FF)FFF104

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	GBC31	GBC30	GBC29	GBC28	GBC27	GBC26	GBC25	GBC24	GBC23	GBC22	GBC21	GBC20	GBC19	GBC18		
TYPE	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 7-4. Chip-Select Group C Base Address Register Description

Name	Description	Setting
GBCx Bits 15–2	**Group C Base Address—**These bits select the high-order bits (31–18) of the starting address for the chip-select range.	The chip-select base address must be set according to the size of the corresponding chip-select signals of the group.
Reserved Bits 1–0	Reserved	These bits are reserved and should be set to 0.

CSGBD Chip-Select Group D Base Address Register 0x(FF)FFF106

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	GBD31	GBD30	GBD29	GBD28	GBD27	GBD26	GBD25	GBD24	GBD23	GBD22	GBD21	GBD20	GBD19	GBD18		
TYPE	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 7-5. Chip-Select Group D Base Address Register Description

Name	Description	Setting
GBDx Bits 15–2	**Group D Base Address—**These bits select the high-order bits (31–18) of the starting address for the chip-select range.	The chip-select base address must be set according to the size of the corresponding chip-select signals of the group.
Reserved Bits 1–0	Reserved	These bits are reserved and should be set to 0.

CSGBE Chip-Select Group E Base Address Register 0x(FF)FFF180

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	GBE31	GBE30	GBE29	GBE28	GBE27	GBE26	GBE25	GBE24	GBE23	GBE22	GBE21	GBE20	GBE19	GBE18		
TYPE	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 7-6. Chip-Select Group E Base Address Register Description

Name	Description	Setting
GBEx Bits 15–2	**Group E Base Address—**These bits select the high-order bits (31–18) of the starting address for the chip-select range.	The chip-select base address must be set according to the size of the corresponding chip-select signals of the group.
Reserved Bits 1–0	Reserved	These bits are reserved and should be set to 0.

CSGBF Chip-Select Group F Base Address Register **0x(FF)FFF182**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	GBF31	GBF30	GBF29	GBF28	GBF27	GBF26	GBF25	GBF24	GBF23	GBF22	GBF21	GBF20	GBF19	GBF18		
TYPE	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 7-7. Chip-Select Group F Base Address Register Description

Name	Description	Setting
GBFx Bits 15–2	**Group F Base Address—**These bits select the high-order bits (31–18) of the starting address for the chip-select range.	The chip-select base address must be set according to the size of the corresponding chip-select signals of the group.
Reserved Bits 1–0	Reserved	These bits are reserved and should be set to 0.

CSGBG Chip-Select Group G Base Address Register **0x(FF)FFF184**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	GBG31	GBG30	GBG29	GBG28	GBG27	GBG26	GBG25	GBG24	GBG23	GBG22	GBG21	GBG20	GBG19	GBG18		
TYPE	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 7-8. Chip-Select Group G Base Address Register Description

Name	Description	Setting
GBGx Bits 15–2	**Group G Base Address—**These bits select the high-order bits (31–18) of the starting address for the chip-select range.	The chip-select base address must be set according to the size of the corresponding chip-select signals of the group.
Reserved Bits 1–0	Reserved	These bits are reserved and should be set to 0.

7.3.2 Chip-Select Registers

There are seven 16-bit chip-select (CSA, CSB, CSC, CSD, CSE, CSF, and CSG) registers for each corresponding chip-select base address register. Each register controls two chip-select signals and can be configured to select the memory type and size of the memory range supported as well as to program the required wait states or use the external $\overline{\text{DTACK}}$ signal. The settings for the registers are described in Table 7-9 through Table 7-15 on page 7-18.

CSA Chip-Select Register A **0x(FF)FFF110**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	RO							FLASH	BSW	WS3–1			SIZ			EN
TYPE	rw							rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0
								0x00B0								

Table 7-9. Chip-Select Register A Description

Name	Description	Setting
RO Bit 15	**Read-Only—**This bit sets the chip-select to read-only. Otherwise, read and write accesses are allowed. A write to a read-only area will generate a bus error if the BETEN bit of the SCR is set. See Section 6.2.1, “System Control Register,” on page 6-2 for more information.	0 = Read/write. 1 = Read-only.
Reserved Bits 14–9	Reserved	These bits are reserved and should be set to 0.
FLASH Bit 8	**Flash Memory Support—**When enabled, this bit provides support for flash memory by forcing the $\overline{LWE}$/$\overline{UWE}$ signal to go active after chip-select. **Note:** This bit is used for expanded memory size for CSD when the DRAM bit in the CSD register is enabled.	0 = The chip-select and $\overline{LWE}$/$\overline{UWE}$ signals go active at the same clock edge. 1 = The chip-select signal goes low 1 clock before $\overline{LWE}$/$\overline{UWE}$.
BSW Bit 7	**Data Bus Width—**This bit sets the data bus width for this chip-select area.	0 = 8 bit. 1 = 16 bit.
WS3–1 Bits 6–4	**Wait State—**This field determines the number of wait states added before an internal $\overline{DTACK}$ signal is returned for this chip-select. **Note:** When using the external $\overline{DTACK}$ signal, you must configure the BUSW/$\overline{DTACK}$/PG0 pin.	000 = 0 + WS0 wait states. 001 = 2 + WS0 wait states. 010 = 4 + WS0 wait states. 011 = 6 + WS0 wait states. 100 = 8 + WS0 wait states. 101 = 10 + WS0 wait states. 110 = 12 + WS0 wait states. 111 = External $\overline{DTACK}$. When using the external $\overline{DTACK}$ signal, you must select $\overline{DTACK}$ function in Port G. WS0 is the FWS0, EWS0, DWS0, CWS0, BWS0, or AWS0 bit in the CSCTRL1 register.
SIZ Bits 3–1	**Chip-Select Size—**This field determines the memory range of the chip-select.	000 = 256 kbyte 001 = 512 kbyte 010 = 1 Mbyte 011 = 2 Mbyte 100 = 4 Mbyte 101 = 8 Mbyte 110 = 16 Mbyte 111 = 32 Mbyte

Table 7-9. Chip-Select Register A Description (Continued)

Name	Description	Setting
EN Bit 0	**Chip-Select Enable—**This write-only bit enables each chip-select.	0 = Disabled. 1 = Enabled.

CSB Chip-Select Register B 0x(FF)FFF112

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	RO	SOP	ROP	UPSIZ				FLASH	BSW	WS4–2			SIZ			EN
TYPE	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 7-10. Chip-Select Register B Description

Name	Description	Setting
RO Bit 15	**Read-Only—**This bit sets the chip-select to read-only. Otherwise, read and write accesses are allowed. A write to a read-only area will generate a bus error if the BETEN bit of the SCR is set. See Section 6.2.1, “System Control Register,” on page 6-2 for more information.	0 = Read/write. 1 = Read-only.
SOP Bit 14	**Supervisor-Use-Only Protected Memory Block—**This bit sets the protected memory block to supervisor-only; otherwise, both supervisor and user accesses are allowed. Attempts to access the supervisor-only area result in a bus error if the BETEN bit of the SCR is set. See Section 6.2.1, “System Control Register,” on page 6-2 for more information.	0 = Supervisor/user. 1 = Supervisor-only.
ROP Bit 13	**Read-Only for Protected Memory Block—**This bit sets the protected memory block to read-only. Otherwise, read and write accesses are allowed. If you write to a read-only area, you will get a bus error.	0 = Read/write. 1 = Read-only.
UPSIZ Bits 12–11	**Unprotected Memory Block Size—**This field determines the unprotected memory range of the chip-select.	00 = 32K. 01 = 64K. 10 = 128K. 11 = 256K.
Reserved Bits 10–9	Reserved	These bits are reserved and should be set to 0.

Table 7-10. Chip-Select Register B Description (Continued)

Name	Description	Setting
FLASH Bit 8	**Flash Memory Support—**When enabled, this bit provides support for flash memory by forcing the $\overline{\text{LWE}}/\overline{\text{UWE}}$ signal to go active after chip-select. **Note:** This bit is used for expanded memory size for CSD when the DRAM bit in the CSD register is enabled.	0 = The chip-select and $\overline{\text{LWE}}/\overline{\text{UWE}}$ signals go active at the same clock edge. 1 = The chip-select signal goes low 1 clock before $\overline{\text{LWE}}/\overline{\text{UWE}}$.
BSW Bit 7	**Data Bus Width—**This bit sets the data bus width for this chip-select area.	0 = 8 bit. 1 = 16 bit.
WS4–2 Bits 6–4	**Wait State—**This field determines the number of wait states added before an internal $\overline{\text{DTACK}}$ signal is returned for this chip-select. **Note:** When using the external $\overline{\text{DTACK}}$ signal, you must configure the BUSW/$\overline{\text{DTACK}}$/PG0 pin.	000 = 0 + BWS1 + BWS0 wait states. 001 = 4 + BWS1 + BWS0 wait states. 010 = 8 + BWS1 + BWS0 wait states. 011 = 12 + BWS1 + BWS0 wait states. 100 = 16 + BWS1 + BWS0 wait states. 101 = 20 + BWS1 + BWS0 wait states. 110 = 24 + BWS1 + BWS0 wait states. 111 = External $\overline{\text{DTACK}}$. When using the external $\overline{\text{DTACK}}$ signal, you must select $\overline{\text{DTACK}}$ function in Port G. BWS1 and BWS0 are programmed in the CSCTRL1 register and CSCTRL3 register, respectively.
SIZ Bits 3–1	**Chip-Select Size—**This field determines the memory range of the chip-select.	000 = 256 kbyte 001 = 512 kbyte 010 = 1 Mbyte 011 = 2 Mbyte 100 = 4 Mbyte 101 = 8 Mbyte 110 = 16 Mbyte 111 = 32 Mbyte
EN Bit 0	**Chip-Select Enable—**This write-only bit enables each chip-select.	0 = Disabled. 1 = Enabled.

CSC — Chip-Select Register C — 0x(FF)FFF114

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	RO	SOP	ROP	UPSIZ				FLASH	BSW	WS3–1			SIZ			EN
TYPE	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 7-11. Chip-Select Register C Description

Name	Description	Setting
RO Bit 15	**Read-Only—**This bit sets the chip-select to read-only. Otherwise, read and write accesses are allowed. A write to a read-only area will generate a bus error if the BETEN bit of the SCR is set. See Section 6.2.1, “System Control Register,” on page 6-2 for more information.	0 = Read/write. 1 = Read-only.
SOP Bit 14	**Supervisor-Use-Only Protected Memory Block—**This bit sets the protected memory block to supervisor-only; otherwise, both supervisor and user accesses are allowed. Attempts to access the supervisor-only area result in a bus error if the BETEN bit of the SCR is set. See Section 6.2.1, “System Control Register,” on page 6-2 for more information.	0 = Supervisor/user. 1 = Supervisor-only.
ROP Bit 13	**Read-Only for Protected Memory Block—**This bit sets the protected memory block to read-only. Otherwise, read and write accesses are allowed. If you write to a read-only area, you will get a bus error.	0 = Read/write. 1 = Read-only.
UPSIZ Bits 12–11	**Unprotected Memory Block Size—**This field determines the unprotected memory range of the chip-select.	00 = 32K. 01 = 64K. 10 = 128K. 11 = 256K.
Reserved Bits 10–9	Reserved	These bits are reserved and should be set to 0.
FLASH Bit 8	**Flash Memory Support—**When enabled, this bit provides support for flash memory by forcing the $\overline{\text{LWE}}$/$\overline{\text{UWE}}$ signal to go active after chip-select. **Note:** This bit is used for expanded memory size for CSD when the DRAM bit in the CSD register is enabled.	0 = The chip-select and $\overline{\text{LWE}}$/$\overline{\text{UWE}}$ signals go active at the same clock edge. 1 = The chip-select signal goes low 1 clock before $\overline{\text{LWE}}$/$\overline{\text{UWE}}$.
BSW Bit 7	**Data Bus Width—**This bit sets the data bus width for this chip-select area.	0 = 8 bit. 1 = 16 bit.

Table 7-11. Chip-Select Register C Description (Continued)

Name	Description	Setting
WS3–1 Bits 6–4	**Wait State—**This field determines the number of wait states added before an internal $\overline{\text{DTACK}}$ signal is returned for this chip-select. **Note:** When using the external $\overline{\text{DTACK}}$ signal, you must configure the BUSW/$\overline{\text{DTACK}}$/PG0 pin.	000 = 0 + WS0 wait states. 001 = 2 + WS0 wait states. 010 = 4 + WS0 wait states. 011 = 6 + WS0 wait states. 100 = 8 + WS0 wait states. 101 = 10 + WS0 wait states. 110 = 12 + WS0 wait states. 111 = External $\overline{\text{DTACK}}$. When using the external $\overline{\text{DTACK}}$ signal, you must select $\overline{\text{DTACK}}$ function in Port G. WS0 is the FWS0, EWS0, DWS0, CWS0, BWS0, or AWS0 bit in the CSCTRL1 register.
SIZ Bits 3–1	**Chip-Select Size—**This field determines the memory range of the chip-select.	000 = 256 kbyte 001 = 512 kbyte 010 = 1 Mbyte 011 = 2 Mbyte 100 = 4 Mbyte 101 = 8 Mbyte 110 = 16 Mbyte 111 = 32 Mbyte
EN Bit 0	**Chip-Select Enable—**This write-only bit enables each chip-select.	0 = Disabled. 1 = Enabled.

CSD Chip-Select Register D **0x(FF)FFF116**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	RO	SOP	ROP	UPSIZ				FLASH	BSW	WS3–1			SIZ			EN
TYPE	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 7-12. Chip-Select Register D Description

Name	Description	Setting
RO Bit 15	**Read-Only—**This bit sets the chip-select to read-only. Otherwise, read and write accesses are allowed. A write to a read-only area will generate a bus error if the BETEN bit of the SCR is set. See Section 6.2.1, “System Control Register,” on page 6-2 for more information.	0 = Read/write. 1 = Read-only.

Table 7-12. Chip-Select Register D Description (Continued)

Name	Description	Setting
SOP Bit 14	**Supervisor-Use-Only Protected Memory Block—**This bit sets the protected memory block to supervisor-only; otherwise, both supervisor and user accesses are allowed. Attempts to access the supervisor-only area result in a bus error if the BETEN bit of the SCR is set. See Section 6.2.1, “System Control Register,” on page 6-2 for more information.	0 = Supervisor/user. 1 = Supervisor-only.
ROP Bit 13	**Read-Only for Protected Memory Block—**This bit sets the protected memory block to read-only. Otherwise, read and write accesses are allowed. If you write to a read-only area, you will get a bus error.	0 = Read/write. 1 = Read-only.
UPSIZ Bits 12–11	**Unprotected Memory Block Size—**This field determines the unprotected memory range of the chip-select.	00 = 32K. 01 = 64K. 10 = 128K. 11 = 256K.
Reserved Bits 10–9	Reserved	These bits are reserved and should be set to 0.
FLASH Bit 8	**Flash Memory Support—**When enabled, this bit provides support for flash memory by forcing the $\overline{\text{LWE}}$/$\overline{\text{UWE}}$ signal to go active after chip-select. **Note:** This bit is used for expanded memory size for CSD when the DRAM bit is enabled.	0 = The chip-select and $\overline{\text{LWE}}$/$\overline{\text{UWE}}$ signals go active at the same clock edge. 1 = The chip-select signal goes low 1 clock before $\overline{\text{LWE}}$/$\overline{\text{UWE}}$.
BSW Bit 7	**Data Bus Width—**This bit sets the data bus width for this chip-select area.	0 = 8 bit. 1 = 16 bit.
WS3–1 Bits 6–4	**Wait State—**This field contains the 3 most significant bits of the 4-bit wait-state value. The least significant bit is located in the chip-select control register 1. The value of these 4 bits determines the number of wait states added to a bus cycle before an internal $\overline{\text{DTACK}}$ is asserted to terminate the chip-select cycle.	000 = 0 + WS0 wait states. 001 = 2 + WS0 wait states. 010 = 4 + WS0 wait states. 011 = 6 + WS0 wait states. 100 = 8 + WS0 wait states. 101 = 10 + WS0 wait states. 110 = 12 + WS0 wait states. 111 = External $\overline{\text{DTACK}}$. When using the external $\overline{\text{DTACK}}$ signal, you must select $\overline{\text{DTACK}}$ function in Port G. WS0 is the FWS0, EWS0, DWS0, CWS0, BWS0, or AWS0 bit in the CSCTRL1 register.

Table 7-12. Chip-Select Register D Description (Continued)

Name	Description	Setting
SIZ Bits 3–1	**Chip-Select Size—**This field determines the memory range of the chip-select.	000 = 256 kbyte 001 = 512 kbyte 010 = 1 Mbyte 011 = 2 Mbyte 100 = 4 Mbyte 101 = 8 Mbyte 110 = 16 Mbyte 111 = 32 Mbyte
EN Bit 0	**Chip-Select Enable—**This write-only bit enables each chip-select.	0 = Disabled. 1 = Enabled.

CSE Chip-Select Register E **0x(FF)FFF190**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	RO	SOP	ROP	UPSIZ			DRAM	FLASH	BSW	WS3–1			SIZ			EN
TYPE	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 7-13. Chip-Select Register E Description

Name	Description	Setting
RO Bit 15	**Read-Only—**This bit sets the chip-select to read-only. Otherwise, read and write accesses are allowed. A write to a read-only area will generate a bus error if the BETEN bit of the SCR is set. See Section 6.2.1, “System Control Register,” on page 6-2 for more information.	0 = Read/write. 1 = Read-only.
SOP Bit 14	**Supervisor-Use-Only Protected Memory Block—**This bit sets the protected memory block to supervisor-only; otherwise, both supervisor and user accesses are allowed. Attempts to access the supervisor-only area result in a bus error if the BETEN bit of the SCR is set. See Section 6.2.1, “System Control Register,” on page 6-2 for more information.	0 = Supervisor/user. 1 = Supervisor-only.
ROP Bit 13	**Read-Only for Protected Memory Block—**This bit sets the protected memory block to read-only. Otherwise, read and write accesses are allowed. If you write to a read-only area, you will get a bus error.	0 = Read/write. 1 = Read-only.
UPSIZ Bits 12–11	**Unprotected Memory Block Size—**This field determines the unprotected memory range of the chip-select.	00 = 32K. 01 = 64K. 10 = 128K. 11 = 256K.
Reserved Bit 10	Reserved	This bit is reserved and should be set to 0.
DRAM Bit 9	**DRAM Selection—**This bit is used to enable DRAM function in $\overline{\text{CSE}}$/RAS0/SDCS0 pin.	0 = Select $\overline{\text{CSE}}$. 1 = Select RAS0/SDCS0.
FLASH Bit 8	**Flash Memory Support—**When enabled, this bit provides support for flash memory by forcing the $\overline{\text{LWE}}$/$\overline{\text{UWE}}$ signal to go active after chip-select. **Note:** This bit is used for expanded memory size for CSD when the DRAM bit in the CSD register is enabled.	0 = The chip-select and $\overline{\text{LWE}}$/$\overline{\text{UWE}}$ signals go active at the same clock edge. 1 = The chip-select signal goes low 1 clock before $\overline{\text{LWE}}$/$\overline{\text{UWE}}$.

Table 7-13. Chip-Select Register E Description (Continued)

Name	Description	Setting
BSW Bit 7	**Data Bus Width—**This bit sets the data bus width for this chip-select area.	0 = 8 bit. 1 = 16 bit.
WS3–1 Bits 6–4	**Wait State—**This field determines the number of wait states added before an internal $\overline{\text{DTACK}}$ signal is returned for this chip-select. **Note:** When using the external $\overline{\text{DTACK}}$ signal, you must configure the BUSW/$\overline{\text{DTACK}}$/PG0 pin.	000 = 0 + WS0 wait states. 001 = 2 + WS0 wait states. 010 = 4 + WS0 wait states. 011 = 6 + WS0 wait states. 100 = 8 + WS0 wait states. 101 = 10 + WS0 wait states. 110 = 12 + WS0 wait states. 111 = External $\overline{\text{DTACK}}$. When using the external $\overline{\text{DTACK}}$ signal, you must select $\overline{\text{DTACK}}$ function in Port G. WS0 is the FWS0, EWS0, DWS0, CWS0, BWS0, or AWS0 bit in the CSCTRL1 register.
SIZ Bits 3–1	**Chip-Select Size—**This field determines the memory range of the chip-select.	000 = 256 kbyte 001 = 512 kbyte 010 = 1 Mbyte 011 = 2 Mbyte 100 = 4 Mbyte 101 = 8 Mbyte 110 = 16 Mbyte 111 = 32 Mbyte
EN Bit 0	**Chip-Select Enable—**This write-only bit enables each chip-select.	0 = Disabled. 1 = Enabled.

CSF Chip-Select Register F **0x(FF)FFF192**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	RO	SOP	ROP	UPSIZ			DRAM	FLASH	BSW	WS3–1			SIZ			EN
TYPE	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 7-14. Chip-Select Register F Description

Name	Description	Setting
RO Bit 15	**Read-Only—**This bit sets the chip-select to read-only. Otherwise, read and write accesses are allowed. A write to a read-only area will generate a bus error if the BETEN bit of the SCR is set. See Section 6.2.1, “System Control Register,” on page 6-2 for more information.	0 = Read/write. 1 = Read-only.
SOP Bit 14	**Supervisor-Use-Only Protected Memory Block—**This bit sets the protected memory block to supervisor-only; otherwise, both supervisor and user accesses are allowed. Attempts to access the supervisor-only area result in a bus error if the BETEN bit of the SCR is set. See Section 6.2.1, “System Control Register,” on page 6-2 for more information.	0 = Supervisor/user. 1 = Supervisor-only.
ROP Bit 13	**Read-Only for Protected Memory Block—**This bit sets the protected memory block to read-only. Otherwise, read and write accesses are allowed. If you write to a read-only area, you will get a bus error.	0 = Read/write. 1 = Read-only.
UPSIZ Bits 12–11	**Unprotected Memory Block Size—**This field determines the unprotected memory range of the chip-select.	00 = 32K. 01 = 64K. 10 = 128K. 11 = 256K.
Reserved Bit 10	Reserved	This bit is reserved and should be set to 0.
DRAM Bit 9	**DRAM Selection—**This bit is used to enable DRAM function in $\overline{\text{CSF}}$/RAS1/SDCS1 pin.	0 = Select $\overline{\text{CSF}}$. 1 = Select RAS1/SDCS1.
FLASH Bit 8	**Flash Memory Support—**When enabled, this bit provides support for flash memory by forcing the $\overline{\text{LWE}}$/$\overline{\text{UWE}}$ signal to go active after chip-select. **Note:** This bit is used for expanded memory size for CSD when the DRAM bit in the CSD register is enabled.	0 = The chip-select and $\overline{\text{LWE}}$/$\overline{\text{UWE}}$ signals go active at the same clock edge. 1 = The chip-select signal goes low 1 clock before $\overline{\text{LWE}}$/$\overline{\text{UWE}}$.

Table 7-14. Chip-Select Register F Description (Continued)

Name	Description	Setting
BSW Bit 7	**Data Bus Width—**This bit sets the data bus width for this chip-select area.	0 = 8 bit. 1 = 16 bit.
WS3–1 Bits 6–4	**Wait State—**This field determines the number of wait states added before an internal $\overline{\text{DTACK}}$ signal is returned for this chip-select. **Note:** When using the external $\overline{\text{DTACK}}$ signal, you must configure the BUSW/$\overline{\text{DTACK}}$/PG0 pin.	000 = 0 + WS0 wait states. 001 = 2 + WS0 wait states. 010 = 4 + WS0 wait states. 011 = 6 + WS0 wait states. 100 = 8 + WS0 wait states. 101 = 10 + WS0 wait states. 110 = 12 + WS0 wait states. 111 = External $\overline{\text{DTACK}}$. When using the external $\overline{\text{DTACK}}$ signal, you must select $\overline{\text{DTACK}}$ function in Port G. WS0 is the FWS0, EWS0, DWS0, CWS0, BWS0, or AWS0 bit in the CSCTRL1 register.
SIZ Bits 3–1	**Chip-Select Size—**This field determines the memory range of the chip-select.	000 = 256 kbyte 001 = 512 kbyte 010 = 1 Mbyte 011 = 2 Mbyte 100 = 4 Mbyte 101 = 8 Mbyte 110 = 16 Mbyte 111 = 32 Mbyte
EN Bit 0	**Chip-Select Enable—**This write-only bit enables each chip-select.	0 = Disabled. 1 = Enabled.

CSG Chip-Select Register G **0x(FF)FFF194**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	RO	SOP	ROP	UPSIZ		BCLKMD										EN
TYPE	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 7-15. Chip-Select Register G Description

Name	Description	Setting
RO Bit 15	**Read-Only—**This bit sets the chip-select to read-only. Otherwise, read and write accesses are allowed. A write to a read-only area will generate a bus error if the BETEN bit of the SCR is set. See Section 6.2.1, “System Control Register,” on page 6-2 for more information.	0 = Read/write. 1 = Read-only.

Table 7-15. Chip-Select Register G Description (Continued)

Name	Description	Setting
SOP Bit 14	**Supervisor-Use-Only Protected Memory Block—**This bit sets the protected memory block to supervisor-only; otherwise, both supervisor and user accesses are allowed. Attempts to access the supervisor-only area result in a bus error if the BETEN bit of the SCR is set. See Section 6.2.1, “System Control Register,” on page 6-2 for more information.	0 = Supervisor/user. 1 = Supervisor-only.
ROP Bit 13	**Read-Only for Protected Memory Block—**This bit sets the protected memory block to read-only. Otherwise, read and write accesses are allowed. If you write to a read-only area, you will get a bus error.	0 = Read/write. 1 = Read-only.
UPSIZ Bits 12–11	**Unprotected Memory Block Size—**This field determines the unprotected memory range of the chip-select.	00 = 32K. 01 = 64K. 10 = eSRAM size. 11 = reserved.
BCLKMD Bit 10	**Burst Clock Mode—**This bit is used to dynamically turn/off the clock to embedded SRAM.	0 (default) = constant clock. 1 = dynamic clock.
Reserved Bits 9–4	Reserved	These bits are reserved and should be set to 0.
EN Bit 0	**Chip-Select Enable—**This write-only bit enables each chip-select.	0 = Disabled. 1 = Enabled.

7.3.3 Emulation Chip-Select Register

In addition to the eight general-purpose chip-select signals, the MC68EZ328 has an emulation chip-select register (EMUCS) that is specifically designed for the in-circuit emulation module. This register provides wait states 12–0, depending on the type of chip used. External logic ($\overline{DTACK}$) may also be used to have longer wait states. EMUCS is only valid for the 0xFFFC0000–0xFFFDFFFF memory location.

EMUCS Emulation Chip-Select Register **0x(FF)FFF118**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
										WS3–1						
TYPE										rw	rw	rw				
RESET	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	0x0060															

Table 7-16. Emulation Chip-Select Register Description

Name	Description	Setting
Reserved Bits 15–7	Reserved	These bits are reserved and should be set to 0.
WS3–1 Bits 6–4	**Wait State—**This field contains the 3 most significant bits of the 4-bit wait-state value. The least significant bit is located in the chip-select control register 1. The value of these 4 bits determines the number of wait states added to a bus cycle before an internal $\overline{DTACK}$ is asserted to terminate the chip-select cycle.	000 = 0 + WS0 wait states. 001 = 2 + WS0 wait states. 010 = 4 + WS0 wait states. 011 = 6 + WS0 wait states. 100 = 8 + WS0 wait states. 101 = 10 + WS0 wait states. 110 = 12 + WS0 wait states. 111 = External $\overline{DTACK}$. When using the external $\overline{DTACK}$ signal, you must select $\overline{DTACK}$ function in Port G. WS0 is the EMUWS0 bit in the CSCTRL1 register.
Reserved Bits 3–0	Reserved	These bits are reserved and should be set to 0.

7.3.4 Chip-Select Control Register 1

The chip-select control register 1 (CSCTRL1) is one of three registers that provide features to control a wide variety of different memory types. The CSCTRL1 register provides supplemental memory-control features for chip-select logic. Control features include 16-bit SRAM support, extended size for unprotected memory space, and extended size for DRAM. See the following register display and Table 7-17 on page 7-21.

CSCTRL1 Chip-Select Control Register 1 **0x(FF)FFF18A**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
FIELD	EWS0	EUPEN	SR16	EMUWS0	DWS0	CWS0	BWS1	AWS0	FWS0	FUPS2	EUPS2	DUPS2		CUPS2		BUPS2
TYPE	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 7-17. Chip-Select Control Register 1 Description

Name	Description	Setting
EWS0 Bit 15	**CSE Wait State Bit 0—**This bit is the lowest significant bit of the CSE wait state register.	Refer to Table 7-13 on page 7-15 on the chip-select register E for the wait state setting.
EUPEN Bit 14	**Extra UPSIZ Bit Enable—**This bit enables the BUPS2, CUPS2, DUPS2, EUPS2, and FUPS2 bits to work with the corresponding UPSIZ configuration bits. Hence, it provides a larger dynamic range with smaller granularity for the unprotected memory sizing.	0 = EUPEN bit not set. 1 = EUPEN bit set.
SR16 Bit 13	**16-Bit SRAM Enable—**This bit enables the use of 16-bit SRAM in chip-select group B memory space. It determines the functions of the $\overline{UWE}$/$\overline{UB}$ and $\overline{LWE}$/$\overline{LB}$ pins in CSB read/write cycles.	0 = $\overline{UWE}$ and $\overline{LWE}$ are selected for all CSB read/write cycles. 1 = $\overline{UB}$ and $\overline{LB}$ are selected for all CSB read/write cycles.
EMUWS0 Bit 12	**EMUCS Wait State Bit 0—**This bit is the lowest significant bit of the EMUCS wait state register.	Refer to Table 7-16 on page 7-20 on the EMUCS register for the wait state setting.
DWS0 Bit 11	**CSD Wait State Bit 0—**This bit is the lowest significant bit of the CSD wait state register.	Refer to Table 7-12 on page 7-12 on the chip-select register D for the wait state setting.
CWS0 Bit 10	**CSC Wait State Bit 0—**This bit is the lowest significant bit of the CSE wait state register.	Refer to Table 7-11 on page 7-11 on the chip-select register C for the wait state setting.
BWS1 Bit 9	**CSB Wait State Bit 1—**This bit represents bit 1 of the CSB wait state register.	Refer to Table 7-10 on page 7-9 on the chip-select register B for the wait state setting.
AWS0 Bit 8	**CSA Wait State Bit 0—**This bit is the lowest significant bit of the CSA wait state register.	Refer to Table 7-9 on page 7-8 on the chip-select register A for the wait state setting.
FWS0 Bit 7	**CSF Wait State Bit 0—**This bit is the lowest significant bit of the CSF wait state register.	Refer to Table 7-14 on page 7-17 on the chip-select register F for the wait state setting.

Table 7-17. Chip-Select Control Register 1 Description (Continued)

Name	Description	Setting
FUPS2 Bit 6	**UPSIZ Bit 2 for CSF Register—**This is the most significant bit for UPSIZ[2:0] when the EUPEN bit is set.	For information on calculating unprotected memory size, see Example 7-1 on page 7-22.
EUPS2 Bit 5	**UPSIZ Bit 2 for CSE Register—**This is the most significant bit for UPSIZ[2:0] when the EUPEN bit is set.	For information on calculating unprotected memory size, see Example 7-1 on page 7-22.
DUPS2 Bit 4	**UPSIZ Bit 2 for CSD Register—**This is the most significant bit for UPSIZ[2:0] when the EUPEN bit is set.	For information on calculating unprotected memory size, see Example 7-1 on page 7-22.
Reserved Bit 3	Reserved	This bit is reserved and should be set to 0.
CUPS2 Bit 2	**UPSIZ Bit 2 CSC Register—**This is the most significant bit for UPSIZ[2:0] when the EUPEN bit is set.	For information on calculating unprotected memory size, see Example 7-1.
Reserved Bit 1	Reserved	This bit is reserved and should be set to 0.
BUPS2 Bit 0	**UPSIZ Bit 2 CSB Register—**This is the most significant bit for UPSIZ[2:0] when the EUPEN bit is set.	For information on calculating unprotected memory size, see Example 7-1.

The unprotected memory size is calculated according to the chip-select addressing space and the UPSIZ value.

Example 7-1. Unprotected Memory Size Calculation

$$\text{Unprotected Size} = \frac{\text{Chip-Select Size}}{2^{(7-\text{UPSIZ})}}$$

For example, if SIZ[2:0] in CSD = 111 and UPSIZ[2:0] = 011, the unprotected size is calculated as follows:

$$32\ \text{Mbyte} / 2^{(7-3)} = 2\text{M}$$

7.3.5 Chip-Select Control Register 2

This register controls early cycle detection for the dynamic type of memory. It improves CPU access performance by generally removing one CPU wait state or by relaxing the timing requirement for the memory.

CSCTRL2 Chip-Select Control Register 2 **0x(FF)FFF10C**

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

Table 7-18. Chip-Select Control Register 2 Description

Name	Description	Setting
ECDD Bit 15	**Early Cycle Detection for Dynamic Memory—**This bit advances the timing, allowing the CPU to be used with dynamic memory access. It reduces wait states by one.	0 = Disabled. 1 = Enabled.
Reserved Bits 14–0	Reserved	These bits are reserved and should be set to 0.

7.3.6 Chip-Select Control Register 3

This register controls minor timing trims for static memory access.

CSCTRL3 Chip-Select Control Register 3 **0x(FF)FFF150**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	EWE	WPEXT	LCWS	AST	DST	CST	BWS0	LCS								
TYPE	rw	rw	rw	rw	rw	rw	rw	rw								
RESET	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
	0x9C00															

Table 7-19. Chip-Select Control Register 3 Description

Name	Description	Setting
EWE Bit 15	**End Write Early—**When this bit is set, the RAM write-enable signal negates before the CS signal is negated.	0 = Disabled. 1 = Enabled.
WPEXT Bit 14	**Write Pulse to CS Negation Margin Extension—**When EWE is set, WPEXT is set to extend the WE negation to CS negation by one more clock.	0 = Disabled. 1 = Enabled.

Table 7-19. Chip-Select Control Register 3 Description (Continued)

Name	Description	Setting
LCWS Bit 13	**Wait State Trim for LCD/DMA-SRAM Access—**When this bit is set, one additional wait state is added to the LCD/DMA-SRAM access cycle. For example, if the wait state is set to zero, the chip-select signal to SRAM lasts 2.5 CPU clock cycles in 68K access, and 2 cycles are used for LCD/DMA access. When LCWS is enabled, the LCD/DMA access is delayed; the LCD/DMA access is increased from 2 to 3 clock cycles.	0 = No additional wait state added. 1 = One additional wait state added.
AST Bit 12	**AS Toggle Enable—**Enables AS toggling between two 8-bit transfers.	0 = Disable AS toggling between two 8-bit transfers. 1 = Enable AS toggling between two 8-bit transfers.
DST Bit 11	**DS Toggle Enable—**Enables DS toggling between two 8-bit transfers.	0 = Disable DS toggling between two 8-bit transfers. 1 = Enable DS toggling between two 8-bit transfers.
CST Bit 10	**CS Toggle Enable—**Enables CS toggling between two 8-bit transfers.	0 = Disable CS toggling between two 8-bit transfers. 1 = Enable CS toggling between two 8-bit transfers.
BWS0 Bit 9	**CSB Wait State Bit 0—**This bit is the lowest significant bit of the CSB wait state register.	Refer to Table 7-10 on page 7-9 on the chip-select register B for the wait state setting.
LCS Bit 8	**Later CS—**When this bit is set, a 1/2 clock delay is added to chip-select B signals in 68K accesses, and a 1 clock delay is added to chip-select B signals during DMA accesses.	0 = Disable delayed $\overline{\text{CSB0}}$ and $\overline{\text{CSB1}}$. 1 = Enable delayed $\overline{\text{CSB0}}$ and $\overline{\text{CSB1}}$.
Reserved Bits 7–0	Reserved	These bits are reserved and should be set to 0.

Chapter 8
DRAM Controller

This chapter describes the DRAM controller for the MC68SZ328. The operation of the DRAM controller is closely linked to the chip-select logic. Refer to Chapter 7, “Chip-Select Module,” for more details.

8.1 DRAM Controller Features

The MC68SZ328 DRAM controller supports both EDO and synchronous DRAM and has the following features:

- Four banks of single data rate 64 Mbit, 128 Mbit, and 256 Mbit SDRAM per chip-select.
- 4 Mbit, 16 Mbit, and 64 Mbit self-refresh type EDO per chip-select with 8, 9, 10, and 11 column address.
- Two independent chip-selects.
- Up to four banks active simultaneously per SDRAM chip-select line.
- 66 MHz system clock.
- Programmed burst length of 4 for 16-bit SDRAM.
- Minimum wait state of 2 at 66 MHz for CPU access.
- Typical access time of 6-1-1-1 at 66 MHz for DMA burst access on SDRAM.
- Typical access time of 6-2-2-2 at 66 MHz for DMA burst access on EDO. Burst length is not limited to 4 for EDO.
- Two internal bus protocols: one 68K bus protocol and one bursty DMA bus protocol.
- Configurable row cycle delay (tRC), row precharge delay (tRP), row to column delay (tRCD), and column to data delay (CAS Latency).
- Data retention capability during user reset.
- Power-down mode and self-refresh mode for SDRAM power saving.
- Self-refresh mode for EDO power saving.

8.2 General Architecture

The memory controller in the MC68SZ328 is composed of a chip-select module, an EDO DRAM controller, and an SDRAM controller. The chip-select module has four chip-select pairs indexed from A to D, and each chip-select pair is split into upper and lower parts, so there are a total of eight chip-select lines available on the chip.

In addition to the eight chip-selects, there are two extra chip-selects for the DRAM controller that are indexed E and F. Control for E and F are independent so that slight variation in device type on E and F is allowed. The maximum size for each chip-select space is 32 MByte; therefore, the total system DRAM capacity is 64 MByte.

The two chip-selects for the DRAM controller can be programmed together as either an EDO DRAM controller or an SDRAM controller. This limits the system in that the EDO DRAM and SDRAM cannot co-exist. Selecting EDO DRAM signals or SDRAM signals to output from the chip occurs just after a power-on reset. The selection is derived from the enable bits controlling the multiplexer shown in Figure 8-1.

The EDO DRAM controller and the SDRAM controller can be considered as two independent modules except that some modules are shared by both. The blocks named SDRAM Controller and EDO Controller in Figure 8-1 are independent, finite state machines. All other modules drawn in the figure are shared by each state machine.

The block named Bus Interface in Figure 8-1 handles the 68K, bursty DMA protocol and converts the signals to a common form for the state machines so that the state machines need not determine whether access is 68K or DMA. To some extent, this arrangement also enhances the portability through different processor buses.

The block named Bank/Page Register in Figure 8-1 is responsible for detecting whether the current access falls within the same page of the previous access. In short, it detects page hits. In SDRAM mode, the maximum bank number is eight, and there can be eight pages activated simultaneously. In EDO mode, the maximum bank number is only two, and there can be two pages activated simultaneously.

The write data bus and read data bus are split and therefore are unidirectional. There is no bidirectional data bus inside the internal data path until the read and write data buses merge at the I/O pad and are ready to connect to external devices.

8.3 Block Diagram

Figure 8-1 is a block diagram of the DRAM controller.

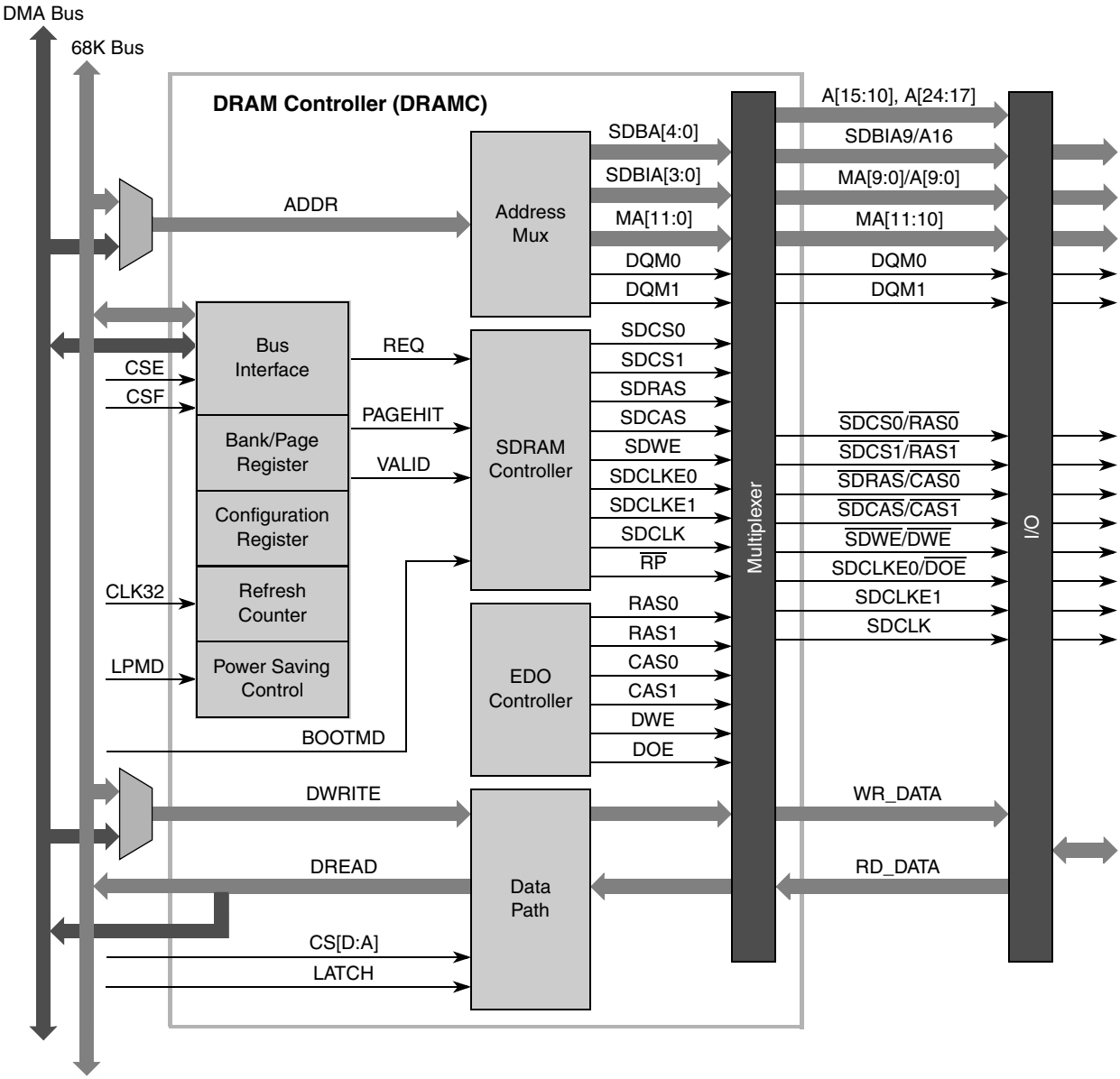


Figure 8-1. DRAM Controller Block Diagram

8.4 External Interface

Most SDRAM control signals are multiplexed with the EDO controller. After power up, the initialization process defines the multiplexer setting and chooses signals between the SDRAM and EDO controllers. Some address lines from the DRAM controller are further multiplexed dynamically with addresses from the chip-select module. Depending on the source of the current access, the output pin will be a DRAM address if the current access is on DRAM space, or the output pin will be a linear address if the current access is on SRAM access space. The output pin will be a linear address if the current access is on SRAM, flash, or ROM space.

8.5 Address Multiplexing

The DRAM controller multiplexed address bus aligns to the column addresses so that address line A0 always appears on pin MA0. With this alignment, the “folding point” in the multiplexor drives solely by the number of column address bits, although interleave mode causes a two bit shift to account for the bank addresses. Non-interleave mode supports column bus widths of 8 to 11 bits, although interleave mode only allows 8 and 9 bit widths. Table 8-1. and Table 8-2. summarize the multiplex options supported by the controller. Column addresses through A10 are driven regardless of the multiplexor configuration, although some of the lines are unused for smaller page sizes.

Table 8-1. SDRAM Address Multiplexing by Column Width

Column Bits		Memory Width	SDRAM Controller Pin											
IAM =0	IAM =1		MA 11	MA 10	MA 9	MA 8	MA 7	MA 6	MA 5	MA 4	MA 3	MA 2	MA 1	MA 0
			ROW											
8	-	16	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8
9	-	16	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9
	8	16	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
-	9	16	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
			COLUMN											
ALL		16	AP	AP	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

Indicates address lines are not required for this memory width.

Table 8-2. EDO Address Multiplexing by Column Width

Column Bits		Memory Width	EDO Controller Pin											
IAM= 0	-		MA 11	MA 10	MA 9	MA 8	MA 7	MA 6	MA 5	MA 4	MA 3	MA 2	MA 1	MA 0
			ROW											
8	-	16	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8
9	-	16	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9
10	-	16	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
11	-	16	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
			COLUMN											
ALL		16	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

Indicates address lines are not required for this memory width.

8.6 Programming Model

The programming model consists of sets of registers for SDRAM and EDO as well as a common register. There are two independent registers for two chip-selects in each type of DRAM. However, the two types of DRAM cannot co-exist in the system. If the SDRAM controller is enabled in chip-select space E, the EDO controller should not be enabled in chip-select space F. Otherwise, there could be unexpected results. Table 8-3 provides an address map of the registers.

Table 8-3. DRAM Controller Register Set

Address	Name	Width	Description	Reset Value
0xFFFFFC00	SDCTLe_H	16	SDRAM control register (high word) for chip-select E	0x0100
0xFFFFFC02	SDCTLe_L	16	SDRAM control register (low word) for chip-select E	0x0300
0xFFFFFC04	SDCTLf_H	16	SDRAM control register (high word) for chip-select F	0x0100
0xFFFFFC06	SDCTLf_L	16	SDRAM control register (low word) for chip-select F	0x0300
0xFFFFFC08	EDOCTLe_H	16	EDO control register (high word) for chip-select E	0x0000
0xFFFFFC0A	EDOCTLe_L	16	EDO control register (low word) for chip-select E	0x0000
0xFFFFFC0C	EDOCTLf_H	16	EDO control register (high word) for chip-select F	0x0000
0xFFFFFC0E	EDOCTLf_L	16	EDO control register (low word) for chip-select F	0x0000
0xFFFFFC10	SECTL	16	Secondary control register	0x0000

8.6.1 Secondary Control Register

The secondary control register provides controls for refresh clock source, external refresh counter clock prescaler, self-refresh mode, and software reset for SDRAM or EDO DRAM. The register settings are described in detail in Table 8-4.

SECTL Secondary Control Register **0x(FF)FFFC10**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	REFCLK			REFPS					RM						RST1	RST0
TYPE	rw			rw	rw	rw	rw	rw	rw						rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
	0x0023															

Table 8-4. Secondary Control Register Settings

Name	Description	Setting
REFCLK Bit 15	**Refresh Clock Source—**This bit selects the refresh counter clock source. This is either the 32.768 kHz clock or the system clock from the clock generator.	0 = 32.768 kHz clock. 1 = System clock from clock generator.
Reserved Bits 14–13	Reserved	These bits are reserved and should be set to 0.
REFPS Bits 12–8	**External Refresh Counter Clock Prescaler—**These bits define the divider value for the refresh counter clock. Using this on the 32 kHz clock source can generate refresh intervals of greater than 15 µs, which is common for low-power DRAM.	Divider value is 16 x (REFPS + 1) if REFCLK = 1. Divider value is (REFPS + 1) if REFCLK = 0.
RM Bit 7	**Self-Refresh Mode—**Setting this bit initiates self-refresh mode. The controller is able to exit self-refresh mode automatically when there is access to the DRAM. When there is no access for a fixed period, self-refresh mode is entered automatically. Setting this bit will, in effect, extend the latency of CPU/DMAC/LCD access because extra time is needed to wake up the DRAM. Therefore, it is best to clear this bit when the DRAM will be used for an extended period of time. The RM bit applies to both EDO DRAM and SDRAM.	0 = Normal mode. 1 = Self-refresh mode.
Reserved Bit 6	Reserved	These bits are reserved and should be set to 0.
Reserved Bit 5	Reserved	These bits are reserved and should be set to 1.
Reserved Bits 4–2	Reserved	These bits are reserved and should be set to 0.
RST1, RST0 Bits 1–0	**Software Initiated Local Module Reset—**These bits are responsible for generating a local module reset to DRAM controller. Writing 1 to RST0 one and 0 to RST1 will result in a 1-cycle reset pulse to the controller, while writing 1 to both RST0 and RST1 will result in a 2-cycle reset pulse to the controller. **Note:** The RST1, RST0 bits always read 1.	See description.

8.6.2 SDRAM Control Registers

SDCTLe_H and SDCTLf_H are SDRAM control registers (high word) for chip-selects E and F, respectively. SDCTLe_L and SDCTLf_L are SDRAM control registers (low word) for chip-selects E and F, respectively. Chip-selects E and F have the same register bit definitions. Table 8-5 and Table 8-6 provide descriptions and settings of each bit field in the registers. Table 8-7 lists the SREFR bit-field encoding values.

SDCTLe_H CSE SDRAM Control Register (High Word) **0x(FF)FFFC00**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	SDE	SMODE					SROW				SCOL		IAM			
TYPE	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
	0x0100															

SDCTLf_H CSF SDRAM Control Register (High Word) **0x(FF)FFFC04**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	SDE	SMODE					SROW				SCOL		IAM			
TYPE	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
	0x0100															

Note: An asterisk (*) indicates that the reset state is affected by bootmd[1:0] (SDCTLx_H only).

Table 8-5. CSE/CSF SDRAM Control Register (High Word) Settings

Name	Description	Setting
SDE Bit 15	**SDRAM Controller Enable—**This bit enables the SDRAM controller. The module is disabled out of a power-up reset.	0 = Disable the SDRAM controller. 1 = Enable the SDRAM controller.
SMODE Bits 14–12	**SDRAM Controller Operating Mode—**These bits determine the operating mode of the SDRAM controller. In addition to normal operating mode, the controller is capable of operating in alternate modes that are listed in Table 8-8 on page 8-10.	See Table 8-8.
Reserved Bits 11–10	Reserved	These bits are reserved and should be set to 0.
SROW Bits 9–8	**SDRAM Row Address Width—**This control field 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). A hardware reset sets this field to 01, resulting in a 12-bit row address.	00 = 11 row addresses. 01 = 12 row addresses. 10 = 13 row addresses. 11 = Reserved.

Table 8-5. CSE/CSF SDRAM Control Register (High Word) Settings (Continued)

Name	Description	Setting
Reserved Bits 7–6	Reserved	These bits are reserved and should be set to 0.
SCOL Bits 5–4	**SDRAM Column Address Width—**The SCOL control field is used to specify the number of column addresses in the memory array and will determine the break point in the address multiplexer. The 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. The column address width is initialized to 8 by hardware reset.	00 = 8 column addresses. 01 = 9 column addresses. 10 = 10 column addresses. 11 = Reserved.
IAM Bit 3	**SDRAM Interleaved Address Mode—**Bank address alignment is controlled by the IAM control bit. The DRAM controller supports two bank address alignments that will satisfy most system requirements. When IAM is disabled, bank addresses are more significant than row and column addresses, resulting in a linear addressing of the banks through the memory map. When IAM is enabled, bank addresses fall between the row and column addresses, resulting in an interleaved memory map with the banks alternately striped through the memory region. Bank address bit placement significantly affects how well the SDRAM page buffers are utilized, with a corresponding impact on system performance.	0 = Interleaved address disabled. 1 = Interleaved address enabled. See Section 8.6.2.2 for details
Reserved Bits 2–0	Reserved	These bits are reserved and should be set to 0.

SDCTLe_L CSE SDRAM Control Register (Low Word) 0x(FF)FFFC02

BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
SREFR		CLKST				SCL			SRP	SRCD			SRC		
TYPE	rw	rw	rw	rw		rw	rw		rw	rw	rw		rw	rw	rw
RESET	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
	0x0300														

SDCTLf_L CSF SDRAM Control Register (Low Word) 0x(FF)FFFC06

BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
SREFR		CLKST				SCL			SRP	SRCD			SRC		
TYPE	rw	rw	rw	rw		rw	rw		rw	rw	rw		rw	rw	rw
RESET	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
	0x0300														

Table 8-6. CSE/CSF SDRAM Control Register (Low Word) Settings

Name	Description	Setting
SREFR Bits 15–14	**SDRAM Refresh Rate—**This control bit field enables/disables SDRAM refresh cycles. It also controls the bulk refresh size and, hence, the refresh rate. Refresh cycles are referenced to a 32 kHz clock or system clock depending on the refresh clock selection bit, REFCLK. The refresh rate is also based on the prescaler value, REFPS. At each rising edge, 1, 2, or 4 rows are refreshed as determined by this bit field. Multiple refresh cycles will be separated by the row cycle delay specified in the SRC control field. Refresh is disabled during hardware reset.	00 = Refresh disabled. 01 = 1 row per REFCLK. 10 = 2 rows per REFCLK. 11 = 4 rows per REFCLK. See Table 8-7 on page 8-10 for details.
CLKST Bits 13–12	**SDRAM Clock Suspend Timeout—**This field determines whether the SDRAM will be placed in a clock suspend condition after a selectable delay from the last access. The suspend timeout can be triggered on either the absence of an active bank (CLKST = 01) or a clock count from the last access (CLKST = 10 or 11). Count-based timeouts do not force the SDRAM into an idle condition (any active banks remain open). The clock suspend feature is disabled by hardware reset. Section 5.3.3, “CPU Clock Control,” on page 5-8 provides additional information about this operating mode.	00 = Timeout disabled. 01 = Timeout occurs when no banks are active. 10 = Timeout occurs 64 clocks after completion of last access. 11 = Timeout occurs 128 clocks after completion of last access.
Reserved Bits 11–10	Reserved	These bits are reserved and should be set to 0.
SCL Bits 9–8	**SDRAM CAS Latency—**This field 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. This control field has no effect on write cycles. CAS latency is initialized to 3 clocks following a hardware reset.	00 = Reserved. 01 = 1 clock. 10 = 2 clocks. 11 = 3 clocks. See Figure 8-3 on page 8-15 for details.
Reserved Bit 7	Reserved	These bits are reserved and should be set to 0.
SRP Bit 6	**SDRAM Row Precharge Delay—**This control bit determines the number of idle clocks inserted between a precharge command and the next row activate command to the same bank. A hardware reset initializes the controller to insert 2 clocks.	0 = 3 clocks will be inserted between a precharge and the next row activate command. 1 = 2 clocks will be inserted between a precharge and the next row activate command. See Figure 8-4 on page 8-15 for details.
SRCD Bits 5–4	**SDRAM Row to Column Delay—**This field determines the number of clocks inserted between a row activate command and a subsequent read or write command to the same bank. Hardware reset initializes the delay to 4 clocks.	00 = 4 clocks. 01 = 1 clock. 10 = 2 clocks. 11 = 3 clocks.
Reserved Bit 3	Reserved	This bit is reserved and should be set to 0.

Table 8-6. CSE/CSF SDRAM Control Register (Low Word) Settings (Continued)

Name	Description	Setting
SRC Bits 2–0	**SDRAM Row Cycle Delay—**This control field determines the minimum delay between a refresh and any subsequent refresh or read/write access. This delay corresponds to the minimum row cycle time captured in the t_{RC}/t_{RFC} memory timing spec. The value programmed in SRC is the number of clocks inserted between the refresh and the subsequent refresh/activate command. An example timing diagram for SRC = 3 appears in Figure 8-6. **Note:** The SRC control field is not used to enforce t_{RC} timing for row activate to row activate within the same bank because this is implicitly guaranteed by the sum of t_{RCD} + t_{CL} + t_{RP}.	000 = 8 clocks. 001 = 1 clock. 010 = 2 clocks. 011 = 3 clocks. 100 = 4 clocks. 101 = 5 clocks. 110 = 6 clocks. 111 = 7 clocks. See Figure 8-6 for details.

Table 8-7. SREFR Bit Field Encoding

SREFR[1:0]	Rows Per Refresh Clock	Rows / 64 mS @ 32 kHz*	Row Rate @ 32 kHz	Row Rate @ 16 kHz (REFPS = 1)
00	Refresh disabled			
01	1	2048	31.25 µs	62.5 µs
10	2	4096	15.62 µs	31.25 µs
11	4	8192	7.81 µs	15.62 µs
*This column shows the approximate number of rows refreshed in 64 mS				

8.6.2.1 SDRAM Controller Operating Mode

The modes listed in Table 8-8 are primarily used for SDRAM initialization. Any access to the SDRAM memory space during operation in one of the alternate modes results in corresponding special cycles being run. Moving from normal mode 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 making a transition out of normal read/write mode. Operating mode details are provided in Section 26.4.3, “SDRAM Operation,” on page 26-10. Reset initializes the operating mode to normal read/write.

Table 8-8. SMODE Bit-Field Encoding

SMODE[2:0]	Operating Mode
000	Normal read/write
001	Precharge command
010	Auto-refresh command
011	Load mode register command
100	Reserved
101	Reserved

Table 8-8. SMODE Bit-Field Encoding (Continued)

SMODE[2:0]	Operating Mode
110	Reserved
111	Reserved

8.6.2.2 SDRAM Interleaved Address Mode

For IAM = 0, the address bus is allocated (from least to most significant) to column addresses, then to row addresses, and finally to the bank addresses. This configuration results in a linear address map that flows through each page in the first bank, into the second bank, and so on. Linear bank addressing is illustrated in the left half of Figure 8-2 and is best suited to applications with large continuous blocks of linear-accessed data such as an LCD display buffer. IAM is forced to 0 at reset.

The second configuration, for IAM = 1, is better suited to CPU code space. In this mode, the addresses are allocated to column addresses, then to the bank addresses, and finally to the row addresses. This results in an interleaved memory map (illustrated by the right half of Figure 8-2) where the banks alternate at each SDRAM page boundary. 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.

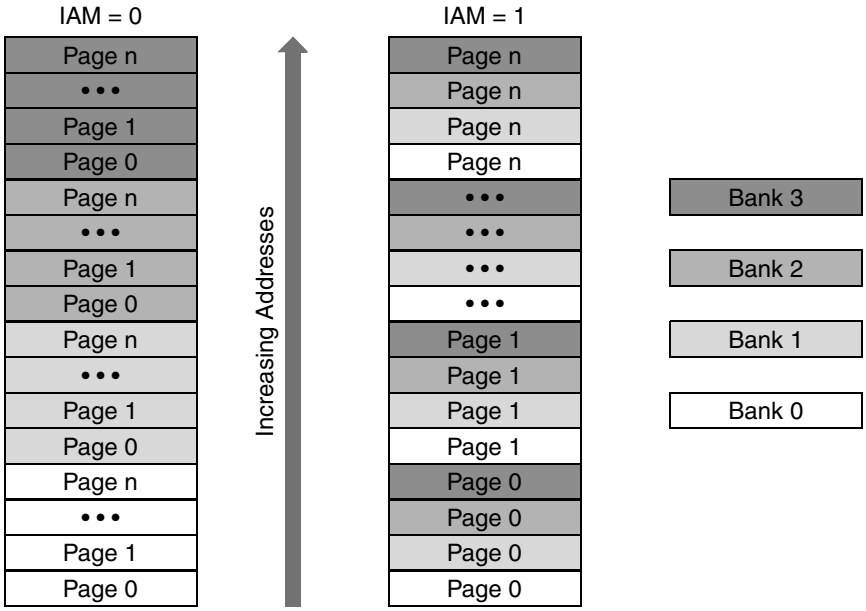


Figure 8-2. Memory Bank Interleaving Options

8.6.3 EDO Control Registers

The EDO control registers for the two chip-select functions have identical structures with different addresses. Table 8-9 and Table 8-10 provide descriptions and settings of each bit field in the registers. Table 8-11 lists the EREFR bit-field encoding values.

EDOCTLe_H CSE EDO Control Register (High Word) **0x(FF)FFFC08**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	EDE		EREFR				EROW				ECOL					
TYPE	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															

EDOCTLf_H CSF EDO Control Register (High Word) **0x(FF)FFFC0C**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	EDE		EREFR				EROW				ECOL					
TYPE	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 8-9. CSE/CSF EDO Control Register (High Word) Settings

Name	Description	Setting
EDE Bit 15	**EDO Controller Enable—**This bit enables or disables the EDO controller. The module is disabled after power-up reset.	0 = Disable. 1 = Enable.
Reserved Bit 14	Reserved	This bit is reserved and should be set to 0.
EREFR Bits 13–12	**EDO Refresh Rate—**These bits determine the bulk refresh size in every refresh request kick. Normally there is a refresh request in every positive edge of the 32 kHz clock. Hence, these bits determine the average refresh rate.	00 = Refresh disabled. 01 = 1 row per REFCLK. 10 = 2 rows per REFCLK. 11 = 4 rows per REFCLK. See Table 8-11 for details.
Reserved Bits 11–10	Reserved	These bits are reserved and should be set to 0.
EROW Bits 9–8	**EDO Row Address Width—**These bits define the row address width of the EDO DRAM used. These bits also affect the page address comparators.	00 = 11 row addresses. 01 = 12 row addresses. 10 = 13 row addresses. 11 = 10 row addresses.
Reserved Bits 7–6	Reserved	These bits are reserved and should be set to 0.

Table 8-9. CSE/CSF EDO Control Register (High Word) Settings (Continued)

Name	Description	Setting
ECOL Bits 5–4	**EDO Column Address Width—**These bits control the column width of the EDO DRAM and, hence, define the break point of the row/column address multiplexer.	00 = 8 column addresses. 01 = 9 column addresses. 10 = 10 column addresses. 11 = 11 column addresses.
Reserved Bits 3–0	Reserved	These bits are reserved and should be set to 0.

EDOCTLe_L CSE EDO Control Register (Low Word) 0x(FF)FFFC0A

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	ETRAS		ETC				ETPR				ETRC				RSTBR	
TYPE	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															

EDOCTLf_L CSF EDO Control Register (Low Word) 0x(FF)FFFC0E

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	ETRAS		ETC				ETPR				ETRC				RSTBR	
TYPE	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 8-10. EDO Control Register (Low Word) Settings

Name	Description	Setting
ETRAS Bits 15–14	**EDO CAS-Before-RAS RAS Pulse Width—**These bits control the RAS pulse width in the CAS-before-RAS refresh cycle. A hardware reset sets the width to 8 clocks.	00 = 8 clocks. 01 = 6 clocks. 10 = 4 clocks. 11 = 3 clocks.
ETC Bits 13–12	**EDO CAS Pulse Width—**These bits control the CAS active pulse width for adapting different memory speed grades and various system operating frequencies. A hardware reset sets the width to 4 clocks.	00 = 4 clocks. 01 = 1 clock. 10 = 2 clocks. 11 = 3 clocks. See Figure 8-7 for details.
Reserved Bits 11–10	Reserved	These bits are reserved and should be set to 0.
ETPR Bits 9–8	**EDO RAS Precharge Width—**These bits control the RAS precharge width for adapting different memory speed grades and various system operating frequencies. A hardware reset sets the width to 6 clocks.	00 = 6 clocks. 01 = 2 clocks. 10 = 3 clocks. 11 = 4 clocks. See Figure 8-8 for details.

Table 8-10. EDO Control Register (Low Word) Settings (Continued)

Name	Description	Setting
Reserved Bits 7–6	Reserved	These bits are reserved and should be set to 0.
ETRC Bits 5–4	**EDO RAS-to-CAS Delay—**These bits determine the RAS to CAS delay and also the row to column address transition time. A hardware reset sets the delay to 4 clocks.	00 = 4 clocks. 01 = 1 clock. 10 = 2 clocks. 11 = 3 clocks. See Figure 8-9 for details.
Reserved Bits 3–2	Reserved	These bits are reserved and should be set to 0.
RSTBR Bit 1	**EDO Reset Burst Reset—**This bit controls the refresh type during manual reset assertion.	0 = Normal distributed CBR refresh. 1 = Continuous burst refresh.
Reserved Bit 0	Reserved	This bit is reserved and should be set to 0.

Table 8-11. EREFR Bit Field Encoding

EREFR[1:0]	Rows Per Refresh Clock	Rows / 64 mS @ 32 kHz*	Row Rate @ 32 kHz	Row Rate @ 16 kHz (REFPS = 1)
00	Refresh disabled			
01	1	2048	31.25 µs	62.5 µs
10	2	4096	15.62 µs	31.25 µs
11	4	8192	7.81 µs	15.62 µs
*This column shows the approximate number of rows refreshed in 64 mS				

8.6.4 Timing Diagrams

Figure 8-3 through Figure 8-9 on page 8-18 illustrate SDRAM controller and EDO controller timing.

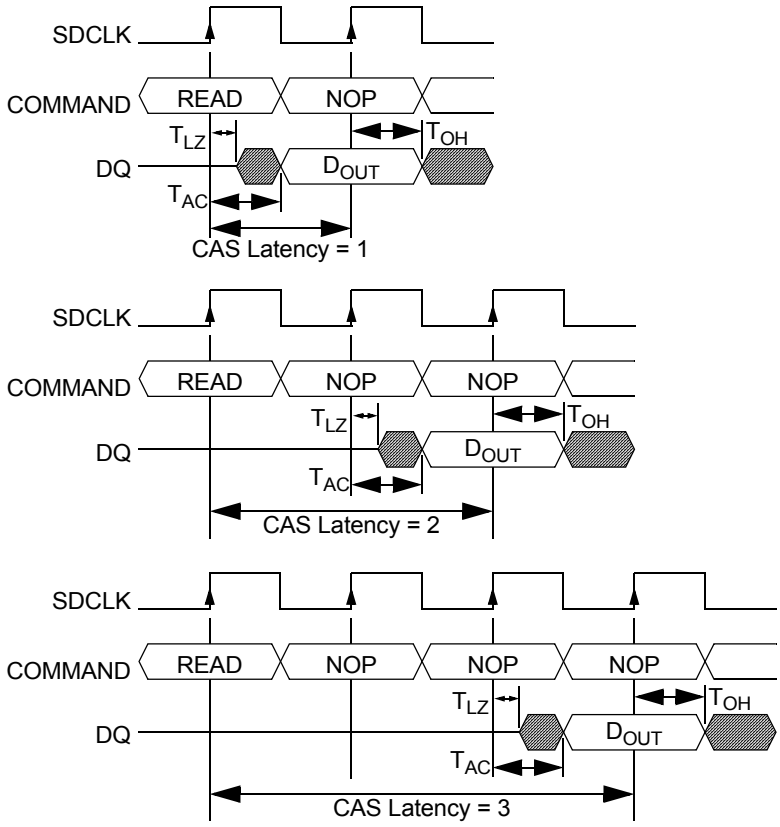


Figure 8-3. SDRAM CAS Latency Timing

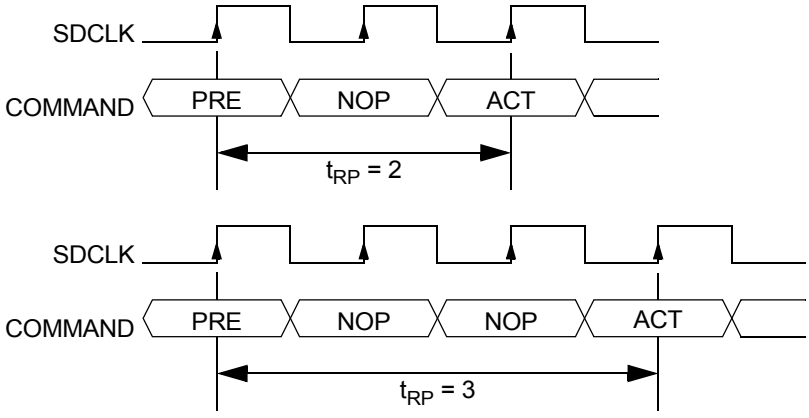


Figure 8-4. SDRAM Precharge Delay Timing

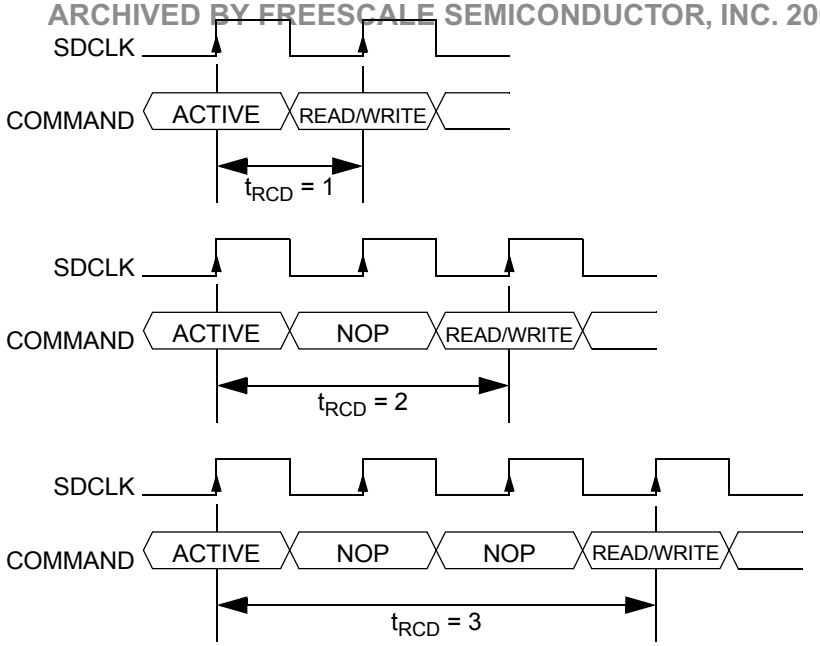


Figure 8-5. SDRAM Row to Column Delay Timing

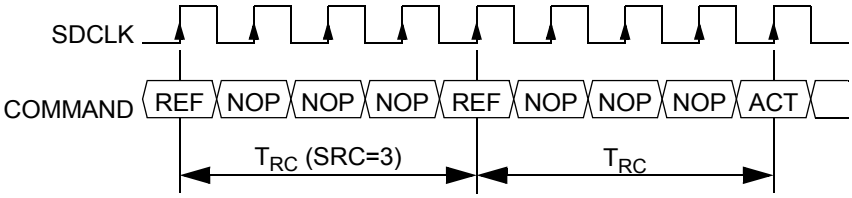


Figure 8-6. SDRAM Row Cycle Timing

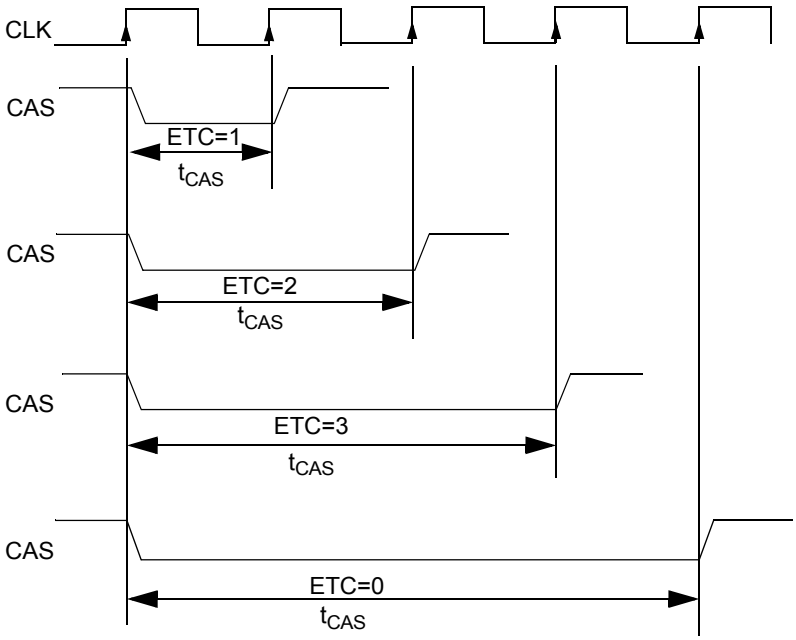


Figure 8-7. EDO CAS Pulse Width Timing

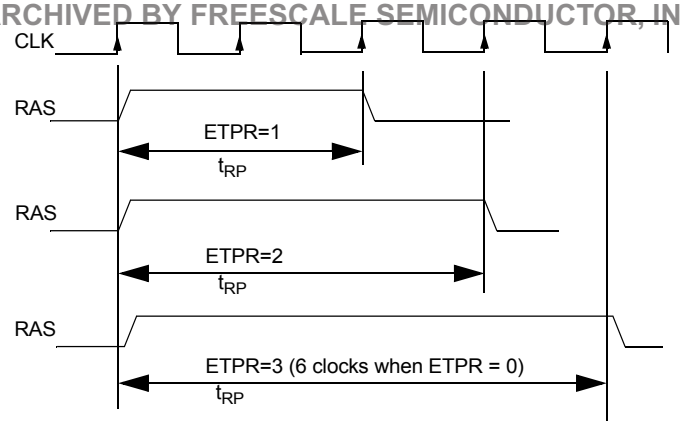


Figure 8-8. EDO RAS Precharge Timing

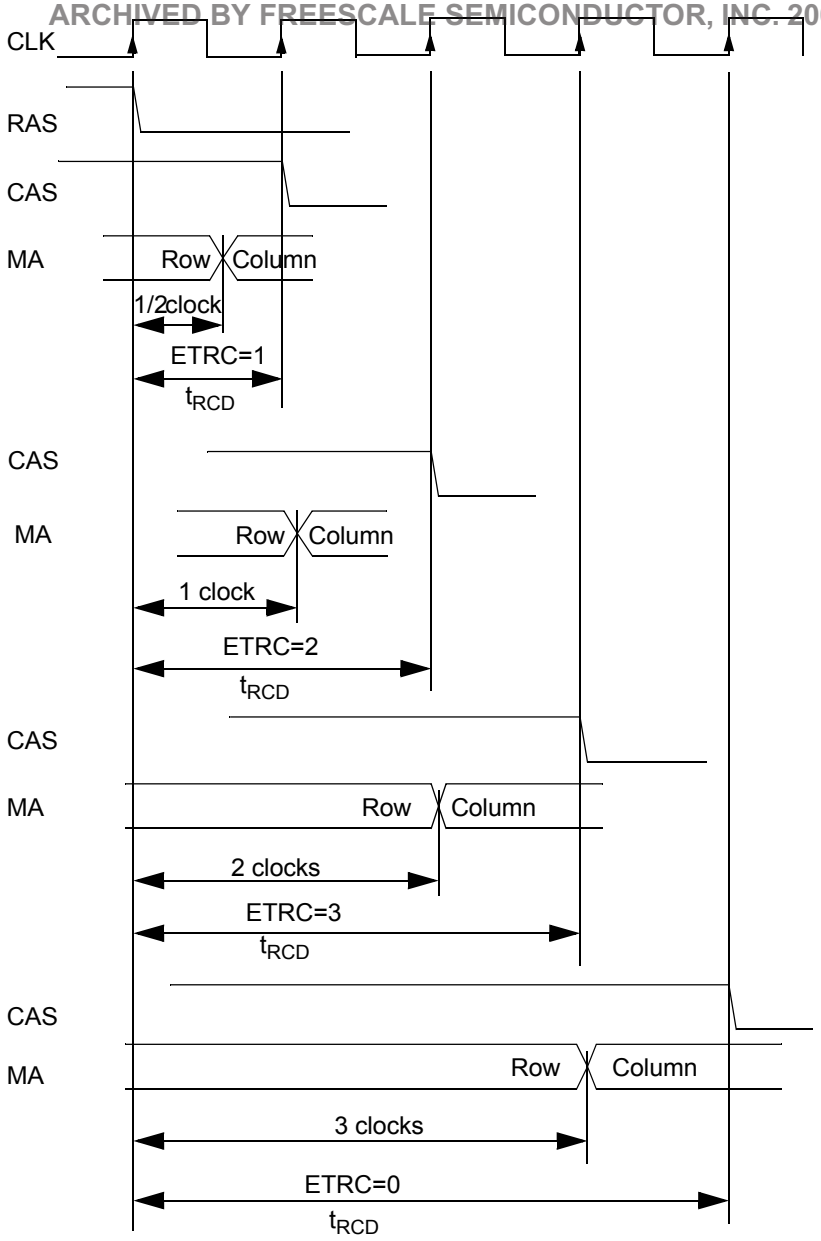


Figure 8-9. EDO RAS-to-CAS Delay and Row to Column Address Timing

8.7 Application Examples

This section describes and gives examples for memory configurations at the application level. Board level connection code examples are given for EDO DRAM and SDRAM.

EDO DRAM and SDRAM offer complex combinations of various device parameters like bus width, row address width, and column address width. Register settings, code examples, and initialization sequences are also provided in this section. Refer to Table 8-12 through Table 8-18 and Figure 8-10 through Figure 8-16, and Code Example 8-1 through Code Example 8-6.

8.7.1 Single 64 Mbit SDRAM (IAM=0, CSE)

Table 8-12 and Figure 8-10 show the SDRAM Control Register values and connection diagram using a single 64 Mbit SDRAM for IAM=0 case on CSE. Example initialization sequence follows in Code Example 8-1 on page 8-20.

Table 8-12. Single 4M x 16 SDRAM (IAM=0, CSE) Control Register Values

Control Field	Value
Density	8 Mbyte
Page size	512 bytes
ROW	12
COL	8
DSIZ	16 (D[15:0])
IAM	Non-interleaved

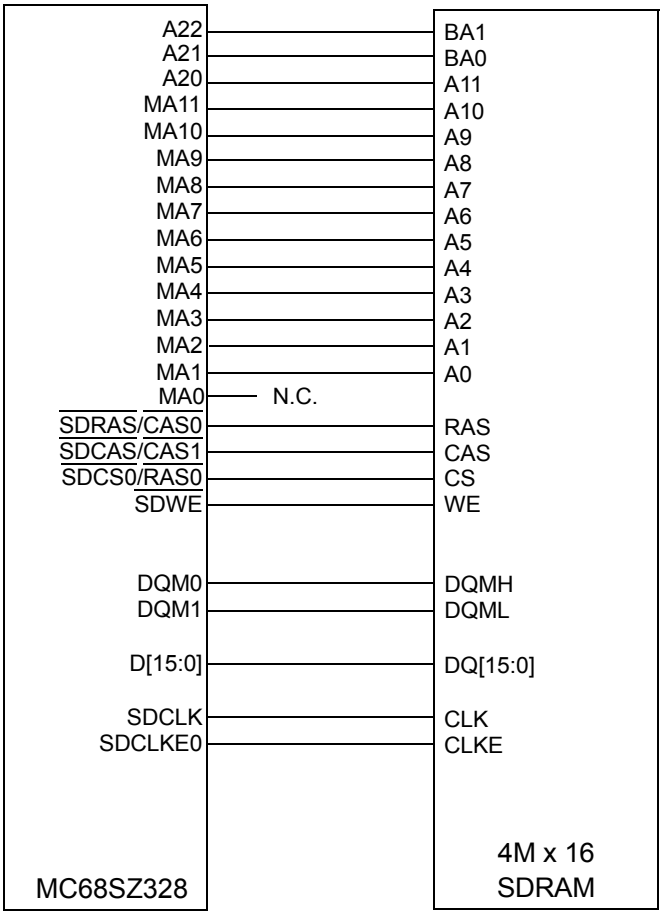


Figure 8-10. Single 64 Mbit SDRAM Connection Diagram (IAM = 0, CSE)

Code Example 8-1. Single 64 Mbit SDRAM (IAM = 0, CSE)

```
# Init SDRAM
# 4Mx16x1
# IAM=0
# ROW=12
# COL=8
# Cas Latency=2
# Chip Select CSE
#

# select dedicated I/O ports
# Port C
write -b 0xffffff412 =0x00
write -b 0xffffff413 =0x00

# Port D
write -b 0xffffff41B =0xF0
write -b 0xffffff41A =0xF0

#Enable CLK0, A23, A22, A21, A20, A24 (PF)
write -b 0xFFFFF42B=0x00
write -b 0xFFFFF423=0x0B

#Enable SDCS0,SDCS1 (PG)
write -b 0xFFFFF433=0x2D
write -b 0xFFFFF430=0xC2

#Enable WEB, SDRAS, SDCAS (PK)
write -b 0xFFFFF443=0x00
write -b 0xFFFFF440=0x0E

#Enable SDRAM signals (PM)
write -b 0xFFFFF44B=0x00
write -b 0xFFFFF448=0xFF

# Port P
write -b 0xffffff45a =0x00
write -b 0xffffff45b =0x00

# Set the Group E Base Address
write -w 0xFFFFF180=0x1000

# Set the Chip Select Register E
write -w 0xFFFFF190=0x029B

# Set Secondary Control Register
write -w 0xFFFFFC10=0x0000

# set precharge mode
write -w 0xFFFFFC00=0x9100
write -w 0xFFFFFC02=0x4200
                    # Base Address +
read -l 0x10080000  # issue precharge all and assert A10

# set auto refresh
write -w 0xFFFFFC00=0xA100
write -w 0xFFFFFC02=0x4200
read -l 0x10000000   # issue refresh
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000

# set mode register
write -w 0xFFFFFC00=0xB100
write -w 0xFFFFFC02=0x4200
read -l 0x10044400

# return to normal mode
write -w 0xFFFFFC00=0x8100
write -w 0xFFFFFC02=0x4200
```

8.7.2 Single 64 Mbit SDRAM (IAM=1, CSF)

Table 8-13 and Figure 8-11 show the SDRAM Control Register values and connection diagram using a single 64 Mbit SDRAM for IAM=1 case on CSF. Example initialization sequence follows in Code Example 8-2 on page 8-22.

Table 8-13. Single 4M x 16 SDRAM (IAM=1, CSF) Control Register Values

Control Field	Value
Density	8 Mbyte
Page size	512 bytes
ROW	12
COL	8
DSIZ	16 (D[15:0])
IAM	Interleaved

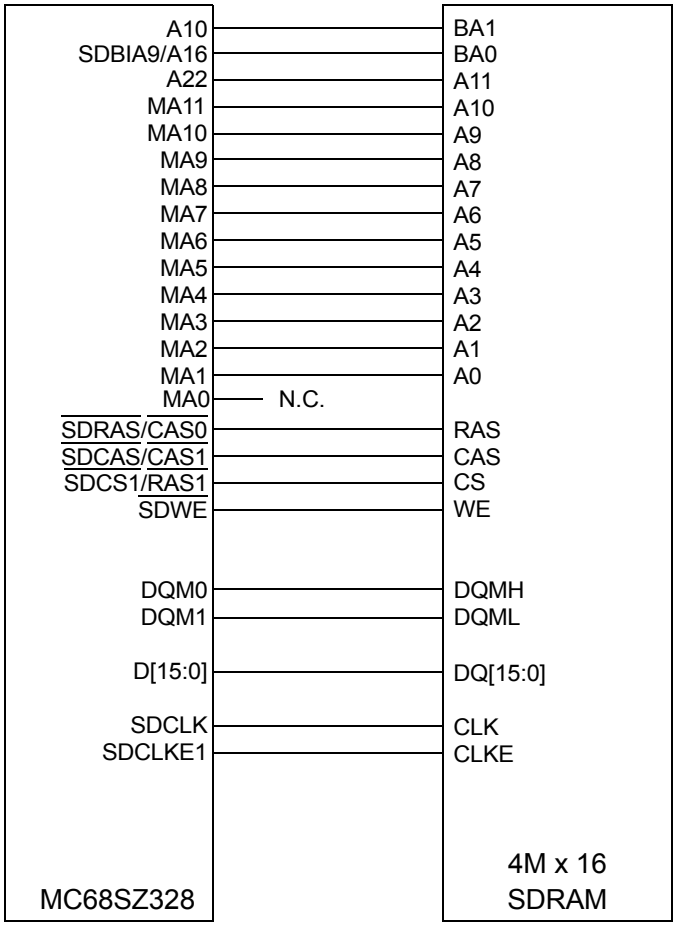


Figure 8-11. Single 64 Mbit SDRAM Connection Diagram (IAM = 1, CSF)

Code Example 8-2. Single 64 Mbit SDRAM (IAM = 1, CSF)

```
# Init SDRAM
# 4Mx16x1
# IAM=1
# ROW=12
# COL=8
# Cas Latency=2
# Chip Select CSE
#

# select dedicated I/O ports
# Port C
write -b 0xffffff412 =0x00
write -b 0xffffff413 =0x00

# Port D
write -b 0xffffff41B =0xF0
write -b 0xffffff41A =0xF0

#Enable CLK0, A23, A22, A21, A20, A24 (PF)
write -b 0xFFFFF42B=0x00
write -b 0xFFFFF423=0x0B

#Enable SDCS0,SDCS1 (PG)
write -b 0xFFFFF433=0x2D
write -b 0xFFFFF430=0xC2

#Enable WEB, SDRAS, SDCAS (PK)
write -b 0xFFFFF443=0x00
write -b 0xFFFFF440=0x0E

#Enable SDRAM signals (PM)
write -b 0xFFFFF44B=0x00
write -b 0xFFFFF448=0xFF

# Port P
write -b 0xffffff45a =0x00
write -b 0xffffff45b =0x00

# Set the Group E Base Address
write -w 0xFFFFF180=0x1000

# Set the Chip Select Register E
write -w 0xFFFFF190=0x029B

# Set Secondary Control Register
write -w 0xFFFFFC10=0x0000

# set precharge mode
write -w 0xFFFFFC00=0x9100
write -w 0xFFFFFC02=0x4200
                    # Base Address +
read -l 0x10200000  # issue precharge all and assert A10

# set auto refresh
write -w 0xFFFFFC00=0xA100
write -w 0xFFFFFC02=0x4200
read -l 0x10000000   # issue refresh
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000

# set mode register
write -w 0xFFFFFC00=0xB100
write -w 0xFFFFFC02=0x4200
read -l 0x10111000

# return to normal mode
write -w 0xFFFFFC00=0x8100
write -w 0xFFFFFC02=0x4200
```

8.7.3 Single 128 Mbit SDRAM (IAM=1, CSF)

Table 8-14 and Figure 8-12 show the SDRAM Control Register values and connection diagram using a single 128 Mbit SDRAM for IAM=1 case on CSF. Example initialization sequence follows in Code Example 8-3 on page 8-24.

Table 8-14. Single 8M x 16 SDRAM (IAM=1, CSF) Control Register Values

Control Field	Value
Density	16 Mbyte
Page Size	1024 byte
ROW	12
COL	9
DSIZ	16 (D[15:0])
IAM	Interleaved

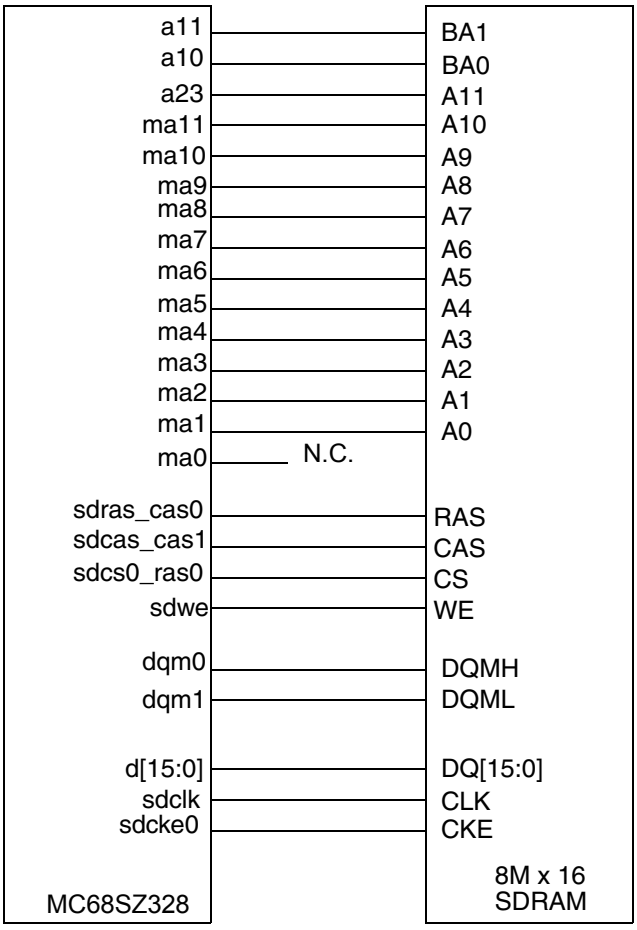


Figure 8-12. Single 128 Mbit SDRAM Connection Diagram (IAM = 1, CSF)

Code Example 8-3. Single 128 Mbit SDRAM (IAM = 1, CSF)

```
# Init SDRAM
# 8Mx16x1
# IAM=1
# CSF
# ROW=12
# COL=9
# Cas Latency=2
# Chip Select CSF
#

# select dedicated I/O ports
# Port C
write -b 0xffffff412 =0x00
write -b 0xffffff413 =0x00

# Port D
write -b 0xffffff41B =0xF0
write -b 0xffffff41A =0xF0

#Enable CLK0, A23, A22, A21, A20, A24 (PF)
write -b 0xFFFFF42B=0x00
write -b 0xFFFFF423=0x0B

#Enable SDCS0,SDCS1 (PG)
write -b 0xFFFFF433=0x2D
write -b 0xFFFFF430=0xC2

#Enable WEB, SDRAS, SDCAS (PK)
write -b 0xFFFFF443=0x00
write -b 0xFFFFF440=0x0E

#Enable SDRAM signals (PM)
write -b 0xFFFFF44B=0x00
write -b 0xFFFFF448=0xFF

# Port P
write -b 0xffffff45a =0x00
write -b 0xffffff45b =0x00

# Set the Group E Base Address
write -w 0xFFFFF182=0x1000

# Set the Chip Select Register E
write -w 0xFFFFF192=0x029D

# Set Secondary Control Register
write -w 0xFFFFFC10=0x0000

# set precharge mode
write -w 0xFFFFFC04=0x9118
write -w 0xFFFFFC06=0x4262
                    # Base Address +
read -l 0x10400000  # issue precharge all and assert A10

# set auto refresh
write -w 0xFFFFFC04=0xA118
write -w 0xFFFFFC06=0x4262
read -l 0x10000000   # issue refresh
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000

# set mode register
write -w 0xFFFFFC04=0xB118
write -w 0xFFFFFC06=0x4262
read -l 0x10222000

# return to normal mode
write -w 0xFFFFFC04=0x8118
write -w 0xFFFFFC06=0x4262
```

8.7.4 Single 128 Mbit SDRAM (IAM=0, CSE)

Table 8-15 and Figure 8-13 show the SDRAM Control Register values and connection diagram using a single 128 Mbit SDRAM for IAM=0 case on CSE. Example initialization sequence follows in Code Example 8-4 on page 8-26.

Table 8-15. Single 8M x 16 SDRAM (IAM=0, CSE) Control Register Values

Control Field	Value
Density	16 Mbyte
Page Size	1024 byte
ROW	12
COL	9
DSIZ	16 (D[15:0])
IAM	Non-interleaved

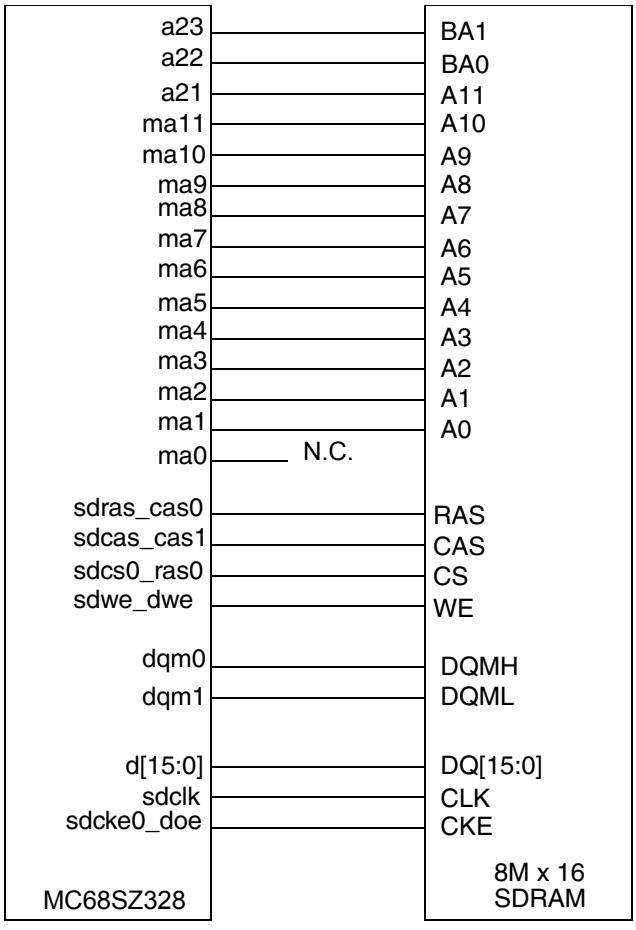


Figure 8-13. Single 128 Mbit SDRAM Connection Diagram (IAM = 0, CSE)

Code Example 8-4. Single 128 Mbit SDRAM (IAM = 0, CSE)

```
# Init SDRAM
# 8Mx16x1
# IAM=0
# CSE
# ROW=12
# COL=9
# IAM = 0
# Cas Latency=2
# Chip Select CSE
#

# select dedicated I/O ports
# Port C
write -b 0xfffff412 =0x00
write -b 0xfffff413 =0x00

# Port D
write -b 0xfffff41B =0xF0
write -b 0xfffff41A =0xF0

#Enable CLK0, A23, A22, A21, A20, A24 (PF)
write -b 0xFFFFF42B=0x00
write -b 0xFFFFF423=0x0B

#Enable SDCS0,SDCS1 (PG)
write -b 0xFFFFF433=0x2D
write -b 0xFFFFF430=0xC2

#Enable WEB, SDRAS, SDCAS (PK)
write -b 0xFFFFF443=0x00
write -b 0xFFFFF440=0x0E

#Enable SDRAM signals (PM)
write -b 0xFFFFF44B=0x00
write -b 0xFFFFF448=0xFF

# Port P
write -b 0xfffff45a =0x00
write -b 0xfffff45b =0x00

# Set the Group E Base Address
write -w 0xFFFFF180=0x1000

# Set the Chip Select Register E
write -w 0xFFFFF190=0x029D

# Set Secondary Control Register
write -w 0xFFFFFC10=0x0000

# set precharge mode
write -w 0xFFFFFC00=0x9110
write -w 0xFFFFFC02=0x4262
                    # Base Address +
read -l 0x10100000  # issue precharge all and assert A10

# set auto refresh
write -w 0xFFFFFC00=0xA110
write -w 0xFFFFFC02=0x4262
read -l 0x10000000   # issue refresh
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000

# set mode register
write -w 0xFFFFFC00=0xB110
write -w 0xFFFFFC02=0x4262
read -l 0x10088800

# return to normal mode
write -w 0xFFFFFC00=0x8110
write -w 0xFFFFFC02=0x4262
```

8.7.5 Single 256 Mbit SDRAM (IAM=1, CSE)

Table 8-16 and Figure 8-14 show the SDRAM Control Register values and connection diagram using a single 256 Mbit SDRAM for IAM=1 case on CSE. Example initialization sequence follows in Code Example 8-5 on page 8-28.

Table 8-16. Single 16M x 16 SDRAM (IAM=1, CSE) Control Register Values

Control Field	Value
Density	32 Mbyte
Page size	1024 bytes
ROW	13
COL	9
DSIZ	16 (D[15:0])
IAM	Interleaved

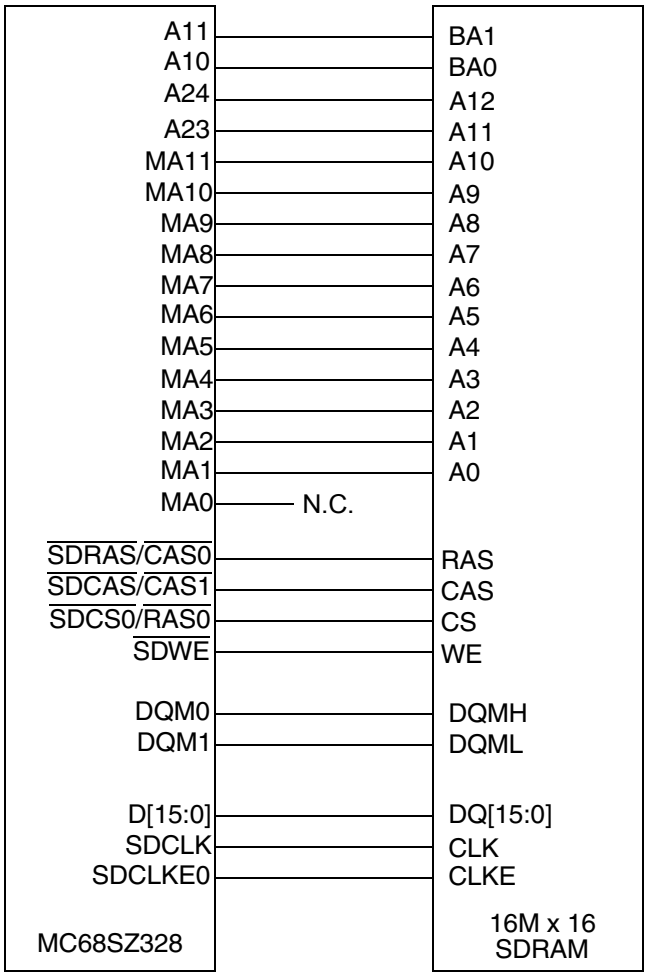


Figure 8-14. Single 256 Mbit SDRAM Connection Diagram (IAM = 1, CSE)

Code Example 8-5. Single 256 Mbit SDRAM (IAM = 1, CSE)

```
# Init SDRAM
# 16Mx16x1
# IAM=1
# CSF
# ROW=13
# COL=9
# Cas Latency=2
# Chip Select CSF
#

# select dedicated I/O ports
# Port C
#write -b 0xfffff412 =0x00
#write -b 0xfffff413 =0x00

# Port D
write -b 0xfffff41B =0xF0
write -b 0xfffff41A =0xF0

# Port E A24
#write -b 0xFFFFF423 =0xFB
write -b 0xFFFFF420 =0x04

#Enable CLK0, A23, A22, A21, A20(PF)
write -b 0xFFFFF42B=0x00
write -b 0xFFFFF423=0x0B

#Enable SDCS0,SDCS1 (PG)
write -b 0xFFFFF433=0x2D
write -b 0xFFFFF430=0xC2

#Enable WEB, SDRAS, SDCAS (PK)
write -b 0xFFFFF443=0x00
write -b 0xFFFFF440=0x0E

#Enable SDRAM signals (PM)
write -b 0xFFFFF44B=0x00
write -b 0xFFFFF448=0xFF

# Port P
write -b 0xfffff45a =0x00
write -b 0xfffff45b =0x00

# Set the Group F Base Address
write -w 0xFFFFF182=0x1000

# Set the Chip Select Register F
write -w 0xFFFFF192=0x029F

# Set Secondary Control Register
write -w 0xFFFFFC10=0x0000

# set precharge mode
write -w 0xFFFFFC04=0x9218
write -w 0xFFFFFC06=0x4262
                    # Base Address +
read -l 0x10400010  # issue precharge all and assert A10


# set auto refresh
write -w 0xFFFFFC04=0xA218
write -w 0xFFFFFC06=0x4262
read -l 0x10000010   # issue refresh
read -l 0x10000010
read -l 0x10000010
read -l 0x10000010
read -l 0x10000010
read -l 0x10000010
read -l 0x10000010
read -l 0x10000010

# set mode register
write -w 0xFFFFFC04=0xB218
write -w 0xFFFFFC06=0x4262
read -l 0x10222000


# return to normal mode
write -w 0xFFFFFC04=0x8218
write -w 0xFFFFFC06=0x4262
```

8.7.6 Single 256 Mbit SDRAM (IAM=0, CSE)

Table 8-17 and Figure 8-15 show the SDRAM Control Register values and connection diagram using a single 256 Mbit SDRAM for IAM=0 case on CSE. Example initialization sequence follows in Code Example 8-6 on page 8-30.

Table 8-17. Single 16M x 16 SDRAM (IAM=0, CSE) Control Register Values

Control Field	Value
Density	32 Mbyte
Page size	1024 bytes
ROW	13
COL	9
DSIZ	16 (D[15:0])
IAM	Non-interleaved

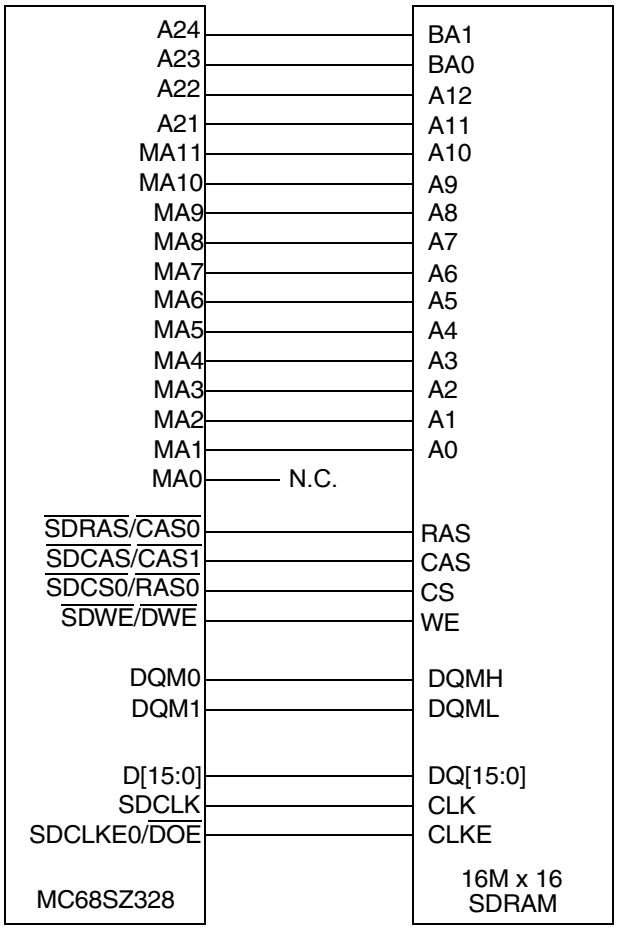


Figure 8-15. Single 256 Mbit SDRAM Connection Diagram (IAM = 0, CSE)

Code Example 8-6. Single 256 Mbit SDRAM (IAM = 0, CSE)

```
# Init SDRAM
# 16Mx16x1
# IAM=0
# CSE
# ROW=13
# COL=9
# Cas Latency=2
# Chip Select CSE
#

# select dedicated I/O ports
# Port C
write -b 0xffffff412 =0x00
write -b 0xffffff413 =0x00

# Port D
write -b 0xffffff41B =0xF0
write -b 0xffffff41A =0xF0

#Enable CLK0, A23, A22, A21, A20, A24 (PF)
write -b 0xFFFFF42B=0x00
write -b 0xFFFFF423=0x0B

#Enable SDCS0,SDCS1 (PG)
write -b 0xFFFFF433=0x2D
write -b 0xFFFFF430=0xC2

#Enable WEB, SDRAS, SDCAS (PK)
write -b 0xFFFFF443=0x00
write -b 0xFFFFF440=0x0E

#Enable SDRAM signals (PM)
write -b 0xFFFFF44B=0x00
write -b 0xFFFFF448=0xFF

# Port P
write -b 0xffffff45a =0x00
write -b 0xffffff45b =0x00

# Set the Group E Base Address
write -w 0xFFFFF180=0x1000

# Set the Chip Select Register E
write -w 0xFFFFF190=0x029F

# Set Secondary Control Register
write -w 0xFFFFFC10=0x0000

# set precharge mode
write -w 0xFFFFFC00=0x9210
write -w 0xFFFFFC02=0x4262
                    # Base Address +
read -l 0x10100000  # issue precharge all and assert A10

# set auto refresh
write -w 0xFFFFFC00=0xA210
write -w 0xFFFFFC02=0x4262
read -l 0x10000000   # issue refresh
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000
read -l 0x10000000

# set mode register
write -w 0xFFFFFC00=0xB210
write -w 0xFFFFFC02=0x4262
read -l 0x10088800

# return to normal mode
write -w 0xFFFFFC00=0x8210
write -w 0xFFFFFC02=0x4262
```

8.7.7 Single 16 Mbit EDO (CSE)

Table 8-14 and Figure 8-12 show the SDRAM Control Register values and connection diagram using a single 16 Mbit EDO on CSE. Example initialization sequence follows in Code Example 8-7.

Table 8-18. Single 1M x 16 EDO (CSE) Control Register Values

Control Field	Value
Density	2 Mbyte
Page size	512 bytes
ROW	12
COL	8
DSIZ	16 (D[15:0])
IAM	N.A.

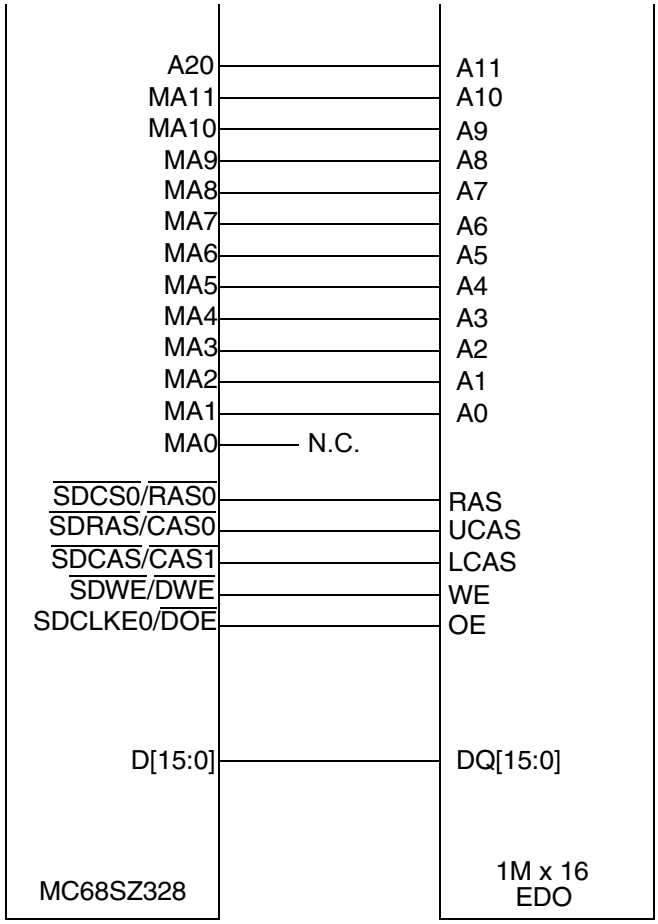


Figure 8-16. Single 16 Mbit EDO Connection Diagram (CSE)

Code Example 8-7. EDO (CSE) DRAM Initialization Code

```
Here (EDO-DRAM:4Mx16x1):
# Set the Group E Base Address
write -w 0xFFFFF180=0x1000

# Set the Chip Select Register E
write -w 0xFFFFF190=0x029B

# Set Secondary Control Register
write -w 0xFFFFFC10=0x0000

# Init EDO-DRAM register
write -w 0xFFFFFC08=0x8120
write -w 0xFFFFFC02=0x0200
```

Chapter 9
DMA Controller

The direct memory access controller (DMAC) of the MC68SZ328 provides two memory channels and four I/O channels to support a wide variety of DMA operations. Section 9.5, “Programming Model,” on page 9-6 includes descriptions of the DMAC registers which is divided into three parts for general, memory and I/O registers. Section 9.9, “DMAC Register Addresses,” on page 9-28 is a summary of the register addresses and memory map. Section 9.10, “DMA Request Table,” on page 9-30 concludes the chapter with a summary of the DMAC request signals and their associated peripherals.

9.1 Features

The following features are incorporated in the MC68SZ328 DMAC:

- Provides two memory channels and four I/O channels.
- Each I/O channel is configurable to respond to any of the defined DMA request signals except the external DMA requests which are supported by the two memory channels only.
- Supports 8- or 16-bit FIFO port size and memory port size for data transfer.
- DMA burst length is configurable for each channel.
- Bus utilization control is provided for memory channels.
- Provides data transfer complete and error (burst time-out or request time-out) interrupts to interrupt controller.
- DMA burst time-out error generation for both memory and I/O channels to terminate DMA cycle when the burst cannot be completed in a programmed timing period.
- DMA request time-out error generation for I/O channels to interrupt FLX68000 when a DMA request is not received during a programmed timing period.
- Supports repeat data transfer function.
- Provides block transfer function for memory channel. Supports byte aligned block transfer from one location to another.
- Provides priority mechanism to serve the DMA requests. Channel numbers 0 and 1 are assigned to memory channels. Channel numbers 2 through 5 are assigned to I/O channels. The channel with larger channel number has higher service priority.
- Provides $\overline{\text{DMA\_REQ0}}$ and $\overline{\text{DMA\_REQ1}}$, two external DMA request pins for external devices to initiate data transfer in memory channel 0 and 1 respectively.

9.1.1 DMAC Organization in MC68SZ328

The DMAC in the DragonBall Super VZ provides full-featured DMA capabilities between both internal and external RAM and the modules shown in Figure 9-1.

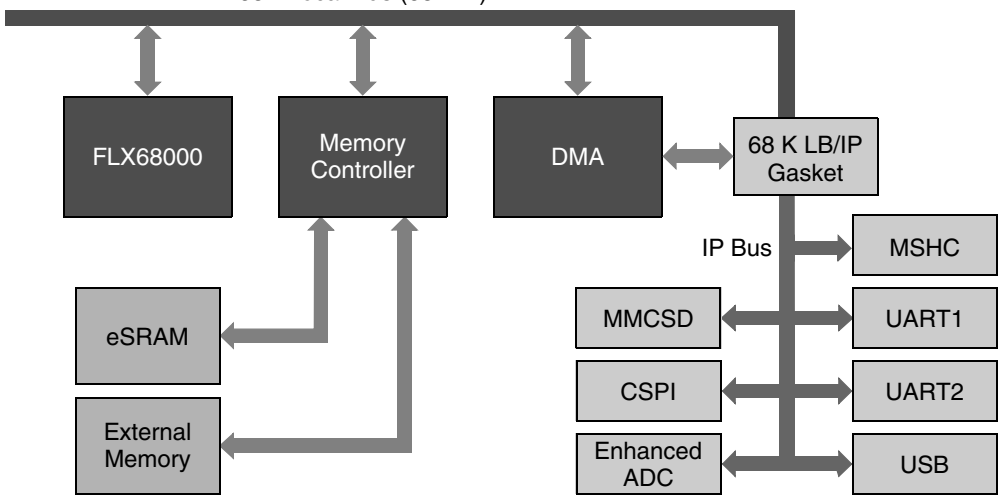


Figure 9-1. DMAC Relationship to Other MC68SZ328 Modules

The modules in the MC68SZ328 that support DMAC operations are shown in Table 9-1.

Table 9-1. Modules with DMA Capability

Module	DMA Capability
CSPI	Yes
UART 1 and 2	Yes
USB	Yes
MMCSD	Yes
ASP	Yes (Enhanced ADC)
I2C	No
PWM 1and 2	No
Timer 1 and 2	No
RTC	No
Bootstrap	No
MSHC	Yes

9.2 DMAC Operational Overview

The DMAC, as shown in the block diagram in Figure 9-2, contains the bus interface to the internal 68K local bus. It also offers a 64-byte data FIFO which is used as a buffer for holding the transfer data during a DMA burst cycle. There is also logic to generate handshaking signals with a USB device controller and logic to generate interrupts. The general system control and status registers control the overall operation of DMA controller. Individual channels have dedicated register sets for configuring their channel properties.

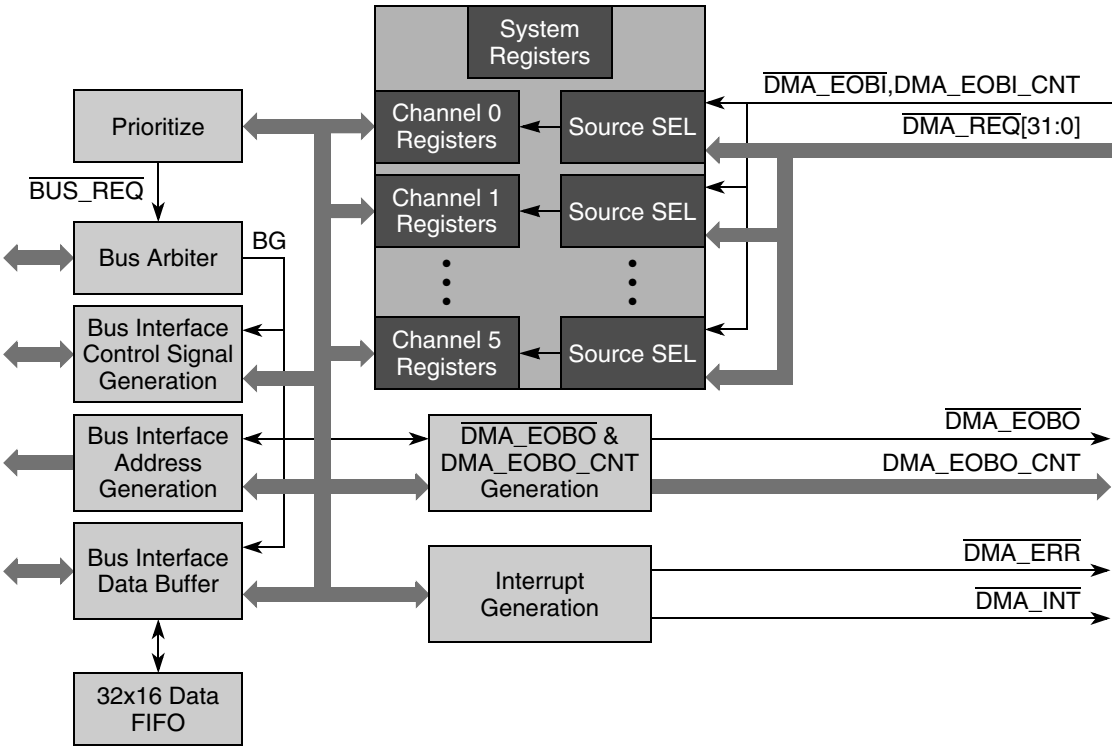


Figure 9-2. DMAC Simplified Block Diagram

9.2.1 DMAC Signal Description

The following seven signals are used to communicate between the DMAC, the other modules in the MC68SZ328, and external peripherals:

- **DMA_REQ—**The DMA request signal is generated by a peripheral. To initiate the DMA transfer, one FIFO generates one DMA_REQ signal. This signal must be negated by the peripheral automatically before data transfer is completed; otherwise, the signal will be treated as the next DMA request.
- **DMA_EOBI—**This signal is asserted by the USB module when the last data of the burst is read from the FIFO.
- **DMA_EOBI_CNT—**This signal is asserted by the USB module when the last data of the burst being read from the FIFO is 8-bit data.
- **DMA_EOBO—**This signal is asserted by the DMA controller when the last data of the burst being written to the FIFO.
- **DMA_EOBO_CNT—**This signal is asserted by the DMA controller when the last data of the burst is being written to the FIFO is 8-bit data.
- **DMA_ERR—**This signal is asserted by the DMA controller when any error is detected.

- **$\overline{\text{DMA\_INT}}$—**This signal is asserted by the DMA controller when data transfer is complete—that is, the data count reaches the desired level.

9.3 External DMA Request Pins

The MC68SZ328 provides two external DMA request pins ($\overline{\text{DMA\_REQ0}}$ and $\overline{\text{DMA\_REQ1}}$) to allow external devices to initiate data transfers using memory channel 0 and 1 respectively. The relationship of the DMA request timing is shown in Figure 9-3. The assertion of the $\overline{\text{DMA\_REQ0}}$ and $\overline{\text{DMA\_REQ1}}$ signal initiates a DMA transfer if the REN bit in memory channel control register is set. The number of data transfers is determined by the corresponding memory channel burst length register (MBLRx). While the negation of $\overline{\text{DMA\_REQ0}}$ and $\overline{\text{DMA\_REQ1}}$ does not affect the completion time of the DMA burst, the signal must be negated before the last data transfer is completed. If the signal is not negated, it will trigger another DMA burst.

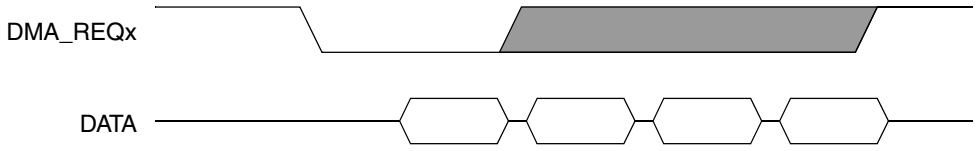


Figure 9-3. Example of External DMA Request Timing

9.3.1 DMA Request and Transfer

Figure 9-4 illustrates an example of DMA request and transfer. After receiving the DMA request from either of the $\overline{\text{DMA\_REQ0}}$ or $\overline{\text{DMA\_REQ1}}$ signals, the DMA controller sets up the transfer order and the appropriate I/O channel to transfer data according to the DMAC priority mechanism.

NOTE:

The maximum length for each read and write burst is 32 words.

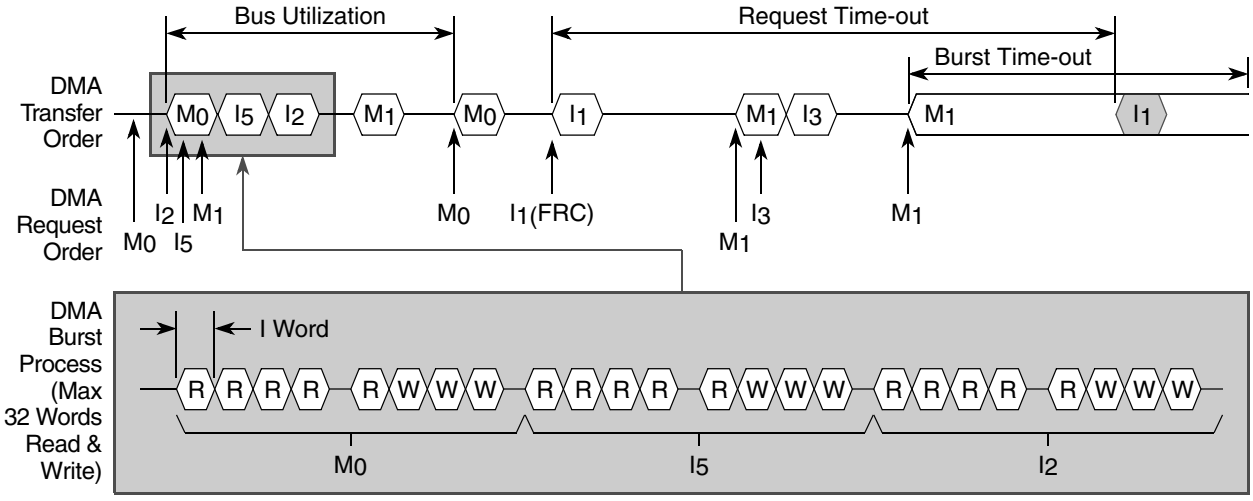


Figure 9-4. Example of DMAC Operation

9.4 Block Transfer Functions Using Memory Channels 0 and 1

The MC68SZ328 provides two memory channels that allow the transfer of blocks of data throughout memory. Memory channels provide discrete data block transfer functions using one of three modes of transfer:

- Discrete-to-discrete
- Discrete-to-continuous
- Continuous-to-discrete

The discrete-to-discrete mode allows the transfer of blocks of data with different separation from one space in memory to another. The separation lengths of the source and destination blocks can be programmed using the Source Block Separation Distance Register (page 9-17) and the Memory Channel Destination Block Separation Distance Register (page 9-19) respectively. The discrete-to-continuous and continuous-to-discrete modes provide transfer of blocks from discrete locations to a continuous space in memory and vice versa. The operation can be visualized in Figure 9-5.

These transfer modes can speed up some display functions such as: image block movement in the LCD display, window effect, retrieval of pre-store image patterns, and storage of the image block.

The block length and block-to-block separation are programmable. The block length is defined by the block length register. The block-to-block separation in all transfer modes is determined by the block separation distance register.

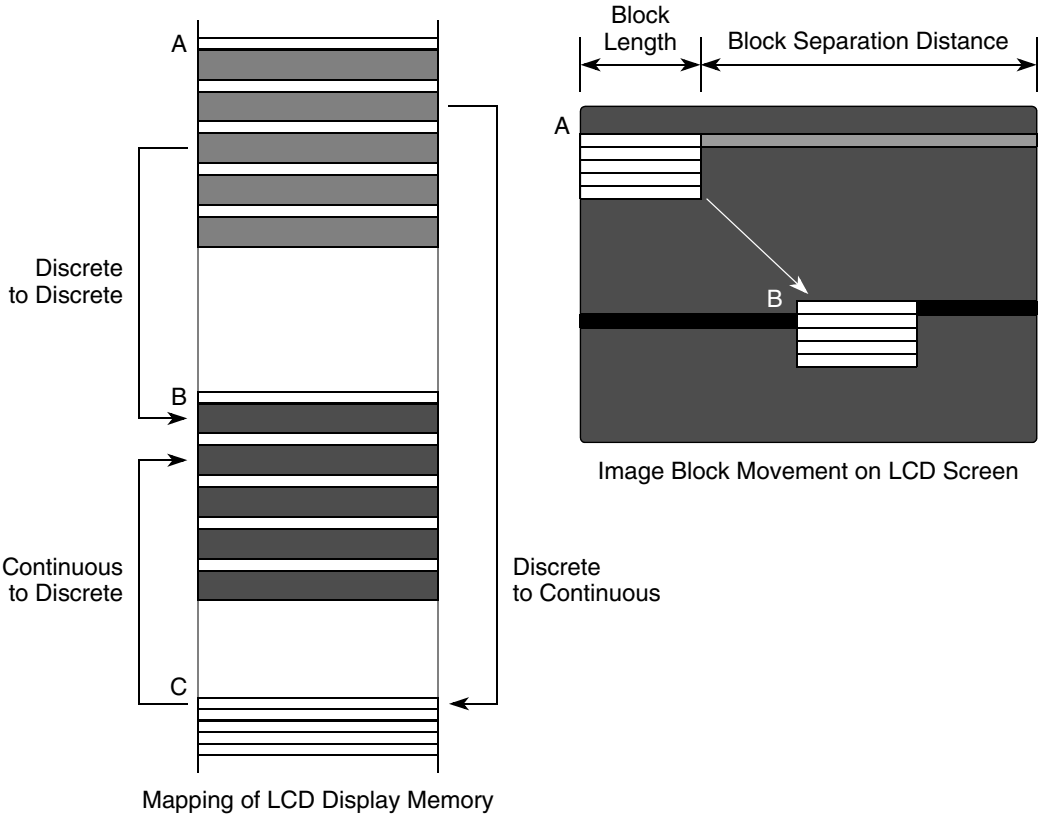


Figure 9-5. Block Transfer Functions Using Memory Channels

9.5 Programming Model

The DMAC registers are divided into three groups and each group is determined by function. The first group contains the general DMAC registers for all functional blocks and are discussed in Section 9.6, “DMAC General Registers,” on page 9-6. The second group of registers controls the memory channels and these are discussed in Section 9.7, “Memory Channel Registers,” on page 9-11. The third group controls the I/O channels and are discussed in Section 9.8, “I/O Channels (Channels 2–5) Registers,” on page 9-20.

9.6 DMAC General Registers

The DMAC general registers control most of the normal DMA functions. The registers including their bit assignments and descriptions are described in through .

9.6.1 DMA Control Register

This register controls the overall functionality of the DMAC. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 9-2.

DCR DMA Control Register **0x(FF)FE0000**

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

Table 9-2. DMA Control Register Description

Name	Description	Setting
Reserved Bits 15–2	Reserved	These bits are reserved and should be set to 0.
DRST Bit 1	**DMA Reset**—Writing a 1 to this bit generates a 3 cycle reset pulse to reset the entire DMAC, which brings the DMAC to the reset condition. Reading from this bit always gives a 0.	See description.
DEN Bit 0	**DMA Enable**—This bit enables/disables the DMA clock to the DMAC.	0 = Disable DMA. 1 = Enable DMA.

9.6.2 DMA Transfer Register

This register indicates both the transfer and error status of the memory and the I/O channels. The transfer status bits CH5–CH0 will be set whenever the corresponding DMA channel has completed data transfer. The error status bits, BTE and RTE, will be set when any bit is set in the burst time-out status register and the request time-out status register respectively.

If any one of the bits CH5–CH0 of the transfer status register is set and the corresponding bit in the interrupt mask register is clear, then $\overline{\text{DMA\_INT}}$ will be asserted to the system interrupt handler to indicate that there is interrupt generated from the corresponding channel.

Because the interrupt controller has a separate transfer complete interrupt (DMA1) and an error interrupt (DMA2), the interrupt service routine reads the DTSR register to verify which channel has completed the transfer. Similarly, it can also verify the type of errors by reading the BTE and RTE bits when DMA2 is set.

The CH5–CH0 bits can be cleared by writing a 1 to them. BTE and RTE are cleared by clearing all the status bits in burst time-out register and request time-out register respectively.

DTSR DMA Transfer Status Register **0x(FF)FE0002**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	BTE	RTE									CH5	CH4	CH3	CH2	CH1	CH0
TYPE	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 9-3. DMA Transfer Status Register Description

Name	Description	Setting
BTE Bit 15	**Burst Time-Out Error**—This bit is generated by logic “Or” of the status bits of the burst time-out error register. It indicates that there is a burst time-out error in the channels.	0 = No burst time-out error. 1 = At least one burst time-out error from the channels.
RTE Bit 14	**Request Time-Out Error**—This bit is generated by logic “Or” the status bits of request time-out error register. It indicates that there is a request time-out error in the channels.	0 = No request time-out error. 1 = At least one request time-out error from the channels.
Reserved Bits 13–6	Reserved	These bits are reserved and should be set to 0.
CH5–CH0 Bits 5–0	**Channel 5 to 0**—These bits show of the transfer status of memory and I/O DMA channels.	0 = Transfer not started or in progress. 1 = Transfer complete.

9.6.3 DMA Interrupt Mask Register

The DMA interrupt mask register (DIMR) is used to mask both the transfer complete interrupt and error interrupt (burst time-out or request time-out) that is generated when servicing the corresponding channel. There is one control bit for each channel. When an interrupt is masked, the interrupt controller will not generate an interrupt request to the interrupt handler, but its status can still be observed in the DMA transfer status register, DMA burst time-out status register, DMA request and the time-out status register.

At reset, all the interrupts are masked and all the bits in the DMA interrupt mask register are set to 1.

DIMR DMA Interrupt Mask Register **0x(FF)FE0004**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
											CH5	CH4	CH3	CH2	CH1	CH0
TYPE											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 9-4. DMA Interrupt Mask Register Description

Name	Description	Setting
Reserved Bits 15–6	Reserved	These bits are reserved and should be set to 0.
CH5–CH0 Bits 5–0	**Channel 5 to 0**—These bits control the interrupt function of all memory and I/O DMA channels. Setting the bit to '0' enables both transfer complete and error (burst time-out or request time-out) interrupts from the corresponding channel. Setting the bit to '1' masks all interrupts from the corresponding channel.	0 = Enable interrupt from the channel. 1 = Disable interrupt from the channel.

9.6.4 DMA Burst Time-Out Status Register

This register indicates which channel, if any, is being serviced and if a burst time-out is detected. Each bit can be cleared by writing a "1" to it. A burst time-out is set when a DMA burst cannot be completed in a pre-assigned number of clock cycles specified in the burst time-out count register of the channel. When any bit is set in this register and the corresponding bit in the interrupt mask register is clear, then $\overline{\text{DMA\_ERR}}$ will be asserted to the system interrupt handler.

DBTOSR DMA Burst Time-Out Status Register **0x(FF)FE0006**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
											CH5	CH4	CH3	CH2	CH1	CH0
TYPE											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 9-5. DMA Burst Time-Out Status Register Description

Name	Description	Setting
Reserved Bits 15–6	Reserved	These bits are reserved and should be set to 0.
CH5–CH0 Bits 5–0	**Channel 5 to 0**—These bits show the burst time-out status of memory and I/O DMA channels.	0 = No burst time-out. 1 = burst time-out.

9.6.5 DMA Request Time-Out Status Register

This register indicates which one of the enabled channels, if any, has detected a DMA request time-out. Each bit can be cleared by writing a "1" to it. A DMA request time-out is set when there is no DMA request from the selected DMA_REQ source within the pre-assigned number of clock cycles specified in the request time-out count register for the channel.

The sources for the DMA_REQ signal are memory channels that have been defined to use the external pins DMA_REQ0 and DMA_REQ1. Bits 0 and 1 of this register will indicate the time-out status of DMA_REQ0 and DMA_REQ1, if so configured, and bits 2–5 indicate the time-out status of the remaining four I/O channels.

When any bit in this register is set and the corresponding bit in the interrupt mask register is clear, then DMA_ERR will be asserted to the system interrupt handler.

DRTOSR DMA Request Time-Out Status Register **0x(FF)FE0008**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
											CH5	CH4	CH3	CH2	CH1	CH0
TYPE											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 9-6. DMA Request Time-Out Status Register Description

Name	Description	Setting
Reserved Bits 15–6	Reserved	These bits are reserved and should be set to 0.
CH5–CH0 Bits 5–0	**Channel 5 to 0**—These bits show the DMA request time-out status of memory and I/O DMA channels.	See description.

9.6.6 DMA Burst Time-Out Control Register

This register sets time-out for the DMA transfer cycle for all DMA channels, so that it can release the 68K bus on an error condition. The count value in this register is latched into an internal counter when a DMA cycle starts. The counter counts down until it is stopped when the DMA cycle is completed, or upon a time-out; then, it asserts an interrupt and sets the corresponding error bit in the DMA burst time-out error register. The DMA clock is used as an input clock to the counter.

DBTOCR DMA Burst Time-Out Status Register **0x(FF)FE000E**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 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 9-7. DMA Burst Time-Out Control Register Description

Name	Description	Setting
EN Bit 15	**Enable**—Burst time-out enable.	0 = Disable burst time-out. 1 = Enable burst time-out.
CNT Bits 14–0	**Count**—This field holds the burst time-out count down value. The programmed count will be a number of DMA clock cycles.	See description.

9.7 Memory Channel Registers

Channels 0 and 1 are the only memory channels which support memory-to-memory data transfers. A memory-to-memory DMA transfer includes a burst read from source memory and then a burst write to the destination memory. A 32 × 16 data buffer in the DMA data buffer and flow control block is used for temporary data storage between burst read and burst write. Therefore, the burst length for the memory channel is a maximum of 64 bytes or 32 words (16-bit). The memory channels have the lowest priority among all the DMA channels. The unique set of memory channel registers including their bit assignments and descriptions are discussed in – .

The memory channels generate a transfer complete or an error interrupt to the interrupt handler as a result of the following conditions:

1. When data count reaches the desired value
2. When a DMA burst time-out is true during a burst cycle
3. When an external DMA request time-out is true

9.7.1 Memory Channel Source Address Register

This register contains the source memory address for the first DMA cycle. The value of this register remains unchanged throughout the DMA process.

MSAR0 Memory Channel Source Address Register **0x(FF)FE0040**
MSAR1 **0x(FF)FE0080**

	BIT 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	BIT 16
	MSA															
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	BIT 0
	MSA															
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 9-8. Memory Channel Source Address Register Description

Name	Description	Setting
MSA Bits 31–0	**Memory Source Address**—These bits hold the system memory address from where data will be read. **Note:** MSA[0] will always read/write as 0 to ensure the correct word-alignment of addresses.	See description.

9.7.2 Memory Channel Destination Address Register

This register contains the destination memory address for the first DMA cycle. The value of this register remains unchanged throughout the DMA process.

MDAR0 Memory Channel Destination Address Register **0x(FF)FE0044**
MDAR1 **0x(FF)FE0084**

	BIT 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	BIT 16
	MDA															
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	BIT 0
	MDA															
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 9-9. Memory Channel Destination Address Register Description

Name	Description	Setting
MDA Bits 31–0	**Memory Destination Address**—These bits hold the system memory address to where data will be written. **Note:** MDA[0] will always read/write as 0 to ensure the word-align address.	See description.

9.7.3 Memory Channel Count Register

This register contains the number of bytes of data to be transferred. There is an internal counter which will up count for every DMA transfer—that is, number of bytes minus 2 for a word and minus 1 for a byte. The internal counter is then compared with the register after every transfer. When the counter value matches the register value, the channel will be disabled until the MEN bit in the memory channel control register has been cleared and set again, or the RPT bit in the memory channel control register is set to 1. The internal counter will be reset to zero when the channel is enabled again. The length of the last DMA burst can be shorter than a regular burst length specified in the memory channel burst length register.

NOTE:

The value of the memory channel count register must be a multiple of the source memory or destination memory port size, whichever is larger. If either the source memory or the destination memory port size is 16-bit, then bit 0 of MCNT will set to 0 before comparing with the internal counter.

MCNTR0
MCNTR1

Memory Channel Count Register

0x(FF)FE0048
0x(FF)FE0088

	BIT 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	BIT 16
									MCNT							
TYPE									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	BIT 0
	MCNT															
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 9-10. Memory Channel Count Register Description

Name	Description	Setting
Reserved Bits 31–24	Reserved	These bits are reserved and should be set to 0.
MCNT Bits 23–0	**Memory Count**—This field holds the number of bytes of data to be transferred.	See description.

9.7.4 Memory Channel Control Register

This register controls the DMAC memory channel operation.

MCR0
MCR1

Memory Channel Control Register

0x(FF)FE004C
0x(FF)FE008C

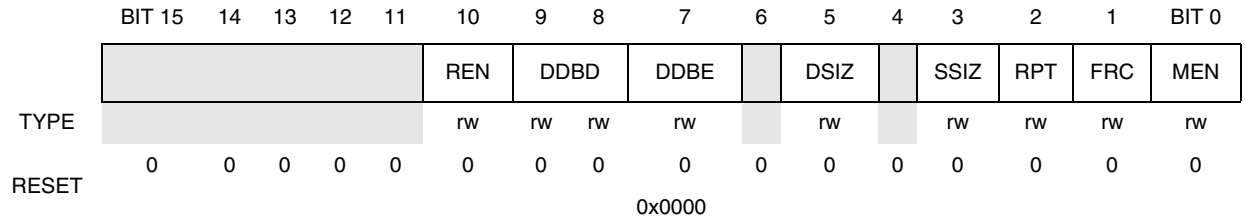


Table 9-11. Memory Channel Control Register Description

Name	Description	Setting
Reserved Bits 15–11	Reserved	These bits are reserved and should be set to 0.
REN Bit 10	**Request Enable**—This bit enables/masks the external DMA request signal ($\overline{\text{DMA\_REQ0}}$ for channel 0, $\overline{\text{DMA\_REQ1}}$ for channel 1). When this bit is set, the DMA transfer is initialized when MEN is set and external DMA request signal is asserted. When this bit is clear, the DMA transfer is initialized only when MEN is asserted.	0 = Masks external DMA request signal, DMA transfer is initialized when MEN is asserted only. 1 = Enable external DMA request signal. A DMA request asserted by a peripheral, with MEN asserted, will trigger a DMA transfer.
DDBD Bits 9–8	**Discrete Data Block Transfer Direction**—These bits control the discrete data block transfer direction.	00 = Discrete-to-discrete 01 = Discrete-to-continuous 10 = Continuous-to-discrete 11 = Reserved
DDBE Bit 7	**Discrete Data Block Transfer Enable**—This bit enables/disables the discrete data block transfer function.	0 = Disable. 1 = Enable.
Reserved Bit 6	Reserved	This bit is reserved and should be set to 0.
DSIZ Bit 5	**Memory Destination Bus Size**—This bit controls the data bus size of the destination memory port size. **Note:** The memory size should not be programmed to 8-bit when using SDRAM.	0 = 16-bit memory bus. 1 = 8-bit memory bus.
Reserved Bit 4	Reserved	This bit is reserved and should be set to 0.
SSIZ Bit 3	**Memory Source Bus Size**—This bit controls the data bus size of the source memory port size. **Note:** The memory size should not be programmed to 8-bit when using SDRAM.	0 = 16-bit memory bus. 1 = 8-bit memory bus.

Table 9-11. Memory Channel Control Register Description (Continued)

Name	Description	Setting
RPT Bit 2	**Repeat**—This bit enable/disable bulk data transfer repeat function. If enable, the Count Register is reset to its original value at the time the channel is enabled after its count value reaches $0. Data transfer will then be carried out endlessly.	0 = Disable repeat function. 1 = Enable repeat function.
FRC Bit 1	**Force DMA Cycle**—Writing a 1 to this bit will force a DMA request, regardless of bus utilization control. Reading to this bit will always give 0.	1 = Force DMA request.
MEN Bit 0	**Memory Channel Enable**—This bit enable/disable DMA channel.	0 = Disable DMA channel. 1 = Enable DMA channel.

9.7.5 Memory Channel Burst Length Register

This register controls the burst length of a DMA cycle. The burst length of the last DMA cycle will be equal to whatever remains in the counter and if interrupt is enabled, MCBLR asserts an interrupt-to-interrupt handler after the last data transfer. The burst length should be selected such that the interrupt latency caused by the DMA burst is acceptable by all other interrupting devices and bus masters.

Because the data buffer size is 32 × 16 in the DMA data buffer and control block, the maximum burst length (MBL) is limited by this buffer size of 64 bytes. For example, if MBL = 64, for a data transfer from a 16-bit port to a 16-bit port the MBL will be a 32-word (16-bit) read burst followed by a 32-word (16-bit) write burst. For a data port transfer from an 8-bit port to a 16-bit port the MBL will be a 64-byte (8-bit) read burst followed by a 32-word (16-bit) write burst.

NOTE:

Memory Channel Burst Length Requirement—The value of the memory channel burst length register must be a multiple of the source memory or destination memory port size, whichever is larger. If either source memory or destination memory port size is 16-bit, then bit 0 of the MBL will set to 0 before comparing with the internal counter.

MBLR0
MBLR1

Memory Channel Burst Length Register

0x(FF)FE004E
0x(FF)FE008E

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
											MBL					
TYPE											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 9-12. Memory Channel Burst Length Register Description

Name	Description	Setting
Reserved Bits 15–6	Reserved	These bits are reserved and should be set to 0.
MBL Bits 5–0	**Memory Burst Length**—These bits hold the number of bytes that will be transferred in a DMA burst. **Requirement:** • *DDBE in MCR Disabled* (discrete block transfer function is not used): If the burst length value is programmed to be larger than the count value, then the DMAC uses the count value as the burst length. • *DDBE in MCR Enabled* (discrete block transfer function is used): If the burst length value is programmed to be larger than block length value, the DMAC uses the block length value as the burst length.	000001–1 byte read followed by 1 byte write … 100000–32 bytes read followed by 32 bytes write 100001–33 bytes read followed by 33 bytes write … 111111—63 bytes read followed by 63 bytes write 000000—64 bytes read followed by 64 bytes write

9.7.6 Memory Channel Bus Utilization Control Register

This register controls the bus utilization of the memory channel. The DMAC will not request the system bus until the counter reaches zero. The count value is latched and starts to count down immediately after the memory channel DMA burst is completed.

MBUCR0
MBUCR1

Memory Channel Bus Utilization Control Register

0x(FF)FE0050
0x(FF)FE0090

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	CCNT															
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 9-13. Memory Channel Bus Utilization Control Register Description

Name	Description	Setting
CCNT Bits 15–0	**Clock Count**—This field contains the number of DMA clocks where the memory channel will release to the 68K bus before the next DMA request for the channel.	See description.

9.7.7 Block Length Register

This register is used in discrete data block transfer only. It defines the block length.

BLR0
BLR1

Block Length Register

0x(FF)FE0052
0x(FF)FE0092

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	BL															
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 9-14. Block Length Register Description

Name	Description	Setting
BL Bits 15–0	**Block Length**—This field holds the block length in bytes. **Requirement:** The value of the memory channel block length register must be a multiple of the source memory or destination memory port size, whichever is larger. If either source memory or destination memory port size is 16-bit, then bit 0 of BLR will be set to 0 before comparing with the internal counter.	See description.

9.7.8 Source Block Separation Distance Register

This register is only used during discrete block transfers. The source block separation distance register (SBSDR) determines the number of bytes skipped between two discrete data blocks in the transfer.

SBSDR0
SBSDR1

Source Block Separation Distance Register

0x(FF)FE0054
0x(FF)FE0094

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	SBSD															
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 9-15. Source Block Separation Distance Register Description

Name	Description	Setting
SBSD Bits 15–0	**Source Block Separation Distance**—This field contains the number of bytes of data to be skipped while reading discrete blocks in discrete block transfer. **Requirement:** The value of this source block separation distance register must be the multiple of the source memory or destination memory port size, whichever is larger. If either source memory or destination memory port size is 16-bit, then bit 0 of SBSDR will set to 0 before comparing the internal counter.	See description.

9.7.9 Memory Channel External DMA Request Time-Out Register

This register sets time-out for the DMA request from the selected external DMA request lines ($\overline{\text{DMA\_REQ0}}$ or $\overline{\text{DMA\_REQ1}}$). It is used to detect the discontinuity of data transfer from an external I/O device. The count value in this register is latched to an internal counter and starts to count when a DMA channel is enabled, and is reset to the original value when a DMA request is detected. Upon time-out, it asserts an interrupt and sets the error bit in the DMA request time-out status register. The input clock is selectable from either the DMA clock or the 32 kHz clock.

MRTOR0
MRTOR1
Memory Channel DMA Request Time-Out Register
0x(FF)FE0056
0x(FF)FE0096

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 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 9-16. Memory Channel DMA Request Time-Out Register Description

Name	Description	Setting
EN Bit 15	**Enable**—This bit enable/disable DMA request time-out.	0 = Disable DMA request time-out. 1 = Enable DMA request time-out.
CLK Bit 14	**Clock Source**—This bit selects the counter of input clock source.	0 = DMACLK 1 = 32.768KHz
PSC Bit 13	**Prescaler Count**—Prescaler of input clock	0 = Divide by 1 1 = Divide by 256
CNT Bits 12–0	**Request Time-Out Count**—These bits hold the time-out count down value for the internal counter. This value remains unchanged throughout the DMA process.	See description.

9.7.10 Destination Block Separation Distance Register

This register is used in discrete block transfer only. It determines the number of bytes skipped between two discrete data blocks when writing the destination block.

DBSDR0 Destination Block Separation Distance Register **0x(FF)FE0058**
DBSDR1 **0x(FF)FE0098**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	DBSD															
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 9-17. Destination Block Separation Distance Register Description

Name	Description	Setting
DBSD Bits 15–0	**Destination Block Separation Distance**—These bits hold the number of bytes of data to be skipped during writing discrete blocks during Discrete Block Transfers. **Requirement:** The value of the destination block separation distance register must be a multiple of the source memory or destination memory port size, whichever is larger. If either source memory or destination memory port size is 16-bit, then bit 0 of SBSDR will set to 0 before comparing the internal counter.	See description.

9.8 I/O Channels (Channels 2–5) Registers

Channels 2–5 support I/O-to-memory and memory-to-I/O data transfer. All channels have an identical set of registers that are discussed with their bit assignments and descriptions in I/O Channel Memory Address Register on page 20 and I/O Channel DMA Request Time-Out Register on page 27.

The I/O channels generate transfer complete or error interrupts to the interrupt handler during the following conditions:

1. When data count reaches the desired value and both IDIR and EOBE bits are not set—only during a burst cycle.
2. When DMA request time-out is true.
3. When DMA burst time-out is true during a burst cycle.

9.8.1 I/O Channel Memory Address Register

This register contains the memory address for the first DMA cycle. The value of this register remains unchanged throughout the DMA process.

IMAR2 **0x(FF)FE00C0**
IMAR3 I/O Channel Memory Address Registers **0x(FF)FE0100**
IMAR4 **0x(FF)FE0140**
IMAR5 **0x(FF)FE0180**

	BIT 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	BIT 16
	IMA															
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	BIT 0
	IMA															
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 9-18. I/O Channel Memory Address Register Description

Name	Description	Setting
IMA Bits 31–0	**I/O Channel Memory Address**—This field holds the system memory address where the data will be written to or read from. **Note:** IMA[0] will always read or write as 0 to ensure a word-align address.	See description.

9.8.2 I/O Channel Peripheral Address Register

This register contains the peripheral address for the DMA cycle. The value of this register remains unchanged throughout the DMA process.

IPAR2 **0x(FF)FE00C4**
IPAR3 I/O Channel Peripheral Address Registers **0x(FF)FE0104**
IPAR4 **0x(FF)FE0144**
IPAR5 **0x(FF)FE0184**

	BIT 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	BIT 16
	IPA															
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	BIT 0
	IPA															
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 9-19. I/O Channel Peripheral Address Register Description

Name	Description	Setting
IPA Bits 31–0	**I/O Channel Memory Address**—This field holds the peripheral address where data will be written to or read from.	See description.

9.8.3 I/O Channel Count Register

This register contains the number of bytes of data to be transferred. There is an internal counter which will up count for every DMA transfer—that is, number of bytes minus 2 for a word and minus 1 for a byte. The internal counter is compared with the register after every transfer. When the counter value matches the register value, the channel is disabled until the CEN bit in the I/O channel control register is cleared and set again, or the RPT bit in the I/O channel control register is set to 1. The internal counter is reset to zero when the channel is enabled again.

The length of the last DMA burst can be shorter than a regular burst length specified in the I/O channel burst length register. However, if data is transferred out from an I/O FIFO and the last burst is less than MBL, the I/O device has to be capable of generating a DMA request for the last transfer. If data is transferred to an I/O FIFO and the last burst is less than MBL, only the remaining count of data is transferred.

When both EOBE and IDIR bit of I/O channel control register are set, this register becomes a read only register and the value of the register will be the number of bytes being transferred.

NOTE:

I/O Channel Count Value Requirement—The value of the I/O channel count register must be a multiple of the peripheral FIFO or memory port size, whichever is larger. If either peripheral FIFO or memory port size is 16-bit, then bit 0 of ICNT will set to 0 before comparing with the internal counter.

ICNTR2 **0x(FF)FE00C8**
ICNTR3 **0x(FF)FE0108**
ICNTR4 **0x(FF)FE0148**
ICNTR5 **0x(FF)FE0188**

I/O Channel Count Registers

	BIT 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	BIT 16
									ICNT							
TYPE									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	BIT 0
	ICNT															
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 9-20. I/O Channel Count Register Description

Name	Description	Setting
Reserved Bits 31–24	Reserved	These bits are reserved and should be set to 0.

Table 9-20. I/O Channel Count Register Description (Continued)

Name	Description	Setting
ICNT Bits 23–0	**I/O Count**—This field contains the number of bytes of data to be transferred.	See description.

9.8.4 I/O Channel Control Register

This register controls and reflects the status of DMAC channel operation.

ICR2 **0x(FF)FE00CC**
ICR3 **0x(FF)FE010C**
ICR4 **0x(FF)FE014C**
ICR5 **0x(FF)FE018C**

I/O Channel Control Register

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
					FLYBY	DIR	REN	EOBE		IFSIZ			MSIZ	RPT	FRC	CEN
TYPE					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 9-21. I/O Channel Control Register Description

Name	Description	Setting
Reserved Bits 15–12	Reserved	These bits are reserved and should be set to 0.
FLYBY Bit 11	**Flyby**—This bit enables/disables a feature by which the destination party can get the data from the data bus in the time that the source party drives the data out. When it is disabled, a DMA burst is accomplished by executing two transfers: source party -> DMA FIFO and DMA FIFO -> destination party. **Note:** This feature does not support byte transfer when the memory is a 16-bit port device. The flyby feature is not supported by the USB device.	0 = Disable Flyby. 1 = Enable Flyby.
DIR Bit 10	**Direction**—This bit controls the data move direction.	0 = Memory to peripheral. 1 = Peripheral-to-memory.
REN Bit 9	**Request Enable**—This bit enables/masks the DMA request signal. When this bit is set, the DMA burst will be initiated by the $\overline{\text{DMA\_REQ}}$ signal from the I/O FIFO. When this bit is clear, no DMA transfer will be issued unless FRC bit is set by software.	0 = Masks DMA request signal, asserting DMA request by the peripheral will not trigger a DMA transfer. 1 = Enables DMA request signal, asserting DMA request by the peripheral will trigger a DMA transfer.

Table 9-21. I/O Channel Control Register Description (Continued)

Name	Description	Setting
EOBE Bit 8	**End of Burst Enable**—This bit enable/disable end-of-burst operation. **Note:** When this bit is set, the end-of-burst operation is enabled. This bit needs to work together with the direction bit DIR (bit-10).	0 = Disable end-of-burst operation. 1 = Enable end-of-burst operation.
Reserved Bit 7	Reserved	This bit is reserved and should be set to 0.
IFSIZ Bit 6	**I/O FIFO Size**—This bit selects the I/O FIFO size. **Note:** IFSIZ will always read/write as 0 if the EOBE bit is set, because end-of-burst operation will only work for 16-bit FIFO. **Note:** When using the flyby function (FLYBY=1) and 8-bit FIFO size (IFSIZ=1), memory port size (MSIZ) should use 8-bit memory bus.	0 = 16-bit I/O FIFO. 1 = 8-bit I/O FIFO.
Reserved Bits 5–4	Reserved	These bits are reserved and should be set to 0.
MSIZ Bit 3	**Memory Port Size**—This bit selects the memory data port size. **Note:** The memory size should not be programmed to 8-bit when using SDRAM.	0 = 16-bit memory bus. 1 = 8-bit memory bus.
RPT Bit 2	**Repeat**—This bit enables/disables the data transfer repeat function. If this bit is enabled when the data count reaches 0, the count register is reset to its original value (at the time the channel is enabled). The memory address generator is reloaded from the memory address register for the next DMA burst. Data transfer will then be carried out repeatedly to and from a memory buffer.	0 = Disable repeat function. 1 = Enable repeat function.
FRC Bit 1	**Force DMA Cycle**—Writing a 1 to this bit will force a DMA cycle. Reading to this bit will always provide 0. This function is normally used for debugging and when REN=0.	1 = Force DMA request.
CEN Bit 0	**DMA Channel Enable**—This bit enables/disables DMA channel.	0 = Disable DMA channel. 1 = Enable DMA channel.

When the direction (DIR) bit is set, which is a peripheral-to-memory transfer operation, the burst length is determined by the input signals $\overline{\text{DMA\_EOBI}}$ and DMA_EOBI_CNT. The DMA burst (from peripheral-to-memory) can only be terminated by disabling the channel (clear the CEN bit in I/O channel control register). The I/O channel count register (ICNTR2–ICNTR5) will become read only and will indicate the number of bytes being transferred. This setting is generally used when the I/O channel is configured to transfer data from an endpoint FIFO of the USB device to an endpoint data packet buffer in system memory.

When the direction bit is clear, which is a memory to peripheral transfer operation, the channel will work like a normal operation, the only difference is that, at the end of each burst, the DMA controller will generate the $\overline{\text{DMA\_EOBO}}$ and DMA_EOBO_CNT signals to the peripheral. This setting is generally used when the I/O channel is configured to transfer data from an endpoint data packet buffer in system memory to an endpoint FIFO of USB device

Table 9-22. DMA_EOBI_CNT and DMA_EOBO_CNT Status

DMA_EOBI_CNT or DMA_EOBO_CNT	Last Data
0	8-bit
1	16-bit

9.8.5 I/O Channel Request Source Select Register

This 8-bit register selects one of the 32 DMA request signals ($\overline{\text{DMA\_REQ}}$[31:00]) which will initiate a DMA transfer for the channel.

IRSSR2 **0x(FF)FE00CE**
IRSSR3 I/O Channel Request Source Select Register **0x(FF)FE010E**
IRSSR4 **0x(FF)FE014E**
IRSSR5 **0x(FF)FE018E**

Table 9-23. I/O Channel Request Source Select Register Description

Name	Description	Setting
Reserved Bits 15–5	Reserved	These bits are reserved and should be set to 0.
IRSS Bits 4–0	**I/O Request Source Select**—These bits select one of the 32 $\overline{\text{DMA\_REQ}}$ signals which will initiate the DMA transfer cycle for this channel.	00000-select $\overline{\text{DMA\_REQ}}$[0] … 11111-select $\overline{\text{DMA\_REQ}}$[31]

9.8.6 I/O Channel Burst Length Register

This register controls the burst length of a DMA cycle. Unlike the memory channel, the I/O channel burst length is normally assigned by the FIFO size of the selected I/O device, or at the FIFO level its $\overline{\text{DMA\_REQ}}$ signal asserts.

For example, if the UART RxD FIFO is 12 × 8 and it asserts $\overline{\text{DMA\_REQ}}$ when it receives more than 8 bytes of data, then IBL in this case is 8. If the memory port size is also 8-bit, then the DMA burst will be 8 times the byte read followed by 8 times the byte write transfer cycle.

If memory port size is smaller than the I/O port size—for example, the I/O port is 16-bit, memory port is 8-bit, and the burst length is set to 16—then the DMA will perform a 16-word burst read and a 32-byte burst write for the I/O-to-memory transfer.

NOTE:

I/O Channel Burst Length Requirement—The value of the I/O channel Burst Length Register *must be* the multiple of the peripheral FIFO or memory port size, whichever is larger. If either the peripheral FIFO or the memory port size is 16-bit, then bit 0 of IBL will set to 0 before comparing with the internal counter.

IBLR2 **0x(FF)FE00D0**
IBLR3 **0x(FF)FE0110**
IBLR4 **0x(FF)FE0150**
IBLR5 **0x(FF)FE0190**

I/O Channel Burst Length Register

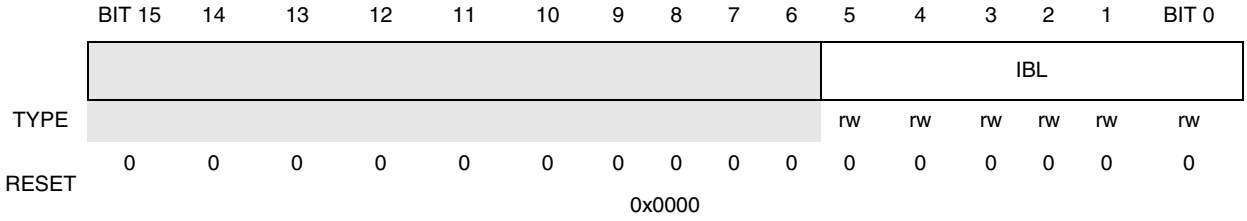


Table 9-24. I/O Channel Burst Length Register Description

Name	Description	Setting
Reserved Bits 15–6	Reserved	These bits are reserved and should be set to 0.
IBL Bits 5–0	**I/O Channel Burst Length**—These bits hold the number of bytes that will be transferred in a DMA burst. IBL is normally equal to the depth of the target FIFO. **Note:** If the user programs the burst length larger than the count value, the DMAC uses the count value as the burst length.	000001-1 byte read followed by 1 byte write … 100000-32 bytes read followed by 32 bytes write 100001-33 bytes read followed by 33 bytes write … 111111-63 bytes read followed by 63 bytes write 000000-64 bytes read followed by 64 bytes write

9.8.7 I/O Channel DMA Request Time-Out Register

This register sets time-out for the DMA request from the selected I/O data FIFO. It is used to detect the discontinuity of I/O data transfer from an external I/O device. The count value in this register is latched to an internal counter and starts to count when a DMA channel is enabled, and then is reset to the original value when a DMA request is detected. Upon time-out, it asserts an interrupt and sets the error bit in the DMA request time-out status register. The input clock is selectable from either the DMA clock or the 32.768 kHz clock.

IRTOR2 **0x(FF)FE00D2**
IRTOR3 I/O Channel DMA Request Time-Out Register **0x(FF)FE0112**
IRTOR4 **0x(FF)FE0152**
IRTOR5 **0x(FF)FE0192**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 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 9-25. I/O Channel DMA Request Time-Out Register Description

Name	Description	Setting
EN Bit 15	**Enable**—This bit enable/disable DMA request time-out.	0 = Disable DMA request time-out. 1 = Enable DMA request time-out.
CLK Bit 14	**Clock Source**—This bit selects the counter of input clock source.	0 = DMACLK 1 = 32.768 kHz clock
PSC Bit 13	**Prescaler Count**—This bit is the prescaler of input clock.	0 = divide by 1 1 = divide by 256
CNT Bits 12–0	**Request Time-Out Count**—These bits hold the time-out count down value for the internal counter. This value remains unchanged throughout the DMA process.	See description.

9.9 DMAC Register Addresses

The base address for DMAC is 0xFFFE0000. A summary of all registers are shown in Table 9-26.

Table 9-26. DMAC Register Map

Address	Name	Width	Description
0xFFFE0000	DCR	16	DMA Control Register
0xFFFE0002	DTSR	16	DMA Transfer status Register
0xFFFE0004	DIMR	16	DMA Interrupt Mask Register
0xFFFE0006	DBTOSR	16	DMA Burst Time-Out Status Register
0xFFFE0008	DRTOSR	16	DMA Request Time-Out Status Register
0xFFFE000A	–	–	Reserved
0xFFFE000C	–	–	Reserved
0xFFFE000E	DBTOCR	16	DMA Burst Time-Out Control Register
0xFFFE0010–03F	–	–	Reserved
0xFFFE0040	MSAR0	32	Memory Channel 0 Source Address Register
0xFFFE0044	MDAR0	32	Memory Channel 0 Destination Address Register
0xFFFE0048	MCNTR0	32	Memory Channel 0 Count Register
0xFFFE004C	MCR0	16	Memory Channel 0 Control Register
0xFFFE004E	MBLR0	16	Memory Channel 0 Burst Length Register
0xFFFE0050	MBUCR0	16	Memory Channel 0 Bus Utilization Control Register
0xFFFE0052	BLR0	16	Memory Channel 0 Block Length Register
0xFFFE0054	SBSDR0	16	Memory Channel 0 Source Block Separation Distance Register
0xFFFE0056	MRTOR0	16	Memory Channel 0 External DMA Request Time-Out Register
0xFFFE0058	DBSDR0	16	Memory Channel 0 Destination Block Separation Distance Register
0xFFFE005A–07F	–	–	Reserved
0xFFFE0080	MSAR1	32	Memory Channel 1 Source Address Register
0xFFFE0084	MDAR1	32	Memory Channel 1 Destination Address Register
0xFFFE0088	MCNTR1	32	Memory Channel 1 Count Register
0xFFFE008C	MCR1	16	Memory Channel 1 Control Register
0xFFFE008E	MBLR1	16	Memory Channel 1 Burst Length Register
0xFFFE0090	MBUCR1	16	Memory Channel 1 Bus Utilization Control Register
0xFFFE0092	BLR1	16	Memory Channel 1 Block Length Register

Table 9-26. DMAC Register Map (Continued)

Address	Name	Width	Description
0xFFFE0094	SBSDR1	16	Memory Channel 1 Source Block Separation Distance Register
0xFFFE0096	MRTOR1	16	Memory Channel 1 External DMA Request Time-Out Register
0xFFFE0098	DBSDR1	16	Memory Channel 1 Destination Block Separation Distance Register
0xFFFE009A–0BF	–	–	Reserved
0xFFFE00C0	IMAR2	32	I/O Channel 2 Memory Address Register
0xFFFE00C4	IPAR2	32	I/O Channel 2 Peripheral Address Register
0xFFFE00C8	ICNTR2	32	I/O Channel 2 Count Register
0xFFFE00CC	ICR2	16	I/O Channel 2 Control Register
0xFFFE00CE	IRSSR2	16	I/O Channel 2 Request Source Select Register
0xFFFE00D0	IBLR2	16	I/O Channel 2 Burst Length Register
0xFFFE00D2	IRTOR2	16	I/O Channel 2 DMA Request Time-Out Register
0xFFFE00D4–0FF	–	–	Reserved
0xFFFE0100	IMAR3	32	I/O Channel 3 Memory Address Register
0xFFFE0104	IPAR3	32	I/O Channel 3 Peripheral Address Register
0xFFFE0108	ICNTR3	32	I/O Channel 3 Count Register
0xFFFE010C	ICR3	16	I/O Channel 3 Control Register
0xFFFE010E	IRSSR3	16	I/O Channel 3 Request Source Select Register
0xFFFE0110	IBLR3	16	I/O Channel 3 Burst Length Register
0xFFFE0112	IRTOR3	16	I/O Channel 3 DMA Request Time-Out Register
0xFFFE0114–13F	–	–	Reserved
0xFFFE0140	IMAR4	32	I/O Channel 4 Memory Address Register
0xFFFE0144	IPAR4	32	I/O Channel 4 Peripheral Address Register
0xFFFE0148	ICNTR4	32	I/O Channel 4 Count Register
0xFFFE014C	ICR4	16	I/O Channel 4 Control Register
0xFFFE014E	IRSSR4	16	I/O Channel 4 Request Source Select Register
0xFFFE0150	IBLR4	16	I/O Channel 4 Burst Length Register
0xFFFE0152	IRTOR4	16	I/O Channel 4 DMA Request Time-Out Register
0xFFFE0154–17F	–	–	Reserved
0xFFFE0180	IMAR5	32	I/O Channel 5 Memory Address Register

Table 9-26. DMAC Register Map (Continued)

Address	Name	Width	Description
0xFFFE0184	IPAR5	32	I/O Channel 5 Peripheral Address Register
0xFFFE0188	ICNTR5	32	I/O Channel 5 Count Register
0xFFFE018C	ICR5	16	I/O Channel 5 Control Register
0xFFFE018E	IRSSR5	16	I/O Channel 5 Request Source Select Register
0xFFFE0190	IBLR5	16	I/O Channel 5 Burst Length Register
0xFFFE0192	IRTOR5	16	I/O Channel 5 DMA Request Time-Out Register
0xFFFE0194–1FF	–	–	Reserved

9.10 DMA Request Table

A summary of all DMA requests are shown in Table 9-27.

Table 9-27. DMA Request Table

DMA Request	Peripheral
$\overline{\text{DMA\_REQ}}$[31]	UART 1 Receive DMA Request
$\overline{\text{DMA\_REQ}}$[30]	UART 1 Transmit DMA Request
$\overline{\text{DMA\_REQ}}$[29]	UART 2 Receive DMA Request
$\overline{\text{DMA\_REQ}}$[28]	UART 2 Transmit DMA Request
$\overline{\text{DMA\_REQ}}$[27]	USB Device End Point 2 DMA Request
$\overline{\text{DMA\_REQ}}$[26]	USB Device End Point 1 DMA Request
$\overline{\text{DMA\_REQ}}$[25]	USB Device End Point 0 DMA Request
$\overline{\text{DMA\_REQ}}$[24]	ASP ADC DMA Request
$\overline{\text{DMA\_REQ}}$[23]	CSPI Transmit DMA Request
$\overline{\text{DMA\_REQ}}$[22]	CSPI Receive DMA Request
$\overline{\text{DMA\_REQ}}$[21]	MMCSD DMA Request
$\overline{\text{DMA\_REQ}}$[20]	Reserved
$\overline{\text{DMA\_REQ}}$[19]	External DMA Request ($\overline{\text{DMA\_REQ0}}$)
$\overline{\text{DMA\_REQ}}$[18]	External DMA Request ($\overline{\text{DMA\_REQ1}}$)
$\overline{\text{DMA\_REQ}}$[17]	USB Device End Point 4 DMA Request
$\overline{\text{DMA\_REQ}}$[16]	USB Device End Point 3 DMA Request
$\overline{\text{DMA\_REQ}}$[15]	MSHC DMA Request

Table 9-27. DMA Request Table (Continued)

DMA Request	Peripheral
$\overline{\text{DMA\_REQ}}$[14]	Reserved
$\overline{\text{DMA\_REQ}}$[13]	Reserved
$\overline{\text{DMA\_REQ}}$[12]	Reserved
$\overline{\text{DMA\_REQ}}$[11]	Reserved
$\overline{\text{DMA\_REQ}}$[10]	Reserved
$\overline{\text{DMA\_REQ}}$[9]	Reserved
$\overline{\text{DMA\_REQ}}$[8]	Reserved
$\overline{\text{DMA\_REQ}}$[7]	Reserved
$\overline{\text{DMA\_REQ}}$[6]	Reserved
$\overline{\text{DMA\_REQ}}$[5]	Reserved
$\overline{\text{DMA\_REQ}}$[4]	Reserved
$\overline{\text{DMA\_REQ}}$[3]	Reserved
$\overline{\text{DMA\_REQ}}$[2]	Reserved
$\overline{\text{DMA\_REQ}}$[1]	Reserved
$\overline{\text{DMA\_REQ}}$[0]	Reserved

Chapter 10
LCD Controller

The liquid crystal display controller (LCDC) provides display data for an external gray-scale or color LCD panel. The LCDC is capable of supporting black and white, gray-scale, passive matrix color, and active matrix color LCD panels.

10.1 Features

The MC68SZ328's LCDC offers the following key features:

- Support for single (non-split) screen monochrome/color LCD panels and self-refresh type LCD panels
- 16 simultaneous gray-scale levels from a palette of 16 for monochrome display
- Support for:
  - 4/8 bits per pixel (bpp) for passive color panel
  - 4/8/12/16 bpp for TFT panel
  - Up to 256 colors out of a palette of 4096 colors for an 8 bpp display and 4096 colors for a 12 bpp display
  - True 64K colors for 16 bpp

  Table 10-1 and Table 10-2 summarize this and additional information.
- Standard panel interface for common LCD drivers
- Panel interface of 16-, 12-, 8-, 4-, 2-, and 1-bit-wide LCD panel data bus for monochrome or color panels
- Interface to passive and active color panel (TFT)
- Direct interface to Sharp 320 x 240 HR-TFT panel
- Capability to share system memory or eSRAM for display memory
- Hardware blinking cursor that is programmable at a maximum 63 x 63 pixels
- Logical operation between color hardware cursor and background
- Hardware panning (soft horizontal scrolling)
- 8-bit pulse-width modulator for software contrast control

Figure 10-1 on page 10-3 is a block diagram of the LCDC.

Table 10-1. Supported Panel Characteristics with External Memory Used for Display Memory

Panel Type	Bits/Pixel	Panel interface (Bits)	Number of Gray-Scale Levels/Colors	Max. Panel Size Supported (Pixels)[1]
Monochrome	1	1, 2, 4, 8	Black and white	640x480
	2	1, 2, 4, 8	4	640x480
	4	1, 2, 4, 8	16	640x480
CSTN	4, 8	8	16, 256	640x480
TFT	4, 8	16	16, 256	640x480
	12, 16	12, 16	4096, 64K	640x480

1. The actual panel size depends on system bandwidth utilization.

Table 10-2. Supported Panel Characteristics with eSRAM Used for Display Memory

Panel Type	Bits/Pixel	Panel interface (Bits)	Number of Gray-Scale Levels/Colors	Max. Panel Size Supported (Pixels)
Monochrome	1	1, 2, 4, 8	Black and white	640×480
	2	1, 2, 4, 8	4	640×480
	4	1, 2, 4, 8	16	320×240 or 240×320
CSTN	4, 8	8	16, 256	320×240 or 240×320
TFT	4, 8	16	16, 256	320×240 or 240×320
	12, 16	12, 16	4096, 64K	240×160 or 160×240

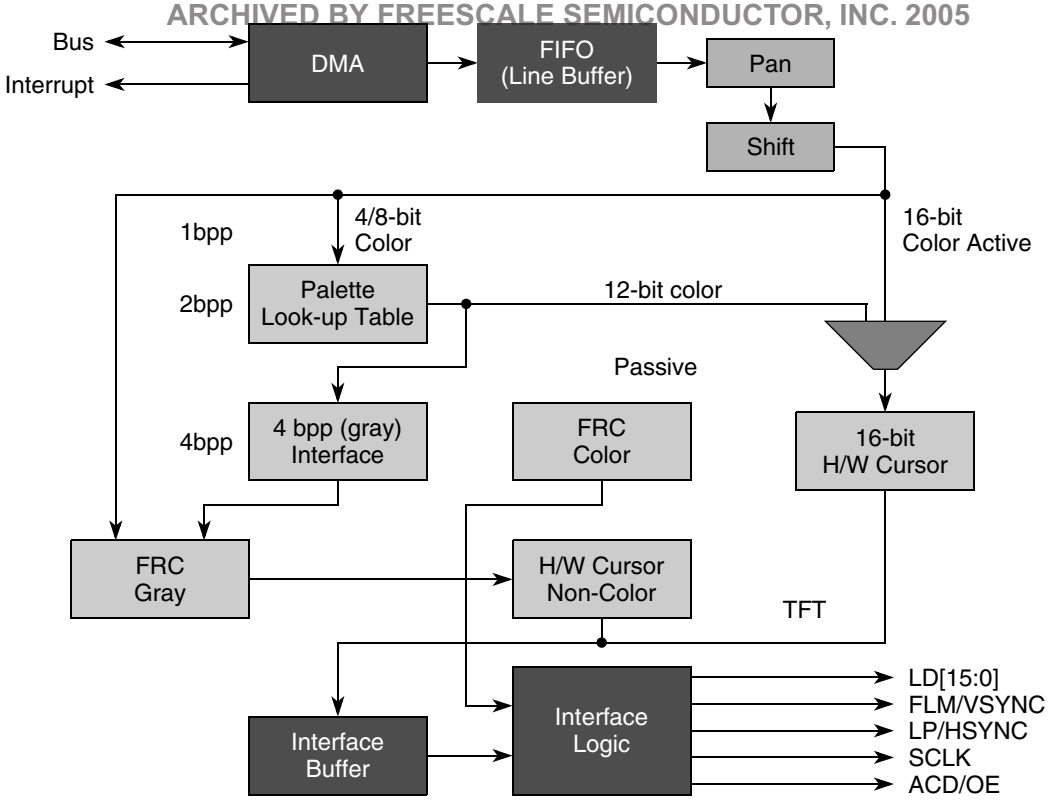


Figure 10-1. LCDC Block Diagram

10.2 Operation

The following sections describe the operation of the LCDC.

10.2.1 LCD Screen Format

The number of pixels forming the screen width and screen height of the LCD panel are software programmable. Figure 10-2 shows the relationship between the portion of a large graphics file displayed on screen and the actual page.

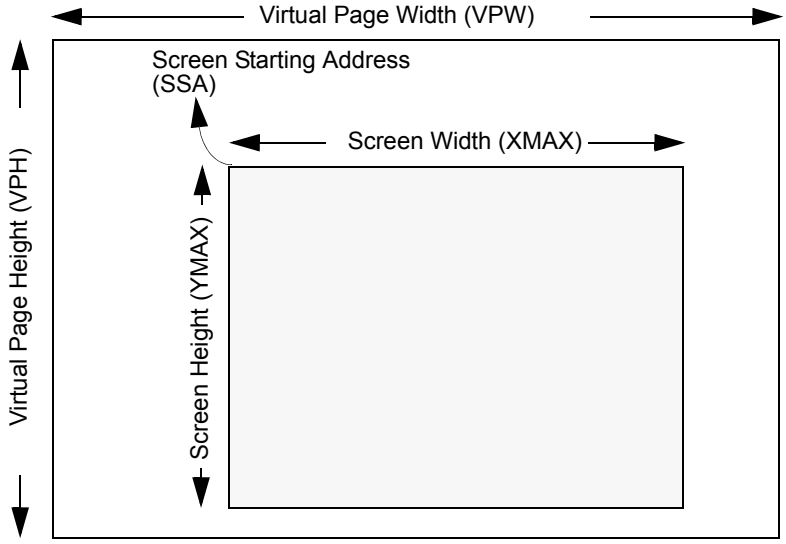


Figure 10-2. LCD Screen Format

The Screen Width (XMAX) and Screen Height (YMAX) parameters specify the LCD panel size. The LCDC will start scanning the display memory at the location pointed to by the Screen Starting Address (SSA) register. Therefore, the shaded area in Figure 10-2 will be displayed 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 Screen Starting Address (SSA) register, a screen-sized window can be vertically or horizontally scrolled (panned) anywhere inside the virtual page boundaries. However, software must position the starting address in SSA properly so that the scanning logic's System Memory Pointer (SMP) does not stretch beyond VPW or VPH; otherwise strange artifacts will appear on the screen. VPH is only used by the programmer for boundary checks. There is no VPH parameter internal to the LCDC.

VPW is used to calculate 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 with SSA to yield the starting address of that line.

10.2.2 Panning

Panning Offset (POS) is expressed in bits, not pixels, so if there is more than 1 bit/pixel, only even pixel boundaries are valid. Notice that in 12 bpp mode, the pixels are aligned to 16-bit boundaries. POS must also align to these boundaries.

SSA and POS are located in isolated registers because they are dynamic parameters that will likely change while the LCDC is running. They are also double buffered for this reason. New values of SSA and POS do not take effect until the beginning of the next frame. A typical panning algorithm enables an interrupt at the beginning of the frame. In the interrupt service routine, POS, SSA, or both are updated (the old values are latched internally). The updates take effect on the next frame.

10.2.3 Display Data Mapping

The LCDC supports 1/2/4 bpp in monochrome mode and 4/8/12/16 bpp in color mode. System memory data mapping in 2/4/8/12/16 bpp modes is shown in Figure 10-3 and in Figure 10-4.

NOTE:

In 12 bpp mode, 16 bits of memory are used for each set of 12 bits, leaving 4 bits unused. In 16 bpp mode, all 16 bits are used. Refer to Figure 10-4 and Table 10-4 for TFT mode LD bit color channels assignments.

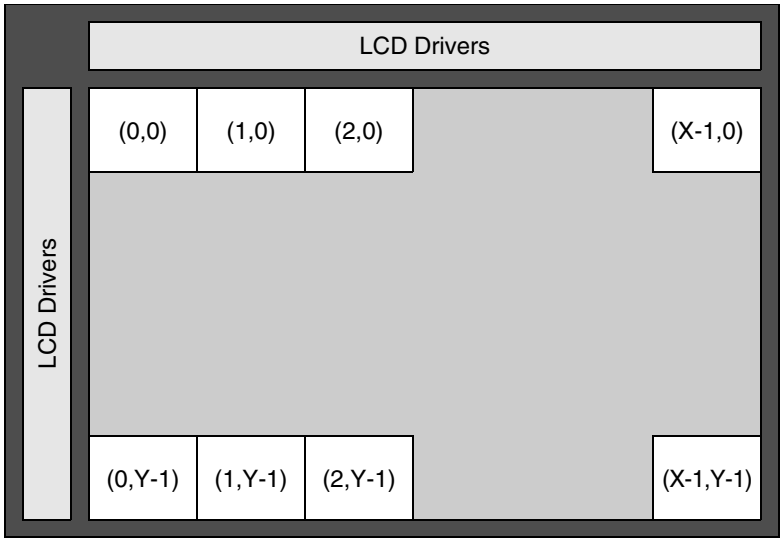


Figure 10-3. Mapping of Memory Data on the Screen

In the binary (1 bpp) mode, each bit in the display memory corresponds to a pixel in the LCD panel. The corresponding pixel on the screen is either fully on or fully off.

1 bpp Mode

7	6	5	4	3	2	1	0
(0,0)	(1,0)	(2,0)	(3,0)	(4,0)	(5,0)	(6,0)	(7,0)

2 bpp Mode

7	6	5	4	3	2	1	0
(0,0)		(1,0)		(2,0)		(3,0)	
(X-8,Y-1)		(X-7,Y-1)		(X-6,Y-1)		(X-5,Y-1)	
(X-4,Y-1)		(X-3,Y-1)		(X-2,Y-1)		(X-1,Y-1)	

4 bpp Mode

7	6	5	4	3	2	1	0
(0,0)				(1,0)			
(2,0)				(3,0)			
(X-4,Y-1)				(X-3,Y-1)			
(X-2,Y-1)				(X-1,Y-1)			

8 bpp Mode

7	6	5	4	3	2	1	0
(0,0)							
(1,0)							
(X-2,Y-1)							
(X-1,Y-1)							

12 bpp Mode (TFT)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R3	R2	R1	R0		G3	G2	G1	G0			B3	B2	B1	B0		(0,0)
R3	R2	R1	R0		G3	G2	G1	G0			B3	B2	B1	B0		(1,0)
R3	R2	R1	R0		G3	G2	G1	G0			B3	B2	B1	B0		(X-2,Y-1)
R3	R2	R1	R0		G3	G2	G1	G0			B3	B2	B1	B0		(X-1,Y-1)

16 bpp Mode (TFT)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0	(0,0)
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0	(1,0)
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0	(X-2,Y-1)
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0	(X-1,Y-1)

Figure 10-4. Memory Mapping Mode Bit Assignments

10.2.4 Gray-Scale Operation

The LCDC is capable of generating up to 16 gray levels. These gray levels are defined by 2 or 4 bits of display data for each pixel. Using 2 bpp, the LCDC can display 4 shades out of a maximum of 16 shades. Using 4 bpp, the LCDC can display all 16 shades. Different 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). It is described in section 10.2.6.

Use of the mapping RAM is shown in Figure 10-5. When using 2 bpp, the mapping RAM is used to map the 2-bit code to 1 of 4 gray levels, but with 4 bpp, 4-bit code is mapped to 1 of 16 gray levels. Because crystal formulations and driving voltages vary, the visual gray effect may or may not be related linearly 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 10-5 illustrates gray-scale pixel generation. The flexible mapping scheme allows the user to optimize the visual effect for a specific panel or application.

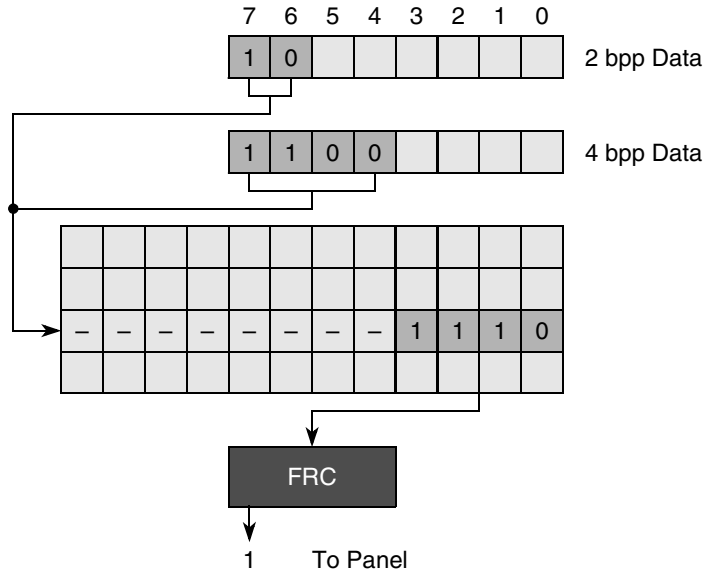


Figure 10-5. Gray-Scale Pixel Generation

10.2.5 Color Generation

The value corresponding to each color pixel on the screen is represented via a 4-, 8-, 12-, or 16-bit code in the display memory. In 12-bit/16-bit mode, pixel data is simply moved from display memory to the 12-bit/16-bit LCDC output bus. The 8- and 4-bit modes use a color mapping RAM programmed inside the LCDC. The 4- or 8-bit data is mapped to a 12-bit RGB code. For active matrix displays, the 12-bit RGB code from the mapping RAM is output to the panel. For passive matrix color displays, the 12-bit RGB code is output to the FRC (Frame Rate Control) blocks, which independently process the code corresponding to the red, green, and blue components of each pixel to generate the required shade and intensity. See Section 10.2.6, “Frame Rate Modulation Control (FRC).” Figure 10-6 and Figure 10-7 illustrate passive matrix and active matrix color pixel generation, respectively.

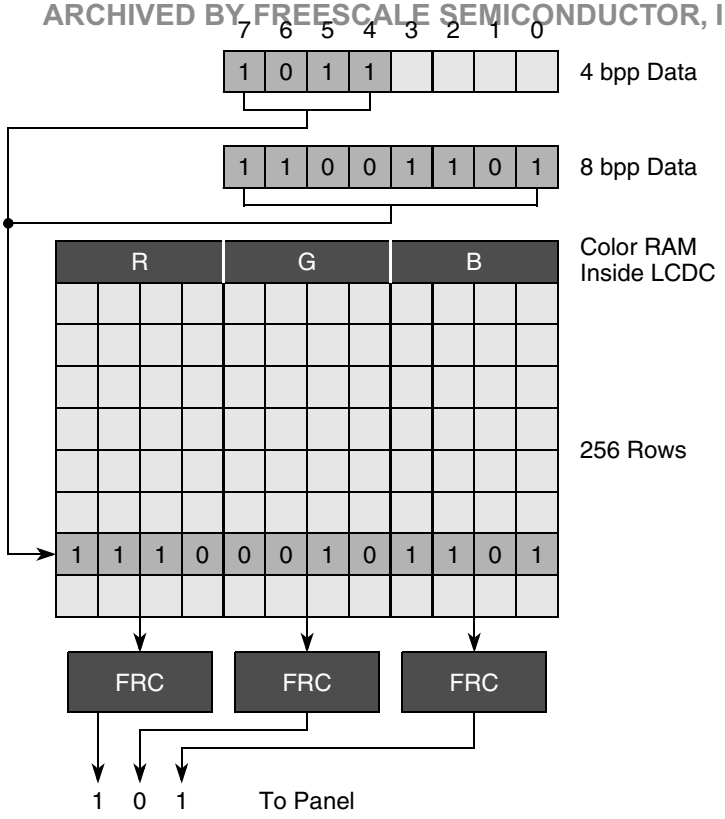


Figure 10-6. Passive Matrix Color Pixel Generation

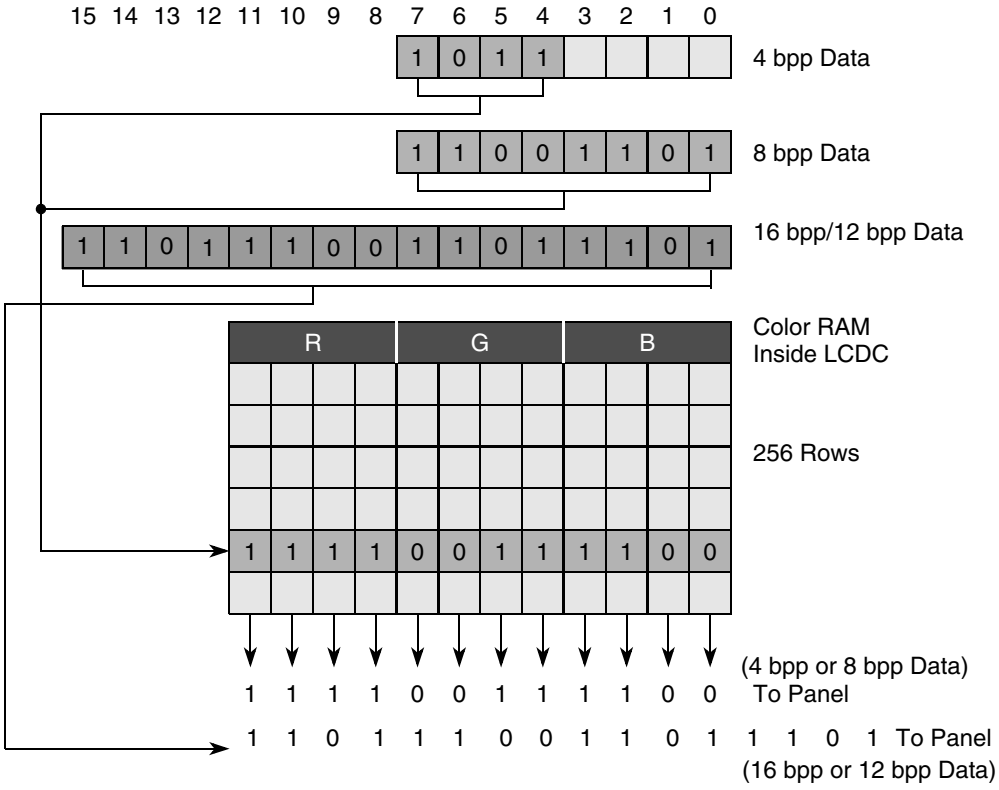


Figure 10-7. Active Matrix Color Pixel Generation

10.2.6 Frame Rate Modulation Control (FRC)

The FRC algorithm used with the MC68SZ328 is the same as for the MC68VZ328. See Table 10-3.

Table 10-3. Grey Palette Density

Gray Code (Hex)	Density	Density (in Decimal)
0	0	0
1	1/8	0.125
2	1/5	0.2
3	1/4	0.25
4	1/3	0.333
5	2/5	0.4
6	4/9	0.444
7	1/2	0.5
8	5/9	0.555
9	3/5	0.6
A	2/3	0.666
B	3/4	0.75
C	4/5	0.8
D	7/8	0.875
E	14/15	0.933
F	1	1

10.2.7 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 10-8.

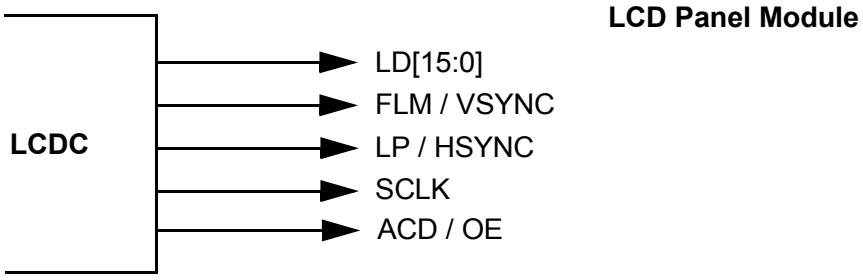


Figure 10-8. LCD Module Interface Signals

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 select either black-and-white mode or color mode.

10.2.7.1 Passive Panel Interface Signals

Figure 10-9 shows the LCD interface timing for monochrome panels, and Figure 10-10 shows the LCD interface timing for passive matrix color panels. Signal polarities are shown as positive, but they can be reversed by clearing the appropriate register bit. The data bus timing for passive panels is controlled by the Shift Clock (SCLK), Line Pulse (LP), First Line Marker (FLM), Alternate Crystal Direction (ACD), and Data (LD) signals. Operation of the panel interface signals is as follows:

1. SCLK 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 data (and 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 is used to prevent degradation of the LCD panel.

NOTE:

LD bus width is programmable to 1, 2, 4, or 8 bits in monochrome mode (color = 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 illustrated in Figure 10-10.

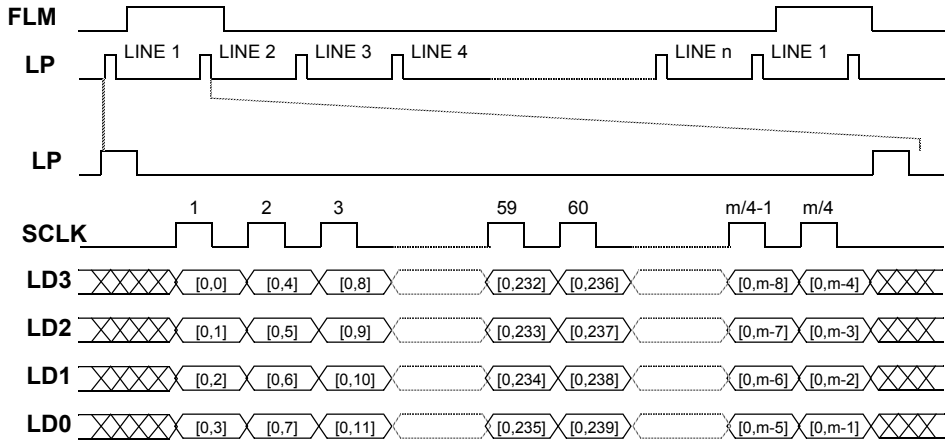


Figure 10-9. LCD Interface Timing for 4-Bit Data Width Gray-Scale Panels

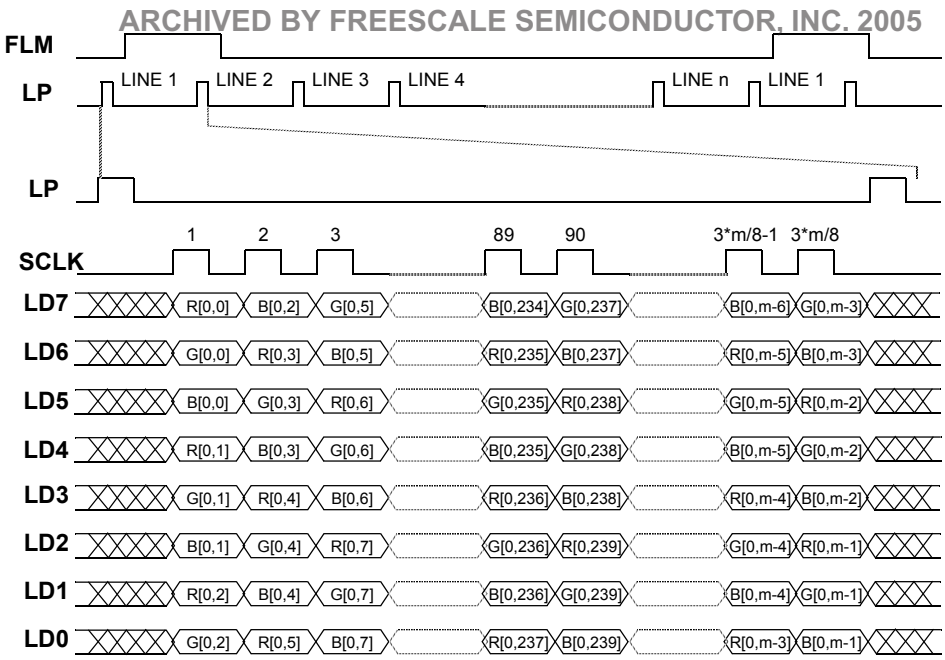


Figure 10-10. LCD Interface Timing for 8-Bit Data Passive Matrix Color Panels

10.2.7.2 Passive Panel Interface Timing

Figure 10-11 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 passive panel interface timing are as follows:

- 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. 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 units of pixel clock period, unless stated otherwise.

Figure 10-12 shows the panel vertical timing (timing of one frame) for passive, color mode.

The delay from the end of one frame until the beginning of the next is programmable. Signal event timing is as follows:

1. Pass_frame_wait defines the delay from the last line of the frame to the beginning of the FLM signal, in Passive Color mode (see Section 10.3, “Programming Model,” for details).
2. YMAX defines the number of lines in a frame.

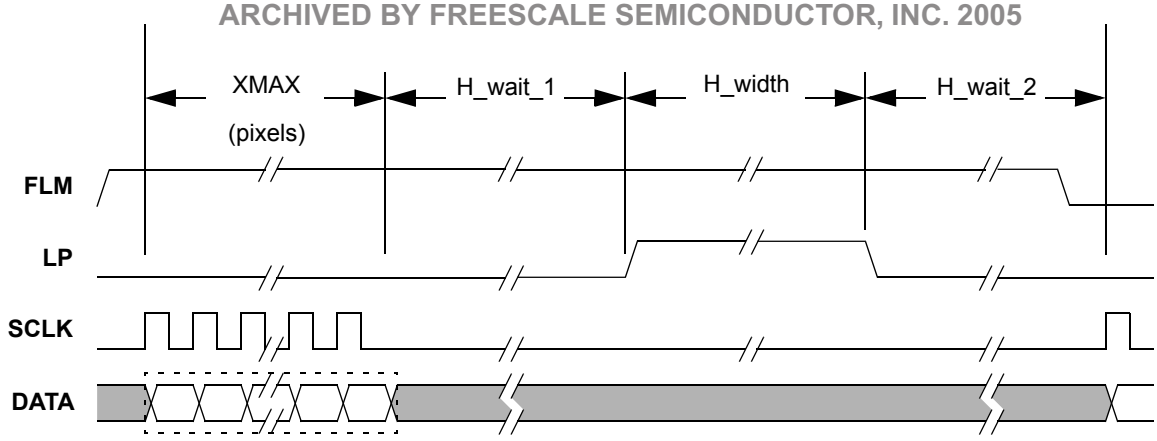


Figure 10-11. Horizontal Sync Pulse Timing in Passive (Non-TFT) Mode

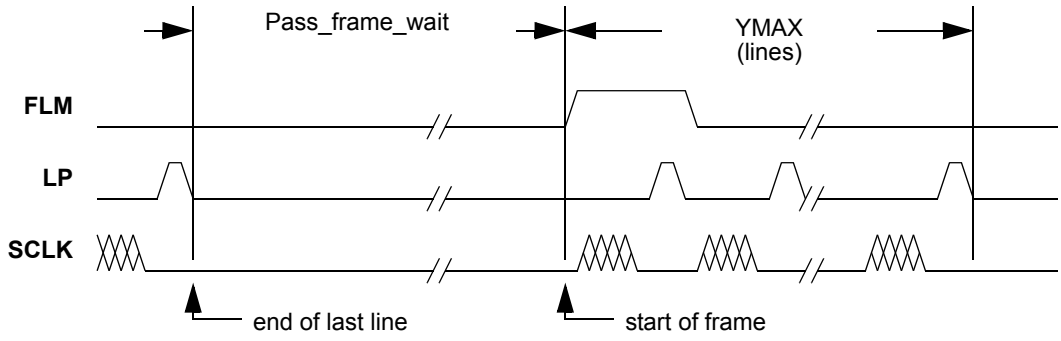


Figure 10-12. Vertical Sync Pulse Timing Passive (Non-TFT) Mode

10.2.7.3 Active Matrix Panel Interface Signals

Figure 10-13 shows the LCD interface timing for an active matrix color TFT panel. This figure shows signals with negative polarity (FLMpol = 0, LPpol = 0, CLKpol = 0, OEpol = 0). The panel interface timing for active matrix panels is sometimes referred to as a “digital CRT.” It is controlled by the Shift Clock (SCLK), 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 Data (LD) signals. The sequence of events for active matrix interface timing is as follows:

1. SCLK latches data into the panel on its negative edge (when positive polarity is selected). Unlike in the passive mode, it runs continuously.
2. HSYNC causes the panel to start a new line.
3. VSYNC causes the panel to start a new frame. It always includes at least one HSYNC pulse.
4. OE functions as an output enable signal to the display. 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.

10.2.7.3.1 Color Channel Assignments

The LD is a 12-bit/16-bit bus width. In 12-bit mode, bits [15:12] are for red, bits [10:7] are for green, and bits [4:1] are for blue. In 16-bit mode, bits [15:11] are for red, bits [10:5] are for green, and bits [4:0] are for blue. The actual TFT color channel assignments are shown in Table 10-4

Table 10-4. TFT Color Channels on LD Bus

	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	R3	R2	R1	R0	X	G3	G2	G1	G0	X	X	B3	B2	B1	B0	X
8 bpp	R3	R2	R1	R0	X	G3	G2	G1	G0	X	X	B3	B2	B1	B0	X
12 bpp	R3	R2	R1	R0	X	G3	G2	G1	G0	X	X	B3	B2	B1	B0	X
16 bpp	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

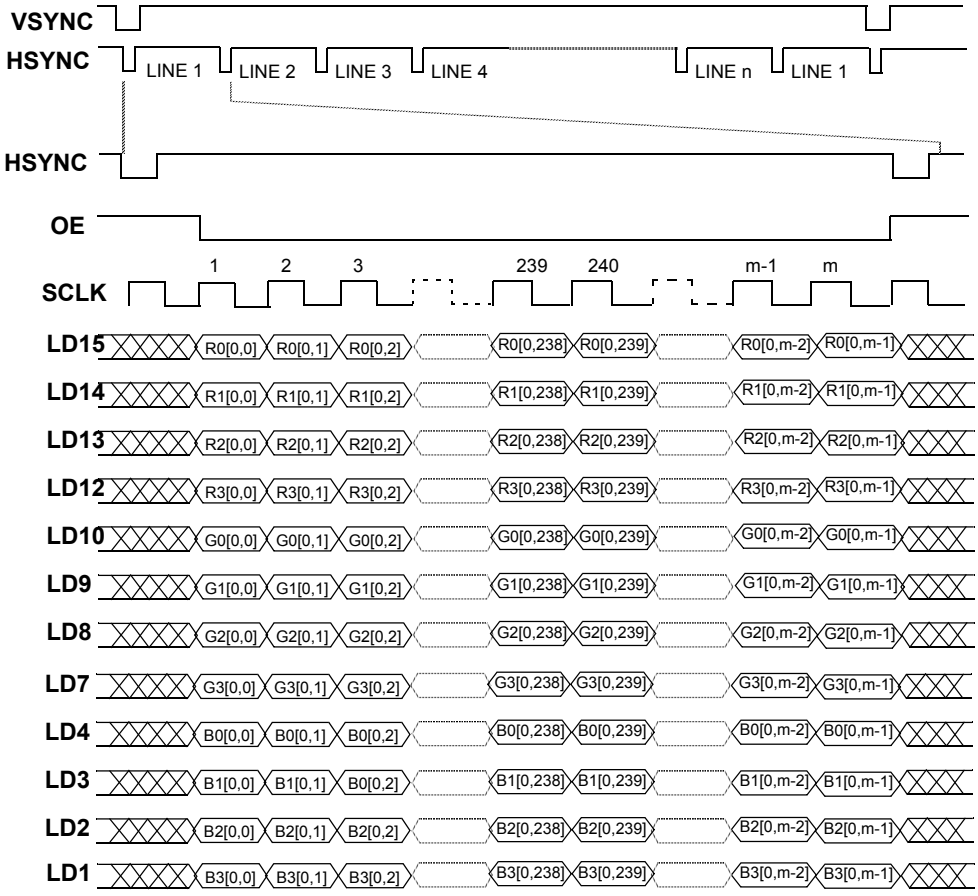


Figure 10-13. LCD Interface Timing for Active Matrix Color Panels

10.2.7.4 Active Panel Interface Timing

Figure 10-14 shows the horizontal timing (timing is shown for only 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. H_width 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 the end of OE to the beginning of the HSYNC pulse.
- XMAX defines the number of pixels per line. XMAX is the total number of pixels per line.

NOTE:

All parameters are defined in pixel periods, not SCLK periods.

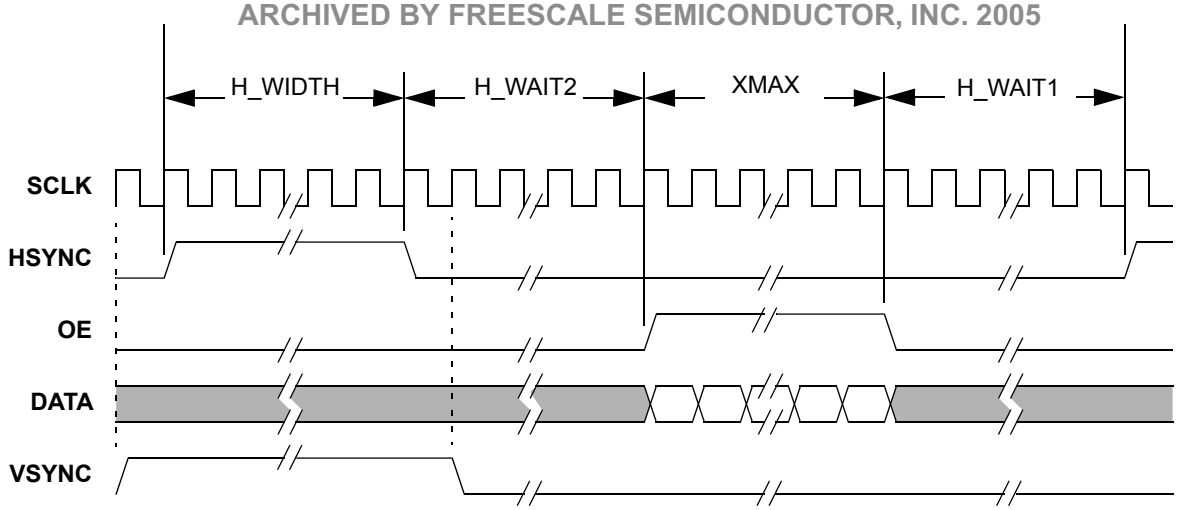


Figure 10-14. Horizontal Sync Pulse Timing in TFT Mode

Figure 10-15 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 timing signal parameters are defined as follows:

- V_wait_1 is a delay measured in lines. For a V_wait_1 of 1, there is a delay of one HSYNC (time equals one line period) before VSYNC, and so on. The HSYNC pulse is output during the V_wait_1 delay.
- For a V_width (Vertical Sync Pulse Width) value of 0, VSYNC includes one HSYNC pulse. For a VSW of 2, VSYNC includes two HSYNC pulses, and so forth.
- V_wait_2 is a delay measured in lines. For a V_wait_2 of 1, there is a delay of one HSYNC (time equals one line period) after VSYNC, and so on. The HSYNC pulse is output during the V_wait_2 delay.

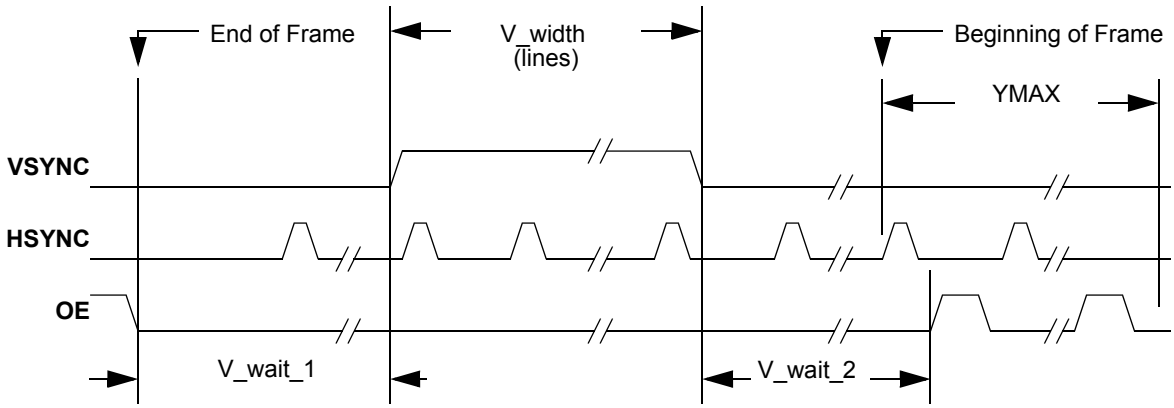


Figure 10-15. Vertical Sync Pulse Timing in TFT Mode

10.3 Programming Model

The LCDC memory space contains twenty-one 16-bit registers for display parameters, a read-only status register, and a 256 × 12 color mapping RAM. The color mapping RAM is physically located inside the Palette Lookup Table module. The base address of the LCDC is 0xFFFE0800 (hexadecimal).

Only *word* access is supported. Byte and half-word accesses are undefined.

Note that there are addressing "gaps" both between the read-write register section and the status register and between the status register and the color mapping RAM. Table 10-5 summarizes the register memory mapping in the LCDC.

Table 10-5. Register Memory Mapping Summary

	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00	Screen Start Address[15:1]															
02	Screen Start Address[31:16]															
04	Screen Width[9:3]							Screen Height[8:0]								
06							Virtual Page Width (words)[9:0]									
08	Cur Ctrl[1:0]			Oper			Cursor X Position[9:0]									
0a								Cursor Y Position[8:0]								
0c	Blink En		Cursor Width[5:0]								Cursor Height[5:0]					
0e									Blink Divisor[7:0]							
10	Cursor Color (Red)[4:0]					Cursor Color (Green)[5:0]						Cursor Color (Blue)[4:0]				
12	ACD Sel	Alternate Crystal Direction Control[6:0]							SClk Sel	Pixel Clock Divisor[6:0]						
14	TFT	Color	Panel Bus Width[1:0		Bit/Pixel[2:0]			Pixel Pol	FLM Pol	LP Pol	OE Pol	Sclk Pol	Sclk Idle			Sharp
16	Horizontal Wait 1[7:0]								Horizontal Wait 2[7:0]							
18	Horizontal Sync Pulse Width[5:0]															
1a	Vertical Wait 1[7:0]								Vertical Wait 2[7:0]							
1c	Vertical Sync Pulse Width[5:0]								Passive Mode Idle Counter[7:0]							
1e													Panning Offset[3:0]			
20									Gray Palette 2[3:0]				Gray Palette 1[3:0]			
22	LD Mask					PWM Clk Source[1:0]		Cont En	Pulse Width[7:0]							
24					DMA End Marker[3:0]								DMA Low Marker[3:0]			
26											Rot. Burst	Enlarge	Rotation [1:0]		LCD En	Self Ref
28														Intr Sync		Intr Cond
2a															EOF Intr	BOF Intr

Table 10-5. Register Memory Mapping Summary (Continued)

	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
:																
200					First Color Mapping RAM Location (R[3:0], G[3:0], B[3:0])[1]											
:					:											
3fe					Last Color Mapping RAM Location (R[3:0], G[3:0], B[3:0])[1]											

1. The color mapping is physically located in the Palette Lookup Table module (lcdc_lut).

10.3.1 LCD Screen Start Address Register

The LCD screen start address register controls the address at which the LCDC begins fetching pixels for a new frame. The settings for the LSSA register are shown in Table 10-6.

LSSA LCD Screen Start Address Register **0x(FF)FE0800**

	BIT 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	BIT 16
	SSA[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
	0x0000															

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	SSA[15:1]															
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	0
	0x0000															

Table 10-6. LCD Screen Start Address Register Settings

Name	Description	Setting
SSA[31:1] Bits 31–1	**Screen Start Address of LCD Panel**—The LCDC begins fetching pixel data for a new frame from the SSA[31:1] address. SSA bits 0 are fixed at 0. This field must start at a location that will enable a complete picture to be stored in a 4 Mbyte memory boundary (A[21:0]). In other words, A[31:22] has a fixed value for a picture's image.	See description.
Reserved Bit 0	Reserved	This bit is reserved and always reads 0.

10.3.2 LCD Screen Size Register

The LCD screen size register defines the width and height of the LCD screen.

LSS LCD Screen Size Register **0x(FF)FE0804**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	XMAX[15:9]							YMAX[8:0]								
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 10-7. LCD Screen Size Register Settings

Name	Description	Setting
XMAX[15:9] Bits 15–9	**Screen Width**—Screen width in multiples of 8 pixels. XMAX[15:9] equals the actual screen size on the X axis divided by 8. For black and white panel, XMAX[10] is ignored, forcing the screen size on X to be a multiple of 16 pixels/line. **Note:** XMAX[15:9] ≥ 4 for black and white mode; otherwise XMAX[15:9] ≥ 2.	See description
YMAX[8:0] Bits 8–0	**Screen Height**—This register 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.	See description

10.3.3 LCD Virtual Page Width Register

The LCD virtual page width register defines the virtual page width of the LCD panel.

LVPW LCD Virtual Page Width Register **0x(FF)FE0806**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
							VPW[9:0]									
TYPE							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 10-8. LCD Virtual Page Width Register Settings

Name	Description	Setting
Reserved Bits 15–10	Reserved	These bits are reserved and always read 0.
VPW[9:0] Bits 9–0	**Virtual Page Width**—The virtual page width of the LCD panel. The VPW[9:0] bits are a count of 16-bit words required to hold the data for one virtual line. VPW is used to calculate the starting address representing the beginning of each displayed line; see Section 10.3.1, “LCD Screen Start Address Register.”	See description

10.3.4 LCD Panel Configuration Registers

There are two LCD panel configuration registers, LPCON0 and LPCON1.

LPCON0 LCD Panel Configuration Register 0 **0x(FF)FE0812**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	ACDSEL	ACD[6:0]							SCLKSEL	PCD[6:0]						
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 10-9. LCD Panel Configuration Register 0 Settings

Name	Description	Setting
ACDSEL Bit 15	**ACD Clock Source Select**—This bit selects the clock source used for the alternate crystal direction count.	0 = FRM is used as clock source for ACD count. 1 = LP/HSYNC is used as clock source for ACD count.
ACD[6:0] Bits 14–8	**Alternate Crystal Direction Control**—The ACD signal toggles once every 1 to 16 FLM cycles based on the value specified by the ACD parameter. The actual number of FLM cycles is the value programmed plus one. Active Mode (TFT = 1) - Un-used Passive Mode (TFT = 0).	See description.
SCLKSEL Bit 7	**SCLK Select**—This bit selects whether to enable or disable SCLK in TFT mode when there is no data output.	0 = Disable OE and SCLK in TFT mode when there is no data output. 1 = Always enable SCLK in TFT mode, even when there is no data output.
PCD[6:0] Bits 6–0	**Pixel Clock Divider**—The system clock is divided by N (PCD plus one) to yield the pixel clock rate. Values of 1-127 will yield values for N of 2-128. The pixel clock rate and the SCLK rate are only the same if PBSIZ = 00 (single pixel output bus). The pixel clock rate is faster than SCLK by a factor equal to the number of pixels in an output vector. PCD must be greater than or equal to 1 (N = 2) in order to have enough clock edges to generate the SCLK output. For passive matrix color panels (COLOR = 1, TFT = 0, PBSIZ = 11), PCD must be greater than or equal to 2.	See description.

LPCON1 LCD Panel Configuration Register 1 **0x(FF)FE0814**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	TFT	COLOR	PBSIZ[1:0]		BPP[2:0]			PIX POL	FLM POL	LP POL	SLCK POL	OE POL	SCLK IDLE			SHARP
TYPE	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 10-10. LCD Panel Configuration Register 1 Settings

Name	Description	Setting
TFT Bit 15	**Interfaces to TFT Display**—TFT controls the format and timing of the output control signals. Active and passive displays use different signal timing formats as described in Section 10.2.7, “Panel Interface Signals and Timing.” TFT also controls the use of the FRC in color mode.	0 = The LCD panel is a passive display. 1 = The LCD panel is an active display: “digital CRT” signal format, FRC is bypassed, LD bus width is fixed at 16 bits.
COLOR Bit 14	**Interfaces to Color Display**—Used in passive mode (TFT = 0) to activate three channels of FRC and to use the special 2 2/3 pixels per output vector format.	0 = The LCD panel is a monochrome display. 1 = The LCD panel is a color display.
PBSIZ[1:0] Bits 13–12	**Panel Bus Width**—LCD Panel Bus Size is applicable for monochrome or passive matrix color monitors (TFT = 0).	00 = 1-bit 01 = 2-bit 10 = 4-bit 11 = 8-bit
BPP[2:0] Bits 11–9	**Bits Per Pixel**—This field indicates the number of bits per pixel in memory.	000 = 1 bpp, FRC bypassed 001 = 2 bpp 010 = 4 bpp 011 = 8 bpp 100 = 12/16 bpp (16 bits of memory used) 101 = reserved 11x = reserved
PIXPOL Bit 8	**Pixel Polarity**—This bit determines the polarity of the pixels.	0 = LD output same as corresponding memory image, 1 for high, 0 for low 1 = LD output is an inversion of corresponding memory image, 0 for high, 1 for low
FLMPOL Bit 7	**First Line Marker Polarity**—This bit determines the polarity of the first line marker signal.	0 = Frame marker is active high 1 = Frame marker is active low (If the TFT and Sharp panel bit are enable, 0 = Frame Marker active low, and 1 = Frame Marker active high)
LPPOL Bit 6	**Line Pulse Polarity**—This bit determines the polarity of the line pulse signal.	0 = Line pulse is active high 1 = Line pulse is active low
SLCKPOL Bit 4	**LCD Shift Clock Polarity**—This bit determines the polarity of the active edge of the LCD shift clock.	0 = Active negative edge of SLCLK (in TFT mode, active on positive edge of SLCLK). 1 = Active positive edge of SLCLK (in TFT mode, active on negative edge of SLCLK).
OEPOL Bit 5	**Output Enable Polarity**—This bit determines the polarity of the output enable signal	0 = Output enable is active high 1 = Output enable is active low
SCLK_IDLE_EN Bit 3	**SCLK Idle Enable**—This bit enables or disables SCLK when VSYNC is idle in TFT mode.	0 = Disable SCLK when VSYNC is idle in TFT mode. 1 = Enable SCLK when VSYNC is idle in TFT mode.

Table 10-10. LCD Panel Configuration Register 1 Settings (Continued)

Name	Description	Setting
Reserved Bits 2–1	Reserved	These bits are reserved and always read 0.
SHARP Bit 0	**Sharp Panel Enable**—This bit enables or disables signals for Sharp HR-TFT 320 × 240 panels. Please refer to Section 10.3.14, “PWM Contrast Control Register,” for additional settings for the Sharp HR-TFT panel.	0 = Disable Sharp HR-TFT 320 × 240 panel signals. 1 = Enable Sharp HR-TFT 320 × 240 panel signals.

10.3.5 LCD Horizontal Configuration Registers

There are two LCD horizontal configuration registers, LHCON0 and LHCON1.

LHCON0 LCD Horizontal Configuration Register 0 **0x(FF)FE0816**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	HWAIT1[7:0								HWAIT2[7:0]							
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 10-11. LCD Horizontal Configuration Register 0 Settings

Name	Description	Setting
HWAIT1[7:0] Bits 15–8	**Wait Between OE and HSYNC**—The number of pixel clock periods between the last LD of each line and the beginning of the HSYNC. It is equal to (HWAIT1 + 1).	See description
HWAIT2[7:0] Bits 7–0	**Wait Between HSYNC and Start of Next Line**—The number of pixel clock periods between the end of HSYNC and the beginning of the first data of next line. It is equal to (HWAIT2 + 1).	See description

LHCON1 LCD Horizontal Configuration Register 1 **0x(FF)FE0818**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	HWIDTH[5:0]															
TYPE	rw	rw	rw	rw	rw	rw										
RESET	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	0x0400															

Table 10-12. LCD Horizontal Configuration Register 1 Settings

Name	Description	Setting
HWAIT[5:0] Bits 15–10	**Horizontal Sync Pulse Width**—The number of pixel clock periods that HSYNC is activated.	See description
Reserved Bits 9–0	Reserved	These bits are reserved and read 0.

10.3.6 LCD Vertical Configuration Registers

There are two LCD vertical configuration registers, LVCON0 and LVCON1.

LVCON0 LCD Vertical Configuration Register 0 **0x(FF)FE081A**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	VWAIT1[7:0								VWAIT2[7:0]							
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 10-13. LCD Vertical Configuration Register 0 Settings

Name	Description	Setting
VWAIT1[7:0] Bits 15–8	**Wait Between Frames 1**—In active mode (TFT = 1), this field is the delay between the end of VSYNC and the beginning of OE, measured in lines. This is one component of the vertical dead time between frames. In passive mode (TFT = 0), this field is the delay, measured in units of the virtual clock period (see the definition of PASS_DIV in Table 10-14), between the last line of the frame to the beginning of the next frame.	See description
VWAIT2[7:0] Bits 7–0	**Wait Between Frames 2**—In active (TFT = 1) mode, this field is the delay between the end of VSYNC and the beginning of OE of the first line, measured in lines.	See description

LVCON1 LCD Vertical Configuration Register 1 **0x(FF)FE081C**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	VWIDTH[5:0]								PASS_DIV[7:0]							
TYPE	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
	0x0401															

Table 10-14. LCD Vertical Configuration Register 1 Settings

Name	Description	Setting
VWIDTH[5:0] Bits 15–10	**Vertical Sync Pulse Width**—In active (TFT = 1) mode, this field is the width of the VSYNC pulse, measured in lines. For a value of 0001, the vertical sync pulse includes one HSYNC pulse. For a value of 0002, the vertical sync pulse includes two HSYNC pulses, and so on.	See description
Reserved Bits 9–8	Reserved	These bits are reserved and read 0.
PASS_DIV[7:0] Bits 7–0	**Passive Clock Divider**—In passive (TFT = 0) mode, this field defines the period of the virtual clock period, which is used for counting the delay between the last line of the frame and the beginning of the next frame. If PASS_DIV = 0, there will be no delay between the last line of the frame to the beginning of the next frame. The value of VWAIT1 is ignored.	See description

10.3.7 LCD Panning Offset Register

The LCD panning offset register contains the panning offset value.

LPANOFF LCD Panning Offset Register **0x(FF)FE081E**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
													POS[3:0]			
TYPE													rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 10-15. LCD Panning Offset Register Settings

Name	Description	Setting
Reserved Bits 15–4	Reserved	These bits are reserved and should read 0.
POS[3:0] Bits 3–0	**Panning Offset**—The number of pixel clock periods that HSYNC is activated. The data from memory is panned to the left by the POS field before processing. To pan the final image by N bits, POS is read by the LCDC once per frame (similar to double buffering) at the beginning of the frame.	See description and Table 10-16

Table 10-16. Panning Offset Values

Bits per Pixel	POS	Effective Number of Bits Panned on image
1	N	N
2	2N	N
4	4N	N
8	8N	N

10.3.8 LCD Cursor X Position Register

The LCD cursor X position (LCXP) register is used to determine the horizontal position of the cursor on the LCD panel.

LCXP LCD Cursor X Position Register **0x(FF)FE0808**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	CC[1:0]			OP			CXP[9:0]									
TYPE	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 10-17. LCD Cursor X Position Register Settings

Name	Description	Setting
CC[1:0] Bits 15–14	**Cursor Control**—These bits are used to control the format of the cursor and the type of arithmetic operations.	When OP = 0 and color mode disabled: 00 = Transparent, cursor is disabled 01 = Full (black) cursor 10 = Reversed video 11 = Full (white) cursor When OP = 0 and color mode enabled: 00 = Transparent, cursor is disabled 01 = Full color cursor 10 = Full color cursor with each bit inverted 11 = Bit operation logical AND between background and cursor When OP = 1 and color mode enabled: 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 13	Reserved	This bit is reserved and reads 0.
OP Bit 12	**Arithmetic Operation**—This bit enables or disables arithmetic operations between the background and the cursor.	0 = Disable arithmetic operation 1 = Enable arithmetic operation between background and cursor
Reserved Bits 11–10	Reserved	These bits are reserved and should read 0.
CXP[9:0] Bits 9–0	**Cursor X Position**—This field represents the cursor's horizontal starting position, X, in a multiple of 8-pixel count (a multiple of 16-pixel count in black and white) from 0 to XMAX.	See description

10.3.9 LCD Cursor Y Position Register

The LCD cursor Y position (LCYP) register is used to determine the vertical position of the cursor on the LCD panel.

LCYP LCD Cursor Y Position Register **0x(FF)FE080A**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
								CYP[8:0]								
TYPE								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 10-18. LCD Cursor Y Position Register Settings

Name	Description	Setting
Reserved Bits 15–9	Reserved	These bits are reserved and read 0.
CYP[8:0] Bits 8–0	**Cursor Y Position**—These bits represent the cursor's vertical starting position, Y, in pixel count (from 0 to YMAX).	See description

10.3.10 LCD Cursor Size Register

The LCD cursor size (LCSR) register is used to determine the width and height of the cursor.

LCSR LCD Cursor Size Register **0x(FF)FE080C**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	BKEN		CW[5:0]								CH[5:0]					
TYPE	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	1
	0x0101															

Table 10-19. LCD Cursor Size Register Settings

Name	Description	Setting
BKEN Bit 15	**Blink Enable**—This bit determines if the cursor will blink or remain steady. **Note:** The cursor is disabled if the CW or CH bits are set to 0.	0 = Blink is disabled 1 = Blink is enabled
Reserved Bit 14	Reserved	This bit is reserved and reads 0.
CW[5:0] Bits 13–8	**Cursor Width**—These bits specify the width of the hardware cursor in pixel count (from 1 to 63). **Note:** The cursor is disabled if these bits are set to 0.	See description
Reserved Bits 7–6	Reserved	These bits are reserved and read 0.
CH[5:0] Bits 5–0	**Cursor Height**—These bits specify the height of the hardware cursor in pixel count (from 1 to 63). **Note:** The cursor is disabled if these bits are set to 0.	See description

10.3.11 LCD Blink Control Register

The LCD blink control register (LBLKC) is used to control how the cursor blinks.

LBLKC LCD Blink Control Register **0x(FF)FE080E**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
									BD[7:0]							
TYPE									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 10-20. LCD Blink Control Register Settings

Name	Description	Setting
Reserved Bits 15–8	Reserved	These bits are reserved and read 0.
BD[7:0] Bits 7–0	**Blink Divisor**—A 32 Hz clock from RTC is used to clock the 7-bit up counter. When the counter value equals BD, the cursor will toggle on/off.	See description

10.3.12 LCD Color Cursor Mapping Register

The LCD color cursor mapping register defines the color of the cursor in color mode (passive and TFT).

LCUR_COL LCD Color Cursor Mapping Register **0x(FF)FE0810**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	CUR_COL_R[4:0]					CUR_COL_B[5:0]						CUR_COL_G[4:0]				
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 10-21. LCD Color Cursor Mapping Register Settings

Name	Description	Setting
CUR_COL_R[4:0] Bits 15–11	**Cursor Color - Red**—Using 16-bit color mapping, this field defines the red component of the cursor color when it is in color mode (passive and TFT).	See description
CUR_COL_B[5:0] Bits 10–5	**Cursor Color - Blue**—Using 16-bit color mapping, this field defines the blue component of the cursor color when it is in color mode (passive and TFT).	See description
CUR_COL_G[4:0] Bits 4–0	**Cursor Color - Green**—Using 16-bit color mapping, this field defines the green component of the cursor color when it is in color mode (passive and TFT).	See description

10.3.13 LCD Gray Palette Mapping Register

For four-level gray-scale displays, full black and full white are the two predefined display levels. The other two intermediate gray-scale shading densities can be adjusted in the LCD gray palette mapping register (LGPMR).

LGPMR LCD Gray Palette Mapping Register **0x(FF)FE0820**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
									GP2[3:0]				GP1[3:0]			
TYPE									rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1
	0x0073															

Table 10-22. LCD Gray Palette Mapping Register Settings

Name	Description	Setting
Reserved Bits 15–8	Reserved	These bits are reserved and read 0.
GP2[3:0] Bits 7–4	**Gray-Scale Palette Mapping 2**—These bits represent one of the two gray-scale shading densities.	See description
GP1[3:0] Bits 3–0	**Gray-Scale Palette Mapping 1**—These bits represent the other gray-scale shading density.	See description

10.3.14 PWM Contrast Control Register

The pulse-width modulator contrast control register (PWMR) is used to control PWMO signal, which controls the contrast of the LCD panel.

PWMR PWM Contrast Control Register **0x(FF)FE0822**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	LDMASK					SCR[1:0]		CCEN	PW[7:0]							
TYPE	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 10-23. PWM Contrast Control Register Settings

Name	Description	Setting
LDMASK Bit 15	**LD Mask**—This bit is used to enable or disable the LD output to 0, 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 read 0.

Table 10-23. PWM Contrast Control Register Settings (Continued)

Name	Description	Setting
SRC[1:0] Bits 10–9	**Clock Source**—These bits select the input clock source for the PWM counter. Therefore, 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 **Note:** To drive Sharp TFT 320 × 240 HR panel, SRC0 is set to 1 and SCR1 is “don’ t care.”
CCEN Bit 8	**Contrast Control Enable**—This bit enables or disables the contrast control function.	0 = Contrast control is off 1 = Contrast control is on
PW[7:0] Bits 7–0	**Pulse Width**—This field controls the pulse width of the built-in pulse-width modulator, which controls the contrast of the LCD screen. **Note:** To drive a Sharp TFT 320 × 240 HR panel, PW[7:0] is reset to 0.	See description

10.3.15 Refresh Mode Control Register

The refresh mode control register controls refresh characteristics.

RMCR Refresh Mode Control Register **0x(FF)FE0826**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
											ROTBUR	ENL	ROT[1:0]		LCDCEN	REFON
TYPE											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 10-24. Refresh Mode Control Register Settings

Name	Description	Setting
Reserved Bits 15–6	Reserved	These bits are reserved and read 0.
ROTBUR Bit 5	**Rotation Burst Mode**—This bit is used to enable or disable the burst access on fetching display data from embedded SRAM. This feature is only active in 16 bpp rotation modes.	0 = Rotation burst mode disabled 1 = Rotation burst mode enabled
ENL Bit 4	**Picture Enlargement Mode**—This bit is used to enable or disable the picture enlargement mode (x2) in all bpp modes.	0 = Picture enlargement mode disabled 1 = Picture enlargement mode enabled

Table 10-24. Refresh Mode Control Register Settings (Continued)

Name	Description	Setting
ROT[1:0] Bits 3–2	**Picture Rotation (16 bpp TFT color mode)**—These bits select the picture rotation orientation. This feature is only active in 16 bpp TFT color mode.	00 = Picture rotation disable (original 0-degree orientation) 01 = 90-degree counter-clockwise orientation 10 = 180-degree counter-clockwise orientation 11 = 270-degree counter-clockwise orientation
LCDCEN Bit 1	**LCD Controller Enable**—This bit enables or disables the LCD controller.	0 = Disable LCDC 1 = Enable LCDC
REFON Bit 0	**Self-Refresh Mode**—This bit enables or disables self-refresh mode.	0 = Disable self-refresh mode 1 = Enable self-refresh mode

NOTE:

1. On entering self-refresh mode, the SCLK and LD[7:0] signals stay low. FRM and LP operate normally.
2. Except for the SSA and color palette mapping register, all configuration data must be set correctly before the LCDC is enabled to avoid a malfunction.
3. The SSA must always match the address range of the RAM selected. The SSA can be switched between various types of RAM on the fly.
4. 4. The rotation mode is designed for image data stored on embedded SRAM. When image data is stored on external RAM, there is a limitation on the screen size supported. The maximum screen size supported is 160 × 160.
5. The LCDC must be disabled before changing the rotation attribute.
6. Assume an LCD panel with the same screen size in x and y direction, the following formula defines the new SSA value after rotation:

   Rot[1:0] = 01, SSA [31:1] = SSA_orig[31:1] + xmax -1

   Rot[1:0] = 10, SSA[31:1] = SSA_orig[31:1] +vpw *(ymax-1) + xmax -1

   Rot[1:0] = 11, SSA[31:1] = SSA_orig[31:1] +vpw *(ymax-1)

   xmax in units of pixel; vpw in units of word

   If the screen size is not equal in x and y direction, some area of the screen may become undefined.
7. The LCDC must be disabled before changing the enlargement mode.
8. The xmax and ymax should be reduced by half before entry to enlargement mode and restored to normal value before return to normal mode.

10.3.16 LCD DMA Control Register

The LCD DMA control register is a 16 × 16-bit line buffer in the LCDC. It stores DMA-in data from system memory. The DMA control register controls when triggers occur and stops the DMA burst cycle by examining the amount of data left in the pixel buffer.

LDMACR LCD DMA Control Register **0x(FF)FE0824**

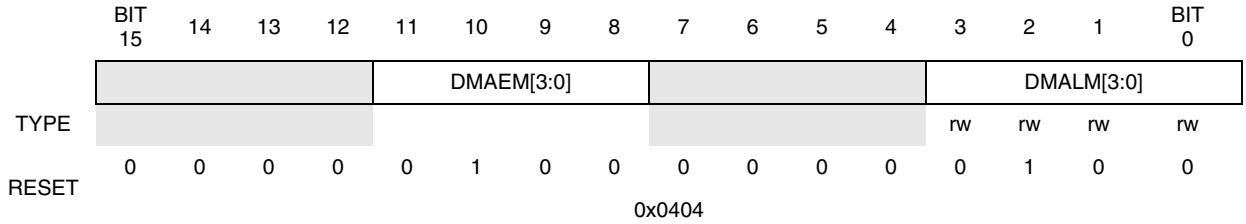


Table 10-25. LCD DMA Control Register Settings

Name	Description	Setting
Reserved Bits 15–12	Reserved	These bits are reserved and read 0.
DMAEM[3:0] Bits 11–8	**DMA End Mark**—This field sets the number of empty locations left in the pixel buffer to stop the DMA burst cycle.	See description
Reserved Bits 7–4	Reserved	These bits are reserved and read 0.
DMALM[3:0] Bits 3–0	**DMA Low Mark**—This field sets the number of data left in the pixel buffer to trigger a DMA request.	See description

10.3.17 LCD Interrupt Configuration Register

The LCD interrupt configuration register is used to configure LCD interrupt conditions.

LICFR LCD Interrupt Configuration Register **0x(FF)FE0828**

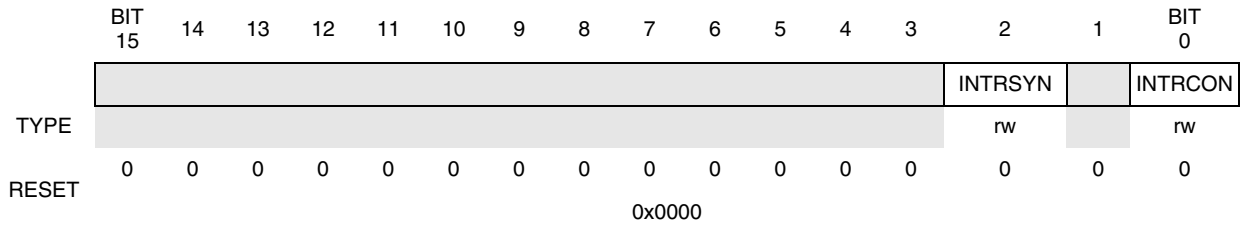


Table 10-26. LCD Interrupt Configuration Register Settings

Name	Description	Setting
Reserved Bits 15–3	Reserved	These bits are reserved and read 0.
INTRSYN Bit 2	**Interrupt Source**—This bit determines if a interrupt flag is set during last data/first data of frame fetch or on last data/first data of frame output to LCD panel.	0 = The interrupt flag is set on fetching the last data/first data of frame from memory. 1 = The interrupt flag is set on output of last data/first data of frame to LCD panel.
Reserved Bit 1	Reserved	These bits are reserved and read 0.

Table 10-26. LCD Interrupt Configuration Register Settings (Continued)

Name	Description	Setting
INTRCON Bit 0	**Interrupt Condition**—This bit determines if a interrupt is generated at the beginning or the end of frame condition. **Note:** For non TFT panel: The EOF is defined as the negative edge of LP pulse for the last line of data. The BOF is defined as the first pixel of the first line appearing in LD signal. For TFT panel: The EOF is defined as the last pixel of the last line appearing in LD. The BOF is defined as the negative edge of the VSYN.	0 = The interrupt flag is set when the end of frame (EOF) is reached. 1 = The interrupt flag is set when the beginning of frame (BOF) is reached.

NOTE:

There is a latency between fetching the last/first data of the frame to the output to the LCD panel. The user should choose according to their system requirements.

10.3.18 LCD Interrupt Status Register

The LCD interrupt status register indicates interrupt status.

LISR LCD Interrupt Status Register **0x(FF)FE082A**

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

Table 10-27. LCD Interrupt Status Register Settings

Name	Description	Setting
Reserved Bits 15 – 2	Reserved	These bits are reserved and read 0.
EOF Bit 1	**End of Frame**—This bit is set when the end of frame is reached. It is cleared by either reading the status register bit or disabling the LCDC.	0 = No end of frame occurred. 1 = End of frame occurred.
BOF Bit 0	**Beginning of Frame**—This bit is set when the beginning of frame is reached. It is cleared by either reading the status register bit or disabling the LCDC.	0 = No beginning of frame occurred. 1 = Beginning of frame occurred.

10.3.19 Mapping RAM Registers

The mapping RAM is used for mapping a 2- or 4-bit gray code to 1 of a maximum of 16 shades and for mapping a 4- or 8-bit color code to 1 of a maximum of 4096 (512) colors for both passive and active panels. The color RAM (0xFFFE0A00 - 0xFFFE0BFF) contains 256 entries, and each entry is 12 bits wide. The RAM can be accessed with word or half-word transactions. Unimplemented bits are read as zeros. Byte access to the RAM will corrupt its contents.

In 2 (4) bpp gray-scale mode, only the first 4 (16) RAM entries are used. In 4 (8) bpp color mode, the first 16 (256, or all) entries are used. The color RAM is not initialized at reset. With any given panel, only one of the following settings will be valid.

10.3.19.1 One Bit/Pixel Monochrome Mode

In this mode the mapping RAM is not used, since the LCDC directly uses the display data in memory to drive the panel.

10.3.19.2 Four Bits/Pixel Gray-Scale Mode

This register maps a 4-bit code into the gray-scale level for a given pixel code. Gray Palette Code is a 4-bit code that represents the gray-scale level for a given pixel code. The first 16 register locations should be written to define the codes for all the 16 combinations. Refer to Figure 10-16.

	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	GPM			
W												
RST												

GPM = Gray Palette Map

Figure 10-16. Gray-Scale Mode

10.3.19.3 Four Bits/Pixel Passive Matrix Color Mode

This register maps a 4-bit code into a 12-bit color. Because only 4 bits are used to encode the color, a maximum of 16 colors can be selected out of a palette of 4096. The first 16 register locations should be written to define the codes for all 16 combinations. Refer to Figure 10-17.

	11	10	9	8	7	6	5	4	3	2	1	0
R	R (4-bit)				G (4-bit)				B (4-bit)			
W												
RST												

R = Red Component Level
G = Green Component Level
B = Blue Component Level

Figure 10-17. Color Mode for Active or Passive Matrices

10.3.19.4 Eight Bits/Pixel Passive Matrix Color Mode

This register maps an 8-bit code into a 12-bit color. Since 8 bits are used to encode the color, a maximum of 256 colors can be selected out of a palette of 4096. All 256 consecutive register locations should be written to define the codes for all 256 combinations. Refer to Figure 10-17.

10.3.19.5 Four Bits/Pixel Active Matrix Color Mode

This register maps a 4-bit code into a 12-bit color. Since just 4 bits are used to encode the color, a maximum of 16 colors can be selected out of a palette of 4096. Sixteen consecutive register locations should be written to define the codes for all sixteen combinations. Refer to Figure 10-17.

10.3.19.6 Eight Bits/Pixel Active Matrix Color Mode

This register maps an 8-bit code into a 9-bit color. Since 8 bits are used to encode the color, a maximum of 256 colors can be selected out of a palette of 512. All 256 register locations should be written to define the codes for the 256 combinations. Refer to Figure 10-17.

10.3.19.7 Twelve/Sixteen Bits/Pixel Active Matrix Color Mode

In this mode, the mapping RAM is not used, since the LCDC directly uses the display data in memory to drive the panel.

Chapter 11
Analog Signal Processor

The analog signal processing (ASP) module of the MC68SZ328 provides support and conversion capabilities for a variety of analog devices, including analog-to-digital controllers (ADC) for voice processing and pen input. The ASP also includes embedded circuity to support a touch panel.

11.1 Features

The ASP module features:

- A (16-bit resolution, 8-bit accuracy) ADC with 3 inputs for touch panel and low-voltage detect
- A (16-bit resolution, 8-bit accuracy) enhanced ADC for voice processing or as a second ADC
- A 12x16 FIFO for pen ADC sample data, an 8x16 FIFO for enhanced ADC data
- Pen ADC supports auto data sampling at a configured sample rate
- Low power management
- Embedded touch panel circuitry
- Provides falling edge or rising edge pen interrupts to interrupt generator
- Provides data ready and FIFO full interrupts to interrupt generator
- Significantly reduces software overhead by using hardware solution for pen input application

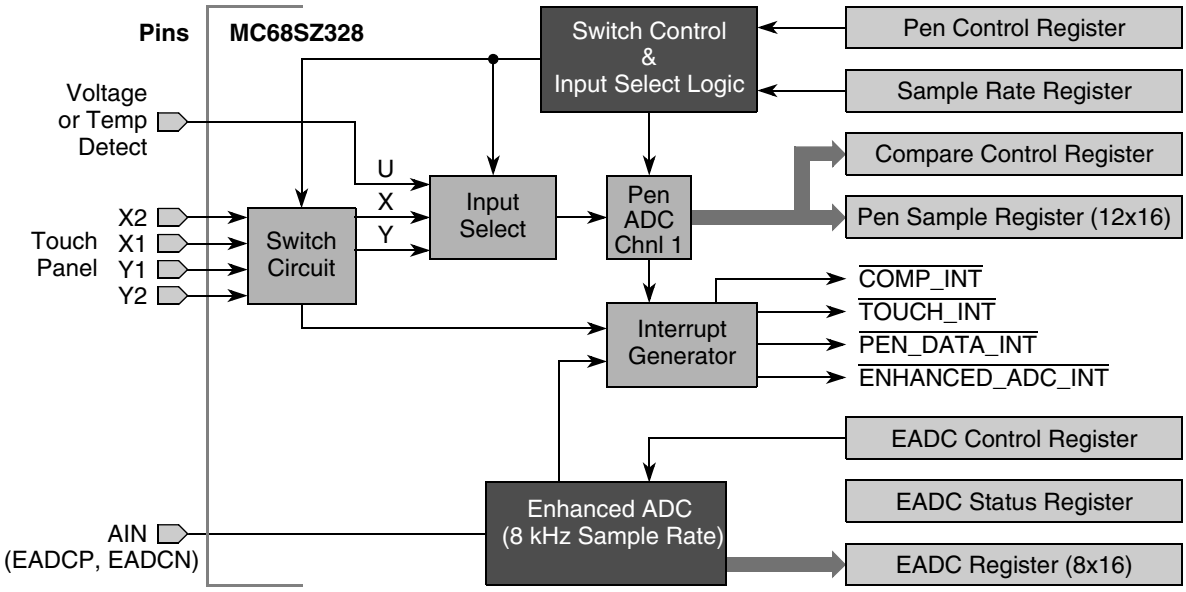


Figure 11-1. ASP System Block Diagram

11.2 ASP Operation

The ASP system block diagram shown in Figure 11-1 illustrates the operation of the individual modules that comprise the ASP. The operation of the individual modules is explained in detail in this section.

11.2.1 Switch Circuit

The touch panel switch circuit block, shown in Figure 11-1, contains 8 switches for the four input signals from the touch panel. The switch circuit toggles the switches on and off in response to input commands from the switch control and input select logic block. The switch circuit outputs two signals, X and Y, that are used as inputs to the input select block. The switch circuit also generates an input to the interrupt generator. The 8 transistor switches that make up the touch panel switch circuit are shown in Figure 11-2.

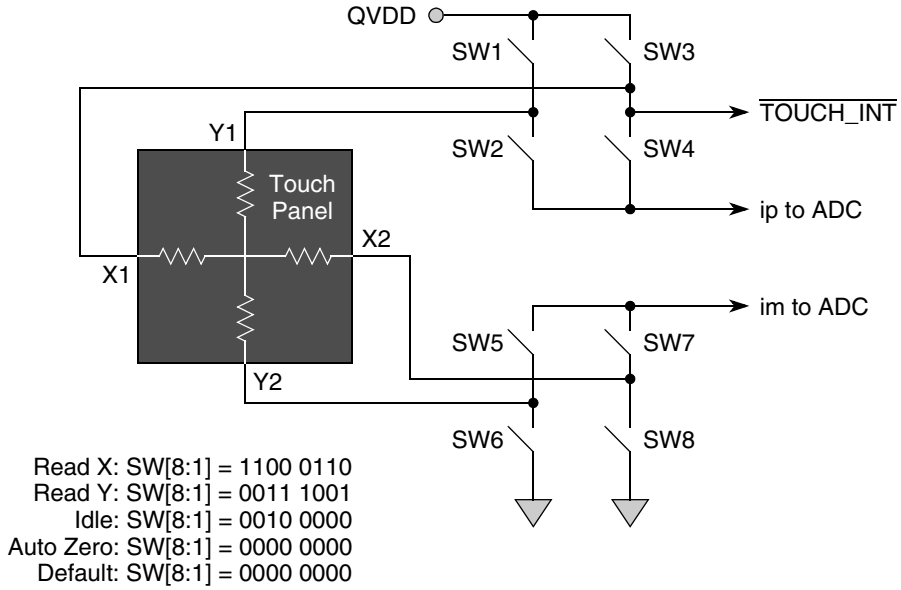


Figure 11-2. Touch Panel Switch Circuit

11.2.2 Switch Control and Input Select Logic

This logic block selects the input signal for Pen ADC sampling. It includes a state machine that automatically generates the proper setting command and selects the target input signal for Pen ADC round robin sampling among the enabled input signals, when the AUTO bit is set.

11.2.3 Input Select

This block selects the three input signals (X, Y, and U) and outputs to Pen ADC for A/D sampling.

NOTE:

The maximum input voltage of the analog input is QVDD (1.9 V nominal).

11.2.4 Enhanced ADC

The ASP includes a 16-bit, 8.29 kHz sample rate enhanced ADC which is normally used for voice processing applications. To minimize software overhead a dedicated ADC FIFO is provided.

11.2.5 Interrupt Generator

This block generates four interrupt signals ($\overline{\text{TOUCH\_INT}}$, $\overline{\text{COMP\_INT}}$, $\overline{\text{PEN\_DATA\_INT}}$ and $\overline{\text{ENHANCED\_ADC\_INT}}$) to an external interrupt handler. The interrupt signals have the following characteristics:

- $\overline{\text{TOUCH\_INT}}$ is asserted when a pen down is detected.
- $\overline{\text{COMP\_INT}}$ is asserted when a pen up is detected.
- $\overline{\text{PEN\_DATA\_INT}}$ is asserted when the following conditions occur:
  - At least one set of valid data is in the Pen Sample FIFO and the PDRE bit of Interrupt Control Register is set. A set of data is defined by the MOD[1:0] bits.
  - The Pen Sample FIFO is full and the PFFE bit of Interrupt Control Register is set.
- $\overline{\text{ENHANCED\_ADC\_INT}}$ is asserted when the following conditions occur:
  - The EADC sample FIFO is full.
  - The EADFFE bit of Interrupt Control Register is set.

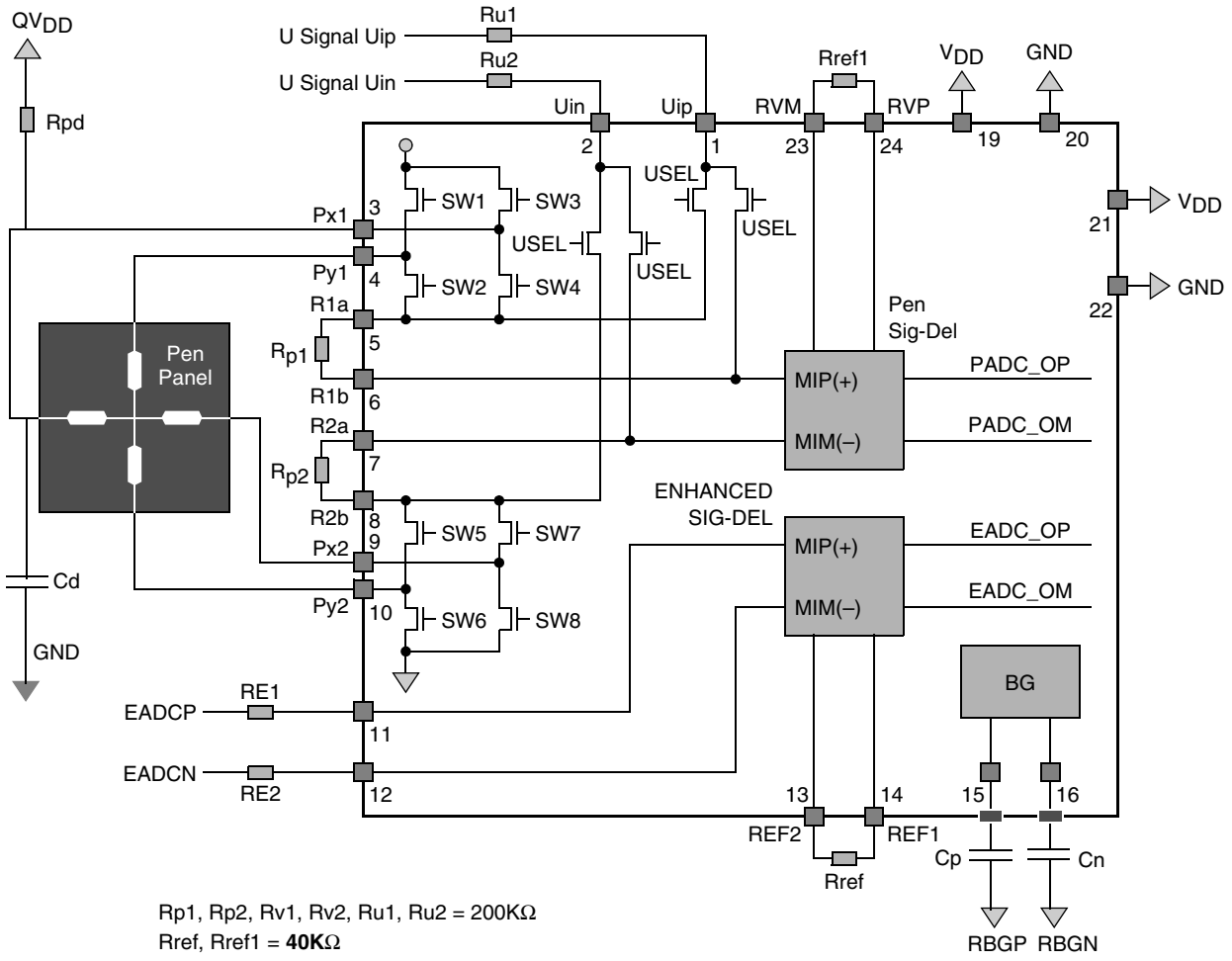


Figure 11-3. Touch Panel Switch Circuits

Table 11-1. Touch Panel Switch Circuit Signal Description

Signal Name	Description
U	External analog input signal.
X1, X2, Y1, Y2	Touch panel interface signals.
Ain	Analog input for enhanced ADC data sampling.
$\overline{\text{TOUCH\_INT}}$	Internal touch detect interrupt signal. This signal can be configured to be a level or an edge detect signal. An external pull-high resistor should be connected to X1. When the panel is touched, X1 is pulled low and the $\overline{\text{TOUCH\_INT}}$ signal is generated.
$\overline{\text{PEN\_DATA\_INT}}$	Internal pen data interrupt signal. If enabled, asserts when data is ready, or when pen sample FIFO is full.
$\overline{\text{ENHANCED\_ADC\_INT}}$	Internal enhanced ADC interrupt signal. If enabled, asserts when EADC FIFO full.
$\overline{\text{COMP\_INT}}$	Internal pen-up interrupt. If enabled, asserts when pen is up.
Rpd	Pen down detect resistor, typical value is 100K ohms.
Cd	Noise decoupling capacitor, typical value is 0.01µF.
RE1, RE2	Enhanced ADC input resistor, typical value is 200K ohms.
Cp, Cn	Bandgap reference capacitor, typical value is 0.01µF.

11.3 Touch Panel Switching Circuit Operation

The touch panel switching circuitry, shown in Figure 11-3, provides all necessary circuitry to support most industry standard touch panels.

11.4 Pen ADC Operation

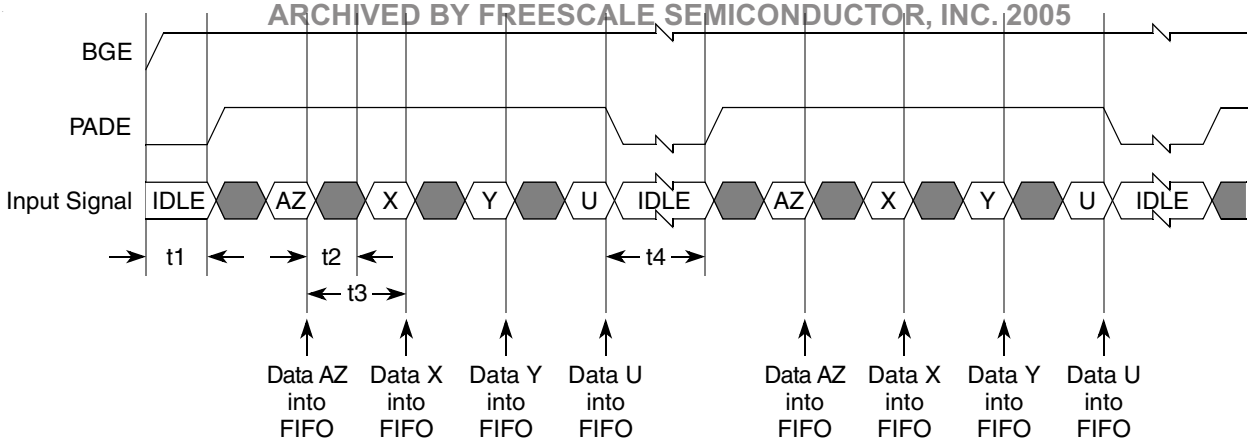
The Pen ADC samples and stores data in the Pen Sample FIFO. The data size is 16-bit with 8-bit accuracy. If data is not read fast enough, data will overflow and old data will be lost. If overflow occurs, the POV status bit is set in the Interrupt/Error Status Register.

This A/D channel can sample data at the rate of up to 6.579 kHz. This sampling speed only occurs when channel U is selected—that is, MODE[1:0] = 11 and ACLK = 8.29 MHz. The Pen ADC samples data and puts the data in the Pen Sample FIFO when it detects a sample command from the Switch Control and Input Select Logic block. Table 11-2 on page 11-5 shows the operation modes supported by the Pen ADC controller.

Table 11-2. Pen ADC Operation

AUTO	PADE	Mode	AZE	Data Format in Pen Sample FIFO	Notes
1	1	00	X	ADC idle	No ADC sampling.
		01	0	x,y,x,y....	Samples pen input only, U input and auto-zero are disabled.
		01	1	AZ,x,y,AZ,x,y....	Samples pen input only, U input is disabled and auto-zero is enabled.
		10	0	x,y,U,x,y,U...	Samples pen input and U input, auto-zero is disabled.
		10	1	AZ,x,y,U,AZ,x,y,U,...	Samples pen input and U input, auto-zero is enabled.
		11	x	U,U,U...	Samples U input only.
x	0	xx	x	No valid data	Pen idle state.
0	1	00	X	ADC idle	No ADC sampling.
		01	0	x	Manual mode, makes one sample when PADE is toggled from 0 to 1 by software.
		01	1	Az, x	
		10	0	y	
		10	1	az, y	
		11	X	U	

When the Pen ADC module is configured for AUTO sampling, the sample operations are carried out continuously. Therefore, the software can read the samples when a Data Ready Interrupt or an ADC FIFO Full interrupt occurs. In manual mode, the AUTO bit is cleared and the Pen ADC samples data only once when the PADE bit is toggled from “0” to “1” by the software. Figure 11-4 shows the sampling timing when AUTO = 1, AZE = 1, and MOD[1:0] = 10.



Notes: t1 = Bandgap warm-up time (when bandgap is ready, BGR bit is set).
t2 = Transistor switching and input selecting setup time, controlled by DSCNT.
t3 = Data setup time, controlled by DSCNT and DMCNT.
t4 = Point-to-point capture idle time, controlled by IDLECNT.

Figure 11-4. Pen Input Sampling Timing

The output data rate is calculated using the equations provided in Table 11-3 on page 11-6. In the equations, DSCNT is the bit for data setup count, IDLECNT is the bit for idle count, and DMCNT is the bit for decimation ratio count. Section 11.9.2, “Pen A/D Sample Rate Control Register,” fully describes DSCNT, DMCNT, and IDLECNT.

If the system clock = 16.58 MHz and PADC_CLK[4:0] = 0x01, then ACLK = 8.29 MHz. The ACLK value can be controlled by PADC_CLK in the ADC clock register. Because the decimation ratio of Filter is 1260, fclk = ACLK/1260. When ACLK = 8.29 MHz, fclk is 6.579 kHz.

NOTE:

The maximum value for the analog clock (ACLK) is 12 MHz. If the value for ACLK is over 12 MHz, the analog circuit will not work efficiently.

NOTE:

For the X and Y channels, the bit for data setup count, DSCNT, must be at least 1 to allow the settling time of the touch panel and the MUX.

Table 11-3. Output Data Rate Equations

Mode	AZE	Equations for Output Data Rate
01	0	(x,y,x,y...) $\frac{fclk}{2\times(DSCNT+DMCNT+1)+IDLECNT}$ (DSCNT>=1)
01	1	(az,x,y,az,x,y...) $\frac{fclk}{3\times(DSCNT+DMCNT+1)+IDLECNT}$ (DSCNT>=1)
10	0	(x,y,u,x,y,u...) $\frac{fclk}{3\times(DSCNT+DMCNT+1)+IDLECNT}$ (DSCNT>=1)

Table 11-3. Output Data Rate Equations (Continued)

Mode	AZE	Equations for Output Data Rate
10	1	(az,x,y,u,az,x,y,u...) $\frac{fclk}{4 \times (DSCNT + DMCNT + 1) + IDLECNT}$ (DSCNT>=1)
11	X	(u,u,u...) $\frac{fclk}{DSCNT + (DMCNT + 1) + IDLECNT}$ (DSCNT>=0)

To illustrate the calculation of the output data rate, a few examples follow:

1. If MODE[1:0] = 01 and AZE = 0, channels X and Y are selected and the auto-zero measurement is disabled. The maximum output data rate for each channel is 1.645 kHz when DSCNT = 1, DMCNT = 0, and IDLECNT = 0.

   NOTE:

   To obtain a 200 Hz output data rate when DSCNT=1 and DMCNT=0, set IDLECNT to 31.

2. If MODE[1:0] = 10 and AZE = 1, all the channels are selected and the auto-zero measurement is performed. The maximum output data rate for each channel is 0.822 kHz when DSCNT=1, DMCNT= 0, and IDLECNT = 0.

   NOTE:

   To obtain a 200Hz output data rate when DSCNT=1 and DMCNT = 7 (the decimation ratio is set to 2) set IDLECNT to 17.

3. If MODE[1:0] = 11, only the U channel will be selected. The bit for data setup count can be set to 0 in this case because there is no need for the settling time of the touch panel and MUX. The maximum output data rate can be up to 6.579 kHz when DSCNT = 0 and IDLECNT = 0.

   NOTE:

   To obtain a 200 Hz output data rate when DSCNT=0 and DMCNT=0, set IDLECNT to 32.

11.5 ASP Pen Up Detection Method

The ASP checks the pen up status after each set of X or Y samples. This is accomplished by using the time slot of the U-channel sampling to check the status of the touch panel. Use this pen-up detection method in Auto mode (XYU mode) and the auto-zero (XYU mode).

During the U-channel sampling time slot of X,Y,U auto-mode and AZ,X,Y,U auto-mode, the configuration of the eight internal switches is changed to SW[8:1] = 0010 0000. This is same as the **IDLE** mode switch configuration, and does not affect U-channel sampling.

When the pen is down, px1 is pulled low which results in a “low” state on the TOUCH_INT signal. When the pen is up, px1 is pulled high, which results in a “high” state on the TOUCH_INT signal. By monitoring the state of px1 (TOUCH_INT) during the U-sampling time, the pen up status (high or low) can be obtained and the interrupt can be generated. Because auto mode is continuously sampled (during every time slot of U-channel sampling) pen up status is generated on the TOUCH_INT signal.

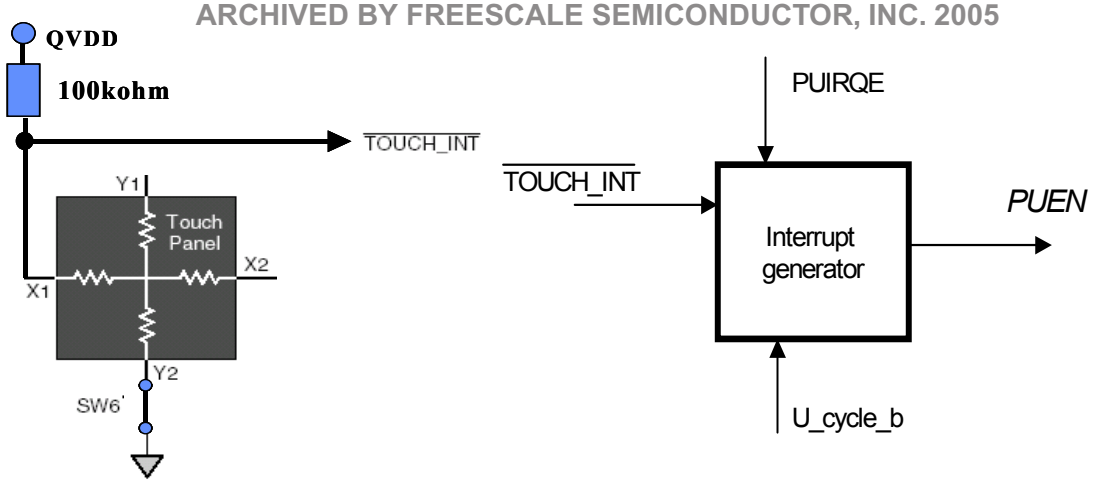


Figure 11-5. TOUCH_INT Pen Up Detection Circuit

NOTE:

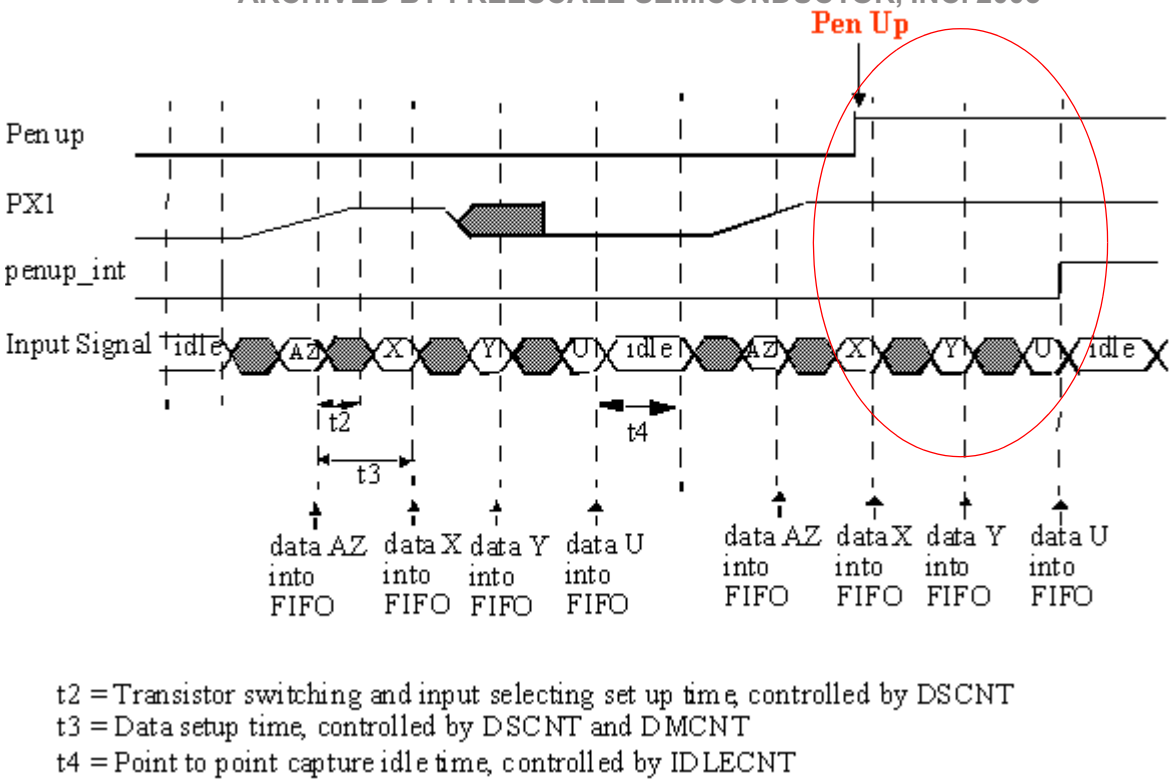
Both pen touch and pen up interrupts use the TOUCH_INT interrupt signal.

11.5.1 Discarding Sample Data Before Pen Up Event

There are two potential timing cases for pen up detection which affect the last sampling data before pen-up interrupt.

1. Pen Up at the beginning of the current U sampling period.
2. Pen Up at the beginning of the next U sampling period.

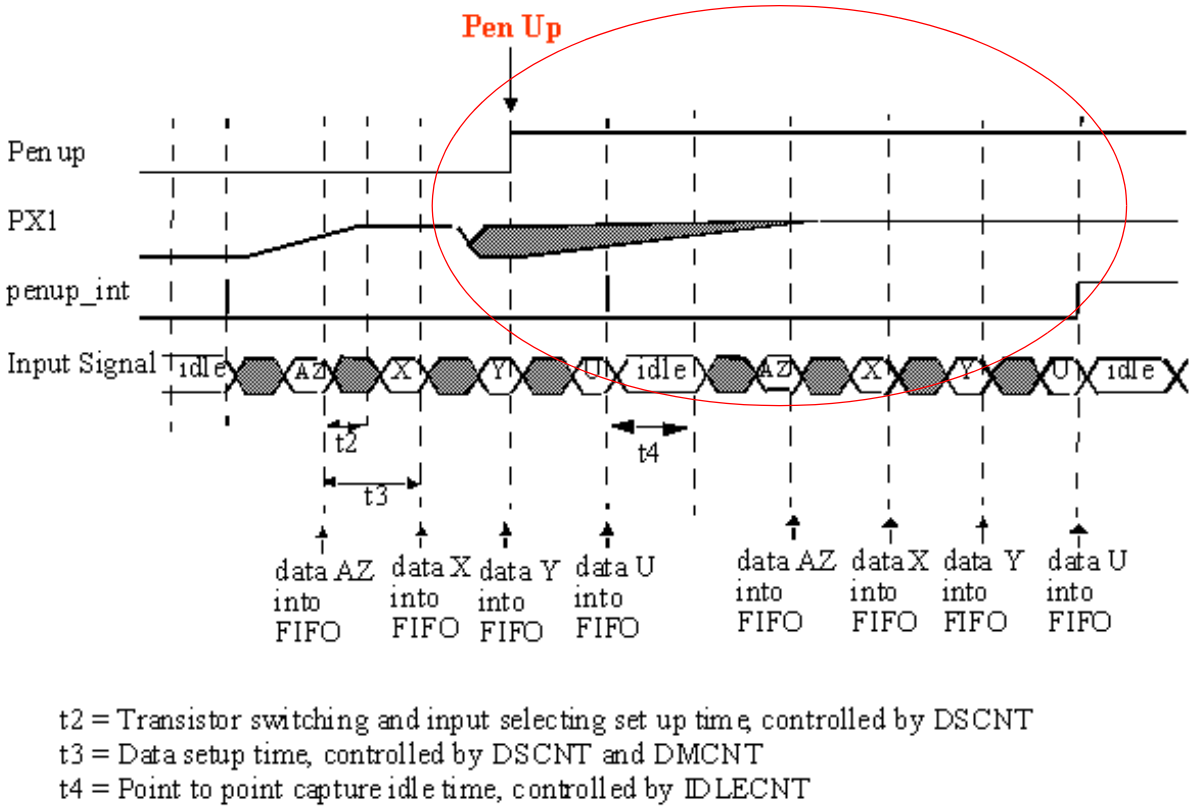
Figure 11-6 on page 11-9 and Figure 11-7 on page 11-10 shows each of these two cases.



t2 = Transistor switching and input selecting set up time, controlled by DSCNT
t3 = Data setup time, controlled by DSCNT and DMCNT
t4 = Point to point capture idle time, controlled by IDLECNT

Figure 11-6. Pen Up (Beginning of Current U Sampling Period)

Pen up requires approximately 5ms for the px1 pin to be completely pulled up. This time period is longer than the U-sampling time. Should pen up occur at the beginning of a U-sampling period, the 1 ms time frame is not long enough for the pen up status to be set.



t2 = Transistor switching and input selecting set up time, controlled by DSCNT
t3 = Data setup time, controlled by DSCNT and DMCNT
t4 = Point to point capture idle time, controlled by IDLECNT

Figure 11-7. Pen Up (Beginning of Next U Sampling Period)

In this case, once the pen is up, the px1 gradually rises to QVDD (this requires approximately 5 ms) and stays at QVDD until the next pen down. This allows pen up status to be set at the next U-sampling period.

As a result, pen up detection can be done within 2 sets of X,Y,U sampling or AZ,X,Y,U sampling after the pen is up. To ensure that the X,Y samples are valid, discard the last set of data before the pen up event.

11.6 ASP Auto-Zero Switch Configuration

Manual operation has two modes:

1. AZ,X mode
2. AZ,Y mode

The switch configuration for reading AZ,X, and AZ,Y is controlled by manually setting the ASP Control Register as follows.

AZ: SW{8:1] = 0000 0000

X: SW[8:1] = 1100 0110

Y: SW[8:1] = 0011 1001

Because the operation of AZ,X and AZ,Y modes only allows users to set the switches once before taking samples, either the AZ, X (or Y) sample is discarded depending on the switch configuration.

In the AZ, X and AZ, Y modes (manual), the switch configuration during the AZ read is automatically set as follows:

SW{8:1] = 0000 0000

Users do not need to manually configure it in the ASP Control Register. Users only need to configure read X, read Y, or read either settings.

11.7 ASP Auto-Calibration Configuration

An auto-calibration function exists in the AZ,X,Y mode when in auto mode. The switch configuration for reading the AZ, X, and Y for auto calibration is as follows:

AZ: SW{8:1] = 0000 0000

X: SW[8:1] = 1100 1100

Y: SW[8:1] = 0011 0011

The sample X,Y value received from auto-calibration will be in the range of the X and Y for that sample. The value is the maximum value of the sample received from the ADC. The AZ value is the minimum value received from the ADC. The range of values from the ADC is defined based on the AZ value and the range values of X and Y.

11.8 Enhanced ADC Operation

The enhanced ADC is used for voice processing or as a second ADC. The controllers operate at 8.29 kHz when the system clock is 16.58 MHz, ADC_FIR_DEC is 4, and ADC_COMB_DEC is 250.

The EADC samples and stores data to the EADC FIFO. The data size is 16-bit with only 8-bit accuracy. User reads to this Data register are 16-bit access. If data is not read fast enough, it will overflow and old data is lost. If overflow occurs, the EOV status bit is set in the Interrupt Status/Error Register.

11.9 Programming Model

11.9.1 ASP Control Register

This register controls the function of Pen ADC and the pen and touch pad interface modules.

ASP_ACNTLCR ASP Control Register **0xFFFE0210**

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							CLKE			SWRST	U_SEL	AZ_SEL		NM		ACE
TYPE							rw			rw	rw	rw		rw		rw
RESET	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0
									0x0304							

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AZE	AUTO	MOD[1:0]		SW8	SW7	SW6	SW5	SW4	SW3	SW2	SW1		EADE	PADE	BGE
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	0
									0x0000							

Table 11-4. ASP Control Register Description

Name	Description	Setting
Reserved Bits 31–26	Reserved	These bits are reserved and should be set to 0.
CLKE Bit 25	**CLKE**—For the software configuration for the low power mode	0 = All clock input into the ASP is gated, ASP will not work except some register read/write. 1 = ASP work state, clock is enabled.
Reserved Bits 24–23	Reserved	This bit is reserved and should be set to 0.
SWRST Bit 22	**Soft Reset**—Writing 1 to this bit resets entire ASP module.	0 = Not reset 1 = Reset
U_SEL Bit 21	**U Channel Select**—This bit selects the external resistor used for U-channel measurement.	
AZ_SEL Bit 20	**AZ_SEL**—This bit selects the position of auto-zero measurement.	
Reserved Bit 19	Reserved	
NM Bit 18	**NM** — Leave always set to 1 (default)	
Reserved Bit 17	Reserved	

Table 11-4. ASP Control Register Description (Continued)

Name	Description	Setting
ACE Bit 16	**Auto Calibration Enable**—This bit enables /disables the auto-calibration in auto-ZXY mode. When this bit is set, the switch settings for X / Y are changed from C6 / 39, to CC / 33 respectively.	0 = Disable 1 = Enable
AZE Bit 15	**Auto-Zero Enable**—This bit enables the auto-zero measurement. When this bit is set, the auto-zero measurement occurs before every pen input measurement.	0 = Disable 1 = Enable
AUTO Bit 14	**Auto Sampling**—This bit enables the Pen ADC auto sampling function. When this bit is set, transistor switches are set properly when measuring selected analog inputs X, Y, or U. After all the selected signals are sampled, this process repeats when the IDLE counter reaches zero. The IDLE counter is reloaded when it reaches zero in AUTO mode.	0 = Disable 1 = Enable
MOD[1:0] Bits 13–12	**Mode**—This field selects the analog input signals for A/D sampling. See Table 11-2 for more information.	00 = No input signal selected 01 = selects X then Y, or only X depending on AUTO bit 10 = selects X, Y then U, or only Y depending on AUTO bit 11 = selects only U
SW8-SW1 Bits 11–4	**Switches**—These bits control the transistor switches when the AUTO function is off or when Pen ADC is off of the touch panel idle state.	0 = Transistor off 1 = Transistor on
Reserved Bit 3	Reserved	This bit is reserved and should be set to 0.
EADE Bit 2	**Enhanced A/D Enable**—This bit enables or disables the Enhanced A/D Controller. When this bit is set, the A/D starts sampling and pushes data to the RxD FIFO at the selected data rate.	0 = Disable 1 = Enable
PADE Bit 1	**Pen ADC Enable**—This bit enables or disables Pen ADC Controller. This bit should be set after all other control bits are set. When this bit is set, the A/D starts sampling. This bit should be cleared by the software when the touch penal is in idle state for power saving—that is, waiting for pen interrupt. When this bit is cleared, the Pen Sample FIFO is flushed.	0 = Disable 1 = Enable
BGE Bit 0	**Band Gap Enable**—This bit enables or disables the bandgap. When the bandgap is stable, the BGR bit in the interrupt/error status register is set. It is normally disabled only when system is in sleep mode for power saving.	0 = Disable 1 = Enable

11.9.2 Pen A/D Sample Rate Control Register

This register controls the sampling rate for touch pen input. Each application may require different pen input sampling rates. The maximum A/D sampling rate is limited by the A/D design and the input signal data setup time. The design supports up to 200 Hz update rate for each input signal. The user should set the optimal data setup time and idle time for maximum power saving.

ASP_PSMPLRG Pen A/D Sample Rate Control Register **0xFFFE0214**

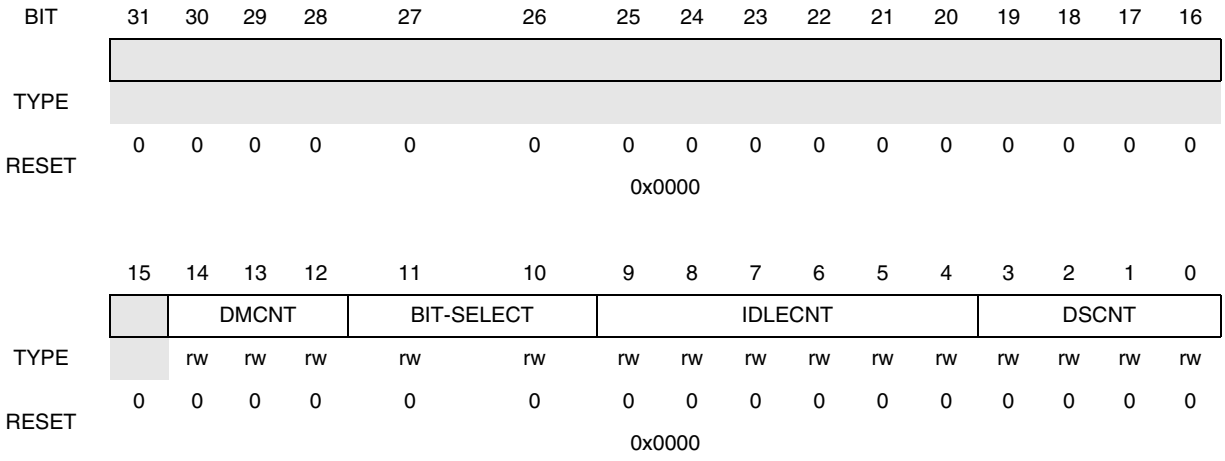


Table 11-5. Pen A/D Sample Rate Control Register Description

Name	Description	Setting
Reserved Bits 31–15	Reserved	These bits are reserved and should be set to 0.
DMCNT[2:0] Bits 14–12	**Decimation Ratio Count**—These bits control the decimation ratio of second-stage FIR. The input clock to this counter is FCLK.	000 = decimation ratio is 1 001 = decimation ratio is 2 … 111 = decimation ratio is 8
BIT-SELECT[1:0] Bits 11–10	**Bit Select**—This field controls which bits are selected from the FIR output.	00 = selecting 16 bits starting from 1st MSB of FIR output 01 = selecting 16 bits starting from 2nd MSB of FIR output 10 = selecting 16 bits starting from 3rd MSB of FIR output 11 = selecting 16 bits starting from 4th MSB of FIR output
IDLECNT[5:0] Bits 9–4	**Idle Count**—These bits control the number of clocks between the last capture and the first capture of two pen input points. Input clock to this counter is FCLK.	$00 = 0 clock $01 = clock … $3F = 63 clocks
DSCNT[3:0] Bits 3–0	**Data Setup Count**—This field controls the input signal data set up time after the transistor switching circuit and input select are settled. Input clock to this counter is FCLK.	0000 = 0 clock 0000 = 1 clocks … 1111 = 15 clocks

11.9.3 Compare Control Register

This register is used to detect the out-of-range samples of the selected input. A typical application of this function is using it as an alarm function when the external input is beyond a specific range.

ASPCMPCNTL Compare Control Register **0xFFFE0230**

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													INT	CC	INSEL[1:0]	
TYPE													rw	rw	rw	
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 11-6. Compare Control Register Description

Name	Description	Setting
Reserved Bits 31–20	Reserved.	These bits are reserved and should be set to 0.
INT Bit 19	**Interrupt**—This interrupt status bit is set when a trigger event is detected. This bit, as well as the interrupt, clears when the software writes a “1” to it.	$0 = No trigger event is detected $1 = A trigger event is detected
CC Bit 18	**Compare Control**—This bit controls the compare operation.	0 = Trigger when compare value is greater than sample 1 - Trigger when sample is greater than compare value
INSEL[1:0] Bits 17–16	**Input Select**—The Input Select field selects the input samples for the compare operation.	00 = No compare, Interrupt disabled 01 = Channel X 10 = Channel Y 11 = Channel U
COMPARE VALUE[15:0] Bits 15–0	**Compare Value**—This field contains the value to compare with the select sample.	See description

11.9.4 Interrupt Control Register

The Interrupt Control Register enables and controls each interrupt function. These interrupts are grouped into three outputs to the system interrupt controller, TOUCH_INT, PEN_DATA_INT, and VOICE_ADC_INT.

ASP_ICNTLR Interrupt Control Register **0xFFFE0218**

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE																
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
									EADMAE	POL	EDGE	PIRQE	PUIRQE	EADFFE	PFFE	PDRE
TYPE									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 11-7. Interrupt Control Register Description

Name	Description	Setting
Reserved Bits 31–8	Reserved	These bits are reserved and should be set to 0.
EADMAE Bit 7	**EADC DMA Enable**—This bit enables or disables the EADC FIFO-FULL DMA request.	0 = Disable 1 = Enable
POL Bit 6	**Pen Interrupt Polarity**—This bit selects the polarity of TOUCH_INT input signal for interrupt trigger.	0 = Active low, or falling edge 1 = Active high, or rising edge
EDGE Bit 5	**EDGE Enable**—This bit selects edge or level trigger on TOUCH_INT input signal.	0 = Level 1 = Edge
PIRQE Bit 4	**Pen Interrupt Enable**—This bit enables or disables generating pen interrupt signal, TOUCH_INT.	0 = Disable 1 = Enable
PUIRQE Bit 3	**Pen Up Interrupt Enable**—This bit enables /disables generating pen-up interrupt signal	0 = Disable 1 = Enable
EADFFE Bit 2	**EADC FIFO Full Enable**—This bit enables or disables EADC FIFO Full interrupt.	0 = Disable 1 = Enable
PFFE Bit 1	**Pen FIFO Full Interrupt Enable**—This bit enables or disables pen sample FIFO full interrupt.	0 = Disable 1 = Enable
PDRE Bit 0	**Pen Data Ready Interrupt Enable**—This bit enables or disables pen sample ready interrupt.	0 = Disable 1 = Enable

11.9.5 Interrupt/Error Status Register

The Interrupt/Error Status Register shows the source of interrupts when there is an interrupt event. Each interrupt status bit in this register can be cleared by either writing a 1 to it or reading/writing to the associated data register, depending on the nature of interrupt.

ASP_ISTATR Interrupt/Error Status Register **0xFFFE021C**

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE																
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
									BGR	EOV	POV	PEN	PUEN	EADFF	PFF	PDR
TYPE									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 11-8. Interrupt/Error Status Register Description

Name	Description	Settings
Reserved Bits 31–8	Reserved.	These bits are reserved and should be set to 0.
BGR Bit 7	**Bandgap Ready**—This status bit shows if the bandgap is ready. All the analog circuits will work only after the bandgap voltage reference is ready.	0 = Bandgap is not ready 1 = Bandgap is ready
EOV Bit 6	**Enhanced ADC Sample Overflow**—This status bit shows if there has been Enhanced ADC sample data overflow in the Enhanced ADC RxD FIFO. It does not generate an interrupt on overflow. It is recommended that the software clears this bit at the beginning of Enhanced ADC sampling and check for an error at the end sampling process. This bit is cleared by writing a 1 to it.	0 = No overflow 1 = FIFO has been overflow
POV Bit 5	**Pen Overflow**—This status bit shows if there has been pen sample data overflow in the Pen Sample FIFO. It will not generate interrupt on overflow. It is recommended that software will clear this bit at the beginning of pen capture and check for error at the end of a stroke. This Bit is clear by writing a 1 to it.	0 = No overflow 1 = FIFO has been overflow
PEN Bit 4	**Pen Interrupt**—When this bit is set, there is a pen touch interrupt pending. Writing a 1 to this bit clears pen interrupt.	0 = No PEN interrupt is pending 1 = PEN interrupt is pending
PUEN Bit 3	**Pen Up Interrupt Status**—When this bit is set, there is a pen up interrupt pending. Writing 1 to this bit clears the pen up interrupt.	0 = No PEN UP interrupt is pending 1 = PEN UP interrupt is pending

Table 11-8. Interrupt/Error Status Register Description (Continued)

Name	Description	Settings
EADFF Bit 2	**EADC FIFO Full**—When this bit is set, it shows that the EADC FIFO is full. Reading the data in EADC Register automatically clears this bit.	0 = RxD FIFO is not full 1 = RxD FIFO is full
PFF Bit 1	**Pen Sample FIFO Full**—When this bit is set, it shows that the pen sample FIFO is full. Reading the data in the Pen sample register clears this bit automatically.	0 = Pen sample FIFO is not full 1 = Pen sample FIFO is full
PDR Bit 0	**Pen Data Ready**—When this bit is set, at least one set of pen data in the pen sample FIFO is ready. Reading the data in the pen sample register clears this bit automatically. A data set is defined by the MOD bits in the control register.	0 = No valid data in Pen Sample FIFO 1 = At least one set of valid data in the Pen sample FIFO

11.9.6 ASP FIFO Pointer Register

This register indicates the FIFO read/write level.

ASP_FIFO_PTR ASP FIFO Pointer Register **0xFFFE0236**

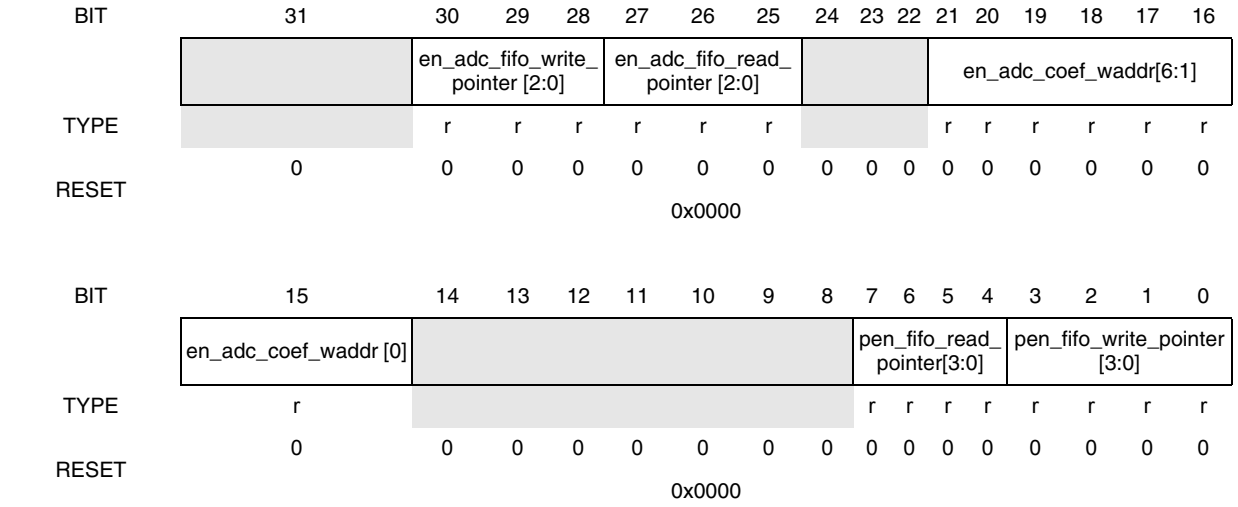


Table 11-9. ASP FIFO Pointer Register Description

Name	Description	Settings
Reserved Bit 31	Reserved	This bit is reserved and should be set to 0.
EN_ADC_FIFO_WRITE_POINTER[2:0] Bits 30–28	**EN_ADC_FIFO_WRITE_POINTER**—Holds the write pointer of enhanced ADC FIFO	
EN_ADC_FIFO_READ_POINTER[2:0] Bits 27–25	**EN_ADC_FIFO_READ_POINTER**—Holds the read pointer of enhanced ADC FIFO	
Reserved Bits 24–22	Reserved	These bits are reserved and should be set to 0.

Table 11-9. ASP FIFO Pointer Register Description

Name	Description	Settings
EN_ADC_COEF_WADDR[6:0] Bits 21–15	**EN_ADC_COEF_WADDR**—Holds the write pointer of VADC coefficient RAM.	
Reserved Bits 14–8	Reserved	These bits are reserved and should be set to 0.
PADC_FIFO_READ_POINTER[3:0] Bits 7–4	**PADC_FIFO_READ_POINTER**—Hold s the write pointer of PADC FIFO	
PADC_FIFO_WRITE_POINTER[3:0] Bits 3–0	**PADC_FIFO_WRITE_POINTER**—Hol ds the write pointer of PADC FIFO	

11.9.7 Pen Sample FIFO Register

This 12x16 FIFO Pen Sample register holds the sample data after Pen ADC sampling. The data structure is controlled by the MOD[1:0] bits of control register.

ASP_PADFIFO Pen Sample FIFO Register **0xFFFE0200**

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

11.9.8 Enhanced ADC Register

This 8x16 FIFO Register stores input data sampled by the enhanced ADC at the assigned sample rate.

ASP_EADFIFO Enhanced ADC Register **0xFFFE0204**

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

11.9.9 Enhanced ADC Control Register

This register controls the decimation ratio and gain of both the comb filter and the FIR filter in ADC.

ASP_EADGAIN Enhanced ADC Control Register **0xFFFE0220**

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE																
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
			ADC_FIR_GAIN					ADC_FIR_DEC		ADC_COMB_GAIN					ADC_COMB_DEC	
TYPE			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 11-10. Enhanced ADC Control Register Description

Name	Description	Setting
Reserved Bits 31–14	Reserved.	These bits are reserved and should be set to 0.

Table 11-10. Enhanced ADC Control Register Description (Continued)

Name	Description	Setting
ADC_FIR_GAIN[4:0] Bits 13–9	**ADC Filter Gain**—This field controls the gain of the FIR filter.	00 = selecting 16 bits from 1st MSB of the output of FIR filter 01 = selecting 16 bits from 2nd MSB of the output of FIR filter … 1F = selecting 16 bits from 32nd MSB of the output of FIR filter
ADC_FIR_DEC[1:0] Bits 8–7	**Decimation Ratio for FIR Filters**—This field controls the decimation ratio of the FIR filter.	00 = decimation ratio is 1 01 = decimation ratio is 2 10 = decimation ratio is 3 11 = decimation ratio is 4
ADC_COMB_GAIN[4:0] Bits 6–2	**ADC Comb Filter Gain**—This field controls the gain of the comb filter.	00 = selecting 16 bits from 1st MSB of the output of comb filter 01 = selecting 16 bits from 2nd MSB of the output of comb filter … 1F = selecting 16 bits from 32nd MSB of the output of comb filter
ADC_COMB_DEC[1:0] Bits 1–0	**Decimation Ratio For Comb Filters**—These bits control the decimation ratio of the comb filter.	00 = decimation ratio is 375 01 = decimation ratio is 125 10 = decimation ratio is 250 11 = decimation ratio is 375

11.9.10 Clock Divide Register

This register is used to control the divider ratio to generate the clock from the system clock for pen ADC and enhanced ADC.

ASP_CLKDIVO Clock Divide Register **0xFFFE022C**

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE																
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
							EADC_CLK					PADC_CLK				
TYPE							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 11-11. Clock Divide Register Description

Name	Description	Setting
Reserved Bits 31–10	Reserved	These bits are reserved and should be set to 0.
EADC_CLK[4:0] Bits 9–5	**EADC Clock**—This field selects the divide ratio used to generate the clock used in the enhanced ADC.	00 = clock disabled 01 = divide ratio is 2 … 1F = divide ratio is 32
PADC_CLK[4:0] Bits 4–0	**Pen ADC Clock**—This field selects the divide ratio to generate the clock for use in the pen ADC.	00 = clock disabled 01 = divide ratio is 2 … 1F = divide ratio is 32

11.9.11 Enhanced ADC FIR Coefficients RAM Register

This 128x16 RAM stores the FIR coefficients used in the enhanced ADC.

ASP_EADCOEF Enhanced ADC FIR Coefficients RAM Register **0xFFFE020C**

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

Chapter 12
General-Purpose Timers

The MC68SZ328 contains two identical general-purpose (GP) 16-bit timers with programmable prescalers, and compare and capture registers. Each timer counter value can be captured using an external event and can be configured to trigger a capture event on the leading or trailing edges of an input pulse. The timer also can generate an interrupt when the timer reaches a programmed value. Each timer has an 8-bit prescaler providing a programmable clock frequency derived from SYSCLK.

Figure 12-1 is the general purpose timer block diagram.

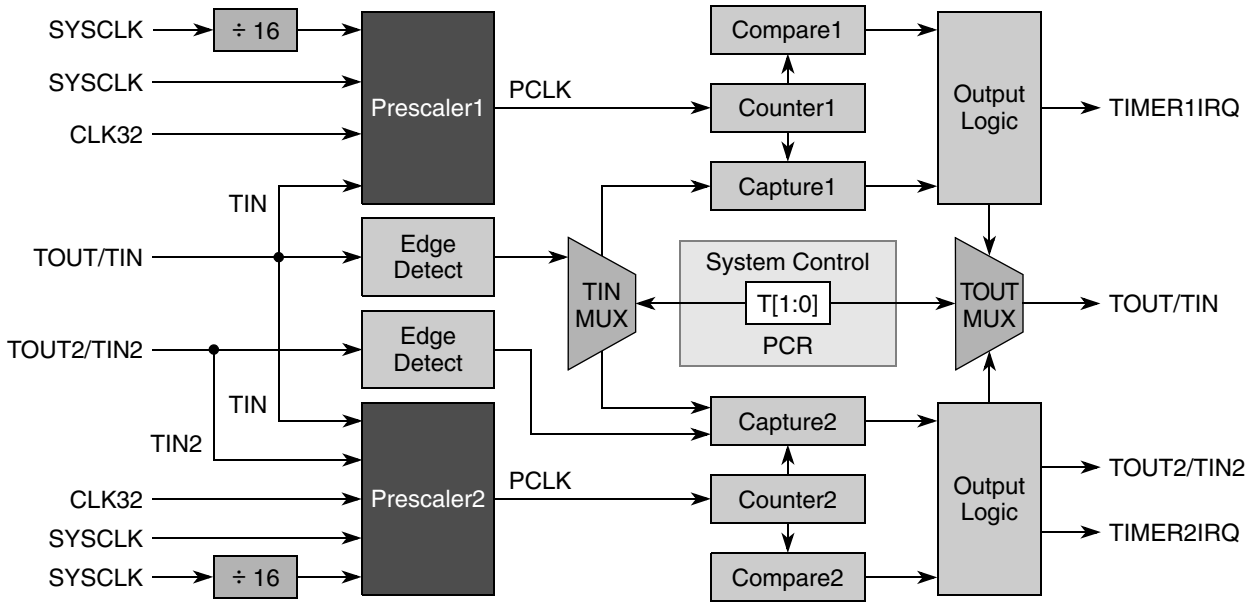


Figure 12-1. General-Purpose Timer Block Diagram

The general-purpose timers have the following features:

- 15 ns resolution at 66.32 MHz
- Programmable sources for the clock input, including external clock
- Input capture capability with programmable trigger edge for interval measurement
- Output compare with programmable mode
- Free-run and restart modes
- Capability to be cascaded together to operate as a single 32-bit timer
- Timer input/output pin for event notification

12.1 Operation

The following sections describe the operation of the modules in the general purpose timers of the MC68SZ328.

12.2 Timer Multiplexing

The GPIO pin PB6 is multiplexed with the TIN (Timer 1 Input or Timer 2 Input) and TOUT (Timer 1 Output or Timer 2 Output) dedicated functions. See section 6.2.2 "Peripheral Control Register", on page 3 for more details. GPIO pin PF1 is multiplexed with TIN2 (Timer 2 Input) and TOUT2 (Timer 2 Output) dedicated functions. This enhanced design, allows increased applications support such as remote control allowing generation of AC code. The General Purpose Timer Multiplex block diagram, shown in Figure 12-2, shows a logic block diagram of the multiplexer. This diagram assumes the GPIO is set for internal dedicated function on pins PB6 and PF1.

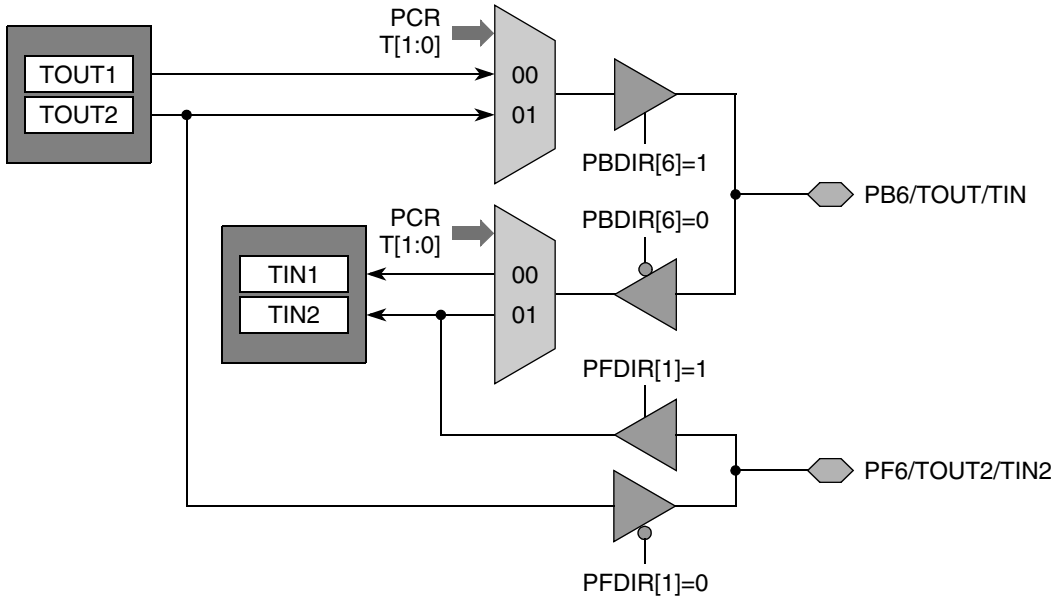


Figure 12-2. General Purpose Timer Multiplexing

12.3 Clock Selection

The clock that feeds the prescaler can be selected from one of three sources: the main clock (divided by 1 or by 16), the timer I/O pin (TIO), or the 32 kHz clock (CLK32). The clock input source is determined by the CLKSOURCE field of the timer control register. The timer prescaler register is used to select the divide ratio of the input clock that drives the main counter. The prescaler can divide the input clock by a value between 1 and 256. When CLK32 is selected as the clock source, the timer operates even while the PLL is in sleep mode (at that time, the SYSCLK from the PLL is not available).

NOTE:

Ensure that the timer is disabled by clearing the TEN bit in the TCTLx register before changing either the clock source or the prescaler setting.

12.3.1 Timer Configuration

The timer can be configured for free-run or restart modes by programming the FRR bit of the timer control register. Restart mode is useful to generate periodic events or audio tones in conjunction with the timer output signals. In restart mode, the following events occur when the compare value is reached:

1. The counter resets to 0x0000.
2. The COMP bit of the timer status register is set.
3. An interrupt is issued if the IRQEN bit of the timer control register is set.
4. The counter resumes counting.

In free-run mode, the compare function operates exactly as it does in restart mode, but the counter continues counting without resetting to 0x0000. When 0xFFFF is reached, the counter rolls over to 0x0000 and keeps counting.

Each timer has a 16-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 timer control register. Pulses that produce the capture edge can be as short as 30 ns. The minimum time between pulses is two PCLK periods.

When a capture or reference event occurs, the corresponding status bit is set in the timer status register, and an interrupt is posted if the capture function is enabled or if the IRQEN bit of the timer control register is set. The timer is disabled at reset.

12.3.2 Cascaded Timers

Both timers can be cascaded together to create a 32-bit counter. The cascade configuration is controlled by the T[1:0] field of the PCR. See Section 6.2.2, "Peripheral Control Register," on page 6-3 for more details.

Table 12-1 shows the two possible configurations of cascaded timers. When T[1:0] = 0x10, Timer 1 becomes the MSW and Timer 2 is the LSW. If the direction of the pin is in (DIR6 = 0), the TIN signal is applied to Timer 2. If the direction is out (DIR6 = 1), the TOUT is connected to Timer 1.

When T[1:0] = 0x11, Timer 2 becomes the MSW and Timer 1 is the LSW. If the direction of the pin is in (DIR6 = 0), the TIN signal is applied to Timer 1. If the direction is out (DIR6 = 1), the TOUT is connected to Timer 2.

Table 12-1. Cascaded Timer Settings

T[1:0] PCR	MSW	LSW	TIN To	TOUT From
10	Timer 1	Timer 2	Timer 2	Timer 1
11	Timer 2	Timer 1	Timer 1	Timer 2

12.3.3 Compare and Capture Using Cascaded Timers

When the timers are cascaded, the associated compare and capture registers are not cascaded. The flow diagram in Figure 12-3 suggests one method for 32-bit compares using a cascaded timer. Captures also can be accomplished using the CAPT status bit instead of the COMP status bit.

After the compare to Timers 1 and 2 is written, the COMP or CAPT status bit of the MSW is checked. When the MSW status bit sets, check the status bit of the LSW. If it is not set, loop until it does set.

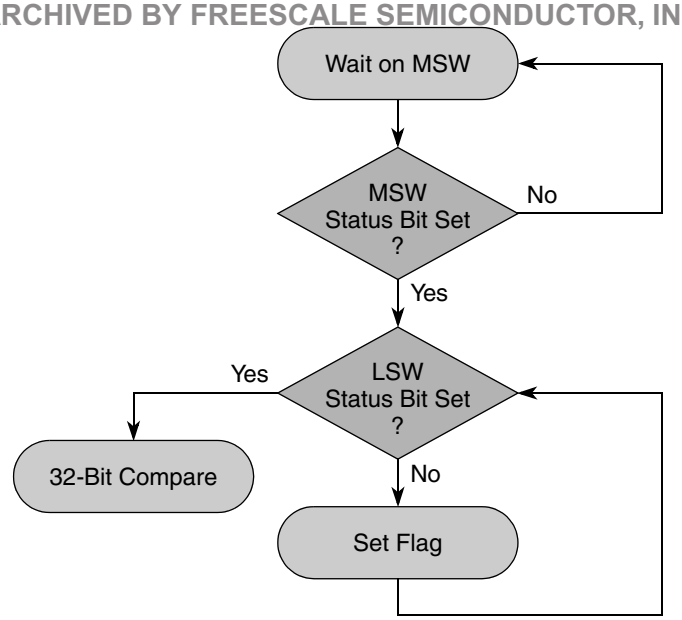


Figure 12-3. Compare Routine for 32-Bit Cascaded Timers

12.4 Programming Model

The base address of Timer 1 is $0x(FF)FFF600, and the base address of Timer 2 is $0x(FF)FFF610. Because the two timers are identical, the register descriptions and the associated tables describing the register settings apply to both registers.

12.4.1 Timer Control Registers

Each timer control register (TCTLx) controls the overall operation of its corresponding general-purpose timer. The settings for the registers are described in Table 12-2. The TCTL registers:

- Select the free-running or restart mode after a compare event
- Select the capture trigger event
- Control the output compare mode
- Enable the compare event interrupt
- Select the prescaler clock source
- Enable and disable the GP timer

TCTL1 Timer Control Register 1 0x(FF)FFF600

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
								FRR	CAP		OM	IRQEN	CLKSOURCE			TEN
TYPE								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															

TCTL2 Timer Control Register 2 0x(FF)FFF610

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
								FRR	CAP		OM	IRQEN	CLKSOURCE			TEN
TYPE								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 12-2. Timer Control Register Description

Name	Description	Setting
Reserved Bits 15–9	Reserved	These bits are reserved and should be set to 0.
FRR Bit 8	**Free-Running/Restart**—This bit controls the counter mode of operation after a compare event occurs. In free-running mode, the counter continues after the compare. In restart mode, the counter resets to 0x0000 and resumes counting.	0 = Restart mode (default). 1 = Free-running mode.
CAP Bits 7–6	**Capture Edge**—This field selects the type of transition on the TIN input that triggers a capture event. **Note:** To use TIN/TOUT as a TIN input, ensure that the SEL6 bit in the Port B select register (PBSEL) is cleared.	00 = Disable capture function (default). 01 = Capture on rising edge. 10 = Capture on falling edge. 11 = Capture on rising or falling edges.
OM Bit 5	**Output Mode**—This bit selects the output mode of the timer after a compare event occurs.	0 = Active-low pulse (default). The output appears for one SYSCLK period. 1 = Toggle output.
IRQEN Bit 4	**Interrupt Request Enable**—This bit enables an interrupt on a compare event.	0 = Disable the compare interrupt (default). 1 = Enable the compare interrupt.
CLKSOURCE Bits 3–1	**Clock Source**—This field controls the clock source to the prescaler. The stop count freezes the counter at its current value. **Note:** To use TIN/TOUT as a TIN input, ensure that the SEL6 bit in the Port B select register (PBSEL) is cleared. Also ensure that DIR6 = 0.	000 = Stop counter (default). 001 = SYSCLK to prescaler. 010 = SYSCLK/16 to prescaler. 011 = TIN to prescaler. 1xx = CLK32 to prescaler.
TEN Bit 0	**Timer Enable**—This bit enables or disables the associated timer.	0 = Timer is disabled (default). 1 = Timer is enabled.

12.4.2 Timer Prescaler Registers

Each timer prescaler register (TPRERx) controls the divide ratio of the associated prescaler. The settings for the registers are described in Table 12-3.

TPRER1 Timer Prescaler Register 1 **0x(FF)FFF602**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	Not Used								Prescaler							
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															

TPRER2 Timer Prescaler Register 2 **0x(FF)FFF612**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	Not Used								Prescaler							
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 12-3. Timer Prescaler Register Description

Name	Description	Setting
Not used Bits 15–8	These bits are not used.	—
PRESCALER Bits 7–0	**Prescaler**—This field controls the frequency output of the prescaler. The clock source is divided by the value contained in this register. The value range of this field is between 1 and 256.	0x00 = Divide by 1 . . . 0xFF = Divide by 256

12.4.3 Timer Compare Registers

Each timer compare register (TCMPx) contains the value that is compared with the counter. A compare event is generated when the counter matches the value in this register. This register is set to 0xFFFF at system reset. The settings for the registers are described in Table 12-4.

TCMP1 Timer Compare Register 1 **0x(FF)FFF604**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	COMPARE															
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															

TCMP2 Timer Compare Register 2 **0x(FF)FFF614**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	COMPARE															
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 12-4. Timer Compare Register Description

Name	Description	Setting
COMPARE Bits 15–0	**Compare Value**—Write this field's value to generate a compare event when the counter matches this value.	This field has a valid range of 0x0000 to 0xFFFF.

12.4.4 Timer Capture Registers

Each timer capture register (TCRx) stores the counter value when a capture event occurs. The settings for the registers are described in Table 12-5.

TCR1 Timer Capture Register 1 **0x(FF)FFF606**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	CAPTURE															
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															

TCR2 Timer Capture Register 2 **0x(FF)FFF616**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	CAPTURE															
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 12-5. Timer Capture Register Description

Name	Description	Setting
CAPTURE Bits 15–0	**Capture Value**—This field stores the counter value that existed at the time of the capture event.	This field has a valid range of 0x0000 to 0xFFFF.

12.4.5 Timer Counter Registers

Each read-only timer counter register (TCNx) contains the current count. The TCNx can be read at any time without affecting the current count. The settings for the registers are described in Table 12-6.

TCN1 Timer Counter Register 1 **0x(FF)FFF608**

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

TCN2 Timer Counter Register 2 **0x(FF)FFF618**

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

Table 12-6. Timer Counter Register Description

Name	Description	Setting
COUNT Bits 15–0	**Timer Counter Value**—This 16-bit field contains the current count value.	This field has a valid range of 0x0000 to 0xFFFF.

12.4.6 Timer Status Register

Each timer status register (TSTATx) indicates the corresponding timer's status. When a capture event occurs, it is indicated by setting the CAPT bit. When a compare event occurs, the COMP bit is set. Both bits are cleared by writing 0x0. To be cleared, these bits must first be examined, and the bit must have a value of 0x1. This ensures that an interrupt will not be missed if it occurs between the status read and when the interrupt is cleared. The settings for the registers are described in Table 12-7.

TSTAT1 Timer Status Register 1 **0x(FF)FFF60A**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	Not Used														CAPT	COMP
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															

TSTAT2 Timer Status Register 2 **0x(FF)FFF61A**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	Not Used														CAPT	COMP
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 12-7. Timer Status Register Description

Name	Description	Setting
Not used Bits 15–2	These bits are not used.	—
CAPT Bit 1	**Capture Event**—When set, this status bit indicates that a capture event occurred.	0 = No capture event occurred. 1 = A capture event occurred.
COMP Bit 0	**Compare Event**—When set, this status bit indicates when a compare event occurred.	0 = No compare event occurred. 1 = A compare event occurred.

Chapter 13
Real-Time Clock

This chapter describes the real-time clock (RTC) module, which is composed of six blocks as shown in Figure 13-1.

- Prescaler
- Time-of-day (TOD) clock
- TOD alarm
- Programmable real-time interrupt
- Watchdog timer
- Minute stopwatch

The control registers and bus interface hardware are also described. The RTC module can generate three different interrupts to the interrupt controller. Each of the interrupts are configurable from level 1 to level 6. The RTC can also generate a watchdog system reset. The following sections describe how each block operates and interacts with other modules in both the RTC and the MC68SZ328.

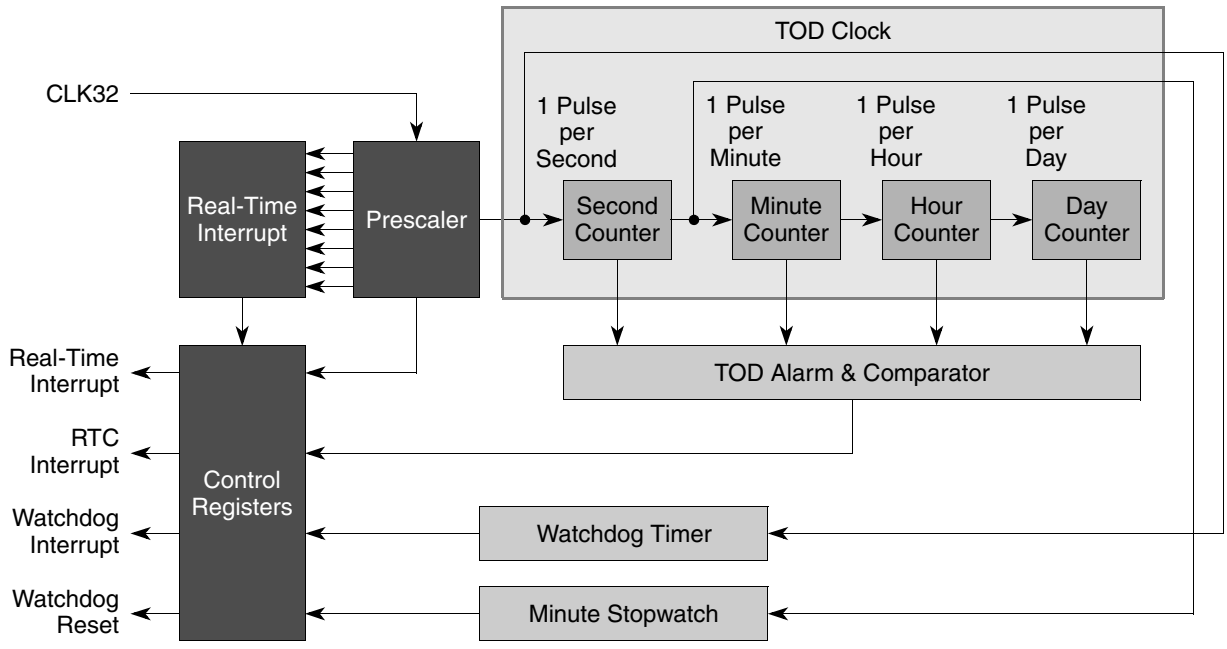


Figure 13-1. Real-Time Clock Module Simplified Block Diagram

13.1 Operation

The prescaler uses the CLK32 clock to create a 1 Hz clock used by all of the blocks in the RTC, as shown in Figure 13-1. The 1 Hz signal is used to increment the counters in the TOD clock. The TOD clock is composed of second, minute, hour, and day counters. When enabled, the TOD alarm generates an RTC interrupt when programmed alarm settings coincide with the TOD counters. The programmable real-time interrupt timer is designed to support application software by providing a fully programmable event timer that generates real-time interrupts to the interrupt controller. In addition, the RTC contains a 2-second watchdog timer and a minute stopwatch.

The RTC can generate 15 event-related interrupts, producing three interrupts that are configurable from level 1 to level 6 to the interrupt controller: a watchdog interrupt, a real-time interrupt, and an RTC interrupt. Each interrupt produced by the RTC, both internally and externally, can be individually enabled or disabled in the real-time interrupt enable register. The mapping of the RTC internal interrupts to the interrupt controller is shown in Table 13-1.

Table 13-1. RTC Interrupt Mapping

Internal Name	Interrupt Controller	Resolution
Real-time interrupt	Real-time interrupt	Eight different rates
Stopwatch	Real-time clock	Minutes
1 HZ	Real-time clock	Seconds
MIN	Real-time clock	Minutes
HR	Real-time clock	Hours
DAY	Real-time clock	Days
ALM	Real-time clock	Seconds
Watchdog	Watchdog	Minutes

The watchdog timer and the entire RTC can also be enabled and disabled. In the following descriptions it is assumed that the real-time clock enable (RTCEN) bit in the real-time control register is set (default), enabling the RTC.

13.1.1 Prescaler

The prescaler divides the CLK32 reference clock down to 1 pulse per second, resulting in a signal labeled 1 HZ. After an initial power up, the CLK32 signal is always available, even when the unit is in a reduced power mode. The actual frequency of the CLK32 is determined by the external crystal used as the crystal oscillator. The MC68SZ328 supports a 32.768 kHz frequency crystal.

The prescaler stages are tapped to support real-time interrupt features. A periodic interrupt at 1 Hz is available, as well as an interrupt at the midnight rollover of the hours counter.

13.1.2 Time-of-Day Counter

Although the four counters that constitute the time-of-day counter are not restricted to operation as a time-of-day counter, most designs use the counters in this fashion. The four counters (seconds, minutes, hours, and days) are toggled by the 1 Hz clock from the prescaler. The seconds and minutes counters (each 6 bits) and the hours counter (5 bits) are maintained in the RTC timer register (RTCTIME). The day

counter (9 bits) can count up to 512 days and is located in its own register (DAYR). The four counters can be read at any time. The seconds, minutes, and hours data is maintained in 24-hour time format, which increments to day counts.

Each of the four counters may be enabled to produce an interrupt on rollover. Upon reaching 59, the seconds and minutes counters each produce an MIN or HR interrupt (when enabled) the next time they are incremented. Both counters reset to 00 and increment the next counter. Likewise, the hours counter, after reaching a count of 23, produces an interrupt (DAY) with the next increment from the minutes counter. The counter resets to 00 and increments the day counter.

13.1.3 Alarm

The alarm is composed of four registers that mirror those found in the time-of-day counter. The seconds, minutes, and hours counters are in the RTC alarm register (RTCALRM). The day alarm register (DAYALRM) contains the 9-bit DAYSAL field.

An alarm is set by accessing the RTCALRM and DAYALRM register and loading the days, hours, minutes, and seconds for the time that the alarm is to generate an interrupt. The alarm is enabled when the ALM bit in the real-time interrupt enable register (RTCIENR) is set. When the time in the TOD counter matches the time in the TOD alarm, the ALM bit in the real-time interrupt status register (RTCISR) is set. If the alarm is not disabled, it will recur every 24 hours. If a single event alarm is desired, then the interrupt service routine should change the values in the alarm registers or disable the ALM bit.

13.1.4 Watchdog Timer

The watchdog timer is an added check to ensure that a program is running and sequencing properly. When application software is running, it is responsible for keeping the 2-second watchdog timer from timing out. If the watchdog timer times out, it is an indication that the software is no longer being executed in the intended sequence. At this time the watchdog timer generates either an interrupt or a reset signal to the system.

Programming the watchdog timer (WATCHDOG) register determines if the 2-second rollover produces a watchdog interrupt or a system reset. At reset, the watchdog timer is enabled and generates a system reset. The watchdog timer is clocked by the 1 Hz clock from the prescaler and therefore has 1-second resolution. It is recommended that the watchdog timer be periodically cleared by software once it is enabled. Otherwise, either a software reset or watchdog interrupt will be generated when the timer reaches a binary value of 10. The timer can be reset by writing any value into it.

13.1.5 Real-Time Interrupt Timer

There is a real-time interrupt available to the user. This interrupt will occur at one of eight different selected rates. Applications for the real-time interrupt can include digitizer sampling, keyboard debouncing, or communication polling.

Each of the eight real-time interrupts operates at a fixed frequency. The frequencies of the real-time interrupts are shown in Table 13-9, “Real-Time Interrupt Frequency Settings,” on page 14-215. Bits RTE0–RTE7 in the RTC interrupt enable register (RTCIENR) enable each of the eight different predefined rates. When the real-time interrupt occurs, it applies an interrupt which is configurable from level 1 to level 6 to the MC68SZ328 interrupt controller. The real-time clock (RTCEN bit in the RTCCTL) or the watchdog timer (EN bit in the watchdog register) must be enabled for the real-time interrupt timer to operate. If the RTC and watchdog timer are disabled, the real-time interrupt stops.

13.1.6 Minute Stopwatch

When enabled, the minute stopwatch performs a countdown that has a 1-minute resolution. The minute stopwatch counts down and remains at decimal -1 until it is reprogrammed. The minute stopwatch can be used to generate an interrupt after a certain number of minutes have elapsed. If the SW bit in the RTCIENR register is enabled with -1 (decimal) in the STPWCH register, an interrupt will be posted on the next minute tick.

13.2 Programming Model

13.2.1 RTC Time of Day Register

The real-time clock hours, minutes, and seconds (RTCTIME) register is used to program the hours, minutes, and seconds. It can be read or written at any time. After a write, the current time assumes the new values. This register cannot be reset since the real-time clock is always enabled at reset. The settings for the RTCTIME register are described in Table 13-2.

RTCTIME RTC Time of Day Register **0x(FF)FFFB00**

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

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
											SECONDS					
TYPE											rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?
	0x00??															

Table 13-2. RTC Time of Day Register Description

Name	Description	Setting
Reserved Bits 31–29	Reserved	These bits are reserved and should be set to 0.
HOURS Bits 28–24	**Hours**—These bits indicate the current hour.	The bits can be set to any value between 0 and 23.
Reserved Bits 23–22	Reserved	These bits are reserved and should be set to 0.
MINUTES Bits 21–16	**Minutes**—These bits indicate the current minute.	The bits can be set to any value between 0 and 59.

Table 13-2. RTC Time of Day Register Description (Continued)

Name	Description	Setting
Reserved Bits 15–6	Reserved	These bits are reserved and should be set to 0.
SECONDS Bits 5–0	**Seconds**—This field indicates the current second.	The bits can be set to any value between 0 and 59.

13.2.2 RTC Day Counter Register

The real-time clock day counter register (DAYR) contains the data from the day counter. The maximum value of DAYR is 512. When the hours counter in RTCTIME reaches 23, the next time increment resets it to 00 and increments the day counter. This register can be read or written at any time. After a write, the current day assumes the new value. This register cannot be reset since it is used to keep the time. The settings for the DAYR register are described in Table 13-3.

DAYR RTC Day Counter Register **0x(FF)FFFB1A**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
								DAYS								
TYPE								rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	?
	0x0???															

Table 13-3. RTC Day Counter Register Description

Name	Description	Setting
Reserved Bits 15–9	Reserved	These bits are reserved and should be set to 0.
DAYS Bits 8–0	**Days**—This field indicates the current setting of the day.	The bits can be set to any value between 0 and 511.

13.2.3 RTC Alarm Register

The real-time clock alarm (RTCALRM) register is used to configure the alarm. The hours, minutes, and seconds can be read or written at any time. After a write, the current time assumes the new values. The settings for the RTCALRM register are described in Table 13-4.

RTCALRM RTC Alarm Register **0x(FF)FFFB04**

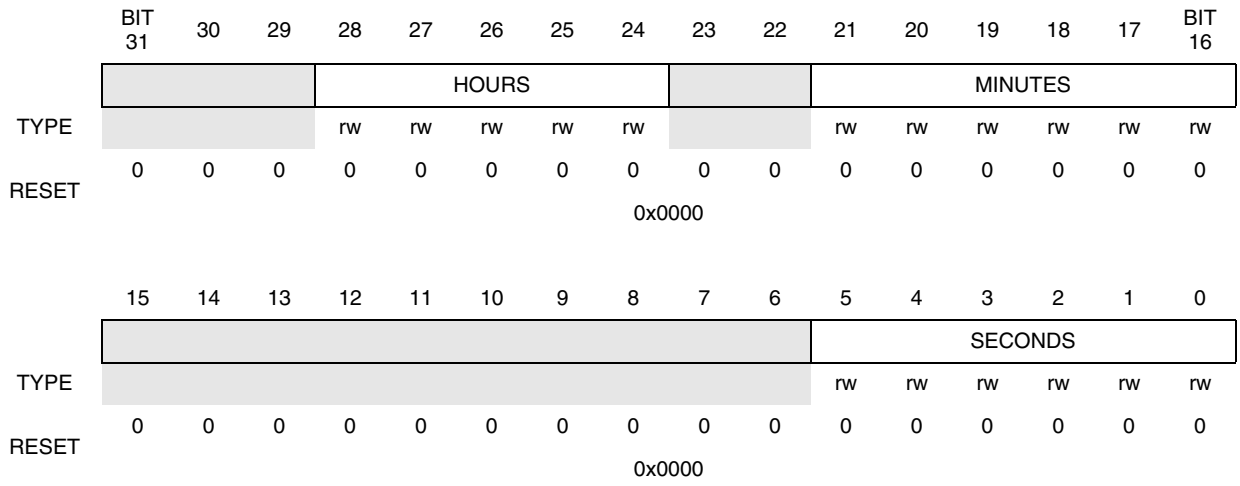


Table 13-4. RTC Alarm Register Description

Name	Description	Setting
Reserved Bits 31–29	Reserved	These bits are reserved and should be set to 0.
HOURS Bits 28–24	**Hours**—This field indicates the value of the hours field in the current alarm setting.	This field can be set to any value between 0 and 23. Default is value 0.
Reserved Bits 23–22	Reserved	These bits are reserved and should be set to 0.
MINUTES Bits 21–16	**Minutes**—This field indicates the value of the minutes field in the current alarm setting.	This field can be set to any value between 0 and 59. Default is value 0.
Reserved Bits 15–6	Reserved	These bits are reserved and should be set to 0.
SECONDS Bits 5–0	**Seconds**—This field indicates the value of the seconds field in the current alarm setting.	This field can be set to any value between 0 and 59. Default is value 0.

13.2.4 RTC Day Alarm Register

The real-time clock day alarm (DAYALRM) register contains the numerical value of the day that generates the alarm. It can be read or written at any time. After a write, the current time assumes the new values. The settings for the DAYALRM register are described in Table 13-5.

DAYALRM RTC Day Alarm Register **0x(FF)FFFB1C**

BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
							DAYSAL								
TYPE							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 13-5. RTC Day Alarm Register Description

Name	Description	Setting
Reserved Bits 15–9	Reserved	These bits are reserved and should be set to 0.
DAYSAL Bits 8–0	**Days Alarm**—This field indicates the numerical setting of the day that will enable the alarm.	The bits can be set to any value between 0 and 511.

13.2.5 Watchdog Timer Register

The watchdog timer (WATCHDOG) register provides all of the control of the watchdog timer. It provides bits to enable the watchdog timer and to determine if the result of a time out is an interrupt or a system reset. The settings for the WATCHDOG register are described in Table 13-6.

WATCHDOG Watchdog Timer Register **0x(FF)FFFB0A**

BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
						CNTR		INTF						ISEL	EN
TYPE						rw	rw	rw						rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	0x0001														

Table 13-6. Watchdog Timer Register Description

Name	Description	Setting
Reserved Bits 15–10	Reserved	These bits are reserved and should be set to 0.
CNTR Bits 9–8	**Counter**—This field represents the value of the watchdog counter, which counts up in 1-second increments. When the watchdog counter counts to 10, it generates a watchdog interrupt. **Note:** Because the watchdog counter is incremented by a 1 Hz signal from the real-time clock, the average tolerance of the counter is 0.5 seconds. Greater accuracy is obtained by polling the 1 Hz flag of the RTCISR.	Writing any value to these bits will reset the counter to 00 (default).

Table 13-6. Watchdog Timer Register Description (Continued)

Name	Description	Setting
INTF Bit 7	**Interrupt Flag**—When this bit is set, a watchdog interrupt has occurred. This bit can be cleared by writing a 1 to it.	0 = No watchdog interrupt occurred. 1 = A watchdog interrupt occurred.
Reserved Bits 6–2	Reserved	These bits are reserved and should be set to 0.
ISEL Bit 1	**Interrupt Selection**—This bit selects the watchdog reset. It is cleared at reset.	0 = Selects the watchdog reset (default). 1 = Select the watchdog interrupt.
EN Bit 0	**Watchdog Timer Enable**—This bit enables the watchdog timer. It is set at reset.	0 = Disable the watchdog timer. 1 = Enable the watchdog timer (default).

13.2.6 RTC Control Register

The real-time clock control (RTCCTL) register is used to enable the real-time clock and provide reference frequency information to the prescaler. The settings for the RTCCTL register are described in Table 13-7.

RTCCTL RTC Control Register **0x(FF)FFFB0C**

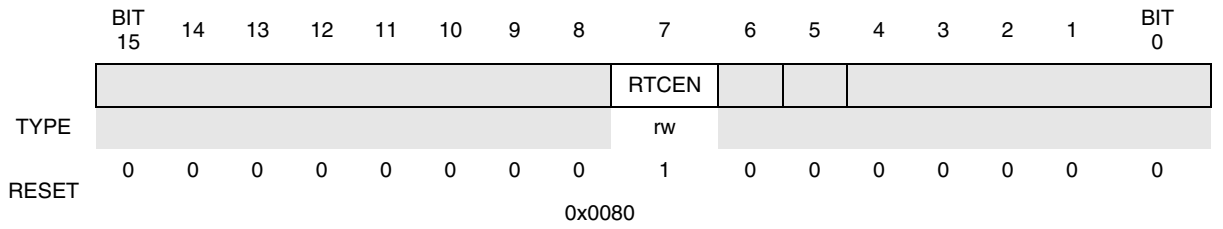


Table 13-7. RTC Control Register Description

Name	Description	Setting
Reserved Bits 15–8	Reserved	These bits are reserved and should be set to 0.
RTCEN Bit 7	**Real-Time Clock Enable**—When set, this bit enables the real-time clock.	0 = Disable the real-time clock 1 = Enable the real-time clock (default)
Reserved Bits 6–0	Reserved	This bit is reserved and should be set to 0.

13.2.7 RTC Interrupt Status Register

The real-time clock interrupt status register (RTCISR) indicates the status of the various real-time clock interrupts. Each bit is set when the corresponding event occurs. You must clear these bits by writing ones, which also clears the interrupt. This register can post interrupts while the system clock is idle or in sleep mode. The settings for the RTCISR register are described in Table 13-8. For more information about the frequency of the RTC interrupts, refer to Table 13-9.

RTCISR RTC Interrupt Status Register **0x(FF)FFB0E**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	RIS7	RIS6	RIS5	RIS4	RIS3	RIS2	RIS1	RIS0			HR	1HZ	DAY	ALM	MIN	SW
TYPE	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 13-8. RTC Interrupt Status Register Description

Name	Description	Setting
RIS7 Bit 15	**Real-Time Interrupt Status Bit 7**—This bit shows the status of real-time interrupt 7.	0 = No RIS7 interrupt occurred. 1 = RIS7 interrupt occurred.
RIS6 Bit 14	**Real-Time Interrupt Status Bit 6**—This bit shows the status of real-time interrupt 6.	0 = No RIS6 interrupt occurred. 1 = RIS6 interrupt occurred.
RIS5 Bit 13	**Real-Time Interrupt Status Bit 5**—This bit shows the status of real-time interrupt 5.	0 = No RIS5 interrupt occurred. 1 = RIS5 interrupt occurred.
RIS4 Bit 12	**Real-Time Interrupt Status Bit 4**—This bit shows the status of real-time interrupt 4.	0 = No RIS4 interrupt occurred. 1 = RIS4 interrupt occurred.
RIS3 Bit 11	**Real-Time Interrupt Status Bit 3**—This bit shows the status of real-time interrupt 3.	0 = No RIS3 interrupt occurred. 1 = RIS3 interrupt occurred.
RIS2 Bit 10	**Real-Time Interrupt Status Bit 2**—This bit shows the status of real-time interrupt 2.	0 = No RIS2 interrupt occurred. 1 = RIS2 interrupt occurred.
RIS1 Bit 9	**Real-Time Interrupt Status Bit 1**—This bit shows the status of real-time interrupt 1.	0 = No RIS1 interrupt occurred. 1 = RIS1 interrupt occurred.
RIS0 Bit 8	**Real-Time Interrupt Status Bit 0**—This bit shows the status of real-time interrupt 0.	0 = No RIS0 interrupt occurred. 1 = RIS0 interrupt occurred.
Reserved Bits 7–6	Reserved	These bits are reserved and should be set to 0.
HR Bit 5	**Hour Flag**—This bit is set on every increment of the hour counter in the TOD clock.	0 = No 1-hour interrupt occurred. 1 = A 1-hour interrupt occurred.
1HZ Bit 4	**1 Hz Flag**—When enabled, this bit is set on every increment of the second counter in the TOD clock.	0 = No 1 Hz interrupt occurred. 1 = A 1 Hz interrupt occurred.
DAY Bit 3	**Day Flag**—When enabled, this bit is set for every 24-hour clock increment (at midnight) of the day counter in the TOD clock, and an interrupt is posted.	0 = No 24-hour rollover interrupt occurred. 1 = A 24-hour rollover interrupt occurred.
ALM Bit 2	**Alarm Flag**—When this bit is enabled, an alarm flag is set on a compare match between the real-time clock and the alarm register's value. **Note:** The alarm will recur every 24 hours. For a single alarm, clear the interrupt enable in the interrupt service routine.	0 = No alarm interrupt occurred. 1 = An alarm interrupt occurred.

Table 13-8. RTC Interrupt Status Register Description (Continued)

Name	Description	Setting
MIN Bit 1	**Minute Flag**—When enabled, this bit is set every increment of the minute counter in the TOD clock.	0 = No 1-minute interrupt occurred. 1 = A 1-minute interrupt has occurred.
SW Bit 0	**Stopwatch Flag**—When enabled, the stopwatch flag is set when the stopwatch minute countdown times out.	0 = The stopwatch did not time out. 1 = The stopwatch timed out.

Table 13-9. Real-Time Interrupt Frequency Settings

Real-Time Interrupt Frequency	32.768 kHz Reference Clock	
RFE7	512 Hz	1.9531 ms
RFE6	256 Hz	3.9062 ms
RFE5	128 Hz	7.8125 ms
RFE4	64 Hz	15.625 ms
RFE3	32 Hz	31.25 ms
RFE2	16 Hz	62.5 ms
RFE1	8 Hz	125 ms
RFE0	4 Hz	250 ms

13.2.8 RTC Interrupt Enable Register

The RTC interrupt enable register (RTCIENR) is used to enable the interrupts in the RTCISR when the corresponding bit is set. The settings for the RTCIENR register are described in Table 13-10. For information about the frequency of the real-time interrupts, refer to Table 13-9.

RTCIENR RTC Interrupt Enable Register **0x(FF)FFFB10**

BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
RIE7	RIE6	RIE5	RIE4	RIE3	RIE2	RIE1	RIE0			HR	1HZ	DAY	ALM	MIN	SW
TYPE 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 13-10. RTC Interrupt Enable Register Description

Name	Description	Setting
RIE7 Bit 15	**Real-Time Interrupt Enable Bit 7**—This bit enables the real-time interrupt 7. The frequency of this interrupt is shown in Table 13-9, “Real-Time Interrupt Frequency Settings,” on page 13-10.	0 = RIE7 interrupt is disabled. 1 = RFE7 interrupt is enabled.
RIE6 Bit 14	**Real-Time Interrupt Enable Bit 6**—This bit enables the real-time interrupt 6. The frequency of this interrupt is shown in Table 13-9, “Real-Time Interrupt Frequency Settings,” on page 13-10.	0 = RIE6 interrupt is disabled. 1 = RIE6 interrupt is enabled.
RIE5 Bit 13	**Real-Time Interrupt Enable Bit 5**—This bit enables the real-time interrupt 5. The frequency of this interrupt is shown in Table 13-9, “Real-Time Interrupt Frequency Settings,” on page 13-10.	0 = RIE5 interrupt is disabled. 1 = RIE5 interrupt is enabled.
RIE4 Bit 12	**Real-Time Interrupt Enable Bit 4**—This bit enables the real-time interrupt 4. The frequency of this interrupt is shown in Table 13-9, “Real-Time Interrupt Frequency Settings,” on page 13-10.	0 = RIE4 interrupt is disabled. 1 = RIE4 interrupt is enabled.
RIE3 Bit 11	**Real-Time Interrupt Enable Bit 3**—This bit enables the real-time interrupt 3. The frequency of this interrupt is shown in Table 13-9, “Real-Time Interrupt Frequency Settings,” on page 13-10.	0 = RIE3 interrupt is disabled. 1 = RIE3 interrupt is enabled.
RIE2 Bit 10	**Real-Time Interrupt Enable Bit 2**—This bit enables the real-time interrupt 2. The frequency of this interrupt is shown in Table 13-9, “Real-Time Interrupt Frequency Settings,” on page 13-10.	0 = RIE2 interrupt is disabled. 1 = RIE2 interrupt is enabled.
RIE1 Bit 9	**Real-Time Interrupt Enable Bit 1**—This bit enables the real-time interrupt 1. The frequency of this interrupt is shown in Table 13-9, “Real-Time Interrupt Frequency Settings,” on page 13-10.	0 = RIE1 interrupt is disabled. 1 = RIE1 interrupt is enabled.

Table 13-10. RTC Interrupt Enable Register Description (Continued)

Name	Description	Setting
RIE0 Bit 8	**Real-Time Interrupt Enable Bit 0**—This bit enables the real-time interrupt 0. The frequency of this interrupt is shown in Table 13-9, "Real-Time Interrupt Frequency Settings," on page 13-10.	0 = RIE0 interrupt is disabled. 1 = RIE0 interrupt is enabled.
Reserved Bits 7–6	Reserved	These bits are reserved and should be set to 0.
HR Bit 5	**Hour Flag**—This bit enables interrupts occurring at a one-per-hour rate.	0 = 1-hour interrupt disabled. 1 = 1-hour interrupt enabled.
1HZ Bit 4	**1 Hz Flag**—This bit enables interrupts occurring at a 1 Hz rate.	0 = 1 Hz interrupt disabled. 1 = 1 Hz interrupt enabled.
DAY Bit 3	**Day Interrupt Enable**—This bit enables the day interrupt occurring at a midnight rollover (0000 hours) of the day counter.	0 = 24-hour rollover interrupt is disabled. 1 = 24-hour rollover interrupt is enabled.
ALM Bit 2	**Alarm Interrupt Enable**—This bit enables the alarm interrupt.	0 = Alarm interrupt is disabled. 1 = Alarm interrupt is enabled.
MIN Bit 1	**Minute Interrupt Enable**—This bit enables the MIN interrupt at the rate of one interrupt per minute.	0 = 1-minute interrupt is disabled. 1 = 1-minute interrupt is enabled.
SW Bit 0	**Stopwatch Interrupt Enable**—This bit enables 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.

13.2.9 Stopwatch Minutes Register

The stopwatch minutes (STPWCH) register contains the current stopwatch countdown value. The stopwatch counter is decremented by the minute (MIN) output from the TOD clock. The average tolerance of the count is 0.5 minutes. The settings for the STPWCH register are described in Table 13-11.

For improved accuracy, enable the stopwatch by polling the MIN bit of the RTCISR register or by polling the minute interrupt service routine.

STPWCH Stopwatch Minutes Register **0x(FF)FFFB12**

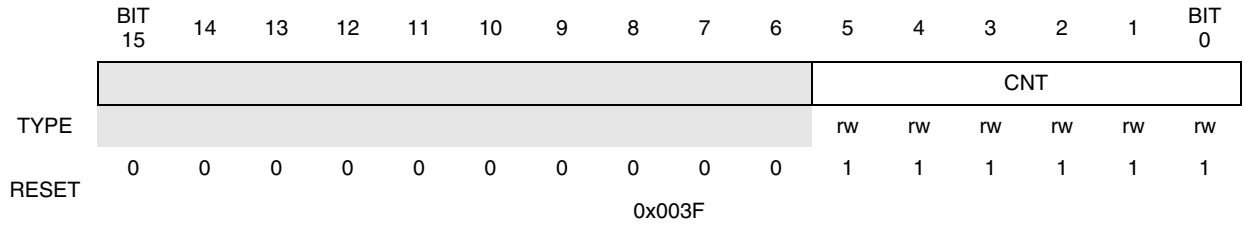


Table 13-11. Stopwatch Minutes Register Description

Name	Description	Setting
Reserved Bits 15–6	Reserved	These bits are reserved and should be set to 0.
CNT Bits 5–0	**Stopwatch Count**—This field contains the stopwatch countdown value.	The highest possible value is 62 minutes. The countdown will not be activated again until a nonzero value, which is less than 63 minutes, is written to this register.

Freescale Semiconductor, Inc.

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Chapter 14
Pulse-Width Modulator 1 and 2

This chapter describes the MC68SZ328's two pulse-width modulators (PWMs). Both pulse-width modulators have three modes of operation—playback, tone, and digital-to-analog (D/A) conversion. Using these modes, the PWM can be used to play back high-quality digital sounds, produce simple tones, or convert digital data into analog waveforms.

14.1 Introduction to PWMs

The output PWMO1 is generated by logically combining the output of both PWMs. The output is available at the PWMO1 external pin. The PWMO2 output is generated solely by PWM 2 and is brought to the PWMO2 external pin. See Figure 14-1.

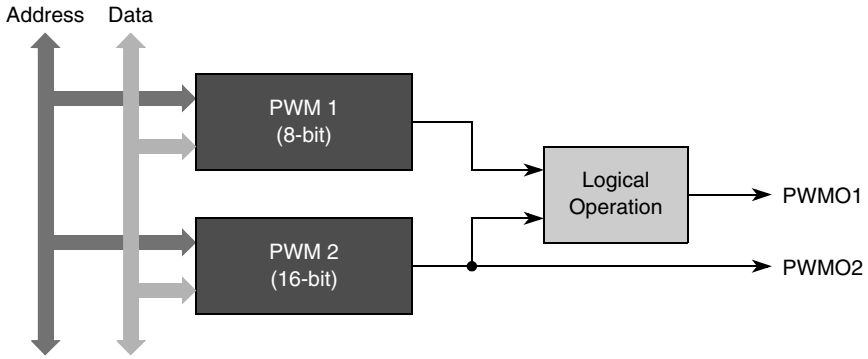


Figure 14-1. PWM 1 and PWM 2 System Configuration Diagram

The operation of the logical block combining the output of PWM 1 and PWM 2 is controlled by programming the P[1:0] bits in the peripheral control register. See Section 6.2.2, "Peripheral Control Register," on page 6-3 for details about the settings of these bits.

14.1.1 PWM 1

PWM 1 is an 8-bit PWM module optimized to generate high-quality sound from stored sample audio files. It can also generate simple or complex tones. It uses 8-bit resolution and a 5-byte FIFO to generate sound. Figure 14-2 illustrates the block diagram of the pulse-width modulator unit 1.

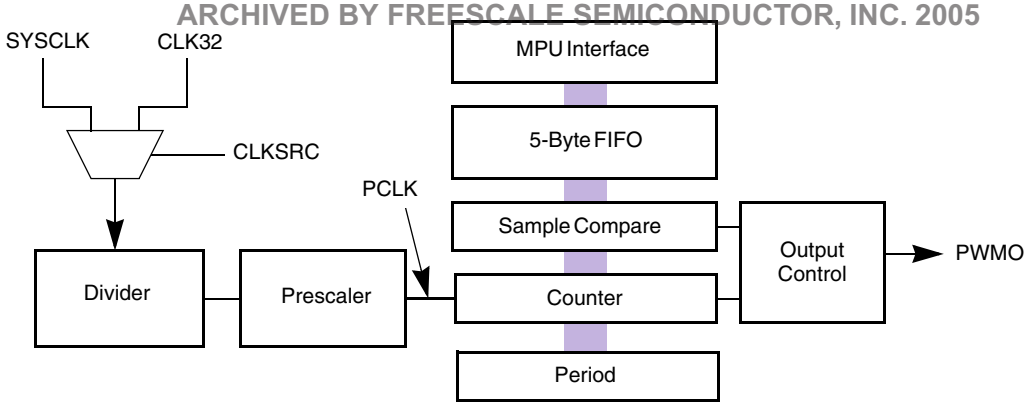


Figure 14-2. PWM 1 Block Diagram

14.1.2 PWM 1 Clock Signals

As shown in Figure 14-2, the prescaler and divider generate the PCLK signal from one of two clock signals—SYSCLK (the default) or CLK32. Selection of the source clock used by pulse-width modulator 1 is made by the clock source (CLKSRC) bit in the PWM 1 control register.

The CLKSEL (clock selection) field in PWMC1 selects the frequency of the output of the divider chain. The incoming clock source is divided by a binary value between 2 and 16.

For 16 kHz audio applications, CLKSEL is equal to %01, divide by 4. For DC-level applications, CLKSEL is equal to %11, divide by 16. In both cases, the following assumptions apply:

- SYSCLK = 16.58 MHz
- Prescaler = 0
- Period = default value

The 7-bit prescaler may be adjusted to achieve lower sampling rates by programming the prescaler field in the PWM 1 control register with any number between 0 and 127, which scales down the incoming clock source by a corresponding factor of 1 to 128.

14.1.3 PWM 2

PWM 2 is a 16-bit PWM module. Besides the difference in the PWM code size (8-bit versus 16-bit), the major difference between PWM 2 and PWM 1 is that PWM 2 does not have a data FIFO. Figure 14-3 illustrates the block diagram of the pulse-width modulator unit 2.

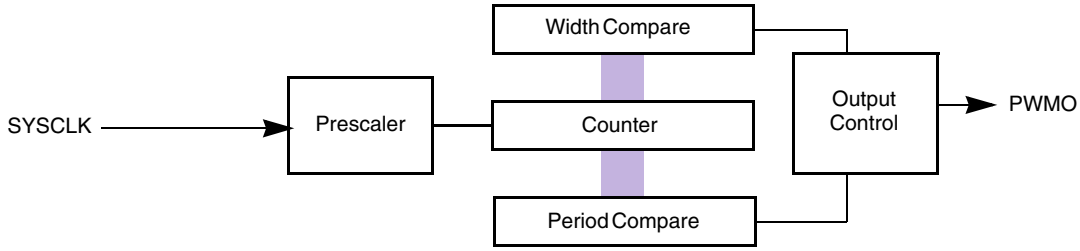


Figure 14-3. PWM 2 Block Diagram

14.2 PWM Operation

The pulse-width modulators have three modes of operation:

1. Playback
2. Tone
3. D/A

14.2.1 Playback Mode

In playback mode, the pulse-width modulators use the data from a sound file to output the resulting audio through an external speaker. Although the PWM can reproduce the contents of a sound file, it is necessary to use a sampling frequency that is equal to or an even multiple of the one used to originally record the sound for the best quality reproduction.

PWM 1 produces variable-width pulses at a constant frequency. The width of the pulse is proportional to the analog voltage of a particular audio sample. At the beginning of a sample period cycle, the PWMO pin is set to 1 and the counter begins counting up from 0x00. The sample value is compared on each count of the prescaler clock. When the sample and count values match, the PWMO signal is cleared to 0. The counter continues counting, and when it overflows from 0xFF to 0x00, another sample period cycle begins. The prescaler clock (PCLK) runs 256 times faster than the sampling rate when the PERIOD field of the PWMP register is at its maximum value; for 16 kHz sampling, PCLK is 4.096 MHz. For human-voice-quality sound, the sampling frequency is either 8 kHz or 16 kHz.

Figure 14-4 illustrates how variable-width pulses affect an audio waveform.

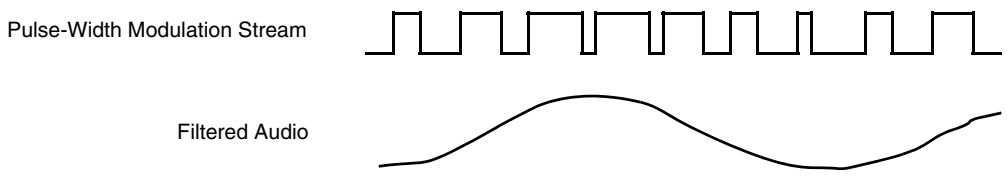


Figure 14-4. Audio Waveform Generation

Digital sample values can be loaded into the pulse-width modulators either as packed 2-sample 16-bit words (big endian format) or as individual 8-bit bytes. A 5-byte FIFO minimizes interrupt overhead. A maskable interrupt is generated when there are 1 or 0 bytes in the FIFO, in which case the software can write either four 1-byte samples or two 2-sample words into the FIFO. When a 16 kHz sampling frequency is being used to play back 8 kHz sampled data while writing 4 bytes at each interrupt, interrupts occur every 500 µs.

14.2.2 Tone Mode

In tone mode, the pulse-width modulators generate a continuous tone at a single frequency when the PWM registers are programmed. The lowest frequency that can be generated is 0.25 Hz.

14.2.3 D/A Mode

The pulse-width modulators can output a frequency with a different pulse width if a low-pass filter is added at the PWMO signal. It can be used to produce a different DC level when programmed using the sample fields in the PWMS1 register. When used in this manner, the PWM becomes a D/A converter.

14.3 Programming Model

This section contains programming information about both PWM 1 and PWM 2.

14.3.1 PWM 1 Control Register

This register controls the operation of the pulse-width modulator 1, and it also contains the status of the PWM 1 FIFO. The register bit assignments are shown in the following register display. The register settings are described in Table 14-1.

PWMC1 PWM 1 Control Register **0x(FF)FFF500**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	CLKSRC	PRESCALER							IRQ	IRQEN	FIFOAV	EN	REPEAT		CLKSEL	
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	0	0	0	0
	0x0020															

Table 14-1. PWM 1 Control Register Description

Name	Description	Setting
CLKSRC Bit 15	**Clock Source**—This bit is used to select the clock source to pulse-width modulator 1.	0 = SYSCLK source is selected (default). 1 = CLK32 is selected. **Note:** 32.768 kHz clock source is selected when using a 32.768 kHz crystal.
PRESCALER Bits 14–8	**Prescaler**—This field is used to scale down the incoming clock to divide by the prescaler + 1. The prescaler is normally used to generate a low single-tone PWMO signal. For voice modulation, these bits are set to 0 (divide by 1). The default value is 0.	Any value between 0 and 127.
IRQ Bit 7	**Interrupt Request**—This bit indicates that the FIFO has one or no bytes remaining, which can be a signal of the need to fill the FIFO by writing no more than two 16-bit words into the PWMS1 register. This bit automatically clears itself after this register is read, thus eliminating an extra write cycle in the interrupt service routine. If the IRQEN bit is 0, this bit can be polled to indicate the status of the period comparator. This bit can be set to immediately post a PWM interrupt for debugging purposes.	0 = The FIFO is not empty. 1 = The FIFO has 1 or no sample bytes remaining.
IRQEN Bit 6	**Interrupt Request Enable**—This bit controls the pulse-width modulator interrupt. While this bit is low, the interrupt is disabled.	0 = The PWM interrupt is disabled (default). 1 = The PWM interrupt is enabled.

Table 14-1. PWM 1 Control Register Description (Continued)

Name	Description	Setting
FIFOAV Bit 5	**FIFO Available**—This bit indicates that the FIFO is available for at least 1 byte of sample data. Data bytes 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 not available. 1 = FIFO available (default).
EN Bit 4	**Enable**—This bit enables or disables pulse-width modulator 1. If this bit is not enabled, writing to other pulse-width modulator 1 registers is ignored.	0 = Disabled.* 1 = Enabled.**
REPEAT Bits 3–2	**Sample Repeats**—This write-only field selects the number of times each sample is repeated. The repeat feature reduces the interrupt overhead, thus reducing CPU loading when audio data is played back at a higher rate, and allows the use of a lower-cost low-pass filter. For example, if the audio data is sampled at 8 kHz and the data is played back at 8 kHz again, an 8 kHz humming noise (carrier) is generated during playback. To filter this carrier, a high-quality low-pass filter is required. For a higher playback rate, it is possible to reconstruct samples at 16 kHz by using the sample twice. This method shifts the carrier from an audible 8 kHz to a less sensitive 16 kHz frequency range, thus providing better sound-quality output.	00 = No samples are repeated (play sample once). This is the default. 01 = Repeat one time (play sample twice). 10 = Repeat three times (play sample four times). 11 = Repeat seven times (play sample eight times).
CLKSEL Bits 1–0	**Clock Selection**—This field selects the output of the divider chain. The approximate sampling rates are calculated using a 16.58 MHz clock source (PRESCALER = 0 and PERIOD = default).	00 = Divide by 2. Provides an approximate 32 kHz sampling rate (default). 01 = Divide by 4. Provides an approximate 16 kHz sampling rate. 10 = Divide by 8. Provides an approximate 8 kHz sampling rate. 11 = Divide by 16. Provides an approximate 4 kHz sampling rate.

Note:

*When pulse-width modulator 1 is disabled, it is in low-power mode, the output pin is forced to 0, and the following events occur:

- The clock prescaler is reset and frozen.
- The counter is reset and frozen.
- The FIFO is flushed.

**When pulse-width modulator 1 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 begin counting.
- The IRQ bit is set, thus indicating that the FIFO is empty.

14.3.2 PWM 1 Sample Register

This register serves as the input to the FIFO. When successive audio sample values are written to this register, they are automatically loaded into the FIFO in big-endian format. If 16-bit words are loaded, the high byte is first placed into the 8-bit FIFO, and then the low byte. When individual sample bytes are being written, they must be written to the low byte (SAMPLE1) only. Pulse-width modulator 1 will revert to free running at the duty-cycle setting that was set last until the FIFO is reloaded or the pulse-width modulator is disabled. If the value in this register is higher than the PERIOD + 1, the output will never be reset, which results in a 100-percent duty cycle.

The register bit assignments are shown in the following register display. The register settings are described in Table 14-2.

PWMS1 PWM 1 Sample Register **0x(FF)FFF502**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	SAMPLE0								SAMPLE1							
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
	0xXXXX															

Table 14-2. PWM 1 Sample Register Description

Name	Description	Setting
SAMPLE0 Bits 15–8	**Sample 0**—This field represents the high byte of a two-sample word. This byte is presented to pulse-width modulator 1 before the SAMPLE1 field.	None
SAMPLE1 Bits 7–0	**Sample 1**—This field represents the low byte of a two-sample word. This byte will be presented to pulse-width modulator 1 after the SAMPLE0 field. With single 8-bit samples, data must be written to this byte.	None

14.3.3 PWM 1 Period Register

This register controls the pulse-width modulator 1 period. When the counter value matches PERIOD + 1, the counter is reset to start another period. Therefore, the following equation applies:

$$PWMO\ (Hz) = PCLK\ (Hz) / (PERIOD + 2) \qquad \textit{Eqn. 14-1}$$

Writing 0xFF to this register achieves the same result as writing 0xFE.

The register bit assignments are shown in the following register display. The register settings are described in Table 14-3.

PWMP1 PWM 1 Period Register **0x(FF)FFF504**

	BIT 7	6	5	4	3	2	1	BIT 0
	PERIOD							
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	0
	0xFE							

Table 14-3. PWM 1 Period Register Description

Name	Description	Setting
PERIOD Bits 7–0	**Period**—This field represents pulse-width modulator 1's period control value.	None

14.3.4 PWM 1 Counter Register

This register contains the current count value and can be read at any time without disturbing the counter. The register bit assignments are shown in the following register display. The register settings are described in Table 14-4.

PWMCNT1 PWM 1 Counter Register **0x(FF)FFF505**

	BIT 7	6	5	4	3	2	1	BIT 0
	COUNT							
TYPE	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 14-4. PWM 1 Counter Register Description

Name	Description	Setting
COUNT Bits 7–0	**Count**—This field represents the value of the current count.	None

14.3.5 PWM 2 Control Register

This register controls how the overall pulse-width modulator 2 operates. Output pin status is also maintained in this register. The register bit assignments are shown in the following register display. The register settings are described in Table 14-5.

PWMC2 PWM 2 Control Register **0x(FF)FFF510**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	PWMIRQ	IRQEN						LOAD	PIN		POL	PWMEN		CLKSEL		
TYPE	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 14-5. PWM 2 Control Register Description

Name	Description	Setting
PWMIRQ Bit 15	**PWM Interrupt**—This bit indicates that a period compare posted an interrupt. This bit may also be set to immediately post a PWM interrupt for debugging purposes. This bit is cleared after it is read while set. If the IRQEN bit is 0, this bit can be polled for the period comparator status.	0 = No PWM period rollover. 1 = PWM period rolled over.
IRQEN Bit 14	**Interrupt Enable**—This bit enables the PWM interrupt.	0 = Disable PWM interrupt. 1 = Enable PWM interrupt.
Reserved Bits 13–9	Reserved	These bits are reserved and should be set to 0.
LOAD Bit 8	**Load New Setting**—This bit forces a new period value and width data to the registers. It automatically clears itself after the loading operation has been performed.	See description.
PIN Bit 7	**Pin Status Indicator**—This bit indicates the current status of the PWM.	0 = PWM output is low. 1 = PWM output is high.
Reserved Bit 6	Reserved	This bit is reserved and should be set to 0.
POL Bit 5	**Output Polarity**—This bit controls the PWM output polarity.	0 = Normal polarity. 1 = Inverted polarity.
PWMEN Bit 4	**PWM Enable**—This bit enables PWM 2.	0 = PWM 2 disabled. 1 = PWM 2 enabled.
Reserved Bit 3	Reserved	This bit is reserved and should be set to 0.
CLKSEL Bits 2–0	**Clock Selection**—This field selects the output of the divider chain.	000 = Divide by 4. 001 = Divide by 8. 010 = Divide by 16. 011 = Divide by 32. 100 = Divide by 64. 101 = Divide by 128. 110 = Divide by 256. 111 = Divide by 512.

14.3.6 PWM 2 Period Register

This register controls the period of PWM 2. When the counter value matches the period control value, an interrupt is generated and the counter is reset to start another period. The register bit assignments are shown in the following register display. The register settings are described in Table 14-5.

PWMP2 PWM 2 Period Register **0x(FF)FFF512**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	PERIOD															
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 14-6. PWM 2 Period Register Description

Name	Description	Setting
PERIOD Bits 15–0	**Period**—This field represents pulse-width modulator 2's period control value.	None

NOTE:

There is a special case: when the register is set to $00, the output will never go high. The pulse signal duty cycle will be 0 percent.

14.3.7 PWM 2 Pulse Width Control Register

This register controls the pulse width of PWM 2. The register bit assignments are shown in the following register display. The register settings are described in Table 14-7.

PWMW2 PWM 2 Pulse Width Control Register **0x(FF)FFF514**

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

Table 14-7. PWM 2 Pulse Width Control Register Description

Name	Description	Setting
WIDTH Bits 15–0	**Width**—When the counter matches the value in this register, the output is reset.	None

NOTE:

If PWMW2 is greater than the period register PWMP2, the output will never be reset. The resulting duty cycle is 100 percent.

14.3.8 PWM 2 Counter Register

This register indicates the current counter value for PWM 2. The register bit assignments are shown in the following register display. The register settings are described in Table 14-8.

PWMCNT2 PWM 2 Counter Register **0x(FF)FFF516**

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

Table 14-8. PWM 2 Counter Register Description

Name	Description	Setting
COUNT Bits 15–0	**Count**—This field indicates the current counter value.	None

Chapter 15
Interrupt Controller

This chapter describes the interrupt controller and all of the signals associated with it. The interrupt controller of the MC68SZ328 supports all internal interrupts as well as external edge- and level-sensitive interrupts. There are seven interrupt levels. Level 7 has the highest priority and level 1 has the lowest. Interrupts can originate from the following sources:

- $\overline{\text{EMUIRQ}}$ or hardware breakpoint interrupt (level 7)
- $\overline{\text{IRQ6}}$ external interrupt (level 6)
- $\overline{\text{IRQ3}}$ external interrupt (level 3)
- $\overline{\text{IRQ2}}$ external interrupt (level 2)
- $\overline{\text{IRQ1}}$ external interrupt (level 1)
- Timer unit 1 (configurable from level 1 to 6)
- Timer unit 2 (configurable from level 1 to 6)
- Pulse-width modulator unit 1 (configurable from level 1 to 6)
- Pulse-width modulator unit 2 (configurable from level 1 to 6)
- LCD Controller unit (configurable from level 1 to 6)
- Configurable serial peripheral interface unit (configurable from level 1 to 6)
- I2C (configurable from level 1 to 6)
- USB (configurable from level 1 to 6)
- DMA unit (configurable from level 1 to 6)
- A/D Converter unit (configurable from level 1 to 6)
- MMCSD unit or MSHC (MS) unit (configurable from level 1 to 6)
- UART unit 1 (configurable from level 1 to 6)
- UART unit 2 (configurable from level 1 to 6)
- Software watchdog timer interrupt (configurable from level 1 to 6)
- Real-time clock (configurable from level 1 to 6)
- Real-time interrupt (configurable from level 1 to 6)
- Ports D, E, F, G, J, K, M, N, P, and R (configurable from level 1 to 6)

15.1 Interrupt Processing

Interrupts on the MC68SZ328 are processed as illustrated in the flowchart shown in Figure 15-1. Details on each stage of the flow diagram are as follows:

1. The interrupt controller collects interrupt events from both on- and off-chip peripherals. Next, it prioritizes them and presents the highest priority request to the CPU if there are no higher interrupts pending; otherwise, the highest priority interrupt is served first.
2. The CPU responds to the interrupt request by executing an interrupt acknowledge bus cycle after the completion of the current instruction.
3. The interrupt controller recognizes the interrupt acknowledge (IACK) cycle and places the interrupt vector for that interrupt request onto the CPU bus.
4. The CPU reads the vector and address of the interrupt handler in the exception vector table and begins execution at that address.

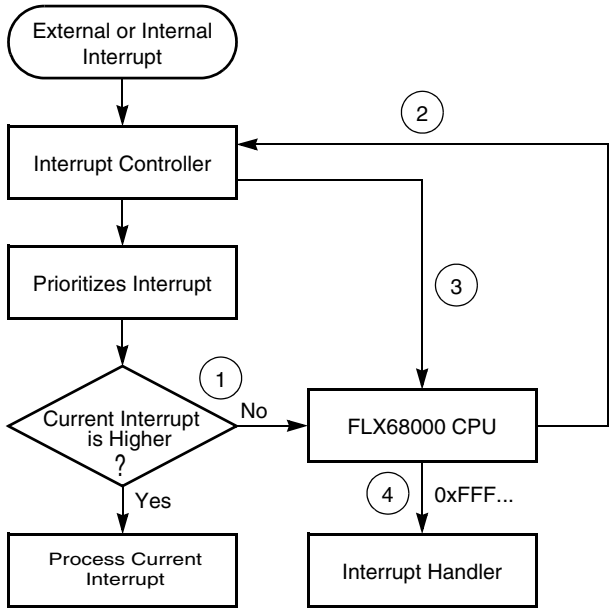


Figure 15-1. Interrupt Processing Flowchart

Steps 2 and 4 are the responsibility of the CPU, whereas steps 1 and 3 are the responsibility of the interrupt controller. External devices must not respond to IACK cycles with a vector because the response is solely the responsibility of the interrupt controller.

On the MC68SZ328, steps 2 and 4 operate exactly as they would on other M68000 devices, which are described in the *M68000 User's Manual*. In step 2, the CPU's status register (SR) is available to mask interrupts globally to determine which priority levels can currently generate interrupts. Also in step 2, the interrupt acknowledge cycle is executed.

In step 4, the CPU reads the vector number, multiplies it by four to get the vector address, fetches a 4-byte program address from that vector address, and then jumps to that 4-byte address. This 4-byte address is the location of the first instruction in the interrupt handler.

The interrupt priority is based on the interrupt level. The interrupts with the same interrupt level are prioritized by the software during the execution of the interrupt service routine. The MC68SZ328 provides one interrupt vector for each interrupt level. The most significant 5 bits of the interrupt vector are

programmable, but the lower 3 bits reflect the interrupt level that is being serviced. All interrupts are maskable. Writing a 1 to a bit in the interrupt mask register disables that interrupt. If an interrupt is masked, you can find out its status in the interrupt pending register.

15.2 Exception Vectors

A vector number is an 8-bit number that can be multiplied by four to obtain the address of an exception vector. An exception vector is the memory location from which the processor fetches the address of a software routine that is used to handle an exception. Each exception has a vector number and an exception vector, as described in Table 15-1. User interrupts are part of the exception processing on the MC68SZ328, and the vector numbers for user interrupts are configurable. For additional information regarding exception processing, see the *M68000 Family Programmer's Reference Manual*.

Table 15-1. Exception Vector Assignment

Vector Number		Address Number		Space[1]	Assignment
Hex	Decimal	Decimal	Hex		
0	0	0	000	SP	Reset: initial SSP[2]
1	1	4	004	SP	Reset: initial PC
2	2	8	008	SD	Bus error
3	3	12	00C	SD	Address error
4	4	16	010	SD	Illegal instruction
5	5	20	014	SD	Divide-by-zero
6	6	24	018	SD	CHK instruction
7	7	28	01C	SD	TRAPV instruction
8	8	32	020	SD	Privilege violation
9	9	36	024	SD	Trace
A	10	40	028	SD	Line 1010 emulator
B	11	44	02C	SD	Line 1111 emulator
C	12	48	030	SD	Unassigned, reserved[3]
D	13	52	034	SD	Unassigned, reserved[3]
E	14	56	038	SD	Unassigned, reserved[3]
F	15	60	03C	SD	Uninitialized interrupt vector
10–17	16–23	64–92	040–05C	SD	Unassigned, reserved[3]
18	24	96	060	SD	Spurious interrupt[4]
19	25	100	064	SD	Level 1 interrupt autovector

Table 15-1. Exception Vector Assignment (Continued)

Vector Number		Address Number		Space[1]	Assignment
Hex	Decimal	Decimal	Hex		
1A	26	104	068	SD	Level 2 interrupt autovector
1B	27	108	06C	SD	Level 3 interrupt autovector
1C	28	112	070	SD	Level 4 interrupt autovector
1D	29	116	074	SD	Level 5 interrupt autovector
1E	30	120	078	SD	Level 6 interrupt autovector
1F	31	124	07C	SD	Level 7 interrupt autovector
20–2F	32–47	128–188	080–0BC	SD	TRAP instruction vectors[5]
30–3F	48–63	192–255	0C0–0FF	SD	Unassigned, reserved[3]
40–FF	64–255	256–1020	100–3FC	SD	User interrupt vectors

1. SP denotes supervisor program space and SD denotes supervisor data space.
2. Reset vector 0 requires four words, unlike the other vectors which only require two words, and it is located in the supervisor program space.
3. Vector numbers 12–14, 16–23, and 48–63 are reserved for future enhancements by Motorola. No peripheral devices should be assigned to these numbers.
4. The spurious interrupt vector is taken when there is a bus error indication during interrupt processing.
5. TRAP #n uses vector number 32 + n (decimal).

NOTE:

The FLX68000 core has reserved vector numbers 0x19 to 0x1F for autovector; however, the MC68SZ328 does not use autovector interrupts. At system startup, the user interrupt vector must be programmed, thereby allowing the processor to handle interrupts properly.

15.3 Reset

The reset exception corresponds to the highest exception level. A reset exception is processed for system initialization and to recover from a catastrophic failure. Any processing that is in progress at the time of the reset is aborted and cannot be recovered. Neither the program counter nor the status register is saved. The processor is forced into the supervisor state. The interrupt priority mask is set at level 7. The address in the first two words of the reset exception vector is fetched by the processor as the initial SSP (supervisor stack pointer), and the address in the next two words of the reset exception vector is fetched as the initial program counter.

At startup or reset, the default chip-select ($\overline{CSA0}$) is asserted and all other chip-selects are negated. The $\overline{CSA0}$ signal should be used to decode an EPROM/ROM memory space. In this case, the first two long words of the EPROM/ROM memory space should be programmed to contain the initial SSP and PC. The initial SSP should point to a RAM space, and the initial PC should point to the startup code within the EPROM/ROM space so that the processor can execute the startup code to bring up the system.

NOTE: The MC68SZ328 supports the **reset** instruction. However, it only resets the CPU, and the $\overline{\text{RESET}}$ pin will not go low when this instruction is issued because it is an input-only signal.

The MC68SZ328's $\overline{\text{RESET}}$ signal should be held low for at least 1.2 s after V_{DD} is applied. After reset, all peripheral function signals and parallel I/O signals appear as inputs with pull-up resistors turned on, unless otherwise specified.

15.4 Interrupt Controller Operation

When interrupts are received by the controller, they are prioritized, and the highest enabled, pending interrupt is posted to the CPU. Before the CPU responds to this interrupt, the status register is copied internally, and then the supervisor bit of the CPU status register is set, placing the processor into supervisor mode. The CPU then responds with an interrupt acknowledge cycle in which the lower 3 bits of the address bus reflect the priority level of the current interrupt. The interrupt controller generates a vector number during the interrupt acknowledge cycle, and the CPU uses this vector number to generate a vector address. Except for the reset exception, the CPU saves the current processor status, including the program counter value (which points to the next instruction to be executed after the interrupt) and the saved copy of the interrupt status register. The new program counter is updated to the content of the interrupt vector, which points to the interrupt service routine. The CPU then resumes instruction execution to execute the interrupt service routine.

15.4.1 Interrupt Priority Processing

Interrupt priority is based on the priority level of the interrupt. If the CPU is currently processing an interrupt service routine and a higher priority interrupt is posted, the process described in Section 15.4 "Interrupt Controller Operation," repeats, and the higher priority interrupt is serviced. If the priority of the newer interrupt is lower than or equal to the priority of the current interrupt, execution of the current interrupt handler continues. The newer interrupt is postponed until its priority becomes the highest. Interrupts within the same level should be prioritized in software by the interrupt handler. The interrupt service routine should end with the **rte** instruction, which restores the processing state prior to the interrupt.

15.4.2 Interrupt Vectors

The MC68SZ328 provides one interrupt vector for each of the seven user interrupt levels. These interrupt vectors form the user interrupt vector section of Table 15-1, "Exception Vector Assignment," on page 254 The user interrupt vectors can be located anywhere within the 0x100 to 0x400 address range. The 5 most significant bits of the interrupt vector number are programmable, but the lower 3 bits reflect the interrupt level being serviced. All interrupts are maskable by the interrupt controller. If an interrupt is masked, its status can still be accessed in the interrupt pending register (IPR).

15.5 Vector Generation

The interrupt controller provides a vector number to the core. You can program the upper 5 bits of the interrupt vector register (IVR) to allow the interrupt vector number to point to any address in the exception vector table. However, many of the vector addresses are assigned to the core's internal exceptions and cannot be reused. This leaves only a small range of address space (0x100 to 0x400) to which you can

configure the IVR to locate user interrupt vectors. For example, if you write a value of 0x40 to the IVR, the interrupt vector base is set to point to 0x100 (0x40<<2), which is the beginning of the user interrupt vectors shown in Table 15-1 on page 15-3. The coding for the vector numbers is provided in Table 15-2.

Table 15-2. Interrupt Vector Numbers

Interrupt	Vector Number
Level 7	xxxxx111
Level 6	xxxxx110
Level 5	xxxxx101
Level 4	xxxxx100
Level 3	xxxxx011
Level 2	xxxxx010
Level 1	xxxxx001
Note: xxxxx is replaced by the upper 5 bits of the interrupt vector register.	

15.6 System Wakeup Interrupts

The following interrupts can be used as wake-up interrupts in sleep mode:

- RTC interrupt
- Timer interrupt
- Watchdog interrupt
- Level interrupt of these ports: PD, PE, PF, PG, PJ, PK, PM, PN, PP, PR
- All interrupts can be used as wake-up interrupts in doze mode

15.7 MMCSD/MS Interrupt

MMCSD/MS interrupt is the logical OR result of the MMCSD interrupt ($\overline{\text{MMCSDIRQ}}$) from the MMC/SD module and Memory Stick interrupt ($\overline{\text{MSIRQ}}$) from MSHC module. In a normal system, either the MMC/SD host controller module or MSHC module is expected to be used and therefore the MMCSD/MS interrupt will be dedicated to one of them. Since their external interface signals are multiplexed on the same I/O port group (PR), it is recommended that the unused module be disabled to ensure proper operation and to guarantee unique interrupt source throughout the operation (the MMC/SD module is disabled by setting MMCSDEN bit of MMC/SD Clock Control Register to 0; MSHC module is disabled by setting MSCEN bit of Memory Stick Control 2 Register to 0).

For I/O multiplexing information on the MSHC and MMC/SD modules, see Chapter 16, “General Purpose I/O Module.”

15.8 Programming Model

This section describes registers that you may need to configure so that the interrupt controller can properly process interrupts, generate vector numbers, and post interrupts to the core.

NOTE:

When programmed as edge-triggered interrupts, all external interrupts ($\overline{IRQ1}$, $\overline{IRQ2}$, $\overline{IRQ3}$, and $\overline{IRQ6}$) can be cleared by writing a 1 to the corresponding status bit in the interrupt status register (ISR). When programmed as level-triggered interrupts, these interrupts are cleared at the requesting sources. All interrupts from internal peripheral devices are level-triggered interrupts to the interrupt handler, and they are cleared at the requesting sources.

15.8.1 Interrupt Vector Register

The interrupt vector register (IVR) is used to program the upper 5 bits of the interrupt vector number. During the interrupt acknowledge cycle, the lower 3 bits, encoded from the interrupt level, are combined with the upper 5 bits to form an 8-bit vector number. The CPU uses the vector number to generate a vector address. During system startup, this register should be configured so that the MC68SZ328's external and internal interrupts can be handled properly by their software handlers. If an interrupt occurs before the IVR has been programmed, the interrupt vector number 0x0F is returned to the CPU as an uninitialized interrupt, which has the interrupt vector 0x3C.

The register bit assignments are shown in the following register display, and their settings are described in Table 15-3.

IVR Interrupt Vector Register **0x(FF)FFF300**

	BIT 7	6	5	4	3	2	1	BIT 0
	VECTOR							
TYPE	rw	rw	rw	rw	rw			
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 15-3. Interrupt Vector Register Description

Name	Description	Settings
VECTOR Bits 7–3	**Vector Number**—This field represents the upper 5 bits of the interrupt vector number.	See description.
Reserved Bits 2–0	Reserved	These bits are reserved and should be set to 0.

15.8.2 Interrupt Control Register

The interrupt control register (ICR) controls the behavior of the external interrupt inputs. It informs the interrupt controller whether the interrupt signal is an edge-triggered or a level-sensitive interrupt, as well as whether it has positive or negative polarity. The bit assignments for this register are shown in the following register display, and the settings for the bit positions are listed in Table 15-4.

ICR Interrupt Control Register **0x(FF)FFF302**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	POL1	POL2	POL3	POL6	ET1	ET2	ET3	ET6								
TYPE	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 15-4. Interrupt Control Register Description

Name	Description	Setting
POL1 Bit 15	**Polarity Control 1**—This bit controls interrupt polarity for the $\overline{\text{IRQ1}}$ signal. In level-sensitive mode, negative polarity produces an interrupt when the signal is at logic level low. Positive polarity produces an interrupt when the signal is at logic level high. In edge-triggered mode, negative polarity produces an interrupt when the signal goes from logic level high to logic level low. Positive polarity generates an interrupt when the signal goes from logic level low to logic level high.	0 = Negative polarity. 1 = Positive polarity.
POL2 Bit 14	**Polarity Control 2**—This bit controls interrupt polarity for the $\overline{\text{IRQ2}}$ signal. In level-sensitive mode, negative polarity produces an interrupt when the signal is at logic level low. Positive polarity produces an interrupt when the signal is at logic level high. In edge-triggered mode, negative polarity produces an interrupt when the signal goes from logic level high to logic level low. Positive polarity generates an interrupt when the signal goes from logic level low to logic level high.	0 = Negative polarity. 1 = Positive polarity.
POL3 Bit 13	**Polarity Control 3**—This bit controls interrupt polarity for the $\overline{\text{IRQ3}}$ signal. In level-sensitive mode, negative polarity produces an interrupt when the signal is at logic level low. Positive polarity produces an interrupt when the signal is at logic level high. In edge-triggered mode, negative polarity produces an interrupt when the signal goes from logic level high to logic level low. Positive polarity generates an interrupt when the signal goes from logic level low to logic level high.	0 = Negative polarity. 1 = Positive polarity.
POL6 Bit 12	**Polarity Control 6**—This bit controls interrupt polarity for the $\overline{\text{IRQ6}}$ signal. In level-sensitive mode, negative polarity produces an interrupt when the signal is at logic level low. Positive polarity produces an interrupt when the signal is at logic level high. In edge-triggered mode, negative polarity produces an interrupt when the signal goes from logic level high to logic level low. Positive polarity generates an interrupt when the signal goes from logic level low to logic level high.	0 = Negative polarity. 1 = Positive polarity.
ET1 Bit 11	**$\overline{\text{IRQ1}}$ Edge Trigger Select**—When this bit is set, the $\overline{\text{IRQ1}}$ signal is an edge-triggered interrupt. In edge-triggered mode, a 1 must be written to the IRQ1 bit in the interrupt status register to clear this interrupt. When this bit is low, $\overline{\text{IRQ1}}$ is a level-sensitive interrupt. In this case, the external source of the interrupt must be cleared.	0 = Level-sensitive interrupt. 1 = Edge-sensitive interrupt.

Table 15-4. Interrupt Control Register Description (Continued)

Name	Description	Setting
ET2 Bit 10	**$\overline{\text{IRQ2}}$ Edge Trigger Select**—When this bit is set, the $\overline{\text{IRQ2}}$ signal is an edge-triggered interrupt. In edge-triggered mode, a 1 must be written to the IRQ2 bit in the interrupt status register to clear this interrupt. When this bit is low, $\overline{\text{IRQ2}}$ is a level-sensitive interrupt. In this case, the external source of the interrupt must be cleared.	0 = Level-sensitive interrupt. 1 = Edge-sensitive interrupt.
ET3 Bit 9	**$\overline{\text{IRQ3}}$ Edge Trigger Select**—When this bit is set, the $\overline{\text{IRQ3}}$ signal is an edge-triggered interrupt. In edge-triggered mode, a 1 must be written to the IRQ3 bit in the interrupt status register to clear this interrupt. When this bit is low, $\overline{\text{IRQ3}}$ is a level-sensitive interrupt. In this case, the external source of the interrupt must be cleared.	0 = Level-sensitive interrupt. 1 = Edge-sensitive interrupt.
ET6 Bit 8	**$\overline{\text{IRQ6}}$ Edge Trigger Select**—When this bit is set, the $\overline{\text{IRQ6}}$ signal is an edge-triggered interrupt. In edge-triggered mode, a 1 must be written to the IRQ6 bit in the interrupt status register to clear this interrupt. When this bit is low, $\overline{\text{IRQ6}}$ is a level-sensitive interrupt. In this case, the external source of the interrupt must be cleared.	0 = Level-sensitive interrupt. 1 = Edge-sensitive interrupt.
Reserved Bits 7–0	Reserved	These bits are reserved and should remain at their default value.
Note: Clear interrupts after changing modes. When modes are changed from level to edge interrupts, an edge can be created, which causes an interrupt to be posted.		

15.8.3 Interrupt Mask Register

The interrupt mask register (IMR) can mask out a particular interrupt if the corresponding bit for the interrupt is set. There is one control bit for each interrupt source. When an interrupt is masked, the interrupt controller will not generate an interrupt request to the CPU, but its status can still be observed in the interrupt pending register. At reset, all the interrupts are masked and all the bits in this register are set to 1.

IMR Interrupt Mask Register **0x(FF)FFF304**

	BIT 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	BIT 16
	M USB	M I2C	MMC SD/ MS	M PK	M PM	M PN	M PP	MA/D C	MEMI Q	MRTI	MCS PI	M PR	MIRQ 6	MIRQ 3	MIRQ 2	MIRQ 1
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	BIT 0
	M DMA 1	M DMA 2	MPW M2	MUA RT2	M PD	M PE	M PF	M PG	MPW M1	M PJ	MTM R2	MRT C	MWD T	MUA RT1	MTM R1	MLCD C
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 15-5. Interrupt Mask Register Description

Name	Description	Settings
MUSB Bit 31	**Mask USB Interrupt**—When set, this bit indicates that the USB interrupt is masked. It is set to 1 after reset.	0 = Enable USB interrupt. 1 = Masked USB interrupt.
MI2C Bit 30	**Mask I2C Interrupt**—When set, this bit indicates that the I2C interrupt is masked. It is set to 1 after reset.	0 = Enable I2C interrupt. 1 = Masked I2C interrupt.
MMCSD/MS Bit 29	**Mask MMCSD/MS Interrupt**—When set, this bit indicates that the MMCSD/MS Interrupt is masked. It is set to 1 after reset.	0 = Enable MMCSD/MS interrupt. 1 = Masked MMCSD/MS interrupt.
MPK Bit 28	**Mask PK Interrupt**—When set, this bit indicates that the PK Port K interrupt is masked. It is set to 1 after reset.	0 = Enable PK interrupt. 1 = Masked PK interrupt.
MPM Bit 27	**Mask PM Interrupt**—When set, this bit indicates that the PM Port M interrupt is masked. It is set to 1 after reset.	0 = Enable PM interrupt. 1 = Masked PM interrupt.
MPN Bit 26	**Mask PN Interrupt**—When set, this bit indicates that the PN Port N interrupt is masked. It is set to 1 after reset.	0 = Enable PN interrupt. 1 = Masked PN interrupt.
MPP Bit 25	**Mask PP Interrupt**—When set, this bit indicates that the PP Port P interrupt is masked. It is set to 1 after reset.	0 = Enable PP interrupt. 1 = Masked PP interrupt.

Table 15-5. Interrupt Mask Register Description (Continued)

Name	Description	Settings
MADC Bit 24	**Mask A/D Converter Interrupt**—When set, this bit indicates that the A/D C interrupt is masked. It is set to 1 after reset.	0 = Enable A/D C interrupt. 1 = Masked A/D C interrupt.
MEMIQ Bit 23	**Mask Emulator Interrupt**—When set, this bit indicates that the $\overline{\text{EMUIRQ}}$ pin and in-circuit emulation breakpoint interrupt functions are masked. It is set to 1 after reset. These interrupts are level 7 interrupts to the CPU.	0 = Enable $\overline{\text{EMUIRQ}}$ interrupt 1 = Mask $\overline{\text{EMUIRQ}}$ interrupt
MRTI Bit 22	**Timer for Real-Time Clock**—When set, this bit indicates that the real-time interrupt timer is masked. It is set to 1 after reset.	0 = Enable real-time interrupt timer interrupt. 1 = Masked real-time interrupt timer interrupt.
MCSPI Bit 21	**Mask CSPI Interrupt**—When set, this bit indicates that the CSPI interrupt is masked. It is set to 1 after reset.	0 = Enable CSPI interrupt. 1 = Mask CSPI interrupt.
MPR Bit 20	**Mask PR Interrupt**—When set, this bit indicates that the PR Port R interrupt is masked. It is set to 1 after reset.	0 = Enable PR interrupt. 1 = Masked PR interrupt.
MIRQ6 Bit 19	**Mask $\overline{\text{IRQ6}}$ Interrupt**—When set, this bit indicates that $\overline{\text{IRQ6}}$ is masked. It is set to 1 after reset.	0 = Enable $\overline{\text{IRQ6}}$ interrupt. 1 = Mask $\overline{\text{IRQ6}}$ interrupt.
MIRQ3 Bit 18	**Mask $\overline{\text{IRQ3}}$ Interrupt**—When set, this bit indicates that $\overline{\text{IRQ3}}$ is masked. It is set to 1 after reset.	0 = Enable $\overline{\text{IRQ3}}$ interrupt. 1 = Mask $\overline{\text{IRQ3}}$ interrupt.
MIRQ2 Bit 17	**Mask $\overline{\text{IRQ2}}$ Interrupt**—When set, this bit indicates that $\overline{\text{IRQ2}}$ is masked. It is set to 1 after reset.	0 = Enable $\overline{\text{IRQ2}}$ interrupt. 1 = Mask $\overline{\text{IRQ2}}$ interrupt.
MIRQ1 Bit 16	**Mask $\overline{\text{IRQ1}}$ Interrupt**—When set, this bit indicates that $\overline{\text{IRQ1}}$ is masked. It is set to 1 after reset.	0 = Enable $\overline{\text{IRQ1}}$ interrupt. 1 = Mask $\overline{\text{IRQ1}}$ interrupt.
MDMA1 Bit 15	**Mask DMA1 Interrupt**—When set, this bit indicates that DMA Transfer Complete interrupt is masked. It is set to 1 after reset.	0 = Enable DMA1 interrupt. 1 = Mask DMA1 interrupt.
MDMA2 Bit 14	**Mask DMA2 Interrupt**—When set, this bit indicates that DMA Transfer Error interrupt is masked. It is set to 1 after reset.	0 = Enable DMA2 interrupt. 1 = Mask DMA2 interrupt.
MPWM2 Bit 13	**Mask PWM2 Interrupt**—Setting this bit indicates that PWM2 interrupt is masked. It is set to 1 after reset.	0 = Enable pulse-width modulator 2 interrupt. 1 = Mask pulse-width modulator 2 interrupt.
MUART2 Bit 12	**Mask UART 2 Interrupt**—When set, this bit indicates that UART2 interrupt is masked. It is set to 1 after reset.	0 = Enable UART 2 interrupt. 1 = Mask UART 2 interrupt.
MPD Bit 11	**Mask PD Interrupt**—When set, this bit indicates that the PD Port D interrupt is masked. It is set to 1 after reset.	0 = Enable PD interrupt. 1 = Masked PD interrupt.
MPE Bit 10	**Mask PE Interrupt**—When set, this bit indicates that the PE Port E interrupt is masked. It is set to 1 after reset.	0 = Enable PE interrupt. 1 = Masked PE interrupt.
MPF Bit 9	**Mask PF Interrupt**—When set, this bit indicates that the PF Port F interrupt is masked. It is set to 1 after reset.	0 = Enable PF interrupt. 1 = Masked PF interrupt.

Table 15-5. Interrupt Mask Register Description (Continued)

Name	Description	Settings
MPG Bit 8	**Mask PG Interrupt**—When set, this bit indicates that the PG Port G interrupt is masked. It is set to 1 after reset.	0 = Enable PG interrupt. 1 = Masked PG interrupt.
MPWM1 Bit 7	**Mask PWM1 Interrupt**—Setting this bit indicates that PWM1 interrupt is masked. It is set to 1 after reset.	0 = Enable pulse-width modulator 1 interrupt. 1 = Mask pulse-width modulator 1 interrupt.
MPJ Bit 6	**Mask PJ Interrupt**—When set, this bit indicates that the PJ Port J interrupt is masked. It is set to 1 after reset.	0 = Enable PJ interrupt. 1 = Masked PJ interrupt.
MTMR2 Bit 5	**Mask Timer 2 Interrupt**—Setting this bit masks the timer interrupt. It is set to 1 after reset.	0 = Enable timer 2 interrupt. 1 = Mask timer 2 interrupt.
MRTC Bit 4	**Mask RTC Interrupt**—Setting this bit masks the real-time clock (time of day) interrupt. It is set to 1 after reset.	0 = Enable real-time clock interrupt. 1 = Mask real-time clock interrupt.
MWDT Bit 3	**Mask Watchdog Timer Interrupt**—Setting this bit masks the watchdog timer interrupt. It is set to 1 after reset.	0 = Enable watchdog timer interrupt. 1 = Mask watchdog timer interrupt.
MUART1 Bit 2	**Mask UART 1 Interrupt**—When set, this bit indicates that the UART 1 interrupt is masked. It is set to 1 after reset.	0 = Enable UART 1 interrupt. 1 = Mask UART 1 interrupt.
MTMR1 Bit 1	**Mask Timer 1 Interrupt**—Setting this bit masks the timer interrupt. It is set to 1 after reset.	0 = Enable timer 1 interrupt. 1 = Mask timer 1 interrupt.
MLCDC Bit 0	**Mask LCDC Interrupt**—Setting this bit indicates that LCDC interrupt is masked. It is set to 1 after reset.	0 = Enable LCDC interrupt. 1 = Mask LCDC interrupt.

15.8.4 Interrupt Status Register

During the interrupt service, the interrupt handler determines the source of interrupts by examining the interrupt status register (ISR). When the bits in this register are set, they indicate that the corresponding interrupt is posted to the core. If there are multiple interrupt sources at the same level, the software handler may need to prioritize them, depending on the application.

Each interrupt status bit in this register reflects the interrupt request from its respective interrupt source. When programmed as edge-triggered interrupts, external interrupts $\overline{\text{IRQ1}}$, $\overline{\text{IRQ2}}$, $\overline{\text{IRQ3}}$, and $\overline{\text{IRQ6}}$ can be cleared by writing a 1 to the corresponding status bit in the register. When programmed as level-triggered interrupts, these interrupts are cleared at the requesting sources. All interrupts from internal peripheral devices are level-triggered interrupts to the interrupt handler, and they are cleared at the requesting sources.

ISR Interrupt Status Register 0xFFFFF30C

	BIT 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	BIT 16
	USB	I2C	MMC SD/ MS	PK	PM	PN	PP	A/D C	EMIQ	RTI	CSPI	PR	1RQ6	IRQ3	IRQ2	IRQ1
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	BIT 0
	DMA1	DMA2	PWM 2	UART 2	PD	PE	PF	PG	PWM 1	PJ	TMR2	RTC	WDT	UART 1	TMR1	LCDC
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 15-6. Interrupt Status Register Description

Name	Description	Settings
USB Bit 31	**USB Interrupt Status**—When set, this bit indicates an interrupt event from USB.	0 = No USB interrupt is pending. 1 = An USB interrupt is pending.
I2C Bit 30	**I2C Interrupt Status**—When set, this bit indicates an interrupt event from I2C.	0 = No I2C interrupt is pending. 1 = An I2C interrupt is pending.
MMCSD/MS Bit 29	**MMCSD/MS Interrupt Status**—When set, this bit indicates an interrupt event from MMCSD/MS unit.	0 = No MMCSD/MS interrupt is pending. 1 = An MMCSD/MS interrupt is pending.
PK Bit 28	**PK Interrupt Status**—When set, this bit indicates an interrupt event from Port K.	0 = No PK interrupt is pending. 1 = An PK interrupt is pending.
PM Bit 27	**PM Interrupt Status**—When set, this bit indicates an interrupt event from Port M.	0 = No PM interrupt is pending. 1 = An PM interrupt is pending.
PN Bit 26	**PN Interrupt Status**—When set, this bit indicates an interrupt event from Port N.	0 = No PN interrupt is pending. 1 = An PN interrupt is pending.

Table 15-6. Interrupt Status Register Description (Continued)

Name	Description	Settings
PP Bit 25	**PP Interrupt Status**—When set, this bit indicates an interrupt event from Port P.	0 = No PP interrupt is pending. 1 = An PP interrupt is pending.
A/D C Bit 24	**A/D Converter Interrupt Status**—When set, this bit indicates an interrupt event from A/D Converter unit.	0 = No A/D C interrupt is pending. 1 = An A/D C interrupt is pending.
EMIQ Bit 23	**Emulator Interrupt Status**—When set, this bit indicates that the in-circuit emulation module or $\overline{\text{EMUIRQ}}$ pin is requesting an interrupt on level 7. This bit can be generated from three interrupt sources: two breakpoint interrupts from the in-circuit emulation module and an external interrupt from $\overline{\text{EMUIRQ}}$, which is an active low, edge-sensitive interrupt. To clear this interrupt, you must read the ICEMSR register to identify the interrupt source and write a 1 to the corresponding bit of that register. See Section 16.2.4 on page 16-8 for more information.	0 = No emulator interrupt is pending. 1 = An emulator interrupt is pending.
RTI Bit 22	**Real-Time Interrupt Status (Real-Time Clock)**—When set, this bit indicates that the real-time timer has reached its predefined frequency count. The frequency can be selected inside the real-time clock module, which can function as an additional timer.	0= Real-time timer has not reached predefined frequency count. 1= Real-time timer has reached predefined frequency count.
CSPI Bit 21	**CSPI Interrupt Status**—When set, this bit indicates an interrupt event from CSPI unit.	0 = No CSPI interrupt is pending. 1 = An CSPI interrupt is pending.
PR Bit 20	**PR Interrupt Status**—When set, this bit indicates an interrupt event from Port R.	0 = No PR interrupt is pending. 1 = An PR interrupt is pending.
IRQ6 Bit 19	**Interrupt Request Level 6**—This bit, when set, indicates that an external device is requesting an interrupt on level 6. If the $\overline{\text{IRQ6}}$ signal is set to be a level-sensitive interrupt, the source of the interrupt must first be cleared. If $\overline{\text{IRQ6}}$ is set to be an edge-triggered interrupt, the interrupt must be cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.	0 = No level 6 interrupt is pending. 1 = A level 6 interrupt is pending.
IRQ3 Bit 18	**Interrupt Request Level 3**—This bit, when set, indicates that an external device is requesting an interrupt on level 3. If the $\overline{\text{IRQ3}}$ signal is set to be a level-sensitive interrupt, the source of the interrupt must first be cleared. If $\overline{\text{IRQ3}}$ is set to be an edge-triggered interrupt, the interrupt must be cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.	0 = No level 3 interrupt is pending. 1 = A level 3 interrupt is pending.

Table 15-6. Interrupt Status Register Description (Continued)

Name	Description	Settings
IRQ2 Bit 17	**Interrupt Request Level 2**—This bit, when set, indicates that an external device is requesting an interrupt on level 2. If the $\overline{\text{IRQ2}}$ signal is set to be a level-sensitive interrupt, the source of the interrupt must first be cleared. If $\overline{\text{IRQ2}}$ is set to be an edge-triggered interrupt, the interrupt must be cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.	0 = No level 2 interrupt is pending. 1 = A level 2 interrupt is pending.
IRQ1 Bit 16	**Interrupt Request Level 1**—This bit, when set, indicates that an external device is requesting an interrupt on level 1. If the $\overline{\text{IRQ1}}$ signal is set to be a level-sensitive interrupt, the source of the interrupt must first be cleared. If $\overline{\text{IRQ1}}$ is set to be an edge-triggered interrupt, the interrupt must be cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.	0 = No level 1 interrupt is pending. 1 = A level 1 interrupt is pending.
DMA1 Bit 15	**DMA1 Interrupt**—This bit, when set, indicates that there is a DMA Transfer Complete interrupt event from DMA unit pending.	0 = No DMA1 interrupt is pending. 1 = A DMA1 interrupt is pending.
DMA2 Bit 14	**DMA2 Interrupt**—This bit, when set, indicates that there is a DMA Transfer Error interrupt event from DMA unit pending.	0 = No DMA2 interrupt is pending. 1 = A DMA2 interrupt is pending.
PWM2 Bit 13	**Pulse-Width Modulator (PWM) 2 Interrupt**—This bit, when set, indicates that there is an interrupt event from PWM unit 2 pending.	0 = No PWM2 interrupt is pending. 1 = A PWM2 interrupt is pending.
UART2 Bit 12	**UART 2 Interrupt Request**—When set, this bit indicates that the UART 2 module needs service. The interrupt level is configurable from level 1 to level 6. See Section 9.6.6, “Interrupt Level Register,” for more details.	0 = No UART 2 interrupt request is pending. 1 = UART 2 interrupt request is pending.
PD Bit 11	**PD Interrupt Status**—When set, this bit indicates an interrupt event from Port D.	0 = No PD interrupt is pending. 1 = An PD interrupt is pending.
PE Bit 10	**PE Interrupt Status**—When set, this bit indicates an interrupt event from Port E.	0 = No PE interrupt is pending. 1 = An PE interrupt is pending.
PF Bit 9	**PF Interrupt Status**—When set, this bit indicates an interrupt event from Port F.	0 = No PF interrupt is pending. 1 = An PF interrupt is pending.
PG Bit 8	**PG Interrupt Status**—When set, this bit indicates an interrupt event from Port G.	0 = No PG interrupt is pending. 1 = An PG interrupt is pending.

Table 15-6. Interrupt Status Register Description (Continued)

Name	Description	Settings
PWM1 Bit 7	**Pulse-Width Modulator (PWM) 1 Interrupt**—This bit, when set, indicates that there is a level 6 interrupt event from PWM unit 1 pending.	0 = No PWM1 interrupt is pending. 1 = A PWM 1interrupt is pending.
PJ Bit 6	**PJ Interrupt Status**—When set, this bit indicates an interrupt event from Port J.	0 = No PJ interrupt is pending. 1 = An PJ interrupt is pending.
TMR2 Bit 5	**Timer 2 Interrupt Status**—This bit indicates that a timer 2 event has occurred. This is a level 4 interrupt.	0 = No timer 2 event occurred. 1 = A timer 2 event has occurred.
RTC Bit 4	**Real-Time Clock Interrupt Request**—This bit, when set, indicates that there is a level 4 interrupt event from the real-time clock that is pending.	0 = No real-time clock interrupt is pending. 1 = A real-time clock interrupt is pending.
WDT Bit 3	**Watchdog Timer Interrupt Request**—This bit indicates that a watchdog timer interrupt is pending. This is a level 4 interrupt.	0 =No watchdog timer interrupt is pending. 1 = A watchdog timer interrupt is pending.
UART1 Bit 2	**UART 1 Interrupt Request**—When set, this bit indicates that the UART 1 module needs service. This is a level 4 interrupt.	0 = No UART1 service request is pending. 1 = UART1 service is needed.
TMR1 Bit 1	**Timer 1 Interrupt Status**—This bit indicates that a timer 1 event has occurred. This is a level 6 interrupt.	0 = No timer 1 event occurred. 1 = A timer 1 event has occurred.
LCDC Bit 0	**LCD Controller Interrupt**—This bit indicates that an interrupt event from LCDC unit is pending. The interrupt level is configurable from level 1 to level 6. See Section 9.6.6, “Interrupt Level Register,” for more details.	0 = No LCDC interrupt is pending. 1 = A LCDC interrupt is pending.

15.8.5 Interrupt Pending Register

The read-only interrupt pending register (IPR) indicates which interrupts are pending. If an interrupt source requests an interrupt, but that interrupt is masked by the interrupt mask register, then that interrupt bit will be set in this register, but not in the interrupt status register. If the pending interrupt is not masked, the interrupt bit will be set in both registers.

IPR Interrupt Pending Register **0x(FF)FFF310**

	BIT 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	BIT 16
	USB	I2C	MMC SD/M S	PK	PM	PN	PP	A/D C	EMIQ	RTI	CSPI	PR	IRQ6	IRQ3	IRQ2	IRQ1
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	BIT 0
	DMA1	DMA2	PWM 2	UART 2	PD	PE	PF	PG	PWM 1	PJ	TMR2	RTC	WDT	UART 1	TMR1	LCDC
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 15-7. Interrupt Pending Register Description

Name	Description	Settings
USB Bit 31	**USB Interrupt Pending**—When set, this bit indicates an interrupt event from USB.	0 = No USB interrupt is pending. 1 = An USB interrupt is pending.
I2C Bit 30	**I2C Interrupt Pending**—When set, this bit indicates an interrupt event from I2C.	0 = No I2C interrupt is pending. 1 = An I2C interrupt is pending.
MMCSD/MS Bit 29	**MMCSD/MS Interrupt Pending**—When set, this bit indicates an interrupt event from MMCSD/MS unit.	0 = No MMCSD/MS interrupt is pending. 1 = An MMCSD/MS interrupt is pending.
PK Bit 28	**PK Interrupt Pending**—When set, this bit indicates an interrupt event from Port K.	0 = No PK interrupt is pending. 1 = An PK interrupt is pending.
PM Bit 27	**PM Interrupt Pending**—When set, this bit indicates an interrupt event from Port M.	0 = No PM interrupt is pending. 1 = An PM interrupt is pending.

Table 15-7. Interrupt Pending Register Description (Continued)

Name	Description	Settings
PN Bit 26	**PN Interrupt Pending**—When set, this bit indicates an interrupt event from Port N.	0 = No PN interrupt is pending. 1 = An PN interrupt is pending.
PP Bit 25	**PP Interrupt Pending**—When set, this bit indicates an interrupt event from Port P.	0 = No PP interrupt is pending. 1 = An PP interrupt is pending.
A/D C Bit 24	**A/D Converter Interrupt Pending**—When set, this bit indicates an interrupt event from A/D Converter unit.	0 = No A/D C interrupt is pending. 1 = An A/D C interrupt is pending.
EMIQ Bit 23	**Emulator Interrupt Pending**—When set, this bit indicates that the in-circuit emulation module or $\overline{\text{EMUIRQ}}$ pin is requesting an interrupt on level 7. This bit can be generated from three interrupt sources: two breakpoint interrupts from the in-circuit emulation module and an external interrupt from $\overline{\text{EMUIRQ}}$, which is an active low, edge-sensitive interrupt. To clear this interrupt, you must read the ICEMSR register to identify the interrupt source and write a 1 to the corresponding bit of that register. See Section 16.2.4, “In-Circuit Emulation Module Status Register,” on page 16-8 for more information.	0 = No emulator interrupt is pending. 1 = An emulator interrupt is pending.
RTI Bit 22	**Real-Time Interrupt Pending (Real-Time Clock)**—When set, this bit indicates that the real-time timer interrupt is pending. The frequency can be selected inside the real-time clock module, which can function as an additional timer.	0 = No real-time timer interrupt is pending. 1 = A real-time timer interrupt is pending.
CSPI Bit 21	**CSPI Interrupt Pending**—When set, this bit indicates an interrupt event from CSPI unit.	0 = No CSPI interrupt is pending. 1 = An CSPI interrupt is pending.
PR Bit 20	**PR Interrupt Pending**—When set, this bit indicates an interrupt event from Port R.	0 = No PR interrupt is pending. 1 = An PR interrupt is pending.
IRQ6 Bit 19	**Interrupt Request Level 6**—This bit, when set, indicates that an external device is requesting an interrupt on level 6. If the $\overline{\text{IRQ6}}$ signal is set to be a level-sensitive interrupt, the source of the interrupt must first be cleared. If $\overline{\text{IRQ6}}$ is set to be an edge-triggered interrupt, the interrupt must be cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.	0 = No level 6 interrupt is pending. 1 = A level 6 interrupt is pending.
IRQ3 Bit 18	**Interrupt Request Level 3**—This bit, when set, indicates that an external device is requesting an interrupt on level 3. If the $\overline{\text{IRQ3}}$ signal is set to be a level-sensitive interrupt, the source of the interrupt must first be cleared. If $\overline{\text{IRQ3}}$ is set to be an edge-triggered interrupt, the interrupt must be cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.	0 = No level 3 interrupt is pending. 1 = A level 3 interrupt is pending.

Table 15-7. Interrupt Pending Register Description (Continued)

Name	Description	Settings
IRQ2 Bit 17	**Interrupt Request Level 2**—This bit, when set, indicates that an external device is requesting an interrupt on level 2. If the $\overline{\text{IRQ2}}$ signal is set to be a level-sensitive interrupt, the source of the interrupt must first be cleared. If $\overline{\text{IRQ2}}$ is set to be an edge-triggered interrupt, the interrupt must be cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.	0 = No level 2 interrupt is pending. 1 = A level 2 interrupt is pending.
IRQ1 Bit 16	**Interrupt Request Level 1**—This bit, when set, indicates that an external device is requesting an interrupt on level 1. If the $\overline{\text{IRQ1}}$ signal is set to be a level-sensitive interrupt, the source of the interrupt must first be cleared. If $\overline{\text{IRQ1}}$ is set to be an edge-triggered interrupt, the interrupt must be cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.	0 = No level 1 interrupt is pending. 1 = A level 1 interrupt is pending.
DMA1 Bits 15	**DMA1 Interrupt**—This bit, when set, indicates that there is a DMA Transfer Complete interrupt event from DMA unit pending.	0 = No DMA1 interrupt is pending. 1 = A DMA1 interrupt is pending.
DMA2 Bits 14	**DMA2 Interrupt**—This bit, when set, indicates that there is a DMA Transfer Error interrupt event from DMA unit pending.	0 = No DMA2 interrupt is pending. 1 = A DMA2 interrupt is pending.
PWM2 Bit 13	**Pulse-Width Modulator (PWM) 2 Interrupt**—This bit, when set, indicates that there is an interrupt event from PWM unit 2 pending.	0 = No PWM2 interrupt is pending. 1 = A PWM 2 interrupt is pending.
UART2 Bit 12	**UART 2 Interrupt Request**—When this bit is set, it indicates that the UART 2 module needs service. The interrupt level is configurable from level 1 to level 6. See Section 9.6.6, “Interrupt Level Register,” for more details.	0 = No UART 2 interrupt request is pending. 1 = UART 2 interrupt request is pending.
PD Bit 11	**PD Interrupt Pending**—When set, this bit indicates an interrupt event from Port D.	0 = No PD interrupt is pending. 1 = An PD interrupt is pending.
PE Bit 10	**PE Interrupt Pending**—When set, this bit indicates an interrupt event from Port E.	0 = No PE interrupt is pending. 1 = An PE interrupt is pending.
PF Bit 9	**PF Interrupt Pending**—When set, this bit indicates an interrupt event from Port F.	0 = No PF interrupt is pending. 1 = An PF interrupt is pending.
PG Bit 8	**PG Interrupt Pending**—When set, this bit indicates an interrupt event from Port G.	0 = No PG interrupt is pending. 1 = An PG interrupt is pending.

Table 15-7. Interrupt Pending Register Description (Continued)

Name	Description	Settings
PWM1 Bit 7	**Pulse-Width Modulator (PWM) 1 Interrupt**—This bit, when set, indicates that there is a level 6 interrupt event from PWM unit 1 pending.	0 = No PWM1 interrupt is pending. 1 = A PWM 1interrupt is pending.
PJ Bit 6	**PJ Interrupt Pending**—When set, this bit indicates an interrupt event from Port J.	0 = No PJ interrupt is pending. 1 = An PJ interrupt is pending.
TMR2 Bit 5	**Timer 2 Interrupt Pending**—This bit indicates that a timer 2 event has occurred. This is a level 4 interrupt.	0 = No timer 2 event occurred. 1 = A timer 2 event has occurred.
RTC Bit 4	**Real-Time Clock Interrupt Request**—This bit, when set, indicates that there is a level 4 interrupt event from the real-time clock that is pending.	0 = No real-time clock interrupt is pending. 1 = A real-time clock interrupt is pending.
WDT Bit 3	**Watchdog Timer Interrupt Request**—This bit indicates that a watchdog timer interrupt is pending. This is a level 4 interrupt.	0 =No watchdog timer interrupt is pending. 1 = A watchdog timer interrupt is pending.
UART1 Bit 2	**UART 1 Interrupt Request**—When this bit is set, it indicates that the UART 1 module needs service. This is a level 4 interrupt.	0 = No UART 1 service request is pending. 1 = UART 1 service is needed.
TMR1 Bit 1	**Timer 1 Interrupt Pending**—This bit indicates that a timer 1 event has occurred. This is a level 6 interrupt.	0 = No timer 1 event occurred. 1 = A timer 1 event has occurred.
LCDC Bit 0	**LCDC Interrupt**—This bit indicates an interrupt event from LCDC unit is pending. The interrupt level is configurable from level 1 to level 6. See Section 9.6.6, “Interrupt Level Register,” for more details.	0 = No LCDC interrupt is pending. 1 = A LCDC interrupt is pending.

15.8.6 Interrupt Level Register

Interrupts generated from all twenty-seven modules are level configurable. The interrupt level control register (ILCR) controls the interrupt level for these interrupts.

ILCR1 Interrupt Level Register 1 0x(FF)FFF314

	BIT15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
		CSPI_LEVEL				UART2_LEVEL				PWM2_LEVEL				TMR2_LEVEL		
TYPE		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw
RESET	0	1	1	0	0	1	0	1	0	0	1	1	0	0	1	1
	0x6533															

ILCR2 Interrupt Level Register 2 0x(FF)FFF316

	BIT15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
		I2C_LEVEL				MMCSD_LEVEL/ MS_LEVEL				LCDC_LEVEL				ADC_LEVEL		
TYPE		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw
RESET	0	1	0	0	0	1	0	1	0	1	0	1	0	0	1	1
	0x4553															

ILCR3 Interrupt Level Register 3 0x(FF)FFF318

	BIT15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
		RTI_LEVEL				DMA1_LEVEL				DMA2_LEVEL				PWM1_LEVEL		
TYPE		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw
RESET	0	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0
	0x4666															

ILCR4 Interrupt Level Register 4 0x(FF)FFF31A

	BIT15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
		PK_LEVEL				RTC_LEVEL				WDT_LEVEL				UART1_LEVEL		
TYPE		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw
RESET	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
	0x4444															

ILCR5 Interrupt Level Register 5 0x(FF)FFF31C

	BIT15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
		PM_LEVEL				PN_LEVEL				PP_LEVEL				PD_LEVEL		
TYPE		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw
RESET	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
	0x4444															

ILCR6 Interrupt Level Register 6 **0x(FF)FFF31E**

	BIT15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
		PE_LEVEL				PF_LEVEL				PG_LEVEL				PJ_LEVEL		
TYPE		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw
RESET	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
	0x4444															

ILCR7 Interrupt Level Register 7 **0x(FF)FFF320**

	BIT15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
						USB_LEVEL				PR_LEVEL				TMR1_LEVEL		
TYPE						rw	rw	rw		rw	rw	rw		rw	rw	rw
RESET	0	0	0	0	0	1	1	0	0	1	0	0	0	1	1	0
	0x0646															

Programming register bits 14–12, 10–8, 6–4, and 2–0 with the values shown in Table 15-8 causes the corresponding interrupt source to generate different interrupt levels.

Table 15-8. Interrupt Level Register Field Values

Interrupt Level	Value in Register Bits 14–12, 10–8, 6–4, and 2–0
Undefined level	111
Level 6	110
Level 5	101
Level 4	100
Level 3	011
Level 2	010
Level 1	001
Undefined level	000
Note: Values 000 and 111 are not allowed to be programmed into these register bits.	

After reset, each of these twenty-seven interrupts are set to the default level indicated in Table 15-9.

Table 15-9. Interrupt Default After Reset

Interrupt	Level
$\overline{\text{I2CIRQ}}$	4
$\overline{\text{MMCSDIRQ}}$/$\overline{\text{MSIRQ}}$	5
$\overline{\text{PKIRQ}}$	4
$\overline{\text{PMIRQ}}$	4

Table 15-9. Interrupt Default After Reset (Continued)

Interrupt	Level
$\overline{\text{PNIRQ}}$	4
$\overline{\text{PPIRQ}}$	4
$\overline{\text{A/DCIRQ}}$	3
$\overline{\text{RTIIRQ}}$	4
$\overline{\text{CSPIIRQ}}$	6
$\overline{\text{PRIRQ}}$	4
$\overline{\text{DMA1IRQ}}$	6
$\overline{\text{DMA2IRQ}}$	6
$\overline{\text{PWM2IRQ}}$	3
$\overline{\text{UART2IRQ}}$	5
$\overline{\text{PDIRQ}}$	4
$\overline{\text{PEIRQ}}$	4
$\overline{\text{PFIRQ}}$	4
$\overline{\text{PGIRQ}}$	4
$\overline{\text{PWM1IRQ}}$	6
$\overline{\text{PJIRQ}}$	4
$\overline{\text{TIMER2IRQ}}$	3
$\overline{\text{RTCIRQ}}$	4
$\overline{\text{WDTIRQ}}$	4
$\overline{\text{UART1IRQ}}$	4
$\overline{\text{TIMER1IRQ}}$	6
$\overline{\text{LCDCIRQ}}$	5
$\overline{\text{USBIRQ}}$	6

Freescale Semiconductor, Inc.

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Chapter 16
General Purpose I/O Module

This chapter describes the 12 multipurpose ports of the MC68SZ328. It also describes how to use the ports for external I/O control and to determine the status of the external signals. All 12 ports (B–G, J, K, M, N, P, and R) are programmable I/O ports with pull-up and pull-down capability. Each port can be used as a general-purpose I/O (GPIO) port, or it can be connected to its dedicated I/O function. Every signal line connects to an external pin. The pin name reflects the functions assigned to the pin. For example, the name PB/$\overline{\text{CSB1}}$/$\overline{\text{SDWE}}$ indicates that the pin is used for any of three separate signals: Port B data, Chip-Select B 1, and SDRAM Write-Enable. This chapter describes pin assignments either programmed as GPIO or programmed to dedicated I/O functions.

NOTE:

Some pins have been multiplexed with more than 2 functions. See Table 16-1.

Table 16-1. Multiplexing Pin Function Configuration

Pin Name	Function	Configuration
$\overline{\text{CSD0}}$/$\overline{\text{DMA\_REQ0}}$	CSD0	1. The pin is programmed as dedicated I/O and DIR4 = 1 in Port B. 2. The REN bit is disabled in the Memory Channel Control register 0 in DMA module. 3. Chip-select enable bit is enabled in chip-select register D in the CS module.
	DMA_REQ0	1. The pin is programmed as dedicated I/O and DIR4 = 0 in Port B. 2. The REN bit is enabled in the Memory Channel Control register 1 in DMA module.
$\overline{\text{CSD1}}$ /$\overline{\text{DMA\_REQ1}}$	CSD1	1. The pin is programmed as dedicated I/O and DIR5 = 1 in Port B. 2. The REN bit is disabled in the Memory Channel Control register 1 in DMA module. 3. Chip-select enabled bit is enable in chip-select register D in the CS module.
	DMA_REQ1	1. The pin is programmed as dedicated I/O and DIR5 = 0 in Port B. 2. The REN bit is enabled in the Memory Channel Control register 1 in DMA module.

Table 16-1. Multiplexing Pin Function Configuration (Continued)

Pin Name	Function	Configuration
$\overline{CSC0}$/$\overline{UDS}$	CSC0	1. The pin is programmed as dedicated I/O. 2. Chip-select enabled bit is enable in chip-select register C in the CS module.
	UDS	1. The pin is programmed as dedicated I/O. 2. Chip-select enabled bit is disable in chip-select register C in the CS module.
$\overline{CSC1}$/$\overline{LDS}$	CSC1	1. The pin is programmed as dedicated I/O. 2. Chip-select enabled bit is enable in chip-select register C in the CS module.
	LDS	1. The pin is programmed as dedicated I/O. 2. Chip-select enabled bit is disabled in chip-select register C in the CS module.
TIN/TOUT	TIN	1. The pin is programmed as dedicated I/O. 2. Bit 6 is set to 0 in the PBDIR register
	TOUT	1. The pin is programmed as dedicated I/O. 2. Bit 6 is set to 1 in the PBDIR register
DATA_READY/PWM2	DATA_READY	1. The pin is programmed as dedicated I/O. 2. Bit 0 is set to 0 in the PKDIR register
	PWM2	1. The pin is programmed as dedicated I/O. 2. Bit 0 is set to 1 in the PKDIR register

When pins are programmed as GPIOs, the direction of individual pins (input or output) can be configured, and pull-up resistors (or the pull-down resistor on port R) can be enabled or disabled. When pins are programmed as dedicated I/Os, a pin's direction cannot be controlled. A few exceptions to this rule are noted in the programming information about the specific ports.

16.1 Port Configuration

Every port is multiplexed with at least one other dedicated I/O function. Twelve ports have pins that can be configured for one of several dedicated I/O functions. Table 16-2 shows the I/O functions available for each port.

Ports are programmed by eight dedicated 8-bit registers: direction, data, pull-up enable, select, interrupt enable, interrupt status, interrupt polarity and interrupt edge. Port B and Port C have no interrupt capability. The remaining registers have select registers controlling whether the pin is assigned as a GPIO or a dedicated I/O function. Some pins have multiple dedicated functions assigned to them. Selection of these functions is controlled by other registers in the MC68SZ328.

The I/O drive control register (IOCR) in the system controller governs the drive strength (in mA) of all I/O signals, including all of the ports. By default, all I/O pins on the MC68SZ328 default to a 4 mA driving current. After reset, it is recommended that the user select a 4 mA drive strength for those signals not requiring high current to ensure maximum power savings.

Table 16-2. Dedicated I/O Functions of Ports

Port	Dedicated I/O Module	Dedicated I/O Module	Dedicated I/O Module	Dedicated I/O Module
B	Chip-select/ DMA signals Bus control	GP timers	PWM output	
C	LCD controller			
D	Sharp panel signals	Interrupt controller		
E	I^2C signals	Address bit 24	UART1	
F	LCD contrast	GP Timer2	CGM	Address bits 23–20
	Chip-select			
G	Bus control	Address bit 0	In-circuit emulation	Chip-select/ DRAM controller
J	CSPI	UART2		
K	CSPI/PWMO2	Bus control	DRAM controller	LCD controller
M	DRAM controller			
N	USBD controller			
P	LCD controller			
R	MMC/SD controller	Memory Stick host controller		

16.2 Status of I/O Ports During Reset

Two types of resets affect the states of the MC68SZ328's I/O ports: warm reset and power-up reset. A warm reset refers to any reset initiated when power to the processor remains uninterrupted. A power-up reset occurs the first time power is supplied to the MC68SZ328. Power-up resets are also called cold start resets.

Table 16-3 summarizes the behavior of all MC68SZ328 I/O ports during the Reset Assertion Time Length for power-up resets and warm resets.

Table 16-3. MC68SZ328 I/O Port Status During the Reset Assertion Time Length

I/O Ports	Warm Reset	Power-up Reset
B	Resets to default state	Resets to default state
C	Resets to default state	Resets to default state
D	Resets to default state	Resets to default state
E	Resets to default state	Resets to default state
F	Resets to default state	Resets to default state

Table 16-3. MC68SZ328 I/O Port Status During the Reset Assertion Time Length (Continued)

I/O Ports	Warm Reset	Power-up Reset
G	Maintains previous state	Resets to default state
J	Resets to default state	Resets to default state
K	Maintains previous state	Resets to default state
M	Maintains previous state	Unknown state
N	Resets to default state	Resets to default state
P	Resets to default state	Resets to default state
R	Resets to default state	Resets to default state

Note: The default state is defined by the reset values of the corresponding I/O port's registers. Please refer to Table 16-2 on page 16-3 and Table 16-3 on page 16-3 for details.

16.3 I/O Port Operation

The following subsections describe details of the I/O ports' operation.

16.3.1 Data Flow from the I/O Module

The operation of a port connected to another module in the MC68SZ328 is shown in Figure 16-1.

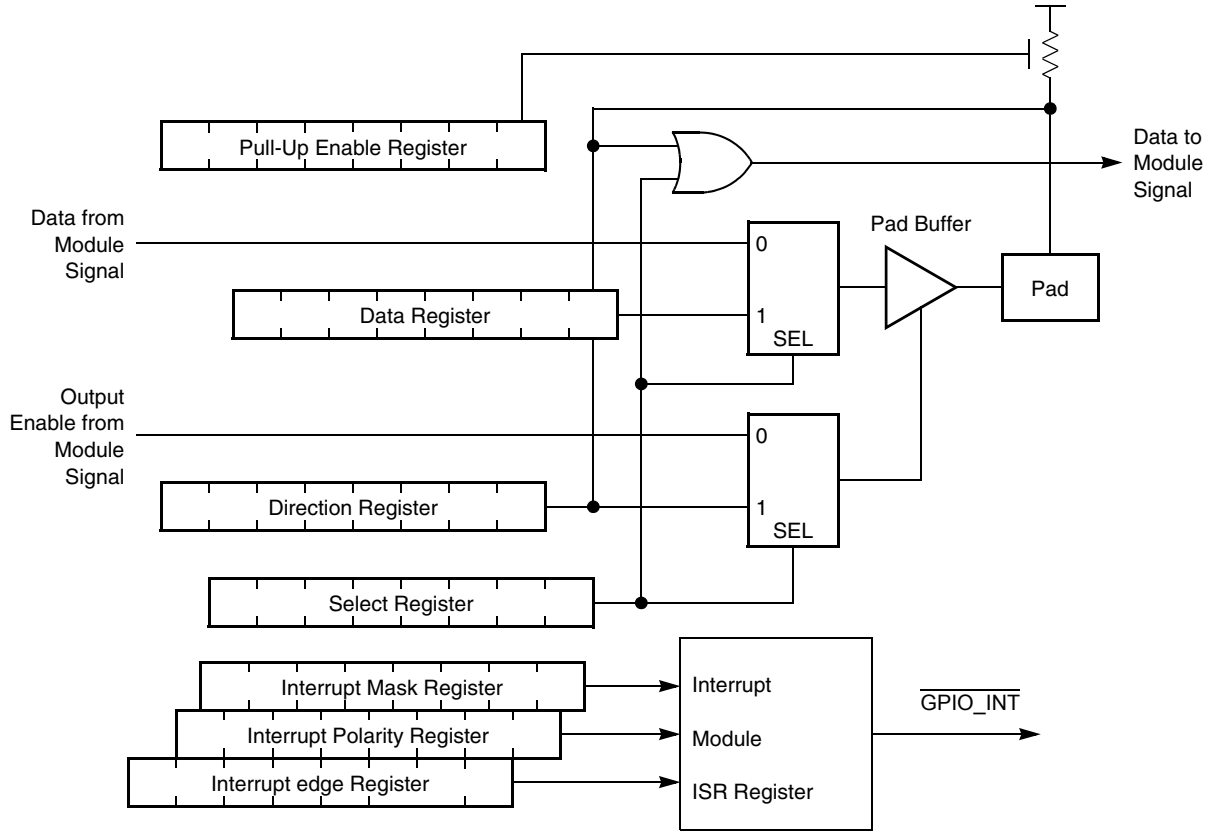


Figure 16-1. I/O Port Operation

For example, if Figure 16-1 represents the D0 bit of Port E, when the SEL5 in the select register is cleared, the “data from module” line is connected to the UART1 module’s TXD1 signal. Because TXD1 is output-only, the MC68SZ328 asserts the “output enable from module” line, thus enabling the output and disabling the “data to module” line. As long as the SELx bit of the port’s select register is clear (the default is set at reset), the UART1 module pin function is enabled. When the dedicated module controls the port, the direction register is ignored. There are a few exceptions that are described in the individual port programming sections that follow.

16.3.2 Data Flow to the I/O Module

An example of data flow to the I/O module is the D4 bit of Port E. This signal’s function is the UART1’s RXD1 signal. In this case, RXD1 is input-only, so the chip negates the “output enable from module” line, and the “data from module” line is not disabled (see Figure 16-1). The “data to module” signal is connected to the RXD1 input of the UART1.

16.3.3 Operating a Port as GPIO

When the SELx bit is set (if the DIRx bit of the PxDIR is 1), data written to the port’s data register is presented to the pin. If the DIRx bit in the direction register is 0 (input), data present on the pin is sampled and presented to the CPU when a read cycle is executed. When the DIRx bit is 0 (output), the actual pin

level is presented during write accesses. This may not be the same as the data that was written if the pin is overdriven. To prevent data loss when changing from one mode to another, the intended data should be written to the PxDATA register before entering the selected mode.

16.3.4 Port Pull-Up and Pull-Down Resistors

The pull-down on R[0] and the pull-up resistors on the other I/Os are enabled by setting the pull-up or pull-down enable register's bits to 1. Pull-up and pull-down resistors can be selected individually regardless of whether the I/O port is selected or not. After reset, Ports F[6:3], G[7:4 & 2:0], J[4], and K[7:0] default to their dedication function. All other ports default to their I/O function. All internal pull-up/pull-down resistors are enabled by default. Only Port R[0] has a pull-down resistor.

16.4 Programming Model

The chapter's remaining sections provide programming information about individual ports.

16.4.1 Port B Registers

Port B consists of the following 8-bit general-purpose I/O registers:

- Port B direction register (PBDIR)
- Port B data register (PBDATA)
- Port B pull-up enable register (PBPUEN)
- Port B select register (PBSEL)

Each signal line connects to an external pin. Each bit on Port B is individually configured.

16.4.1.1 Port B Direction Register

The Port B direction register controls the direction (input or output) of the line associated with the PBDATA bit position. When the data bit is assigned to a dedicated I/O function, the direction bits are ignored. The settings for the bit positions are shown in Table 16-4.

PBDIR Port B Direction Register **0x(FF)FFF408**

	BIT 7	6	5	4	3	2	1	BIT 0
	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-4. Port B Direction Register Description

Name	Description	Setting
DIRx Bits 7–0	**Direction**—This field controls the direction of the pins. They reset to 0. With the exception of bit 6, if a bit is selected as a dedicated I/O in PBSEL, the DIR bit is ignored.	0 = Inputs 1 = Output

16.4.1.2 Port B Data Register

The settings for the PBDATA bit positions are shown in Table 16-5.

PBDATA Port B Data Register **0x(FF)FFF409**

	BIT 7	6	5	4	3	2	1	BIT 0
	D7	D6	D5	D4	D3	D2	D1	D0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
	1	1	1	1	1	1	1	1
RESET	0xFF*							

*Actual bit value depends on external circuits connected to pin.

Table 16-5. Port B Data Register Description

Name	Description	Setting
Dx Bits 7–0	**Data**—This field reflects the status of the I/O signal in an 8-bit system.	0 = Drives the output signal low when DIRx is set to 1 or the external signal is low when DIRx is set to 0. 1 = Drives the output signal high when DIRx is set to 1 or the external signal is high when DIRx is set to 0.

Port B is multiplexed with chip-select, TIN/TOUT, and PWM dedicated I/O signals. These pins can be programmed as GPIO when these other assignments are not used.

This field controls or reports the data on the pins when the associated SELx bits are high. When the DIRx bits are high (output), the Dx bits control the pins. When the DIRx bits are low (input), the Dx bits report the signal driving the pins. The Dx bits can be written at any time. Bits that are configured as inputs accept the data, but the data written to each cannot be accessed until the corresponding pin is configured as an output. The actual value on the pin is reported when these bits are read, regardless of whether they are configured as input or output.

16.4.1.3 Port B Dedicated I/O Functions

The eight PBDATA lines are multiplexed with the chip-select, TIN/TOUT, and PWM dedicated I/O signals, these assignments are shown in Table 16-6 on page 16-7.

Bits 1–5 operate as chip-select signals.

The TIN/TOUT line can be specified as either timer-input or timer-output by programming bit 6 in the PBDIR register. Clearing the bit makes the line TIN. Setting the bit to 1 makes it TOUT. Unlike other port register pins, the TOUT/TIN/PB6 pin direction is still controlled by the DIR6 bit in the Port B register even though the pin is assigned to the GP timers. Refer to “TOUT/TIN/PB6 Pin,” for details about the operation and programming of the pin.

The PWMO1 signal is an output signal resulting from the logical operation (AND or OR) of both the PWM 1 and PWM 2 modules. Bits 3–2 (P[1:0]) of the peripheral control register (PCR) select the logic used for combining the modules. The PB7/PWMO1 pin defaults to a GPIO input pulled high. Refer to “Pulse-Width Modulator 1 and 2,” for additional information.

Table 16-6. Port B Dedicated Function Assignments

Bit	GPIO Function	Dedicated I/O Functions
0	Data bit 0	$\overline{\text{CSB0}}$

Table 16-6. Port B Dedicated Function Assignments (Continued)

Bit	GPIO Function	Dedicated I/O Functions
1	Data bit 1	CSB1
2	Data bit 2	CSC0/UDS
3	Data bit 3	CSC1/LDS
4	Data bit 4	CSD0/DMA_REQ0
5	Data bit 5	CSD1/DMA_REQ1
6	Data bit 6	TIN/TOUT
7	Data bit 7	PWMO1

16.4.1.4 Port B Pull-Up Enable Register

The Port B pull-up enable register (PBPUEN) controls the pull-up resistors for each line in Port B. The settings for the bit positions are shown in Table 16-7.

PBPUEN Port B Pull-Up Enable Register **0x(FF)FFF40A**

	BIT 7	6	5	4	3	2	1	BIT 0
	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-7. Port B Pull-Up Enable Register Description

Name	Description	Setting
PUx Bits 7–0	**Pull-Up**—This field enables the pull-up resistors on the port.	0 = Pull-up resistors are disabled. 1 = Pull-up resistors are enabled.

16.4.1.5 Port B Select Register

The Port B select register (PBSEL) determines if a bit position in the data register (PBDATA) is assigned as a general purpose I/O or to a dedicated I/O function. The settings for the bit positions are shown in Table 16-8.

PBSEL Port B Select Register **0x(FF)FFF40B**

	BIT 7	6	5	4	3	2	1	BIT 0
	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-8. Port B Select Register Description

Name	Description	Setting
SELx Bits 7–0	**Select** —This field selects whether the internal chip function or I/O port signals are connected to the pins.	0 = The dedicated function pins are connected. 1 = The I/O port function pins are connected.

16.4.2 Port C Registers

Port C consists of the following 8-bit general-purpose I/O registers:

- Port C direction register (PCDIR)
- Port C data register (PCDATA)
- Port C pull-up enable register (PCPDEN)
- Port C select register (PCSEL)

Each signal in the PCDATA register connects to an external pin. As with the other ports, each bit on Port C is individually configured.

16.4.2.1 Port C Direction Register

The Port C direction register controls the direction (input or output) of the line associated with the PCDATA bit position. When the data bit is assigned to a dedicated I/O function by the PCSEL register, the DIR bits are ignored. The settings for the bit positions are shown in Table 16-9.

PCDIR Port C Direction Register **0x(FF)FFF410**

	BIT 7	6	5	4	3	2	1	BIT 0
	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-9. Port C Direction Register Description

Name	Description	Setting
DIRx Bits 7–0	**Direction**—This field controls the direction of the pins in an 8-bit system. They reset to 0.	0 = Input 1 = Output

16.4.2.2 Port C Data Register

The settings for the PCDATA bit positions are shown in Table 16-10.

PCDATA Port C Data Register **0x(FF)FFF411**

	BIT 7	6	5	4	3	2	1	BIT 0
	D7	D6	D5	D4	D3	D2	D1	D0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
	0	0	0	0	0	0	0	0
RESET	0x00*							

*Actual bit value depends on external circuits connected to pin.

Table 16-10. Port C Data Register Description

Name	Description	Setting
Dx Bits 7–0	**Data**—This field reflects the status of the I/O signal.	0 = Drives the output signal low when DIRx is set to 1 or the external signal is low when DIRx is set to 0. 1 = Drives the output signal high when DIRx is set to 1 or the external signal is high when DIRx is set to 0.

Port C is primarily multiplexed with the LCD controller's signals. These pins can be programmed as GPIO when the LCD controller is not used. See "Connecting the LCD Controller to an LCD Panel," for more detailed information.

This field controls or reports the data on the pins when the associated SELx bits are high. When the DIRx bits are high (output), the Dx bits control the pins. When the DIRx bits are low (input), the Dx bits report the signal driving the pins. The Dx bits can be written at any time. Bits that are configured as inputs accept the data, but the data written to each cannot be accessed until the corresponding pin is configured as an output. The actual value on the pin is reported when these bits are read, regardless of whether they are configured as input or output.

16.4.2.3 Port C Dedicated I/O Functions

The eight PCDATA lines are multiplexed with the LCD controller dedicated I/O signals, these assignments are shown in Table 16-11.

Table 16-11. Port C Dedicated Function Assignments

Bit	GPIO Function	Dedicated I/O Function
0	Data bit 0	LD0
1	Data bit 1	LD1
2	Data bit 2	LD2
3	Data bit 3	LD3
4	Data bit 4	FLM/VSYNC
5	Data bit 5	LP/HSYNC
6	Data bit 6	SCLK

Table 16-11. Port C Dedicated Function Assignments (Continued)

Bit	GPIO Function	Dedicated I/O Function
7	Data bit 7	ACD/OE

16.4.2.4 Port C Pull-Up Enable Register

The Port C pull-up enable register (PCPUEN) controls the pull-up resistors for each line in Port C. The settings for the bit positions are shown in Table 16-12.

PCPUEN Port C Pull-Up Enable Register **0x(FF)FFF412**

	BIT 7	6	5	4	3	2	1	BIT 0
	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-12. Port C Pull-Up Enable Register Description

Name	Description	Setting
PUx Bits 7–0	**Pull-**Up—This field enables the pull-up resistors on the port.	0 = Pull-Up resistors are disabled. 1 = Pull-Up resistors are enabled.

16.4.2.5 Port C Select Register

The Port C select register (PCSEL) determines if a bit position in the Port C data register (PCDATA) is assigned as a GPIO or to a dedicated I/O function. The settings for the bit positions are shown in Table 16-13.

PCSEL Port C Select Register **0x(FF)FFF413**

	BIT 7	6	5	4	3	2	1	BIT 0
	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-13. Port C Select Register Description

Name	Description	Setting
SELx Bits 7–0	**Select**—This field selects whether the internal chip function or I/O port signals are connected to the pins.	0 = The dedicated function pins are connected. 1 = The I/O port function pins are connected.

16.4.3 Port D Registers

Port D consists of the following 8-bit general-purpose I/O registers:

- Port D direction register (PDDIR)
- Port D data register (PDDATA)
- Port D pull-up enable register (PDPUEN)
- Port D select register (PDSEL)
- Port D interrupt mask register (PDIMR)
- Port D interrupt status register (PDISR)
- Port D interrupt edge register (PDIER)
- Port D interrupt polarity register (PDIPR)

16.4.3.1 Port D Direction Register

The Port D direction register controls the direction (input or output) of the line associated with the PDDATA bit position. When the data bit is assigned to a dedicated I/O function by the PDSEL register, the DIR bits are ignored. The settings for the PDDIR bit positions are shown in Table 16-14.

PDDIR Port D Direction Register **0x(FF)FFF418**

	BIT 7	6	5	4	3	2	1	BIT 0
	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-14. Port D Direction Register Description

Name	Description	Setting
DIRx Bits 7–0	**Direction**—This field controls the direction of the pins in an 8-bit system. They reset to 0.	0 = Input 1 = Output

16.4.3.2 Port D Data Register

The settings for the PDDATA bit positions are shown in Table 16-15.

PDDATA Port D Data Register **0x(FF)FFF419**

	BIT 7	6	5	4	3	2	1	BIT 0
	D7	D6	D5	D4	D3	D2	D1	D0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
	1	1	1	1	1	1	1	1
RESET	0xFF*							

*Actual bit value depends on external circuits connected to pin.

Table 16-15. Port D Data Register Description

Name	Description	Setting
Dx Bits 7–0	**Data**—This field reflects the status of the I/O signal.	0 = Drives the output signal low when DIRx is set to 1 or the external signal is low when DIRx is set to 0. 1 = Drives the output signal high when DIRx is set to 1 or the external signal is high when DIRx is set to 0.

The eight PDDATA lines are multiplexed with the Sharp panel signals and $\overline{\text{IRQx}}$ dedicated I/O signals, these assignments are shown in Table 16-16. Port D signals can be programmed as GPIO when not used for handling external interrupts.

This field controls or reports the data on the pins when the associated SELx bits are high. When the DIRx bits are high (output), the Dx bits control the pins. When the DIRx bits are low (input), the Dx bits report the signal driving the pins. The Dx bits can be written at any time. Bits that are configured as inputs accept the data, but the data written to each cannot be accessed until the corresponding pin is configured as an output. The actual value on the pin is reported when these bits are read, regardless of whether they are configured as input or output.

16.4.3.3 Port D Dedicated I/O Functions

The eight PDDATA lines are multiplexed with the Sharp panel dedicated I/O signals and IRQ1, 2, 3, 6, these assignments are shown in Table 16-16.

Table 16-16. Port D Dedicated Function Assignments

Bit	GPIO Function	Dedicated I/O Function
0	Data bit 0	SPL/SPR
1	Data bit 1	PS
2	Data bit 2	CLS
3	Data bit 3	REV
4	Data bit 4	$\overline{\text{IRQ1}}$
5	Data bit 5	$\overline{\text{IRQ2}}$
6	Data bit 6	$\overline{\text{IRQ3}}$

Table 16-16. Port D Dedicated Function Assignments (Continued)

Bit	GPIO Function	Dedicated I/O Function
7	Data bit 7	$\overline{IRQ6}$

16.4.3.4 Port D Pull-Up Enable Register

The Port D pull-up enable register (PDPUEN) controls the pull-up resistors for each line in Port D. The settings for the bit positions in PDPUEN are shown in Table 16-17.

PDPUEN Port D Pull-Up Enable Register **0x(FF)FFF41A**

	BIT 7	6	5	4	3	2	1	BIT 0
	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-17. Port D Pull-Up Enable Register Description

Name	Description	Setting
PUx Bits 7–0	**Pull-Up**—This field enables the pull-up resistors on the port.	0 = Pull-up resistors are disabled. 1 = Pull-up resistors are enabled.

16.4.3.5 Port D Select Register

The Port D select register (PDSEL) determines if a bit position in the Port D data register (PDDATA) is assigned as a GPIO or to a dedicated I/O function. The settings for the bit positions of PDSEL are shown in Table 16-18.

PDSEL Port D Select Register **0x(FF)FFF41B**

	BIT 7	6	5	4	3	2	1	BIT 0
	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-18. Port D Select Register Description

Name	Description	Setting
SELx Bits 7–0	**Select**—This field selects whether the internal chip function or I/O port signals are connected to the pins.	0 = The dedicated function pins are connected. 1 = The I/O port function pins are connected.

16.4.3.6 Port D Interrupt Mask Register

The Port D interrupt mask register (PDIMR) controls the interrupt whether be masked or not be masked for each line in Port D. The settings for the bit positions are shown in Table 16-19.

PDIMR Port D Interrupt Mask Register **0x(FF)FFF41C**

	BIT 7	6	5	4	3	2	1	BIT 0
	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-19. Port D Interrupt Mask Register Description

Name	Description	Setting
IMx Bits 7–0	**Interrupt Mask**—This field masks the interrupt on the port.	0 = Interrupt is masked 1 = Interrupt is not masked

16.4.3.7 Port D Interrupt Status Register

The Port D interrupt status register (PDISR) display the interrupt status for each line in Port D. The settings for the bit positions are shown in Table 16-20.

PDISR Port D Interrupt Status Register **0x(FF)FFF41D**

	BIT 7	6	5	4	3	2	1	BIT 0
	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
TYPE	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-20. Port D Interrupt Status Register Description

Name	Description	Setting
ISx Bits 7–0	**Interrupt Status**—This field displays the interrupt status on the port.	0 = Interrupt not generated 1 = Interrupt generated

16.4.3.8 Port D Interrupt Edge Register

The Port D interrupt edge register (PDIER) controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt to configure for each line in Port D. The settings for the bit positions are shown in Table 16-21.

PDIER Port D Interrupt Edge Register **0x(FF)FFF41E**

	BIT 7	6	5	4	3	2	1	BIT 0
	EE7	EE6	EE5	EE4	EE3	EE2	EE1	EE0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-21. Port D Interrupt Edge Register Description

Name	Description	Setting
EEx Bits 7–0	**Edge Enable**—This field controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt.	0 = Edge-sensitive interrupt 1 = Level-sensitive interrupt

16.4.3.9 Port D Interrupt Polarity Register

The Port D interrupt polarity register (PDIPR) controls the interrupt sensitive polarity type to configure for each line in Port D. The settings for the bit positions are shown in Table 16-22.

PDIPR Port D Interrupt Polarity Register **0x(FF)FFF41F**

	BIT 7	6	5	4	3	2	1	BIT 0
	POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-22. Port D Interrupt Polarity Register Description

Name	Description	Setting
POLx Bits 7–0	**Polarity**—This field controls interrupt polarity.	0 = Positive polarity 1 = Negative polarity

16.4.4 Port E Registers

Port E consists of the following 8-bit general-purpose I/O registers:

- Port E direction register (PEDIR)
- Port E data register (PEDATA)
- Port E pull-up enable register (PEPUEN)
- Port E select register (PESEL)

- Port E interrupt mask register (PEIMR)
- Port E interrupt status register (PEISR)
- Port E interrupt edge register (PEIER)
- Port E interrupt polarity register (PEIPR)

Each signal in the PEDATA register connects to an external pin. As with the other ports, each bit on Port E is individually configured.

16.4.4.1 Port E Direction Register

The Port E direction register controls the direction (input or output) of the line associated with the PEDATA bit position. When the data bit is assigned to a dedicated I/O function by the PESEL register, the DIR bits are ignored. The settings for the bit positions of the PEDIR register are shown in Table 16-23.

PEDIR Port E Direction Register **0x(FF)FFF420**

	BIT 7	6	5	4	3	2	1	BIT 0
	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-23. Port E Direction Register Description

Name	Description	Setting
DIRx Bits 7–0	**Direction**—This field controls the direction of the pins in an 8-bit system. They reset to 0.	0 = Input 1 = Output

16.4.4.2 Port E Data Register

The settings for the bit positions of the PEDATA register are shown in Table 16-24.

PEDATA Port E Data Register **0x(FF)FFF421**

	BIT 7	6	5	4	3	2	1	BIT 0
	D7	D6	D5	D4	D3	D2	D1	D0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF*							

*Actual bit value depends on external circuits connected to pin.

Table 16-24. Port E Data Register Description

Name	Description	Setting
Dx Bits 7–0	**Data**—This field reflects the status of the I/O signal in an 8-bit system.	0 = Drives the output signal low when DIRx is set to 1 or the external signal is low when DIRx is set to 0. 1 = Drives the output signal high when DIRx is set to 1 or the external signal is high when DIRx is set to 0.

Port E is multiplexed with the UART1, and I^2C control signals. These pins can be programmed as GPIO when the UART1, and I^2C control features are not used. See "UART Signals" and "I^2C Control Signals," for more detailed information.

This field controls or reports the data on the pins when the associated SELx bits are high. When the DIRx bits are high (output), the Dx bits control the pins. When the DIRx bits are low (input), the Dx bits report the signal driving the pins. The Dx bits can be written at any time. Bits that are configured as inputs accept the data, but the data written to each cannot be accessed until the corresponding pin is configured as an output. The actual value on the pin is reported when these bits are read, regardless of whether they are configured as input or output.

16.4.4.3 Port E Dedicated I/O Functions

The eight PEDATA lines are multiplexed with the UART dedicated I/O signals, these assignments are shown in Table 16-25.

Table 16-25. Port E Dedicated Function Assignments

Bit	GPIO Function	Dedicated I/O Function
0	Data bit 0	SDA
1	Data bit 1	SCL
2	Data bit 2	A24
3	Data bit 3	UCLK
4	Data bit 4	RXD1
5	Data bit 5	TXD1
6	Data bit 6	$\overline{\text{RTS1}}$
7	Data bit 7	$\overline{\text{CTS1}}$

16.4.4.4 Port E Pull-Up Enable Register

The Port E pull-up enable register (PEPUEN) controls the pull-up resistors for each line in Port E. The settings for the bit positions of the PEPUEN register are shown in Table 16-26.

PEPUEN Port E Pull-Up Enable Register **0x(FF)FFF422**

	BIT 7	6	5	4	3	2	1	BIT 0
	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-26. Port E Pull-Up Enable Register Description

Name	Description	Setting
PUx Bits 7–0	**Pull-Up**—This field enables the pull-up resistors on the port.	0 = Pull-up resistors are disabled. 1 = Pull-up resistors are enabled.

16.4.4.5 Port E Select Register

The Port E select register (PESEL) determines if a bit position in the Port E data register (PEDATA) is assigned as a GPIO or to a dedicated I/O function. The settings for the bit positions of the PEDIR register are shown in Table 16-27.

PESEL Port E Select Register **0x(FF)FFF423**

	BIT 7	6	5	4	3	2	1	BIT 0
	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-27. Port E Select Register Description

Name	Description	Setting
SELx Bits 7–0	**Select**—This field selects whether the internal chip function or I/O port signals are connected to the pins.	0 = The dedicated function pins are connected. 1 = The I/O port function pins are connected.

16.4.4.6 Port E Interrupt Mask Register

The Port E interrupt mask register (PEIMR) controls the interrupt whether be masked or not be masked for each line in Port E. The settings for the bit positions are shown in Table 16-28.

PEIMR Port E Interrupt Mask Register **0x(FF)FFF424**

	BIT 7	6	5	4	3	2	1	BIT 0
	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-28. Port E Interrupt Mask Register Description

Name	Description	Setting
IMx Bits 7–0	**Interrupt Mask**—This field masks the interrupt on the port.	0 = Interrupt is masked 1 = Interrupt is not masked

16.4.4.7 Port E Interrupt Status Register

The Port E interrupt status register (PEISR) display the interrupt status for each line in Port E. The settings for the bit positions are shown in Table 16-29.

PEISR Port E Interrupt Status Register **0x(FF)FFF425**

	BIT 7	6	5	4	3	2	1	BIT 0
	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
TYPE	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-29. Port E Interrupt Status Register Description

Name	Description	Setting
ISx Bits 7–0	**Interrupt Status**—This field displays the interrupt status on the port.	0 = Interrupt not generated 1 = Interrupt generated

16.4.4.8 Port E Interrupt Edge Register

The Port E interrupt edge register (PEIER) controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt to configure for each line in Port E. The settings for the bit positions are shown in Table 16-30.

PEIER Port E Interrupt Edge Register **0x(FF)FFF426**

	BIT 7	6	5	4	3	2	1	BIT 0
	EE7	EE6	EE5	EE4	EE3	EE2	EE1	EE0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-30. Port E Interrupt Edge Register Description

Name	Description	Setting
EEx Bits 7–0	**Edge Enable**—This field controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt.	0 = Edge-sensitive interrupt 1 = Level-sensitive interrupt

16.4.4.9 Port E Interrupt Polarity Register

The Port E interrupt polarity register (PEIPR) controls the interrupt sensitive polarity type to configure for each line in Port E. The settings for the bit positions are shown in Table 16-31.

PEIPR Port E Interrupt Polarity Register **0x(FF)FFF427**

	BIT 7	6	5	4	3	2	1	BIT 0
	POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-31. Port E Interrupt Polarity Register Description

Name	Description	Setting
POLx Bits 7–0	**Polarity**—This field controls interrupt polarity.	0 = Positive polarity 1 = Negative polarity

16.4.5 Port F Registers

Port F consists of the following 8-bit general-purpose I/O registers:

- Port F direction register (PFDIR)
- Port F data register (PFDATA)
- Port F pull-up enable register (PFPUEN)
- Port F select register (PFSEL)
- Port F interrupt mask register (PFIMR)
- Port F interrupt status register (PFISR)
- Port F interrupt edge register (PFIER)
- Port F interrupt polarity register (PFIPR)

Each signal in the PFDATA register connects to an external pin. As on the other ports, each bit on Port F is individually configured.

16.4.5.1 Port F Direction Register

The Port F direction register controls the direction (input or output) of the line associated with the PFDATA bit position. When the data bit is assigned to a dedicated I/O function by the PFSEL register, the DIR bits are ignored. The settings for the PFDIR bit positions are shown in Table 16-32.

PFDIR Port F Direction Register **0x(FF)FFF428**

	BIT 7	6	5	4	3	2	1	BIT 0
	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-32. Port F Direction Register Description

Name	Description	Setting
DIRx Bits 7–0	**Direction**—This field controls the direction of the pins in an 8-bit system. They reset to 0.	0 = Input 1 = Output

16.4.5.2 Port F Data Register

The settings for the bit positions of the PFDATA register are shown in Table 16-33.

PFDATA Port F Data Register **0x(FF)FFF42**

	BIT 7	6	5	4	3	2	1	BIT 0
	D7	D6	D5	D4	D3	D2	D1	D0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF*							

*Actual bit value depends on external circuits connected to pin.

Table 16-33. Port F Data Register Description

Name	Description	Setting
Dx Bits 7–0	**Data**—This field reflects the status of the I/O signal in an 8-bit system.	0 = Drives the output signal low when DIRx is set to 1 or the external signal is low when DIRx is set to 0. 1 = Drives the output signal high when DIRx is set to 1 or the external signal is high when DIRx is set to 0.

Port F is multiplexed with address lines A[23:20] and several dedicated functions. These pins can be programmed as GPIO when the address bus and the dedicated I/O signals are not in use.

This field controls or reports the data on the pins when the associated SELx bits are high. When the DIRx bits are high (output), the Dx bits control the pins. When the DIRx bits are low (input), the Dx bits report the signal driving the pins. The Dx bits can be written at any time. Bits that are configured as inputs accept the data, but the data written to each cannot be accessed until the corresponding pin is configured as an output. The actual value on the pin is reported when these bits are read, regardless of whether they are configured as input or output.

16.4.5.3 Port F Dedicated I/O Functions

The eight PFDATA lines are multiplexed with the dedicated I/O signals, these assignments are shown in Table 16-34.

Table 16-34. Port F Dedicated I/O Function Assignments

Bit	GPIO Function	Dedicated I/O Function
0	Data bit 0	LCONTRAST
1	Data bit 1	TIN2/TOUT2
2	Data bit 2	CLKO
3	Data bit 3	A20
4	Data bit 4	A21
5	Data bit 5	A22
6	Data bit 6	A23
7	Data bit 7	$\overline{\text{CSA1}}$

The LCONTRAST function controls the pulse-width modulator (PWM) inside the LCD controller to adjust the supply voltage to the LCD panel.

The CLKO output clock signal is internally connected to the SYSCLK clock output of the internal CGM. This signal is provided for external reference. The output can be disabled to reduce power consumption and electromagnetic emission. This signal defaults to a PF2 input signal. See “CGM Operational Overview,” for more information about this signal.

Bit 7 is used for the chip-select signal $\overline{\text{CSA1}}$. See “Chip-Select Operation,” for detailed information.

16.4.5.4 Port F Pull-Up Enable Register

The Port F pull-up enable register (PFPUEN) controls the pull-up resistors for each line in Port F. The settings for the PFPUEN bit positions are shown in Table 16-35.

PFPUEN Port F Pull-Up Enable Register **0x(FF)FFF42A**

	BIT 7	6	5	4	3	2	1	BIT 0
	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-35. Port F Pull-Up Enable Register Description

Name	Description	Setting
PU7 Bit 7–0	**Pull-Up**—These bits enable the pull-up resistor on the port.	0 = Pull-up resistor is disabled. 1 = Pull-up resistor is enabled.

16.4.5.5 Port F Select Register

The Port F select register (PFSEL) determines if a bit position in the data register (PFDATA) is assigned as a GPIO or to a dedicated I/O function. The settings for the PFSEL bit positions are shown in Table 16-36.

PFSEL Port F Select Register **0x(FF)FFF42B**

	BIT 7	6	5	4	3	2	1	BIT 0
	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	0	0	0	0	1	1	1
	0x87							

Table 16-36. Port F Select Register Description

Name	Description	Setting
SELx Bits 7–0	**Select**—This field selects whether the internal chip function or I/O port signals are connected to the pins.	0 = The dedicated function pins are connected. 1 = The I/O port function pins are connected.

16.4.5.6 Port F Interrupt Mask Register

The Port F interrupt mask register (PFIMR) controls the interrupt whether be masked or not be masked for each line in Port F. The settings for the bit positions are shown in Table 16-37.

PFIMR Port F Interrupt Mask Register **0x(FF)FFF42C**

	BIT 7	6	5	4	3	2	1	BIT 0
	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-37. Port F Interrupt Mask Register Description

Name	Description	Setting
IMx Bits 7–0	**Interrupt Mask**—This field masks the interrupt on the port.	0 = Interrupt is masked 1 = Interrupt is not masked

16.4.5.7 Port F Interrupt Status Register

The Port F interrupt status register (PFISR) display the interrupt status for each line in Port F. The settings for the bit positions are shown in Table 16-38.

PFISR Port F Interrupt Status Register **0x(FF)FFF42D**

	BIT 7	6	5	4	3	2	1	BIT 0
	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
TYPE	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-38. Port F Interrupt Status Register Description

Name	Description	Setting
ISx Bits 7–0	**Interrupt Status**—This field displays the interrupt status on the port.	0 = Interrupt not generated 1 = Interrupt generated

16.4.5.8 Port F Interrupt Edge Register

The Port F interrupt edge register (PFIER) controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt to configure for each line in Port F. The settings for the bit positions are shown in Table 16-39.

PFIER Port F Interrupt Edge Register **0x(FF)FFF42E**

	BIT 7	6	5	4	3	2	1	BIT 0
	EE7	EE6	EE5	EE4	EE3	EE2	EE1	EE0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-39. Port F Interrupt Edge Register Description

Name	Description	Setting
EEx Bits 7–0	**Edge Enable**—This field controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt.	0 = Edge-sensitive interrupt 1 = Level-sensitive interrupt

16.4.5.9 Port F Interrupt Polarity Register

The Port F interrupt polarity register (PFIPR) controls the interrupt sensitive polarity type to configure for each line in Port F. The settings for the bit positions are shown in Table 16-40.

PFIPR Port F Interrupt Polarity Register **0x(FF)FFF42F**

	BIT 7	6	5	4	3	2	1	BIT 0
	POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-40. Port F Interrupt Polarity Register Description

Name	Description	Setting
POLx Bits 7–0	**Polarity**—This field controls interrupt polarity.	0 = Positive polarity 1 = Negative polarity

16.4.6 Port G Registers

Port G is comprised of the following 8-bit general-purpose I/O registers:

- Port G direction register (PGDIR)
- Port G data register (PGDATA)
- Port G pull-up enable register (PGPUEN)
- Port G select register (PGSEL)
- Port G interrupt mask register (PGIMR)
- Port G interrupt status register (PGISR)
- Port G interrupt edge register (PGIER)
- Port G interrupt polarity register (PGIPR)

Each signal in the PGDATA register connects to an external pin. Port G provides a total of six pins, and each bit is individually configured.

16.4.6.1 Port G Direction Register

The Port G direction register controls the direction (input or output) of the line associated with the PGDATA bit position. When the data bit is assigned to a dedicated I/O function by the PGSEL register, the DIR bits are ignored. The settings for the PGDIR bit positions are shown in Table 16-41.

PGDIR Port G Direction Register **0x(FF)FFF430**

	BIT 7	6	5	4	3	2	1	BIT 0
	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-41. Port G Direction Register Description

Name	Description	Setting
DIRx Bits 7–0	**Direction**—This field controls the direction of the pins in an 8-bit system. They reset to 0.	0 = Input 1 = Output

16.4.6.2 Port G Data Register

The settings for the bit positions of the PGDATA register are shown in Table 16-42.

PGDATA Port G Data Register **0x(FF)FFF431**

	BIT 7	6	5	4	3	2	1	BIT 0
	D7	D6	D5	D4	D3	D2	D1	D0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF*							

*Actual bit value depends on external circuits connected to pin.

Table 16-42. Port G Data Register Description

Name	Description	Setting
Dx Bits 7–0	**Data**—This field reflects the status of the I/O signal in an 8-bit system.	0 = Drives the output signal low when DIRx is set to 1 or the external signal is low when DIRx is set to 0. 1 = Drives the output signal high when DIRx is set to 1 or the external signal is high when DIRx is set to 0.

Port G is multiplexed with address line A0 and several dedicated I/O functions. These pins can be programmed as GPIO when the address bus and the dedicated I/O signals are not in use.

All of the bits control or report the data on the pins when the associated SELx bits are high. When the DIRx bits are high (output), the Dx bits control the pins. When the DIRx bits are low (input), the Dx bits report the signal driving the pins. The Dx bits can be written at any time. Bits that are configured as inputs accept the data, but the data written to each cannot be accessed until the corresponding pin is configured as an output. The actual value on the pin is reported when these bits are read, regardless of whether they are configured as input or output. See Table 16-43 for information about setting the bits in the PGDIR register.

16.4.6.3 Port G Dedicated I/O Functions

The six PGDATA lines are multiplexed with the dedicated I/O signals, these assignments are shown in Table 16-43.

Table 16-43. Port G Dedicated I/O Function Assignments

Bit	GPIO Function	Dedicated I/O Function
0	Data bit 0	BUSW/$\overline{\text{DTACK}}$
1	Data bit 1	A0/MA0
2	Data bit 2	$\overline{\text{EMUIRQ}}$
3	Data bit 3	P/$\overline{\text{D}}$
4	Data bit 4	$\overline{\text{EMUCS}}$
5	Data bit 5	$\overline{\text{EMUBRK}}$
6	Data bit 6	$\overline{\text{CSE}}$/$\overline{\text{SDCS0}}$/$\overline{\text{RAS0}}$
7	Data bit 7	$\overline{\text{CSF}}$/$\overline{\text{SDCS1}}$/$\overline{\text{RAS1}}$

BUSW is the default bus width for the $\overline{\text{CSA0}}$ signal. The $\overline{\text{DTACK}}$ signal is the external input data acknowledge signal. The MC68SZ328 microprocessor will latch the BUSW signal at the rising edge of the Reset signal. Its mode determines the default bus width for $\overline{\text{CSA0}}$. Bit 1 is Address 0. After system reset, this signal defaults to A0.

Bit 3 is P/$\overline{\text{D}}$ (Program/Data). During system reset, a logic low of this input signal will put the MC68SZ328 into Hi-Z mode, in which all MC68SZ328 pins are three-stated after reset release. For normal operation, this pin must be pulled high during system reset or left unconnected. This pin defaults to a GPIO input pulled high, but can be programmed as the P/$\overline{\text{D}}$ function. P/$\overline{\text{D}}$ is a status signal used in conjunction with in-circuit emulation that shows whether the current bus cycle is in program space or in data space during emulation mode. The remaining bits are dedicated in-circuit emulation controls. See “In-Circuit Emulation,” for detailed information on their operation.

16.4.6.4 Port G Operational Considerations

Port G can be used as a GPIO as long as caution is exercised. After reset, the Port G pins default to the dedicated function, except bit 3, which has an I/O function. To ensure normal operation, the $\overline{\text{EMUIRQ}}$ and $\overline{\text{EMUBRK}}$ pins must stay high or not be connected during system reset. Otherwise, the chip will enter emulation mode.

When bits 2–5 are used as I/O, the emulation mode cannot be used during development and debugging. Once development is complete, bits 2–5 can be used as I/O in the final system. Bit 1 (A0) can be used as I/O when the system is 16-bit and there is no pull-up after reset for this pin.

16.4.6.5 Port G Pull-Up Enable Register

The pull-up enable register (PGPUEN) controls the pull-up resistors for each line in Port G. See Table 16-44 for the bit settings of the PGPUEN register.

PGPUEN Port G Pull-Up Enable Register **0x(FF)FFF432**

	BIT 7	6	5	4	3	2	1	BIT 0
	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-44. Port G Pull-Up Enable Register Description

Name	Description	Setting
PUx Bits 7–0	**Pull-Up**—This field enables the pull-up resistors on the port.	0 = Pull-up resistors are disabled. 1 = Pull-up resistors are enabled.

16.4.6.6 Port G Select Register

The select register (PGSEL) determines if a bit position in the data register (PGDATA) is assigned as a GPIO or to a dedicated I/O function. See Table 16-45 for information about setting the bits in the PGSEL register.

PGSEL Port G Select Register **0x(FF)FFF433**

	BIT 7	6	5	4	3	2	1	BIT 0
	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	1	0	0	0
	0x08							

Table 16-45. Port G Select Register Description

Name	Description	Setting
SELx Bits 7–0	**Select**—This field selects whether the internal chip function or I/O port signals are connected to the pins.	0 = The dedicated function pins are connected. 1 = The I/O port function pins are connected.

16.4.6.7 Port G Interrupt Mask Register

The Port G interrupt mask register (PGIMR) controls the interrupt whether be masked or not be masked for each line in Port G. The settings for the bit positions are shown in Table 16-46.

PGIMR Port G Interrupt Mask Register **0x(FF)FFF434**

	BIT 7	6	5	4	3	2	1	BIT 0
	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-46. Port G Interrupt Mask Register Description

Name	Description	Setting
IMx Bits 7–0	**Interrupt Mask**—This field masks the interrupt on the port.	0 = Interrupt is masked 1 = Interrupt is not masked

16.4.6.8 Port G Interrupt Status Register

The Port G interrupt status register (PGISR) display the interrupt status for each line in Port G. The settings for the bit positions are shown in Table 16-47.

PGISR Port G Interrupt Status Register **0x(FF)FFF435**

	BIT 7	6	5	4	3	2	1	BIT 0
	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
TYPE	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-47. Port G Interrupt Status Register Description

Name	Description	Setting
ISx Bits 7–0	**Interrupt Status**—This field displays the interrupt status on the port.	0 = Interrupt not generated 1 = Interrupt generated

16.4.6.9 Port G Interrupt Edge Register

The Port G interrupt edge register (PGIER) controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt to configure for each line in Port G. The settings for the bit positions are shown in Table 16-48.

PGIER Port G Interrupt Edge Register **0x(FF)FFF436**

	BIT 7	6	5	4	3	2	1	BIT 0
	EE7	EE6	EE5	EE4	EE3	EE2	EE1	EE0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-48. Port G Interrupt Edge Register Description

Name	Description	Setting
EEx Bits 7–0	**Edge Enable**—This field controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt.	0 = Edge-sensitive interrupt 1 = Level-sensitive interrupt

16.4.6.10 Port G Interrupt Polarity Register

The Port G interrupt polarity register (PGIPR) controls the interrupt sensitive polarity type to configure for each line in Port G. The settings for the bit positions are shown in Table 16-49.

PGIPR Port G Interrupt Polarity Register **0x(FF)FFF437**

	BIT 7	6	5	4	3	2	1	BIT 0
	POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-49. Port G Interrupt Polarity Register Description

Name	Description	Setting
POLx Bits 7–0	**Polarity**—This field controls interrupt polarity.	0 = Positive polarity 1 = Negative polarity

16.4.7 Port J Registers

Port J consists of the following 8-bit general-purpose I/O registers:

- Port J direction register (PJDIR)
- Port J data register (PJDATA)
- Port J pull-up enable register (PJPUEN)
- Port J select register (PJSEL)
- Port J interrupt mask register (PJIMR)
- Port J interrupt status register (PJISR)
- Port J interrupt edge register (PJIER)
- Port J interrupt polarity register (PJIPR)

Each signal in the PJDATA register connects to an external pin. As on the other ports, each bit on Port J is individually configured.

16.4.7.1 Port J Direction Register

The direction register controls the direction (input or output) of the line associated with the PJDATA bit position. When the data bit is assigned to a dedicated I/O function by the PJSEL register, the DIR bits are ignored. The settings for the bit positions are shown in Table 16-50.

PJDIR Port J Direction Register **0x(FF)FFF438**

	BIT 7	6	5	4	3	2	1	BIT 0
	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-50. Port J Direction Register Description

Name	Description	Setting
DIRx Bits 7–0	**Direction**—This field controls the direction of the pins in an 8-bit system. They reset to 0.	0 = Input 1 = Output

16.4.7.2 Port J Data Register

The bit settings for the PJDATA register are shown in Table 16-51.

PJDATA Port J Data Register **0x(FF)FFF439**

	BIT 7	6	5	4	3	2	1	BIT 0
	D7	D6	D5	D4	D3	D2	D1	D0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
	1	1	1	1	1	1	1	1
RESET	0xFF*							

*Actual bit value depends on external circuits connected to pin.

Table 16-51. Port J Data Register Description

Name	Description	Setting
Dx Bits 7–0	**Data**—This field reflects the status of the I/O signal in an 8-bit system.	0 = Drives the output signal low when DIRx is set to 1 or the external signal is low when DIRx is set to 0. 1 = Drives the output signal high when DIRx is set to 1 or the external signal is high when DIRx is set to 0.

Port J is multiplexed with the configurable CSPI (with internal FIFO) and UART 2 signals. These pins can be programmed as GPIO when the dedicated I/O signals are not in use.

This field controls or reports the data on the pins when the associated SELx bits are high. When the DIRx bits are high (output), the Dx bits control the pins. When the DIRx bits are low (input), the Dx bits report the signal driving the pins. The Dx bits can be written at any time. Bits that are configured as inputs accept the data, but the data written to each cannot be accessed until the corresponding pin is configured as an output. The actual value on the pin is reported when these bits are read, regardless of whether they are configured as input or output.

16.4.7.3 Port J Dedicated I/O Functions

The eight PJDATA lines are multiplexed with the dedicated I/O signals, these assignments are shown in Table 16-52. Bits 0–3 are control signals connected to CSPI. Their operation is detailed in Chapter 2, “Signal Descriptions,” Configurable Serial Peripheral Interface (CSPI). The remaining 4 bits are control signals for UART 2; more information appears in Chapter 2, “Signal Descriptions,” Universal Asynchronous Receiver/Transmitter 2 (UART2).

Table 16-52. Port J Dedicated I/O Function Assignments

Bit	GPIO Function	Dedicated I/O Function
0	Data bit 0	MOSI
1	Data bit 1	MISO
2	Data bit 2	SPICLK
3	Data bit 3	$\overline{SS}$
4	Data bit 4	RXD2
5	Data bit 5	TXD2

Table 16-52. Port J Dedicated I/O Function Assignments (Continued)

Bit	GPIO Function	Dedicated I/O Function
6	Data bit 6	$\overline{RTS2}$
7	Data bit 7	$\overline{CTS2}$

16.4.7.4 Port J Pull-Up Enable Register

The pull-up enable register (PJPUEN) controls the pull-up resistors for each line in Port J. The bit settings for the PJPUEN register are shown in Table 16-53.

PJPUEN Port J Pull-Up Enable Register **0x(FF)FFF43A**

	BIT 7	6	5	4	3	2	1	BIT 0
	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-53. Port J Pull-Up Enable Register Description

Name	Description	Setting
PUx Bits 7–0	**Pull-Up**—This field enables the pull-up resistors on the port	0 = Pull-up resistors are disabled. 1 = Pull-up resistors are enabled.

16.4.7.5 Port J Select Register

The Port J select register (PJSEL) determines if a bit position in the data register (PJDATA) is assigned as a GPIO or to a dedicated I/O function. The bit settings for the PJSEL register are shown in Table 16-54.

PJSEL Port J Select Register **0x(FF)FFF43B**

	BIT 7	6	5	4	3	2	1	BIT 0
	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	0	1	1	1	1
	0xFF							

Table 16-54. Port J Select Register Description

Name	Description	Setting
SELx Bits 7–0	**Select**—This field selects whether the internal chip function or I/O port signals are connected to the pins.	0 = The dedicated function pins are connected. 1 = The I/O port function pins are connected.

16.4.7.6 Port J Interrupt Mask Register

The Port J interrupt mask register (PJIMR) controls the interrupt whether be masked or not be masked for each line in Port J. The settings for the bit positions are shown in Table 16-55.

PJIMR Port J Interrupt Mask Register **0x(FF)FFF43C**

	BIT 7	6	5	4	3	2	1	BIT 0
	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-55. Port J Interrupt Mask Register Description

Name	Description	Setting
IMx Bits 7–0	**Interrupt Mask**—This field masks the interrupt on the port.	0 = Interrupt is masked. 1 = Interrupt is not masked.

16.4.7.7 Port J Interrupt Status Register

The Port J interrupt status register (PJISR) display the interrupt status for each line in Port J. The settings for the bit positions are shown in Table 16-56.

PJISR Port J Interrupt Status Register **0x(FF)FFF43D**

	BIT 7	6	5	4	3	2	1	BIT 0
	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
TYPE	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-56. Port J Interrupt Status Register Description

Name	Description	Setting
ISx Bits 7–0	**Interrupt Status**—This field displays the interrupt status on the port.	0 = Interrupt not generated 1 = Interrupt generated

16.4.7.8 Port J Interrupt Edge Register

The Port J interrupt edge register (PJIER) controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt to configure for each line in Port J. The settings for the bit positions are shown in Table 16-57.

PJIER Port J Interrupt Edge Register **0x(FF)FFF43E**

	BIT 7	6	5	4	3	2	1	BIT 0
	EE7	EE6	EE5	EE4	EE3	EE2	EE1	EE0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-57. Port J Interrupt Edge Register Description

Name	Description	Setting
EEx Bits 7–0	**Edge Enable**—This field controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt	0 = Edge-sensitive interrupt 1 = Level-sensitive interrupt

16.4.7.9 Port J Interrupt Polarity Register

The Port J interrupt polarity register (PJIPR) controls the interrupt sensitive polarity type to configure for each line in Port J. The settings for the bit positions are shown in Table 16-58.

PJIPR Port J Interrupt Polarity Register **0x(FF)FFF43F**

	BIT 7	6	5	4	3	2	1	BIT 0
	POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-58. Port J Interrupt Polarity Register Description

Name	Description	Setting
POLx Bits 7–0	**Polarity**—This field controls interrupt polarity.	0 = Positive polarity 1 = Negative polarity

16.4.8 Port K Registers

Port K consists of the following 8-bit general-purpose I/O registers:

- Port K direction register (PKDIR)
- Port K data register (PKDATA)
- Port K pull-up enable register (PKPUEN)
- Port K select register (PKSEL)
- Port K interrupt mask register (PKIMR)

- Port K interrupt status register (PKISR)
- Port K interrupt edge register (PKIER)
- Port K interrupt polarity register (PKIPR)

Each signal in the PKDATA register connects to an external pin. As on the other ports, each bit on Port K is individually configured.

16.4.8.1 Port K Direction Register

The direction register controls the direction (input or output) of the line associated with the PKDATA bit position. When the data bit is assigned to a dedicated I/O function by the PKSEL register, the DIR bits are ignored. The settings for the PKDIR register bit positions are shown in Table 16-59.

PKDIR Port K Direction Register **0x(FF)FFF440**

	BIT 7	6	5	4	3	2	1	BIT 0
	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-59. Port K Direction Register Description

Name	Description	Setting
DIRx Bits 7–0	**Direction**—This field controls the direction of the pins in an 8-bit system. They reset to 0.	0 = The pins are inputs. 1 = The pins are outputs.

16.4.8.2 Port K Data Register

The settings for the PKDATA register bit positions are shown in Table 16-60.

PKDATA Port K Data Register **0x(FF)FFF441**

	BIT 7	6	5	4	3	2	1	BIT 0
	D7	D6	D5	D4	D3	D2	D1	D0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	1	1	1	1
	0x0F*							

*Actual bit value depends on external circuits connected to pin.

Table 16-60. Port K Data Register Description

Name	Description	Setting
Dx Bits 7–0	**Data**—This field reflects the status of the I/O signal in an 8-bit system.	0 = Drives the output signal low when DIRx is set to 1 or the external signal is low when DIRx is set to 0. 1 = Drives the output signal high when DIRx is set to 1 or the external signal is high when DIRx is set to 0.

This field controls or reports the data on the pins when the associated SELx bits are high. When the DIRx bits are high (output), the Dx bits control the pins. When the DIRx bits are low (input), the Dx bits report the signal driving the pins. The Dx bits can be written at any time. Bits that are configured as inputs accept the data, but the data written to each cannot be accessed until the corresponding pin is configured as an output. The actual value on the pin is reported when these bits are read, regardless of whether they are configured as input or output.

16.4.8.3 Port K Dedicated I/O Functions

The eight PKDATA lines are multiplexed with the dedicated I/O signals, these assignments are shown in Table 16-61.

Table 16-61. Port K Dedicated I/O Function Assignments

Bit	GPIO Function	Dedicated I/O Function
0	Data bit 0	$\overline{\text{DATA\_READY}}$/PWMO2
1	Data bit 1	R/$\overline{\text{W}}$
2	Data bit 2	$\overline{\text{SDRAS}}$/$\overline{\text{CAS0}}$
3	Data bit 3	$\overline{\text{SDCAS}}$/$\overline{\text{CAS1}}$
4	Data bit 4	LD4
5	Data bit 5	LD5
6	Data bit 6	LD6
7	Data bit 7	LD7

When bit 0 is set as $\overline{\text{DATA\_READY}}$, it can be used in master mode to signal the CSPI master to clock out data. PWMO2 is an output signal from the PWM 2 module. If this pin is configured as this dedicated function and PKDIR0 is set to 1, the PWMO2 signal is selected. If PKDIR0 is 0, $\overline{\text{DATA\_READY}}$ is selected. This pin defaults to Port K data bit 0, GPIO input, pulled high.

16.4.8.4 Port K Pull-Up Enable Register

The pull-up enable register (PKPUEN) controls the pull-up resistors for each line in Port K. The settings for the PKPUEN register bit positions are shown in Table 16-62.

PKPUEN Port K Pull-Up Enable Register **0x(FF)FFF442**

	BIT 7	6	5	4	3	2	1	BIT 0
	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-62. Port K Pull-Up Enable Register Description

Name	Description	Setting
PUx Bits 7–0	**Pull-Up Enable**—This field enables the pull-up resistors on the port.	0 = Pull-up resistors are disabled. 1 = Pull-up resistors are enabled.

16.4.8.5 Port K Select Register

The Port K select register (PKSEL) determines if a bit position in the data register (PKDATA) is assigned as a GPIO or to a dedicated I/O function. The settings for the PKSEL register bit positions are shown in Table 16-63.

PKSEL Port K Select Register **0x(FF)FFF443**

	BIT 7	6	5	4	3	2	1	BIT 0
	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-63. Port K Select Register Description

Name	Description	Setting
SELx Bits 7–0	**Select**—This field selects whether the internal chip function or I/O port signals are connected to the pins.	0 = The dedicated function pins are connected. 1 = The I/O port function pins are connected.

16.4.8.6 Port K Interrupt Mask Register

The Port K interrupt mask register (PKIMR) controls the interrupt whether be masked or not be masked for each line in Port K. The settings for the bit positions are shown in Table 16-64.

PKIMR Port K Interrupt Mask Register **0x(FF)FFF444**

	BIT 7	6	5	4	3	2	1	BIT 0
	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-64. Port K Interrupt Mask Register Description

Name	Description	Setting
IMx Bits 7–0	**Interrupt Mask**—This field masks the interrupt on the port.	0 = Interrupt is masked. 1 = Interrupt is not masked.

16.4.8.7 Port K Interrupt Status Register

The Port K interrupt status register (PKISR) display the interrupt status for each line in Port K. The settings for the bit positions are shown in Table 16-65.

PKISR Port K Interrupt Status Register **0x(FF)FFF445**

	BIT 7	6	5	4	3	2	1	BIT 0
	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
TYPE	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-65. Port K Interrupt Status Register Description

Name	Description	Setting
ISx Bits 7–0	**Interrupt Status**—This field displays the interrupt status on the port.	0 = Interrupt not generated. 1 = Interrupt generated.

16.4.8.8 Port K Interrupt Edge Register

The Port K interrupt edge register (PKIER) controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt to configure for each line in Port K. The settings for the bit positions are shown in Table 16-66.

PKIER Port K Interrupt Edge Register **0x(FF)FFF446**

	BIT 7	6	5	4	3	2	1	BIT 0
	EE7	EE6	EE5	EE4	EE3	EE2	EE1	EE0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-66. Port K Interrupt Edge Register Description

Name	Description	Setting
EEx Bits 7–0	**Edge Enable**—This field controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt	0 = Edge-sensitive interrupt. 1 = Level-sensitive interrupt.

16.4.8.9 Port K Interrupt Polarity Register

The Port K interrupt polarity register (PKIPR) controls the interrupt sensitive polarity type to configure for each line in Port K. The settings for the bit positions are shown in Table 16-67.

PKIPR Port K Interrupt Polarity Register **0x(FF)FFF447**

	BIT 7	6	5	4	3	2	1	BIT 0
	POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-67. Port K Interrupt Polarity Register Description

Name	Description	Setting
POLx Bits 7–0	**Polarity**—This field controls interrupt polarity.	0 = Positive polarity 1 = Negative polarity

16.4.9 Port M Registers

Port M consists of the following eight general-purpose I/O registers:

- Port M direction register (PMDIR)
- Port M data register (PMDATA)
- Port M pull-up enable register (PMPUEN)
- Port M select register (PMSEL)

- Port M interrupt mask register (PMIMR)
- Port M interrupt status register (PMISR)
- Port M interrupt edge register (PMIER)
- Port M interrupt polarity register (PMIPR)

Each signal in the PMDATA register connects to an external pin except for the PM0 pin. It should be noted that pins 6 and 7 are not connected to external pins.

16.4.9.1 Port M Direction Register

The direction register controls the direction (input or output) of the line associated with the PMDATA bit position. When the data bit is assigned to a dedicated I/O function by the PMSEL register, the DIR bits are ignored. The settings for the PMDIR register bit positions are shown in Table 16-68.

PMDIR Port M Direction Register **0x(FF)FFF448**

	BIT 7	6	5	4	3	2	1	BIT 0
	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	
TYPE	rw	rw	rw	rw	rw	rw	rw	
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-68. Port M Direction Register Description

Name	Description	Setting
DIRx Bits 7–1	**Direction**—This field controls the direction of the pins in an 8-bit system. They reset to 0.	0 = Inputs 1 = Outputs
Reserved Bit 0	Reserved	This bit is reserved and should be set to 0.

16.4.9.2 Port M Data Register

The settings for the PMDATA register bit positions are shown in Table 16-69.

PMDATA Port M Data Register **0x(FF)FFF449**

	BIT 7	6	5	4	3	2	1	BIT 0
	D7	D6	D5	D4	D3	D2	D1	
TYPE	rw	rw	rw	rw	rw	rw	rw	
RESET	0	0	1	0	0	0	0	0
	0x20*							

*Actual bit value depends on external circuits connected to pin.

Table 16-69. Port M Data Register Description

Name	Description	Setting
Dx Bits 7–1	**Data**—This field reflects the status of the I/O signal in an 8-bit system.	0 = Drives the output signal low when DIRx is set to 1 or the external signal is low when DIRx is set to 0. 1 = Drives the output signal high when DIRx is set to 1 or the external signal is high when DIRx is set to 0.
Reserved Bit 0	Reserved	This bit is reserved and should be set to 0.

This field controls or reports the data on the pins when the associated SELx bits are high. When the DIRx bits are high (output), the Dx bits control the pins. When the DIRx bits are low (input), the Dx bits report the signal driving the pins. The Dx bits can be written at any time. Bits that are configured as inputs accept the data, but the data written to each cannot be accessed until the corresponding pin is configured as an output. The actual value on the pin is reported when these bits are read, regardless of whether they are configured as input or output.

16.4.9.3 Port M Dedicated I/O Functions

The eight PMDATA lines are multiplexed with the dedicated I/O signals, these assignments are shown in Table 16-70.

Table 16-70. Port M Dedicated I/O Function Assignments

Bit	GPIO Function	Dedicated I/O Function
0		
1	Data bit 1	SDCLKE0/$\overline{\text{DOE}}$
2	Data bit 2	DQM0
3	Data bit 3	DQM1
4	Data bit 4	$\overline{\text{SDWE}}$/$\overline{\text{DWE}}$
5	Data bit 5	SDCLKE1
6	Data bit 6	MA10

Table 16-70. Port M Dedicated I/O Function Assignments (Continued)

Bit	GPIO Function	Dedicated I/O Function
7	Data bit 7	MA11

All of the dedicated I/O functions are involved in the operation of the DRAM controller. See “DRAM Controller,” for more details.

16.4.9.4 Port M Pull-Up Enable Register

The pull-up enable register (PMPUEN) controls the pull-up resistors for each line in Port M. The settings for the PMPUEN register bit positions are shown in Table 16-71.

PMPUEN Port M Pull-Up Enable Register **0x(FF)FFF44A**

	BIT 7	6	5	4	3	2	1	BIT 0
	PU7	PU6	PU5	PU4	PU3	PU2	PU1	
TYPE	rw	rw	rw	rw	rw	rw	rw	
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-71. Port M Pull-Up Enable Register Description

Name	Description	Setting
PUx Bits 7–1	**Pull-Up**—This field enables the pull-up resistors on the port.	0 = Pull-up resistors are disabled. 1 = Pull-up resistors are enabled.
Reserved Bit 0	Reserved	This bit is reserved and should be set to 0.

16.4.9.5 Port M Select Register

The Port M select register (PMSEL) determines if a bit position in the data register (PMDATA) is assigned as a GPIO or to a dedicated I/O function. The settings for the PMSEL register bit positions are shown in Table 16-72.

PMSEL Port M Select Register **0x(FF)FFF44B**

	BIT 7	6	5	4	3	2	1	BIT 0
	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	
TYPE	rw	rw	rw	rw	rw	rw	rw	
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-72. Port M Select Register Description

Name	Description	Setting
SELx Bits 5–1	**Select**—This field selects whether the internal chip function or I/O port signals are connected to the pins.	0 = The dedicated function pins are connected. 1 = The I/O port function pins are connected.
Reserved Bit 0	Reserved	This bit is reserved and should be set to 0.

16.4.9.6 Port M Interrupt Mask Register

The Port M interrupt mask register (PMIMR) controls the interrupt whether be masked or not be masked for each line in Port M. The settings for the bit positions are shown in Table 16-73.

PMIMR Port M Interrupt Mask Register **0x(FF)FFF44C**

	BIT 7	6	5	4	3	2	1	BIT 0
	IM7	IM6	IM5	IM4	IM3	IM2	IM1	
TYPE	rw	rw	rw	rw	rw	rw	rw	
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-73. Port M Interrupt Mask Register Description

Name	Description	Setting
IMx Bits 7–1	**Interrupt Mask**—This field masks the interrupt on the port.	0 = Interrupt is masked. 1 = Interrupt is not masked.
Reserved Bit 0	Reserved	This bit is reserved and should be set to 0.

16.4.9.7 Port M Interrupt Status Register

The Port M interrupt status register (PMISR) display the interrupt status for each line in Port M. The settings for the bit positions are shown in Table 16-74.

PMISR Port M Interrupt Status Register **0x(FF)FFF44D**

	BIT 7	6	5	4	3	2	1	BIT 0
	IS7	IS6	IS5	IS4	IS3	IS2	IS1	
TYPE	r	r	r	r	r	r	r	
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-74. Port M Interrupt Status Register Description

Name	Description	Setting
ISx Bits 7–1	**Interrupt Status**—This field displays the interrupt status on the port.	0 = Interrupt not generated 1 = Interrupt generated
Reserved Bit 0	Reserved	This bit is reserved and should be set to 0.

16.4.9.8 Port M Interrupt Edge Register

The Port M interrupt edge register (PMIER) controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt to configure for each line in Port M. The settings for the bit positions are shown in Table 16-75.

PMIER Port M Interrupt Edge Register **0x(FF)FFF44E**

	BIT 7	6	5	4	3	2	1	BIT 0
	EE7	EE6	EE5	EE4	EE3	EE2	EE1	
TYPE	rw	rw	rw	rw	rw	rw	rw	
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-75. Port M Interrupt Edge Register Description

Name	Description	Setting
EEx Bits 7–1	**Edge Enable**—This field controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt.	0 = Edge-sensitive interrupt. 1 = Level-sensitive interrupt.
Reserved Bit 0	Reserved	This bit is reserved and should be set to 0.

16.4.9.9 Port M Interrupt Polarity Register

The Port M interrupt polarity register (PMIPR) controls the interrupt sensitive polarity type to configure for each line in Port M. The settings for the bit positions are shown in Table 16-76.

PMIPR Port M Interrupt Polarity Register **0x(FF)FFF44F**

	BIT 7	6	5	4	3	2	1	BIT 0
	POL7	POL6	POL5	POL4	POL3	POL2	POL1	
TYPE	rw	rw	rw	rw	rw	rw	rw	
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-76. Port M Interrupt Polarity Register Description

Name	Description	Setting
POLx Bits 7–1	**Polarity**—This field controls interrupt polarity.	0 = Positive polarity. 1 = Negative polarity.
Reserved Bit 0	Reserved	This bit is reserved and should be set to 0.

16.4.10 Port N Registers

Port N consists of the following 8-bit general-purpose I/O registers:

- Port N direction register (PNDIR)
- Port N data register (PNDATA)
- Port N pull-up enable register (PNPUEN)
- Port N select register (PNSEL)
- Port N interrupt mask register (PNIMR)
- Port N interrupt status register (PNISR)
- Port N interrupt edge register (PNIER)
- Port N interrupt polarity register (PNIPR)

16.4.10.1 Port N Direction Register

The direction register controls the direction (input or output) of the line associated with the PNDATA bit position. When the data bit is assigned to a dedicated I/O function by the PNSEL register, the DIR bits are ignored. The settings for the PNDIR register bit positions are shown in Table 16-77.

PNDIR Port N Direction Register **0x(FF)FFF450**

	BIT 7	6	5	4	3	2	1	BIT 0
	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-77. Port N Direction Register Description

Name	Description	Setting
DIRx Bits 7–0	**Direction**—This field controls the direction of the pins in an 8-bit system. They reset to 0.	0 = Input 1 = Output

16.4.10.2 Port N Data Register

The settings for the PNDATA register bit positions are shown in Table 16-78.

PNDATA Port N Data Register **0x(FF)FFF451**

	BIT 7	6	5	4	3	2	1	BIT 0
	D7	D6	D5	D4	D3	D2	D1	D0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	1	0	0	0	0	0
	0x20*							

*Actual bit value depends on external circuits connected to pin.

Table 16-78. Port N Data Register Description

Name	Description	Setting
Dx Bits 7–0	**Data**—This field reflects the status of the I/O signal in an 8-bit system.	0 = Drives the output signal low when DIRx is set to 1 or the external signal is low when DIRx is set to 0. 1 = Drives the output signal high when DIRx is set to 1 or the external signal is high when DIRx is set to 0.

This field controls or reports the data on the pins when the associated SELx bits are high. When the DIRx bits are high (output), the Dx bits control the pins. When the DIRx bits are low (input), the Dx bits report the signal driving the pins. The Dx bits can be written at any time. Bits that are configured as inputs accept the data, but the data written to each cannot be accessed until the corresponding pin is configured as an output. The actual value on the pin is reported when these bits are read, regardless of whether they are configured as input or output.

16.4.10.3 Port N Dedicated I/O Functions

The eight PNDATA lines are multiplexed with the dedicated I/O signals, these assignments are shown in Table 16-79.

Table 16-79. Port N Dedicated I/O Function Assignments

Bit	GPIO Function	Dedicated I/O Function
0	Data bit 0	USBD_AFE
1	Data bit 1	$\overline{\text{USBD\_ROE}}$
2	Data bit 2	USBD_VMO
3	Data bit 3	USBD_VPO
4	Data bit 4	USBD_SUSPND
5	Data bit 5	USBD_RCV
6	Data bit 6	USBD_VP
7	Data bit 7	USBD_VM

16.4.10.4 Port N Pull-Up Enable Register

The pull-up enable register (PNPUEN) controls the pull-up resistors for each line in Port N. The settings for the PNPUEN register bit positions are shown in Table 16-80.

PNPUEN Port N Pull-Up Enable Register **0x(FF)FFF452**

	BIT 7	6	5	4	3	2	1	BIT 0
	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-80. Port N Pull-Up Enable Register Description

Name	Description	Setting
PUx Bits 7–0	**Pull-Up Enable**—This field enables the pull-up resistors on the port.	0 = Pull-up resistors are disabled. 1 = Pull-up resistors are enabled.

16.4.10.5 Port N Select Register

The Port N select register (PNSEL) determines if a bit position in the data register (PNDATA) is assigned as a GPIO or to a dedicated I/O function. The settings for the PNSEL register bit positions are shown in Table 16-81.

PNSEL Port N Select Register **0x(FF)FFF453**

	BIT 7	6	5	4	3	2	1	BIT 0
	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-81. Port N Select Register Description

Name	Description	Setting
SELx Bits 7–0	**Select**—This field selects whether the internal chip function or I/O port signals are connected to the pins.	0 = The dedicated function pins are connected. 1 = The I/O port function pins are connected.

16.4.10.6 Port N Interrupt Mask Register

The Port N interrupt mask register (PNIMR) controls the interrupt whether be masked or not be masked for each line in Port N. The settings for the bit positions are shown in Table 16-82.

PNIMR Port N Interrupt Mask Register **0x(FF)FFF454**

	BIT 7	6	5	4	3	2	1	BIT 0
	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-82. Port N Interrupt Mask Register Description

Name	Description	Setting
IMx Bits 7–0	**Interrupt Mask**—This field masks the interrupt on the port.	0 = Interrupt is masked 1 = Interrupt is not masked

16.4.10.7 Port N Interrupt Status Register

The Port N interrupt status register (PNISR) display the interrupt status for each line in Port N. The settings for the bit positions are shown in Table 16-83.

PNISR Port N Interrupt Status Register **0x(FF)FFF455**

	BIT 7	6	5	4	3	2	1	BIT 0
	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
TYPE	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-83. Port N Interrupt Status Register Description

Name	Description	Setting
ISx Bits 7–0	**Interrupt Status**—This field displays the interrupt status on the port.	0 = Interrupt not generated 1 = Interrupt generated

16.4.10.8 Port N Interrupt Edge Register

The Port N interrupt edge register (PNIER) controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt to configure for each line in Port N. The settings for the bit positions are shown in Table 16-84.

PNIER Port N Interrupt Edge Register **0x(FF)FFF456**

	BIT 7	6	5	4	3	2	1	BIT 0
	EE7	EE6	EE5	EE4	EE3	EE2	EE1	EE0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-84. Port N Interrupt Edge Register Description

Name	Description	Setting
EEx Bits 7–0	**Edge Enable**—This field controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt.	0 = Edge-sensitive interrupt. 1 = Level-sensitive interrupt.

16.4.10.9 Port N Interrupt Polarity Register

The Port N interrupt polarity register (PNIPR) controls the interrupt sensitive polarity type to configure for each line in Port N. The settings for the bit positions are shown in Table 16-85.

PNIPR Port N Interrupt Polarity Register **0x(FF)FFF457**

	BIT 7	6	5	4	3	2	1	BIT 0
	POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-85. Port N Interrupt Polarity Register Description

Name	Description	Setting
POLx Bits 7–0	**Polarity**—This field controls interrupt polarity.	0 = Positive polarity. 1 = Negative polarity.

16.4.11 Port P Registers

Port P consists of the following eight general-purpose I/O registers:

- Port P direction register (PPDIR)
- Port P data register (PPDATA)
- Port P pull-up enable register (PPPUEN)
- Port P select register (PPSEL)
- Port P interrupt mask register (PPIMR)
- Port P interrupt status register (PPISR)
- Port P interrupt edge register (PPIER)
- Port P interrupt polarity register (PPIPR)

16.4.11.1 Port P Direction Register

The direction register controls the direction (input or output) of the line associated with the PPDATA bit position. When the data bit is assigned to a dedicated I/O function by the PPSEL register, the DIR bits are ignored. The settings for the PPDIR register bit positions are shown in Table 16-86.

PPDIR Port P Direction Register **0x(FF)FFF458**

	BIT 7	6	5	4	3	2	1	BIT 0
	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-86. Port P Direction Register Description

Name	Description	Setting
DIRx Bits 7–0	**Direction**—This field controls the direction of the pins in an 8-bit system. They reset to 0.	0 = Inputs 1 = Outputs

16.4.11.2 Port P Data Register

The settings for the PPDATA register bit positions are shown in Table 16-87.

PPDATA Port P Data Register **0x(FF)FFF459**

	BIT 7	6	5	4	3	2	1	BIT 0
	D7	D6	D5	D4	D3	D2	D1	D0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	1	0	0	0	0	0
	0x20*							

*Actual bit value depends on external circuits connected to pin.

Table 16-87. Port P Data Register Description

Name	Description	Setting
Dx Bits 7–0	**Data**—This field reflects the status of the I/O signal in an 8-bit system.	0 = Drives the output signal low when DIRx is set to 1 or the external signal is low when DIRx is set to 0. 1 = Drives the output signal high when DIRx is set to 1 or the external signal is high when DIRx is set to 0.

This field controls or reports the data on the pins when the associated SELx bits are high. When the DIRx bits are high (output), the Dx bits control the pins. When the DIRx bits are low (input), the Dx bits report the signal driving the pins. The Dx bits can be written at any time. Bits that are configured as inputs accept the data, but the data written to each cannot be accessed until the corresponding pin is configured as an output. The actual value on the pin is reported when these bits are read, regardless of whether they are configured as input or output.

16.4.11.3 Port P Dedicated I/O Functions

The eight PPDATA lines are multiplexed with the dedicated I/O signals, these assignments are shown in Table 16-88.

Table 16-88. Port P Dedicated I/O Function Assignments

Bit	GPIO Function	Dedicated I/O Function
0	Data bit 0	LD8
1	Data bit 1	LD9
2	Data bit 2	LD10
3	Data bit 3	LD11
4	Data bit 4	LD12
5	Data bit 5	LD13
6	Data bit 6	LD14
7	Data bit 7	LD15

16.4.11.4 Port P Pull-Up Enable Register

The pull-up enable register (PPPUEN) controls the pull-up resistors for each line in Port P. The settings for the PPPUEN register bit positions are shown in Table 16-89.

PPPUEN Port P Pull-Up Enable Register **0x(FF)FFF45A**

	BIT 7	6	5	4	3	2	1	BIT 0
	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-89. Port P Pull-Up Enable Register Description

Name	Description	Setting
PUx Bits 7–0	**Pull-Up Enable**—This field enables the pull-up resistors on the port.	0 = Pull-up resistors are disabled. 1 = Pull-up resistors are enabled.

16.4.11.5 Port P Select Register

The Port P select register (PPSEL) determines if a bit position in the data register (PPDATA) is assigned as a GPIO or to a dedicated I/O function. The settings for the PPSEL register bit positions are shown in Table 16-90.

PPSEL Port P Select Register **0x(FF)FFF45B**

	BIT 7	6	5	4	3	2	1	BIT 0
	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-90. Port P Select Register Description

Name	Description	Setting
SELx Bits 7–0	**Select**—This field selects whether the internal chip function or I/O port signals are connected to the pins.	0 = The dedicated function pins are connected. 1 = The I/O port function pins are connected.

16.4.11.6 Port P Interrupt Mask Register

The Port P interrupt mask register (PPIMR) controls the interrupt whether be masked or not be masked for each line in Port P. The settings for the bit positions are shown in Table 16-91.

PPIMR Port P Interrupt Mask Register **0x(FF)FFF45C**

	BIT 7	6	5	4	3	2	1	BIT 0
	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-91. Port P Interrupt Mask Register Description

Name	Description	Setting
IMx Bits 7–0	**Interrupt Mask**—This field masks the interrupt on the port.	0 = Interrupt is masked 1 = Interrupt is not masked

16.4.11.7 Port P Interrupt Status Register

The Port P interrupt status register (PPISR) display the interrupt status for each line in Port P. The settings for the bit positions are shown in Table 16-92.

PPISR Port P Interrupt Status Register **0x(FF)FFF45D**

	BIT 7	6	5	4	3	2	1	BIT 0
	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
TYPE	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-92. Port P Interrupt Status Register Description

Name	Description	Setting
ISx Bits 7–0	**Interrupt Status**—This field displays the interrupt status on the port.	0 = Interrupt not generated 1 = Interrupt generated

16.4.11.8 Port P Interrupt Edge Register

The Port P interrupt edge register (PPIER) controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt to configure for each line in Port P. The settings for the bit positions are shown in Table 16-93.

PPIER **Port P Interrupt Edge Register** **0x(FF)FFF45E**

	BIT 7	6	5	4	3	2	1	BIT 0
	EE7	EE6	EE5	EE4	EE3	EE2	EE1	EE0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-93. Port P Interrupt Edge Register Description

Name	Description	Setting
EEx Bits 7–0	**Edge Enable**—This field controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt.	0 = Edge-sensitive interrupt 1 = Level-sensitive interrupt

16.4.11.9 Port P Interrupt Polarity Register

The Port P interrupt polarity register (PPIPR) controls the interrupt sensitive polarity type to configure for each line in Port P. The settings for the bit positions are shown in Table 16-94.

PPIPR Port P Interrupt Polarity Register **0x(FF)FFF45F**

	BIT 7	6	5	4	3	2	1	BIT 0
	POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-94. Port P Interrupt Polarity Register Description

Name	Description	Setting
POLx Bits 7–0	**Polarity**—This field controls interrupt polarity.	0 = Positive polarity 1 = Negative polarity

16.4.12 Port R Registers

Port R consists of the following eight general-purpose I/O registers:

- Port R direction register (PRDIR)
- Port R data register (PRDATA)
- Port R pull-up/pull-down enable register (PRPUEN)
- Port R select register (PRSEL)
- Port R interrupt mask register (PRIMR)
- Port R interrupt status register (PRISR)
- Port R interrupt edge register (PRIER)
- Port R interrupt polarity register (PRIPR)

16.4.12.1 Port R Direction Register

The direction register controls the direction (input or output) of the line associated with the PRDATA bit position. When the data bit is assigned to a dedicated I/O function by the PRSEL register, the DIR bits are ignored. The settings for the PRDIR register bit positions are shown in Table 16-95.

PRDIR Port R Direction Register **0x(FF)FFF460**

	BIT 7	6	5	4	3	2	1	BIT 0
			DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
TYPE			rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-95. Port R Direction Register Description

Name	Description	Setting
Reserved Bits 7–6	Reserved	These bits are reserved and should be set to 0.
DIRx Bits 5–0	**Direction**—This field controls the direction of the pins in an 8-bit system. They reset to 0.	0 = Inputs 1 = Outputs

16.4.12.2 Port R Data Register

The settings for the PRDATA register bit positions are shown in Table 16-96.

PRDATA Port R Data Register **0x(FF)FFF461**

	BIT 7	6	5	4	3	2	1	BIT 0
			D5	D4	D3	D2	D1	D0
TYPE			rw	rw	rw	rw	rw	rw
RESET	0	0	1	0	0	0	0	0
	0x20*							

*Actual bit value depends on external circuits connected to pin.

Table 16-96. Port R Data Register Description

Name	Description	Setting
Reserved Bits 7–6	Reserved	These bits are reserved and should be set to 0.
Dx Bits 5–0	**Data**—This field reflects the status of the I/O signal in an 8-bit system.	0 = Drives the output signal low when DIRx is set to 1 or the external signal is low when DIRx is set to 0. 1 = Drives the output signal high when DIRx is set to 1 or the external signal is high when DIRx is set to 0.

This field controls or reports the data on the pins when the associated SELx bits are high. When the DIRx bits are high (output), the Dx bits control the pins. When the DIRx bits are low (input), the Dx bits report the signal driving the pins. The Dx bits can be written at any time. Bits that are configured as inputs accept

the data, but the data written to each cannot be accessed until the corresponding pin is configured as an output. The actual value on the pin is reported when these bits are read, regardless of whether they are configured as input or output.

16.4.12.3 Port R Dedicated I/O Functions

The eight PRDATA lines are multiplexed with the dedicated I/O signals, these assignments are shown in Table 16-97.

Table 16-97. Port R Dedicated I/O Function Assignments

Bit	GPIO Function	Dedicated I/O Function
0	Data bit 0	MMCSD_CLK/MS_SDIO
1	Data bit 1	MMCSD_CMD/MS_PI0
2	Data bit 2	MMCSD_DAT[0]/MS_BS
3	Data bit 3	MMCSD_DAT[1]/MS_PI1
4	Data bit 4	MMCSD_DAT[2]/MS_SCLKI
5	Data bit 5	MMCSD_DAT[3]/MS_SCLKO
6		
7		

16.4.12.4 Port R Pull-Up/Pull-Down Enable Register

The pull-up/pull-down enable register (PRPUEN) controls the pull-up and pull-down resistors for each line in Port R. The settings for the PRPUEN register bit positions are shown in Table 16-98.

PRPUEN Port R Pull-Up/Pull-Down Enable Register **0x(FF)FFF462**

	BIT 7	6	5	4	3	2	1	BIT 0
			PU5	PU4	PU3	PU2	PU1	PD0
TYPE			rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-98. Port R Pull-Up/Pull-Down Enable Register Description

Name	Description	Setting
Reserved Bits 7–6	Reserved	These bits are reserved and should be set to 0
PUx Bits 5–0	**Pull-Up/Pull-Down Enable**—This field enables the pull-up and pull-down resistors on the port.	0 = Pull-up and pull-down resistors are disabled 1 = Pull-up and pull-down resistors are enabled

16.4.12.5 Port R Select Register

The Port R select register (PRSEL) determines if a bit position in the data register (PRDATA) is assigned as a GPIO or to a dedicated I/O function. The settings for the PRSEL register bit positions are shown in Table 16-99.

PRSEL Port R Select Register **0x(FF)FFF463**

	BIT 7	6	5	4	3	2	1	BIT 0
			SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
TYPE			rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
	0xFF							

Table 16-99. Port R Select Register Description

Name	Description	Setting
Reserved Bits 7–6	Reserved	These bits are reserved and should be set to 0.
SELx Bits 5–0	**Select**—This field selects whether the internal chip function or I/O port signals are connected to the pins.	0 = The dedicated function pins are connected. 1 = The I/O port function pins are connected.

16.4.12.6 Port R Interrupt Mask Register

The Port R interrupt mask register (PRIMR) controls the interrupt enable for each line in Port R. The settings for the bit positions are shown in Table 16-100.

PRIMR Port R Interrupt Mask Register **0x(FF)FFF464**

	BIT 7	6	5	4	3	2	1	BIT 0
			IM5	IM4	IM3	IM2	IM1	IM0
TYPE			rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-100. Port R Interrupt Mask Register Description

Name	Description	Setting
Reserved Bits 7–6	Reserved	These bits are reserved and should be set to 0.
IMx Bits 5–0	**Interrupt Mask**—This field masks the interrupt on the port.	0 = Interrupt is masked. 1 = Interrupt is not masked.

16.4.12.7 Port R Interrupt Status Register

The Port R interrupt status register (PRISR) display the interrupt status for each line in Port R. The settings for the bit positions are shown in Table 16-101.

PRISR Port R Interrupt Status Register **0x(FF)FFF465**

	BIT 7	6	5	4	3	2	1	BIT 0
			IS5	IS4	IS3	IS2	IS1	IS0
TYPE			r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-101. Port R Interrupt Status Register Description

Name	Description	Setting
Reserved Bits 7–6	Reserved	These bits are reserved and should be set to 0.
ISx Bits 5–0	**Interrupt Status**—This field displays the interrupt status on the port.	0 = Interrupt not generated. 1 = Interrupt generated.

16.4.12.8 Port R Interrupt Edge Register

The Port R interrupt edge register (PRIER) controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt to configure for each line in Port R. The settings for the bit positions are shown in Table 16-102.

PRIER Port R Interrupt Edge Register **0x(FF)FFF466**

	BIT 7	6	5	4	3	2	1	BIT 0
			EE5	EE4	EE3	EE2	EE1	EE0
TYPE			rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-102. Port R Interrupt Edge Register Description

Name	Description	Setting
Reserved Bits 7–6	Reserved	These bits are reserved and should be set to 0.
EEx Bits 5–0	**Edge Enable**—This field controls whether the interrupt signal is an edge-triggered or a level-sensitive interrupt	0 = Edge-sensitive interrupt. 1 = Level-sensitive interrupt.

16.4.12.9 Port R Interrupt Polarity Register

The Port R interrupt polarity register (PRIPR) controls the interrupt sensitive polarity type to configure for each line in Port R. The settings for the bit positions are shown in Table .

PRIPR Port R Interrupt Polarity Register **0x(FF)FFF467**

	BIT 7	6	5	4	3	2	1	BIT 0
			POL5	POL4	POL3	POL2	POL1	POL0
TYPE			rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 16-103. Port R Interrupt Polarity Register Description

Name	Description	Setting
Reserved Bits 7–6	Reserved	These bits are reserved and should be set to 0.
POLx Bits 5–0	**Polarity**—This field controls interrupt polarity.	0 = Positive polarity 1 = Negative polarity

Chapter 17
Multimedia Card/Secure Digital Host Controller

The multimedia card (MMC) is a flash-based, mass storage product that uses a seven-pin medium-speed serial interface. The MMC bus uses a multi-drop master/slave configuration requiring a host device to control the multimedia card.

The secure digital memory card (SD) is an evolution of MMC. With two additional pins at the form factor, it is specifically designed to meet the security, capacity performance, and environment requirements inherent in newly emerging audio and video consumer electronic devices. With some additions made, the physical form factor, pin assignment, and data transfer protocol are forward compatible with the multimedia card. The SD memory card includes a copyright-protection mechanism that complies with the security of the SDMI (Secure Digital Music Initiative) standard, is faster, and can have higher memory capacity.

The MMC/SD host controller is designed to support both MMCs and SD memory cards. The SD memory card security system uses mutual authentication and a cipher algorithm to protect from illegal use of the card content. The main difference between MMCs and SD memory cards is their respective initialization processes.

17.1 Features

The MMC/SD module acts as the host device by providing command interrupter, protocol handler, error detection, and clock control features as well as an application interface.

Features of the MMC/SD host controller module include the following:

- Compatible with the MMC system specification version 2.2
- Compatible with the SD Memory Card specification version 1.0
- Ability to achieve maximum data rate with up to 10 cards; only 1 SD card is allowed
- Error auto-detection for Response CRC and time-out and Access CRC and time-out errors
- Built-in programmable frequency counter for MMC/SD bus
- Maskable hardware interrupt for internal status and FIFO status indicator
- 8 x 16-bit built-in FIFO with fast DMA access
- Automatic operation that can be paused upon internal FIFO full condition, allowing flexible packet transaction for dynamic DMA transfer
- Built-in 3-bit prescaler and 3-bit bus clock divider to maximize the performance of data transaction between memory card and host
- Built-in 7/16 CRC generate and check for command and data

17.2 Block Diagram

Figure 17-1 is a block diagram of the MMC/SD host controller, and Figure 17-2 shows the interconnection between the host controller and an MMC or SD memory card.

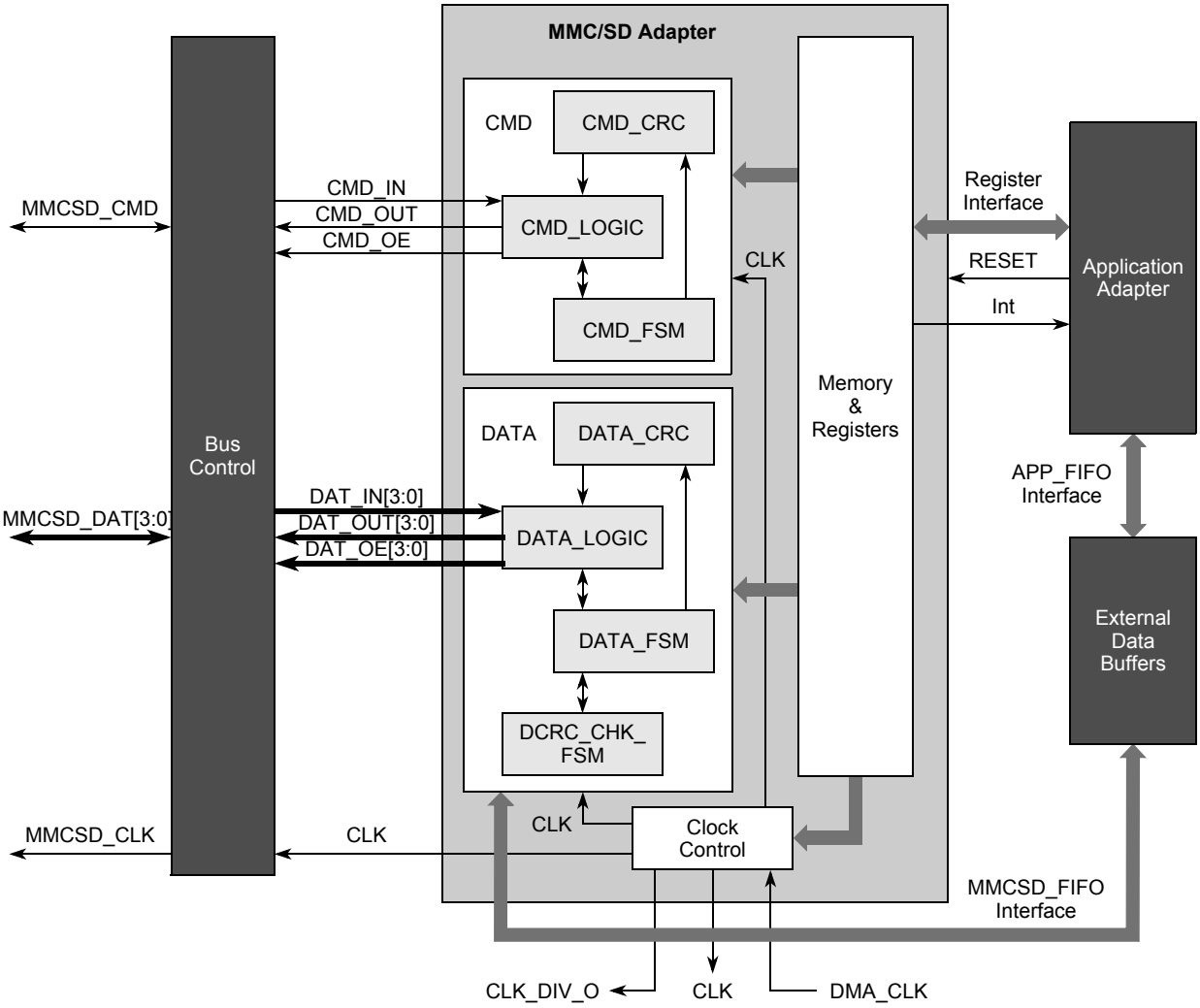


Figure 17-1. MMC/SD Host Controller Block Diagram

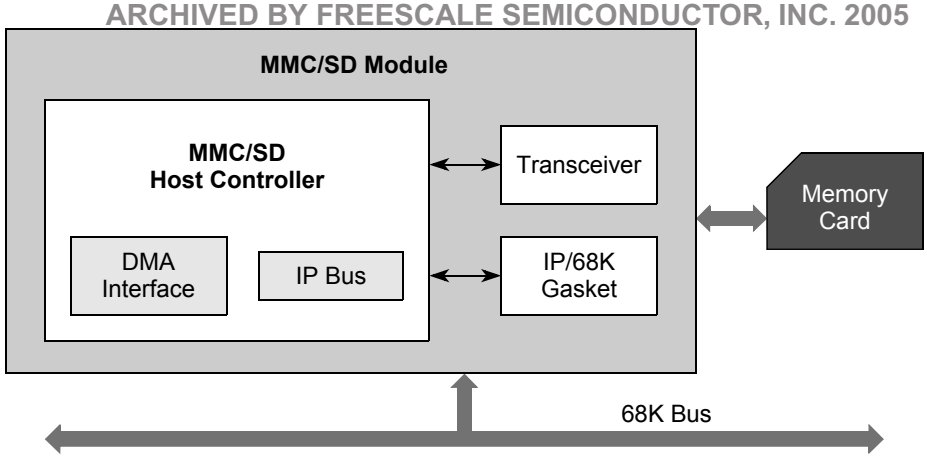


Figure 17-2. Interconnection with MMC/SD Host Controller

17.3 MMC/SD Host Controller Interface

This section provides information on signals and I/O multiplexing.

17.3.1 Signal Description

The MC68SZ328's MMC/SD host controller supports the standard MMC and SD interfaces. The MMC/SD host controller has six signals to interface with the external MMC and SD devices.

- MMCSD_CLK—Host to card clock signal (Output)
- MMCSD_CMD—Bidirectional Command/Response signal between host and the card (Input/Output)
- MMCSD_DAT[3:0]—Four Bidirectional Data signals (Input/Output)

17.3.2 MMC/SD Host Controller Interface I/O Multiplexing

The MMC/SD host controller I/O pins are multiplexed with the Memory Stick I/O pins in Port R of the MC68SZ328. Table 17-1 shows the muxing assignment. To select an MMC/SD dedicated I/O function, program the Port R Select register to 0 (dedicated I/O function).

Table 17-1. MMC/SD and MS Port R Muxing Assignment

Port R Pin	GPIO Function	Dedicated I/O Function
0	Data bit 0	MMCSD_CLK
1	Data bit 1	MMCSD_CMD
2	Data bit 2	MMCSD_DAT[0]
3	Data bit 3	MMCSD_DAT[1]
4	Data bit 4	MMCSD_DAT[2]
5	Data bit 5	MMCSD_DAT[3]

17.4 Functional Blocks

The following sections provide brief functional descriptions of the system blocks shown in Figure 17-1, including the FIFO and DMA interface, card error detection, clock controller, and interrupt operation.

17.4.1 FIFO and DMA Interface

To maximize the throughput between the system memory and the external memory card, an access-strobe driven, buffer-structured FIFO using DMA is provided. This approach also resolves possible synchronization problems between the bus clock (66.32 MHz) and the MMCSD_CLK (0 to 16.58 MHz). The FIFO is composed of an 8 x 16-bit register file acting as RAM. The FIFO unit is implemented by adding two counters to the RAM memories:

- Access address counter
- FIFO full and FIFO empty flags counter

The buffer structure is implemented by multiplexing the two different access strobes from external DMA and the MMC/SD host controller. The strobe multiplexer is controlled by using an FSM. This structure achieves a faster data path to and from the application, and it also allows the MMC/SD bus clock (MMCSD_CLK) to stop smoothly when waiting for DMA access. Figure 17-3 shows the block architecture of the FIFO interface.

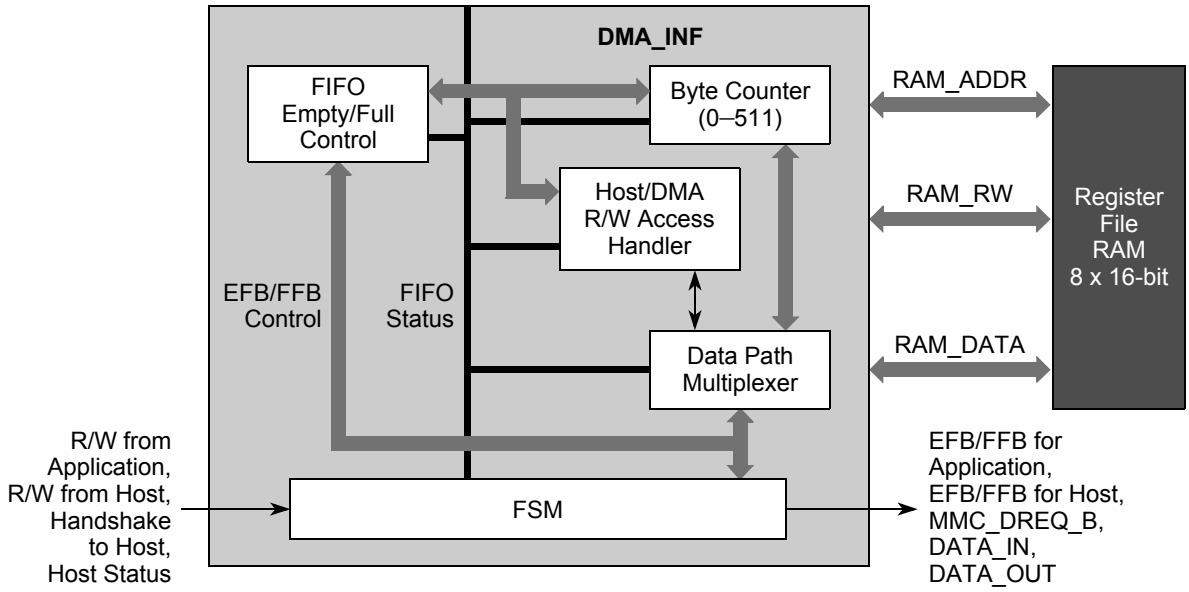


Figure 17-3. Data Transfer FIFO Buffers Interface Block

17.4.2 Card Error Detection

The MMC/SD controller detects the following errors on the MMC/SD bus:

- Response CRC error - There was a CRC calculation error of the command response.
- Response time out - The response did not appear after the number of clock cycles specified.
- Write CRC error - The CRC response in the data write sequence returned a CRC error.
- Read CRC error - The CRC calculation in the data read block is wrong.
- Time out read data - The data in the read command timed out.

The CRC is intended for protecting MMC/SD memory card commands, responses, and data transfers against transmission errors on the MMC/SD memory card bus. One CRC is generated for every command and checked for every response on the CMD line. The generation and detection of CRC is provided by hardware in the MMC/SD module. To minimize the gate count, the internal command shift register is re-used for the CRC shift register.

17.4.3 System Clock Control

The system clock control unit controls the rate of the host main clock and checks whether it is on or off. The clock is turned on by setting bit 1 (START_CLK) and is turned off by setting bit 0 (STOP_CLK) of the STR_STP_CLK register (write 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.

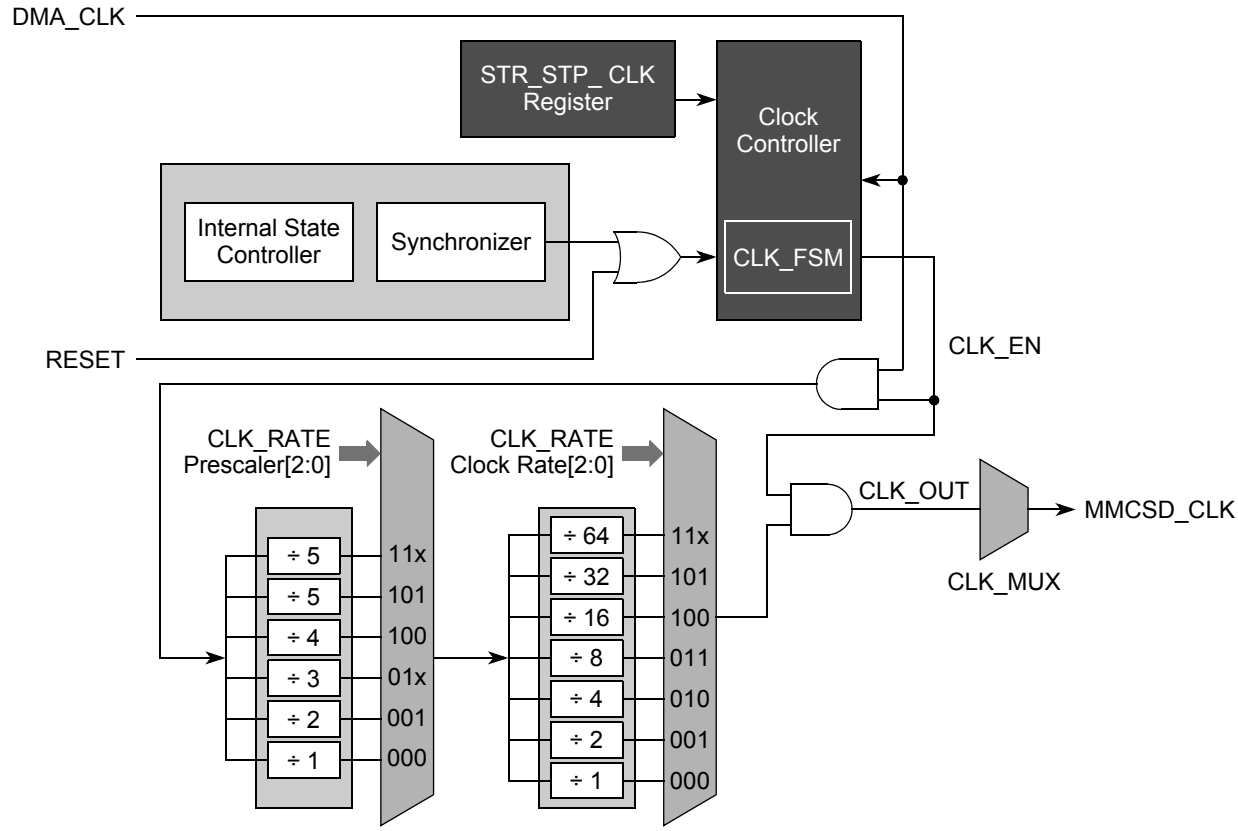


Figure 17-4. System Clock Control Unit

The purpose of the clock control unit is to enable the application to divide the clock and to stop and start the clock without generating a glitch in the process. The system clock control unit ensures that every time the clock stops, it stops when it is low. This is done by ensuring that all the clocks from the CLK_DIV counter and the master clock are low. The stop clock data unit enables the data FSM to stop the clock if the application is too slow, thus preventing the update of the data FIFO on time during a multiple block or stream write and read.

17.4.4 MMC/SD Host Controller Interrupt Operation

The MMC/SD host controller interrupt to the MC68SZ328 interrupt controller is shared with the Memory Stick interrupt. The default value is MMCSDIRQ/MSIRQ; the default interrupt level is 5. As with the Memory Stick interrupt, the MMC/SD interrupt is level programmable using Interrupt Level Register 2 (ILCR2) bits 10–8. MMCSDIRQ is enabled by setting Interrupt Mask Register (IMR) bit 29 (MMCSD/MS) to 0, and it is masked by setting IMR bit 29 to 1. MMCSDIRQ status is provided by Interrupt Status Register (ISR) bit 29 (MMCSD/MS), and pending status is provided in Interrupt Pending Register (IPR) bit 29 (MMCSD/MS).

17.5 Functional Example of MMC/SD Communication

All communication between the host 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. The commands GO_IDLE_STATE, SEND_OP_COND, ALL_SEND_CID, and SET_RELATIVE_ADDR are broadcast to all cards. During broadcast mode, all cards are in open-drain mode to avoid bus contention.

The addressed (point-to-point) commands are sent to the addressed card and cause a response from this card. The commands SELECT/DESELECT_CARD, SEND_STATUS, and STOP_TRANSMISSION are addressed to the specific card.

After the broadcast command SET_RELATIVE_ADDR is issued, cards enter standby mode. At this point addressed commands can be used to control a single card. CMD/DAT returns to push-pull mode in order to have maximum driving for maximum operation frequency.

Two operation modes are available for the MMC/SD host controller: card identification mode and data transfer mode.

17.5.1 Card Identification Mode

When 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.

17.5.1.1 Reset

The host has three types of reset:

- Hardware reset, which is driven by POR.
- Software reset is enacted by the write operation on the register STR_STP_CLK. Follow the recommended sequence mentioned in Section 17.4.3, “System Clock Control.”
- Card reset: the command Go_Idle_State (CMD0) is the software reset command for the card bus, and it puts each card in the idle state regardless of the current card state. Cards in Inactive State are not affected by this command.

After power on or CMD0, all cards’ CMD lines are in input mode, waiting for the start bit of the next command. 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).

17.5.1.2 Voltage Validation

All cards can establish communication with the host using any operating voltage within the maximum allowed voltage range specified in the MMC standard. However, the supported minimum and maximum values for V_{DD} are defined in the operation conditions register (OCR) and might not cover the whole range. Cards that store the CID (Card Identification Number) and CSD (Card Specific Data) data in the payload memory would be able to communicate this information only under data transfer V_{DD} conditions. That is, if the host and card have incompatible V_{DD} ranges, the card will not be able to complete the identification cycle or to send CSD data.

Therefore, special commands, SEND_OP_COND (CMD1 for MMC) and SD_SEND_OP_COND (ACMD41 for SD), are designed to provide a mechanism to identify and reject cards that do not match the V_{DD} range desired by the host. The host sends the required V_{DD} voltage window as the operand of this command. Cards that cannot perform data transfers in the specified range must discard themselves from further bus operations and enter the Inactive State. ACMD41 is an application-specific command; therefore, APP_CMD (CMD55) always precedes ACMD41. The RCA to be used for CMD55 in idle state is the card's default RCA, which is 0x0000.

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 unusable cards in the stack is desired.

During the initialization procedure, the host is not allowed to change the operating voltage range. If there is a real change in the operating conditions, the host must reset the card stack (sending CMD0 to all cards) and restart the initialization procedure.

17.5.1.3 Card Registry

The identification process of the SD memory card starts at clock rate Fod, when the CMD line output drivers are push-pull instead of open-drain. After the bus is activated, the host asks the cards to send their valid operation conditions (APP_CMD, which is CMD55, with RCA = 0x0000 followed by ACMD41). The response to ACMD41 is the operation condition register of the card. The same command is sent to all new cards in the system. Incompatible cards are sent into Inactive State. The host then issues the command ALL_SEND_CID (CMD2) to each card to get its unique card identification (CID) number. A card that is unidentified (a card that is in Ready State) sends its CID number as the response. After the CID is sent by the card, the card enters Identification State. Thereafter, the host issues SEND_RELATIVE_ADDR (CMD3) to ask the card to publish a new relative card address (RCA), which is shorter than CID and addresses 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 the card to have another RCA, it asks the card to publish a new number by sending another SEND_RELATIVE_ADDR (CMD3) command to the card. The last published RCA is the actual RCA of the card.

For each card in the system, the host repeats the identification process (the cycles with CMD2 and CMD3).

In the MMC, the host starts the card identification process in open-drain mode with the identification clock rate Fod. The open-drain driver stages on the CMD line allow parallel card operation during card identification. After the bus is activated, the host asks 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 that are in Ready State) simultaneously begin sending their CID numbers serially while bit-wise monitoring their outgoing bitstream. Cards with outgoing CID bits that do not match the corresponding bits on the command line in any one of the bit periods, stops sending their CID immediately and must wait for the next identification cycle. Because the CID is unique for each card, only one card can 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 this card a relative card address (RCA). Once the RCA is received, the card state changes to the Stand-by State and does not react to further identification cycles, and its output switches from open-drain to push-pull.

The host repeats the process of issuing CMD2 and CMD3 until the host receives a time-out condition to recognize the completion of the identification process.

17.5.2 Data Transfer Mode

All data communication in the data transfer mode is point-to-point between the host and the selected card (using addressed commands). The broadcast command SET_DSR (CMD4) configures the driver stages of all identified cards. It programs their DSRs (Driver Stage Registers) corresponding to the application bus layout (length) and the number at that point. SELECT/DESELECT_CARD (CMD7) is used to select one card and put it into the Transfer State. Only one card can be in the Transfer State at a given time. If a previously selected card is in the Transfer State, the connection with the host is released and the selected card moves back to the Stand-by State. When SELECT/DESELECT_CARD (CMD7) is issued with the reserved relative card address 0x0000, all cards are put back to Stand-by State. This can be used before identifying new cards without resetting other already registered cards.

The card deselection is used if a certain card gets SELECT/DESELECT_CARD (CMD7) with an unmatched RCA. Deselection happens automatically if another card is selected and the CMD lines are common. It is the host's responsibility either to work with a common CMD line (after initialization is done)—in which case the card deselection occurs automatically (as in the MMC system)—or, if the CMD lines are separate, to be aware of the necessity to deselect cards.

17.5.2.1 Block Access: Block Write and Block Read

17.5.2.1.1 Block Write

For block-oriented write data transfers, the CRC check bits are added to each data block. The card performs 1 bit or 4 bits CRC parity check for each received data block prior to the write operation. This mechanism can prevent the writing of erroneously transferred data.

During block write (CMD24–27, 42, 56(w)), 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 can always accept a block of data defined by WRITE_BL_LEN. If the CRC fails, the card indicates the failure on the DAT line; the transferred data is discarded and not written, and all further transmitted blocks (in multiple block write mode) are ignored.

If the host uses partial blocks with an accumulated length that is not block aligned, and if block misalignment is not allowed (the CSD parameter WRITE_BLK_MISALIGN is not set), the card detects the block misalignment error and aborts programming before the beginning of the first misaligned block. The card sets the ADDRESS_ERROR error bit in the status register, ignores all further data transfers, and waits in the receive-data state for a stop command. The write operation is also aborted if the host tries to write over a write-protected area. When this occurs, the card sets the WP_VIOLATION bit.

Programming the CID and CSD registers does not require a previous block length setting. The transferred data is also CRC protected. If 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 reports an error and does not change any register content. Some cards require long and unpredictable amounts of time to write a block of data. After receiving a block of data and completing the CRC check, the card begins to write 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 can poll the status of the card with a SEND_STATUS command (CMD13) at any time, and the card responds 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 can deselect the card by issuing CMD7 (to select a different card), which displaces the card into the disconnect state and release the DAT line without interrupting the write operation. When reselecting the card, the host reactivates the busy indication by pulling DAT low when programming is still in progress and the write buffer is unavailable.

17.5.2.1.2 Block Read

Block read is a block-oriented data transfer. The data transfer format for block read is similar to the data write format. The basic unit of data transfer is a block with a maximum size defined in the CSD (READ_BL_LEN). If READ_BL_PARTIAL is set, smaller blocks with starting and ending addresses entirely contained within one physical block (as defined by READ_BL_LEN) can also be transmitted. It is required that SD memory cards have the capability to transfer blocks of 512 bytes. 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 are continuously transferred until a STOP_TRANSMISSION (CMD12) command is issued. 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 host uses partial blocks with an accumulated length that is not block aligned and block misalignment is not allowed the card detects a block misalignment at the beginning of the first misaligned block, sets the ADDRESS_ERROR error bit in the status register, aborts transmission, and waits in the data state for a stop command.

17.5.2.2 Stream Access: Stream Write and Stream Read

17.5.2.2.1 Stream Write

The stream write command WRITE_DAT_UNTIL_STOP (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 (the CSD parameter WRITE_BL_PARTIAL is set), the data stream can start and stop at any address within the card address space; otherwise it starts and stops only at block boundaries. Because the amount of data to be transferred is not determined in advance, CRC cannot be used. If the end of the memory range is reached when 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:

max. speed = min. (TRAN_SPEED, (8 * 2 WRITE_BL_LEN - NSAC) / (TAAC * R2W_FACTOR))

Table 17-2. Selected Parameters in CSD Field

Field	Name	Width	Cell Type[1]	CSD Slice
TRANS_SPEED	Max data transfer rate	8	r	[103:96]
NSAC	Data read access-time-2 in CLK cycles (NSAC * 100)	8	r	[111:104]
TAAC	Data read access-time-1	8	r	[119:112]
R2W_FACTOR	Write speed factor	3	w	[28:26]

1. r = readable; w = writeable

If the host attempts to use a higher frequency, the card might not be able to process the data and stops programming; sets the OVERRUN error bit in the status register, ignores all further data transfers, and waits (in the receive-data state) for a stop command. The write operation also aborts if the host tries to write over a write-protected area. When this occurs, the card sets the WP_VIOLATION bit.

17.5.2.2.2 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 when sending data and no stop command has been sent 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. (TRANS_SPEED, (8 * 2 $^{READ\_BL\_LEN}$ - NSAC) / TAAC)

These parameters are defined in Table 17-2. If the host attempts to use a higher frequency, the card might not be able to sustain the data transfer. If this happens, the card sets the UNDERRUN error bit in the status register, aborts the transmission, and waits in the data state for a stop command.

17.5.2.3 Erase: Group Erase and Sector Erase

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 can 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 are selected for erasing. After a range is selected, an individual sector (or group) within that range can be spared from erasure 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 sets the ERASE_SEQ_ERROR bit in the status register and resets the entire sequence. If an out of sequence command (except SEND_STATUS) is received, the card sets the ERASE_RESET status bit in the status register, resets the erase sequence, and executes the last command. If the erase range includes write-protected sectors, they are left intact; only the unprotected sectors are erased. The WP_ERASE_SKIP status bit in the status register is set. The address field in the tag commands is a sector or a group address in byte units. The card ignores all LSBs below the group or sector size.

The number of untag commands (CMD34 and CMD37) that are used in a sequence is limited to a maximum of 16. As described in Section 17.5.2.1.1, “Block Write,” the card indicates that an erase is in progress by holding DAT low. The actual erase time can be quite long, and the host can issue CMD7 to deselect the card.

17.5.3 Protection Management

Three write-protect methods are supported in the host for MMC/SD cards: card internal write protect (the card's responsibility), mechanical write protect switch (the host's responsibility only), and password protection card lock operation.

17.5.3.1 Card Internal Write Protection

Card data can be protected against erasure or writing. The entire card can be permanently write protected by the manufacturer or content provider by setting the permanent or temporary write-protect bits in the CSD. For cards that support the write protection of groups of sectors by setting the WP_GRP_ENABLE bit in the CSD, portions of the data can be protected (in units of WP_GRP_SIZE sectors as specified in the CSD), and the write protection can 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 sends 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 ignores all LSBs below the group size.

17.5.3.2 Mechanical Write Protect Switch

The user can use a mechanical sliding tablet on the side of the card to indicate whether a given card is write protected or not. If the sliding tablet is positioned such that the window is open, the card is write protected. If the window is closed, the card is not write protected.

A properly matched switch on the socket side indicates to the host whether 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 unknown to the internal circuitry of the card.

17.5.3.3 Password Protection

The password protection feature enables the host to lock a card when providing a password, which later can be used for unlocking the card. The password and its size are kept in the 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, and so on, but not to access data on the card. If the password was previously set (the value of PWD_LEN is not 0), the card is locked automatically after power on. Similar to the existing CSD and CID register write commands, the lock/unlock command is available only in transfer state. This means that it does not include an address argument, and the card has to be selected before being used. 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, and so on). Table 17-3 describes the structure of the command data block.

Table 17-3. Structure of Command Data Block

Byte Number	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

Table 17-3. Structure of Command Data Block (Continued)

1	PWD_LEN
2	Password Data
...	
PWD_LEN + 1	

The following setting definitions apply:

- **ERASE:** A value of 1 defines Forced Erase Operation (all other bits shall be 0), and only the command byte is sent.
- **LOCK/UNLOCK**: A value of 1 locks the card. A value of 0 unlocks 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**: A value of 1 clears PWD.
- **SET_PWD**: A value of 1 set a 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 is defined by the host before it sends the card lock/unlock command. This allows different password sizes.

The following sections define the various lock/unlock command sequences.

17.5.3.3.1 Setting a Password

The command sequence for setting and changing passwords is as follows:

1. Select a card (CMD7) if one was not selected already.
2. Define the block length (CMD16) given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the new password. In the case of a password replacement, the block size must take into account that both passwords, the old and the new one, are sent with the command.
3. Send the card lock/unlock command with the appropriate data block size on the data line, including 16-bit CRC. The data block must indicate the mode (SET_PWD), the length (PWD_LEN), and the password itself. In the case of a password replacement, the length value (PWD_LEN) includes both the old and new passwords, and the PWD field includes the old password (currently used) followed by the new password.

If the sent, old password is not correct (not equal in size and content), then the LOCK_UNLOCK_FAILED error bit is set in the status register, and the old password does not change. If PWD matches the sent, old password, then the given new password and its size are 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 0, the card locks itself after power up. It is possible to lock the card immediately in the current power session by setting the LOCK/UNLOCK bit (when setting the password) or by sending an additional command for card locking.

Note that it is possible to set the password and to lock the card in the same sequence. See Section 17.5.3.3.3, “Locking a Card.”

17.5.3.3.2 Resetting a Password

The command sequence for resetting passwords is as follows:

1. Select a card (CMD7) if one was not selected already.
2. Define the block length (CMD16) given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the currently used password.
3. Send the card lock/unlock command with the appropriate data block size on the data line, including 16-bit CRC. The data block indicates 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, the LOCK_UNLOCK_FAILED error bit is set in the status register.

17.5.3.3.3 Locking a Card

The command sequence for locking a card is as follows:

— Select a card (CMD7) if one was not selected already.

— Define the block length (CMD16) given by the 8-bit card lock/unlock mode, the 8-bit password 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 indicates the mode LOCK, the length (PWD_LEN), and the password (PWD) itself.

If the PWD content equals the sent password, then the card becomes locked and the card-locked status bit is set in the status register. If the password is not correct, then the LOCK_UNLOCK_FAILED error bit is set in the status register.

Note that it is possible to set the password and to lock the card in the same sequence. In this case the host performs all the required steps for setting the password, as described in Section 17.5.3.3.1, “Setting a Password,” including the bit LOCK set, when the new password command is sent. If the password was previously set (PWD_LEN is not 0), then the card is locked automatically after a power on reset.

An attempt to lock a card that is already locked or to lock a card that does not have a password fails, and the LOCK_UNLOCK_FAILED error bit is set in the status register.

17.5.3.3.4 Unlocking a Card

The command sequence for unlocking a card is as follows:

1. Select a card (CMD7) if one was not selected already.
2. Define the block length (CMD16) given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the currently used password.
3. Send the card lock/unlock command with the appropriate data block size on the data line, including 16-bit CRC. The data block indicates the mode UNLOCK, the length (PWD_LEN), and the password (PWD) itself.

If the PWD content equals the sent password, then the card becomes unlocked and the card-locked status bit is cleared in the status register. If the password is not correct, then the LOCK_UNLOCK_FAILED error bit is 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 is locked automatically on the next power up. The only way to unlock the card permanently is by clearing the password.

An attempt to unlock a card that is already unlocked fails, and the LOCK_UNLOCK_FAILED error bit is set in the status register.

17.5.3.3.5 Forced Erase

If the user forgets the password (the PWD content), it is possible to erase all card data content along with the PWD content. This operation is called Forced Erase:

1. Select a card (CMD7) if one was not selected already.
2. Define the block length (CMD16) to 1 byte (the 8-bit card lock/unlock command). Send the card lock/unlock command with the appropriate data block of 1 byte on the data line, including 16-bit CRC. The data block indicates the mode ERASE (the ERASE bit is the only bit set).

If the ERASE bit is not the only bit in the data field, then the LOCK_UNLOCK_FAILED error bit is set in the status register and the erase request is rejected. If the command is accepted, then *all card content will be erased* including the PWD and PWD_LEN register content, and the locked card is unlocked.

An attempt to force erase on an unlocked card fails, and the LOCK_UNLOCK_FAILED error bit is set in the status register.

17.5.4 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 can 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 are in accordance with the CSD entry SPEC_VERS, indicating the version of the response formats.

Table 17-4 defines the different entries of the status. The abbreviations of the type and clear condition fields in the table are defined 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 X bits.
- Clear condition:
  - A: According to the card's current state.
  - B: Always related to the previous command. Reception of a valid command clears it (with a delay of one command).
  - C: Clear by read.

Table 17-4. MMC/SD Card Status Entries

Bit or Field	Identifier	Type	Value	Description	Clear Condition
31	OUT_OF_RANGE	E R	1 = Error	The command's argument was out of the allowed range for this card.	C

Table 17-4. MMC/SD Card Status Entries (Continued)

Bit or Field	Identifier	Type	Value	Description	Clear Condition
30	ADDRESS_ERROR	E R X	1 = Error	A misaligned address that 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 an 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, this bit 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 when 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
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

Table 17-4. MMC/SD Card Status Entries (Continued)

Bit or Field	Identifier	Type	Value	Description	Clear Condition
16	CID/CSD_OVERWRITE	E R X	1 = Error	Can be of the following errors: • The CID register has been already written and cannot be overwritten. • The read-only section of the CSD does not match the card content. • An attempt was made to reverse the copy (set as original) or permanent WP (unprotected) bits.	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 an 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 is visible to the host in the response to the next command. The 4 bits are interpreted as a 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	When set, this bit causes the card to expect ACMD, or it indicates that the command has been interpreted as ACMD.	C
4–0	Reserved				

17.5.5 I/O Control (Interrupt Mode)

The interrupt mode on the MMC/SD system enables the master (MMC/SD host) to grant the transmission allowance to all slaves (cards) simultaneously. This mode reduces the polling load for the host—and, hence, the power consumption of the system—while maintaining adequate responsiveness of the host to a

card's request for service. Supporting MMC/SD interrupt mode is an option, both for the host and the cards.

- The host must ensure that all cards (including those cards that do not support interrupt mode) are in the Stand-by State before issuing the GO_IRQ_STATE (CMD40) command. When waiting for an interrupt response from a card, the host must keep the clock signal active. The clock rate can be changed according to the required response time.
- The host sets the cards into interrupt mode using the GO_IRQ_STATE (CMD40) command.
- All cards in the Wait-IRQ-State are waiting for an internal interrupt trigger event. Once the event occurs, the card starts to send its response to the host. This response is sent in the open-drain mode.
- When waiting for the internal interrupt event, the cards are also waiting for a start bit on the command line. On reception of a start bit (generated by either another card or the host), the card aborts interrupt mode and switches to the Stand-by State.
- Because the interrupt request message (response to command 40) is being sent in open-drain mode, the host receives a single valid response even when multiple cards respond simultaneously. (This can happen in the rare event that an interrupt event occurs in multiple cards that are synchronizing their start bit, preventing the cards from detecting each other's start bit.)
- Regardless of winning or losing bus control during the CMD40 response, all cards switch to the Stand-by State (as opposed to CMD2).
- After the interrupt response is received by the host, the host returns to the standard data communication procedure.
- If the host wants to terminate the interrupt mode before an interrupt response is received, it can generate the CMD40 response by itself (with card bit = 0) using the reserved RCA address 0x000. This brings all cards from Wait-IRQ-State back into Stand-by-State. Now the host resumes the standard communication procedure.

17.5.6 Application-Specific Command Handling

The MMC/SD system is designed to provide a standard interface for a variety of application types, including customer-specific application features. To enable a common way of implementing these features, two types of generic commands are defined in the MMC standard: Application-Specific Command, ACMD, and General Command, GEN_CMD.

When received by the card, APP_CMD causes the card to interpret the following command as an application-specific command, ACMD. The ACMD has the same structure as regular MMC-standard commands, and it can have the same CMD number. The card recognizes it as an 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 is used. For example, if a card has a definition for ACMD13 but not for ACMD7, and if each of these commands is received immediately after the APP_CMD command, command 13 is interpreted as the non-standard ACMD13 but command 7 is interpreted as the standard CMD7.

In order to use one of the manufacturer-specific ACMDs, the host:

1. Sends APP_CMD. The response will have the APP_CMD bit (new status bit) set, signaling to the host that an ACMD is now expected.
2. Sends the required ACMD. The response will have the APP_CMD bit set, indicating that the accepted command was interpreted as an ACMD. If a non-ACMD is sent, then it will be treated by the card as a normal MMC command, and the APP_CMD bit in the Card Status stays clear.

If an invalid command is sent (neither ACMD nor CMD), then it is handled as a standard MMC 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 differences are that the argument denotes the direction of the data transfer (rather than the address) and the data block is not memory payload data but has a vendor-specific format and meaning.

The card must be selected (tran_state) before sending CMD56. The data block size is the BLOCK_LEN that was defined with CMD16. The response to CMD56 is R1b (card status + busy indication).

17.6 Commands for MMC/SD

Table 17-5 lists and details MMC/SD commands.

Table 17-5. MMC/SD Commands (Sheet 1 of 6)

CMD Index	Type	Argument	Response	Abbreviation	Description
CMD0	BC	[31:0] stuff bits	-	GO_IDLE_STATE	Resets all cards to idle state.
CMD1	BCR	[31:0] OCR without busy	R3	SEND_OP_COND	Asks all 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	[31:16] RCA [15:0] stuff bits	R1	SET_RELATIVE_ADDR	Assigns relative address to the card.
CMD4	BC	[31:16] DSR [15:0] stuff bits	-	SET_DSR	Programs the DSR of all cards.
CMD5–6	Reserved				
CMD7	AC	[31:16] 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.

Table 17-5. MMC/SD Commands (Sheet 2 of 6)

CMD Index	Type	Argument	Response	Abbreviation	Description
CMD8	Reserved				
CMD9	AC	[31:16] RCA [15:0] stuff bits	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	AC	[31:16] 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_TRANSMIS-SION follows.
CMD12	AC	[31:0] stuff bits	R1b	STOP_TRANSMISSION	Forces the card to stop transmission.
CMD13	AC	[31:16] RCA [15:0] stuff bits	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	Reserved				
CMD15	AC	[31:16] 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.

Table 17-5. MMC/SD Commands (Sheet 3 of 6)

CMD Index	Type	Argument	Response	Abbreviation	Description
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_TRANSMIS-SION 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_TRANSMIS-SION follows.
CMD26	ADTC	[31:0] stuff bits	R1	PROGRAM_CID	For programming the card identification register. This command is issued only once per card. The card contains hardware to prevent this operation after the initial programming. Normally this command is reserved for the manufacturer.
CMD27	ADTC	[31:0] stuff bits	R1	PROGRAM_CSD	For programming 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).

Table 17-5. MMC/SD Commands (Sheet 4 of 6)

CMD Index	Type	Argument	Response	Abbreviation	Description
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 erase group, or the address 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.

Table 17-5. MMC/SD Commands (Sheet 5 of 6)

CMD Index	Type	Argument	Response	Abbreviation	Description
CMD39	AC	[31:16] 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. This command addresses 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 that are not defined in the MMC and SD standards.
CMD40	BCR	[31:0] stuff bits	R5	GO_IRQ_STATE	Sets the system into interrupt mode.
CMD41	Reserved				
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.
CMD43–54	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 than 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 or application-specific commands. The size of the data block is set by the SET_BLOCK_LEN command.

Table 17-5. MMC/SD Commands (Sheet 6 of 6)

CMD Index	Type	Argument	Response	Abbreviation	Description
CMD57–63	Reserved				
ACMDs must be preceded by the APP_CMD command. The following commands are used for SD only. Other SD commands not in this list are not supported in this module.					
ACMD6	AC	[31:2] stuff bit [1:0] bus width	R1	SET_BUS_WIDTH	Defines the data bus width (00 = 1-bit or 10 = 4-bit bus) to be used for data transfer. The allowed data bus widths are given in the 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_BLOCKS	Send the number of written (without errors) write blocks. Responds with 32 bits + CRC data block.
ACMD23	AC	[31:23] stuff bits [22:0] number of blocks	R1	SET_WR_BLK_ERASE_COUNT	Set the number of write blocks to be pre-erased before writing (to be used for a faster Multiple Block WR command). 1 = default (one write block)
ACMD41	BCR	[31:0] OCR without busy	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	AC	[31:1] stuff bits [0] set_cd	R1	SET_CLR_CARD_DETECT	Connect (1) or disconnect (0) the 50 kohm pull-up resistor on CD/DAT3 (pin1) of the card. The pull-up can be used for card detection.
ACMD51	ADTC	[31:0] stuff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).

17.7 Programming Model

This section describes the registers for the MMC/SD host controller. Section 17.7.1, “MMC/SD Host Register Set,” summarizes the MMC/SD registry, and Section 17.7.2, “Register Descriptions,” provides detailed information including bit names, addresses, and descriptions. All registers are 16 bits wide and 16-bit aligned. The user accesses on a byte or a word basis.

17.7.1 MMC/SD Host Register Set

The MMC/SD controller is controlled by a set of registers that is configured by the application before each operation on the MMC/SD bus. Table 17-6 describes the registers' addresses and reset states.

Table 17-6. Register Set Summary in MMC/SD Host Controller

Address	Name	Width	Description	Reset Value
0xFFFE0300	STR_STP_CLK	16	MMC/SD Clock Control Register	0x0000
0xFFFE0304	STATUS	16	MMC/SD Status Register	0x0000
0xFFFE0308	CLK_RATE	16	MMC/SD Clock Rate Register	0x0036
0xFFFE030C	-	-	Reserved	-
0xFFFE0310	CMD_DAT_CONT	16	MMC/SD Command and Data Control Register	0x0000
0xFFFE0314	RES_TO	16	MMC/SD Response Time-Out Register	0x0040
0xFFFE0318	READ_TO	16	MMC/SD Read Time-Out Register	0xFFFF
0xFFFE031C	BLK_LEN	16	MMC/SD Block Length Register	0x0000
0xFFFE0320	NOB	16	MMC/SD Number of Blocks Register	0x0000
0xFFFE0324	REV_NO	16	MMC/SD Revision Number Register	0x0380
0xFFFE0328	INT_MASK	16	MMC/SD Interrupt Mask Register	0x0000
0xFFFE032C	CMD	16	MMC/SD Command Number Register	0x0000
0xFFFE0330	ARGUMENTH	16	MMC/SD HIGHER Argument Register	0x0000
0xFFFE0332	ARGUMENTL	16	MMC/SD LOWER Argument Register	0x0000
0xFFFE0334	RES_FIFO	16	MMC/SD Response FIFO Register	0x0000
0xFFFE0338	BUFFER_ACCESS	16	MMC/SD Buffer Access Register	0xXXXX
0xFFFE033C	BUF_PART_FULL	16	MMC/SD Buffer Part Full Register	0x0000

17.7.2 Register Descriptions

The following sections provide details on individual registers.

17.7.2.1 MMC/SD Clock Control Register

The MMC/SD Clock Control Register provides controls for the MMC/SD clock, for MMC/SD reset, and for enabling/disabling the MMC/SD module. The register settings are described in detail in Table 17-7.

NOTE:

To perform a system soft reset and an MMC/SD module enable, a specific sequence is required. The following values must be written to the STR_STP_CLK register in the following sequence: 0x0008, 0x000d, 0x0005.

STR_STP_CLK MMC/SD Clock Control Register **0xFFFE0300**

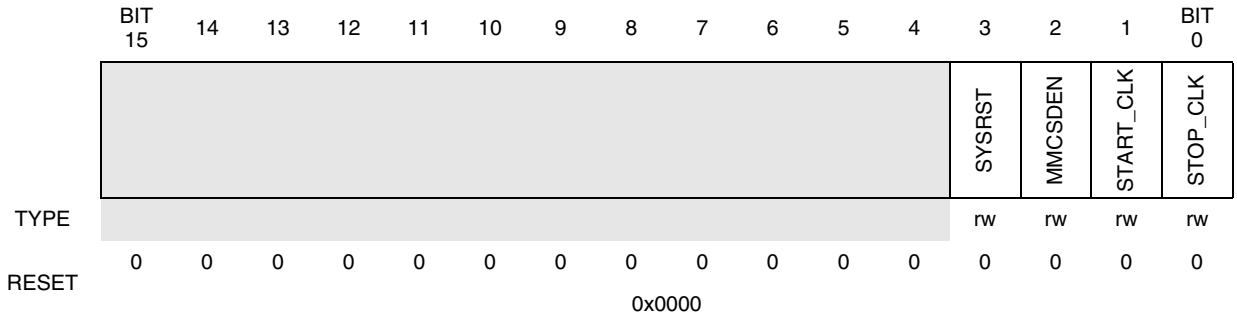


Table 17-7. MMC/SD Clock Control Register Settings

Name	Description	Setting
Reserved Bits 15–4	These bits are reserved and read 0.	Reserved
SYSRST Bit 3	**System Reset**—MMC/SD module reset. When this bit is set to 1, the MMC/SD module resets. When it is set to 0, no reset is initiated (default). **Note:** To perform a system soft reset and an MMC/SD module enable, a specific sequence is required. The following values must be written to the STR_STP_CLK register in the following sequence: • 0x0008 • 0x000d • 0x0005	0 = No reset (default) 1 = Reset MMC/SD module
MMCSDEN Bit 2	**MMC/SD Module Enable**—When this bit is set to 1, the MMC/SD module is enabled. When it is set to 0, the module is disabled. **Note:** To perform a system soft reset and an MMC/SD module enable, a specific sequence is required. The following values must be written to the STR_STP_CLK register in the following sequence: • 0x0008 • 0x000d • 0x0005	0 = MMC/SD module disabled (default) 1 = MMC/SD module enabled

Table 17-7. MMC/SD Clock Control Register Settings (Continued)

Name	Description	Setting
START_CLK Bit 1	**Start Clock**—The START_CLK and STOP_CLK bits (1–0) are used to enable/disable the MMC/SD clock (MMCSD_CLK) during non-transmission mode. For example, during access operation, the MMC/SD bus clock (MMCSD_CLK) is enabled/disabled according to the bit settings of the START_CLK and STOP_CLK bits. MMCSDEN must be set to 1 to use the START_CLK and STOP_CLK bits. **WARNING:** Inserting a value of 11 on bits 1–0 is prohibited.	0 = MMCSD_CLK controlled by internal state machine (default) 1 = MMCSD_CLK enabled
STOP_CLK Bit 0	**Stop Clock**—The START_CLK and STOP_CLK bits (1–0) are used to enable/disable the MMC/SD clock (MMCSD_CLK) during non-transmission mode. For example, during access operation, the MMC/SD bus clock (MMCSD_CLK) is enabled/disabled according to the bit settings of the START_CLK and STOP_CLK bits. MMCSDEN must be set to 1 to use the START_CLK and STOP_CLK bits. **WARNING:** Inserting a value of 11 on bits 1–0 is prohibited.	0 = MMCSD_CLK controlled by internal state machine (default) 1 = MMCSD_CLK disabled

17.7.2.2 MMC/SD Status Register

The MMC/SD Status Register provides the programmer with information about the status of both MMC and SD operations, the application FIFO buffer, and error conditions. The register settings are described in detail in Table 17-8.

STATUS MMC/SD Status Register **0xFFFE0304**

BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
		ECR	AOD	DTD			MMCSDCR	ABFF	ABFE	RCRCERR		CRCRDERR	CRCWRERR	TORERR	TORDDATERR
TYPE		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														

Table 17-8. MMC/SD Status Register Settings

Name	Description	Setting
Reserved Bits 15–14	These bits are reserved and read 0.	Reserved

Table 17-8. MMC/SD Status Register Settings (Continued)

Name	Description	Setting
ECR Bit 13	**End Command Response**—A response package is expected after each successful command operation. ECR is normally received to indicate that a successful command has been transmitted to the card.	0 = No ECR received 1 = ECR received
AOD Bit 12	**Access Operation Done**—This status bit indicates when an access operation is complete.	0 = Access operation not complete or in progress 1 = Access operation complete (default)
DTD Bit 11	**Data Transfer Done**—This status bit indicates that a data transfer is complete. This bit is set to 0 if a data transfer is in progress or not complete.	0 = Data transfer not complete or in progress 1 = Data transfer complete (default)
Reserved Bits 10–9	These bits are reserved and read 0.	
MMCSDCR Bit 8	**MMC/SD Clock Running**—This status bit indicates the status of the MMC/SD clock signal (MMCSD_CLK).	0 = MMCSD_CLK is not running (default) 1 = MMCSD_CLK is running
ABFF Bit 7	**Application Buffer FIFO Full**—This bit indicates when the application buffer FIFO is full. When DMA is enabled, this bit generates a DMA_REQ_B signal to the DMA controller when the buffer is full.	0 = Buffer is not full 1 = Buffer is full
ABFE Bit 6	**Application Buffer FIFO Empty**—This bit indicates when the application buffer FIFO is empty. When DMA is enabled, this bit generates the DMA_REQ_B signal to the DMA controller when the buffer is empty.	0 = Buffer is not empty 1 = Buffer is empty (default)
RCRCERR Bit 5	**Response CRC Error**—This status bit indicates when a response CRC error has occurred. **Note:** Error status can be removed only by an internal status change or by the source of the error being removed.	0 = No error (default) 1 = Response CRC error occurred
Reserved Bit 4	This bit is reserved and reads 0.	Reserved
CRCRDERR Bit 3	**CRC Read Error**—This status bit indicates when a CRC read error has occurred. **Note:** Error status can be removed only by an internal status change or by the source of the error being removed.	0 = No error (default) 1 = CRC read error occurred
CRCWRERR Bit 2	**CRC Write Error**—This status bit indicates when a CRC write error has occurred. **Note:** Error status can be removed only by an internal status change or by the source of the error being removed.	0 = No error (default) 1 = CRC write error occurred

Table 17-8. MMC/SD Status Register Settings (Continued)

Name	Description	Setting
TORERR Bit 1	**Time-Out Response Error**—This status bit indicates if a time-out response error has occurred. **Note:** Error status can only be removed by either internal status change or by the source of the error being removed.	0 = No error (default) 1 = Time-out response error occurred
TORDDATERR Bit 0	**Time-Out Read Data Error**—This status bit indicates if a time-out read data error has occurred. **Note:** Error status can only be removed by either internal status change or by the source of the error being removed.	0 = No error (default) 1 = Time-out read data error occurred

17.7.2.3 MMC/SD Clock Rate Register

The MMC/SD Clock Rate Register contains two fields that control the prescaler and clock rate divider values for the MMC/SD module. The MC68SZ328 internal peripheral clock (SYSCLK) is divided by PRESCALER and CLOCKRATE to obtain MMCSD_CLK. The register settings are described in detail in Table 17-9.

NOTE:

The PRESCALER and CLOCKRATE fields can be only be written to after the application has checked the MMC/SD Status Register (STATUS) to verify that the MMC/SD clock has stopped.

CLK_RATE MMC/SD Clock Rate Register **0xFFFE0308**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
											PRESCALER			CLOCKRATE		
TYPE											rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0
	0x0036															

Table 17-9. MMC/SD Clock Rate Register Settings

Name	Description	Setting
Reserved Bits 15–6	These bits are reserved and read 0.	Reserved
PRESCALER Bits 5–3	**Prescaler**—This field controls the prescaler for the input clock to the MMC/SD module. The input to the prescaler is SYSCLK and the output is CLK.	000 = SYSCLK/1 001 = SYSCLK/2 01x: = SYSCLK/3 100 = SYSCLK/4 101 = SYSCLK/5 11x: = SYSCLK/5 (default is 110)

Table 17-9. MMC/SD Clock Rate Register Settings (Continued)

Name	Description	Setting
CLOCKRATE Bits 2–0	**Clock Rate**—This field controls the clock rate divider. The input to the clock rate divider is CLK and the output is MMCSD_CLK.	000 = CLK/1 001 = CLK/2 010 =CLK /4 011 = CLK/8 100 = CLK/16 101 = CLK/32 11x: = CLK/64 (default is 110)

17.7.2.4 MMC/SD Command and Data Control Register

The MMC/SD Command and Data Control register provides the controls for selecting format response and other commands relating to MMC/SD operation. The register settings are described in detail in Table 17-10.

CMD_DAT_CONT MMC/SD Command and Data Control Register **0xFFFE0310**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
				CRLOFF	STPRDW	STRRW	BUSW		INIT	BSY	STRBLK	WRRD	DATEN	FRES		
TYPE				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 17-10. MMC/SD Command and Data Control Register Settings

Name	Description	Setting
Reserved Bits 15–13	These bits are reserved and read 0.	Reserved
CRLOFF Bit 12	**CmdResLongOff (CRLOFF)**—Set this bit to allow bit clearance when status is read (0 - default).	See description
STPRDW Bit 11	**Stop Read Wait (STPRDW)**—Set this bit to end the read-wait cycle (0 - default).	0 = disabled (default) 1 = end Read-Wait Cycle
STRRW Bit 10	**Start Read Wait (STRRW)**—Set this bit to start the read-wait cycle (0 - default).	0 = disabled (default) 1 = start Read-Wait Cycle
BUSW Bits 9–8	**Bus Width (BUSW)**—This field sets the data bus width.	00 = 1 bit (default) 01 = reserved 10 = 4 bit 11 = reserved
INIT Bit 7	**INITIALIZE**—This bit, when set to 1, enables the 80 bits for initializing the MMC/SD cards.	0 = disabled (default) 1 = 80 bits enabled

Table 17-10. MMC/SD Command and Data Control Register Settings (Continued)

Name	Description	Setting
BSY Bit 6	**Busy Bit**—This bit specifies whether a busy signal is expected after the current command.	0 = Busy signal not expected after current command (default) 1 = Busy signal expected after current command
STRBLK Bit 5	**Stream/Block**—This bit specifies whether the data transfer of the current command is in stream or block mode.	0 = Block mode (default) 1 = Stream mode
WRRD Bit 4	**Write/Read**—This bit specifies whether the data transfer of the current command is a write or read operation.	0 = Read operation (default) 1 = Write operation
DATEN Bit 3	**Data Enable**—This bit specifies whether this command includes a data transfer (1 as true).	0 = no data transfer included 1 = data transfer included
FRES Bits 2–0	**Format of Response**—See Table 17-11.	000 = No Response 001 = Format R1 010 = Format R2 011 = Format R3 100 = Format R4 101 = Format R5 110 = Format R6

Table 17-11. Response Format Settings

Description	FRES MultiMediaCard
No Response	000
Format R1	001
Format R2	010
Format R3	011
Format R4	100
Format R5	101
Format R6	110

17.7.2.5 MMC/SD Response Time-Out Register

RES_TO MMC/SD Response Time Out Register **0xFFFE0314**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
									RESPONSE TIME OUT							
TYPE									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

Table 17-12. MMC/SD Response Time-Out Register Settings

Name	Description	Setting
Reserved Bits 15–8	These bits are reserved and read 0.	Reserved
RESTO Bits 7–0	**Response Time-Out**—This 8-bit field specifies the number of MMC/SD clock counts that can occur after the command before a time-out response error (TORERR) occurs in the MMC/SD status register for the received response. The default value of this register is 64 (decimal).	0x00 = 1 clock count 0x01 = 2 clock counts 0x02 = 3 clock counts : 0xFF = 256 clock counts

17.7.2.6 MMC/SD Read Time-Out Register

READ_TO MMC/SD Read Time Out Register **0xFFFE0318**

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

Table 17-13. MMC/SD Read Time-Out Register Settings

Name	Description	Setting
RDTO Bits 15–0	**Read Time-Out**—This 16-bit register specifies the number of clocks after the command before the host turns on the time-out error for the received data (TORDDATERR). The units are (Master clock/256).	See description

17.7.2.7 MMC/SD Block Length Register

BLK_LEN MMC/SD Block Length Register **0xFFFE031C**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
							BLOCK LENGTH									
TYPE							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 17-14. MMC/SD Block Length Register Settings

Name	Description	Setting
Reserved Bits 15–10	These bits are reserved and read 0.	Reserved

Table 17-14. MMC/SD Block Length Register Settings (Continued)

Name	Description	Setting
BLKLEN Bits 9–0	**Block Length**—This field specifies the “number of bytes” in a block. Normally for MMC/SD data transactions, the number is set to 512. The “number of bytes” in a block is specified in the card's CSD.	0x000 = 0 byte 0x001 = 1 bytes : 0x3FF = 1023 bytes

17.7.2.8 MMC/SD Number of Blocks Register

NOB MMC/SD Number of Blocks Register **0xFFFE0320**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	NUMBER OF BLOCKS															
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 17-15. MMC/SD Number of Blocks Register Settings

Name	Description	Setting
NOB Bits 15–0	**Number of Blocks**—In block mode, this 16-bit register specifies the number of blocks. One block is also a possibility.	0x0000 = 0 blocks 0x0001 = 1 block : 0xFFFF = 65,535 blocks

17.7.2.9 MMC/SD Revision Number Register

REV_NO MMC/SD Revision Number Register **0xFFFE0324**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	HARDWARE REVISION NUMBER															
TYPE																
RESET	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0

Table 17-16. MMC/SD Revision Number Register Settings

Name	Description	Setting
HWREVNUM Bits 15–0	**Hardware Revision Number**—The revision number of this module is fixed to 0x0380.	See description

17.7.2.10 MMC/SD Interrupt Mask Register

Writing to this register twice allow the clearance of a current interrupt $\overline{\text{MMCSDIRQ}}$. To clear the interrupt, rewrite the register with any pattern. The $\overline{\text{MMCSDIRQ}}$ active state clears to inactive state. Programmers should normally use this write operation after $\overline{\text{MMCSDIRQ}}$ is handled.

INT_MASK MMC/SD Interrupt Mask Register **0xFFFE0328**

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

Table 17-17. MMC/SD Interrupt Mask Register Settings

Name	Description	Setting
Reserved Bits 15–4	These bits are reserved and read 0.	Reserved
BUFRDY Bit 3	**Buffer Ready**—Buffer ready interrupt mask	0 = Not masked 1 = Masked (disabled) (default)
ECR Bit 2	**End Command Response**—End command response interrupt mask	0 = Not masked 1 = Masked (disabled) (default)
PDONE Bit 1	**Program Done**—Program done interrupt mask	0 = Not masked 1 = Masked (disabled) (default)
DTRAN Bit 0	**Data Transfer**—Data transfer done interrupt mask	0 = Not masked 1 = Masked (disabled) (default)

17.7.2.11 MMC/SD Command Number Register

CMD MMC/SD Command Number Register **0xFFFE032C**

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

Table 17-18. MMC/SD Command Number Register Settings

Name	Description	Setting
Reserved Bits 15–6	These bits are reserved and read 0.	Reserved
CMDNUM Bits 5–0	**Command Number**—This 6-bit register specifies the command number.	0 = CMD0 1 = CMD1 : 63 = CMD63

17.7.2.12 MMC/SD Higher Argument Register

ARGUMENTH MMC/SD Higher Argument Register **0xFFFE0330**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	HIGHER ARGUMENT															
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 17-19. MMC/SD Higher Argument Register Settings

Name	Description	Setting
HARG Bits 15–0	**Higher Argument**—This register specifies the higher argument in the current command.	See description

17.7.2.13 MMC/SD Lower Argument Register

ARGUMENTL MMC/SD Lower Argument Register **0xFFFE0332**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	LOWER ARGUMENT															
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 17-20. MMC/SD Lower Argument Register Settings

Name	Description	Setting
LARG Bits 15–0	**Lower Argument**—This register specifies the lower argument in the current command.	See description

17.7.2.14 MMC/SD Response FIFO Register

RES_FIFO MMC/SD Response FIFO Register **0xFFFE0334**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	RESPONSE 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

Table 17-21. MMC/SD Response FIFO Register Settings

Name	Description	Setting
RESPONSE CONTENT Bits 15 - 0	**Response Content**—This FIFO register contains the responses after every command that are sent by the host. The size of this FIFO register is 8 x 16-bit.	See description

17.7.2.15 MMC/SD Buffer Access Register

BUFFER_ACCESS MMC/SD Buffer Access Register **0xFFFE0338**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	FIFO CONTENT															
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 17-22. MMC/SD Buffer Access Register Settings

Name	Description	Setting
FIFOCON Bits 15–0	**FIFO Content**—The size of FIFO buffer is 8 x 16-bit. The MMC/SD requires 512 bytes for each sector access. This can be achieved with 32 DMA operations where each DMA operation consists of 8 FIFO accesses of 16 bits per access. During DMA access to this module, the reading of this register clears the DMA request during card reading. The operation is similar to card writing via the “Write” to BUFFER ACCESS.	See description

17.7.2.16 MMC/SD Buffer Partial Full Register

BUF_PART_FULL MMC/SD Buffer Partial Full Register **0xFFFE033C**

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

Table 17-23. MMC/SD Buffer Partial Full Register Settings

Name	Description	Setting
Reserved Bits 15–1	These bits are reserved and read 0.	Reserved
BUFPFULL Bit 0	**Buffer Partial Full**—This 1-bit register is used when a partial buffer is written in stream mode write.	0 = Full buffer written (default) 1 = Partial buffer written

Chapter 18
Memory Stick Host Controller

This chapter describes how data is transferred to a Memory Stick device. It also discusses how to configure and program the Memory Stick host controller (MSHC) module, which has the following features:

- Built-in 8-byte (4-word) FIFO buffers for transmit and receive
- Built-in CRC circuit
- Host bus clock up to 66.32 MHz (BUSCLK maximum setting)
- DMA supported; DMA request condition is selectable based upon FIFO status
- Automatic command execution when an interrupt from the Memory Stick is detected (can be turned on/off)
- Built-in Serial Clock Divider: maximum 33.16 Mhz serial data transfer rate
- External clock source pin for Serial Clock Divider can be input at up to half of BUSCLK
- Protocol is started by writing to the command register from the CPU
- Data is requested by DMA or interrupt to the CPU when entering the data period
- RDY timeout period can be set by the number of serial clock cycles
- Interrupt can also be output to the CPU when a timeout occurs
- CRC can be turned off during test mode
- Two built-in general-purpose input pins for detecting Memory Stick insertion/extraction
- 16-bit host bus access (byte access not supported)

18.1 Block Diagram

Figure 18-1 on page 18-2 is a high-level block diagram of the Memory Stick host controller.

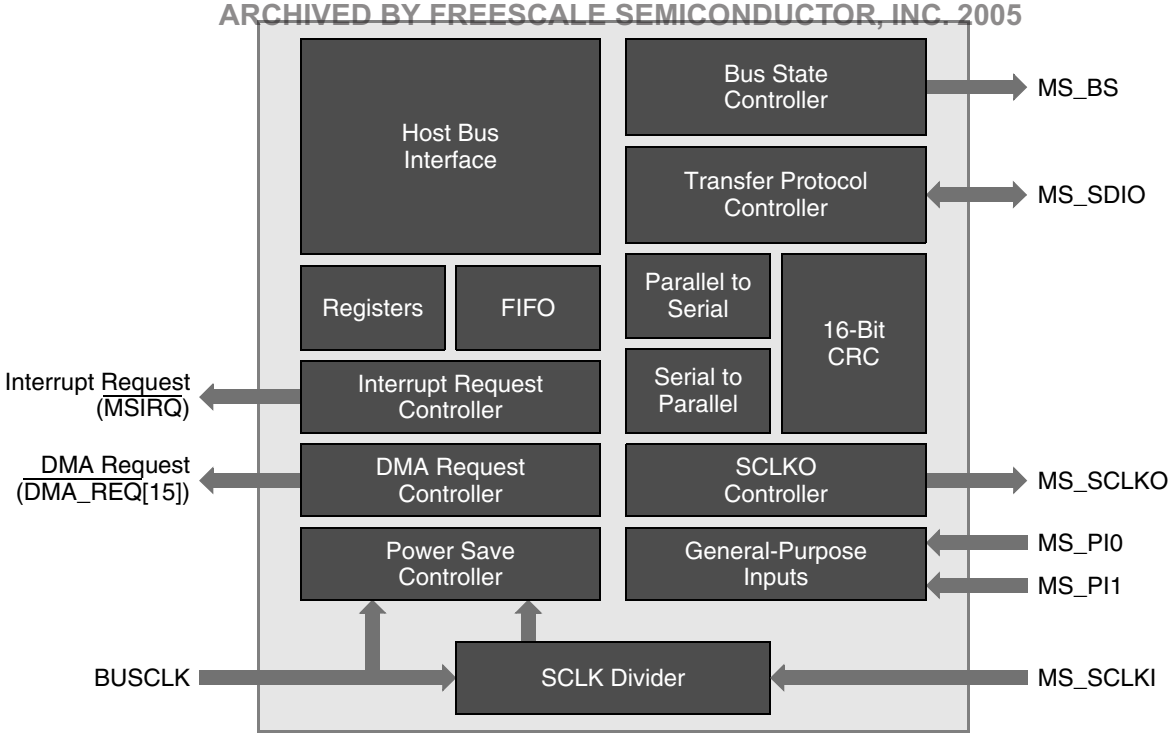


Figure 18-1. Memory Stick Host Controller Simplified Block Diagram

18.2 Memory Stick Interface

The MC68SZ328 provides support for the standard Memory Stick interface. Devices that conform to both the Memory Stick form factor and protocol can be supported. Figure 18-2 depicts the signals required by the Memory Stick hardware interface.

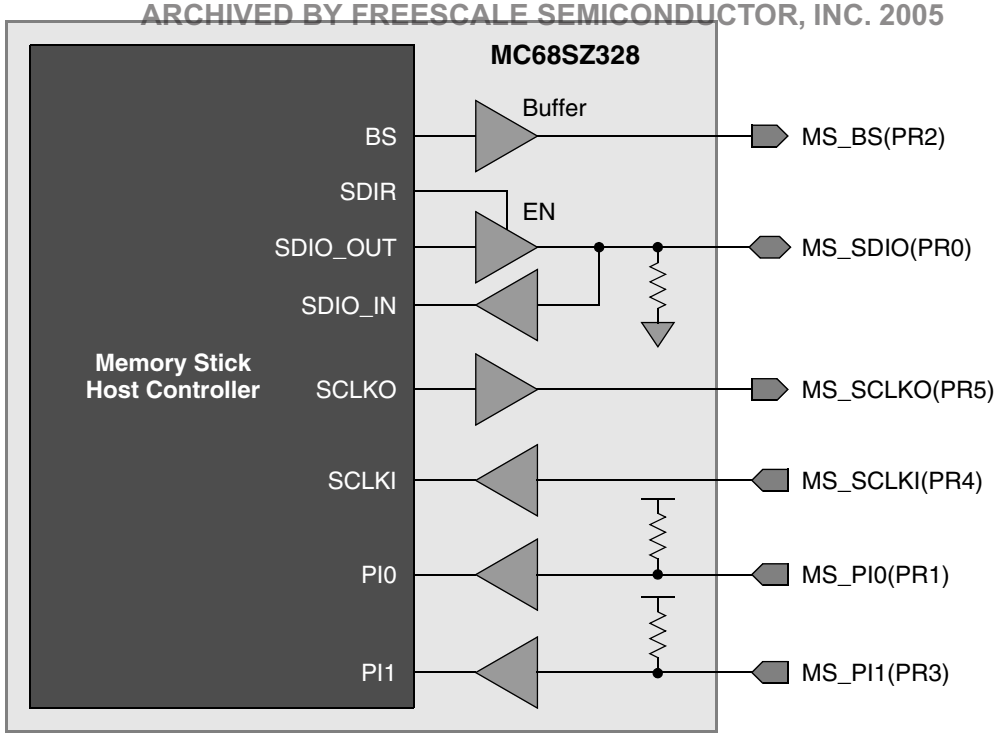


Figure 18-2. Memory Stick Interface

NOTE:

The Memory Stick interface signals are multiplexed with GPIO signals. Refer to Chapter 16, "General Purpose I/O Module," and Section 18.3, "Operation," for detailed information.

18.2.1 Signal Descriptions

The Memory Stick host controller module has six signals that may be used to interface with the external Memory Stick device:

- **MS_BS**—Memory Stick Bus State (Output): Serial bus control signal
- **MS_SDIO**—Memory Stick Serial Data (Input/Output)
- **MS_SCLKO**—Memory Stick Serial Clock (Output): Serial Protocol clock signal
- **MS_SCLKI**—Memory Stick External Clock (Input): External clock source for SCLK Divider
- **MS_PI0**— General-Purpose Input0: Can be used Memory Stick insertion/extraction detection
- **MS_PI1**— General-Purpose Input1: Can be used Memory Stick insertion/extraction detection

18.2.2 Memory Stick Interface I/O Multiplexing

The MSHC I/O pins are muxed with the MMCSD I/O pins in Port R of the MC68SZ328. Table 18-1 shows the multiplexing assignment and how to configure the internal pull-up/pull-down resistor for the MSHC.

Table 18-1. MSHC Port R Multiplexing Assignments

Port R Pin	GPIO Function	Dedicated I/O Function	Internal Resistor Configuration
0	Data bit 0	MMCSD_CLK/MS_SDIO	Pull-down
1	Data bit 1	MMCSD_CMD/MS_PI0	Pull-up
2	Data bit 2	MMCSD_DAT[0]/MS_BS	Disabled
3	Data bit 3	MMCSD_DAT[1]/MS_PI1	Pull-up
4	Data bit 4	MMCSD_DAT[2]/MS_SCLKI	Disabled
5	Data bit 5	MMCSD_DAT[3]/MS_SCLKO	Disabled

Table 18-2 shows pin direction and configuration when MSHC is disabled and enabled. After reset, Port R pins are configured as GPIO pins, with all pins being inputs with internal resistors enabled and MSHC disabled. If the user configures Port R pins as a dedicated function while MSHC is disabled, a signal conflict may arise.

After system reset, if MSHC is disabled and Port R is configured as a dedicated function, the MMC/SD dedicated pins, the PR0 pin, and the PR4 pin direction become output. Also note that Port R can be operated as MSHC dedicated pins only while MSHC is enabled. Therefore, there is a possibility of causing signal conflict on the PR4 pin if it is configured as a dedicated function while MSHC is disabled. To prevent the possibility of signal conflict on a user's application, it is recommended that bit 0 and bit 4 of the GPIO PRSEL register be set as a dedicated function only after the MSHC is enabled.

An example of configuration sequence is as shown below.

To configure the mux of the MSHC I/O to PR[5:0] pins:

1. Enable the Memory Stick Host Controller (set MSCEN bit in MSC2 register)
2. Configure the Internal Pull-Up/Pull-Down Resistors
3. Set Port R Select Register (PRSEL) for Dedicated I/O Function

To prevent signal conflicts when disabling MSHC from its enabled state:

1. Set Port R Select Register (PRSEL) for GPIO Function. (PRDIR register bits should be in input mode)
2. Disable the Memory Stick Host Controller (clear MSCEN bit in MSC2 register)

Table 18-2. Port R Pin Direction and Configuration when MSHC is Disabled/Enabled

GPIO PRSEL	GPIO Function (after Reset)	Dedicated I/O Function	
MSCEN Bit	0 (default) or 1	0 (MSHC Disabled)	1 (MSHC Enabled)
Port R 0 pin	Input with pull-down	Output	Input (MS_SDIO)
Port R 1 pin	Input with pull-up	Input	Input (MS_PI0)
Port R 2 pin	Input with pull-up	Input	Output (MS_BS)
Port R 3 pin	Input with pull-up	Input	Input (MS_PI1)
Port R 4 pin	Input with pull-up	Output	Input (MS_SCLKI)
Port R 5 pin	Input with pull-up	Input	Output (MS_SCLKO)

18.3 Operation

The Memory Stick host controller operation consists of the following:

- Data FIFO operation
- Bus state control operation
- Interrupt operation
- Reset operation
- Power save mode operation
- Auto command function operation
- Serial clock divider operation
- MC68SZ328 system-level DMA transfer operation

18.3.1 Data FIFO Operation

The Memory Stick host controller features a built-in 8-byte (4-word) data FIFO for transmit and receive data. FIFO access is provided through writes/reads to and from the Memory Stick Receive/Transmit FIFO Data register.

The FIFO status flags Receive Buffer Empty (RBE) and Receive Buffer Full (RBF) are provided for receive data to determine the type of accesses allowed or expected. The operation of the status flags is designed so that when RBE = 0, data that is read from the buffer is valid, and when RBE = 1, reads from the buffer are invalid. When RBF = 0 the Receiver FIFO is not full, while RBF = 1 signifies that the Receiver FIFO is full and requires service.

The FIFO status flags for transmit data are Transmit Buffer Empty (TBE) and Transmit Buffer Full (TBF). When TBE = 0, the transmit buffer contains data that is pending transmission. When TBE = 1, the transmit buffer is full and data writes to the buffer are ignored.

18.3.2 Bus State Control Operation

The Memory Stick protocol uses three interface signal line connections for data transfers: MS_BS, MS_SDIO, and MS_SCLKO (or MS_SCLKI). Communication is always started by the Memory Stick host controller (MSHC), and it operates the bus in either Four State Access or Two State Access Mode.

The MS_BS signal line classifies data on the SDIO line into one of four states (BS0, BS1, BS2, or BS3) according to the attribute and transfer direction. BS0 is INT Transfer State; during this state, no packet transmission occurs. During the other three states (BS1, BS2, and BS3), packet communication is executed. BS1, BS2, and BS3 are regarded as one packet length, and one communication transfer is always completed within one packet length (in Four State Access Mode).

The Memory Stick is usually operated in Four State Access Mode from BS0 through BS3. However, if an error occurs during packet communication, the mode is shifted to Two State Access Mode in which states BS0 and BS1 are automatically repeated to avoid bus collisions on SDIO.

18.3.3 MSHC Interrupt Operation

The MSHC interrupt to the MC68SZ328 interrupt controller is shared with the MMCSD interrupt; the default value is $\overline{\text{MMCSDIRQ}}$/$\overline{\text{MSIRQ}}$ (level 5). As with the MMCSD interrupt, the MSHC interrupt is level programmable using Interrupt Level Register 2 (ILCR2) bits 10–8. The default interrupt level is 5. $\overline{\text{MSIRQ}}$ is masked by setting IMR bit 29 (MMCSD/MS) to 1 and enabled by setting IMR bit 29 to 0. $\overline{\text{MSIRQ}}$ status is provided by ISR bit 29 (MMCSD/MS), and pending status is provided in IPR bit 29 (MMCSD/MS). For more information, see Chapter 15, “Interrupt Controller.”

18.3.3.1 Interrupt Sources

The MSHC module provides interrupt source/status flags for the programmer. Generally, after a $\overline{\text{MSIRQ}}$ assertion (interrupt event occurrence), there is distinction in the MSHC about how to clear the interrupt to the core ($\overline{\text{MSIRQ}}$ negate) and clearing the interrupt condition in the MSHC (Interrupt Flag Clear). Table 18-3 summarizes the interrupt sources that may assert $\overline{\text{MSIRQ}}$ to the MC68SZ328 core.

Table 18-3. MSHC Interrupt Sources Summary

Interrupt Flag Name and (Register)	$\overline{\text{MSIRQ}}$ Interrupt Enable Setting(s)	$\overline{\text{MSIRQ}}$ Interrupt Disable Setting(s)	Interrupt Flag Clear	$\overline{\text{MSIRQ}}$ Negate
INT(MSCS)	MSICS[INTEN] = 1	MSICS[INTEN] = 0	Read MSICS	Depends on interrupt source
RDY(MSICS)	MSICS[INTEN] = 1 MSCS[SIEN] = 1	MSICS[INTEN] = 0	Write MSCMD	Read MSICS
SIF(MSICS)	MSICS[INTEN] = 1 MSCS[SIEN] = 1 MSC2[ACD] = 0	MSICS[INTEN] = 0 or MSC2[ACD] = 1	Write MSCMD	Read MSICS
DRQ(MSICS)*	MSICS[DRQSL] = 1 MSICS[INTEN] = 1	MSICS[INTEN] = 0 or MSICS[DRQSL] = 0	- Write TXDATA for Write TPC** - Read RXDATA for Read TPC	Read MSICS or - Write TXDATA for Write TPC - Read RXDATA for Read TPC

Table 18-3. MSHC Interrupt Sources Summary (Continued)

Interrupt Flag Name and (Register)	$\overline{\text{MSIRQ}}$ Interrupt Enable Setting(s)	$\overline{\text{MSIRQ}}$ Interrupt Disable Setting(s)	Interrupt Flag Clear	$\overline{\text{MSIRQ}}$ Negate
PIN(MSICS)	MSICS[INTEN] = 1 MSICS[PINEN] = 1 MSPPCD[PIENx]*** = 1	MSICS[INTEN] = 0	Read MSPPCD	Read MSICS
FAE(MSICS)	MSICS[INTEN] = 1 MSFAECS[FAEEN] = 1	MSICS[INTEN] = 0	Read MSFAECS	Read MSICS
CRC(MSICS)	MSICS[INTEN] = 1 MSCS[SIEN] = 1	MSICS[INTEN] = 0	Write MSCMD	Read MSICS
TOE(MSICS)	MSICS[INTEN] = 1 MSCS[SIEN] = 1 MSCS[BSYCNT] > 0	MSICS[INTEN] = 0	Write MSCMD	Read MSICS

* DRQ(MSICS):
If DAKEN(MSCS) = 0
DRQ(MSICS) = 1 when RxFIFO receives at least 1 word (RFF = don't care) for receive
DRQ(MSICS) = 1 when TxFIFO has at least 1 empty slot available (TFE = don't care) for transmit
If DAKEN(MSCS) = 1
DRQ(MSICS) = 1 when RxFIFO is full (RFF = 1) for receive
DRQ(MSICS) = 1 when TxFIFO is empty (TFE = 1) for transmit
or
DRQ(MSICS) = 1 when RxFIFO receives at least 1 word (RFF = 0) for receive
DRQ(MSICS) = 1 when TxFIFO has at least 1 empty slot available (TFE = 0) for transmit

**TPC: Transfer Protocol Command
***PIENx: PIEN0 or PIEN1 bit of MSPPCD register

18.3.3.2 SDIO Interrupt Operation

An Interrupt Transfer State (INT) from Memory Stick to MSHC may occur during BS0. See Figure 18-3.

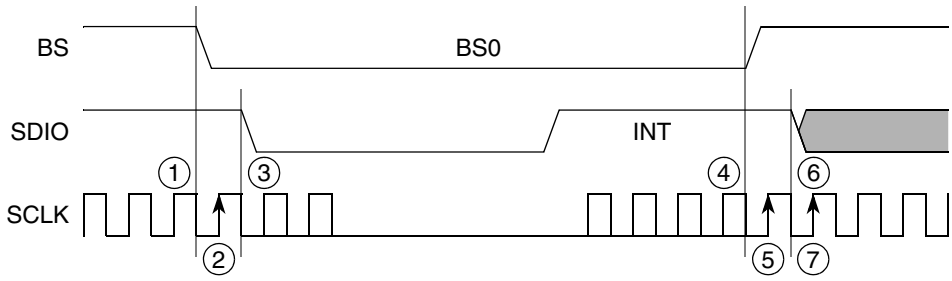


Figure 18-3. Memory Stick Interrupt Transfer State (BS0) Operation

Once the Memory Stick detects BS0 at timing 2, INT Transfer State is started at timing 3. The MSHC may terminate MS_SCLKO after timing 3 (MS_SCLKO = low). An interrupt occurrence is reflected in the Memory Stick Interrupt Control/Status Register, and SDIO is asserted high (interrupt) by the Memory Stick. When SDIO = high (INT) is detected during BS0, the INT Register of the Memory Stick is read and the interrupt factor is checked.The MSHC recognizes INT when SDIO is kept high for 3 SCLK cycles during BS0.

NOTE:

The detection of INT from the Memory Stick is available when the MC68SZ328 is in Normal operation mode or Doze mode and the Memory Stick host controller module is enabled (the MSCEN bit is set). Also the Memory Stick host controller module should be in Normal operation mode (that is, PWS bit = 0, Power Save Disable).

18.3.4 Reset Operation

The RST bit of the Memory Stick Control Status Register (MSCS) provides a mechanism for software resets. When the user writes a 1 to the RST bit, the Memory Stick host controller module will initiate a module and an associated I/O reset.

To reset the Memory Stick module, a value of 1 must be maintained for the RST bit for more than 2 SCLK clocks, and then user software must set to 0 in order to perform the internal reset.

Resetting the Memory Stick module results in the following:

1. Register operation (Status after RST = 1 and immediately after RST = 0)
   - Memory Stick Command Register, MSCMD = 0x0000
   - Memory Stick Control/Status Register, MSCS = 0x050A
   - Memory Stick Receive/Transmit FIFO Data buffer, MSTDATA/MSRDATA = 0x0000
   - Memory Stick Interrupt Control/Status Register, MSICS = 0x0080
   - Memory Stick Parallel Port Control/Data Register, MSPPCD = 0x0000
   - Memory Stick Control Register 2, MSC2 = ACD, RED and LEND bits = 0; MSCEN bit is not changed
   - Memory Stick Auto Command Register, MSACD = 0x7001
   - Memory Stick FIFO Access Error Control/Status Register, MSFAECS = 0x0000
   - Memory Stick Serial Clock Divider Register, MSCLKD = no change
   - Memory Stick DMA Request Control Register, MSDMRQC = 0x0000
2. Output signal status
   - MS_BS -> Low level
   - MS_SDIO_OUT -> Low level
   - MS_SCLK -> Low level
3. Internal operation
   - Internal Interrupt Request signal ($\overline{\text{MSIRQ}}$) -> High level (negated)
   - Internal DMA Request signal -> High level (negated)
   - Transmit/Receive FIFO is cleared
4. The executing protocol is terminated.

18.3.5 Power Save Mode Operation

Figure 18-4 depicts the procedure in Power Save Mode of the Memory Stick host controller.

NOTE:

The Power Save Mode of the Memory Stick controller is a feature of the

Memory Stick host controller and should not be confused with the power save mode features of the MC68SZ328 system as described in Chapter 5, "Clock Generation Module and Power Control Module."

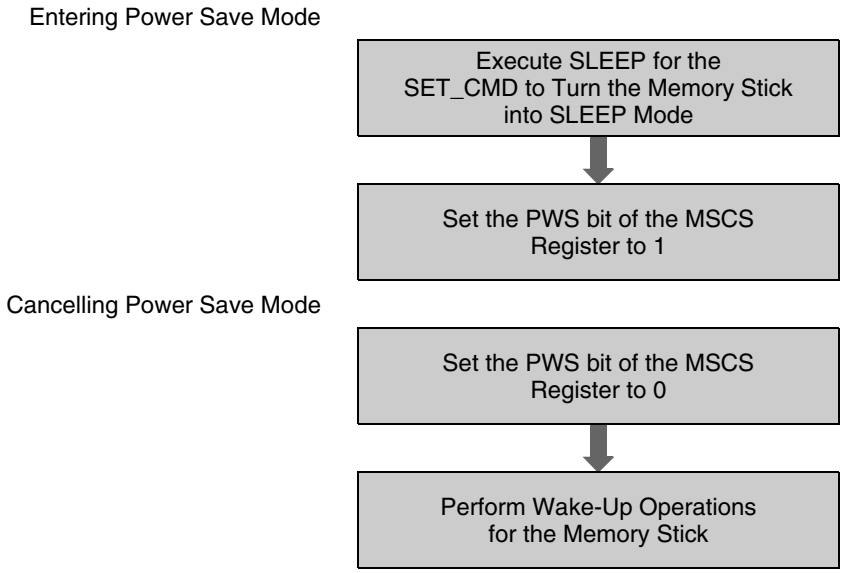


Figure 18-4. Power Save Mode

After the Memory Stick host controller is placed in Power Save Mode (PWS = 1), the Memory Stick cannot be placed in SLEEP mode because the protocol cannot be started. The user must place the Memory Stick into SLEEP mode before placing the Memory Stick host controller in Power Save Mode (PWS = 1). Also, it is necessary to cancel the Power Save mode (PWS = 0) before waking up the Memory Stick.

In Power Save Mode, the Memory Stick host controller can detect an MS_PI[1:0] input status interrupt change. The following table shows MC68SZ328 and MSHC Power Save Mode combinations and whether or not MSHC can detect them.

Table 18-4. Interrupt Detect Capability on Power Save Mode

MC68SZ328 Power Save Mode	MSHC PWS	MS_PI[1:0] Interrupt	MS_SDIO Interrupt
Doze	0 (No PWS)	Detectable	Detectable
Doze	1 (PWS)	Detectable	Not detectable
Sleep	X	Not detectable	Not detectable

18.3.5.1 Register Access During Power Save Mode

Note that the following registers cannot be written while the MSHC is in Power Save Mode (PWS bit = 1):

- MSCMD
- MSTDATA
- MSC2 (except the MSCEN bit)
- MSACD
- MSFAECS
- MSDRQC

18.3.5.2 Register Access while MSHC Module is Disabled

Only the following register should be written while the MSHC module is in disable mode (MSCEN = 0):

- SRC bit and DIV[1:0] bits of MSCLKD register

Setting the MSCEN bit from 1 to 0 causes all of the MSHC registers to initialize except the MSCEN bit of MSC2 Register and the MSCLKD register.

18.3.6 Auto Command Function

The Memory Stick host controller supports the Auto Command function. Auto Command is a function used to automatically execute GET_INT or READ_REG on the host interface for checking status after SET_CMD ends.

With this function, the INT signal from the Memory Stick is detected, and the command set in the ACD Command Register is executed. The result of an automatically executed command (the value of the read register) is put in the receive data buffer.

The time required and CPU load are lower with this function than when the CPU executes SET_CMD and then GET_INT (or READ_REG).

NOTE:

Be sure that READ_SIZE is set to 4 words or less when executing READ_REG using the ACD.

Figure 18-5 indicates the CPU processing and host interface operations when the ACD is used. This figure is for when BLOCK_READ in SET_CMD is executed.

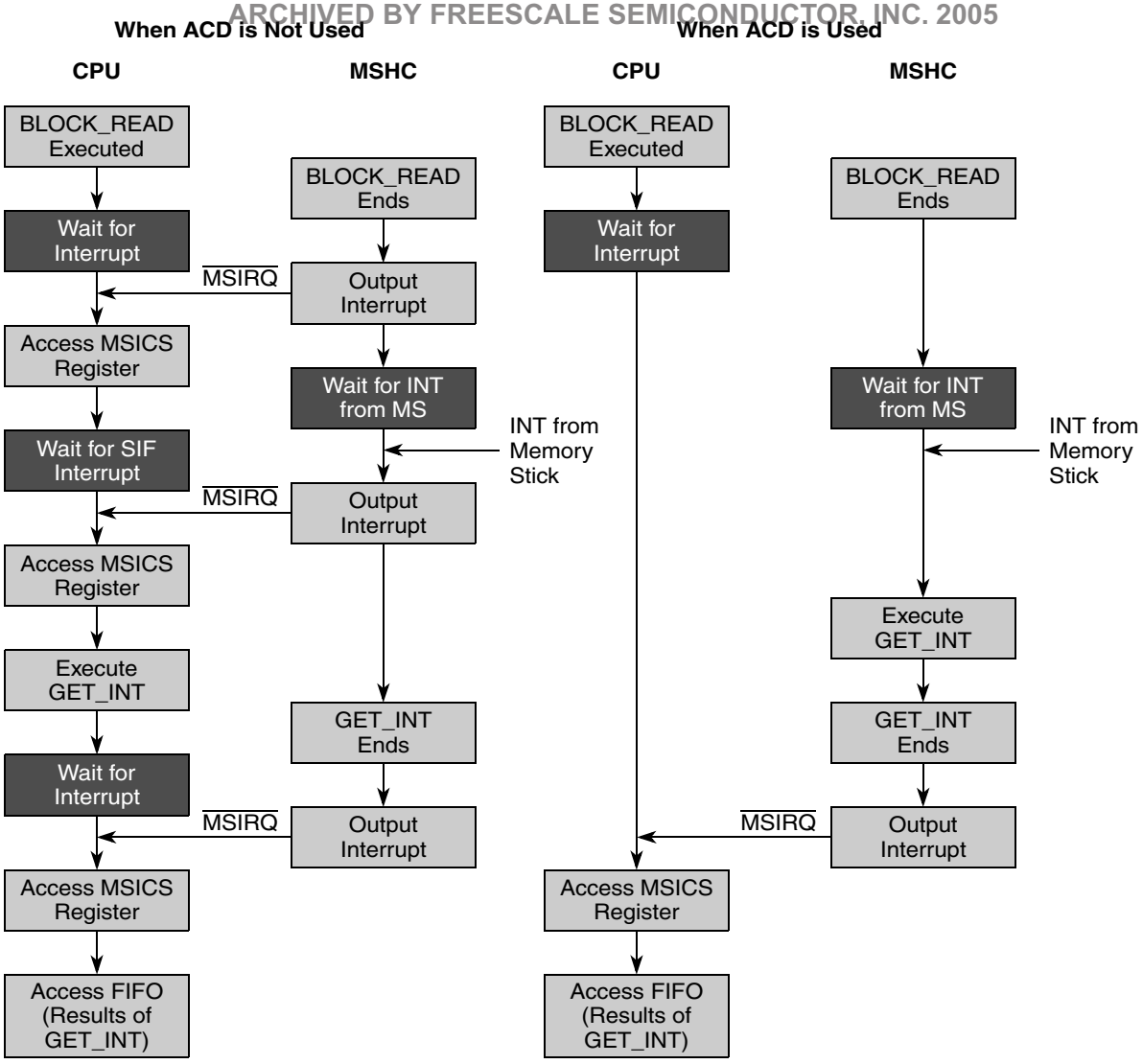


Figure 18-5. Auto Command Function Operation

NOTE:

When a CRC error or TOE occurs, processing terminates without performing ACD and the interrupt signal $\overline{\text{MSIRQ}}$ is output.

Be sure to set the ACD bit of Control Register 2 to 1 immediately before executing ACD (immediately before executing BLOCK_READ in the example in Figure 18-5).

The ACD bit of Memory Stick Control Register 2 is automatically set to 0 after ACD ends (after GET_INT ends in the example in Figure 18-5).

18.3.7 Serial Clock Divider Operation

The MC68SZ328 Memory Stick host controller provides for flexible transfer rate control with a configurable Serial Clock Divider. The divider supports four settings: /1, /2, /4, and /8. It also supports the ability to select one of two clock sources. When internal BUSCLK is used, the /1 setting is not supported.

Figure 18-6 illustrates the Memory Stick Host controller Serial Clock Divider.

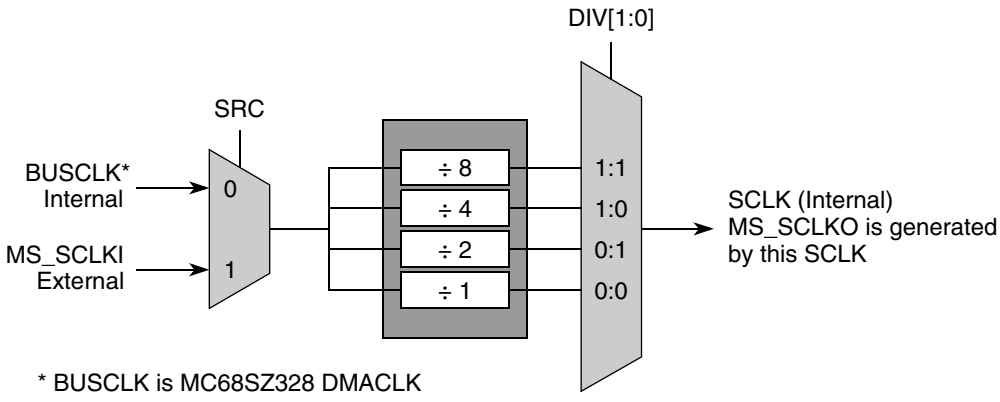


Figure 18-6. Memory Stick Host Controller Serial Clock Divider

Table 18-5. Serial Clock Divider Settings

Source Select Bit (SRC Bit of MSCLKD)	Divide Bits (DIV[1:0] of MSCLKD)	Serial Clock Source	Divide Ratio	SCLKO Output
0	0:0	-	-	not supported
0	0:1	BUSCLK	divide by 2	1/2 BUSCLK
0	1:0	BUSCLK	divide by 4	1/4 BUSCLK
0	1:1	BUSCLK	divide by 8	1/8 BUSCLK
1	0:0	SCLKI	divide by 1	SCLKI
1	0:1	SCLKI	divide by 2	1/2 SCLKI
1	1:0	SCLKI	divide by 4	1/4 SCLKI
1	1:1	SCLKI	divide by 8	1/8 SCLKI

18.3.8 MC68SZ328 System-Level DMA Transfer Operation

The MSHC DMA request bit is assigned to $\overline{\text{DMA\_REQ}}$[15] in the MC68SZ328 DMA request signals. As a result, the I/O Request Source Select (IRSS[4:0]) bits of the I/O Channel Request Source Register should be set to 15. See Chapter 9, “DMA Controller.”

The DMA controller general registers must first be programmed for using all MC68SZ328 I/O peripherals. However, I/O channels must be selected to work solely with the MSHC by the system programmer. Table 18-6 summarizes the important MSHC data for configuring the DMA general registers and DMA I/O registers.

Table 18-6. MSHC DMA Configuration Options

	MSHC RX	MSHC TX
FIFO Size	16 bit	16 bit

Table 18-6. MSHC DMA Configuration Options (Continued)

	MSHC RX	MSHC TX
Memory Size	8, 16 bit	8, 16 bit
DMA Burst Length Setting	2 bytes or 8 bytes	2 bytes or 8 bytes
DMA Source Select Setting	$\overline{\text{DMA\_REQ}}$[15]	$\overline{\text{DMA\_REQ}}$[15]
DMA Channels Available for Use	Channel 2-5 (I/O to Memory channel)	Channel 2-5 (Memory to I/O channel)
Memory Address	User Specified	User Specified
Peripheral Address	0xFFFE0604 (MSRDATA)	0xFFFE0604 (MSTDATA)
Byte Count	User Specified	User Specified
Request Time Out	Supported	Supported
DMA Interrupt	Supported	Supported

18.4 Programming Model

This section describes the registers for the Memory Stick Host controller. Section 18.4.1, “Memory Stick Host Register Set Summary,” summarizes the Memory Stick host registry, while subsequent subsections provide bit names, addresses, and detailed descriptions. All registers are 16 bits wide and 16-bit aligned. Since the Memory Stick host controller does not support byte access, the user should always access by word.

18.4.1 Memory Stick Host Register Set Summary

The Memory Stick host controller is controlled by a set of registers. Table 18-7 describes the registers’ addresses and reset states.

Table 18-7. Register Set in Memory Stick Host Controller

Address	Name	Width	Description	Reset Value
0xFFFE0600	MSCMD	16	Memory Stick Command Register	0x0000
0xFFFE0602	MSCS	16	Memory Stick Control/Status Register	0x050A
0xFFFE0604	MSTDATA	16	Memory Stick Transmit FIFO Data Register	0x0000
0xFFFE0604	MSRDATA	16	Memory Stick Receive FIFO Data Register	0x0000
0xFFFE0606	MSICS	16	Memory Stick Interrupt Control/Status Register	0x0080

Table 18-7. Register Set in Memory Stick Host Controller (Continued)

Address	Name	Width	Description	Reset Value
0xFFFE0608	MSPPCD	16	Memory Stick Parallel Port Control/Data Register	0x0000
0xFFFE060A	MSC2	16	Memory Stick Control 2 Register	0x0000
0xFFFE060C	MSACD	16	Memory Stick Auto Command Register	0x7001
0xFFFE060E	MSFAECS	16	Memory Stick FIFO Access Error Control/Status Register	0x0000
0xFFFE0610	MSCLKD	16	Memory Stick Serial Clock Divider Control Register	0x0002
0xFFFE0612	MSDRQC	16	DMA Request Control Register	0x0000

18.4.2 Memory Stick Command Register

The bit position assignments for the Memory Stick Command Register are shown in the following register display. The settings for this register are described in Table 18-8.

The protocol is started by writing to the Command Register. The data transfer direction is extracted from the PID code. The CRC (16-bit) is transferred during the data period even if the data size is 0. The CRC is disabled if the data size is 0 and the NOCRC bit of the Control Register 1 is 1. Data cannot be written to the Command Register when the RDY bit of the MSICS Register is 0 (while the protocol is executing).

MSCMD Memory Stick Command Register **0x(FF)FE0600**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	PID						DATASIZE									
TYPE	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 18-8. Memory Stick Command Register Description

Name	Description	Setting
PID Bits 15–12	**PID (Packet ID)**—Packet ID code	0001 = RESERVED 0010 = READ_PAGE_DATA 0100 = READ_REG 0111 = GET_INT 1000 = SET_R/W_REG_ADRS 1011 = WRITE_REG 1101 = WRITE_PAGE_DATA 1110 = SET_CMD
Reserved Bits 11–10	Reserved	These bits are reserved and should be set to 0.
DATASIZE Bits 9–0	**Data Size**—Data size value determined for each PID code.	DATASIZE should be byte size.

18.4.3 Memory Stick Control/Status Register

The bit position assignments for the Memory Stick Control/Status Register are shown in the following register display. The settings for this register are described in Table 18-9.

This register is initialized on power up or when the RST bit of Memory Stick Control/Status Register is 1.

MSCS Memory Stick Control/Status Register **0x(FF)FE0602**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	RST	PWS	SIEN	DAKEN	NOCRC	BSYCNT			INT	DRQ			RBE	RBF	TBE	TBF
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	r	r			r	r	r	r
RESET	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0
	0x050A															

Table 18-9. Memory Stick Control/Status Register Description

Name	Description	Setting
RST Bit 15	**Reset**—Internal reset occurs when this bit is set to 1 and canceled when it is set to 0 (initial value). **Note:** A value of '1' must be maintained for the RST bit for more than 2 SCLK clocks and then user software must set to '0' in order to perform the internal reset.	0 = No Reset (default) 1 = Reset Memory Stick module
PWS Bit 14	**Power Save**—Power save mode is enabled when bit is set to 1 and canceled when it is set to 0 (initial value). Data can only be written to the MSCS register and the MSICS register when PWS is 1. It is not possible to write to PWS while the protocol is executing.	0 = Power Save disabled (default) 1 = Power Save enabled
SIEN Bit 13	**Serial Interface Enable**—Serial interface output is enabled when the bit is set to 1 and disabled when it is set to 0 (initial value). Normally set to 1 during operation.	0 = Serial Interface disabled (default) 1 = Serial Interface enabled
DAKEN Bit 12	**XDAK Enable**— This bit configures the internal DMA transfer protocol. The internal DMA acknowledge signal XDAK input is enabled when the bit is set to 1 and disabled when it is set to 0 (initial value). This XDAK signal is used for supporting the 4-words burst DMA transfer. Therefore, if the user needs to configure the DMA transfer as 4-words DMA burst transfer mode, the DAKEN bit must be set to 1. If the user needs to configure the transfer as a 1-word burst DMA transfer mode, the DAKEN bit may be set either to 0 or 1. **Note:** See the note on the RFF and TFE bits of the MSDRQC register.	0 = XDAK Input disabled (default) 1 = XDAK Input enabled
NOCRC Bit 11	**No CRC**—CRC is off when this bit is set to 1. It is on when the bit is set to 0 (initial value). When the bit is set to 1, data is transmitted without adding a CRC (16-bit) at the end of the data array. Normally, this bit remains 0 during operation.	0 = CRC on (default) 1 = CRC off

Table 18-9. Memory Stick Control/Status Register Description (Continued)

Name	Description	Setting
BSYCNT Bits 10–8	**Busy Count**—This field contains the RDY timeout time setting serial clock count (SCLK).This field is set to the maximum BSY timeout time (set value × 4 + 2) to wait until the RDY signal is output from the Memory Stick card. RDY timeout error detection is not performed when BSYCNT = 0. The initial value is 05h. (Exceeding 5 × 4 + 2 = 22 SCLK causes an RDY timeout error.)	000 = No RDY timeout error detection performed 001 = 1 × 4 + 2 = 6 SCLK 010 = 2 × 4 + 2 = 10 SCLK 011 = 3 × 4 + 2 = 14 SCLK 100 = 4 × 4 + 2 = 18 SCLK 101 = 5 × 4 + 2 = 22 SCLK 110 = 6 × 4 + 2 = 26 SCLK 111 = 7 × 4 + 2 = 30 SCLK
INT Bit 7	**Interrupt (Read Only)**—This bit is set to 1 when an interrupt condition is generated. Otherwise, it remains at 0 (Initial value). The bit changes even when the INTEN bit of the MSICS register is 0.	0 = No Interrupt generated (default) 1 = Interrupt generated
DRQ Bit 6	**DMA Request (Read Only)**—This bit is set to 1 when data is requested. Otherwise, it remains at its initial value of 0. If the DRQEN bit of the MSDRQC register is set to 0, the internal DMA request signal is not generated even if this DRQ bit is 1.	0 = No DMA request generated (default) 1 = DMA request generated
Reserved Bits 5–4	Reserved	These bits are reserved and should be set to 0.
RBE Bit 3	**Receive Buffer Empty Flag (Read Only)**—This bit is set to 1 (initial value) when the receive data buffer is empty. It changes to 0 when there is data in the receive data buffer.	0 = Data available in Receiver data buffer 1 = Receiver data buffer EMPTY (default)
RBF Bit 2	**Receive Buffer Full Flag (Read Only)**—This bit is set to 1 when the receive data buffer is full. It changes to 0 when there is space in the receive data buffer (initial value is 0).	0 = Receiver data buffer NOT FULL (default) 1 = Receiver data buffer FULL
TBE Bit 1	**Transmit Buffer Empty Flag (Read Only)**—This bit is set to 1 when the transmit data buffer is empty. It changes to 0 when there is data in the transmit data buffer (initial value is 1).	0 = Data in the Transmit Data buffer 1 = Transmit data buffer EMPTY (default)
TBF Bit 0	**Transmit Buffer Full Flag (Read Only)**—This bit is set to1 when the transmit data buffer is full. It changes to 0 when there is space in the transmit data buffer (initial value is 0).	0 = Transmit data buffer NOT FULL (default) 1 = Transmit data buffer FULL

18.4.4 Memory Stick Transmit FIFO Data Register

The Memory Stick Transmit FIFO Data Register is a 16-bit register. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 18-10.

This register value and the FIFO pointers are initialized on power up or when the RST bit of the Memory Stick Control/Status Register is 1.

Big- or little-endian mode of the FIFO Data Register can be set via the LEND bit of the MSC2 register. The default setting is big-endian. When LEND is 0, the Memory Stick host controller handles the FIFO data in big-endian mode. In big-endian mode, to send only 1 byte of data, the data byte must be written in

bits 15 through 8. When LEND is 1, the Memory Stick host controller handles the FIFO data in little-endian mode. In little-endian mode, to send only 1 byte of data, the byte data must be written in bits 7 through 0.

When the TBF bit in the Memory Stick Control/Status Register is 1, write data is ignored and is not stored to the FIFO. The Transmit FIFO Data Register should be written only when the MSCS Register's DRQ bit or the MSICS Register's DRQ bit is 1, and it should not be written before setting a write command to the MSCMD Register.

MSTDATA Memory Stick Transmit FIFO Data Register **0x(FF)FE0604**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	TX DATA BUFFER															
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 18-10. Memory Stick Transmit FIFO Data Register Description

Name	Description	Setting
TX DATA BUFFER Bits 15–0	**Transmit FIFO Data Buffer**—When the TBF bit in the Memory Stick Control/Status Register = 0, the transmit buffer is available for data writes. When the TBF bit is 1, write data is ignored.	See description

18.4.5 Memory Stick Receive FIFO Data Register

The Memory Stick Receive FIFO Data Register is a 16-bit register. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 18-11.

This register value and the FIFO pointers are initialized on power up or when the RST bit of the Memory Stick Control/Status Register is 1.

Big- or little-endian mode of the FIFO Data Register can be set via the LEND bit of the MSC2 Register. The default setting is big-endian. When LEND is 0, MSHC handles the FIFO data in big-endian mode. In big-endian mode, the first byte of data incoming via the MS_SDIO pin will be received to bits 15 through 8, and the next byte of data will be received to bits 7 through 0 in MSRDATA. Therefore, when only 1 byte of data is received from the Memory Stick, the valid byte data is put onto bits 15 through 8 in MSRDATA. In this instance, bits 7 through 0 contain invalid data. It should be noted that the data in the invalid byte will not be 0.

When the LEND bit is 1, MSHC handles the FIFO data in little-endian mode. In little-endian mode, the first byte of data incoming via the MS_SDIO pin will be received to bits 7 through 0, and the next byte of data will be received to bits 15 through 8 in MSRDATA. Therefore, when only 1 byte of data is received from the Memory Stick, the valid byte data is put onto bits 7 through 0 in MSRDATA. In this instance, bits 15 through 8 contain invalid data. It should be noted that the data in the invalid byte will not be 0.

When RBE is 1, invalid data is read and the FIFO read operation is ignored. The Receive FIFO Data Register should be read only when the MSCS Register's DRQ bit or the MSICS Register's DRQ bit is 1.

MSRDATA Memory Stick Receive FIFO Data Register **0x(FF)FE0604**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	RX DATA BUFFER															
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 18-11. Memory Stick Receive FIFO Data Register Description

Name	Description	Setting
RX DATA BUFFER Bits 15–0	**Receiver FIFO Data Buffer**—When the RBE bit in the Memory Stick Control/Status Register = 0, valid receive data is available in the buffer. When the RBE is 1, invalid data is read.	See description

18.4.6 Memory Stick Interrupt Control/Status Register

The Memory Stick Interrupt Control/Status Register is a 16-bit register. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 18-12.

This register is initialized on power up or when the RST bit of the Memory Stick Control/Status Register is 1. When the MSICS Register is read by the host, the internal interrupt signal $\overline{\text{MSIRQ}}$ is set to high level (negated).

MSICS Memory Stick Interrupt Control/Status Register **0x(FF)FE0606**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	INTEN	DRQSL	PINEN						RDY	SIF	DRQ	PIN		FAE	CRC	TOE
TYPE	rw	rw	rw						r	r	r	r		r	r	r
RESET	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	0x0080															

Table 18-12. Memory Stick Interrupt Control/Status Register Description

Name	Description	Setting
INTEN Bit 15	**$\overline{\text{MSIRQ}}$ Enable**—This bit is set to 1 when the internal interrupt request $\overline{\text{MSIRQ}}$ signal output is enabled; it is 0 when disabled (initial value). The $\overline{\text{MSIRQ}}$ interrupt signal is generated when an interrupt condition occurs after INTEN has been set to 1.	0 = interrupt disabled (default) 1 = interrupt enabled

Table 18-12. Memory Stick Interrupt Control/Status Register Description (Continued)

Name	Description	Setting
DRQSL Bit 14	**Data Transfer Request $\overline{\text{MSIRQ}}$ Enable**—This bit is set to 1 when $\overline{\text{MSIRQ}}$ output is enabled during data transfer request; it is 0 when disabled (initial value). Also, this bit enables operation of the DRQ bit of the MSICS Register. **Note:** The DRQSL bit should be disabled if DMA transfer is used to/from the MSHC's FIFO. For data transfers to the FIFO by CPU (non DMA operation), the DRQSL bit may be enabled. Setting DRQSL to 1 should be used with DAKEN = 1. Upon detection of a Data Transfer Request interrupt (DRQ bit in the MSICS register is set) during the interrupt service routine, the MSICS register must be read after either writing to TXDATA (for Write TPC) or reading from RXDATA (for Read TPC). This operation is needed to accurately detect the status of the interrupt status bits.	0 = disabled (default) 1 = enabled
PINEN Bit 13	**MS_PI[1:0] Level Change $\overline{\text{MSIRQ}}$ Enable**—This bit is set to 1 when $\overline{\text{MSIRQ}}$ output is enabled due to a change in the level at the MS_PI[1:0] pin; it is 0 when disabled (initial value). If a level change on MS_PI[1:0] occurred while PINEN = 0, MSIRQ may be output if the user sets PINEN to 1. To avoid this, the user must wait more than 32 BUSCLK cycles before setting PINEN = 1 after setting 1 to PIENx bit of MSPPCD register.	0 = disabled (default) 1 = enabled
Reserved Bits 12–8	Reserved	These bits are reserved and should be set to 0.
RDY Bit 7	**Ready (Read Only)**—The protocol has ended when this bit is set to 1 (initial value). Communications with Memory Stick are in progress when the bit is set to 0. It is cleared to 0 when writing to the Command Register. $\overline{\text{MSIRQ}}$ asserts when RDY transitions from 0 to 1 to signal that the protocol has ended. An internal interrupt request ($\overline{\text{MSIRQ}}$) for this bit will be negated by reading the MSICS register (if INTEN = 1). **Note:** Data cannot be written to the Command Register when the RDY bit of the MSICS Register is 0 (while the protocol is executing). When using the Auto Command function (ACD bit of MSC2 register is set to 1), the RDY bit will not be set after the SET_CMD TPC ends. The RDY bit is set after the Auto Command PID (APID in MSACD register) is executed.	0 = Protocol in progress 1 = Protocol Ended (default)
SIF Bit 6	**Serial I/F Interrupt (Read Only)**—This bit is set to 1 when the Serial I/F receives INT; otherwise it remains 0 (initial value). For SIF, an interrupt signal is output separately from RDY (see Figure 18-3 on page 18-7). This bit is cleared to 0 when writing to the MSCMD. An internal interrupt request ($\overline{\text{MSIRQ}}$) for this bit will be negated by reading the MSICS register (if INTEN = 1). **Note:** When using the Auto Command function (ACD bit of MSC2 register set to 1), the SIF interrupt will not be generated.	0 = No Serial I/F Interrupt (default) 1 = Serial I/F Interrupt generated

Table 18-12. Memory Stick Interrupt Control/Status Register Description (Continued)

Name	Description	Setting
DRQ Bit 5	**Data Transfer Request (Read Only)**—This bit is set to 1 when a data transfer request condition occurs. It changes to 0 when there is no data transfer request (initial value). The DRQ bit can be changed only when the DRQSL bit of the MSICS Register is set to 1. This bit is cleared to 0 by writing to the FIFO (if PID is a write command) or reading the FIFO (if PID is a read command). After it is cleared, an internal interrupt request ($\overline{\text{MSIRQ}}$) is negated (if DRQSL = 1). Also, the interrupt request ($\overline{\text{MSIRQ}}$) for this bit will be negated by reading the MSICS register. **Note:** If the DRQEN bit of the MSDRQC Register is set to 0, the internal DMA request signal is not generated even if the DRQ bit is 1.	0 = No Data transfer request condition occurs (default) 1 = Data transfer request condition occurs
PIN Bit 4	**Parallel Input**—This bit is set to 1 when the parallel input level changes on pins MS_PI[1:0]. It remains at 0 (initial value) as long as the parallel input level remains unchanged. This bit will be cleared to 0 by reading the MSPPCD. An internal interrupt request ($\overline{\text{MSIRQ}}$) for this bit will be negated by reading the MSICS register (if INTEN = 1). **Note:** The PIN bit can change regardless of the PINEN bit setting.	0 = Parallel Input level unchanged (default) 1 = Parallel Input level change
Reserved Bit 3	Reserved	This bits is reserved and should be set to 0.
FAE Bit 2	**FIFO Access Error (Read Only)**—This bit is set to 1 when a FIFO access error occurs. It remains at 0 (initial value) until an error occurs. This bit is cleared to 0 when the MSFAECS register is read. This status bit is enabled by setting the FAEEN bit (MSFAECS register) to 1, and it can be disabled by setting FAEEN bit to 0. An internal interrupt request ($\overline{\text{MSIRQ}}$) for this bit will be negated by reading the MSICS register (if INTEN = 1).	0 = No FIFO access error (default) 1 = FIFO access error occurred
CRC Bit 1	**CRC Error (Read Only)**—This bit is set to 1 when a CRC error occurs while receiving data; it is 0 when normal (initial value). The bit is cleared to 0 when data is written to the Command Register. The BS output is set to Low level when a CRC error occurs; the RDY sets to 1 and an interrupt signal is output. An internal interrupt request ($\overline{\text{MSIRQ}}$) for this bit will be negated by reading the MSICS register (if INTEN = 1).	0 = No CRC error (default) 1 = CRC error occurred

Table 18-12. Memory Stick Interrupt Control/Status Register Description (Continued)

Name	Description	Setting
TOE Bit 0	**Time Out Error (Read Only)**—This bit is set to 1 when a BSY signal timeout error occurs; it remains at 0 (initial value) during normal operations. This TOE bit is cleared to 0 when data is written to the Command Register. If the number of clocks set determined by the BSYCNT of the Control Register is exceeded and the BSY bit from the Memory Stick continues to remain set, it is taken as a card malfunction and an RDY timeout error (TOE) is sent out. Also, the RDY bit becomes 1 and an interrupt signal is output. An internal interrupt request ($\overline{\text{MSIRQ}}$) for this bit will be negated by reading the MSICS register (if INTEN = 1).	0 = No BSY Timeout error (default) 1 = BSY Timeout Error

18.4.7 Memory Stick Parallel Port Control/Data Register

The Memory Stick Parallel Port Control/Data Register is a 16-bit register. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 18-13. Be aware that the MC68SZ328 supports only the PI[1:0] pin.

This register is initialized on power up or when the RST bit of the Memory Stick Control/Status Register is 1.

The parallel input MS_PIN[1:0] is configured using two flip-flops, each running at 1/16 clock.

- The parallel input pin MS_PI[1:0] is pulled up internally.
- The XPIN[1:0] bit is 1 when the MS_PI[1:0] pin is Low level, and it is 0 when the pin is High level.

It takes 30 BUSCLK cycles for a value from the parallel input pin PI[1:0] to be reflected on bits XPIN[1:0]. This detection is available in following conditions:

- MC68SZ328 is in Normal operation mode or Doze mode, and the Memory Stick host controller module is enabled (MSCEN bit is set).
- Under the preceding condition, the Memory Stick host controller module is in Normal operation mode (PWS bit = 0) or Power Save Mode (PWS bit = 1).

MSPPCD Memory Stick Parallel Port Control/Data Register **0x(FF)FE0608**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
			PIEN1	PIEN0							XPIN1	XPIN0				
TYPE			rw	rw							r	r				
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 18-13. Memory Stick Parallel Port Control/Status Register Description

Name	Description	Setting
Reserved Bits 15–14	Reserved	These bits are reserved and should be set to 0.

Table 18-13. Memory Stick Parallel Port Control/Status Register Description (Continued)

Name	Description	Setting
PIEN1 Bit 13	**Parallel Input Enable 1 (Read/Write)**—Write 1 to this bit to enable parallel port data input (MS_PI1). Write 0 to disable MS_PI1 (initial value).	0 = parallel input port disabled (default) 1 = parallel input port enabled
PIEN0 Bit 12	**Parallel Input Enable 2 (Read/Write)**—Write 1 to this bit to enable parallel port data input (MS_PI0). Write 0 to disable MS_PI0 (initial value).	0 = parallel input port disabled (default) 1 = parallel input port enabled
Reserved Bits 11–6	Reserved	These bits are reserved and should be set to 0.
XPIN1 Bit 5	**MS_PI1 Pin Status (Read Only)**—Input only parallel port data bit. This status bit reports the status of the MS_PI1 pin. XPIN1 is 1 when MS_PI1 is low level. XPI1 is 0 when MS_PI1 is high level.	0 = parallel input port is high level 1 = parallel input port is low level
XPIN0 Bit 4	**MS_PI0 Pin Status (Read Only)**—Input only parallel port data bit. This status bit reports the status of the MS_PI0 pin. XPIN0 is 1 when MS_PI0 is low level. XPI0 is 0 when MS_PI0 is high level.	0 = parallel input port is high level 1 = parallel input port is low level
Reserved Bits 3–0	Reserved	These bits are reserved and should be set to 0.

18.4.8 Memory Stick Control 2 Register

The Memory Stick Control 2 Register is a 16-bit register. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 18-14. For a description of other operation, see Section 18.3.6, “Auto Command Function.”

This register is initialized on power up or when the RST bit of the Memory Stick Control/Status Register is 1.

MSC2 Memory Stick Control 2 Register **0x(FF)FE060A**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	ACD	RED													LEND	MSCEN
TYPE	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. Memory Stick Control 2 Register Description

Name	Description	Setting
ACD Bit 15	**Auto Command**—Write 1 to this bit when a command is to be automatically executed after an INT is detected from the Memory Stick; otherwise write 0 (initial value).	1 = Auto Command enabled 0 = Auto Command disabled (default)

Table 18-14. Memory Stick Control 2 Register Description (Continued)

Name	Description	Setting
RED Bit 14	**Rise Edge Data**—Write 1 to this bit when serial data input is loaded at the falling edge of the clock, and write 0 when it is loaded at the rising edge (initial value).	1 = Serial Data loaded at falling edge of the clock 0 = Serial Data loaded at rising edge of the clock.
Reserved Bits 13–2	Reserved	These bits are reserved and should be set to 0.
LEND Bit 1	**Little Endian Enable**—This read/write bit is used to configure FIFO data between big- and little-endian mode. Set to 1 to handle the FIFO data in little-endian. Set to 0 to handle the FIFO data in big-endian.	1 = Little endian 0 = Big endian (default)
MSCEN Bit 0	**Memory Stick Host Controller Enable**—This bit enables the Memory Stick host controller (MSHC) module. This bit resets to 0. **Note:** The MSCEN bit is *not* reset by setting the RST bit of the MSCS Register.	1 = Memory Stick host controller is enabled 0 = Memory Stick host controller is disabled (default)

18.4.9 Memory Stick Auto Command Register

The Memory Stick Auto Command Register is a 16-bit register. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 18-15.

MSACD Memory Stick Auto Command Register **0x(FF)FE060C**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	APID						ADATASIZE									
TYPE	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1
	0x7001															

Table 18-15. Memory Stick Auto Command Register Description

Name	Description	Setting
APID Bits 15–12	**Auto Command PID**—This field sets the PID to be automatically executed. Initial value is GET_INT (0x7).	0100 = READ_REG 0111 = GET_INT
Reserved Bits 11–10	Reserved	These bits are reserved and should be set to 0.
ADATASIZE Bits 9–0	**Auto Command Data Size**—This field sets the data size. Initial value is 0x001.	ADATASIZE should be byte size.

18.4.10 FIFO Access Error Control/Status Register

The FIFO Access Error Control/Status Register is a 16-bit register. This register's purpose is to detect an invalid FIFO access from the host bus side. For example, when RxFIFO is empty, if the host or DMAC reads the FIFO, the access means that an underrun operation is caused. When TxFIFO is full, if the host or DMAC writes the FIFO, the access means that an overrun operation has occurred. However, since the FIFO's pointer does not advance during invalid accesses, the user does not need to clear the FIFO in such cases. This register is useful for debugging the host's and the DMAC's FIFO access operation.

This register is initialized on power up or when the RST bit of the Memory Stick Control/Status Register is 1.

The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 18-16.

MSFAECS FIFO Access Error Control/Status Register **0x(FF)FE060E**

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

Table 18-16. FIFO Access Error Control/Status Register Description

Name	Description	Setting
Reserved Bits 15–9	Reserved	These bits are reserved and should be set to 0.
FAEEN Bit 8	**FIFO Access Error Detection Enable**—Set this bit to detect a host's invalid FIFO access on the RUN and TOV bits. This bit also enables the FIFO Access Error (FAE) interrupt status bit (MSICS register). Set this bit to 0 to disable detection.	1 = FIFO Access Error Detection disabled (default) 0 = FIFO Access Error Detection enabled
Reserved Bits 7–2	Reserved	These bits are reserved and should be set to 0.
RUN Bit 1	**RxFIFO Underrun Access**—This bit indicates when a host's read access occurs when RxFIFO is empty (RBE = 1). Also, when this bit is set, the FAE interrupt status bit is set in the MSICS register. This detection is available when FAEEN = 1. Writing 1 to this bit clears it.	1 = RxFIFO access error 0 = No RxFIFO access error
TOV Bit 0	**TxFIFO Overrun Access**—This bit indicates when a host's write access occurs when TxFIFO is full (TBF = 1). Also, when this bit is set, the FAE interrupt status bit is set in the MSICS register. This detection is available when FAEEN = 1. Writing 1 to this bit clears it.	1 = TxFIFO access error 0 = No TxFIFO access error (default)

18.4.11 SCLK Divider Control Register

The Serial Clock Divider Control Register is a 16-bit register. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 18-17.

This register is initialized on power up. Note that this register will not be initialized by the RST bit of the Memory Stick Control/Status Register.

MSCLKD SCLK Divider Control Register **0x(FF)FE0610**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	SRC														DIV[1:0]	
TYPE	rw														rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	0x0002															

Table 18-17. SCLK Divider Control Register Description

Name	Description	Setting
SRC Bit 15	**Source Clock of Divider**—Set this bit to select the SCLKI pin as the source clock of the divider. Set it to 0 to select the internal BUSCLK as the source clock of divider. This bit should not be written after setting the MSCEN bit to 1. This bit should be modified only when the MSCEN bit is 0 (disabled). **Note:** The SRC bit is *not* reset by setting the RST bit of the MSCS Register. A DIV setting of 00 is not supported when SRC = 0.	1 = select SCLKI as source clock 0 = select BUSCLK as source clock (default)
Reserved Bits 14–2	Reserved	These bits are reserved and should be set to 0.
DIV[1:0] Bits 1–0	**Divide Ratio**—This field contains the 2^N divide ratio (N = 0,1,2,3). **Note:** SRC = 0 and N = 0 (divide by 1) should not be used. This condition should be used only for debug purpose. This bit should not be written after setting the MSCEN bit to 1. This bit should be modified only when the MSCEN bit is 0 (disabled). DIV bits are *not* reset by setting the RST bit of the MSCS Register. A DIV setting of 00 is not supported when SRC = 0.	00= divide by 1 01= divide by 2 10= divide by 4 (default) 11= divide by 8

18.4.12 DMA Request Control Register

The DMA Request Control Register is a 16-bit register. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 18-18.

This register is initialized on power up or when the RST bit of the Memory Stick Control/Status Register is 1.

When the user uses the MC68SZ328's DMA controller with the MSHC, the DMAC Burst Length Register value must be either "2-byte" or "8-byte" because that MSHC's FIFO depth is 4 words (8 bytes) and the user can configure a DMA request condition as either "1-word" or "4-word." The following describes the MSHC's DMA request operation in a special case.

When MSRDATA is transferred out by the DMA controller and the last burst data is less than the DMAC Burst Length Register's value, the MSHC will generate a DMA request signal for the last burst transfer when it has received the last byte data of a Read type TPC. This DMA request capability is needed to communicate with the DMA controller.

MSDRQC DMA Request Control Register **0x(FF)FE0612**

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

Table 18-18. DMA Request Control Register Description

Name	Description	Setting
DRQEN Bit 15	**DMA Request Enable**—Set this bit to 1 to enable operation of the DMA request signal. **Note:** This bit must be set to 1 before initiating a DMA transfer.	1 = Enables DMA transfer requests 0 = Disables DMA transfer requests
Reserved Bits 14–5	Reserved	These bits are reserved and should be set to 0.
RFF Bit 4	**RxFIFO Full DMA Request**—This bit controls the DMA request signal when PID/APID is a read command. **Note:** This bit is effective only when the DAKEN bit of the MSCS register is 1. If DAKEN is 0, the MSHC generates a DMA request if RFF = 0. Therefore, if the user needs to use the functionality when RFF = 1, the DAKEN bit must be set to 1.	0 = Generates DMA request when RxFIFO has received at least one word 1 = Generates DMA request when RxFIFO is full (4 words)
Reserved Bits 3–1	Reserved	These bits are reserved and should be set to 0.
TFE Bit 0	**TxFIFO Empty DMA Request**—This bit controls the DMA request signal when the PID is a write command. **Note:** This bit is effective only when the DAKEN bit of the MSCS register is 1. If DAKEN is 0, the MSHC generates a DMA request if the TFE bit = 0. Therefore, if the user needs to request when TFE = 1, the DAKEN bit must be set to 1.	0 = Generate DMA request if at least 1 empty slot is available in TxFIFO 1 = Generate DMA request if TxFIFO is empty

18.5 Programmer's Reference

This section provides reference material for the programmer.

18.5.1 Memory Stick Serial Interface Overview

The Memory Stick protocol requires three interface signal line connections for data transfers: MS_BS, MS_SDIO, and MS_SCLKO (or MS_SCLKI). The MS_PI[1:0] pin inputs may be used to detect the insertion and removal of a Memory Stick. Communication is always started from the Memory Stick host controller (MSHC) and will operate the bus in either Four State Access Mode or Two State Access mode.

MS_BS classifies data on SDIO into one of four states (BS0, BS1, BS2, or BS3) according to the attribute and transfer direction. BS0 state has no packet communication occurring, while three states (BS1, BS2, and BS3) have packet communication being executed. BS1, BS2, and BS3 are regarded as one packet, and one communication transfer is always completed within one packet (in Four State Access Mode). See Figure 18-7, Table 18-19, and Table 18-20.

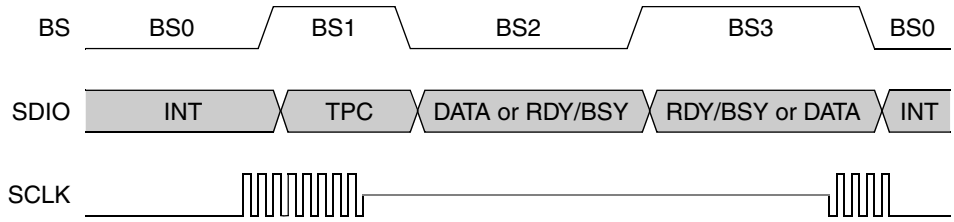


Figure 18-7. Memory Stick Bus Four State Access Protocol

Table 18-19. Serial Interface Signal Specifications

Signal	MC68SZ328 Port	I/O Host (MSHC) Side	Description
MS_SDIO	PR0	Input/output	Serial Data Bus. The direction of the data and the data itself will change at each Bus State. Data is 8 bit, MSB first.
MS_PI0	PR1	Input	Parallel Port Data Input. Memory Stick Insertion/Extraction detect 0
MS_BS	PR2	Output	Indicates Bus State (0:3) on SDIO and its timing of starting transfer
MS_PI1	PR3	Input	Parallel Port Data Input. Memory Stick Insertion/Extraction detect 1
MS_SCLKI	PR4	Input	External Clock input to the serial clock generation circuit.
MS_SCLKO (Serial Clock)	PR5	Output	Signal on MS_BS and MS_SDIO is output on trailing edge and input (latched) at leading edge. It is always output except during BS0 period.

Table 18-20. Four State Access Mode

State	MS_BS	State Name	Description
BS0	LOW	INT Transfer State	A state in which packet communication is not active, and (MS_SDIO) is used as a transmission line for INT signals (interruption)
BS1	HIGH	TPC State	Packet starts and transfers Transfer Protocol Command (TPC) from MSHC to Memory Stick
BS2	LOW	Hand Shake State (Read Protocol)	Waiting for RDY signal
		Data Transfer State (Write Protocol)	Transferring data to Memory Stick
BS3	HIGH	Data Transfer State (Read Protocol)	Reading data from Memory Stick
		Handshake State (Write Protocol)	Waiting for RDY signal

The Memory Stick usually operates in Four State Access Mode from BS0 through BS3. However, if an error occurs during packet communication, the mode is shifted to Two State Access Mode in which states BS0 and BS1 are automatically repeated to avoid bus collisions on SDIO. See Section 18.5.3, “Transfer Protocol Command (TPC),” for additional information.

18.5.2 Protocol

Bus state sequences of write packets that transfer data from MSHC to the Memory Stick differ from those of read packets that transfer data from the Memory Stick to MSHC.

18.5.2.1 Write Packets

Figure 18-8 and Table 18-21 provide information about write packets.

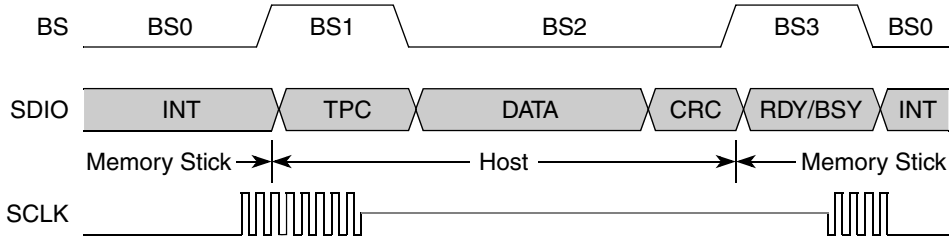


Figure 18-8. Write Packet

Table 18-21. Write Packet Description

Bus State	Direction	Description
BS1 (TPC)	MSHC to Memory Stick	Transfers Write TPC.
BS2 (Data)	MSHC to Memory Stick	Transfer Data + CRC to SDIO from Host.
BS3 (Handshake)	Memory Stick to MSHC	During BSY output (High/Low signal) on SDIO, the Memory Stick decides whether the packet can be terminated normally or not, reflects the result to the corresponding register, and then outputs RDY (a signal inverting at every 1 SCLK) on SDIO.
BS0 (INT)	Memory Stick to MSHC	If some interruption factors occur as a result of Memory Stick inner operation, an INT (HIGH signal) is output on SDIO. During the BS0 period, the SDIO signal line is used as an INT signal line, which does not synchronize with SCLK.

18.5.2.2 Read Packets

Figure 18-9 and Table 18-22 provide information about read packets.

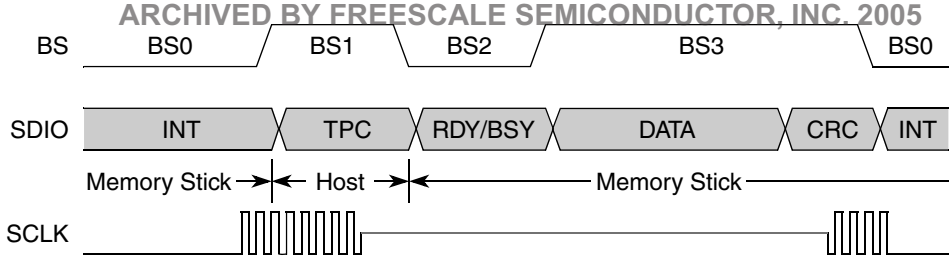


Figure 18-9. Read Packet

Table 18-22. Read Packet Description

Bus State	Direction	Description
BS1 (TPC)	MSHC to Memory Stick	Transfers read TPC.
BS2 (Handshake)	Memory Stick to MSHC	Memory Stick outputs BSY (High/Low signal) on SDIO until reading data is ready to transfer. When ready, the MSHC outputs RDY (inverting signal at every 1 SCLK).
BS3 (Data)	Memory Stick to MSHC	Data + CRC are output on MS_SDIO from the Memory Stick.
BS0 (INT)	Memory Stick to MSHC	If some interruption factors occur as a result of Memory Stick inner operation, an INT (High) signal is output on SDIO. During the BS0 period, the SDIO signal line is used as an INT signal line, which does not synchronize with SCLK.

18.5.3 Transfer Protocol Command (TPC)

MSHC can directly access registers and PageBuffer on the Memory Stick via TPC. TPC code is 8-bit data that is coded by TPC 4-bit data and one's-complement TPC 4-bit data for error checking. Table 18-23 details TPC 4-bit code.

Table 18-23. TPC Code Specification

Name	TPC[3:0]				Operation	Description
READ_PAGE_DATA	0	0	1	0	Transfer from PageBuffer	TPC for reading from PageBuffer in units of a page (512 bytes). Data is a fixed length of 512 bytes + CRC (16-bit).
READ_REG	0	1	0	0	Read Register	TPC for reading from the register whose address was set. Address and Data length are set by SET_R/W_REG_ADRS. Actual Data length: the value which was set + CRC (16-bit).
GET_INT	0	1	1	1	Read INT Register	The only INT register: 1 byte is read. Setting by SET_R/W_REG_ADRS TPC is not necessary. Read INT Register operation is provided as an independent TPC because the INT Register is accessed frequently. Data is a fixed length of 1 byte + CRC (16-bit).

Table 18-23. TPC Code Specification (Continued)

Name	TPC[3:0]				Operation	Description
WRITE_PAGE_DATA	1	1	0	1	Transfer to PageBuffer	TPC for writing to PageBuffer in units of a page (512 bytes). Address and Data are a fixed length of 512 bytes + CRC (16-bit).
WRITE_REG	1	0	1	1	Write Register	TPC for writing to the register whose address was set. Address and Data length are set by SET_R/W_REG_ADRS. Actual Data length: the value which was set + CRC (16-bit).
SET_R/W_REG_ADRS	1	0	0	0	Address setting of READ_REG and WRITE_REG	TPC for setting values that determine the register accessed by WRITE_REG and READ_REG. Values to be set are the following 4 bytes (fixed length). Data is a fixed length of 4 bytes + CRC (16-bit). READ_REG: - Starting address for READ_REG: Starting address of the register to be read - Consecutive size for READ_REG: The number of registers to be read consecutively. WRITE_REG: - Starting address for WRITE_REG: Starting address of the register to be written. - Consecutive size for WRITE_REG: The number of registers to be written consecutively.
SET_CMD	1	1	1	0	Set CMD	CMD to be executed by the Flash Memory Controller, such as an operation for transferring flash memory. Data is a fixed length of CMD (8-bit) + CRC (16-bit). The Flash Memory Controller starts operation when CMD is set by this TPC and posts the result by INT.
Reserved	0	0	0	1	-	-

18.5.4 Protocol Errors

This section provides information about protocol errors.

18.5.4.1 Overview

Since High and Low MS_BS signals express corresponding bus states, a bus collision occurs if a difference in bus state arises between the host and the Memory Stick. To avoid collisions, the Memory Stick shifts to Two State Access Mode automatically when an error occurs in a packet.

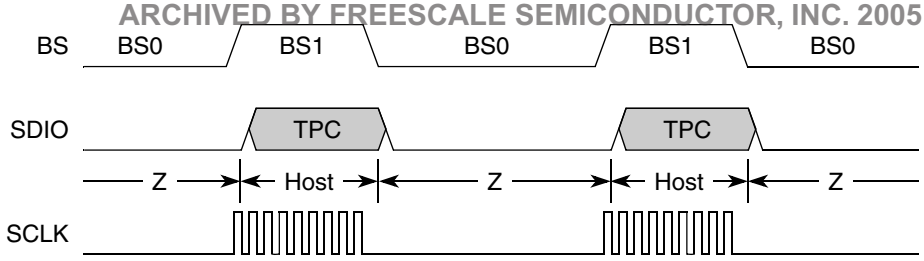


Figure 18-10. Two State Access Mode

In Two State Access Mode, the operation is performed with the recognition that MS_BS = LOW is BS0 and MS_BS = HIGH is BS1. See Table 18-24.

Table 18-24. Bus State in Two State Access Mode

Bus State	Direction	Description
BS0	-	Under normal conditions, BS0 is regarded as high impedance state, regardless of the INT signal output period. There is no output even if the INT signal is active.
BS1 (TPC)	MSHC to Memory Stick	The Memory Stick accepts TPC.

When the Memory Stick shifts to Two State Access Mode, a timeout occurs during the handshake state of the packet on the host (see Figure 18-11 and Figure 18-12), and the failure of the communicating packet is detected.

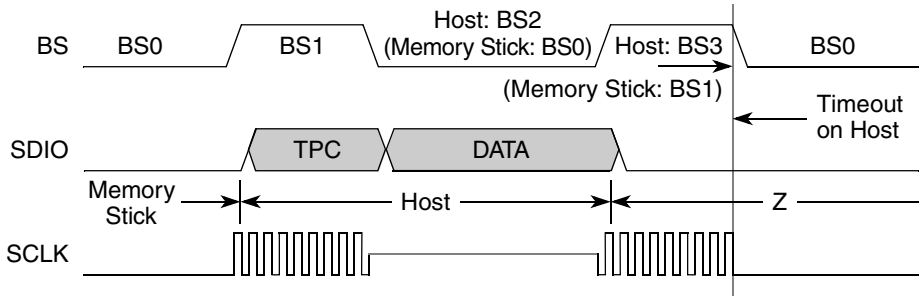


Figure 18-11. Write Packet Timeout

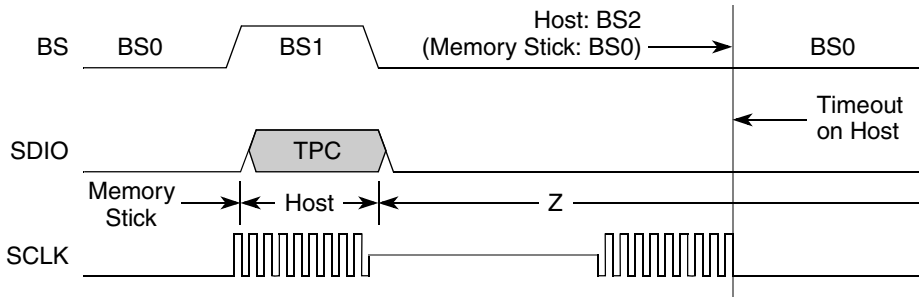


Figure 18-12. Read Packet Timeout

NOTE:

If a timeout occurs in BS2 of a read packet, the bus state will not shift to

18.5.4.2 Two State Access Mode Error Factors

Table 18-25 summarizes Two State Access Mode errors.

Table 18-25. Two State Access Mode Error Factors

Error	Description	
TPC Code Error	4-Bit Error Check Code Error	-
	Undefined TPC	-
	Unacceptable TPC	TPC is received but Memory Stick is not capable of executing it due to internal status
	Short TPC State	When BS1 is under 8 SCLKO
Data Error	Write Packet CRC Error	CRC Error occurred in the data transferred from MSHC
	Short Data State	Not all data is accepted because data state of BS is shorter than the setting on Memory Stick
Handshake Error	Short Handshake State	BS is switched before the output of RDY, though Memory Stick is operating normally
Power Supply On	-	-

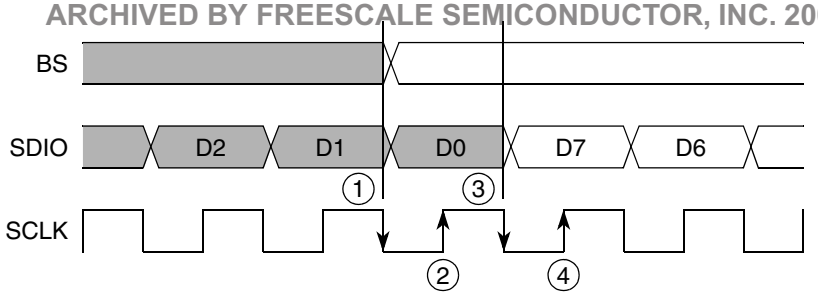
If any error factor described in Table 18-25 does not occur in BS1, the mode will shift to BS2 and BS3 and enter Four State Access Mode. When an error described in Table 18-25 occurs, the Memory Stick will return to Two State Access Mode.

18.5.5 Signal Timing

This section provides signal timing details.

18.5.5.1 Timing

Figure 18-13 illustrates signal timing.



NOTES: (1): Change timing of BS; timing to output LSB of final data.
(2): Timing to detect BS change on Memory Stick side; timing to latch LSB of final data.
(3): Timing to output MSB of first data.
(4): Timing to latch MSB of first data.

Figure 18-13. Signal Timing

The timing of SCLK, SDIO, and BS is as follows:

- The sender outputs the SDIO signal at SCLK fall (output side) and latches it at SCLK rise (input side).
- BS signal is output, synchronizing with SCLK fall.
- Relation between BS change and data: When BS changes to shift to the next state, and the Bus State is not extended, the new BS is output synchronizing with the output timing of the final data LSB on SDIO in the previous state.
- TPC, data, and CRC are MSB first.

18.5.5.2 Bus State Extension

If it is difficult to switch the BS signal in the same timing as the final data, in TPC State and Data State it is possible to continue that Bus State without switching the BS signal even after the final data transfer. However, in Data State, HIGH must be output on SDIO during the period when Bus State is continued without switching after the transfer of the last bit. In TPC State, signals on SDIO in this period are not prescribed. See Figure 18-14.

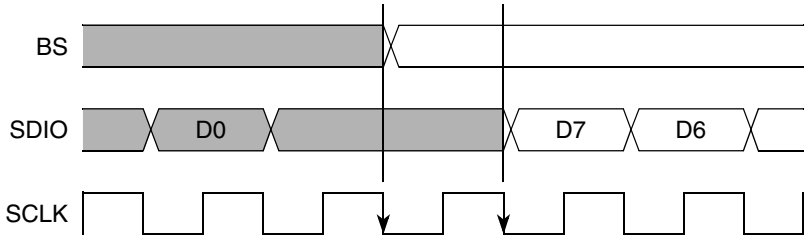


Figure 18-14. Bus State Extension

18.5.5.3 Data Transfer Extension

When data transferred from MSHC cannot be output to catch the next fall of SCLK, or when data transferred from Memory Stick to MSHC cannot be received from the rise of the next SCLK because the buffer is full on MSHC, the next data transfer can be delayed by keeping SCLK high. See Figure 18-15.

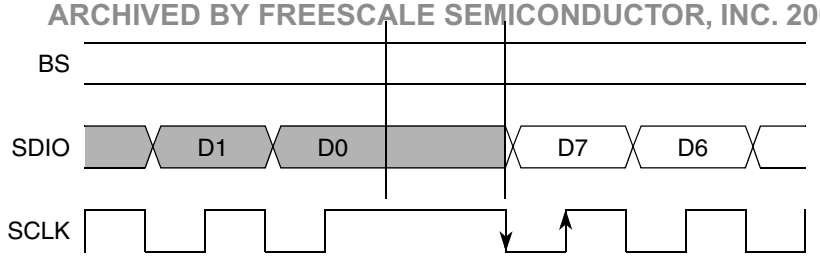


Figure 18-15. SCLK Extension for Data Wait

Chapter 19
Universal Asynchronous Receiver/Transmitter 1 and 2

This chapter describes the universal asynchronous receiver transmitters used to communicate with external serial devices. Because the UART modules are identical, the signal nomenclature throughout this chapter uses an *x* suffix to represent either 1 or 2. For example, TXD*x* represents either TXD1 or TXD2.

Regarding legacy issues with previous DragonBall processors, UART 1 and 2 in the DragonBall Super VZ integrated processor are nearly identical to UART 2 of the DragonBall VZ integrated processor the difference being the FIFO size and DMA support.

19.1 Introduction to the UARTs

This section describes how data is transported in character blocks using the standard “start-stop” format and discusses how to configure and program the UART modules. The UART modules have the following features:

- Fully software compatible with the MC68SZ328 UART core
- Full-duplex operation
- Flexible 5-wire serial interface
- Direct “glueless” support of IrDA physical layer protocol
- Robust receiver data sampling with noise filtering
- 32-byte FIFO for receive, 32-byte FIFO for transmit
- Flexible DMA burst access to both UART 1 and UART 2 FIFO architecture
- “Old data” timer on receive FIFO
- 7- and 8-bit operation with optional parity
- Break generation and detection
- Baud rate generator
- Flexible clocking options
- Standard baud rates of 600 bps to 460.8 kbps with 16x sample clock
- External 1x clock for high-speed synchronous communication
- Eight maskable interrupts
- Low-power idle model
- User selectable RxFIFO and TxFIFO half-mark levels
- The $\overline{\text{RTS}}$ signal can be triggered by either a near RxFIFO full condition or at the level defined by the RxFIFO level marker.

- Non-integer prescaler value can be programmed to 1. This will provide the maximum non-standard baud rate of 4.14 Mbps with 16x sample clock.

Both UART modules perform all of the normal operations associated with start-stop asynchronous communication. Serial data is transmitted and received at standard bit rates using the internal baud rate generator. For applications that require other bit rates, a 1x clock mode provides a data-bit clock. Figure 19-1 is a simplified block diagram that illustrates both UART modules.

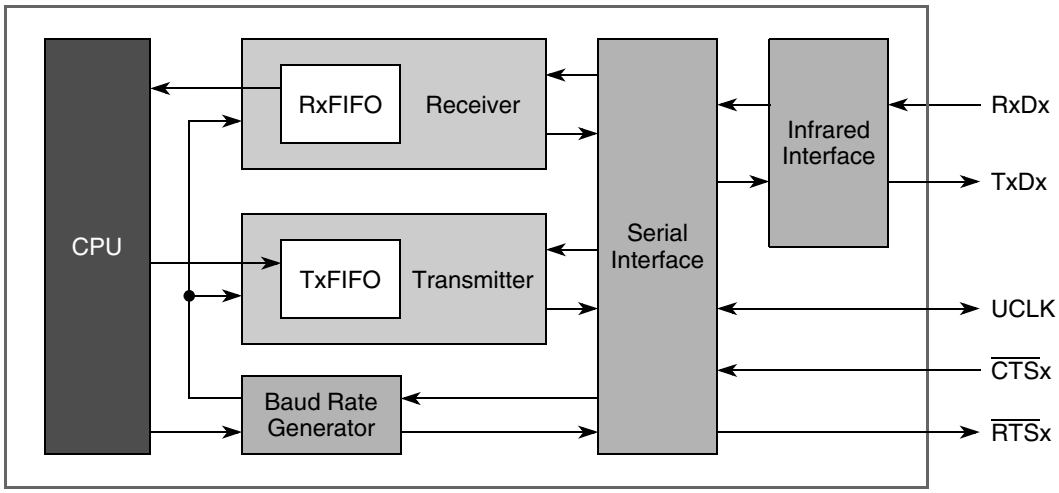


Figure 19-1. UART Simplified Block Diagram

19.2 Serial Operation

The UART modules have two modes of operation, nonreturn to zero (NRZ) and infrared (IrDA). Section 19.2.1, “NRZ Mode,” and Section 19.2.2, “IrDA Mode,” describe these two modes of operation.

19.2.1 NRZ Mode

The NRZ mode is primarily associated with RS-232. Each character is transmitted as a frame delimited by a start bit at the beginning and a stop bit at the end. Data bits are transmitted least significant bit first, and each bit occupies a period of time equal to 1 full bit. If parity is used, the parity bit is transmitted after the most significant bit. Figure 19-2 illustrates a character in NRZ mode.

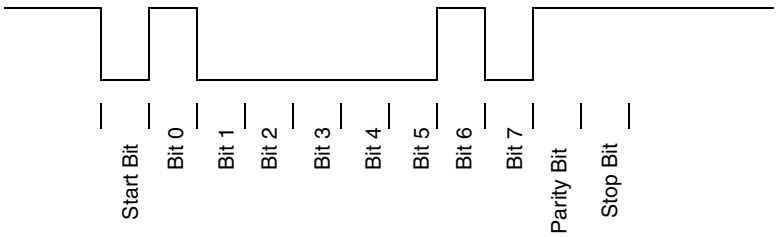


Figure 19-2. NRZ ASCII “A” Character with Odd Parity

19.2.2 IrDA Mode

Infrared (IrDA) mode uses character frames as NRZ mode does, however uses a different method for transmitting. Instead of driving ones and zeros for a full bit-time period, zeros are transmitted as three-sixteenth (or less) bit-time pulses, and ones remain low. The polarity of transmitted pulses and expected receive pulses can be inverted so that a direct connection can be made to external IrDA transceiver modules that use active low pulses. Figure 19-3 illustrates a character in IrDA mode.

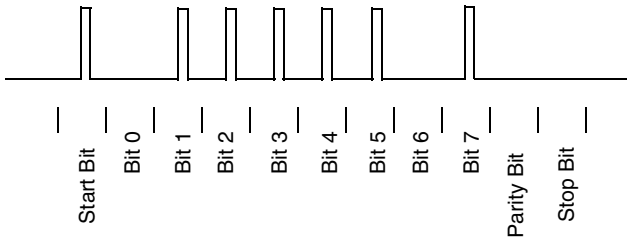


Figure 19-3. IrDA ASCII "A" Character with Odd Parity

19.2.3 Serial Interface Signals

The UART module has five signals that are used to communicate with external UART-compatible devices. The pins of both UART modules operate identically. Exceptions in pin and register nomenclature are noted in the following descriptions:

- **TXD1/TXD2**—The RS-232 transmit data signal multiplexed with PE5 in UART 1 (PJ5 in UART 2), is the RS-232 transmitter serial output. This pin connects to standard RS-232 or infrared transceiver modules. When the UART is in NRZ mode, normal data is output with "marks" transmitted as logic high and "spaces" transmitted as logic low. In IrDA mode, this pin, a configurable narrow pulse, is output for each zero bit that is transmitted.
- **$\overline{\text{CTS1}}$/$\overline{\text{CTS2}}$**—The clear-to-send signal multiplexed with PE7 (PJ7 in UART 2), is an active low input used for transmitter flow control. The transmitter waits until this signal is asserted (low) before it starts transmitting a character. If this signal is negated when a character is being transmitted, the character will be completed, but no additional characters are transmitted until this signal is asserted again. The current value of this pin can be read in the CTSx STAT bit of the corresponding UART transmitter (UTX*x*) register.

  NOTE:

  If the NOCTS*x* bit of the UTX*x* register is set, the transmitter sends a character whenever a character is ready to be transmitted. The $\overline{\text{CTS}x}$ pin can be programmed to post an interrupt on rising and falling edges if the CTSD bit is set in the corresponding UART control (USTCNT) register.

- **RXD1/RXD2**—The receive data signal multiplexed with PE4 (PJ4 in UART 2), is the receiver serial input. As for the TXD*x* pin, when the UART is in NRZ mode, standard NRZ data is expected. In IrDA mode, a pulse of at least 1.63 µs is expected for each zero bit received. The required pulse polarity is controlled by the RXPOL bit of the corresponding UART miscellaneous (UMISC) register. This pin interfaces to standard RS-232 and infrared transceiver modules.
- **$\overline{\text{RTS1}}$/$\overline{\text{RTS2}}$**—The request-to-send signal multiplexed with PE6 (PJ6 in UART 2) serves two purposes. Normally, this signal is used for flow control, in which the receiver indicates that it is ready to receive data by asserting this pin (low). This pin is then connected to the far-end transmitter's $\overline{\text{CTS}}$ pin. When the receiver FIFO is nearly full (four slots are remaining), which

indicates a pending FIFO overrun, this pin is negated (high). When not being used for flow control, this pin can be used as a general-purpose output controlled by the RTS1 bit (RTS2 bit in UART 2) of the corresponding UMISC register.

- **UCLK**—The UART Clock input/output signal serves two purposes. It can function as the source of the clock to the baud rate generator, or it can output the bit clock at the selected baud rate for synchronous operation. The external UCLK pin connects to the UCLK of both UART 1 and UART 2. For UCLK output, only one UART at a time is selected to drive this signal. Please refer to Section 6.2.2, “Peripheral Control Register,” on page 6-3 for more details.

19.3 UART Operation

Both UART modules consist of three sub-blocks:

- Transmitter
- Receiver
- Baud rate generator

Section 19.3.1, “Transmitter Operation,” through Section 19.3.3, “Baud Rate Generator Operation,” discuss these sub-blocks in detail.

19.3.1 Transmitter Operation

The transmitter accepts a character (byte) from the CPU bus and transmits it serially. When the FIFO is empty, the transmitter outputs a continuous idle (which is 1 bit in NRZ mode and selectable polarity in IrDA mode). When a character is available for transmission, the start, stop, and parity (if enabled) bits are added to the character, and it is serially shifted (LSB first) at the selected bit rate. The transmitter presents a new bit on each falling edge of the bit clock.

19.3.1.1 Tx FIFO Buffer Operation

The transmitter posts a maskable interrupt when it needs parallel data (TX AVAIL). There are three maskable interrupts. To take maximum advantage of the 32-byte FIFO, the FIFO EMPTY interrupt should be enabled. The interrupt service routine should load data until the TX AVAIL bit in the UTX*x* register is clear or until there is no more data to transmit. The transmitter does not generate another interrupt until the FIFO has completely emptied.

If the driver software has excessive interrupt service latency time, use the FIFO HALF interrupt. With UART 1 and 2, the transmitter generates an interrupt when the FIFO has a number of empty slots that is less than or equal to the number specified by the TxFIFO level marker of the FIFO level marker interrupt register.

If the FIFO buffer is not needed, only the TX AVAIL interrupt is required. This interrupt is generated when at least one space is available in the FIFO. Any data that is written to the FIFO when the TX AVAIL bit is clear is ignored.

19.3.1.2 $\overline{\text{CTS}}$ Signal Operation

$\overline{\text{CTS}}x$ is used for hardware flow control. If $\overline{\text{CTS}}x$ is negated (high), the transmitter finishes sending the character in progress (if any) and then waits for $\overline{\text{CTS}}x$ to become asserted (low) again before starting the next character. The current state of the $\overline{\text{CTS}}x$ pin is sampled by the bit clock and can be monitored by reading the CTS*x* STAT bit of the UTX*x* register. An interrupt can be generated when the $\overline{\text{CTS}}x$ pin changes state. The CTS*x* DELTA bit of the UTX*x* register goes high when the $\overline{\text{CTS}}x$ pin toggles. For

applications that do not need hardware flow control, such as IrDA, the NOCTS*x* bit of the UTX*x* register should be set. When this bit is set, characters will be sent as soon as they are available in the FIFO. Parity errors can be generated for debugging purposes by setting the FORCE PERR bit in the corresponding UMISC register.

The SEND BREAK bit of the corresponding UTX*x* register is used to generate a Break character (continuous zeros). Use the following procedure to send the minimum number of valid Break characters.

1. Make sure the BUSY bit in the UTX*x* register is set.
2. Wait until the BUSY bit goes low.
3. Clear the TXEN bit in the USTCNT register, to flush the FIFO.
4. Wait until the BUSY bit goes low.
5. Set the TXEN bit.
6. Set the SEND BREAK bit in the UTX*x* register.
7. Load a dummy character into the FIFO.
8. Wait until the BUSY bit goes low.
9. Clear the SEND BREAK bit.

After the procedure finishes, the FIFO is empty and the transmitter is idle and waiting for the next character.

If the TXEN bit of the USTCNT*x* register is negated when a character is being transmitted, the character is completed before the transmitter returns to IDLE. The transmit FIFO is immediately flushed when the TXEN bit is cleared. To disable the UART after the message has been completely sent, monitor the BUSY bit to determine when the transmitter completes sending the final character. Remember that there may be a long time delay, depending on the baud rate. It is safe to clear the UEN bit of the corresponding USTCNT register after the BUSY bit becomes clear. The BUSY bit can also be used to determine when to disable the transmitter and turn the link around to receive IrDA applications.

When IrDA mode is enabled, the transmitter produces a pulse that is less than or equal to three-sixteenths of bit time for each zero bit sent. Ones are sent as “no pulse.” When the TXPOL bit of the UMISC register is low, pulses are active high. When the TXPOL bit is high, pulses are active low and idle is high.

19.3.2 Receiver Operation

The receiver block of the UART accepts a serial data stream and converts it into parallel characters. The receiver operates in two modes—asynchronous and synchronous. In asynchronous mode, it searches for a start bit, qualifies it, and then samples the succeeding data bits at the perceived bit center. Jitter tolerance and noise immunity are provided by sampling 16 times per bit and using a voting circuit to enhance sampling. IrDA operation must use asynchronous mode. In synchronous mode, RXD*x* is sampled on each rising edge of the bit clock, which is generated by the UART module or supplied externally. When a start bit is identified, the remaining bits are shifted in and loaded into the FIFO.

If parity is enabled, the parity bit is checked and its status is reported in the URX*x* register. Similarly, frame errors, breaks, and overruns are checked and reported. The 4 character status bits in the high byte (bits 11–8) of the URX*x* register are valid only when read as a 16-bit word with the received character byte.

19.3.2.1 Rx FIFO Buffer Operation

As with the transmitter, the receiver FIFO is configurable. If the software being used has a short interrupt latency time, the FIFO FULL interrupt in the URX register can be enabled to indicate that the FIFO has no remaining space available when this interrupt is generated. If the DATA READY bit in the URX register indicates that more data is remaining in the FIFO, the FIFO is emptied byte by byte.

If the software has a longer latency time, the FIFO HALF interrupt of the URX register can be used. This interrupt is generated when no more than *n* empty bytes remain in the FIFO where *n* is the number specified by the RxFIFO level marker of the FIFO level marker interrupt register.

If the receiver FIFO is not needed, the DATA READY interrupt should be used. This interrupt is generated when one or more characters are present in the FIFO. The OLD DATA bit in the URX register indicates that there is data in the FIFO and that the receive line has been idle for more than 30 bit times. This is useful in determining the end of a block of characters.

When IrDA mode is enabled, the receiver expects narrow (1.63 µs minimum) pulses for each zero bit received. Otherwise, normal NRZ is expected. An infrared transceiver directly connected to the RXD*x* pin transforms the infrared signal into an electrical signal. Polarity is programmable so that RXD*x* can be connected directly to an external IrDA transceiver.

19.3.3 Baud Rate Generator Operation

The baud generator provides the bit clocks to the transmitter and receiver blocks. It consists of two prescalers, an integer prescaler, and a second non-integer prescaler, as well as a 2^n divider. Figure 19-4 illustrates a block diagram of the baud rate generator.

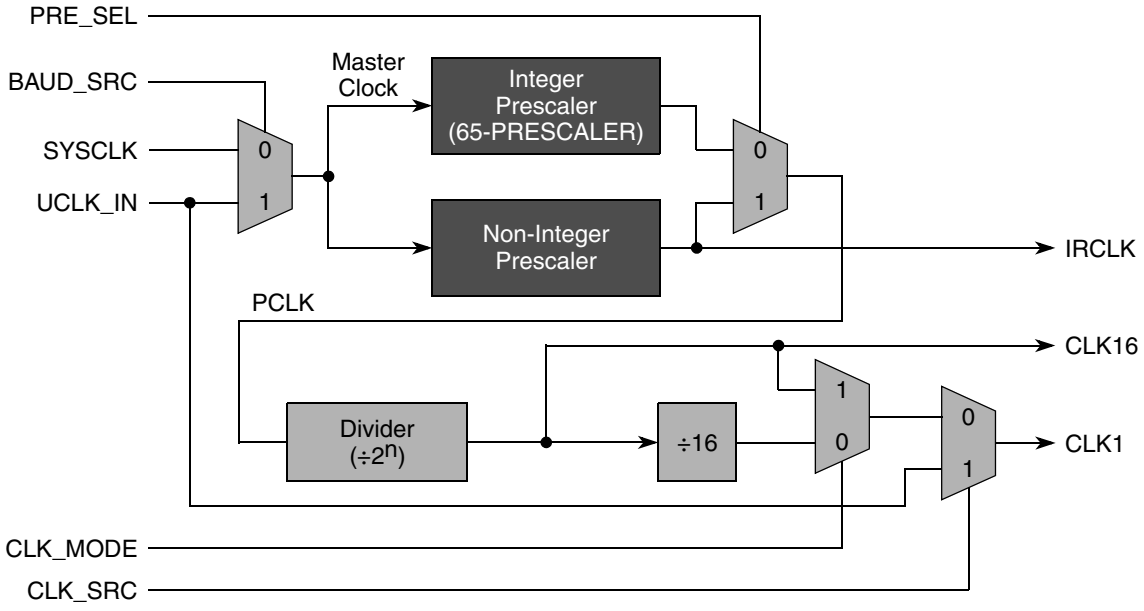


Figure 19-4. Baud Rate Generator Block Diagram

The baud rate generator's master clock source can be the system clock (SYSCLK), or it can be provided by the UCLK pin (input mode). By setting the BAUD SRC bit of the corresponding UART baud control (UBAUD) register to 1, an external clock can directly drive the baud rate generator. For synchronous applications, the UCLK signal can be configured as an input or output for the 1x bit clock.

19.3.3.1 Divider

The divider is a 2^n binary divider with eight taps—1, 2, 4, 8, 16, 32, 64, and 128. The selected tap is the 16x clock (CLK16) for the receiver. This clock is further divided by 16 to provide a 50-percent duty-cycle 1x clock (CLK1) to the transmitter. When the CLKM bit of the USTCNT register is high, CLK1 is directly sourced by the CLK16 signal.

19.3.3.2 Non-Integer Prescaler

The non-integer prescaler is used to generate special, nonstandard baud frequencies.

Table 19-1 contains the values to use for the non-integer prescaler operation.

Table 19-1. Non-Integer Prescaler Values

Select (Binary)	Minimum Divisor	Maximum Divisor	Step Size
000	2	3 127/128	1/128
001	4	7 63/64	1/64
010	8	15 31/32	1/32
011	16	31 15/16	1/16
100	32	63 7/8	1/8
101	64	127 3/4	1/4
110	128	255 1/2	1/2
111	1	1	—

19.3.3.2.1 Non-Integer Prescaler Programming Example

To generate a 3.072 MHz clock frequency from a 16.580608 MHz clock source, use the following steps:

1. Calculate the divisor:
   divisor = 16.580608 MHz ÷ 3.072000 MHz = 5.397333
2. Find the value for the SELECT field in the non-integer prescaler register (NIPR*x*). Because the value of the divisor calculated in the previous step is between 4 and 8, Table 19-1 indicates that the SELECT field is 001. The divisor step size for the selected range is 1/64.
3. Determine the number of steps to program into the STEP VALUE field by subtracting the minimum divisor from the divisor (5.397333 - 4 = 1.397333) and dividing this value by the step size, which is one sixty-fourth or 0.015625 (1.397333 ÷ 0.015625 = 89.43). The result should be rounded to the nearest integer value and converted to the hex equivalent:

   89 (decimal) = 59 (hex)

   The actual divisor will be 5.390625, which will produce a frequency of 3.075823 MHz (0.12 percent above the preferred frequency).

19.3.3.2.2 IrDA Mode Non-Integer Prescaler Programming Example

When IrDA mode is enabled, zeros are transmitted as 3/16 (or less) bit-time pulses.

NOTE:

In IrDA mode, the baud rate is determined by the integer prescaler. The non-integer prescaler is then used for controlling the pulse width, but it must be less than or equal to 3/16 of bit time.

For example, a 115.200 kHz IrDA transfer rate is desired. The non-integer prescaler must provide a clock at 1.843200 MHz (115.200 kHz × 16). This clock is used to generate transmit pulses, which will be 3/16 of a 115.200 kHz bit time. The following example provides a sample divisor calculation assuming a 33.16 MHz SYSCLK.

Sample Divisor Calculation:

Using a 33.16 MHz SYSCLK, a 1.8432 MHz IRCLK (for IrDA bit time) is desired.

Non-integer prescaler required = 33.16 MHz/1.8432 MHz = 18.0

18.0 = 16 + ($20 × 1/16)

Where:

16 = minimum divisor

$20 = step value

1/16 = step size

Table 19-2 contains the values inserted in the non-integer prescaler register (NIPR) for IrDA operation in the example. Section 19.5.6, “UART 1 Non-Integer Prescaler Register,” on page 19-19 and Section 19.5.13, “UART 2 Non-Integer Prescaler Register,” on page 19-29 describes all settings for NIPR.

Table 19-2. Non-Integer Prescaler Settings for Example

Mode	Select (Binary)	Step Value (Hex)
IrDA	011	0x20

19.3.3.3 Integer Prescaler

The baud rate generator can provide standard baud rates from many system clock frequencies. Table 19-3 contains the values that should be used in the UBAUD register for a 33.16 MHz system clock frequency.

Table 19-3. Selected Baud Rate Settings

Baud Rate	UBAUD*x* DIVIDE Setting (Decimal)	UBAUD*x* PRESCALER Setting (Decimal)
230400	0	56
115200	1	56
57600	2	56
28800	3	56
14400	4	56
38400	1	38
19200	2	38

Table 19-3. Selected Baud Rate Settings (Continued)

Baud Rate	UBAUD*x* DIVIDE Setting (Decimal)	UBAUD*x* PRESCALER Setting (Decimal)
9600	3	38
4800	4	38
2400	5	38
1200	6	38
600	7	38

19.4 UART Direct Memory Access Operation

The UART 1 and 2 modules support DMA burst reads from memory to TXFIFO 1and 2 and DMA burst writes to memory from RXFIFO 1 and 2 using the MC68SZ328 DMA controller.

19.4.1 DMA Configuration Options

The DMA controller general registers must be initialized before any I/O peripherals may be used. In additon to this, I/O channels must be selected to work solely with the UART by the system programmer. Table 19-4, “UART DMA Configuration Options,” on page 19-10 summarizes important data for configuring the DMA general and DMA I/O registers to be used with UART 1 and 2.

Table 19-4. UART DMA Configuration Options

	UART 1 TX	UART 1 RX	UART 2 TX	UART 2 RX
FIFO size	8 bit	8 bit	8 bit	8 bit
Memory size	8, 16 bit **Note:** When the flyby function is used, memory size must be set to 8-bit, the same as the FIFO size.	8, 16 bit **Note:** When the flyby function is used, memory size must be set to 8-bit, the same as the FIFO size.	8, 16 bit **Note:** When the flyby function is used, memory size must be set to 8-bit, the same as the FIFO size.	8, 16 bit **Note:** When the flyby function is used, memory size must be set to 8-bit, the same as the FIFO size.
DMA burst length setting for DMA flag use	* byte – TFH 1bit 32 byte – TFE 1 bit	* byte – RFH 1bit 32 byte – RFF 1 bit	* byte – TFH 2bit 32 byte – TFE 2 bit	* byte – TFH 2bit 32 byte – TFE 2 bit
DMA source select setting	30 $\overline{\text{DMA\_REQ}}$[30]	31 $\overline{\text{DMA\_REQ}}$[31]	28 $\overline{\text{DMA\_REQ}}$[28]	29 $\overline{\text{DMA\_REQ}}$[29]
DMA channels available for use	Channel 2–5 (memory to I/O channel)	Channel 2–5 (I/O to memory channel)	Channel 2–5 (memory to I/O channel)	Channel 2–5 (I/O to memory channel)
Memory address	User specified	User specified	User specified	User specified
Peripheral address	0xFFFFF907	0xFFFFF905	0xFFFFF917	0xFFFFF915
Byte count	User specified	User specified	User specified	User specified
Request time out	Supported	Supported	Supported	Supported
DMA interrupt	Supported	Supported	Supported	Supported

Note: * the DMA burst length setting in the DMA controller is equal to the TXFIFO and RXFIFO level marker setting.

19.4.2 DMA Configuration Procedure

To configure the UART 1 and UART 2 modules for DMA operation, use the following procedure:

1. Initialize the UART.
2. Initialize the DMA controller general registers.
3. Initialize the DMA controller I/O registers for UART DMA.
4. Enable the DMA I/O channel.
5. Enable the DMA flag in the HMARK1 and HMARK2 registers.
6. Poll DMA ISR for end of burst status.

19.5 Programming Model

Section 19.5.1, “UART 1 Status/Control Register,” through Section 19.5.14, “UART2 FIFO Level Marker Interrupt Register,” describe the UART registers and detailed information about their settings. The UART 1 registers are described first. Section 19.6, “UART Register Set,” on page 19-31, provides a summary of the entire set of registers for both UART 1 and 2.

19.5.1 UART 1 Status/Control Register

The UART 1 status/control register (USTCNT1) controls the overall operation of the UART 1 module. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 19-5.

USTCNT1 UART 1 Status/Control Register **0x(FF)FFF900**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	UEN	RXEN	TXEN	CLKM	PEN	ODD	STOP	8/7	ODEN	CTSD	RXFE	RXHE	RXRE	TXEE	TXHE	TXAE
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 19-5. UART 1 Status/Control Register Description

Name	Description	Setting
UEN Bit 15	**UART 1 Enable**—This bit enables the UART 1 module. This bit resets to 0. **Note:** When the UART 1 module is first enabled after a hard reset and before the interrupts are enabled, set the UEN and RXEN bits and perform a word read operation on the URX register to initialize the FIFO and character status bits.	0 = UART 1 module is disabled 1 = UART 1 module is enabled
RXEN Bit 14	**Receiver Enable**—This bit enables the receiver block. This bit resets to 0.	0 = Receiver is disabled and the receive FIFO is flushed 1 = Receiver is enabled
TXEN Bit 13	**Transmitter Enable**—This bit enables the transmitter block. This bit resets to 0.	0 = Transmitter is disabled and the transmit FIFO is flushed 1 = Transmitter is enabled
CLKM Bit 12	**Clock Mode Selection**—This bit selects the receiver's operating mode. When this bit is low, the receiver is in 16x mode, in which it synchronizes to the incoming datastream and samples at the perceived center of each bit period. When this bit is high, the receiver is in 1x mode, in which it samples the datastream on each rising edge of the bit clock. In 1x mode, the bit clock is driven by CLK16. This bit resets to 0.	0 = 16x clock mode (asynchronous mode) 1 = 1x clock mode (synchronous mode)
PEN Bit 11	**Parity Enable**—This bit controls the parity generator in the transmitter and the parity checker in the receiver.	0 = Parity is disabled 1 = Parity is enabled
ODD Bit 10	**Odd Parity**—This bit controls the sense of the parity generator and checker. This bit has no function if the PEN bit is low.	0 = Even parity 1 = Odd parity
STOP Bit 9	**Stop Bit Transmission**—This bit controls the number of stop bits transmitted after a character. This bit has no effect on the receiver, which expects one or more stop bits.	0 = One stop bit is transmitted 1 = Two stop bits are transmitted
8/7 Bit 8	**8- or 7-Bit**—This bit controls the character length. When this bit is set to 7-bit operation, the transmitter ignores data bit 7 and, when receiving, the receiver forces data bit 7 to 0.	0 = 7-bit transmit-and-receive character length 1 = 8-bit transmit-and-receive character length

Table 19-5. UART 1 Status/Control Register Description (Continued)

Name	Description	Setting
ODEN Bit 7	**Old Data Enable**—This bit enables an interrupt when the OLD DATA bit in the URX register is set.	0 = OLD DATA interrupt is disabled 1 = OLD DATA interrupt is enabled
CTSD Bit 6	**$\overline{\text{CTS1}}$ Delta Enable**—When this bit is high, it enables an interrupt when the $\overline{\text{CTS1}}$ pin changes state. When it is low, this interrupt is disabled. The current status of the $\overline{\text{CTS1}}$ pin is read in the UTX register.	0 = $\overline{\text{CTS1}}$ interrupt is disabled 1 = $\overline{\text{CTS1}}$ interrupt is enabled
RXFE Bit 5	**Receiver Full Enable**—When this bit is high, it enables an interrupt when the receiver FIFO is full. This bit resets to 0.	0 = RX FULL interrupt is disabled 1 = RX FULL interrupt is enabled
RXHE Bit 4	**Receiver Half Enable**—When this bit is high, it enables an interrupt when the receiver FIFO is more than half full. This bit resets to 0.	0 = RX HALF interrupt is disabled 1 = RX HALF interrupt is enabled
RXRE Bit 3	**Receiver Ready Enable**—When this bit is high, it enables an interrupt when the receiver has at least 1 data byte in the FIFO. When it is low, this interrupt is disabled.	0 = RX interrupt is disabled 1 = RX interrupt is enabled
TXEE Bit 2	**Transmitter Empty Enable**—When this bit is high, it enables an interrupt when the transmitter FIFO is empty and needs data. When it is low, this interrupt is disabled.	0 = TX EMPTY interrupt is disabled 1 = TX EMPTY interrupt is enabled
TXHE Bit 1	**Transmitter Half Empty Enable**—When this bit is high, it enables an interrupt when the transmit FIFO is less than half full. When it is low, the TX HALF interrupt is disabled. This bit resets to 0.	0 = TX HALF interrupt is disabled 1 = TX HALF interrupt is enabled
TXAE Bit 0	**Transmitter Available for New Data**—When this bit is high, it enables an interrupt if the transmitter has a slot available in the FIFO. When it is low, this interrupt is disabled. This bit resets to 0.	0 = TX AVAIL interrupt is disabled 1 = TX AVAIL interrupt is enabled

19.5.2 UART 1 Baud Control Register

The UART 1 baud control (UBAUD1) register controls the operation of the baud rate generator, the integer prescaler, and the UCLK signal. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 19-6.

UBAUD1 UART 1 Baud Control Register **0x(FF)FFF902**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
			UCLK DIR		BAUD SRC	DIVIDE					PRESCALER					
TYPE			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	1	0
	0x0002															

Table 19-6. UART 1 Baud Control Register Description

Name	Description	Setting
Reserved Bits 15–14	Reserved	These bits are reserved and should be set to 0.
UCLKDIR Bit 13	**UCLK Direction**—This bit controls the direction of the UCLK signal. When this bit is low, the signal is an input, and when it is high, it is an output. However, the SEL*x* bit in the Port E registers must be 0. See Section 16.4.4, “Port E Registers,” on page 16-16 for more information.	0 = UCLK is an input. 1 = UCLK is an output.
Reserved Bit 12	Reserved	This bit is reserved and should be set to 0.
BAUD SRC Bit 11	**Baud Source**—This bit controls the clock source to the baud rate generator.	0 = Baud rate generator source is from system clock. 1 = Baud rate generator source is from UCLK pin (UCLKDIR must be set to 0).
DIVIDE Bits 10–8	**Divide**—This field controls the clock frequency produced by the baud rate generator.	000 = Divide by 1. 001 = Divide by 2. 010 = Divide by 4. 011 = Divide by 8. 100 = Divide by 16. 101 = Divide by 32. 110 = Divide by 64. 111 = Divide by 128.
Reserved Bits 7–6	Reserved	These bits are reserved and should be set to 0.
PRESCALER Bits 5–0	**Prescaler**—This field controls the division value of the baud generator prescaler. The division value is determined by the following formula: Prescaler division value = 65 (decimal) – PRESCALER	0x00: (not supported) 0x01: Integer prescaler = 65-(1) = 64 0x02: Integer prescaler = 65-(2) = 63 0x03: Integer prescaler = 65-(3) = 62 : 0x3F: Integer prescaler = 65-(63) = 2

19.5.3 UART 1 Receiver Register

The UART 1 receiver (URX1) register indicates the status of the receiver FIFO and character data. The FIFO status bits reflect the current status of the FIFO. At initial power up, these bits contain random data. Before enabling the receiver interrupts, the UEN and RXEN bits in the USTCNT register should be set. Reading the UART 1 receiver register initializes the FIFO status bits. The receiver interrupts can then be enabled. However, the character status bits are only valid when read with the character bits in a 16-bit read access. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 19-7.

URX1 UART 1 Receiver Register **0x(FF)FFF904**

BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
FIFO FULL	FIFO HALF	DATA READY	OLD DATA	OVRUN	FRAME ERROR	BREAK	PARITY ERROR	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	0	0	0	0	0	0	0	0
	0x0000															

Table 19-7. UART 1 Receiver Register Description

Name	Description	Setting
FIFO FULL Bit 15	**FIFO Full (FIFO Status)**—This read-only bit indicates that the receiver FIFO is full and may generate an overrun. This bit generates a maskable interrupt. **Note:** FIFO full status flag will assert—set to 1—after receiving 31 bytes. This is due in part because the buffer is a circular link list structure.	0 = Receiver FIFO is not full 1 = Receiver FIFO is full
FIFO HALF Bit 14	**FIFO Half (FIFO Status)**—This read-only bit indicates that the receiver FIFO has four or fewer slots remaining in the FIFO. This bit generates a maskable interrupt.	0 = Receiver FIFO has more than four slots remaining 1 = Receiver FIFO has four or fewer slots remaining
DATA READY Bit 13	**Data Ready (FIFO Status)**—This read-only bit indicates that at least 1 byte is present in the receive FIFO. The character bits are valid only when this bit is set. This bit generates a maskable interrupt.	0 = No data in the receiver FIFO 1 = Data in the receiver FIFO
OLD DATA Bit 12	**Old Data (FIFO Status)**—This read-only bit indicates that data in the FIFO is older than 30 bit times. It is useful in situations where the FIFO FULL or FIFO HALF interrupts are used. If there is data in the FIFO, but the amount is below the FIFO HALF interrupt threshold, a maskable interrupt can be generated to alert the software that unread data is present. This bit clears when the character bits are read.	0 = FIFO is empty or the data in the FIFO is < 30 bit times old 1 = Data in the FIFO is > 30 bit times old
OVRUN Bit 11	**FIFO Overrun (Character Status)**—This read-only bit indicates, when HIGH, that the receiver ignored data to prevent overwriting the data in the FIFO. It indicates that the user's software is not keeping up with the incoming data rate. This bit is set for the last character written to the FIFO indicating that all characters following this character will be ignored if a read is not performed by software. This bit is updated and valid for each received character. Under normal circumstances, this bit should never be set, unless the overrun condition occurs—that is, it should be set on the 32nd character in the FIFO.	0 = No FIFO overrun occurred 1 = A FIFO overrun was detected
FRAME ERROR Bit 10	**Frame Error (Character Status)**—This read-only bit indicates that the current character had a framing error (missing stop bit), which indicates that there may be corrupted data. This bit is updated for each character read from the FIFO.	0 = Character has no framing error 1 = Character has a framing error

Table 19-7. UART 1 Receiver Register Description (Continued)

Name	Description	Setting
BREAK Bit 9	**Break (Character Status)**—This read-only bit indicates that the current character was detected as a BREAK. The data bits are all 0 and the stop bit is also 0. The FRAME ERROR bit will always be set when this bit is set, and if odd parity is selected, PARITY ERROR will also be set. This bit is updated and valid with each character read from the FIFO.	0 = Character is not a break character 1 = Character is a break character
PARITY ERROR Bit 8	**Parity Error (Character Status)**—This read-only bit indicates that the current character was detected with a parity error, which indicates that there may be corrupted data. This bit is updated and valid with each character read from the FIFO. When parity is disabled, this bit always reads 0.	See description
RX DATA Bits 7–0	**Rx Data (Character Data)**—This read-only field is the top receive character in the FIFO. The bits have no meaning if the DATA READY bit is 0. In 7-bit mode, the most significant bit is forced to 0, and in 8-bit mode, all bits are active.	See description

19.5.4 UART 1 Transmitter Register

The UART 1 transmitter (UTX1) register controls how the transmitter operates. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 19-8.

UTX1 UART 1 Transmitter Register **0x(FF)FFF906**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	FIFO EMPTY	FIFO HALF	TX AVAIL	SEND BREAK	NO CTS1	BUSY	CTS1 STAT	CTS1 DELTA	TX DATA							
TYPE	r	r	r	rw	rw	r	r	rw	w	w	w	w	w	w	w	w
RESET	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
						0xE800										

Table 19-8. UART 1 Transmitter Register Description

Name	Description	Setting
FIFO EMPTY Bit 15	**FIFO Empty (FIFO Status)**—This read-only bit indicates that the transmitter FIFO is empty. This bit generates a maskable interrupt. **Note:** FIFO empty status flag will assert—set to 1—when there are 31free bytes (slots) in the FIFO. This is due in part because the buffer is a circular link list structure.	0 = Transmitter FIFO is not empty 1 = Transmitter FIFO is empty
FIFO HALF Bit 14	**FIFO Half (FIFO Status)**—This read-only bit indicates that the transmitter FIFO is less than half full. This bit generates a maskable interrupt.	0 = Transmitter FIFO is more than half full 1 = Transmitter FIFO is less than half full
TX AVAIL Bit 13	**Transmit FIFO Available (FIFO Status)**—This read-only bit indicates that the transmitter FIFO has at least one slot available for data. This bit generates a maskable interrupt.	0 = Transmitter does not need data 1 = Transmitter needs data
SEND BREAK Bit 12	**Send Break (Tx Control)**—This bit forces the transmitter to immediately send continuous zeros, which creates a break character. See Section 19.3.1.2, “CTS Signal Operation,” for a description of how to generate a break.	0 = Normal transmission 1 = Send break (continuous zeros)
NOCTS1 Bit 11	**Ignore $\overline{\text{CTS1}}$ (Tx Control)**—When this bit is high, it forces the $\overline{\text{CTS1}}$ signal that is presented to the transmitter to always be asserted, which effectively ignores the external pin.	0 = Transmit only when the $\overline{\text{CTS1}}$ signal is asserted 1 = Ignore the $\overline{\text{CTS1}}$ signal
BUSY Bit 10	**Busy (Tx Status) (Read-Only)**—When this bit is high, it indicates that the transmitter is busy sending a character. This bit is asserted when the transmitter state machine is not idle or the FIFO has data in it.	0 = Transmitter is not sending a character 1 = Transmitter is sending a character
CTS1 STAT Bit 9	**$\overline{\text{CTS1}}$ Status ($\overline{\text{CTS1}}$ Bit) (Read-Only)**—This bit indicates the current status of the $\overline{\text{CTS1}}$ signal. A “snapshot” of the pin is taken immediately before this bit is presented to the data bus. When the NOCTS1 bit is high, this bit can function as a general-purpose input.	0 = $\overline{\text{CTS1}}$ signal is low 1 = $\overline{\text{CTS1}}$ signal is high

Table 19-8. UART 1 Transmitter Register Description (Continued)

Name	Description	Setting
CTS1 DELTA Bit 8	**$\overline{\text{CTS1}}$ Delta ($\overline{\text{CTS1}}$ Bit)**—When this bit is high, it indicates that the $\overline{\text{CTS1}}$ signal changed state and generates a maskable interrupt. The current state of the $\overline{\text{CTS1}}$ signal is available on the CTS1 STAT bit. An immediate interrupt may be generated by setting this bit high. The $\overline{\text{CTS1}}$ interrupt is cleared by writing 0 to this bit.	0 = $\overline{\text{CTS1}}$ signal did not change state since it was last cleared 1 = $\overline{\text{CTS1}}$ signal has changed state
TX DATA Bits 7–0	**Tx Data (Character) (Write-Only)**—This write-only field is the parallel transmit-data input. In 7-bit mode, bit 7 is ignored, and in 8-bit mode, all of the bits are used. Data is transmitted with the least significant bit first. A new character is transmitted when this field is written and has passed through the FIFO.	See description

19.5.5 UART 1 Miscellaneous Register

The UART 1 miscellaneous (UMISC1) register contains miscellaneous bits to control test features of the UART 1 module. Some bits, however, are only used for factory testing and should not be used. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 19-9.

UMISC1 UART 1 Miscellaneous Register **0x(FF)FFF908**

BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
BAUD TEST	CLKSRC	FORCE PERR	LOOP	BAUD RESET	IRTEST		RTS1 TRIG	RTS1 CONT	RTS1	IRDAEN	IRDA LOOP	RXPOL	TXPOL	RXES	RXEE
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	0
0x0000															

Table 19-9. UART 1 Miscellaneous Register Description

Name	Description	Setting
BAUD TEST Bit 15	**Baud Rate Generator Testing**—This bit puts the baud rate generator in test mode. The integer and non-integer prescalers, as well as the divider, are broken into 4-bit nibbles for testing. This bit should remain 0 for normal operation.	0 = Normal mode. 1 = Test mode.
CLKSRC Bit 14	**Clock Source**—This bit selects the source of the 1x bit clock for transmission and reception. When this bit is high, the bit clock is derived directly from the UCLK pin (it must be configured as an input). When it is low (normal), the bit clock is supplied by the baud rate generator. This bit allows high-speed synchronous applications, in which a clock is provided by the external system.	0 = Bit clock is generated by the baud rate generator. 1 = Bit clock is supplied by the UCLK pin.

Table 19-9. UART 1 Miscellaneous Register Description (Continued)

Name	Description	Setting
FORCE PERR Bit 13	**Force Parity Error**—When this bit is high, it forces the transmitter to generate parity errors, if parity is enabled. This bit is for system debugging.	0 = Generate normal parity. 1 = Generate inverted parity (error).
LOOP Bit 12	**Loopback**—This bit controls loopback for system testing purposes. When this bit is high, the receiver input is internally connected to the transmitter and ignores the RXD1 pin. The TXD1 pin is unaffected by this bit.	0 = Normal receiver operation. 1 = Internally connects the transmitter output to the receiver input.
BAUD RESET Bit 11	**Baud Rate Generator Reset**—This bit resets the baud rate generator counters.	0 = Normal operation. 1 = Reset baud counters.
IRTEST Bit 10	**Infrared Testing**—This bit connects the output of the IrDA circuitry to the TXD1 pin. This provides test visibility to the IrDA module. This bit is for manufacturer testing only and should be set to 0 in the normal application.	0 = Normal operation. 1 = IrDA test mode.
Reserved Bit 9	Reserved	This bit is reserved and should be set to 0.
RTS1 TRIG Bit 8	**$\overline{\text{RTS1}}$ Trigger Source Select**—This bit selects the control of the $\overline{\text{RTS}}$ pin when bit 7 RTS1 CONT =1.	0 = $\overline{\text{RTS1}}$ pin is asserted when the number of filled slots is equal to RxFIFO level marker. 1 = $\overline{\text{RTS1}}$ pin is asserted when only one slot is available in the receiver FIFO.
RTS1 CONT Bit 7	**$\overline{\text{RTS1}}$ Control**—This bit selects the function of the $\overline{\text{RTS1}}$ pin.	0 = $\overline{\text{RTS1}}$ pin is controlled by the RTS1 bit. 1 = $\overline{\text{RTS1}}$ pin is controlled by the receiver FIFO and RTS1 TRIG bit-.
RTS1 Bit 6	**Request to Send Pin**—This bit controls the $\overline{\text{RTS1}}$ pin when the RTS1 CONT bit is 0.	0 = $\overline{\text{RTS1}}$ pin is 1. 1 = $\overline{\text{RTS1}}$ pin is 0.
IRDAEN Bit 5	**Infrared Enable**—This bit enables the IrDA interface.	0 = Normal NRZ operation. 1 = IrDA operation.
IRDA LOOP Bit 4	**Loop Infrared**—This bit controls the loopback from the transmitter to the receiver in the IrDA interface. This bit is used for system testing purposes.	0 = No infrared loop. 1 = Connect the infrared transmitter to an infrared receiver.
RXPOL Bit 3	**Receive Polarity**—This bit controls the polarity of the received data.	0 = Normal polarity (1 = idle). 1 = Inverted polarity (0 = idle).
TXPOL Bit 2	**Transmit Polarity**—This bit controls the polarity of the transmitted data.	0 = Normal polarity (1 = idle). 1 = Inverted polarity (0 = idle).
RXES Bit 1	**Receive Error Interrupt Status**— This bit, when set, indicates that one of the error status bits in receiver register has been set once before. The interrupt must be cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.	0 = No receive error interrupt occurred. 1 = Receive error interrupt occurred.

Table 19-9. UART 1 Miscellaneous Register Description (Continued)

Name	Description	Setting
RXEE Bit 0	**Receive Error Interrupt Enable**— When this bit is high, it enables an interrupt if any of the error status bits in the receiver register are set during the DMA transfer. This bit is designed for DMA operation. When there is a receive error (for example, overrun, parity, or frame errors) during DMA burst transfer, interrupt is generated. However, the user's program cannot read the receiver register to locate the exact error source after the burst because the receive error status bits change on each received character basis.	0 = Receive error interrupt is disabled. 1 = Receive error interrupt is enabled.

19.5.6 UART 1 Non-Integer Prescaler Register

The UART 1 non-integer prescaler register (NIPR1) contains the control bits for the non-integer prescaler. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 19-10.

NIPR1 UART 1 Non-Integer Prescaler Register **0x(FF)FFF90A**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	PRE SEL					SELECT			STEP VALUE							
TYPE	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 19-10. UART 1 Non-Integer Prescaler Register Description

Name	Description	Setting
PRESEL Bit 15	**Prescaler Selection**—This bit selects the input to the baud rate generator divider. Refer to Figure 19-4 on page 19-6 for information about selecting the prescaler.	0 = Divider source is from the integer prescaler. 1 = Divider source is from the non-integer prescaler.
Reserved Bits 14–11	Reserved	These bits are reserved and should be set to 0.
SELECT Bits 10–8	**Tap Selection**—This field selects a tap from the non-integer divider.	000 = Divide range is 2 to 3 127/128 in 1/128 steps. 001 = Divide range is 4 to 7 63/64 in 1/64 steps. 010 = Divide range is 8 to 15 31/32 in 1/32 steps. 011 = Divide range is 16 to 31 15/16 in 1/16 steps. 100 = Divide range is 32 to 63 7/8 in 1/8 steps. 101 = Divide range is 64 to 127 3/4 in 1/4 steps. 110 = Divide range is 128 to 255 1/2 in 1/2 steps. 111 = Divide value is 1.

Table 19-10. UART 1 Non-Integer Prescaler Register Description (Continued)

Name	Description	Setting
STEP VALUE Bits 7–0	**Step Value**—This field selects the non-integer prescaler's step value.	0000 0000. Step = 0. 0000 0001. Step = 1. . . . 1111 1110. Step = 254. 1111 1111. Step = 255.

19.5.7 UART 1 FIFO Level Marker Interrupt Register

The UART 1 FIFO level marker register configures the level at which either the RxFIFO or the TxFIFO of UART1 reports a half-full condition. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 19-19.

HMARK1 UART 1 FIFO Level Marker Interrupt Register **0x(FF)FFF90C**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	TFE DMAEN	TFH DMAEN				TXFIFO LEVEL MARKER			RFF DMAEN	RFH DMAEN				RXFIFO LEVEL MARKER		
TYPE	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	1	0
	0x0102															

Table 19-11. FIFO Level Marker Interrupt Register Description

Name	Description	Setting
TFE DMAEN Bit 15	**UART 2 Transmit FIFO Empty DMA Enable**—This active high bit can be set to enable a TXFIFO DMA request when the UART 2 Transmit FIFO Empty status flag sets.	0 = Disable (default). 1 = Enable.
TFH DMAEN Bit 14	**UART 2 Transmit FIFO Half DMA Enable**—This active high bit can be set to enable a TXFIFO DMA request when the UART 2 transmit FIFO half status flag sets.	0 = Disable (default). 1 = Enable.
Reserved Bits 13–11	Reserved	These bits are reserved and should be set to 0.
TXFIFO LEVEL MARKER Bits 10–8	**TxFIFO Level Marker**—This field defines the level at which the TxFIFO marker is set. When the TxFIFO status matches the level marker selected here, the TxFIFO half status bit is set and the TXFIFO HALF interrupt is generated if it is enabled.	See Table 19-20 for settings.
RFF DMAEN Bit 7	**UART 2 Receive FIFO Full DMA Enable**—This active high bit can be set to enable an RXFIFO DMA request when the UART 2 receive FIFO full status flag sets.	0 = Disable (default). 1 = Enable.
RFH DMAEN Bit 6	**UART 2 Receive FIFO Half DMA Enable**—This active high bit can be set to enable an RXFIFO DMA request when the UART 2 receive FIFO half status flag sets.	0 = Disable (default). 1 = Enable.

Table 19-11. FIFO Level Marker Interrupt Register Description (Continued)

Name	Description	Setting
Reserved Bits 5–3	Reserved	These bits are reserved and should be set to 0.
RXFIFO LEVEL MARKER Bits 2–0	**RxFIFO Level Marker**—This field defines the level at which the RxFIFO marker is set. When the RxFIFO status matches the level marker selected here, the RxFIFO half status bit is set and the RXFIFO HALF interrupt is generated if it is enabled.	See Table 19-20 for settings.

Table 19-12. FIFO Level Marker Settings

Tx FIFO Level Marker	Number of Slots Empty
000	Disable
001	>= 4
010	>= 8
011	>= 12
100	>= 16
101	>= 20
110	>= 24
111	>= 28

Rx FIFO Level Marker	Number of Bytes Received
000	Disable
001	>= 4
010	>= 8
011	>= 12
100	>= 16
101	>= 20
110	>= 24
111	>= 28

19.5.8 UART 2 Status/Control Register

The UART 2 status/control register (USTCNT2) controls the overall operation of the UART 2 module. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 19-13.

USTCNT2 UART 2 Status/Control Register **0x(FF)FFF910**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	UEN	RXEN	TXEN	CLKM	PEN	ODD	STOP	8/7	ODEN	CTSD	RXFE	RXHE	RXRE	TXEE	TXHE	TXAE
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 19-13. UART 2 Status/Control Register Description

Name	Description	Setting
UEN Bit 15	**UART 2 Enable**—This bit enables the UART 2 module. This bit resets to 0. **Note:** When the UART 2 module is first enabled after a hard reset and before the interrupts are enabled, set the UEN and RXEN bits and perform a word read operation on the URX register to initialize the FIFO and character status bits.	0 = UART 2 module is disabled 1 = UART 2 module is enabled
RXEN Bit 14	**Receiver Enable**—This bit enables the receiver block. This bit resets to 0.	0 = Receiver is disabled and the receive FIFO is flushed 1 = Receiver is enabled
TXEN Bit 13	**Transmitter Enable**—This bit enables the transmitter block. This bit resets to 0.	0 = Transmitter is disabled and the transmit FIFO is flushed 1 = Transmitter is enabled
CLKM Bit 12	**Clock Mode Selection**—This bit selects the receiver's operating mode. When this bit is low, the receiver is in 16x mode, in which it synchronizes to the incoming datastream and samples at the perceived center of each bit period. When this bit is high, the receiver is in 1x mode, in which it samples the datastream on each rising edge of the bit clock. In 1x mode, the bit clock is driven by CLK16. This bit resets to 0.	0 = 16x clock mode (asynchronous mode) 1 = 1x clock mode (synchronous mode)
PEN Bit 11	**Parity Enable**—This bit controls the parity generator in the transmitter and the parity checker in the receiver.	0 = Parity is disabled 1 = Parity is enabled
ODD Bit 10	**Odd Parity**—This bit controls the sense of the parity generator and checker. This bit has no function if the PEN bit is low.	0 = Even parity 1 = Odd parity
STOP Bit 9	**Stop Bit Transmission**—This bit controls the number of stop bits transmitted after a character. This bit has no effect on the receiver, which expects one or more stop bits.	0 = One stop bit is transmitted 1 = Two stop bits are transmitted
8/7 Bit 8	**8- or 7-Bit**—This bit controls the character length. When this bit is set to 7-bit operation, the transmitter ignores data bit 7 and, when receiving, the receiver forces data bit 7 to 0.	0 = 7-bit transmit-and-receive character length 1 = 8-bit transmit-and-receive character length
ODEN Bit 7	**Old Data Enable**—This bit enables an interrupt when the OLD DATA bit in the URX register is set.	0 = OLD DATA interrupt is disabled 1 = OLD DATA interrupt is enabled
CTSD Bit 6	**$\overline{\text{CTS2}}$ Delta Enable**—When this bit is high, it enables an interrupt when the $\overline{\text{CTS2}}$ pin changes state. When it is low, this interrupt is disabled. The current status of the $\overline{\text{CTS2}}$ pin is read in the UTX register.	0 = $\overline{\text{CTS2}}$ interrupt is disabled 1 = $\overline{\text{CTS2}}$ interrupt is enabled
RXFE Bit 5	**Receiver Full Enable**—When this bit is high, it enables an interrupt when the receiver FIFO is full. This bit resets to 0.	0 = RX FULL interrupt is disabled 1 = RX FULL interrupt is enabled
RXHE Bit 4	**Receiver Half Enable**—When this bit is high, it enables an interrupt when the receiver FIFO is more than half full. This bit resets to 0.	0 = RX HALF interrupt is disabled 1 = RX HALF interrupt is enabled
RXRE Bit 3	**Receiver Ready Enable**—When this bit is high, it enables an interrupt when the receiver has at least 1 data byte in the FIFO. When it is low, this interrupt is disabled.	0 = RX interrupt is disabled 1 = RX interrupt is enabled

Table 19-13. UART 2 Status/Control Register Description (Continued)

Name	Description	Setting
TXEE Bit 2	**Transmitter Empty Enable**—When this bit is high, it enables an interrupt when the transmitter FIFO is empty and needs data. When it is low, this interrupt is disabled.	0 = TX EMPTY interrupt is disabled 1 = TX EMPTY interrupt is enabled
TXHE Bit 1	**Transmitter Half Empty Enable**—When this bit is high, it enables an interrupt when the transmit FIFO is less than half full. When it is low, the TX HALF interrupt is disabled. This bit resets to 0.	0 = TX HALF interrupt is disabled 1 = TX HALF interrupt is enabled
TXAE Bit 0	**Transmitter Available for New Data**—When this bit is high, it enables an interrupt if the transmitter has a slot available in the FIFO. When it is low, this interrupt is disabled. This bit resets to 0.	0 = TX AVAIL interrupt is disabled 1 = TX AVAIL interrupt is enabled

19.5.9 UART 2 Baud Control Register

The UART 2 baud control (UBAUD2) register controls the operation of the baud rate generator, the integer prescaler, and the UCLK signal. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 19-14.

UBAUD2 UART 2 Baud Control Register **0x(FF)FFF912**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
			UCLKDIR		BAUD SRC	DIVIDE					PRESCALER					
TYPE			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	1	0
	0x0002															

Table 19-14. UART 2 Baud Control Register Description

Name	Description	Setting
Reserved Bits 15–14	Reserved	These bits are reserved and should be set to 0.
UCLKDIR Bit 13	**UCLK Direction**—This bit controls the direction of the UCLK signal. When this bit is low, the signal is an input, and when it is high, it is an output. However, the SEL*x* bit in the Port E registers must be 0. See Section 16.4.4, “Port E Registers,” on page 16-16 for more information.	0 = UCLK is an input. 1 = UCLK is an output.
Reserved Bit 12	Reserved	This bit is reserved and should be set to 0.
BAUD SRC Bit 11	**Baud Source**—This bit controls the clock source to the baud rate generator.	0 = Baud rate generator source is from system clock. 1 = Baud rate generator source is from UCLK pin (UCLKDIR must be set to 0).

Table 19-14. UART 2 Baud Control Register Description (Continued)

Name	Description	Setting
DIVIDE Bits 10–8	**Divide**—This field controls the clock frequency produced by the baud rate generator.	000 = Divide by 1. 001 = Divide by 2. 010 = Divide by 4. 011 = Divide by 8. 100 = Divide by 16. 101 = Divide by 32. 110 = Divide by 64. 111 = Divide by 128.
Reserved Bits 7–6	Reserved	These bits are reserved and should be set to 0.
PRESCALER Bits 5–0	**Prescaler**—This field controls the division value of the baud generator prescaler. The division value is determined by the following formula: Prescaler division value = 65 (decimal) – PRESCALER	0x00: (not supported) 0x01: Integer prescaler = 65-(1) = 64 0x02: Integer prescaler = 65-(2) = 63 0x03: Integer prescaler = 65-(3) = 62 : 0x3F: Integer prescaler = 65-(63) = 2

19.5.10 UART 2 Receiver Register

The UART 2 receiver (URX2) register indicates the status of the receiver FIFO and character data. The FIFO status bits reflect the current status of the FIFO. At initial power up, these bits contain random data. Before the receiver interrupts are enabled, the UEN and RXEN bits in the USTCNT register should be set. Reading the UART 2 receiver register initializes the FIFO status bits. The receiver interrupts can then be enabled. However, the character status bits are only valid when read with the character bits in a 16-bit read access. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 19-15.

URX2 UART 2 Receiver Register **0x(FF)FFF914**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	FIFO FULL	FIFO HALF	DATA READY	OLD DATA	OVRUN	FRAME ERROR	BREAK	PARITY ERROR	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	0	0	0	0	0	0	0	0
	0x0000															

Table 19-15. UART 2 Receiver Register Description

Name	Description	Setting
FIFO FULL Bit 15	**FIFO Full (FIFO Status)**—This read-only bit indicates that the receiver FIFO is full and may generate an overrun. This bit generates a maskable interrupt. **Note:** FIFO full status flag will assert—set to 1—after receiving 31 bytes. This is due in part because the buffer is a circular link list structure.	0 = Receiver FIFO is not full 1 = Receiver FIFO is full

Table 19-15. UART 2 Receiver Register Description (Continued)

Name	Description	Setting
FIFO HALF Bit 14	**FIFO Half (FIFO Status)**—This read-only bit indicates that the receiver FIFO has four or fewer slots remaining in the FIFO. This bit generates a maskable interrupt.	0 = Receiver FIFO has more than four slots remaining 1 = Receiver FIFO has four or fewer slots remaining
DATA READY Bit 13	**Data Ready (FIFO Status)**—This read-only bit indicates that at least 1 byte is present in the receive FIFO. The character bits are valid only when this bit is set. This bit generates a maskable interrupt.	0 = No data in the receiver FIFO 1 = Data in the receiver FIFO
OLD DATA Bit 12	**Old Data (FIFO Status)**—This read-only bit indicates that data in the FIFO is older than 30 bit times. It is useful in situations where the FIFO FULL or FIFO HALF interrupts are used. If there is data in the FIFO, but the amount is below the FIFO HALF interrupt threshold, a maskable interrupt can be generated to alert the software that unread data is present. This bit clears when the character bits are read.	0 = FIFO is empty or the data in the FIFO is < 30 bit times old 1 = Data in the FIFO is > 30 bit times old
OVRUN Bit 11	**FIFO Overrun (Character Status)**—This read-only bit indicates that the receiver overwrote data in the FIFO. The character with this bit set is valid, but at least one previous character was lost. In normal circumstances, this bit should never be set. It indicates the software is not keeping up with the incoming data rate. This bit is updated and valid for each received character.	0 = No FIFO overrun occurred 1 = A FIFO overrun was detected
FRAME ERROR Bit 10	**Frame Error (Character Status)**—This read-only bit indicates that the current character had a framing error (missing stop bit), which indicates that there may be corrupted data. This bit is updated for each character read from the FIFO.	0 = Character has no framing error 1 = Character has a framing error
BREAK Bit 9	**Break (Character Status)**—This read-only bit indicates that the current character was detected as a BREAK. The data bits are all 0 and the stop bit is also 0. The FRAME ERROR bit will always be set when this bit is set, and if odd parity is selected, PARITY ERROR will also be set. This bit is updated and valid with each character read from the FIFO.	0 = Character is not a break character 1 = Character is a break character
PARITY ERROR Bit 8	**Parity Error (Character Status)**—This read-only bit indicates that the current character was detected with a parity error, which indicates that there may be corrupted data. This bit is updated and valid with each character read from the FIFO. When parity is disabled, this bit always reads 0.	See description
RX DATA Bits 7–0	**Rx Data (Character Data)**—This read-only field is the top receive character in the FIFO. The bits have no meaning if the DATA READY bit is 0. In 7-bit mode, the most significant bit is forced to 0, and in 8-bit mode, all bits are active.	See description

19.5.11 UART 2 Transmitter Register

The UART 2 transmitter (UTX2) register controls how the transmitter operates. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 19-16.

UTX2 UART 2 Transmitter Register **0x(FF)FFF916**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	FIFO EMPTY	FIFO HALF	TX AVAIL	SEND BREAK	NO CTS2	BUSY	CTS2 STAT	CTS2 DELTA	TX DATA							
TYPE	r	r	r	rw	rw	r	r	rw	w	w	w	w	w	w	w	w
RESET	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
	0xE800															

Table 19-16. UART 2 Transmitter Register Description

Name	Description	Setting
FIFO EMPTY Bit 15	**FIFO Empty (FIFO Status)**—This read-only bit indicates that the transmitter FIFO is empty. This bit generates a maskable interrupt. **Note:** FIFO empty status flag will assert—set to 1—when there are 31 free bytes (slots) in the FIFO. This is due in part because the buffer is a circular link list structure.	0 = Transmitter FIFO is not empty 1 = Transmitter FIFO is empty
FIFO HALF Bit 14	**FIFO Half (FIFO Status)**—This read-only bit indicates that the transmitter FIFO is less than half full. This bit generates a maskable interrupt.	0 = Transmitter FIFO is more than half full 1 = Transmitter FIFO is less than half full
TX AVAIL Bit 13	**Transmit FIFO Has a Slot Available (FIFO Status)**—This read-only bit indicates that the transmitter FIFO has at least one slot available for data. This bit generates a maskable interrupt.	0 = Transmitter does not need data 1 = Transmitter needs data
SEND BREAK Bit 12	**Send Break (Tx Control)**—This bit forces the transmitter to immediately send continuous zeros, which creates a break character. See Section 19.3.1.2, “CTS Signal Operation,” for a description of how to generate a break.	0 = Normal transmission 1 = Send break (continuous zeros)
NOCTS2 Bit 11	**Ignore $\overline{\text{CTS2}}$ (Tx Control)**—When this bit is high, it forces the $\overline{\text{CTS2}}$ signal that is presented to the transmitter to always be asserted, which effectively ignores the external pin.	0 = Transmit only when the $\overline{\text{CTS2}}$ signal is asserted 1 = Ignore the $\overline{\text{CTS2}}$ signal
BUSY Bit 10	**Busy (Tx Status) (Read-Only)**—When this bit is high, it indicates that the transmitter is busy sending a character. This bit is asserted when the transmitter state machine is not idle or the FIFO has data in it.	0 = Transmitter is not sending a character 1 = Transmitter is sending a character
CTS2 STAT Bit 9	**$\overline{\text{CTS2}}$ Status ($\overline{\text{CTS2}}$ Bit) (Read-Only)**—This bit indicates the current status of the $\overline{\text{CTS2}}$ signal. A “snapshot” of the pin is taken immediately before this bit is presented to the data bus. When the NOCTS2 bit is high, this bit can function as a general-purpose input.	0 = $\overline{\text{CTS2}}$ signal is low 1 = $\overline{\text{CTS2}}$ signal is high

Table 19-16. UART 2 Transmitter Register Description (Continued)

Name	Description	Setting
CTS2 DELTA Bit 8	**$\overline{\text{CTS2}}$ Delta ($\overline{\text{CTS2}}$ Bit)**—When this bit is high, it indicates that the $\overline{\text{CTS2}}$ signal changed state and generates a maskable interrupt. The current state of the $\overline{\text{CTS2}}$ signal is available on the CTS2 STAT bit. An immediate interrupt may be generated by setting this bit high. The $\overline{\text{CTS2}}$ interrupt is cleared by writing 0 to this bit.	0 = $\overline{\text{CTS2}}$ signal did not change state since it was last cleared 1 = $\overline{\text{CTS2}}$ signal has changed state
TX DATA Bits 7–0	**Tx Data (Character) (Write-Only)**—This write-only field is the parallel transmit-data input. In 7-bit mode, bit 7 is ignored, and in 8-bit mode, all of the bits are used. Data is transmitted with the least significant bit first. A new character is transmitted when this field is written and has passed through the FIFO.	See description

19.5.12 UART 2 Miscellaneous Register

The UART 2 miscellaneous (UMISC2) register contains miscellaneous bits to control test features of the UART 2 module. Some bits, however, are only used for factory testing and should not be used. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 19-17.

UMISC2 UART 2 Miscellaneous Register **0x(FF)FFF918**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	BAUD TEST	CLKSRC	FORCE PERR	LOOP	BAUD RESET	IRTEST		RTS2 TRIG	RTS2 CONT	RTS2	IRDAEN	IRDA LOOP	RXPOL	TXPOL	RXES	RXEE
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	0
	0x0000															

Table 19-17. UART 2 Miscellaneous Register Description

Name	Description	Setting
BAUD TEST Bit 15	**Baud Rate Generator Testing**—This bit puts the baud rate generator in test mode. The integer and non-integer prescalers, as well as the divider, are broken into 4-bit nibbles for testing. This bit should remain 0 for normal operation.	0 = Normal mode. 1 = Test mode.
CLKSRC Bit 14	**Clock Source**—This bit selects the source of the 1x bit clock for transmission and reception. When this bit is high, the bit clock is derived directly from the UCLK pin (it must be configured as an input). When it is low (normal), the bit clock is supplied by the baud rate generator. This bit allows high-speed synchronous applications, in which a clock is provided by the external system.	0 = Bit clock is generated by the baud rate generator. 1 = Bit clock is supplied by the UCLK pin.

Table 19-17. UART 2 Miscellaneous Register Description (Continued)

Name	Description	Setting
FORCE PERR Bit 13	**Force Parity Error**—When this bit is high, it forces the transmitter to generate parity errors, if parity is enabled. This bit is for system debugging.	0 = Generate normal parity. 1 = Generate inverted parity (error).
LOOP Bit 12	**Loopback**—This bit controls loopback for system testing purposes. When this bit is high, the receiver input is internally connected to the transmitter and ignores the RXD2 pin. The TXD2 pin is unaffected by this bit.	0 = Normal receiver operation. 1 = Internally connects the transmitter output to the receiver input.
BAUD RESET Bit 11	**Baud Rate Generator Reset**—This bit resets the baud rate generator counters.	0 = Normal operation. 1 = Reset baud counters.
IRTEST Bit 10	**Infrared Testing**—This bit connects the output of the IrDA circuitry to the TXD2 pin. This provides test visibility to the IrDA module. This bit is for manufacturer testing only and should be set to 0 in the normal application.	0 = Normal operation. 1 = IrDA test mode.
Reserved Bit 9	Reserved	This bit is reserved and should be set to 0.
RTS2 TRIG Bit 8	**$\overline{\text{RTS2}}$ Trigger Source Select**—This bit selects the control of the $\overline{\text{RTS}}$ pin when bit 7 RTS2 CONT =1.	0 = $\overline{\text{RTS2}}$ pin is asserted when the number of filled slots is equal to RxFIFO level marker. 1 = $\overline{\text{RTS2}}$ pin is asserted when only one slot is available in the receiver FIFO.
RTS2 CONT Bit 7	**$\overline{\text{RTS2}}$ Control**—This bit selects the function of the $\overline{\text{RTS2}}$ pin.	0 = $\overline{\text{RTS2}}$ pin is controlled by the RTS2 bit. 1 = $\overline{\text{RTS2}}$ pin is controlled by the receiver FIFO and RTS2 TRIG bit.
RTS2 Bit 6	**Request to Send Pin**—This bit controls the $\overline{\text{RTS2}}$ pin when the RTS2 CONT bit is 0.	0 = $\overline{\text{RTS2}}$ pin is 1. 1 = $\overline{\text{RTS2}}$ pin is 0.
IRDAEN Bit 5	**Infrared Enable**—This bit enables the IrDA interface.	0 = Normal NRZ operation. 1 = IrDA operation.
IRDA LOOP Bit 4	**Loop Infrared**—This bit controls the loopback from the transmitter to the receiver in the IrDA interface. This bit is used for system testing purposes.	0 = No infrared loop. 1 = Connect the infrared transmitter to an infrared receiver.
RXPOL Bit 3	**Receive Polarity**—This bit controls the polarity of the received data.	0 = Normal polarity (1 = idle). 1 = Inverted polarity (0 = idle).
TXPOL Bit 2	**Transmit Polarity**—This bit controls the polarity of the transmitted data.	0 = Normal polarity (1 = idle). 1 = Inverted polarity (0 = idle).
RXES Bit 1	**Receive Error Interrupt Status**— This bit, when set, indicates that one of the error status bits in the receiver register has been set once before. The interrupt must be cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.	0 = No Receive error interrupt occurred. 1 = Receive error interrupt occurred.

Table 19-17. UART 2 Miscellaneous Register Description (Continued)

Name	Description	Setting
RXEE Bit 0	**Receive Error Interrupt Enable**— When this bit is high, it enables an interrupt if any of the error status bits in the receiver register are set during the DMA transfer. This bit is designed for DMA operation. When there is a receive error (for example, overrun, parity, or frame errors) during DMA burst transfer, interrupt is generated. However, the user's program cannot read the receiver register to locate the exact error source after the burst because the receive error status bits change on each received character basis.	0 = Receive error interrupt is disabled 1 = Receive error interrupt is enabled.

19.5.13 UART 2 Non-Integer Prescaler Register

The UART 2 non-integer prescaler register (NIPR2) contains the control bits for the non-integer prescaler. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 19-18.

NIPR2 UART 2 Non-Integer Prescaler Register **0x(FF)FFF91A**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	PRE SEL					SELECT			STEP VALUE							
TYPE	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 19-18. UART 2 Non-Integer Prescaler Register Description

Name	Description	Setting
PRESEL Bit 15	**Prescaler Selection**—This bit selects the input to the baud rate generator divider. Refer to Figure 19-4 on page 19-6 for information about selecting the prescaler.	0 = Divider source is from the integer prescaler. 1 = Divider source is from the non-integer prescaler.
Reserved Bits 14–11	Reserved	These bits are reserved and should be set to 0.
SELECT Bits 10–8	**Tap Selection**—This field selects a tap from the non-integer divider.	000 = Divide range is 2 to 3 127/128 in 1/128 steps. 001 = Divide range is 4 to 7 63/64 in 1/64 steps. 010 = Divide range is 8 to 15 31/32 in 1/32 steps. 011 = Divide range is 16 to 31 15/16 in 1/16 steps. 100 = Divide range is 32 to 63 7/8 in 1/8 steps. 101 = Divide range is 64 to 127 3/4 in 1/4 steps. 110 = Divide range is 128 to 255 1/2 in 1/2 steps. 111 = Divide value is 1.

Table 19-18. UART 2 Non-Integer Prescaler Register Description (Continued)

Name	Description	Setting
STEP VALUE Bits 7–0	**Step Value**—This field selects the non-integer prescaler's step value.	0000 0000. Step = 0. 0000 0001. Step = 1. . . . 1111 1110. Step = 254. 1111 1111. Step = 255.

19.5.14 UART2 FIFO Level Marker Interrupt Register

The UART2 FIFO level marker register configures the level at which either the RxFIFO or the TxFIFO of UART2 reports a half-full condition. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 19-19.

HMARK2 UART 2 FIFO Level Marker Interrupt Register **0x(FF)FFF91C**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	TFE DMAEN	TFH DMAEN				TXFIFO LEVEL MARKER			RFF DMAEN	RFH DMAEN				RXFIFO LEVEL MARKER		
TYPE	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	1	0
	0x0102															

Table 19-19. FIFO Level Marker Interrupt Register Description

Name	Description	Setting
TFE DMAEN Bit 15	**UART 2 Transmit FIFO Empty DMA Enable**—This active high bit can be set to enable a TXFIFO DMA request when the UART 2 transmit FIFO empty status flag sets.	0 = Disable (default). 1 = Enable.
TFH DMAEN Bit 14	**UART 2 Transmit FIFO Half DMA Enable**—This active high bit can be set to enable a TXFIFO DMA request when the UART 2 transmit FIFO half status flag sets.	0 = Disable (default). 1 = Enable.
Reserved Bits 13–11	Reserved	These bits are reserved and should be set to 0.
TXFIFO LEVEL MARKER Bits 10–8	**TxFIFO Level Marker**—This field defines the level at which the TxFIFO marker is set. When the TxFIFO status matches the level marker selected here, the TxFIFO half status bit is set and the TXFIFO HALF interrupt is generated if it is enabled.	See Table 19-20 for settings.
RFF DMAEN Bit 7	**UART 2 Receive FIFO Full DMA Enable**—This active high bit can be set to enable an RXFIFO DMA request when the UART 2 receive FIFO full status flag sets.	0 = Disable (default). 1 = Enable.
RFH DMAEN Bit 6	**UART 2 Receive FIFO Half DMA Enable**—This active high bit can be set to enable an RXFIFO DMA request when the UART 2 receive FIFO half status flag sets.	0 = Disable (default). 1 = Enable.

Table 19-19. FIFO Level Marker Interrupt Register Description (Continued)

Name	Description	Setting
Reserved Bits 5–3	Reserved	These bits are reserved and should be set to 0.
RXFIFO LEVEL MARKER Bits 2–0	**RxFIFO Level Marker**—This field defines the level at which the RxFIFO marker is set. When the RxFIFO status matches the level marker selected here, the RxFIFO half status bit is set and the RXFIFO HALF interrupt is generated if it is enabled.	See Table 19-20 for settings.

Table 19-20. FIFO Level Marker Settings

Tx FIFO Level Marker	Number of Slots Empty
000	Disable
001	>= 4
010	>= 8
011	>= 12
100	>= 16
101	>= 20
110	>= 24
111	>= 28

Rx FIFO Level Marker	Number of Bytes Received
000	Disable
001	>= 4
010	>= 8
011	>= 12
100	>= 16
101	>= 20
110	>= 24
111	>= 28

19.6 UART Register Set

The UART device controller is regulated by the set of registers described in Section 19.5. Table 19-21 summarizes the register addresses and reset states of both units.

Table 19-21. Register Set in UART 1 and UART 2

Address	Name	Width	Description	Reset Value
0xFFFFF900	USTCNT1	16	UART 1 status/control register	0x0000
0xFFFFF902	UBAUD1	16	UART 1 baud control register	0x0002
0xFFFFF904	URX1	16	UART 1 receiver register	0x0000
0xFFFFF906	UTX1	16	UART 1 transmitter register	0xE800
0xFFFFF908	UMISC1	16	UART 1 miscellaneous register	0x0000
0xFFFFF90A	NIPR1	16	UART 1 non-integer prescaler register	0x0000
0xFFFFF90C	HMARK1	16	UART 1 FIFO level mark register	0x0102

Table 19-21. Register Set in UART 1 and UART 2

0xFFFFF90E	-	-	Reserved	-
0xFFFFF910	USTCNT2	16	UART 2 status/control register	0x0000
0xFFFFF912	UBAUD2	16	UART 2 baud control register	0x0002
0xFFFFF914	URX2	16	UART 2 receiver register	0x0000
0xFFFFF916	UTX2	16	UART 2 transmitter register	0xE800
0xFFFFF918	UMISC2	16	UART 2 miscellaneous register	0x0000
0xFFFFF91A	NIPR2	16	UART 2 non-integer prescaler register	0x0000
0xFFFFF91C	HMARK2	16	UART 2 FIFO level mark register	0x0102
0xFFFFF91E	-	-	Reserved	-

Chapter 20
Configurable Serial Peripheral Interface

The MC68SZ328 contains a single configurable serial peripheral interface (CSPI) module. This chapter describes the operation and programming of the CSPI module.

The major architectural change from MC68VZ328, in which SPI 2 can only operate as a master-mode SPI module, while SPI 1 may operate as a master- or slave-configurable SPI interface module; the MC68SZ328 contains only one CSPI, which is an enhanced version of the MC68VZ328 SPI 1. The CSPI can be configured to be master or slave with DMA support.

20.1 CSPI Overview

This section discusses how the CSPI may be used to communicate with external devices. The CSPI contains an 8 × 16 data-in FIFO and an 8 × 16 data-out FIFO. Incorporating the $\overline{\text{DATA\_READY}}$ and $\overline{\text{SS}}$ control signals enables faster data communication with fewer software interrupts. Figure 20-1 is the CSPI block diagram.

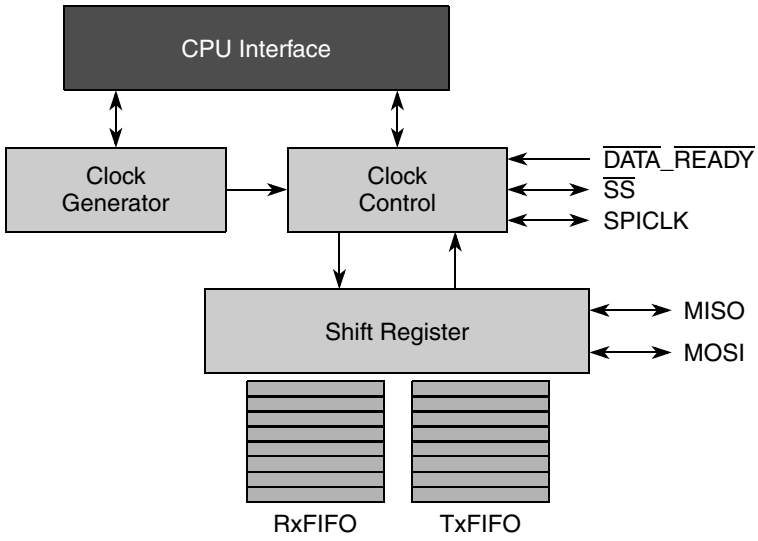


Figure 20-1. CSPI Block Diagram

20.2 CSPI Operation

This section describes the operation of the CSPI.

20.2.1 Using CSPI as Master

When the CSPI is configured as master, it uses a serial link to transfer data between the MC68SZ328 and a peripheral device. A chip-enable signal and a clock signal are used to transfer data between the two devices. If the external device is a transmit-only device, then the output port of the CSPI master can be ignored and used for other purposes. In order to utilize the internal TxD and RxD data FIFOs, two auxiliary output signals, slave select ($\overline{SS}$) and CSPI data ready ($\overline{DATA\_READY}$), are used for data transfer rate control. The user may also program the sample period control register to a fixed data transfer rate.

20.2.2 Using CSPI as Slave

When the CSPI is configured as slave, the CSPI control register can be configured to match the external CSPI master's timing. $\overline{SS}$ becomes an input signal and can be used for data latching from and loading to the internal data shift registers, as well as to increment the data FIFO. Figure 20-2 shows the generic CSPI timing.

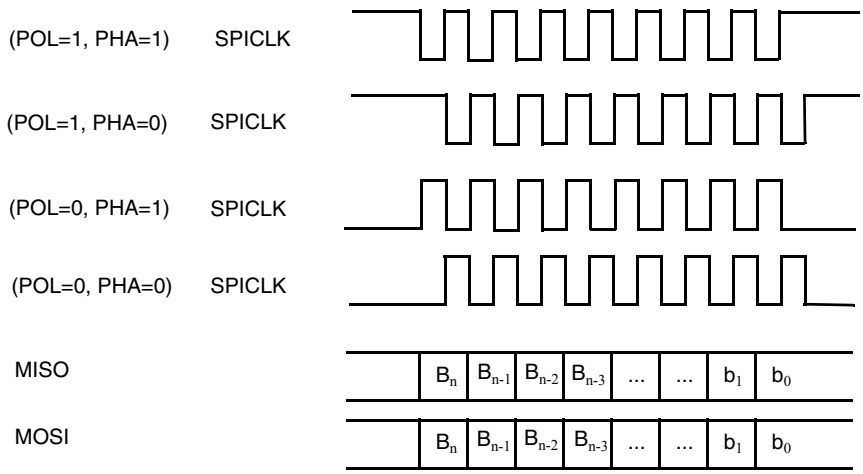


Figure 20-2. CSPI Generic Timing

NOTE:

CSPI does not consume any power when it is disabled.

20.2.3 CSPI Phase and Polarity Configurations

When CSPI is used as master, the CSPI clock (SPICLK) signal is used to transfer data in and out of the shift register. Data is clocked using any one of four programmable clock phase and polarity variations. During phase 0 operation, output data changes on the falling clock edges, and input data is shifted in on rising edges. The most significant bit is output when the CPU loads the transmitted data. In phase 1 operation, 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 SPICLK edge. The polarity of SPICLK may be configured (to invert the SPICLK signal), but it does not change the edge-triggered events that are internal to the CSPI. This flexibility allows it to operate with most serial peripheral devices available in the marketplace.

20.2.4 SPI Signals

The following signals are used to control CSPI :

- MOSI—Master Out/Slave In bidirectional signal. It is the TxD output signal from the data shift register when in master mode. In slave mode it is the RxD input to the data shift register.
- MISO—Master In/Slave Out bidirectional signal. It is the RxD input signal to the data shift register in master mode. In slave mode it is the TxD output from the data shift register.
- SPICLK—CSPI Clock bidirectional signal. It is the CSPI clock output in master mode. In slave mode it is the input CSPI clock signal.
- $\overline{SS}$—Slave Select bidirectional signal. It is an output in master mode and an input in slave mode.
- $\overline{DATA\_READY}$—CSPI Data Ready input signal is used only in master mode. It edge- or level-triggers a CSPI burst when used.

20.3 CSPI Programming Model

This section provides information for programming CSPI.

20.3.1 CSPI Receive Data Register

This read-only register holds the top of the 8 × 16 RxFIFO, which receives data from an external CSPI device during a data transaction. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 20-1.

SPIRXD CSPI Receive Data Register **0x(FF)FFF700**

	BIT15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	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	0	0	0	0	0	0	0	0
	0x0000															

Table 20-1. CSPI Receive Data Register Description

Name	Description	Setting
DATA Bits15–0	**Data**—Top of CSPI's RxFIFO (8 × 16)	The data in this register has no meaning if the RR bit in the interrupt control/status register is cleared.

20.3.2 CSPI Transmit Data Register

This write-only data register is the top of the 8 × 16 TxFIFO. Writing to TxFIFO is permitted as long as TxFIFO is not full, even if the exchange (XCH) bit is set. For example, a user may write to TxFIFO during the CSPI data exchange process. In either master or slave mode, a maximum of 8 data words are loaded. Data written to this register can be either 8-bit or 16-bit size. The number of bits to be shifted out of a 16-bit FIFO element is determined by the bit count setting in the CSPI status/control register. The unused MSBs are discarded and may be written with any value. For example, to transfer 10-bit data, a 16-bit word is written to the CSPI transmit data register (SPITXD) register, and the 6 MSBs are treated as “don’t care” and will not be shifted out. In slave mode, if no data is loaded to the TxFIFO, data bits received from MOSI pin in last transfer are shifted out serially as the TxD signal. Writes to this register are ignored while the CSPI enable (SPIEN) bit in the CSPI control/status register is clear. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 20-2.

SPITXD CSPI Transmit Data Register **0x(FF)FFF702**

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

Table 20-2. CSPI Transmit Data Register Description

Name	Description	Setting
DATA Bits 15–0	**Data**—Top CSPI data to be loaded to the 8 × 16 TxFIFO	See description

20.3.3 CSPI Control/Status Register

This register controls the configuration and operation of the CSPI module. The bit position assignments for this register are shown in the following register display. The settings for this register are described inTable 20-3.

SPICONT CSPI Control/Status Register **0x(FF)FFF704**

BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
DATA RATE			DRCTL		MODE	SPIEN	XCH	SS POL	SS CTL	PHA	POL	BIT COUNT			
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 20-3. CSPI Control/Status Register Description

Name	Description	Setting
DATA RATE Bits 15–13	**Data Rate**—This field selects the bit rate of the SCLK based on the division of the system clock. The master clock for CSPI in master mode is SYSCLK.	000 = Divide SYSCLK by 4 001 = Divide SYSCLK by 8 010 = Divide SYSCLK by 16 011 = Divide SYSCLK by 32 100 = Divide SYSCLK by 64 101 = Divide SYSCLK by 128 110 = Divide SYSCLK by 256 111 = Divide SYSCLK by 512
DRCTL Bits 12–11	**$\overline{\text{DATA\_READY}}$ Control**—In master mode, these 2 bits select the waveform of the $\overline{\text{DATA\_READY}}$ input signal. In slave mode, they have no effect.	00 = Don't care $\overline{\text{DATA\_READY}}$ 01 = Falling edge trigger input 10 = Active low level trigger input 11 = RSV
MODE Bit 10	**CSPI Mode Select**—This bit selects the mode of CSPI.	0 = CSPI is slave mode 1 = CSPI is master mode
SPIEN Bit 9	**CSPI Enable**—This bit enables the CSPI and must be asserted before initiating an exchange. Writing a 0 to this bit flushes the Rx and Tx FIFOs.	0 = Serial peripheral interface is disabled 1 = Serial peripheral interface is enabled
XCH Bit 8	**Exchange**—In master mode, writing a 1 to this bit triggers a data exchange. This bit remains set while either the exchange is in progress or CSPI 1 is waiting for active DATA_READY input while DATA_READY is enabled. This bit is cleared automatically when all data in the TxFIFO and shift registers are shifted out. In slave mode, this bit must be clear.	1 = Initiates exchange (write) or busy (read) 0 = Idle
SSPOL Bit 7	**$\overline{\text{SS}}$ Polarity Select**—In both master and slave modes, this bit selects the polarity of $\overline{\text{SS}}$ signal.	0 = Active low 1 = Active high

Table 20-3. CSPI Control/Status Register Description (Continued)

Name	Description	Setting
SSCTL Bit 6	**$\overline{SS}$ Waveform Select**—In master mode, this bit selects the output wave form for the $\overline{SS}$ signal. In slave mode, this bit controls RxFIFO advancement.	Master Mode: 0 = $\overline{SS}$ stays low between CSPI bursts 1 = Insert pulse between CSPI bursts Slave Mode: 0 = RxFIFO advanced by Bit Count 1 = RxFIFO advanced by $\overline{SS}$ rising edge
PHA Bit 5	**Phase**—This bit controls the clock/data phase relationship.	0 = Phase 0 operation 1 = Phase 1 operation
POL Bit 4	**Polarity**—This bit controls the polarity of the SCLK signal.	0 = Active high polarity (0 = idle) 1 = Active low polarity (1 = idle)
BIT COUNT Bits 3–0	**Bit Count**—This field selects the length of the transfer. A maximum of 16 bits can be transferred. In master mode, a 16-bit data word is loaded from TxFIFO to the shift register, and only the least significant n bits (n = BIT COUNT) are shifted out. The next 16-bit word is then loaded to the shift register. In slave mode, when the SSCTL bit is 0, this field controls the number of bits received as a data word loaded to RxFIFO. When the SSCTL bit is 1, this field is ignored.	0000 = 1-bit transfer 0001 = 2-bit transfer . . . 1110 = 15-bit transfer 1111 = 16-bit transfer

20.3.4 CSPI Interrupt Control/Status Register

This register is used to provide interrupt control and status of various operations in CSPI. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 20-4.

SPIINTCS CSPI Interrupt Control/Status Register **0x(FF)FFF706**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	BOEN	ROEN	RFEN	RHEN	RREN	TFEN	THEN	TEEN	BO	RO	RF	RH	RR	TF	TH	TE
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 20-4. CSPI Interrupt Control/Status Register Description

Name	Description	Setting
BOEN Bit 15	**Bit Count Overflow Interrupt Enable**—This bit, when set, allows an interrupt to be generated when an overflow bit count condition exists. See the description of the bit count overflow (BO), bit 7, for details.	0 = Disable bit count overflow interrupt. 1 = Enable bit count overflow interrupt.
ROEN Bit 14	**RxFIFO Overflow Interrupt Enable**—This bit, when set, allows an interrupt to be generated when an overflow occurs in the RxFIFO. See the description of the RxFIFO overflow (RO), bit 6, for details.	0 = Disable RxFIFO overflow interrupt. 1 = Enable RxFIFO overflow interrupt.
RFEN Bit 13	**RxFIFO Full Interrupt Enable**—This bit, when set, allows an interrupt to be generated when there are 8 data words in the RxFIFO. See the description of the RxFIFO full status (RF), bit 5, for details.	0 = Disable RxFIFO full interrupt enable. 1 = Enable RxFIFO full interrupt enable.
RHEN Bit 12	**RxFIFO Half Interrupt Enable**—This bit, when set, allows an interrupt to be generated when the contents of the RxFIFO is more than or equal to 4 data words. See the description of the RxFIFO half status (RH), bit 4, for details.	0 = Disable half interrupt enable. 1 = Enable half interrupt enable.
RREN Bit 11	**RxFIFO Data Ready Interrupt Enable**—This bit, when set, allows an interrupt to be generated when at least 1 data word is ready in the RxFIFO. See the description of the RxFIFO data ready status (RR), bit 3, for details.	0 = Disable data ready interrupt enable. 1 = Enabled data ready interrupt enable.
TFEN Bit 10	**TxFIFO Full Interrupt Enable**—This bit, when set, causes an interrupt to be generated when the TxFIFO buffer is full and the RFEN bit is set.	0 = Disable TxFIFO full interrupt. 1 = Enable TxFIFO full interrupt.
THEN Bit 9	**TxFIFO Half Interrupt Enable**—This bit, when set, causes an interrupt to be generated when the TxFIFO buffer is half empty and the THEN bit is set.	0 = Disable TxFIFO half interrupt. 1 = Enable TxFIFO half interrupt.

Table 20-4. CSPI Interrupt Control/Status Register Description (Continued)

Name	Description	Setting
TEEN Bit 8	**TxFIFO Empty Interrupt Enable**—This bit, when set, causes an interrupt to be generated when the TxFIFO buffer is empty and the TxFIFO empty status (TE), bit 0, is set.	0 = Disable TxFIFO empty interrupt. 1 = Enable TxFIFO empty interrupt.
BO Bit 7	**Bit Count Overflow**—This bit is set when the CSPI is in “slave CSPI FIFO advanced by $\overline{SS}$ rising edge” mode and the slave is receiving more than 16 bits in one burst. This bit is cleared after a data read from the SPIRXD register. **Note:** There is nothing to indicate which data word has 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 1 data word in RxFIFO has bit count overflow error.
RO Bit 6	**RxFIFO Overflow**—This bit indicates that the RxFIFO has overflowed and at least 1 data word is has been overwritten. The RO flag is automatically cleared after a data read.	0 = RxFIFO has not overflowed. 1 = RxFIFO has overflowed. At least 1 data word in the RxFIFO is overwritten.
RF Bit 5	**RxFIFO Full Status**—This bit, when set, indicates that there are 8 data words in RxFIFO.	0 = Less than 8 data words in RxFIFO. 1 = 8 data words in RxFIFO.
RH Bit 4	**RxFIFO Half Status**—This bit, when set, indicates the contents of the RxFIFO is more than or equal to 4 data words.	0 = Contents of RxFIFO is less than 4 data words. 1 = Contents of RxFIFO is greater than or equal to 4 data words.
RR Bit 3	**RxFIFO Data Ready Status**—This bit, when set, indicates that at least 1 data word is ready in the Rx FIFO.	0 = RxFIFO is empty. 1 = At least 1 data word is ready in the RxFIFO.
TF Bit 2	**TxFIFO Full Status**—This bit, when set, indicates there are 8 data words in the TxFIFO.	0 = Less than 8 data words in TxFIFO. 1 = 8 data words in TxFIFO.
TH Bit 1	**TxFIFO Half Status**—This bit, when set, indicates that the contents of the TxFIFO is more than or equal to 4 data words.	0 = Less than four empty slots in TxFIFO. 1 = More than or equal to 4 empty slots in TxFIFO.
TE Bit 0	**TxFIFO Empty Status**—This bit, when set, causes an interrupt to be generated when the TxFIFO buffer is empty and the TEEN bit is set. **Note:** When the FIFO is empty, data shifting may still be ongoing. To ensure no data transaction is ongoing, read the XCH bit in control register.	0 = At least 1 data word is in Tx FIFO. 1 = TxFIFO is empty.

20.3.5 CSPI Test Register

The configurable CSPI test (SPITEST) register indicates the state machine status of CSPI as well as the number of words currently in the TxFIFO and RxFIFO. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 20-5.

SPITEST CSPI Test Register **0x(FF)FFF708**

BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
				SSTATUS				RXCNT				TXCNT			
TYPE				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 20-5. CSPI Test Register Description

Name	Description	Setting
Reserved Bits 15–12	Reserved	These bits are reserved and should be set to 0.
SSTATUS Bits 11–8	**State Machine Status**—This field indicates the state machine status. These bits are used for test purposes only.	See description.
RXCNT Bits 7–4	**RxFIFO Counter**—This field indicates the number of data words in the RxFIFO.	0000 = RXFIFO is empty. 0001 = 1 data word in RXFIFO. 0010 = 2 data words in RXFIFO. 0011 = 3 data words in RXFIFO. 0100 = 4 data words in RXFIFO. 0101 = 5 data words in RXFIFO. 0110 = 6 data words in RXFIFO. 0111 = 7 data words in RXFIFO. 1000 = 8 data words in RXFIFO.
TXCNT Bits 3–0	**TxFIFO Counter**—This field indicates the number of data words in the TxFIFO.	0000 = TxFIFO is empty. 0001 = 1 data word in TxFIFO. 0010 = 2 data words in TxFIFO. 0011 = 3 data words in TxFIFO. 0100 = 4 data words in TxFIFO. 0101 = 5 data words in TxFIFO. 0110 = 6 data words in TxFIFO. 0111 = 7 data words in TxFIFO. 1000 = 8 data words in TxFIFO.

20.3.6 CSPI Sample Period Control Register

This register controls the time inserted between data transactions in master mode. The time inserted between samples can be from 0 to about 1 second at the resolution of the data rate clock (SPICLK) or the CLK32 signal. Unless a different crystal is used, the CLK32 signal is 32.768 kHz. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 20-6.

SPISPC CSPI Sample Period Control Register **0x(FF)FFF70A**

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

Table 20-6. CSPI Sample Period Control Register Description

Name	Description	Setting
CSRC Bit 15	**Counter Clock Source**—This bit selects the clock source for the sample period counter.	0 = SPICLK clock 1 = CLK32 (32.768 kHz normal crystal used)
WAIT Bits 14–0	**Wait**—Number of clock periods inserted between data transactions in master mode.	When CLK32 is used: 0000 = 0 clocks 0001 = 1 clock 0002 = 2 clocks . . . 7FFF = 32767 clocks (approximately 1 second)

20.3.7 DMA Control Register

This register controls the DMA operation of SP. This is a new register to support DMA function. The settings for this register are described in Table 20-7.

SPIDMA CSPI DMA Control Register **0x(FF)FFF70C**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	THDEN	TEDEN	RFDEN	RHDEN					THDMA	TEDMA	RFDMA	RHDMA				
TYPE	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 20-7. CSPI DMA Control Register Description

Name	Description	Setting
THDEN Bit 15	**TxFIFO Half DMA Request Enable**—	0 = disable. 1 = enable.
TEDEN Bit 14	**TxFIFO Empty DMA Request Enable**—	0 = disable. 1 = enable.
RFDEN Bit 13	**RxFIFO Full DMA Request Enable**—	0 = disable. 1 = enable.
RHDEN Bit 12	**RxFIFO Half DMA Request Enable**—	0 = disable. 1 = enable.
Reserved Bits 11–8	Reserved	These bits are reserved and should be set to 0.
THDMA Bit 7	**TxFIFO Half Status**—	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**—	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**—	0 = Less than 8 data words in RxFIFO. 1 = 8 data words in RxFIFO.
RHDMA Bit 4	**RxFIFO Half Status**—	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 be set to 0.

Chapter 21
USB Device Module

This chapter describes the features and programming model of the MC68SZ328's USB device module.

21.1 Introduction

The Universal Serial Bus (USB) is a peripheral bus standard developed that brings "plug and play" capability to computer peripherals located outside the computer or other device. The MC68SZ328 integrated processor has an integrated USB device controller. This controller is compliant with Universal Serial Bus Specification revision 1.1, which allows for fast "hot" synchronization between the external device and the host computer.

The Universal Serial Bus specification describes three types of devices that can connect to the bus. The USB host is the bus master and periodically polls peripherals to initiate data transfers. There is only one host on the bus. The USB function (or USB device) is a bus slave. It can communicate only with a USB host. It does not generate bus traffic, but only responds to requests from the host. A USB hub is a special class of USB function that adds additional connection points to the bus for more USB devices. The USB controller integrated into the MC68SZ328 is by definition a USB device.

From the user's perspective, this module hides all direct interaction with the USB protocol. The registers allow the user to enable or disable the module, control characteristics of individual endpoints, and monitor traffic flow through the module without ever seeing the low-level details of the USB protocol.

While this module hides all direct interaction with the protocol, some knowledge of the USB is required to properly configure the device for operation on the bus. Programming requirements are covered in this chapter.

21.2 Features

The MC68SZ328's USB device module provides the following USB features to the user:

- Complies with Universal Serial Bus Specification revision 1.1.
- Provides endpoint configurations as shown in Table 21-1. Five pipes are available for mapping. Endpoint 0 is required by the USB specification, while endpoints 1, 2, 3, and 4 can be configured as bulk or interrupt pipes (IN or OUT).
- Supports a frame match interrupt feature to notify the user when a specific USB frame occurs. For DMA access, the maximum packet size for each endpoint is restricted by the FIFO size of the endpoint.
- Supports four bulk/interrupt pipes for 12 Mbps data transfer. The packet sizes are limited to 8, 16, 32, or 64 bytes, and the maximum packet size depends on the FIFO size of the endpoint.
- Requires no power drain from the USB bus.
- Supports a remote wake-up feature through a register bit.

- Provides complete FIFO interrupts (full, empty, error, high, low).
- Supports end-of-frame and start-of-frame interrupt.
- Provides full-speed (12 Mbps) operation.
- Provides intelligence related to packet retries and data framing through the FIFO controller.

Table 21-1. Endpoint Configurations

Endpoint	Direction	Physical FIFO Size	Endpoint Configuration	Maximum Packet Size
0	IN and OUT	32 bytes	Control	32
1	IN or OUT	16 bytes	Bulk or interrupt	16
2	IN or OUT	16 bytes	Bulk or interrupt	16
3	IN or OUT	128 bytes	Bulk or interrupt	64
4	IN or OUT	128 bytes	Bulk or interrupt	64

21.3 Module Components

Figure 21-1 is a block diagram of the complete USB device module. This section briefly describes each of the components in the USB device module.

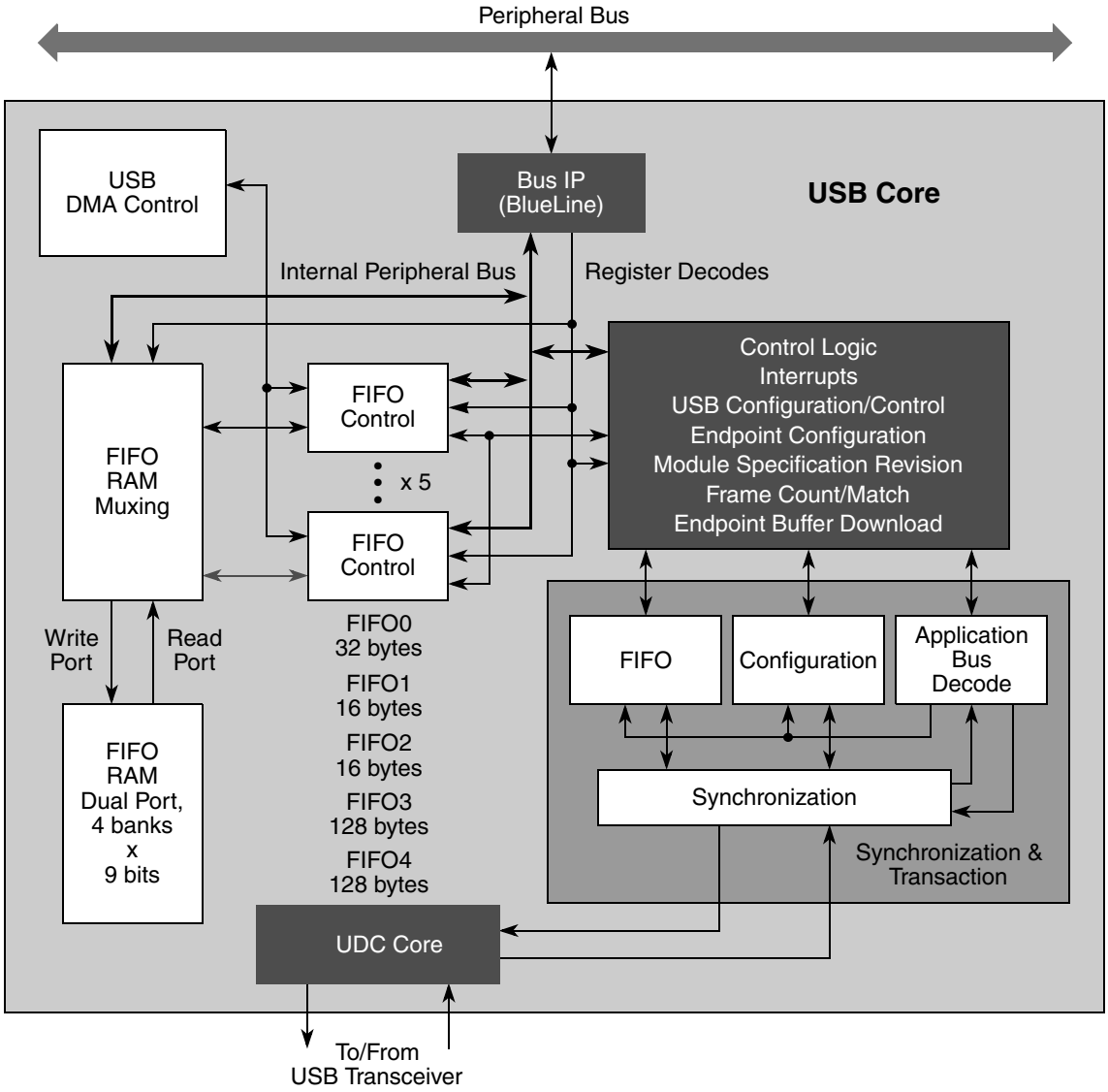


Figure 21-1. USB Device Module Block Diagram

21.3.1 Universal Serial Bus Device Controller Core (UDC)

The Universal Serial Bus Device Controller Core (UDC) interfaces the USB function device to the Universal Serial Bus. The UDC handles all the USB protocol and provides a simple read/write protocol on the function interface (application bus).

The UDC handles all details of managing the USB protocol and presents a simple set of handshakes to the application for managing data flow, vendor commands, and configuration information. It provides the following features:

- Complies with USB Specification revision 1.1.
- Supports USB protocol handling

- Requires no microcontroller or firmware support
- Provides USB device state handling
- Enables clock and data recovery from USB
- Supports bit stripping and bit stuffing functions
- Supports CRC5 checking and CRC16 generation and checking
- Provides serial to parallel data conversion
- Maintains data synchronization bits (DATA0/DATA1 toggle bits)
- Understands and decodes standard USB commands to endpoint 0

21.3.2 Synchronization and Transaction Decoder

The synchronization module performs two functions. It synchronizes the front-end logic timing to the UDC's application bus timing. The front-end logic is targeted for 66.32 MHz operation, while the UDC's application bus runs at 12 MHz for full-speed devices.

The synchronization layer contains a transaction decoder. The application bus protocol is very simple and makes no distinction between RAM, FIFO, and configuration access. The decoder examines the type of transaction requested by the UDC and generates control signals appropriate to that transaction type (for example, RAM or FIFO).

The transaction decoder ensures that all packet transfers occur in units of the maximum packet size for the selected endpoint. This block decodes the buffer address from the UDC's application bus to determine maximum packet size for the current endpoint. It also looks at bytes free and end of frame information from the FIFO module to determine if a packet transfer should occur.

In general, all transfers are the maximum packet size *except* when both of the following conditions apply:

- The transmit FIFO has less than the maximum packet size worth of data available, and
- There is an end of frame indicator in the FIFO.

The transaction decoder also handles hardware retries of USB packets containing errors. The hardware is capable of retransmitting an IN packet to the host or discarding an OUT packet from the host.

The synchronization and transaction decode module contains logic related to the UDC module's clock enable. The UDC module is designed with low-power operation in mind. It includes a gated clock and part of the enable logic for that clock.

21.3.3 Endpoint FIFO Architecture

The USB protocol has some specialized requirements that affect FIFO implementation and essentially create a need for one FIFO per USB endpoint. The USB host can access any endpoint on the function in any order, including the same endpoint in back-to-back transactions. There also is a latency requirement. The USB device must respond to the USB host within a certain number of USB bit times, otherwise the device loses its time slice on the USB until some point in the future. To achieve maximum USB bandwidth use, the USB device must be able to provide full packets of data to the USB host immediately upon request and receive full packets from the host on request. This requirement results in one FIFO per USB endpoint.

Five endpoint FIFOs are available in the USB device module. They are:

- FIFO0: 32 bytes
- FIFO1: 16 bytes
- FIFO2: 16 bytes

- FIFO3: 128 bytes
- FIFO4: 128 bytes

21.3.4 Control Logic

The control logic portion of the module implements the control logic and registers that allow the user to control and communicate with the USB module. The register functions include interrupt status/mask, USB configuration download, FIFO control and access, and processing device requests from the control endpoint.

21.4 USB Transceiver Interface

The MC68SZ328 provides support for a low-cost external USB transceiver interface. Generic USB transceivers that support 3.0 V I/O levels are supported.

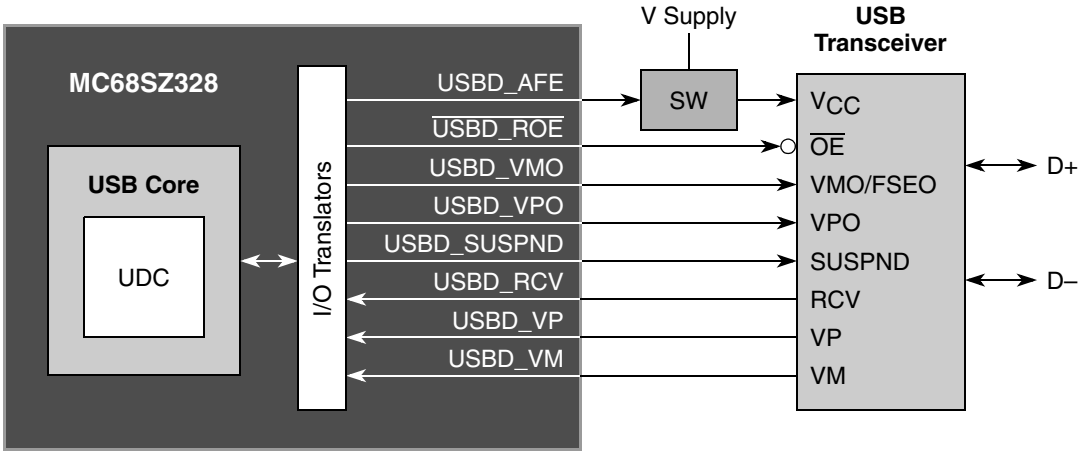


Figure 21-2. USB Module/Transceiver Interface

21.4.1 Signal Description

The USB module has seven signals that may be used to interface with the external USB driver.

- **USBD_AFE**—Analog Front-End Enable. This signal is used to enable the front-end transceiver.
- **$\overline{\text{USBD\_ROE}}$**—Reverse Output Enable. This active low signal is used to control the transceiver to drive its D+/D- signal (to the host connection) according to the signal in USBD_VPO and USBD_VMO (output signal), respectively.
- **USBD_VMO/USBD_VPO**—USB Module Data Output. These signals provide single ended data to the USB Transceiver Transmitter differential driver.
- **USBD_SUSPND**—Transceiver Suspend Enable. This signal, when high, activates a low-power state in the USB transceiver. Normally, when suspended, the transceiver will drive USBD_RCV low and tri-state the USB bus signals D+ and D-.
- **USBD_RCV**—USB Module Receive Data. This signal is a CMOS level driven signal provided by the external USB transceiver. The signal is derived from the D+ and D- differential input to the transceiver.
- **USBD_VP**—Input D+ signal connected directly to the D+.

- **USBD_VM**—Input D- signal connected directly to the D-.

21.5 Interrupt Services

The USB module generates a number of interrupts to indicate conditions requiring attention from the host. There are two categories of interrupts, USB general interrupts and endpoint-specific interrupts.

21.5.1 USB General Interrupts

The USB general interrupts indicate global configuration and status changes that pertain to the USB. All of these interrupts are maskable. When an event occurs that causes an interrupt condition to occur, and the corresponding bit in the interrupt mask register is 0, an interrupt signal is asserted on the module's interface. Writing a 1 to the associated bit in the interrupt register clears the interrupt. After a hard reset, all interrupts are masked by default. This section describes the USB general interrupts.

- **MSOF**—Missed start-of-frame. The MSOF interrupt informs the software that it did not service the SOF interrupt before another SOF was received.
- **SOF**—Start-of-frame. The SOF interrupt indicates that a start-of-frame token was received by the device. The current USB frame number may be read from the USB_FRAME register.
- **RESET_STOP**—End of USB Reset Signalling. The RESET_STOP interrupt indicates that reset signalling on the USB has stopped.
- **RESET_START**—Start of USB Reset Signalling. The RESET_START interrupt indicates that reset signalling on the USB has begun. When reset signalling is active on the USB, the device should not expect to receive any transactions on the bus. This interrupt results in a reset of the UDC into the powered state, but does not cause any specific actions in the USB device module's front-end logic. User software must clear any pending interrupts and ensure that the module is configured properly after the reset. This interrupt only indicates the start of reset signalling. Status of the USB at any given moment can be verified by examining the USB_STAT register.

  NOTE:

  The user is cautioned that the presence of USB reset signalling invalidates any transaction in progress. On detecting RESET_START, user software should read any remaining valid data from the receive FIFOs and flush all the others. When reset signalling is detected, user software must clear any pending interrupts and ensure that the module is configured properly after the reset.

- **WAKEUP**—Resume Signalling Detected. This interrupt asserts when resume signalling is detected on the USB. The device uses this interrupt to wake up from suspend mode and resume active mode. This interrupt is independent of the clock to the module and therefore can be used when the USB device is powered down. The status is reported in the USB_ISR register. In USB power down mode, the USB_ISR register does not update; however, the programmer may check USB_GEN_ISR to validate an asynchronous WAKEUP interrupt assertion.
- **RES**—Resume Active Mode. This interrupt asserts when a state change from suspend to resume occurs in the UDC module. The RES interrupt only indicates the change from suspended to active mode. This interrupt will not assert when the USB device is powered down. In this case, the programmer uses the asynchronous WAKEUP interrupt.
- **SUSP**—USB Suspended. This interrupt asserts when the USB goes into suspend mode. Suspend mode occurs when the device fails to receive any traffic from the USB for a period of 6 ms. Once suspend mode is detected, the device puts itself into a low-power standby mode.

- **FRAME_MATCH**—Match detected in USB_FRAME register. This interrupt asserts when the frame number programmed into the MATCH field of the USB_FRAME register is the same as the FRAME field of the USB_FRAME register.
- **CFG_CHG**—Host changed USB device configuration. This interrupt indicates that the USB host selected a different configuration or alternate interface. Software should read the USB_STAT register to determine the current configuration and interface, and then reconfigure itself accordingly.

21.5.2 Endpoint Interrupts

The endpoint interrupts indicate requests for service by specific USB endpoints. All bits are maskable. Each endpoint's interrupt output is connected to a separate hardware interrupt line. When an event occurs that causes an interrupt condition to occur, and the corresponding bit in the interrupt mask register is 0, an interrupt signal will assert on the module's interface. Writing a 1 to the associated bit in the interrupt register clears the interrupt.

- **FIFO_FULL**—This interrupt asserts when the FIFO is full.
- **FIFO_EMPTY**—This interrupt asserts when the FIFO is empty.
- **FIFO_ERROR**—This interrupt indicates that some abnormal condition occurred in the FIFO. The cause of the error can be verified by reading the USB_EPn_FSTAT register associated with the FIFO that had the error.
- **FIFO_HIGH**—Each FIFO has an alarm register. This interrupt asserts if the byte count in the FIFO is above the level specified by the alarm register.
- **FIFO_LOW**—Each FIFO has an alarm register. This interrupt asserts if the byte count in the FIFO is below the level specified by the alarm register.
- **EOT**—End of Transfer. This interrupt asserts after the last data byte of a USB transfer crosses from the USB device into the UDC module or vice versa. The end of a USB transfer is indicated by either a zero byte packet or by a data packet shorter than the maximum packet size for the endpoint.

NOTE:

The EOT is never asserted at the same time as the DEVREQ interrupt for setup packets. The EOT interrupt asserts after every interrupt packet transfer, complete bulk data transfer, and data phase of control transfer. The EOT interrupt will generally assert along with an EOF interrupt, although an EOT interrupt without an EOF interrupt is possible if a transfer terminated on a USB packet boundary.

- **DEVREQ**—Device Request (Setup Packet). This interrupt indicates that the most recently received packet was a setup or device request packet. Software on the USB device must decode and respond to the packet in order to complete a Vendor, Class, or Standard request. This interrupt will only assert for the control endpoint.
- **MDEVREQ**—Multiple Device Request. This interrupt asserts when two or more setup packets have been received before the DEVREQ interrupt was cleared. This interrupt is used to determine when the USB host aborted a transfer in progress. In this case, the device receives a setup packet, followed by a new setup packet before it completed processing the original command. This interrupt only asserts for the control endpoint.
- **EOF**—End of Frame. This interrupt indicates that an end of frame marker was sent or received on the FIFO/UDC interface. This interrupt asserts if a DEVREQ is received. This interrupt asserts for bulk, control, and interrupt data.

21.5.3 Interrupts, Missed Interrupts, and the USB

There are a number of cases where improper operation of the device could result if interrupts are not serviced in a timely manner. An example case: a CFG_CHG interrupt is received, the device does not service it, and then another CFG_CHG interrupt is received. In this example, a condition exists that could leave the device in an incorrect operating mode. The interrupts of concern are SOF, CFG_CHG, EOT, and DEVREQ.

The missed-interrupt behaviors are as follows:

- **SOF**—If the device misses a start-of-frame interrupt, the MSOF bit asserts in the USB_GEN_ISR register.
- **CFG_CHG**—When the device receives a CFG_CHG interrupt, the module will NAK all traffic from the USB host until software clears the interrupt bit. This prevents the device configuration from getting out of sync with host requests.
- **EOT**—When an end of transfer is received on a BULK OUT endpoint, the device will NAK all traffic on that endpoint until the software clears the interrupt bit. This prevents data from two different transfers from becoming combined in a FIFO.
- **DEVREQ**—When a device request is received, the device will NAK all IN/OUT traffic on the affected endpoint until the software clears the interrupt bit. This ensures that the device correctly identifies the setup packet in the FIFO and clears the FIFO before the data phase is allowed to begin. If multiple setup packets are received, the MDEVREQ interrupt asserts as a warning.

21.6 Programming Model

This section describes the registers for the USB device controller. Section 21.6.1, “USB Register Set Summary,” summarizes the USB device registry, while Section 21.6.2, “Register Descriptions,” details bit names, addresses, and descriptions. All but five registers are 32 bits wide and 32-bit aligned and are accessible by byte, word, or long word. The exceptions are the endpoint n FIFO data registers, which are 16 bits wide and are accessible by byte or word.

21.6.1 USB Register Set Summary

The USB device is controlled by a set of registers. Table 21-2 identifies the register addresses and reset states.

Table 21-2. Register Set in USB Device Controller (USBCORE)

Address	Name	Width	Description	Reset Value
0xFFFE0400	USB_FRAME	32	USB Frame Number and Match	0x00000000
0xFFFE0404	USB_SPEC	32	USB Specification/Release Number	0x00000110
0xFFFE0408	USB_STAT	32	USB Status	0x00000000
0xFFFE040C	USB_CTRL	32	USB Control	0x00000010
0xFFFE0410	USB_CFGSTAT	32	USB Configuration Status	0x80000000
0xFFFE0414	USB_DDAT	32	USB Endpoint Buffer Data	0x000000XX
0xFFFE0418	USB_GEN_ISR	32	USB General Interrupt Status	0x00000000

Table 21-2. Register Set in USB Device Controller (USBCORE) (Continued)

Address	Name	Width	Description	Reset Value
0xFFFE041C	USB_MASK	32	USB General Interrupt Mask	0x800000FF
0xFFFE0420	RESERVED	32	Reserved	0x00000000
0xFFFE0424	USB_ENAB	32	USB Enable	0x00000000
0xFFFE0428	USB_ISR	32	USB Interrupt Status Register	0x00000000
0xFFFE042C	RESERVED	32	Reserved	-
0xFFFE0430	USB_EP0_STATCR	32	USB Endpoint 0 Status/Control	0x00000000
0xFFFE0434	USB_EP0_ISR	32	USB Endpoint 0 Interrupt Status	0x00000080
0xFFFE0438	USB_EP0_MASK	32	USB Endpoint 0 Interrupt Mask	0x000001FF
0xFFFE043C	USB_EP0_FDAT	16	USB Endpoint 0 FIFO Data	0x0000
0xFFFE0440	USB_EP0_FSTAT	32	USB Endpoint 0 FIFO Status	0x00010000
0xFFFE0444	USB_EP0_FCTRL	32	USB Endpoint 0 FIFO Control	0x01000000
0xFFFE0448	USB_EP0_LRFP	32	USB Endpoint 0 FIFO Last Read Frame Pointer	0x00000000
0xFFFE044C	USB_EP0_LWFP	32	USB Endpoint 0 FIFO Last Write Frame Pointer	0x00000000
0xFFFE0450	USB_EP0_FALRM	32	USB Endpoint 0 FIFO Alarm	0x00000000
0xFFFE0454	USB_EP0_FRDP	32	USB Endpoint 0 FIFO Read Pointer	0x00000000
0xFFFE0458	USB_EP0_FWRP	32	USB Endpoint 0 FIFO Write Pointer	0x00000000
0xFFFE045C	RESERVED	32	Reserved	-
0xFFFE0460	USB_EP1_STATCR	32	USB Endpoint 1 Status/Control	0x00000000
0xFFFE0464	USB_EP1_ISR	32	USB Endpoint 1 Interrupt Status	0x00000080
0xFFFE0468	USB_EP1_MASK	32	USB Endpoint 1 Interrupt Mask	0x000001FF
0xFFFE046C	USB_EP1_FDAT	16	USB Endpoint 1 FIFO Data	0x0000
0xFFFE0470	USB_EP1_FSTAT	32	USB Endpoint 1 FIFO Status	0x00010000
0xFFFE0474	USB_EP1_FCTRL	32	USB Endpoint 1 FIFO Control	0x01000000
0xFFFE0478	USB_EP1_LRFP	32	USB Endpoint 1 FIFO Last Read Frame Pointer	0x00000000
0xFFFE047C	USB_EP1_LWFP	32	USB Endpoint 1 FIFO Last Write Frame Pointer	0x00000000
0xFFFE0480	USB_EP1_FALRM	32	USB Endpoint 1 FIFO Alarm	0x00000000
0xFFFE0484	USB_EP1_FRDP	32	USB Endpoint 1 FIFO Read Pointer	0x00000000

Table 21-2. Register Set in USB Device Controller (USBCORE) (Continued)

Address	Name	Width	Description	Reset Value
0xFFFE0488	USB_EP1_FWRP	32	USB Endpoint 1 FIFO Write Pointer	0x0000000
0xFFFE048C	RESERVED	32	Reserved	-
0xFFFE0490	USB_EP2_STATCR	32	USB Endpoint 2 Status/Control	0x00000000
0xFFFE0494	USB_EP2_ISR	32	USB Endpoint 2 Interrupt Status	0x00000080
0xFFFE0498	USB_EP2_MASK	32	USB Endpoint 2 Interrupt Mask	0x000001FF
0xFFFE049C	USB_EP2_FDAT	16	USB Endpoint 2 FIFO Data	0x0000
0xFFFE04A0	USB_EP2_FSTAT	32	USB Endpoint 2 FIFO Status	0x00010000
0xFFFE04A4	USB_EP2_FCTRL	32	USB Endpoint 2 FIFO Control	0x01000000
0xFFFE04A8	USB_EP2_LRFP	32	USB Endpoint 2 FIFO Last Read Frame Pointer	0x00000000
0xFFFE04AC	USB_EP2_LWFP	32	USB Endpoint 2 FIFO Last Write Frame Pointer	0x00000000
0xFFFE04B0	USB_EP2_FALRM	32	USB Endpoint 2 FIFO Alarm	0x00000000
0xFFFE04B4	USB_EP2_FRDP	32	USB Endpoint 2 FIFO Read Pointer	0x00000000
0xFFFE04B8	USB_EP2_FWRP	32	USB Endpoint 2 FIFO Write Pointer	0x00000000
0xFFFE04BC	RESERVED	32	Reserved	-
0xFFFE04C0	USB_EP3_STATCR	32	USB Endpoint 3 Status/Control	0x00000000
0xFFFE04C4	USB_EP3_ISR	32	USB Endpoint 3 Interrupt Status	0x00000080
0xFFFE04C8	USB_EP3_MASK	32	USB Endpoint 3 Interrupt Mask	0x000001FF
0xFFFE04CC	USB_EP3_FDAT	16	USB Endpoint 3 FIFO Data	0x0000
0xFFFE04D0	USB_EP3_FSTAT	32	USB Endpoint 3 FIFO Status	0x00010000
0xFFFE04D4	USB_EP3_FCTRL	32	USB Endpoint 3 FIFO Control	0x01000000
0xFFFE04D8	USB_EP3_LRFP	32	USB Endpoint 3 FIFO Last Read Frame Pointer	0x00000000
0xFFFE04DC	USB_EP3_LWFP	32	USB Endpoint 3 FIFO Last Write Frame Pointer	0x00000000
0xFFFE04E0	USB_EP3_FALRM	32	USB Endpoint 3 FIFO Alarm	0x00000000
0xFFFE04E4	USB_EP3_FRDP	32	USB Endpoint 3 FIFO Read Pointer	0x00000000
0xFFFE04E8	USB_EP3_FWRP	32	USB Endpoint 3 FIFO Write Pointer	0x00000000
0xFFFE04EC	RESERVED	32	Reserved	-
0xFFFE04F0	USB_EP4_STATCR	32	USB Endpoint 4 Status/Control	0x00000000

Table 21-2. Register Set in USB Device Controller (USBCORE) (Continued)

Address	Name	Width	Description	Reset Value
0xFFFE04F4	USB_EP4_ISR	32	USB Endpoint 4 Interrupt Status	0x00000080
0xFFFE04F8	USB_EP4_MASK	32	USB Endpoint 4 Interrupt Mask	0x000001FF
0xFFFE04FC	USB_EP4_FDAT	16	USB Endpoint 4 FIFO Data	0x0000
0xFFFE0500	USB_EP4_FSTAT	32	USB Endpoint 4 FIFO Status	0x00010000
0xFFFE0504	USB_EP4_FCTRL	32	USB Endpoint 4 FIFO Control	0x01000000
0xFFFE0508	USB_EP4_LRFP	32	USB Endpoint 4 FIFO Last Read Frame Pointer	0x00000000
0xFFFE050C	USB_EP4_LWFP	32	USB Endpoint 4 FIFO Last Write Frame Pointer	0x00000000
0xFFFE0510	USB_EP4_FALRM	32	USB Endpoint 4 FIFO Alarm	0x00000000
0xFFFE0514	USB_EP4_FRDP	32	USB Endpoint 4 FIFO Read Pointer	0x00000000
0xFFFE0518	USB_EP4_FWRP	32	USB Endpoint 4 FIFO Write Pointer	0x00000000
0xFFFE051C	RESERVED	32	Reserved	-

21.6.2 Register Descriptions

This section gives detailed descriptions of user-accessible registers and bits within the USB module.

21.6.2.1 USB Frame Number and Match Register

The USB frame number and match register is used for detecting a specific frame.

USB_FRAME USB Frame Number and Match Register **0xFFFE0400**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						MATCH										
TYPE						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															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						FRAME										
TYPE						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 21-3. USB Frame Number and Match Register Settings

Name	Description	Setting
Reserved Bits 31–27	Reserved	These bits are reserved should be set to 0.
MATCH Bits 26–16	**Match**—When the value in the Match field equals the value in FRAME, an interrupt (FRAME_MATCH in the USB_GEN_ISR register) is generated (if not masked).	See description
Reserved Bits 15–11	Reserved	These bits are reserved should be set to 0.
FRAME Bits 10–0	**Frame**—This field is the 11-bit frame number decoded from the SOF packet that leads off each USB frame.	See description

21.6.2.2 USB Specification/Release Number Register

The USB specification/release number register stores information about the version of the USB specification with which the module complies.

USB_SPEC USB Specification/Release Number Register **0xFFFE0404**

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					SPEC											
TYPE					r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
	0x0110															

Table 21-4. USB Specification/Release Number Register Settings

Name	Description	Setting
Reserved Bits 31–12	Reserved	These bits are reserved should be set to 0.
SPEC Bits 11–0	**Specification Version**—This field contains the version of the USB specification to which the underlying USB core is compliant.	0x110 = version 1.1

21.6.2.3 USB Status Register

The USB status register reports the current state of various features of the module. Certain bits are used only for hardware debug mode and read back zeros when debug mode is not enabled.

USB_STAT USB Status Register **0xFFFE0408**

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								RST	SUSP	CFG		INTF		ALTSET		
TYPE								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 21-5. USB Status Register Settings

Name	Description	Setting
Reserved Bits 31–9	Reserved	These bits are reserved should be set to 0.
RST Bit 8	**Reset Signaling Indicator**—This bit indicates the USB reset signalling.	0 = Normal signalling in progress on USB 1 = Reset signalling in progress on USB
SUSP Bit 7	**Suspend**—This bit indicates USB suspend.	0 = USB is not suspended 1 = USB is suspended
CFG[1:0] Bits 6–5	**Configuration Field**—This field indicates the USB configuration. CFG[1:0] reports the current configuration number. See Section 21.7.2, “Configuration Download,” for more information.	See description
INTF[1:0] Bits 4–3	**Interface Indicator Field**—This field reports the current interface number. See Section 21.7.2, “Configuration Download,” for more information.	See description
ALTSET[2:0] Bits 2–0	**Alternate Setting Indicator Field**—This field indicates the current USB alternate setting selection. ALTSET[2:0] indicates the alternate setting number. See Section 21.7.2, “Configuration Download,” for more information.	See description

21.6.2.4 USB Control Register

The USB control register configures numerous features of the USB module.

USB_CTRL USB Control Register **0xFFFE040C**

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									CMD_OVER	CMD_ERROR	USB_SPD	USB_ENA	UDC_RST	AFE_ENA	RESUME
TYPE									rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	0x0010															

Table 21-6. USB Control Register Settings

Name	Description	Setting
Reserved Bits 31–7	Reserved	These bits are reserved should be set to 0.
CMD_OVER Bit 6	**Command Over**—This bit is used to indicate that processing of a device request command has been completed by the device. See Table 21-7 for usage. This bit clears automatically after the UDC core has completed the status phase of a control transfer. CMD_OVER and CMD_ERROR combine to create the handshake code for the status phase of a device request transaction.	See description
CMD_ERROR Bit 5	**Command Error**—This bit is used to indicate that an error has been encountered during processing of a device request. See Table 21-7 for usage. CMD_OVER and CMD_ERROR combine to create the handshake code for the status phase of a device request transaction.	See description
USB_SPD Bit 4	**USB Speed**—This bit sets the low- or full-speed operation for the USB module. **WARNING:** *This module supports only full-speed operation of the device. Selecting slow speed will result in unpredictable operation.*	0 = Low speed 1 = Full speed
USB_ENA Bit 3	**USB Enable**—This bit determines if the USB device module responds to requests from the USB host. The module comes out of reset in the disabled state. The user must ensure that the USB endpoint configuration and USB registers are programmed appropriately before enabling communications. **Note:** This bit does not affect the UDC core's ability to communicate with the host, but affects only the front-end logic's ability to communicate with the UDC core.	0 = USB module front-end logic is disabled. All transactions to or from the UDC will be ignored. (default). 1 = USB module front-end logic is enabled and ready to communicate with the host.
UDC_RST Bit 2	**UDC Reset**—This bit executes a hard reset sequence on the UDC module per the UDC specification. This bit allows the system software to force a reset of the UDC's logic when the system is first connected to the USB, or for debug purposes. Reads of this bit return 0.	See description
AFE_ENA Bit 1	**Analog Front End Enable**—When set to 1, the USBD_AFE pin is high. When set to 0, the USB_AFE pin is low. This pin can be used as a GPIO pin controlled through the USB device registry.	0 = USBD_AFE pin is low 1 = USBD_AFE pin is high

Table 21-6. USB Control Register Settings (Continued)

Name	Description	Setting
RESUME Bit 0	**Resume**—This bit initiates resume signalling on the USB. If the remote wake-up capability is enabled for the current USB configuration, writing a 1 to this bit will cause the USB module to initiate resume signalling on the bus. This bit automatically resets to 0 after a write. Writing a 0 to this bit has no effect. If the remote wake-up capability has been disabled in the UDC via the CLEAR_FEATURE request, then this bit will have no effect. User software should have a time-out feature that aborts the remote wake-up attempt after some time if the RESUME interrupt does not occur. Reads of this bit return 0.	0 = No effect 1 = Initiate resume signalling on the USB if the remote wake-up capability is enabled in the UDC. No effect if the remote wake-up capability is disabled.

Table 21-7. Device Request Status

Result of Transfer	CMD_OVER	CMD_ERROR
Application processes device request successfully	1	0
Application encountered an error processing the request	1	1
Application is busy completing the request	0	X

21.6.2.5 USB Configuration Status Register

The USB_CFGSTAT register provides information about the status of endpoint buffer configuration downloading.

USB_CFGSTAT USB Configuration Status Register **0xFFFE0410**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFG	BSY														
TYPE	r	r														
RESET	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
								0x8000								

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

Table 21-8. USB Configuration Status Register Settings

Name	Description	Setting
CFG Bit 31	**Configuration (Read-Only)**—This bit indicates the configuration status. This bit is set automatically at power-on or hard reset and clears after the last byte of endpoint buffer configuration data has been downloaded into the UDC.	0 = Configuration load has completed. The USB_DDAT writes have no effect. 1 = The USB_DDAT register is currently set to download endpoint buffer configuration data to the UDC.
BSY Bit 30	**Busy (Read-Only)**—This is the configuration download interface busy signal. Because the front-end logic and the UDC module operate on different clocks, a busy indicator is provided to ensure that writes from the USB_DDAT register have a sufficient amount of time to successfully enter the UDC's clock domain.	0 = No write is in progress. 1 = A write is in progress to the UDC module's endpoint buffer. Attempt no other operations on the USB device module until BSY has cleared.
Reserved Bits 29–0	Reserved	These bits are reserved should be set to 0.

21.6.2.6 USB Endpoint Buffer Register

The USB endpoint buffer register allows user access to the endpoint buffer download facility.

USB_DDAT USB Endpoint Buffer Register **0xFFFE0414**

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									DDAT[7:0]							
TYPE									r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x
	0x00xx															

Table 21-9. USB Endpoint Buffer Register Settings

Name	Description	Setting
Reserved Bits 31–8	Reserved	These bits are reserved should be set to 0.

Table 21-9. USB Endpoint Buffer Register Settings (Continued)

Name	Description	Setting
DDAT[7:0] Bits 7–0	**Endpoint Buffer Data Register**—This register allows user access to the endpoint buffer download facility. For endpoint buffer access, the CFG bit in USB_CFGSTAT is set; writes to this register cause the data to be loaded into the UDC module's endpoint buffers, and reads are undefined. When the CFG bit in USB_CFGSTAT is cleared, writes have no effect and reads back will be 0. **Note:** Access to this register is only allowed when the USB_ENA bit in the USB_CTRL register is set to 0. Access at other times is ignored, and reads are undefined.	See description

21.6.2.7 USB General Interrupt Status Register

The USB general interrupt register maintains the status of interrupt conditions pertaining to USB functions.

An interrupt, once set, remains set until it is cleared by writing a 1 to the corresponding bit. Interrupts do not clear automatically if the event that caused them goes away. For example, if reset signalling comes and goes with no intervention from software, both RESET_START and RESET_STOP would be set. Writing a 0 has no effect.

If a register write occurs at the same time an interrupt is received, the interrupt takes precedence over the write.

USB_GEN_ISR USB General Interrupt Status Register **0xFFFE0418**

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									MSOF	SOF	RESET_STOP	RESET_START	RES	SUSP	FRAME_MATCH	CFG_CHG
TYPE									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 21-10. USB General Interrupt Status Register Settings

Name	Description	Setting
WAKEUP Bit 31	**Wake-Up**—This bit indicates a state change from suspend to resume (wake-up) in the UDC module. This bit only indicates the change of state. Clearing the interrupt has no effect on the actual state of the USB. (This bit will still be writable even when the module is disabled.) **Note:** *Use the asynchronous WAKEUP interrupt to power down the module clocks and subsequently use the interrupt to power up the module.*	0 = No state change since last interrupt. 1 = USB has changed state from suspend to resume (wake-up).
Reserved Bits 30–8	Reserved	These bits are reserved should be set to 0.
MSOF Bit 7	**Missed Start-of-Frame**—This bit is the missed start-of-frame interrupt.	0 = No missed start-of-frame. 1 = An SOF interrupt was set, but not cleared, before the next one was received.
SOF Bit 6	**Start-of-Frame**—This bit is the USB start-of-frame interrupt.	0 = No start-of-frame received. 1 = A start-of-frame token has been received by the UDC module. The USB frame number is current.
RESET_STOP Bit 5	**Reset Stop**—This bit indicates the end of reset signalling on the USB.	0 = Reset signalling has not stopped. Does not imply that reset signalling is occurring, but only that no end-of-reset event has occurred. 1 = Reset signalling has stopped.
RESET_START Bit 4	**Reset Start**—This bit indicates start of reset signalling on the USB.	0 = Reset signalling has not started. Does not imply that reset signalling is occurring, but only that no start of reset event has occurred. 1 = Reset signalling has started.
RES Bit 3	**Resume**—This bit indicates a state change from suspend to resume in the UDC module. This bit only indicates the change from suspended to active mode. Clearing the interrupt has no effect on the actual state of the USB.	0 = Indicates that USB has not left the suspended state (but does not imply that the bus is, or ever was, suspended). 1 = USB has left suspend state.
SUSP Bit 2	**Suspend**—This bit indicates a suspend state in the UDC module. This bit only indicates the change from active to suspended mode. Clearing the interrupt has no effect on the actual state of the USB.	0 = USB is not suspended, or the interrupt was cleared. 1 = USB is suspended.
FRAME_MATCH Bit 1	**Frame Match**—This bit indicates a match between the USB frame number and the frame match register.	0 = No match. 1 = Match occurred.
CFG_CHG Bit 0	**Configuration Change**—This bit indicates that a USB configuration (configuration, interface, alternate) change occurred and the software will need to reread the USB status register.	0 = No activity. 1 = Configuration change occurred.

21.6.2.8 USB General Interrupt Mask Register

Setting a bit in the USB general interrupt mask register masks the corresponding interrupt in the USB_GEN_ISR register.

USB_MASK USB General Interrupt Mask Register **0xFFFE041C**

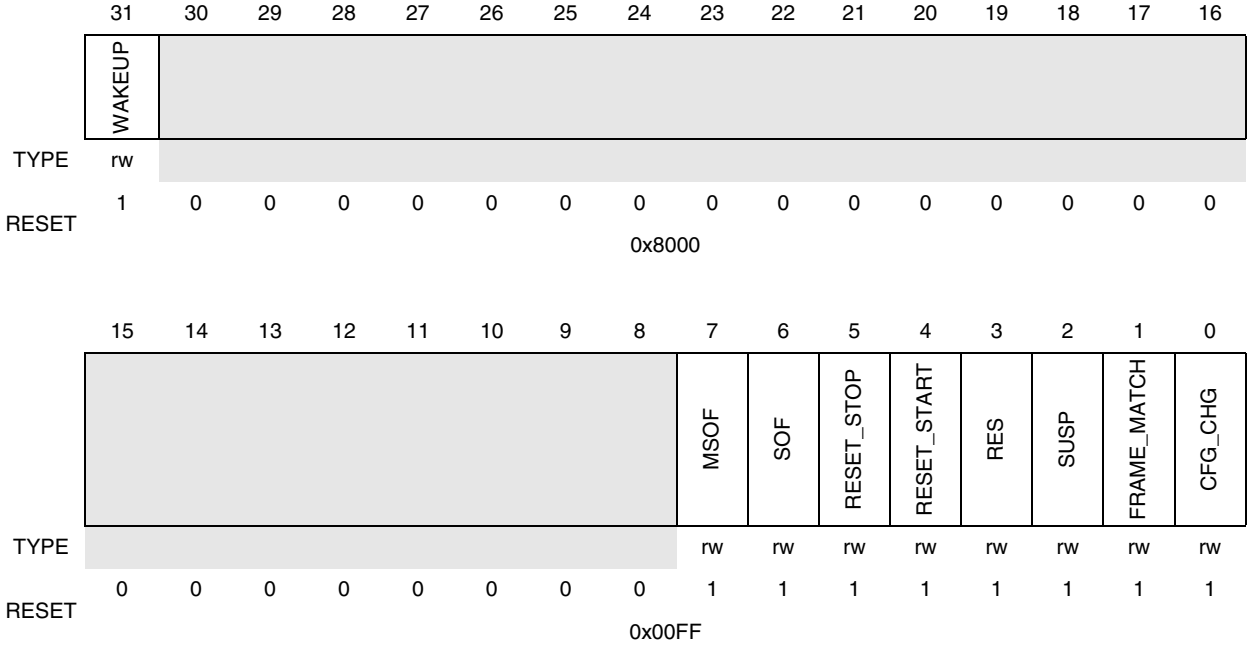


Table 21-11. USB General Interrupt Mask Register Settings

Name	Description	Setting
WAKEUP Bit 31	**Wake-Up Interrupt Mask**—See setting.	0 = Interrupt enabled (unmasked). 1 = Interrupt disabled (masked).
Reserved Bits 30–8	Reserved	These bits are reserved should be set to 0.
MSOF Bit 7	**Missed Start-of-Frame Interrupt Mask**—See setting.	0 = Interrupt enabled (unmasked). 1 = Interrupt disabled (masked).
SOF Bit 6	**Start-Of-Frame Interrupt Mask**—See setting.	0 = Interrupt enabled (unmasked). 1 = Interrupt disabled (masked).
RESET_STOP Bit 5	**Reset Stop Interrupt Mask**—See setting.	0 = Interrupt enabled (unmasked). 1 = Interrupt disabled (masked).
RESET_START Bit 4	**Reset Start**—See setting.	0 = Interrupt enabled (unmasked). 1 = Interrupt disabled (masked).
RES Bit 3	**Resume Interrupt Mask**—See setting.	0 = Interrupt enabled (unmasked). 1 = Interrupt disabled (masked).
SUSP Bit 2	**Suspend Interrupt Mask**—See setting.	0 = Interrupt enabled (unmasked). 1 = Interrupt disabled (masked).
FRAME_MATCH Bit 1	**Frame Match Interrupt Mask**—See setting.	0 = Interrupt enabled (unmasked). 1 = Interrupt disabled (masked).

Table 21-11. USB General Interrupt Mask Register Settings (Continued)

Name	Description	Setting
CFG_CHG Bit 0	**Configuration Change Interrupt Mask**—See setting.	0 = Interrupt enabled (unmasked). 1 = Interrupt disabled (masked).

21.6.2.9 USB Enable Register

The USB enable register controls the operation of the USB functions.

USB_ENAB USB Enable Register **0xFFFE0424**

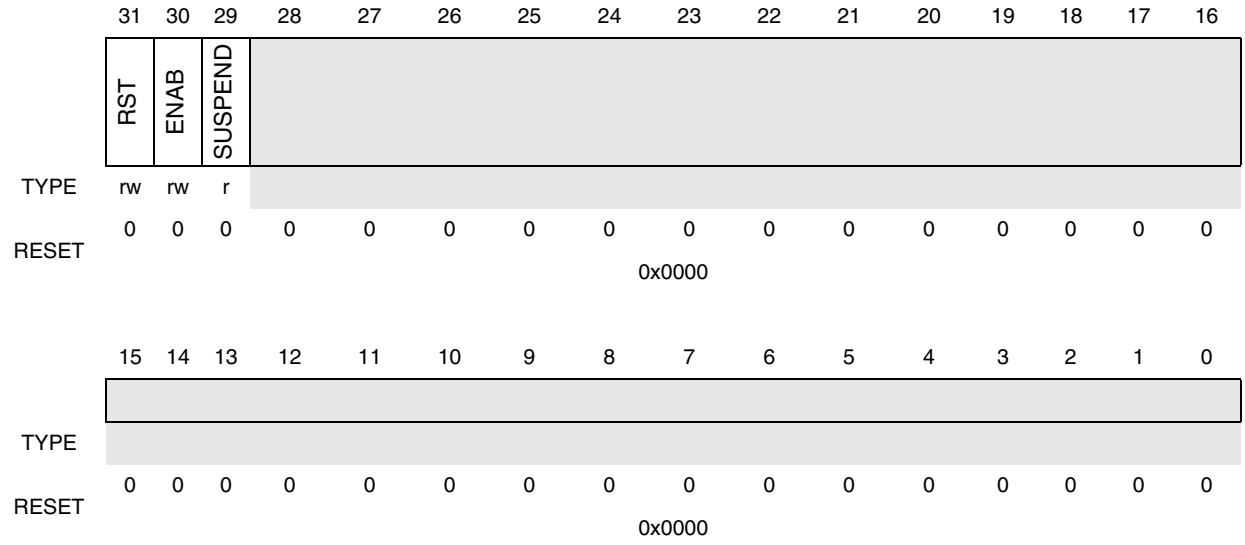


Table 21-12. USB Enable Register Settings

Name	Description	Setting
RST Bit 31	**Reset**—This bit indicates the software reset state of the USB. Writing a 1 to this bit will reset the USB device. The bit clears automatically after the reset is completed. **Note:** When this bit is set, the ENAB bit is automatically set.	0 = No reset in progress 1 = USB reset in progress
ENAB Bit 30	**Enable**—This bit indicates the enabling of the USB. Writing a 1 will enable the device, and a 0 will disable the device. When the USB is disabled, all write access to registers will be ignored except to the USB_GEN_ISR register's WAKEUP bit and this register (USB_ENAB).	0 = Disable USB 1 = Enable USB
SUSPEND Bit 29	**Suspend**—This bit indicates suspend state of the UDC module.	0 = Module is in resume/active state 1 = Module is in suspend state
Reserved Bits 28–0	Reserved	These bits are reserved should be set to 0.

21.6.2.10 USB Interrupt Status Register

The USB_ISR register reports the source of asserted interrupts to the interrupt controller. These bits, if asserted, are cleared at the interrupting source.

USB_ISR USB Interrupt Status Register **0xFFFE0428**

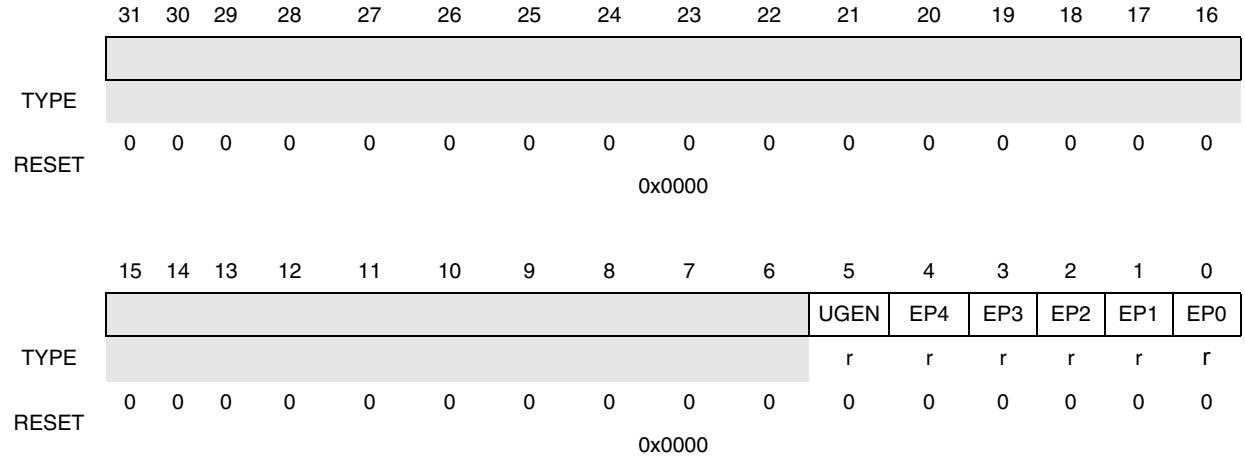


Table 21-13. USB Interrupt Status Register Settings

Name	Description	Setting
Reserved Bits 31–6	Reserved	These bits are reserved and should be set to 0.
UGEN Bit 5	**USB General Interrupt Status**—This bit indicates that a USB general interrupt is asserted and must be cleared in the USB_GEN_ISR register.	0 = No interrupt (default) 1 = Interrupt asserted
EP4 Bit 4	**Endpoint 4 INterrupt Status**—This bit indicates an interrupt from Endpoint 4 is asserted and must be cleared in the USB_EP4_ISR register.	0 = No interrupt (default) 1 = Interrupt asserted
EP3 Bit 3	**Endpoint 3 INterrupt Status**—This bit indicates an interrupt from Endpoint 3 is asserted and must be cleared in the USB_EP3_ISR register.	0 = No interrupt (default) 1 = Interrupt asserted
EP2 Bit 2	**Endpoint 2INterrupt Status**—This bit indicates an interrupt from Endpoint 2 is asserted and must be cleared in the USB_EP2_ISR register.	0 = No interrupt (default) 1 = Interrupt asserted
EP1 Bit 1	**Endpoint 1INterrupt Status**—This bit indicates an interrupt from Endpoint 1 is asserted and must be cleared in the USB_EP1_ISR register.	0 = No interrupt (default) 1 = Interrupt asserted
EP0 Bit 0	**Endpoint 0 INterrupt Status**—This bit indicates an interrupt from Endpoint 0 is asserted and must be cleared in the USB_EP0_ISR register.	0 = No interrupt (default) 1 = Interrupt asserted

21.6.2.11 USB Endpoint n Status/Control Register

The USB endpoint n status/control register allows the user to configure specific aspects of an individual endpoint and monitor endpoint status.

USB_EPn_STATCR — USB Endpoint n Status/Control Register — **0xFFFE0430+(n*0x30)**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										BYTE_COUNT[7:0]						
TYPE										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															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SIP	DIR	MAX[1:0]		TYP[1:0]		ZLPS	FLUSH	FORCE_STALL
TYPE								rw	rw	rw	rw	rw	rw	rw	w	rw
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x0000															

Table 21-14. USB Endpoint n Status/Control Register Settings

Name	Description	Setting
Reserved Bits 31–23	Reserved	These bits are reserved should be set to 0.
BYTE_COUNT[7:0] Bits 22–16	**Byte Count**—This field indicates the number of bytes currently stored in the associated FIFO.	See description.
Reserved Bits 15–9	Reserved	These bits are reserved should be set to 0.
SIP Bit 8	**Setup Packet In Progress**—Setup packet in progress indicator.	0 = Setup data is not being transferred from host to device. 1 = Setup data is currently being transferred from host to device.
DIR Bit 7	**Direction**—This bit controls the transfer direction. This is ignored for the control endpoint.	0 = OUT endpoint (from host to device). 1 = IN endpoint (from device to host).
MAX[1:0] Bits 6–5	**Maximum [1:0]**—This field contains the maximum packet size for the endpoint. **Note:** The maximum packet size cannot be greater than the physical FIFO size for the given endpoint.	00 = 08 bytes 01 = 16 bytes 10 = 32 bytes 11 = 64 bytes
TYP[1:0] Bits 4–3	**Type [1:0]**—This field is the endpoint type. Endpoint 0 is a control endpoint only, while endpoints 1 to 4 may be configured by the user.	00 = Control 01 = Reserved 10 = Bulk 11 = Interrupt

Table 21-14. USB Endpoint n Status/Control Register Settings (Continued)

Name	Description	Setting
ZLPS Bit 2	**Zero Length Packet Send**—This bit signals the USB host that the end of data has been reached in a data transmission when the end of data lands on a packet boundary and there is no short packet to signify the end of data. This bit allows the software to signify that an empty packet should go back to the host. See Section 21.8.2.1, “Data Transfers to the Host,” for more information.	0 = No zero-length packet send. 1 = If the FIFO is empty, and the USB host requests an IN transaction, the module will send a zero-length packet in response. This bit automatically clears once the transaction completes successfully.
FLUSH Bit 1	**Flush**—This write-only bit causes the associated FIFO to be flushed to its empty state. Reading this bit returns 0.	0 = Do nothing. 1 = Initiate flush operation.
FORCE_STALL Bit 0	**Force Stall**—When written, this bit causes the endpoint to respond with a STALL to the next poll. This bit automatically clears when the stall takes effect. **Note:** There is no endpoint stalled indicator; one is not returned from the UDC. The USB host is expected to communicate with the USB device via device requests to fix the stall condition. The USB host will then send a CLEAR_FEATURE request to the UDC module to clear the stall and resume normal operations.	0 = Do nothing. 1 = Force a stall condition.

21.6.2.12 USB Endpoint n Interrupt Status Register

The USB endpoint n interrupt status register monitors the status of a specific endpoint and generates a CPU interrupt each time a monitored event occurs.

An interrupt, once set, remains set until cleared by writing a 1 to the corresponding bit. Interrupts do not clear automatically if the event that caused them goes away. For example, if reset signalling comes and goes with no intervention from software, both RESET_START and RESET_STOP are set. Writing a 0 has no effect.

If a register write occurs at the same time that an interrupt is received, the interrupt takes precedence over the write. If the corresponding bit in the USB_EPn_MASK register is set (interrupt disabled), the interrupt status is still reflected in USB_EPn_ISR.

USB_EPn_ISR USB Endpoint n Interrupt Status Register **0xFFFE0434+(n*0x30)**

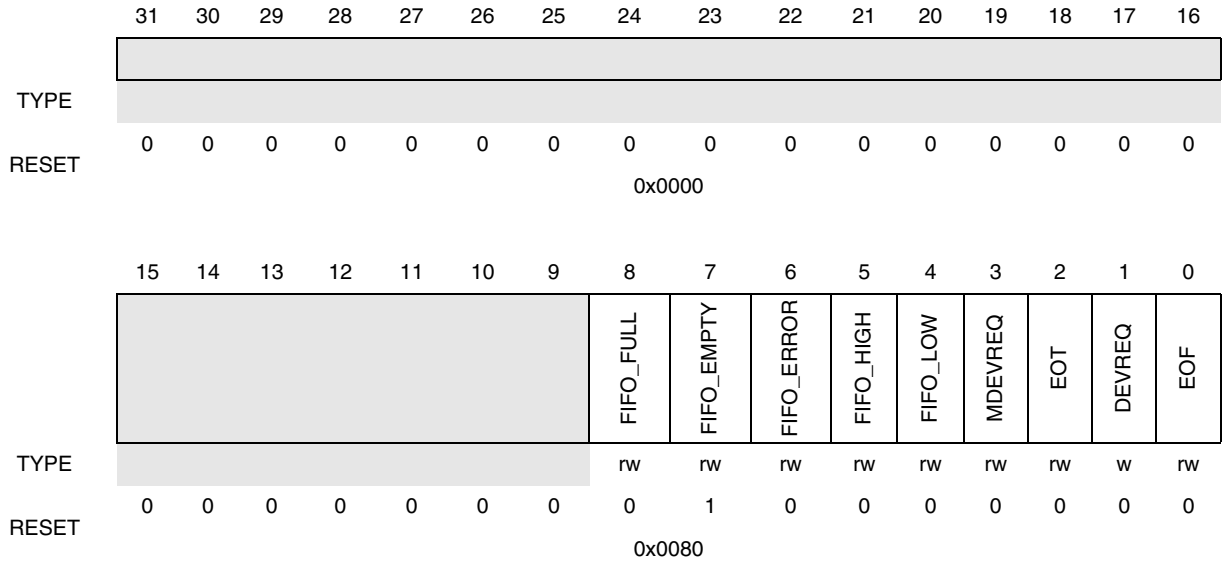


Table 21-15. USB Endpoint n Interrupt Status Register Settings

Name	Description	Setting
Reserved Bits 31–9	Reserved	These bits are reserved should be set to 0.
FIFO_FULL Bit 8	**FIFO Full**—This bit, when set, indicates that the FIFO is full.	0 = Indicates that the FIFO is not full 1 = Indicates that the FIFO is full
FIFO_EMPTY Bit 7	**FIFO Empty**—This bit, when set, indicates that the FIFO is empty.	0 = Indicates that the FIFO is not empty 1 = Indicates that the FIFO is empty
FIFO_ERROR Bit 6	**FIFO Error**—This bit, when set, indicates an error condition in the FIFO controller. The error condition can be checked by reading the FIFO status register (USB_EPn_FSTAT).	0 = Indicates no error condition pending 1 = Indicates an error condition pending
FIFO_HIGH Bit 5	**FIFO High**—For OUT FIFO (receive). This bit indicates that the number of bytes in the FIFO surpasses the high level alarm value.	0 = Less than GR[2:0] data bytes in the FIFO 1 = Less than ALARM[5:0] free bytes in the FIFO
FIFO_LOW Bit 4	**FIFO Low**—For IN FIFO (transmit). This bit indicates that the number of bytes in the FIFO falls below the FIFO low level alarm value.	0 = Less than 4 x GR[2:0] free bytes remaining in the FIFO 1 = Less than ALARM[5:0] data bytes in the FIFO
MDEVREQ Bit 3	**Multiple Device Request**—This bit indicates multiple device requests are pending. This interrupt asserts when a DEVREQ interrupt is pending and another setup packet is received. This bit only asserts for the control endpoint.	0 = Multiple setup packets are not pending 1 = Multiple setup packets are pending

Table 21-15. USB Endpoint n Interrupt Status Register Settings (Continued)

Name	Description	Setting
EOT Bit 2	**End of Transfer**—This bit indicates that the last packet of a USB data transfer has crossed into, or out of, the UDC module. The last packet is identified by its length. Any packet shorter than the maximum packet size for the associated endpoint is considered to be an end of transfer marker. In addition, for the control endpoint only, the EOT interrupt will assert when the number of bytes specified in the wLength field of the setup packet has been transferred.	0 = USB end of transfer not detected 1 = USB end of transfer detected
DEVREQ Bit 1	**Device Request**—This bit indicates a device request on the current endpoint. It only asserts for the control endpoint.	0 = Indicates no request is pending 1 = Indicates a request is pending
EOF Bit 0	**End of Frame**—This bit indicates the end of frame activity for this endpoint. It monitors the data flow between the FIFO and the UDC and indicates when the end of a USB packet is written into the FIFO or the UDC as the end of a frame.	0 = End of frame (USB packet) was not sent/received 1 = End of frame (USB packet) sent/received

21.6.2.13 USB Endpoint n Interrupt Mask Register

The USB endpoint n interrupt mask register allows software to mask individual interrupts for each endpoint. Writing a 1 to a bit in this register masks the corresponding interrupt in the USB_EPn_ISR register. Writing a 0 unmasks the interrupt.

USB_EPn_MASK USB Endpoint n Interrupt Mask Register **0xFFFE0438+(n*0x30)**

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE																
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0x000															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								FIFO_FULL	FIFO_EMPTY	FIFO_ERROR	FIFO_HIGH	FIFO_LOW	MDEVREQ	EOT	DEVREQ	EOF
TYPE								rw	rw	rw	rw	rw	rw	rw	w	rw
RESET	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	0x01FF															

Table 21-16. USB Endpoint n Interrupt Mask Register Settings

Name	Description	Setting
Reserved Bits 31–9	Reserved	These bits are reserved should be set to 0.
FIFO_FULL Bit 8	**FIFO Full Interrupt Mask**—See Setting.	0 = Interrupt enabled (unmasked). 1 = Interrupt disabled (masked).
FIFO_EMPTY Bit 7	**FIFO Empty Interrupt Mask**—See Setting.	0 = Interrupt enabled (unmasked). 1 = Interrupt disabled (masked).
FIFO_ERROR Bit 6	**FIFO Error Interrupt Mask**—See Setting.	0 = Interrupt enabled (unmasked). 1 = Interrupt disabled (masked).
FIFO_HIGH Bit 5	**FIFO High Interrupt Mask**—See Setting.	0 = Interrupt enabled (unmasked). 1 = Interrupt disabled (masked).
FIFO_LOW Bit 4	**FIFO Low Interrupt Mask**—See Setting.	0 = Interrupt enabled (unmasked). 1 = Interrupt disabled (masked).
MDEVREQ Bit 3	**Multiple Device Request Interrupt Mask**—See Setting.	0 = Interrupt enabled (unmasked). 1 = Interrupt disabled (masked).
EOT Bit 2	**End of Transfer Interrupt Mask**—See Setting.	0 = Interrupt enabled (unmasked). 1 = Interrupt disabled (masked).
DEVREQ Bit 1	**Device Request Interrupt Mask**—See Setting.	0 = Interrupt enabled (unmasked). 1 = Interrupt disabled (masked).
EOF Bit 0	**End of Frame Interrupt Mask**—See Setting.	0 = Interrupt enabled (unmasked). 1 = Interrupt disabled (masked).

21.6.2.14 USB Endpoint n FIFO Data Register

The USB endpoint n FIFO data register is the main interface port for the corresponding FIFO. Data that is to be buffered in the FIFO, or has been buffered in the FIFO, is accessed through this register. This register can access data from the FIFO independently of the FIFO's transmit or receive configuration. This register is byte or word accessible.

The FIFO direction (IN or OUT) is controlled by the DIR bit in the USB_EPn_STATCR register. See Section 21.7.2, "Configuration Download," on the direction of the endpoint, which should agree with the FIFO setting.

USB_EPn_FDAT — USB Endpoint n FIFO Data Register — **0xFFFE043C+(n*0x30)**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXDATA[15:0]/TXDATA[15:0]															
TYPE	rw	rw	rw	rw	w	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 21-17. USB Endpoint n FIFO Data Register

Name	Description	Setting
RXDATA[15:0] Bits 15–0	**Receive Data**—This field contains the receive FIFO read data.	See description
TXDATA[15:0] Bits 15–0	**Transmit Data**—This field contains the transmit FIFO write data.	See description

21.6.2.15 USB Endpoint n FIFO Status Register

The USB endpoint n FIFO status register provides information about FIFO status.

USB_EPn_FSTAT — USB Endpoint n FIFO Status Register — **0xFFFE0440+(n*0x30)**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				FRAME[0]	FRAME[1]				ERROR	UFERR	OFERR	FRMRDY	FIFO_FULL	ALARM_STAT	EMPTY
TYPE				r	r				rw	rw	rw	r	r	r	r
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	0x0001														

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

Table 21-18. USB Endpoint n FIFO Status Register Settings

Name	Description	Setting
Reserved Bits 31–28	Reserved	These bits are reserved should be set to 0.
FRAME[0] Bit 27	**Frame [0]**—This read-only bit provides a frame status indicator for non-DMA applications.	0 = Indicates no frame boundary has occurred. 1 = Indicates that a frame boundary has occurred on the [15:8] byte of the bus.
FRAME[1] Bit 26	**Frame [1]**—This read-only bit provides a frame status indicator for non-DMA applications.	0 = Indicates no frame boundary has occurred. 1 = Indicates that a frame boundary has occurred on the [7:0] byte of the bus.

Table 21-18. USB Endpoint n FIFO Status Register Settings (Continued)

Name	Description	Setting
Reserved Bits 25–23	Reserved	These bits are reserved should be set to 0.
ERROR Bit 22	**Error**—This bit indicates that an error occurred in the FIFO controller.	0 = Indicates no error. 1 = Indicates underflow or overflow. Writing a 1 to this bit clears the error indicator. Writing a 0 has no effect.
UFERR Bit 21	**Under Flow Error**—This bit, when set, indicates FIFO underflow.	0 = Indicates no overflow. 1 = Indicates that the read pointer has passed the write pointer. Writing a 1 to this bit clears the UFERR indicator. Writing a 0 has no effect.
OFERR Bit 20	**Over Flow Error**—This bit, when set, indicates FIFO overflow.	0 = Indicates no overflow. 1 = Indicates that the write pointer has passed the read pointer. Writing a 1 to this bit clears the OFERR indicator. Writing a 0 has no effect.
FRMRDY Bit 19	**Frame Ready**—This read-only bit is the frame ready indicator. This bit is inactive when the FIFO is not programmed for frame mode.	0 = No complete frames exist in the FIFO. 1 = One or more complete frames exist in the FIFO.
FIFO_FULL Bit 18	**FIFO Full**—This read-only bit is the FIFO full indicator.	0 = The FIFO is not full. 1 = The FIFO has requested attention because it is full. The FIFO must be read to clear this alarm.

Table 21-18. USB Endpoint n FIFO Status Register Settings (Continued)

Name	Description	Setting
ALARM_STAT Bit 17	**Alarm Status**—This read-only bit indicates a FIFO alarm. The specific alarm condition detected depends upon the FIFO direction. For IN (transmit) FIFOs, the ALARM_STAT bit indicates a “low” service request alarm (low level alarm). • ALARM_STAT asserts when there are less than ALRM[6:0] data bytes in the FIFO, USB_EPn_FDAT[TXDATA(15:0)]. • ALARM_STAT deasserts when there are less than 4 x GR[2:0] free slots (bytes) in the FIFO, USB_EPn_FDAT[TXDATA(15:0)]. For OUT (receive) FIFOs, the ALARM_STAT bit indicates a “high” service request alarm (high level alarm). • ALARM_STAT asserts: — When there are less than ALRM[6:0] free slots (bytes) in the FIFO, USB_EPn_FDAT[RXDATA(15:0)] Or, — If there are end of frame bytes in the FIFO. • ALARM_STAT deasserts: — When there are less than GR[2:0] data bytes in the FIFO, USB_EPnFDAT[RXDATA(15:0)] Or, — If there are no end of frame bytes in the FIFO.	0 = The alarm is not set. 1 = The FIFO has requested attention because it has determined an alarm condition. See the description for detailed information.
EMPTY Bit 16	**Empty**—This read-only bit is the FIFO empty indicator.	0 = The FIFO is not empty. 1 = The FIFO has requested attention because it is empty. The FIFO must be written to clear this alarm.
Reserved Bits 15–0	Reserved	These bits are reserved should be set to 0.

21.6.2.16 USB Endpoint 0 FIFO Control Register

The USB endpoint 0 FIFO control register configures the corresponding endpoint FIFO.

USB_EP0_FCTRL — USB Endpoint 0 FIFO Control Register — **0xFFFE0444**

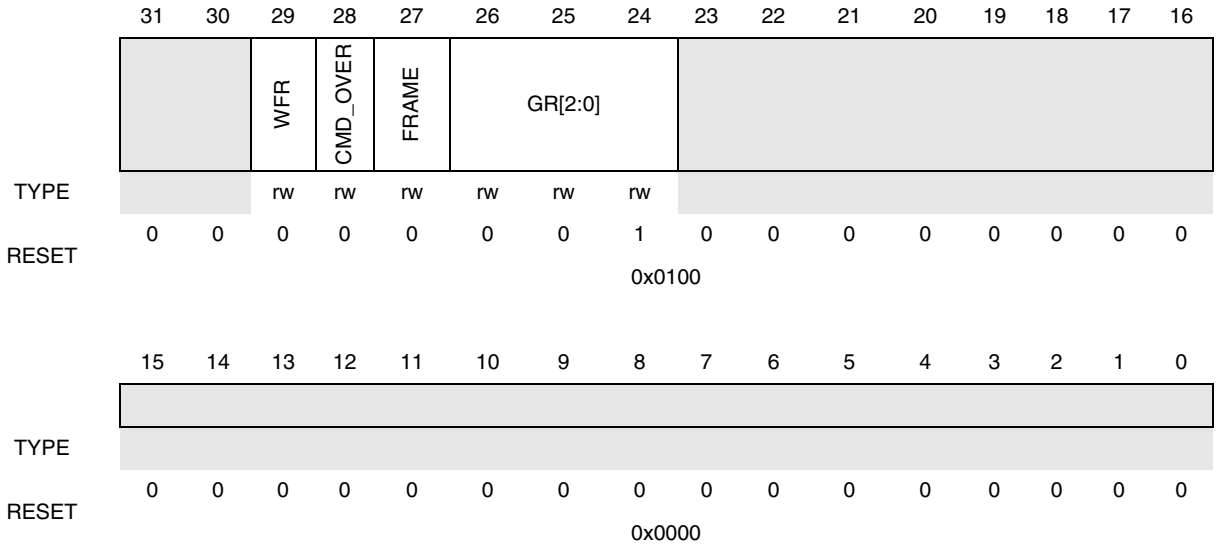


Table 21-19. USB Endpoint n FIFO Control Register Settings

Name	Description	Setting
Reserved Bits 31–30	Reserved	These bits are reserved should be set to 0.
WFR Bit 29	**Write Frame**—This bit determines the end of the current data frame in the FIFO. When set to 1, this bit indicates to the UDC that the next write to the FIFO data register is the end of the frame. For an odd byte packet, the next write will be a byte; for an even byte packet, the next write will be a word. **Note:** This bit only applies for IN (transmit) FIFOs.	0 = Indicates next write to FIFO data register is not the end of frame 1 = Indicates next write to FIFO data register is the end of frame
CMD_OVER Bit 28	CMD_OVER bit —Set the bit to 1 for proper operation of the USB module. Default is 0.	0 = Disables CMD_OVER 1 = Enables CMD_OVER
FRAME Bit 27	**Frame Mode**—This bit indicates the frame mode. In FRAME mode, the FIFO uses its internal frame pointer and information from the peripheral to transfer only full frames of data, as defined by the peripheral. Since the controller only keeps a pointer to the end of the last complete frame, a read request may contain more than one frame of data.	0 = Indicates frame mode enabled 1 = Indicates frame mode disabled

Table 21-19. USB Endpoint n FIFO Control Register Settings (Continued)

Name	Description	Setting
GR[2:0] Bits 26–24	**Granularity**—This field defines the deassertion point of ALARM_STAT for a “high” service request (high level alarm) and for a “low” service request (low level alarm). For OUT (receive) FIFOs, a “high” service request (high level alarm) is deasserted when there are less than GR[2:0] data bytes remaining in the FIFO. For IN (transmit) FIFOs, a “low” service request is deasserted when there are less than (4 x GR[2:0]) free slots (bytes) remaining in the FIFO. **Note:** The direction, type, and packet size are defined in the USB_EPn_STATCR register.	000 = 1 data byte or free slot (byte) 001 = 2 data bytes or free slots (bytes) 010 = 3 data bytes or free slots (bytes) 011 = 4 data bytes or free slots (bytes) 100 = 5 data bytes or free slots (bytes) 101 = 6 data bytes or free slots (bytes) 110 = 7 data bytes or free slots (bytes) 111 = 8 data bytes or free slots (bytes)
Reserved Bits 23–0	Reserved	These bits are reserved should be set to 0.

21.6.2.17 USB Endpoint n Last Read Frame Pointer Register

The USB endpoint n last read frame pointer register stores the last read frame pointer (LRFP).

USB_EPn_LRFP — USB Endpoint n Last Read Frame Pointer Register — **0xFFFE0448+(n*0x30)**

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE										LRFP[6:0]						
										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 21-20. USB Endpoint n Last Read Frame Pointer Register Settings

Name	Description	Setting
Reserved Bits 31–7	Reserved	These bits are reserved should be set to 0.

Table 21-20. USB Endpoint n Last Read Frame Pointer Register Settings (Continued)

Name	Description	Setting
LRFP[6:0] Bits 6–0	**Last Read Frame Pointer**—This field is a FIFO-maintained pointer that indicates the start of the most recently read frame or the start of the frame currently in transmission. The LRFP can be read and written for debug purposes. For the frame retransmit function, the LRFP indicates at which point to begin retransmitting the data frame. There are no safeguards to prevent retransmitting data that has been overwritten. When FRAME is not set, this pointer has no meaning.	See description

21.6.2.18 USB Endpoint n Last Write Frame Pointer Register

The USB endpoint n last write frame pointer register stores the last write frame pointer (LWFP).

USB_EPn_LWFP — USB Endpoint n Last Write Frame Pointer Register — **0xFFFE044C+(n*0x30)**

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE										LWFP[6:0]						
										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 21-21. USB Endpoint n Last Write Frame Pointer Register Settings

Name	Description	Setting
Reserved Bits 31–7	Reserved	These bits are reserved should be set to 0.
LWFP[6:0] Bits 6–0	**Last Write Frame Pointer**—This field is a FIFO-maintained pointer that indicates the start of the last frame written into the FIFO. The LWFP can be read and written for debug purposes. For the frame retransmit function, the LWFP indicates at which point to begin retransmitting the data frame. For the frame discard function, the LWFP divides the valid data region of the FIFO (the area between the read and write pointers) into framed and unframed data. Data between the LWFP and write pointer is of an incomplete frame, while data between the read pointer and the LWFP has been received as whole frames. When FRAME is not set, this pointer has no meaning.	See description

21.6.2.19 USB Endpoint n FIFO Alarm Register

The USB endpoint n FIFO alarm register sets the trigger level for ALARM_STAT for the corresponding endpoint FIFO.

USB_EPn_FALRM USB Endpoint n FIFO Alarm Register **0xFFFE0450+(n*0x30)**

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE										ALRM[6:0]						
										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 21-22. USB Endpoint n FIFO Alarm Register Settings

Name	Description	Setting
Reserved Bits 31–7	Reserved	These bits are reserved should be set to 0.
ALRM[6:0] Bits 6–0	**Alarm [6:0]**—This field sets the trigger level for ALARM_STAT upon a FIFO alarm condition. A low level alarm reports a lack of data, while a high level alarm reports a lack of space. The integrator must decide which alarm is necessary for each application. The deassertion of ALARM_STAT is based upon two conditions: the set status of the granularity bits, GR[2:0], and whether an end of frame byte is present in the FIFO. This programmable alarm warns the system when the FIFO is almost full of data (“high” service request) or when the FIFO is almost out of data (“low” service request). This register is programmed with the upper limit for the number of bytes of data or space in the FIFO before an internal alarm is set. If the amount of data or space in the FIFO is above the indicated amount, the alarm is set in non-frame mode. In frame mode, the alarm is set if the amount of data or space in the FIFO is above the amount indicated by the ALRM setting or if there are end of frame bytes in the FIFO. See the description of frame mode operation in Section 21.6.2.16, “USB Endpoint 0 FIFO Control Register.” The alarm is cleared when there is less data or space than is programmed in the FIFO granularity bits.	0x00 = 1 data or space 0x01 = 2 data or space . . . 0x7f = 128 data or space See description for more information

21.6.2.20 USB Endpoint n FIFO Read Pointer Register

The USB endpoint n FIFO read pointer register stores the read pointer for the corresponding endpoint FIFO.

USB_EPn_FRDP — USB Endpoint n FIFO Read Pointer Register — **0xFFFE0454+(n*0x30)**

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE										RP[6:0]						
										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 21-23. USB Endpoint n FIFO Read Pointer Register Settings

Name	Description	Setting
Reserved Bits 31–7	Reserved	These bits are reserved should be set to 0.
RP[6:0] Bits 6–0	**Read Pointer**—This field is a FIFO-maintained pointer that points to the next FIFO location to be read. The physical address of this FIFO location is actually the sum of the read pointer and the FIFO base, which is provided through a port to the FIFO controller. This base address can vary, but, if chosen properly, the FIFO RAM address can be concatenated with the read pointer instead of requiring hardware for addition. The read pointer can be both read and written. This ability facilitates the debug of the FIFO controller and peripheral drivers. The current maximum size of the write pointer is 12 bits, but it can be reduced through parameterization.	See description

21.6.2.21 USB Endpoint n FIFO Write Pointer Register

The USB endpoint n FIFO write pointer register stores the write pointer for the corresponding endpoint FIFO.

USB_EPn_FWRP — USB Endpoint n FIFO Write Pointer Register — **0xFFFE0458+(n*0x30)**

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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE										WP[6:0]						
										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 21-24. USB Endpoint n FIFO Write Pointer Register Settings

Name	Description	Setting
Reserved Bits 31–7	Reserved	These bits are reserved should be set to 0.
WP[6:0] Bits 6–0	**Write Pointer**—This field is a FIFO-maintained pointer that points to the next FIFO location to be written. The physical address of this FIFO location is actually the sum of the write pointer and the FIFO base, which is provided through a port to the FIFO controller. This base address can vary, but, if chosen properly, the FIFO RAM address can be concatenated with the write pointer instead of requiring hardware for addition. The write pointer can be both read and written. This ability facilitates the debug of the FIFO controller and peripheral drivers.	See description

21.7 Programmer's Reference

This programmer's reference section explains how to program the USB module. This section documents device initialization, the processing of vendor requests, normal data path operations, interrupt services, and reset operation.

21.7.1 Device Initialization

During device initialization, user software downloads critical configuration information to the UDC module and prepares the USB device module data path for processing. This process is performed at two different times, at reset (hard reset or software reset via RST bit in the USB_ENAB register) and when the device is first connected to the USB.

At power-up time, the USB module contains no configuration information. The UDC module does not know how many endpoints it has available or how to find the descriptors. To initialize the device, download this information to the appropriate memories and configure the data path to match the intended application. The following steps are involved in the initialization process:

1. Perform a hard reset or a software reset (RST bit in USB_ENAB register).
2. Wait for the device to be properly reset (after writing to RST bit, wait for it to clear) before accessing the device registers.
3. Wait for the CFG bit in the USB_CFGSTAT register to assert before attempting to communicate with the UDC. The RST bit set will make the ENAB bit set automatically.
4. Download configuration data (EndPtBufs) to the device via the USB_DDAT register. See Section 21.7.2, "Configuration Download."
5. Program the USB interrupt mask register (USB_MASK) to enable USB general interrupts.
6. Program each endpoint's control (USB_EPn_STATCR) and interrupt (USB_EPn_MASK) registers to support the intended data transfer modes and interrupts.
7. Program endpoint type, direction, and maximum packet size in the USB_EPn_STATCR register for each endpoint.
8. Program frame mode, granularity (USB_EPn_FCTRL), alarm level (USB_EPn_FALRM), and so on for each enabled endpoint. Normally, all endpoints should be programmed with FRAME mode enabled. To ensure proper operation of the DMA request lines for frame mode endpoints, program the alarm level to the packet size or a multiple of the packet size, depending on the depth of the FIFO buffer (that is, for 8-byte packets and a 16-byte FIFO, the alarm would be programmed to 8 bytes).
9. Enable the USB for processing (USB_ENA bit in USB_CTRL).

Module initialization is a time-critical process. The USB host waits about 100 ms after powering on or for a connection event to begin enumerating devices on the bus. This device must have all of the configuration information available when the host requests it.

Once the device is enumerated, the USB host selects a specific configuration and set of interfaces on the device. Software on the device must be aware of USB configuration changes in order to maintain proper communication with the USB Host. The software retains sole responsibility for knowing the current configuration and alternate interface at any given time. The CFG_CHG interrupt in the USB_GEN_ISR register reports changes in device configuration and alternate interface settings to the software. The software is required to respond to the CFG_CHG interrupt. To prevent the state of the device from becoming out of sync with respect to the host, the device halts further traffic on the USB while this interrupt is pending.

21.7.2 Configuration Download

The configuration download process initializes five endpoint buffers (EndPtBufs) within the UDC module that defines its personality on the USB. The first EndPtBuf is reserved for the default pipe (control endpoint) and should contain the value 0x0000080000.

The endpoint buffers are 40-bit data strings per physical endpoint (pipe) that are loaded directly into the UDC module. They associate "logical" endpoint numbers in the USB software stack with hardware within the UDC. Specifically, they attach each endpoint to a USB configuration, interface, and alternate setting. Then they specify transfer type, packet size, data direction, and hardware FIFO number, among other characteristics.

Table 21-25. EndPtBuf - UDC Endpoint Buffers Format

Bit Field	Type	Description
[39:36]	EpNum	Logical endpoint number
[35:34]	Config	Configuration number. Maximum up to one configurations.
[33:32]	Interface	Interface number. Maximum up to one interface.
[31:29]	AltSetting	Alternate setting number. Maximum of four alternate settings.
[28:27]	Type	Type of endpoint: 00 - Control 01 - Reserved 10 - Bulk 11 - Interrupt
[26]	Dir	Direction of the endpoint: 0 - OUT endpoint 1 - IN endpoint
[25:16]	MaxPktSize	Maximum packet size for this endpoint: 0x08 = 8 bytes 0x10 = 16 bytes 0x20 = 32 bytes 0x40 = 64 bytes
[15:14]	TRXTYP	For this revision of the module, these bits must be set to 2'b00 for endpoint 0 and 2'b11 for all other endpoints.
[13:3]	Reserved	
[2:0]	FifoNum	This field maps the endpoint to one of the USB device module's hardware FIFOs. Multiple UDC endpoints may map to a single hardware FIFO. It is up to the software to monitor and control any data hazards related to operation in this way. The hardware FIFOs that are available are: FIFO0 (32 bytes) FIFO1 (16 bytes) FIFO2 (16 bytes) FIFO3 (128 bytes) FIFO4 (128 bytes)

Downloading the configuration data consists of the following steps:

1. Verify that the CFG bit is set in the USB_CFGSTAT register. To ensure that the UDC has been properly reset and is prepared to take data.
2. Write the endpoint buffers (EndPtBufs) to the USB_DDAT register (first byte written is EPn[39:32] and last byte written is EPn[7:0]). After writing each byte, and before performing any other operation on the USB device module, check that the BSY bit has cleared in the USB_CFGSTAT register.
3. After writing all endpoint buffer configuration bytes, check the CFG bit of USB_CFGSTAT to verify that the configuration load has completed. This bit changes from 1 to 0 after the last byte is loaded into the UDC.

21.7.3 USB Endpoint to FIFO Mapping

The USB protocol recognizes up to 31 endpoints on a USB device. The endpoint numbers available on a specific USB device vary based on the functionality present in the device along with the configuration and alternate interfaces selected at any given time. Regardless of the "logical" endpoint number programmed in the interface descriptors, some hardware must be associated with each endpoint.

This module supports a maximum of five endpoints, including endpoint 0. Each hardware endpoint consists of a single FIFO that can be programmed independently for direction, transfer type, frame mode, and low/high alarms. The hardware endpoints are mapped to the USB's "logical" endpoints in the UDC module's endpoint buffer (EndPtBufs). Endpoint FIFO0 is reserved for the default pipe (control endpoint).

Endpoint type, direction, and packet size are defined via the USB_EPn_STATCR register for each endpoint. FIFO characteristics are programmed via the USB_EPn_FCTRL and USB_EPn_FALRM registers. The user can map USB endpoints to specific hardware FIFOs when the UDC endpoint buffers (EndPtBufs) are downloaded to the device from the CPU, see Section 21.7.1, "Device Initialization." The endpoint buffer makes 16 bits available to the user for the purpose of identifying a hardware endpoint. These bits are the UDC_BufAdrPtr[15:0] bus. Depending upon the transaction selected by the UDC, this bus includes the indicator for a setup packet or endpoint-specific encoding. The fields of the UDC_BufAdrPtr[15:0] defined for this module are shown in Table 21-25—EndPtBuf[15:0].

21.7.3.1 USB Interrupt Register

If the application uses the interrupt registers, the specific interrupts to be used must be enabled. During a reset, all interrupts revert to the masked state. USB global interrupts (affecting the whole module) are programmed separately from those affecting a single endpoint.

21.7.3.2 Endpoint Registers

The characteristics of the FIFO and a number of interrupt sources may be programmed for each endpoint. The integrator must program the following registers:

- USB endpoint interrupt mask (USB_EPn_MASK)
  - Separate interrupt registers are provided for each hardware FIFO. Enable the interrupts pertaining to the application by writing a 0 to the mask bit for that interrupt.
- Endpoint FIFO controller configuration (USB_EPn_FCTRL)
  - Each FIFO is programmed for the type of data transmission used by the endpoint. Normally, all endpoints are programmed with FRAME mode.
- FIFO alarm register (USB_EPn_FALRM).
  - For bulk traffic (FRAME = 1), the alarm level is normally programmed to a multiple of the USB packet size (that is, for 8-byte packets and a 16-byte FIFO, the alarm would be programmed to 8 bytes) to allow the USB device DMA request lines to request full packets. For single buffered endpoints (packet size = 16, FIFO depth = 16 bytes), the alarm is normally programmed to 0.

21.7.3.3 Enable the Device

The last step in initializing the module is to enable it for processing via the USB_CTRL register. Most applications will set the USB_SPD (full-speed operation) and USB_AFE (analog front end enable) bits along with the USB_ENA (module enable) bit.

NOTE:

The USB_SPD bit must be set to 1 for high-speed mode (default). This

USB device does not support low-speed applications.

21.7.4 Exception Handling

Exception handling occurs in two situations: when corrupted frames must be discarded, or when an error situation occurs on the USB.

Corrupted frames are automatically discarded by the hardware. No software intervention is required.

If the device cannot respond to the host in the time allotted, the hardware automatically handles retries. No software intervention is required.

The following error situations must be dealt with by the software:

21.7.4.1 Unable to Complete Device Request

In the event that the software receives a device request that it cannot interpret, it must assert the CMD_ERROR and CMD_OVER bits back to the UDC. This results in a STALL to the endpoint in question and requires intervention from the USB host to clear. When the CMD_OVER bit clears, it indicates that the USB host has cleared the stall condition.

21.7.4.2 Aborted Device Request

When the host sends a setup packet to the device, the ACK handshake from the device can be corrupted and lost on its way back to the host. If this occurs, the host will retry the setup packet, and the device can wind up with two or more setup packets in its FIFO. There are two ways to detect this condition:

- The presence of the MDEVREQ interrupt
- The SIP bit in the USB endpoint n status/control register (USB_EPn_STATCR) is active

In either case, the presence of more than one setup packet invalidates the first one in the FIFO. When MDEVREQ is active, software must discard the first setup packet and process the second one. When SIP is active, software must discard the first setup packet, clear the device request, and wait for it to reassert.

21.7.4.3 Unable to Fill or Empty FIFO Due to Temporary Problem

If the module is unable to fill or empty a FIFO due to a temporary problem (for example, the OS did not service the FIFO in time and it overflowed), the software stalls the endpoint via the FORCE_STALL bit in the USB endpoint n status/control register (USB_EPn_STATCR). This aborts the transfer in progress and forces intervention from the USB host to clear the stall condition. The FORCE_STALL register bit automatically clears once the stall takes effect. The application software on the host must deal with the stall condition and notify the device as to how to proceed.

21.7.4.4 Catastrophic Error

In the event of a catastrophic error, the software executes a hard reset, re-initializes the USB module, and waits for the USB host to re-enumerate the bus.

21.8 Data Transfer Operations

Three types of data transfer modes exist for this module: control transfers, bulk transfers, and interrupt transfers. From the perspective of this module, the interrupt transfer type is identical to the bulk data transfer mode, and no additional hardware is supplied to support it. This section covers the transfer modes and how they work from the ground up.

All data is moved across the USB in packets. Groups of packets are combined to form data transfers. The same packet transfer mechanism applies to bulk, interrupt, and control transfers.

21.8.1 USB Packets

Data moves across the USB protocol in units called packets. Packets range in size from 0 to 1023 bytes, and depending on the transfer mode, the packet size is restricted to a small set of values. For bulk, control, and interrupt traffic, packet sizes are limited to 8, 16, 32, or 64 bytes. The packet size is programmable within the UDC module endpoint by endpoint.

The terms *packet* and *frame* are used interchangeably in this document. While USB traffic occurs in packets, the FIFO mechanism uses the term *frames* for the same blocks of data. The only difference between frames and packets from the user's standpoint is that packets may be as little as 0 bytes in length, while a frame must be at least 1 byte in length.

21.8.1.1 Short Packets

Each endpoint has a maximum packet size associated with it. In most cases, packets transferred across an endpoint will be sent at the maximum size. Since the USB does not indicate a data transfer size or include an end of transfer token, short packets are used to mark the end of data. Software indicates the end of data by writing short packets into the FIFO. Incoming short packets are indicated by examining the length of a received packet or by looking at the end of transfer and end of frame interrupts.

21.8.1.2 Sending Packets

Perform the following steps to send a packet of data to the USB host using programmed I/O.

Odd byte size packets:

1. For an N (N = odd) byte packet, write the first N-1 bytes to the FIFO data register (USB_EPn_FDAT). Data may be written as words.
2. Set the WFR bit in the USB_EPn_FCTRL register, then write the last data byte (byte N) to the USB_EPn_FDAT register.

Even byte size packets:

1. For an N (N = even) byte packet, write the first N-2 bytes to the FIFO data register (USB_EPn_FDAT). Data may be written as words.
2. Set the WFR bit in the USB_EPn_FCTRL register and then write the last data word (byte N-1 and byte N) to the USB_EPn_FDAT register.

When using the DMA controller for a Memory to I/O transfer to send packets, the sequence is as follows:

1. Program the DMA controller to write data to the endpoint FIFO and enable the DMA channel for the USB endpoint DMA.
2. When the FIFO byte count has dropped below the alarm level, a DMA request will be generated. The DMA will write data to the FIFO until the DMA request deasserts.

3. For an N byte packet, the first N-1 bytes are written to the FIFO data register (USB_EPn_FDAT) as words.
4. On the Nth byte, to signal the end of a frame, the DMA controller will signal to the USB that it is writing the final byte of the USB_EPn_FDAT register. The last byte in the transfer will get the end of frame tag (bits 15:8 for byte, bits 7:0 for word).

In a double-buffered system, the FIFO depth is twice the size of the USB packet size. Program the FIFO alarm level to be the same as a single packet. This causes the DMA request to assert whenever there is one packet's worth of data or less in the FIFO. The system may write data until the DMA request deasserts, as long as the last byte of each USB packet is tagged as the end of the frame.

21.8.1.3 Receiving Packets

Perform the following steps to receive a packet of data from the USB Host using programmed I/O.

1. Monitor the EOF interrupt for the endpoint.
2. prepare to read a complete packet of data on receiving the EOF interrupt. Clear the EOF interrupt so that software will receive notification of the next frame.
3. Read the USB_EPN_FDAT register to read in the next piece of data.
4. Read the USB_EPn_FSTAT register to get the end of frame status bits (see the following note). If the end of frame bit is set for the current transfer, then stop reading data.
5. Go to step 3.

NOTE:

The USB_EPn_FSTAT(FRAME[0:1]) field indicates valid frame byte lanes used on the bus (15:8 and 7:0). Currently, more than 1 bit may be on if there are multiple end of frame bytes on word transfers. Extra software may be required to determine the first valid end of frame marker. The value of this field is computed directly from the "frame boundary" bits that are stored in the RAM. The user is responsible for ensuring that the RAM data is valid when accessing these bits. For example, if there is only 1 byte of data in the RAM, it is marked EOF, and the user does a word (16-bit) access, then bit 1 (FRAME[1]) of FRAME[0:1] is undefined.

Perform the following steps to use the DMA Controller for an I/O to Memory transfer to receive packets.

1. Program the DMA controller to read data in units of the packet size. Enable the channel for the USB endpoint.
2. When a packet EOF is detected from the UDC to the FIFO, a DMA request will be generated, and the DMA must read the FIFO to allow for the next transaction.
3. DMA reading occurs until the request expires. Normally the expiration will occur when the FIFO no longer needs service or all the data has been read by the DMA.
4. Monitor the EOT interrupt for the endpoint. If EOT is asserted, it indicates that the data in the FIFO represents the end of a USB data transfer. The USB_EPn_STATCR register can be used to determine how many bytes remain to be read from the FIFO. This data may end on a short frame.

21.8.1.4 Programming the FIFO Controller

The FIFO controller module has two modes of operation, Frame and Non-Frame. For the typical USB application, only Frame mode is used.

In Frame mode, the FIFO controller handles automatic hardware retries of bad packets. During device initialization, the user configures the FIFOs via the USB_EPn_FCTRL register for FRAME mode. Data flow is controlled with the end of frame (EOF) and end of transfer (EOT) interrupts, or with the DMA request lines.

21.8.2 USB Transfers

Data transfers on the USB are composed of one or more packets. Instead of maintaining a transfer count, the USB host and device send groups of packets to each other in units called transfers. In a transfer, all packets are the same size, except the last one. The last packet in a transfer will be a short packet that is as small as 0 bytes when the last data byte ends on a packet boundary.

This section describes how data transfers work both from the device to the host and from the host to the device.

21.8.2.1 Data Transfers to the Host

Given an arbitrary-sized block of data to be sent to the host, break it into a number of packets sized at the maximum packet size of the target endpoint.

If the number of packets is an integer, the transfer ends on a packet boundary. A zero-length packet is required to terminate the transfer. If the number of packets is not an integer, the last packet of the transfer will be a short packet and no zero-length packet is required.

For each packet in the transfer, write the data to the USB_EPn_FDAT register. The last byte in each packet must be tagged with the end of frame marker via the USB_EPn_FCTRL register or the USB device DMA request lines. See Section 21.8.1.2, “Sending Packets.” Monitor the FIFO_LOW interrupt, EOF interrupt, or DMA transfer status register in the DMA controller to determine when the FIFO can accept another packet.

After the last byte of the transfer has been written to the FIFO, if a zero-length packet is required to terminate the transfer, then set the ZLPS bit for the endpoint.

Wait for the EOT interrupt to determine when the transfer has completed.

NOTE:

For DMA operation, a zero-length frame is not defined, so it is necessary to have the user software monitor the EOT interrupts and use them as a basis for delineating individual transfers. USB traffic flow is halted (NACKed) by the UDC until the EOT interrupt has been serviced to ensure that data from different data transfers is not combined in the FIFOs.

21.8.2.2 Data Transfers to the Device

The length of a data transfer from the host is generally not known in advance. The device receives a continuous stream of packets and uses the EOT interrupt to determine when the transfer ended.

Software on the device monitors the EOF interrupt and/or DMA transfer status register to manage packet traffic. Each time a packet is received, the device must pull the data from the FIFO. Each time an end of frame is transferred from the UDC module into the data FIFO, the EOF interrupt asserts. At the end of a complete transfer, the EOT interrupt asserts. For bulk endpoints, until the CPU has serviced the EOT interrupt, the device will NAK any further requests to that endpoint from the host. This guarantees that data from two different transfers will never be intermixed within the FIFO.

21.8.2.2.1 Control Transfers

The USB host sends commands to the device via control transfers. Control transfers can be addressed to any control endpoint. Control transfers consist of up to three distinct phases: a setup phase, an optional data phase, and a status phase. Command processing occurs in the following order:

1. Receive the SETUP packet on the control endpoint. The DEVREQ and EOF interrupts assert for that endpoint.
2. Read 8 bytes of the setup packet from the appropriate FIFO data register and decode the command.
3. Clear the EOF and DEVREQ interrupts.
4. Set up and perform the data transfer if a data transfer is implied by the command. Be careful not to send back more bytes to the USB host than were requested in the wLength field of the SETUP packet. The hardware does not check for an incorrect data phase length. The EOT interrupt will assert on completion of the data phase.
5. Assert the CMD_OVER/CMD_ERROR bits to indicate processing or error status. The UDC module will generate appropriate handshakes on the USB to implement the status phase. CMD_OVER automatically clears at the end of the status phase.
6. Wait for CMD_OVER to clear, indicating that the device request has completed.

The USB device module assumes that the UDC module will handle most of the standard requests without software intervention. User software does not need to handle any of the so-called “Chapter 9” requests listed in the USB specification, except for SYNCH_FRAME, GET_DESCRIPTOR, and SET_DESCRIPTOR. The requests are passed through endpoint 0 as a device request and must be processed by the device driver software

21.8.2.2.2 Bulk Traffic

Bulk traffic guarantees the error-free delivery of data in the order that it was sent, but the rate of transfer is not guaranteed. Bandwidth is allocated to bulk, interrupt, and control packets based on the bandwidth usage policy of the USB host.

- **BULK OUT**
  - Internal logic marks the start of the packet location in the FIFO for OUT transfers (from host to device). If an error occurs in a transfer, the logic forces the FIFO to back up to the start of the current packet and try again. No software intervention is required to handle packet retries.
  - User software reads packets from the FIFOs as they appear and stops when an EOT interrupt is received. To enable further data transfers, software services and clears the pending interrupts (EOF or EOT) and then waits for the next transfer to begin. For a Bulk Out endpoint, until the CPU has serviced the EOT interrupt, the device will NAK any further requests to that endpoint from the host. This guarantees that data from two different transfers will never be intermixed within the FIFO.
- **BULK IN**
  - Software tags the last byte in a packet to mark the end of the frame for IN transfers (from device to host). If an error occurs in a transfer, the hardware automatically forces the FIFO to back up to the start of the current packet and resend the data. User software is expected to write data to the FIFO data register in units of the associated endpoint’s maximum packet size. The end of frame may be indicated via the WFR bit in the endpoint FIFO control register (USB_EPN_FCTRL) or via the End of Frame tag signal from the DMA controller.
  - In the USB protocol, the last packet in a transfer is allowed to be short (smaller than the endpoint’s maximum packet size) or even of zero length. In order to indicate a zero-length

packet, the software should set the ZLPS bit in the associated endpoint's control register. The ZLPS bit automatically clears after the zero-length packet has been successfully sent to the host.

— The EOT interrupt asserts to indicate that the last packet of the IN transfer has completed. Software should clear any pending interrupts (EOT and EOF) to begin the next data transfer.

21.8.2.2.3 Interrupt Traffic

Interrupt endpoints are a special case of bulk traffic. Interrupt endpoints are serviced on a periodic basis by the USB host. Interrupt endpoints are guaranteed to transfer one packet per polling interval, so an endpoint with an 8-byte packet size that is serviced every 2 ms moves at 16 kbps across the USB.

The only difference between interrupt transfers and bulk transfers from the device standpoint is that every time an interrupt packet is transferred, regardless of size, the EOT interrupt asserts. The device driver software must service this interrupt packet before the next interrupt servicing interval to prevent the device from NAKing the poll.

Device driver software must ensure that the interrupt endpoint polling interval is longer than the device's interrupt service latency.

21.9 Reset Operation

This module includes four reset modes: hard reset, software reset, UDC reset, and USB reset signalling.

The UDC reset allows software to force a hard reset of the UDC module only, leaving all register bits in the front-end logic intact. A UDC reset will normally only be used as a debug option, but may also be used in the event of a connect/disconnect bus event.

21.9.1 Hard Reset

A hard reset is generated on the module's bus interface by the reset module of the MC68SZ328. It resets all storage elements in both the front-end logic and in the UDC module and issues a UDC reset. A hard reset requires that MCU PLL and USB PLL be locked.

21.9.2 USB Software Reset

The USB device allows the reset of all the storage elements in both the front-end logic and in the UDC module via the RST bit in the USB_ENAB register. On initial power up, the user issues a software reset. This enables the module and resets the internal logic. Note that both MCU PLL and USB PLL should be locked before the software reset is issued.

21.9.3 UDC Reset

According to the specification for the UDC module, the UDC must be reset whenever a connect/disconnect occurs on the USB. Any time the device is plugged in or unplugged from the USB, the software must initiate either a hard reset or a UDC reset (see the UDC_RST bit in the USB_CTRL register) to ensure that the module can properly communicate with the USB host. Reset signalling is discussed in Chapter 7 of the USB Specification. A UDC reset may invalidate data remaining in the data FIFOs. Depending on the application, software may need to flush the data FIFOs before proceeding.

21.9.4 USB Reset Signalling

Reset signalling can occur on the USB for a number of reasons. When the device receives reset signalling, it means that the host is preparing to re-enumerate the bus. All transactions in progress must be invalidated, and the device software should prepare to receive configuration changes.

Because the device may have valid but as yet unread data remaining in the FIFOs when USB reset signalling occurs, the hardware does not flush the FIFOs. When the device software receives reset signalling, it should first finish reading any unread data from the FIFOs and then execute FIFO flush operations on all of the FIFOs. This guarantees that the data path is empty and ready for new data transfer operations when reset signalling and re-enumeration are complete.

Chapter 22
I²C

This chapter describes the MC68SZ328 I²C module, including I²C protocol, clock synchronization, and the registers in the I²C programing model.

22.1 Overview

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 system is a true multiple-master bus including collision detection and arbitration that prevents data corruption if multiple masters attempt to control the bus simultaneously. This feature allows for 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.

22.2 I²C Features

The I²C module has the following key features:

- Complies with I²C bus standard (supports standard-mode and fast-mode)
- Support for 7-bit address
- Support for 3.0 V devices (Up to 3.3 V)
- 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 a block diagram of the I²C module and illustrates the relationships of the I²C registers listed below. These registers and their settings are described in detail in Section 22.5, “Programming Model,” on page 22-5.

- I2C address register (IADR)
- I2C frequency divider register (IFDR)
- I2C control register (I2CR)
- I2C status register (I2SR)
- I2C data I/O register (I2DR)
- I2C byte counter register (IBCR)

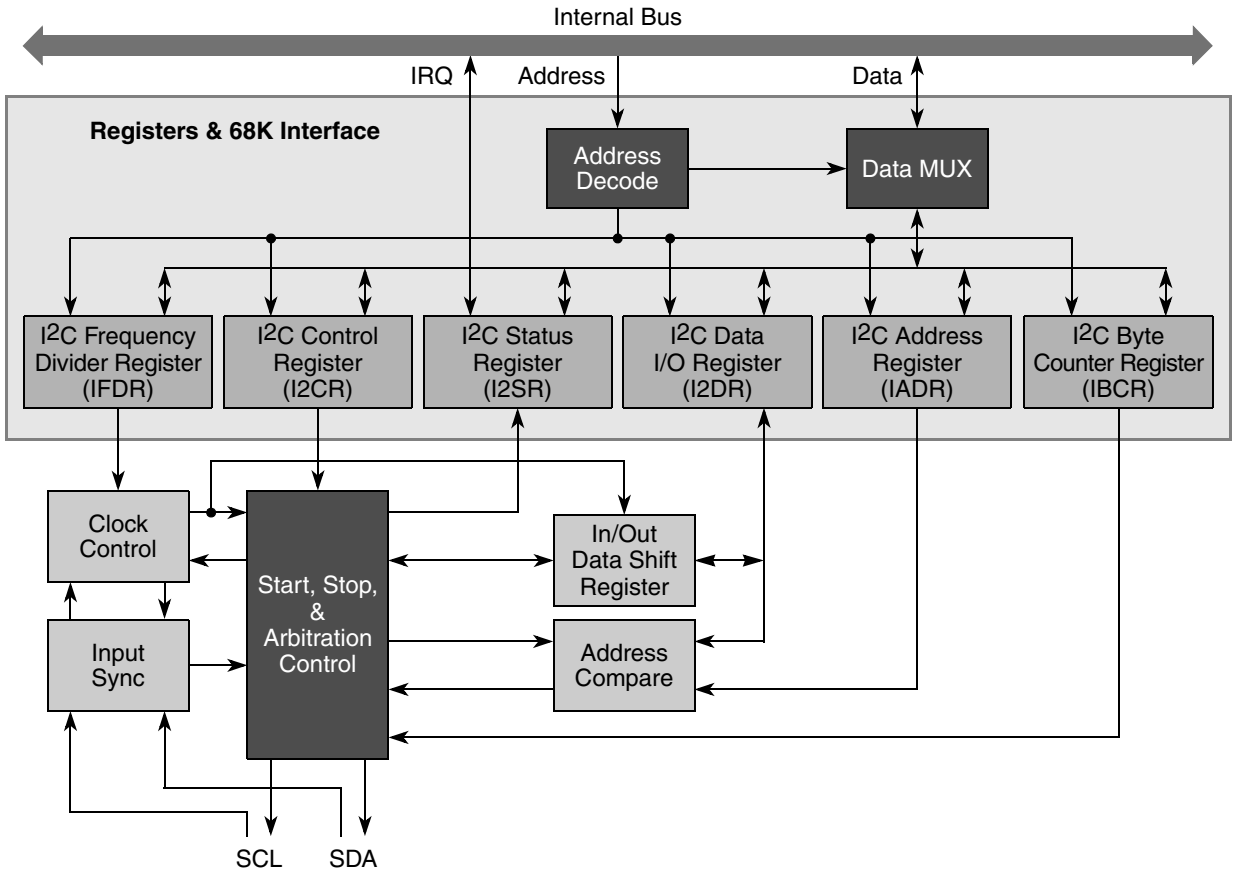


Figure 22-1. I2C Module Block Diagram

22.3 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.

The default state of the I²C is the same as a slave receiver after reset. When not programmed to be a master or responding to a slave transmit address, the I²C module always returns to the default state of the slave receiver. Exceptions to this are described in Section 22.6.1, “Initialization Sequence,” on page 22-12.

NOTE:

The I²C module is designed to be compatible with *The* I²C-*Bus Specification*, Rev. 2.1, (Philips Semiconductor, 2000). For detailed information on system configuration, protocol, and restrictions see the Philips’ I²C standard.

22.4 I²C Protocol

I²C protocol uses standard communication processes. This section describes and illustrates the standard 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 as shown in Figure 22-2. 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 will be several bytes long) and awakens all slaves.

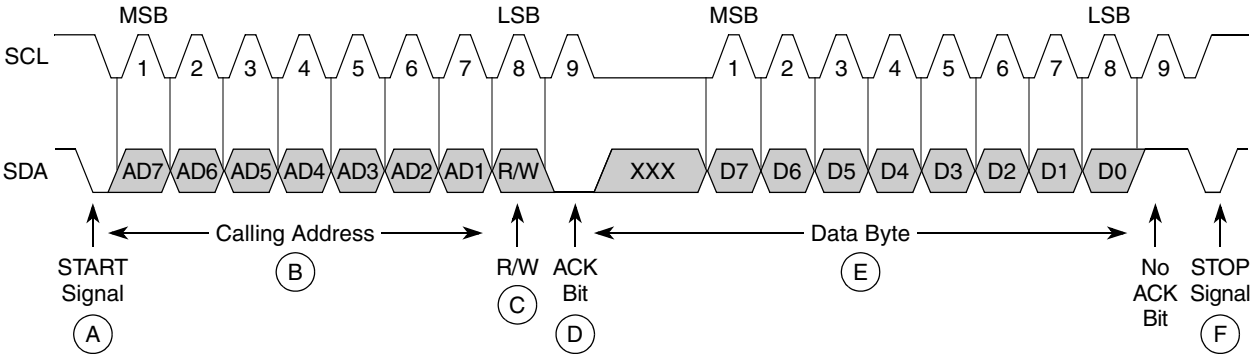


Figure 22-2. I²C Standard Communication Protocol

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—that is, it cannot be master and slave at the same time. The slave with an address that matches the address sent by the master by returns an acknowledge bit by pulling SDA low at the ninth clock (D).
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 22-2 shows. SCL is pulsed once for each data bit, with the most significant

bit (MSB) sent first. The receiving device acknowledges each byte by pulling SDA low at the ninth clock, therefore, a data byte transfer takes nine clock pulses.

If the slave receiver does not acknowledge the master, it leaves SDA high. The master then generates a STOP signal to abort the data transfer or generate a START signal (repeated start, shown in Figure 22-3) 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 1 (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.

   Instead of signalling a STOP, the master can repeat the START signal, followed by a calling command, indicated in Figure 22-3. A repeated START signal occurs when a START signal is generated without first generating a STOP signal to end the communication.

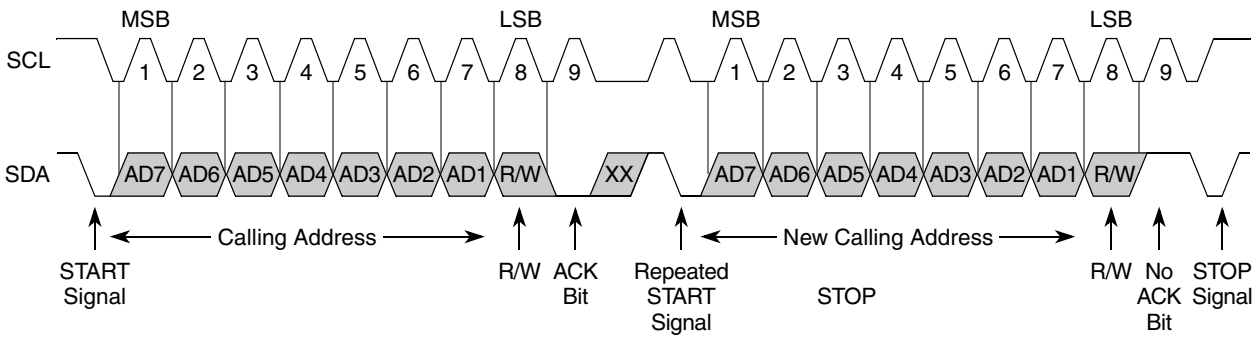


Figure 22-3. Repeated START

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.

22.4.1 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 1 while another sends logic 0—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 the IAL bit in the I2SR register to indicate loss of arbitration.

22.4.2 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 the 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 as illustrated in Figure 22-4 on page 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.

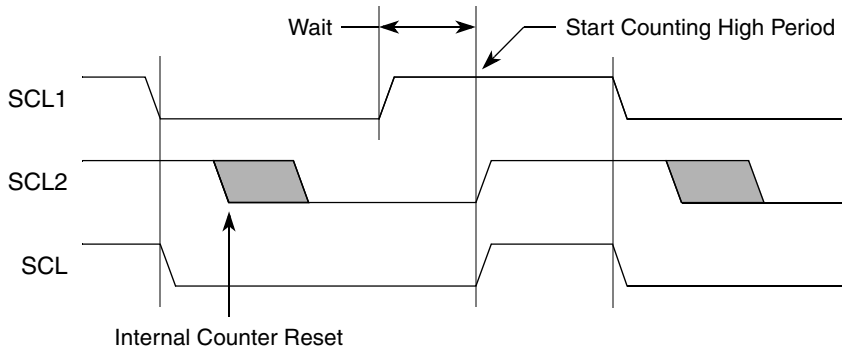


Figure 22-4. Synchronized Clock SCL

22.4.3 Handshaking

The clock synchronization mechanism can be used as a handshake in data transfers. Slave devices can hold SCL low after completing a 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.

22.4.4 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.

22.5 Programming Model

This section describes the detailed programming information about the I^2C module. Table 22-1 is a summary of the registers and the addresses used in the I^2C module.

Table 22-1. I^2C Module Memory Map

Address	Register	Width	Description	Reset Value
0x(FF)FFF800	IADR	8	I^2C Address Register	0x00
0x(FF)FFF804	IFDR	8	I^2C Frequency Divider Register	0x00
0x(FF)FFF808	I2CR	8	I^2C Control Register	0x00
0x(FF)FFF80C	I2SR	8	I^2C Status Register	0x81
0x(FF)FFF810	I2DR	8	I^2C Data I/O Register	0x00

Table 22-1. I2C Module Memory Map (Continued)

Address	Register	Width	Description	Reset Value
0x(FF)FFF814	IBCR	8	I2C Byte Counter Register	0x00

22.5.1 I2C Address Register (IADR)

The I2C address register (IADR) holds the address to which the I2C responds when addressed as a slave. Note that it is not the address sent on the bus during the address transfer. The settings for the bit positions of IADR are shown in Table 22-2.2

IADR I2C Address Register **0x(FF)FFF800**

	BIT 7	6	5	4	3	2	1	BIT 0
	ADR							
TYPE	rw	rw	rw	rw	rw	rw	rw	
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 22-2. I2C Address Register Description

Name	Description	Settings
ADR Bits 7–1	**Address**—Slave address. This field 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.	See description.
Reserved Bit 0	Reserved	This bit is reserved and should be set to 0.

22.5.2 I²C Frequency Divider Register (IFDR)

The I²C frequency divider register (IFDR), provides a programmable prescaler to configure the clock for bit-rate selection.

IFDR I²C Frequency Divider Register **0x(FF)FFF804**

	BIT 7	6	5	4	3	2	1	BIT 0
			IC					
TYPE			rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 22-3. I²C Frequency Divider Register Description

Name	Description	Settings
Reserved Bits 7–6	Reserved	These bits are reserved and should be set to 0.
IC Bits 5–0	I²C **Clock Rate**—Used to prescale the clock for bit-rate selection. Due to the potential slow rise and fall times of SCL and SDA, bus signals are sampled at the prescaler frequency. The serial bit clock frequency is equal to SYSCLK divided by the divider shown in the Table 22-4. Note that the IC value can be changed at any point in a program. **Note:** As a slave with an unknown SCL clock speed, choose a small prescale value (for example, 0x00) that benefits the I²C operation. Choose a large prescale value for slow transmission and noisy environments.	Table 22-4 provides the I²C clock rate settings.

Table 22-4. I²C Clock Rate Divider Settings

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

Table 22-4. I2C Clock Rate Divider Settings (Continued)

IC	Divider	IC	Divider	IC	Divider	IC	Divider
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 I2C control register (I2CR) is used to enable the I2C module and the I2C interrupt. This register also contains bits that govern operation as a slave or a master. The settings for I2CR are described in Table 22-5.

I2CR I2C Control Register **0x(FF)FFF808**

	BIT 7	6	5	4	3	2	1	BIT 0
	IEN	IIEN	MSTA	MTX	TXAK	RSTA		
TYPE	rw	rw	rw	rw	rw	w		
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 22-5. I2C Control Register Description

Name	Description	Settings
IEN Bit 7	I2C **Enable**—Controls the software reset of the entire I2C module. If the module is enabled in the middle of a byte transfer, slave mode ignores the current bus transfer and starts operating whenever a subsequent start condition is detected. Master mode is not aware that the bus is busy, therefore, if a start cycle is initiated, the current bus cycle can become corrupted ultimately causing either the current bus master or the I2C module to lose arbitration, after which bus operation returns to normal.	0 = The module is disabled, but registers can still be accessed. 1 = The module is enabled. This bit must be set before any other I2CR and I2DR bits have any effect.
IIEN Bit 6	I2C **Interrupt Enable**—This bit enables the I2C interrupts.	0 = The module interrupts are disabled, but the currently pending interrupt condition is not cleared. 1 = The module interrupts are enabled. An I2C interrupt occurs if I2CR[IIF] is also set.

Table 22-5. I²C Control Register Description (Continued)

Name	Description	Settings
MSTA Bit 5	**Master/Slave Mode Select**—If the master loses arbitration, MSTA is cleared without generating a STOP signal.	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**—This bit selects the direction of master and slave transfers. **Note:** During transmit mode, 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 for the master and 0 for the slave.	0 = Receive. 1 = Transmit.
TXAK Bit 3	**Transmit Acknowledge Enable**—This bit specifies the value driven into SDA during acknowledge cycles for both master and slave receivers. Note that writing TXAK applies only when the I²C bus is a receiver, not a transmitter.	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.
RSTA Bit 2	**Repeat Start**—This bit is 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 be set to 0.

22.5.4 I²C Status Register (I2SR)

The I²C status register (I2SR) contains bits that indicate transaction direction and status. The settings for the registers are described in Table 22-6.

I2SR — I²C Status Register — **0x(FF)FFF80C**

	BIT 7	6	5	4	3	2	1	BIT 0
	ICF	IAAS	IBB	IAL	IWDR	SRW	IIF	RXAK
TYPE	r	rw	r	rw	r	r	rw	r
RESET	1	0	0	0	0	0	0	1
	0x81							

Table 22-6. I²C Status Register Description

Name	Description	Settings
ICF Bit 7	**Data Transferring**—While one byte of data is being transferred, ICF is cleared.	0 = Transfer in progress. 1 = Transfer complete. Set by the falling edge of the ninth clock of a byte transfer.

Table 22-6. I2C Status Register Description (Continued)

Name	Description	Settings
IAAS Bit 6	I2C **Addressed As Slave**—Interrupted the CPU 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**—This bit indicates the status of the bus.	0 = Bus is idle. If a STOP is detected, IBB is cleared. 1 = Bus is busy. When a START is detected, IBB is set.
IAL Bit 4	**Arbitration Lost**—This bit is set by hardware in the following circumstances. The IAL bit must be cleared by software by writing a 1to 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. • 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.	See description.
IWDR Bit 3	**Write Data Ready**	Data is written into I2DR only when this bit is set.
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 must be cleared by software by writing a 1 to it in the interrupt routine. This bit is 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.	0 = No I2C interrupt pending. 1 = An interrupt is pending, which causes a processor interrupt request (if IIEN=1).
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 I²C data I/O register (I2DR) allows a read to occur and initiates receive for the next data byte. In slave mode, the same function is available after it is addressed.

I2DR — I²C Data I/O Register — **0x(FF)FFF810**

	BIT 7	6	5	4	3	2	1	BIT 0
FIELD	DATA							
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 22-7. I²C Data I/O Register Description

Name	Description	Settings
DATA Bits 7–0	I²C **Data**—Next I²C data to be sent/last I²C data received.	See description.

22.5.6 I²C Byte Counter Register (IBCR)

The register indicates the number of byte successfully transferred. Start/Re-start signal on I²C or writing to this register will reset its value to 0. Arbitration loss will not reset this counter but the byte results in arbitration loss will not be counted.

IBCR — I²C Byte Counter Register — **0x(FF)FFF814**

	BIT 7	6	5	4	3	2	1	BIT 0
FIELD	BC							
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 22-8. I²C Byte Counter Register Description

Name	Description	Settings
BC Bits 7–0	I²C **Byte Counter**—Number of bytes transmitted/received	See description.

22.6 I²C Programming Information

This section describes programming sequences for I²C, including initialization, signaling START, post-transfer software response, signalling STOP, and generating a repeated START. Figure 22-5, “Flow-Chart of Typical I²C Interrupt Routine,” on page 22-14, illustrates the typical process.

22.6.1 Initialization Sequence

Before the I^2C module can transfer serial data, the registers must be initialized, as follows:

1. Set the IC bit in the IFDR register to obtain SCL frequency from the system bus clock. See Section 22.5.2, "I^2C Frequency Divider Register (IFDR)," on page 22-7.
2. Update the IADR to define its slave address. See Section 22.5.1, "I^2C Address Register (IADR)," on page 22-6.
3. Set the IEN bit in the I2CR register to enable the I^2C bus interface system. See Section 22.5.3, "I^2C Control Register (I2CR)," on page 22-8.
4. Modify the I2CR to select master/slave mode, transmit/receive mode, and whether to enable interrupt.

Before enabling the I ^{2}C, ensure that there is not another external I ^{2}C communication in process.

The following instructions are an example of code for an initialization sequence:

Code Listing 22-1. I^2C Instructions

```
IICBUSFREE btst #5,I2SR          check if I2C bus is free
bne.s IICBUSFREE
mov.b #$F0,I2CR                  set master Tx mode, interrupt enable
IICWRRDY btst #3,I2SR            wait for data ready
beq.s IICWRRDY
mov.b CALLING_ADDR,I2DR
```

22.6.2 Generation of START

After completing the initialization procedure, serial data can be transmitted by selecting the master transmitter mode. On a multiple-master bus system, the IBB bit in the I2SR register 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 least significant bit (LSB) indicates the transfer direction after IWDR bit in the I2SR register is set. The bus free time (the time between a STOP condition and the following START condition) is built into the hardware that generates the START cycle.

22.6.3 Post-Transfer Software Response

The ICF bit in the I2SR register is set upon completion of sending or receiving a byte, which indicates one byte communication is finished. The IIF bit in the I2SR register is also set. An interrupt is generated if the interrupt function is enabled by setting the IIEN bit in the I2CR register. Software must first clear IIF in the interrupt routine. ICF is cleared by reading from I2DR in receive mode or by writing to I2DR in transmit mode: the next communication byte then starts.

Software services the I^2C 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, then the MTX bit in the I2CR register should be toggled.

During slave-mode address cycles (I2SR[IAAS] = 1), the SRW bit in the I2SR register is read to determine the direction of the next transfer and 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 occurs 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 the TXAK bit in the I2CR register 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 before the data register (I2DR) is accessed.

22.6.6 Slave Mode

In the slave interrupt service routine, the module addressed as a slave bit (IAAS) tests to see 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 SRW bit in the I2SR register. 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, the RXAK bit in the I2SR register 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 attempt 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, however 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 IAL bit of the I2SR = 1 and the MSTA bit of the I2CR register = 0.

If a device that is not a master attempts to transmit or generate 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.

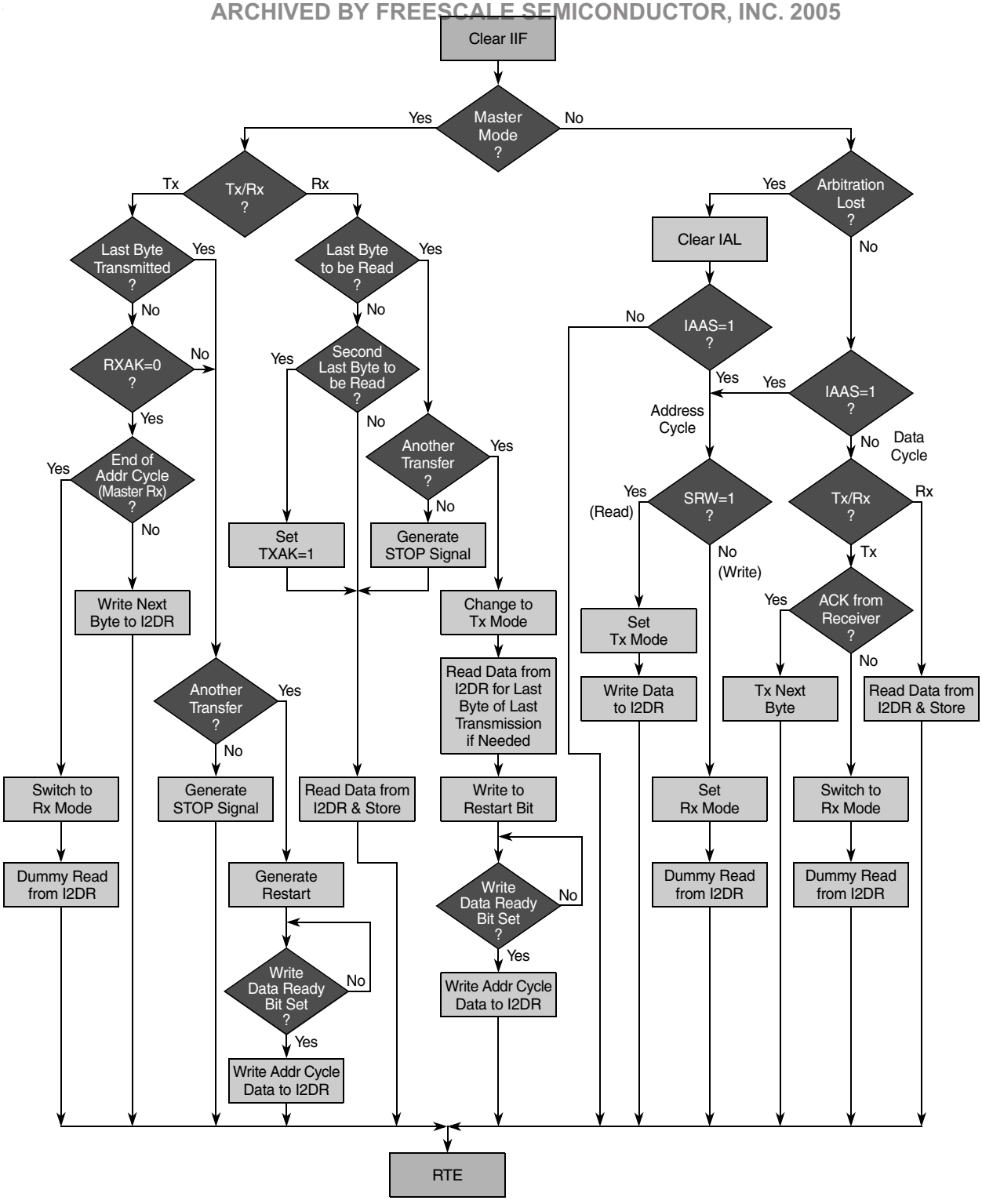


Figure 22-5. Flow-Chart of Typical I²C Interrupt Routine

Chapter 23
Bootstrap Mode

This chapter describes the operation and programming information of the bootstrap mode of the MC68SZ328. The bootstrap mode is designed to allow the initialization of a target system and the ability to download programs or data to the target system RAM using either the UART 1 or the UART 2 controller. See Chapter 19, “Universal Asynchronous Receiver/Transmitter 1 and 2,” for information on operating and programming the UART controllers. Once a program is downloaded to the MC68SZ328, it can be executed, providing a simple debugging environment for failure analysis and a channel to update programs stored in flash memory. The features of bootstrap mode are as follows:

- Allows system initialization and the ability to download both programs and data to system memory using UART 1 or UART 2
- Accepts execution commands to run programs stored in system memory
- Provides a 32-byte instruction buffer for 68000 instruction storage and execution

23.1 Bootstrap Mode Operation

In bootstrap mode, the MC68SZ328’s UART 1 and UART 2 controllers are initialized to 19,200 baud, no parity, 8-bit character, and 1 stop bit, and then they are ready to accept bootstrap data download. The first character received is used to instruct the MC68SZ328 to determine which UART port is being used for bootstrapping. The first character can be any value and is not part of the program or data being downloaded. Downloading the data or program requires the user to convert the code to a bootstrap format file, which is a text file that contains bootstrap records. A DOS-executable program, STOB.EXE, can be downloaded from the DragonBall Web site (http://www.Motorola.com/DragonBall) to convert an S-record file to a bootstrap format file.

Before a program is downloaded to system memory, the MC68SZ328’s internal registers should be set to initialize the target system. Since internal registers are treated as a type of memory, each of them can be initialized by issuing a bootstrap record.

The bootstrap design provides a 32-byte instruction buffer to which 68000 instructions may be downloaded. This feature enables the 68000 instructions to execute even if the memory systems are disabled or the MC68SZ328 is operating in a CPU standalone system. The instruction buffer starts at 0xFFFFD2. Whether initializing internal registers, downloading a program to system RAM, or issuing a core instruction, bootstrap mode will only accept bootstrap record transfers that are made using the UART. The type of bootstrap record (b-records) received determines the action of the system.

23.1.1 Entering Bootstrap Mode

Bootstrap mode is one of the three operation modes (normal, emulation, and bootstrap) of the MC68SZ328. Of the three modes, bootstrap has the highest priority. To enter bootstrap mode, the BST2 /BST0 signal must be driven low, the BST1 signal must be driven high, and a system reset must be performed. After reset, bootstrap reset vectors are internally generated for reset vector fetch cycles.

Figure 23-1 illustrates the bootstrap mode reset vector fetch timing. These two-long-word reset vectors are loaded to the stack pointer and program counter of the CPU, and then the built-in bootstrap program runs and accepts data transfers.

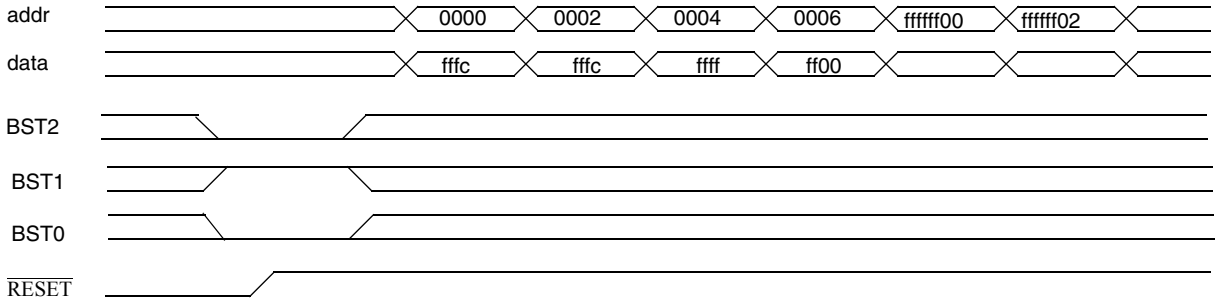


Figure 23-1. Bootstrap Mode Reset Timing Diagram

23.1.2 Bootstrap Record Types and Formats

There are two types of b-records, the execution b-record and the data b-record. Bootstrap mode data transfers will only accept bootstrap records whose format is shown in Table 23-1. Both types of b-records use this format. The two major attributes of b-records are that they are in uppercase and end with a carriage return.

Table 23-1. Bootstrap Record Format

Field	Address	Count	Data
Character	4-Byte	1-Byte	N-Byte

23.1.2.1 Execution B-Record

The execution b-record tells the bootloader to run a program starting at the location specified by the address field of the b-record. The count field for an execution b-record always contains 0x00, and no data is in the data field.

An execution b-record is used in two situations.

Situation 1:

After a program is downloaded to system RAM, an execution b-record initiates program execution. The address field of the b-record will be the start address of the program.

Situation 2:

When loading a 68000 instruction into the instruction buffer and filling the remainder of the unused buffer space with `nop--$4e71`, an execution b-record executes the 68000 instruction that is stored in IBUFF and returns to bootloader mode. The address field of the b-record will be the start address of IBUFF.

23.1.2.2 Data B-Record

The data b-record contains gives the instructions for the data to be transferred. The 4-byte address field indicates where the data will be stored, and this address could be any MC68SZ328 internal register location. The count field of the record contains the number of data bytes to be transferred. The data field contains the data to be transferred.

23.1.3 Setting Up the RS-232 Terminal

To set up communication between your target system and the PC, set the communication specifications to 19,200 bps, no parity, 8-bit, and 1 stop bit. It is permissible to pause after each line (b-record) is transferred to ensure that each transferred ASCII character is echoed.

After the hardware is set up, the system is powered up, and bootstrap mode is entered, sending any ASCII character to the target system will initiate the link. The bootloader automatically determines which UART port is being used for bootstrap by sensing the receive FIFO in each UART. The first UART to have data is selected. Next, the bootloader adjusts the baud rate to 19,200 bps. If the link is successful, the bootloader returns a unique character (@) as an acknowledgement. In addition, the bootloader echoes to the target system the same ASCII character that the target system initially transmitted.

NOTE:

The TXD2 pin of UART 2 is not enabled by default. Therefore, no character is echoed before bit 5 (TXD2) of the Port J select register is cleared. To re-enable the TXD2 pin in bootstrap mode, download the following b-record: “FFFFF43B01CF.”

23.1.4 Changing the Speed of Communication

The communication baud rate may be changed after 19,200 bps is initially used to set up the RS-232 terminal. Simply issue a b-record to reinitialize the baud control register of the UART controller, which is described in Section 19.5.2, “UART 1 Baud Control Register,” on page 19-12. For example, if the system uses a 32.768 kHz external crystal, the baud control register is initialized to 0x0126 after 19,200 bps is set up, assuming that the system clock is 16.58 MHz. Changing the baud control register from 0x0126 to 0x0026 will switch the baud rate from 19,200 bps to 38,400 bps by issuing a b-record. After the last character of this b-record is sent (0), the echo of this last character will be in the new speed (38,400 bps). At this time, the host speed must immediately be adjusted to 38,400 bps.

The baud control register is a 2-byte register, and bootstrap mode data transfers are byte-sized write cycles. Therefore, changing both bytes of the baud control register requires two steps, and each byte change must be issued at the standard communication speed for the host to set up new communication. For example, to change the speed from 19,200 bps to 115,200 bps, follow these steps:

1. Issue the b-record “FFFFF9020100” to change the baud control register from 0x0126 to 0x0026, and the new speed changes to 38,400 bps. Next, change the host speed to 38,400 bps to synchronize with the target system.
2. Issue another b-record to change the baud control register from 0x0026 to 0x0038 of the final 115,200 bps speed, and readjust the host speed to 115,200 bps.

23.1.5 Application Programming Example

The code shown in Code Example 23-1 on page 23-4 can be used to calculate a CRC value. The example demonstrates how assembly code is assembled and downloaded to system RAM.

Code Example 23-1. Application Programming Example

```
            section code
START:
copy clr.l   d1                ;d1 is used to count the number of words copied.
             clr.w   d2             ;d2 is used to count the number of words copied.
nextwd move.w  (a0,d2),d6
             move.w  d6,(a1)+
             add.l   #2,d1          ;Count the number of words copied.
             add.w   #2,d2          ;Count the number of words copied.
             cmpi.w  #16,d2
             blt     nextwd         ;until the whole section has been copied.
             clr.w   d2
             cmp.l   d0,d1          ;Copy the next word (nextwd)
             blt     nextwd         ;until the whole section has been copied.

crc    clr.l D0
lp2    add.l (A0)+,D0
             cmp.l A0,A1
             bpl.b lp2
             nop
             rts
```

After assembling and linking the program in Example 23-1, generate the following s-record file.

```
S0030000FC
S1134000428142423C30200032C6548154420C4228
S113401000106DF04242B2806DEA4280D098B3C87D
S10940206AFA4E714E75B0
S9030000FC
```

Run the DOS program STOB.EXE to convert the preceding s-records to bootstrap format.

```
000040001O428142423C30200032C6548154420C42
000040101000106DF04242B2806DEA4280D098B3C8
00004020066AFA4E714E75
```

Download the preceding b-record file to the target system using the UART port in bootstrap mode. Since this b-record file will be loaded into system RAM, initialize the system by downloading an init b-record file.

To run the preceding program after it is downloaded to RAM, issue an execution b-record “0000400000”, where 00004000 is the start address of the program and the last two zeros identify the record as an execution b-record and not a data record.

To resume bootstrap mode operation after running a program, make the last instruction in the application program a `jmp $FFFFFF6C` to start receiving a new b-record.

Any b-record may be entered in a RS-232 terminal environment, but when a key is pressed, the character produced by the keystroke is sent to the bootloader to be assembled. Although the backspace capability is not implemented, the b-record can be terminated at any time by pressing the ENTER key. As long as a program execution b-record is not issued, the MC68SZ328 will remain in bootstrap mode.

23.2 Bootloader Flowchart

The following flowchart illustrates how the bootloader program operates inside the MC68SZ328. The bootloader starts when the MC68SZ328 enters bootstrap mode.

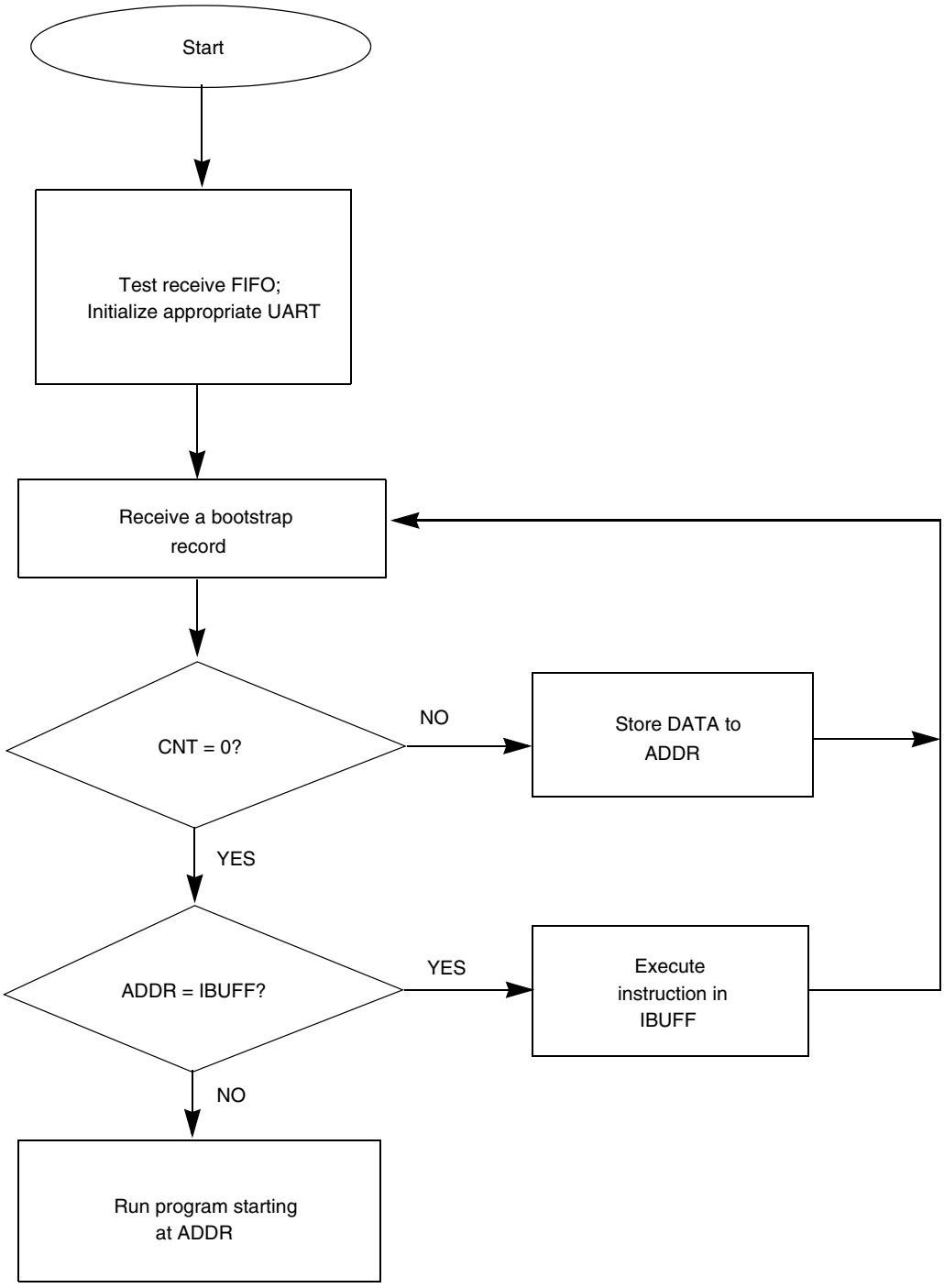


Figure 23-2. Bootloader Program Operation

23.3 Special Notes

The following information may be useful when the MC68SZ328 is in bootstrap mode.

- A b-record is a string of uppercase hex characters with optional comments that follow.
- Comments in a executable b-record or b-record data file must not contain any word or symbol that is longer than nine characters. However, the following characters can be used in a string of any length (all of these have an ASCII code value that is less than 0x30):
  - space
  - ! (exclamation point)
  - “ (quotation mark)
  - # (number sign)
  - $ (dollar sign)
  - % (percentage symbol)
  - & (ampersand)
  - ((opening parenthesis)
  -) (closing parenthesis)
  - * (asterisk)
  - + (plus sign)
  - - (minus sign)
  - . (period)
  - / (forward slash)
- The bootloader program echoes all characters being received, but only those having an ASCII code value greater than or equal to 0x30 are kept for b-record assembling. Sending a character that is not a b-record (ASCII code value < 0x30) will force the bootloader to start a new b-record.
- The D[6:0] and A0 registers are used by the bootloader program. Writing to these registers may corrupt the bootloader program.
- Visit the DragonBall Web site at http://www.Motorola.com/DragonBall for bootstrap utility programs.

Chapter 24
In-Circuit Emulation

This chapter describes the in-circuit emulation (ICE) module of the MC68SZ328 and provides detailed information about its operation and registers. The ICE module is designed to support low-cost emulator designs using the MC68SZ328 microprocessor. Using four interface signals that are extended to external pins, the ICE module has access to the 68000 CPU resources, with minimal restrictions. The features of the in-circuit emulation module are as follows:

- Dedicated chip-select for emulator debug monitor (using the $\overline{\text{EMUCS}}$ signal)
- Dedicated level 7 interrupt for in-circuit emulation
- One address signal comparator and one control signal comparator with masking to support single or multiple hardware execution and bus breakpoints
- One breakpoint instruction insertion unit

Figure 24-1 illustrates the block diagram of the in-circuit emulation module.

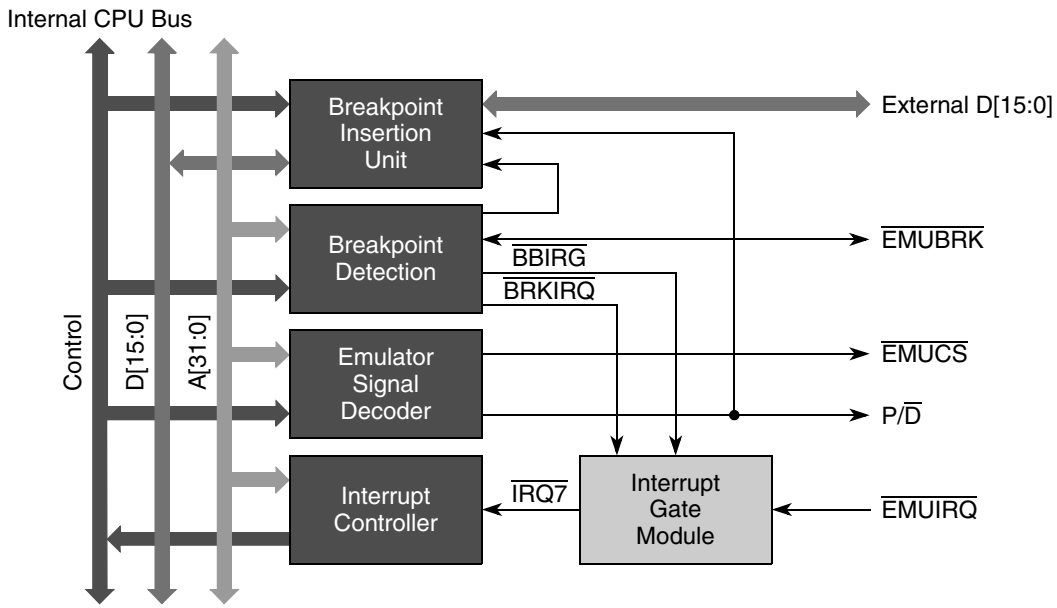


Figure 24-1. In-Circuit Emulation Module Block Diagram

24.1 ICE Operation

The in-circuit emulation module's operation consists of the following tasks:

- Entering emulation mode
- Detecting breakpoints
- Using the signal decoder
- Using the interrupt gate module
- Using the A-line insertion unit

24.1.1 Entering Emulation Mode

The reset module will latch the state of the BST2~0 signal on the rising edge of the $\overline{\text{RESET}}$ signal. To put the MC68SZ328 in emulation mode, the BST2~0 must be set to 001. For emulation mode, the $\overline{\text{CSA0}}$ signal is not asserted for reset fetch, because it is not in normal operation mode. The in-circuit emulation module internally generates a reset vector to the processor on reset vector fetch cycles.

This hard-coded reset vector is PC = 0xFFFC0020 and SSP = 0xFFFCFFFC, which means that the monitor or boot code must start at 0xFFFC0020. The $\overline{\text{EMUCS}}$ signal is designed to cover system memory space from 0xFFFC0000 to 0xFFFDFFFF, and it is an 8-bit data bus width chip-select signal. If $\overline{\text{EMUIRQ}}$ is logic high during system reset, the in-circuit emulation module is disabled and the MC68SZ328 begins another operation mode.

24.1.2 Detecting Breakpoints

The execution breakpoint detector has one 32-bit address comparator and one control signal comparator. When the in-circuit emulation module is configured to operate in single breakpoint mode, in which $\overline{\text{EMUBRK}}$ is an output, the generation of the $\overline{\text{EMUBRK}}$ signal is internally qualified by the $\overline{\text{AS}}$ signal. The active time for this signal will vary, depending on the setting and width (wait state) of the bus cycle. The $\overline{\text{EMUBRK}}$ signal is asserted throughout the address matched cycle. When the in-circuit emulation module is in multiple breakpoint mode, $\overline{\text{EMUBRK}}$ is an input that is asserted by the external address comparator. The external address comparator will compare the lower address while the internal comparator, with masking, compares the hidden address signals. The $\overline{\text{EMUBRK}}$ signal, together with the internal compare result, generates the match signal to the breakpoint insertion unit.

Because the processor does not have built-in emulation support, the execution breakpoint is implemented external to the core and will use the A-line instruction and level 7 interrupt. To accurately catch the execution breakpoint, the in-circuit emulation module inserts the 0xA0000 opcode at the location where a breakpoint is set. For more information regarding the insertion mechanism, refer to Section 24.1.5, "Using the A-Line Insertion Unit." When the 0xA000 opcode is being executed, which means the breakpoint is reached, an exception vector fetch for an A-line exception will occur. At this point, $\overline{\text{EMUBRK}}$ is asserted to stop the process and switch control to the emulation monitor (selected by the $\overline{\text{EMUCS}}$ signal).

An exception vector fetch for an A-line exception consists of two consecutive word reads at addresses 0x28 and 0x2A. The A-line exception vector fetch will cause an $\overline{\text{IRQ7}}$ assertion if a breakpoint is activated in emulation mode. However, normal memory reads to these two words will not cause an $\overline{\text{IRQ7}}$ assertion.

24.1.2.1 Execution Breakpoints vs. Bus Breakpoints

An execution breakpoint is a breakpoint at which the current program execution stops and gives control to the monitor. To set up a single execution breakpoint, initialize the compare and mask registers; set the SB, PBEN, and CEN bits in the in-circuit emulation module control register (ICEMCR); and then clear the

BBIEN and HMDIS bits in the same register. For multiple execution breakpoint mode, clear the SB bit. A bus breakpoint is a breakpoint at which the current program execution stops when there is a memory write or read at a defined address location. To enter single bus breakpoint mode, set the SB, BBIEN, and CEN bits, and then clear the PBEN and HMDIS bits. For multiple bus breakpoint mode, clear the SB bit.

24.1.3 Using the Signal Decoder

The emulator requires a local resident debug monitor to be mapped at a specific location that is transparent to the user. This monitor resides in the dedicated memory space 0xFFFC0000–0xFFFDFFFF (128K), which is selected by the $\overline{\text{EMUCS}}$ signal with internal $\overline{\text{DTACK}}$ generation. In emulation mode, the respected memory map is reserved for the emulator, and memory should not be assigned to this area. The port size of this monitor is 8-bit and the data bus is D[15:8].

The P/$\overline{\text{D}}$ signal indicates the characteristics of the current cycle. A 0 indicates a data access cycle (FC[2:0] = x01), and a 1 indicates a program access (FC[2:0] = x10). The emulator uses this signal to disassemble assembly code during trace.

24.1.4 Using the Interrupt Gate Module

There are three level 7 interrupt sources: two are internal and one is external. An internal level 7 interrupt is generated, if it is enabled, when a program or bus breakpoint is hit. An external level 7 interrupt is directly connected to the $\overline{\text{EMUIRQ}}$ pin, which is a falling edge trigger signal. The level 7 interrupt vector is hard coded to 0xFFFC0010 if the HMDIS bit in the ICEMCR register is clear. If HMDIS is set, refer to Chapter 15, “Interrupt Controller,” for information about generating a level 7 interrupt vector number.

When there is a level 7 interrupt, the software needs to check the in-circuit emulation module status register (ICEMSR) to determine the source of the interrupt. Each of these interrupts can be cleared by writing a 1 to the associated status bit. If the in-circuit emulation module is disabled, the $\overline{\text{EMUIRQ}}$ pin is the only source for level 7 interrupts.

24.1.5 Using the A-Line Insertion Unit

The A-line insertion unit will physically replace the data bus contents with 0xA000 in an instruction fetch cycle when the address of this bus cycle matches the breakpoint address. When an A-line insertion occurs, the in-circuit emulation module will wait for an A-line exception to occur. If an A-line exception occurs, a level 7 interrupt is generated to the signal that a program breakpoint hits.

24.2 Programming Model

This section contains information about the ICE registers and programming information about their settings.

24.2.1 In-Circuit Emulation Module Address Compare and Mask Registers

The in-circuit emulation module address compare register (ICEMACR) is used to store the address of the breakpoint, and the in-circuit emulation module address mask register (ICEMAMR) is used to mask the corresponding address bit in the ICEMACR. The in-circuit emulation module’s address comparator will compare the address bus value together with the control bus value to generate the $\overline{\text{EMUBRK}}$ signal. A range can be set by using the address mask bits to break in a range of memory so that the external address

comparator can take action if extra hardware breakpoints are needed. The register bit assignments are shown in the following register displays, and the settings of the bit assignments for both registers are described in Table 24-1 on page 24-4.

ICEMACR ICE Module Address Compare Register 0x(FF)FFFD00

	BIT 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	BIT 16
	AC 31	AC 30	AC 29	AC 28	AC 27	AC 26	AC 25	AC 24	AC 23	AC 22	AC 21	AC 20	AC 19	AC 18	AC 17	AC 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
	0x0															

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	AC 15	AC 14	AC 13	AC 12	AC 11	AC 10	AC 9	AC 8	AC 7	AC 6	AC 5	AC 4	AC 3	AC 2	AC 1	AC 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
	0x0															

ICEMAMR ICE Module Address Mask Register 0x(FF)FFFD04

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

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	AM 15	AM 14	AM 13	AM 12	AM 11	AM 10	AM 9	AM 8	AM 7	AM 6	AM 5	AM 4	AM 3	AM 2	AM 1	AM 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 24-1. ICE Module Address Compare and Mask Registers Description

Name	Description	Setting
ACx Bits 31–0	**Address Compare 31–0**—This field represents the value of the execution/bus breakpoint address. A match of address bits 31–0 with qualification of $\overline{AS}$ will generate a match signal.	See description.

Table 24-1. ICE Module Address Compare and Mask Registers Description

Name	Description	Setting
AMx Bits 31–0	**Address Mask 31–0**—This field masks the corresponding bits in the ACx field. With this masking scheme, a break can be made when the core is accessing a certain range of addresses.	0 = The address is compared to the current address cycle. 1 = Forces a true comparison (“don’t care”) on the corresponding bit.

24.2.2 ICE Module Control Compare and Mask Register

The in-circuit emulation module control compare register (ICEMCCR) is used to set the breakpoint at a specific bus cycle, and the in-circuit emulation module control mask register (ICEMCMR) is used to mask the corresponding control bit in the ICEMCMR. In bus breakpoint mode, the control signal comparator will compare the predefined control signals with the address compare match signal to generate the $\overline{\text{EMUBRK}}$ signal in single breakpoint mode. In multiple breakpoint mode, $\overline{\text{EMUBRK}}$ is an input signal and will AND with the result from the address comparator and control comparator to generate the internal match signal. For program break mode, these two registers are “don’t care.” The register bit assignments for both the compare and mask registers are shown in the following register displays. The settings for the bits are described in Table 24-2 and Table 24-3.

ICEMCCR ICE Module Control Compare Register **0x(FF)FFFD08**

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

Table 24-2. ICE Module Control Compare Register Description

Name	Description	Setting
Reserved Bits 15–2	Reserved	These bits are reserved and should be set to 0.
RW Bit 1	**Read or Write Cycle Selection**—This bit is used to select the break at a read cycle or write cycle. When a break at a read cycle is selected, a breakpoint at the ROM location is possible.	0 = Write cycle breakpoint. 1 = Read cycle breakpoint.
PD Bit 0	**Program or Data Cycle Selection**—This bit is used to select the break at a program cycle or data cycle.	0 = Data bus cycle. 1 = Instruction bus cycle.

ICEMCMR ICE Module Control Mask Register **0x(FF)FFFD0A**

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

Table 24-3. ICE Module Control Mask Register Description

Name	Description	Setting
Reserved Bits 15–2	Reserved	These bits are reserved and should be set to 0.
RWM Bit 1	**Read or Write Cycle Mask**—This bit masks the RW bit of the ICEMCCR.	0 = Enable the comparator to compare itself against the RW bit. 1 = Force a true comparison (“don’t care”) on the corresponding bit.
PDM Bit 0	**Program or Data Cycle Mask**—This bit masks the PD bit of the ICEMCCR.	0 = Enable the comparator to compare itself against the PD bit. 1 = Force a true comparison (“don’t care”) on the corresponding bit.

24.2.3 ICE Module Control Register

The in-circuit emulation module control register (ICEMCR) is used to control the in-circuit emulation module. The bit assignments for the ICE module control register are shown in the following register display. The settings for the bits are described in Table 24-4.

ICEMCR ICE Module Control Register **0x(FF)FFFD0C**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
											SWEN	BBIEN	HMDIS	SB	PBEN	CEN
TYPE											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 24-4. ICE Module Control Register Description

Name	Description	Setting
Reserved Bits 15–6	Reserved	These bits are reserved and should be set to 0.
SWEN Bit 5	**Software Enable EMU Module**—In normal mode, writing to this bit enables the breakpoint function.	0 = Disable breakpoint function. 1 = Enable breakpoint function.
BBIEN Bit 4	**Bus Break Interrupt Enable**—When set, this bit enables the generation of a level 7 interrupt on a bus breakpoint.	0 = Disable level 7 interrupt generation on a bus breakpoint. 1 = Enable level 7 interrupt generation on a bus breakpoint.

Table 24-4. ICE Module Control Register Description (Continued)

Name	Description	Setting
HMDIS Bit 3	**Hard-Map Disable**—In emulation mode, this bit activates the internal hard-map operation. When this bit is clear, some memory locations are hard-coded to the specific values shown inTable 24-5. If this bit is set or in normal mode, memory reads to these locations refer to the external memory. **Note:** It is important to note that when writing to these locations, all writes are occurring to external memory. When the HMDIS bit is disabled, reads to these addresses are in word or long-word sizes.	See Table 24-5 on page 24-7.
SB Bit 2	**Single BreakPoint**—This bit controls the direction of the $\overline{\text{EMUBRK}}$ signal. In multiple breakpoint mode, the external address comparator will compare the lower address bits and the internal comparator will compare the higher address bits to generate a breakpoint matched signal.	0 = Configure the $\overline{\text{EMUBRK}}$ signal as an input (multiple breakpoint mode with external address compare for the lower addresses). 1 = Configure the $\overline{\text{EMUBRK}}$ signal as an output (single breakpoint based on the internal address compare register).
PBEN Bit 1	**Program Break Enable**—This bit is used to select a program or bus break.	0 = Select a bus break. 1 = Select a program break.
CEN Bit 0	**Compare Enable**—This bit is used to activate the comparison logic. It is recommended that the address compare and mask registers be programmed before setting this bit to valid.	0 = Disable the breakpoint comparison logic. 1 = Enable the breakpoint comparison logic.

Table 24-5. Emulation Mode Hard Coded Memory Locations

Address	Hard Code
0x0	0xFFFC
0x2	0xFFFC
0x4	0xFFFC
0x6	0x0020
0x28	0xFFFC
0x2A	0x0010
(IRQ7 vector upper word)	0xFFFC
(IRQ7 vector lower word)	0x0010

24.2.4 ICE Module Status Register

The in-circuit emulation module status register (ICEMSR) is used to determine the source of an interrupt. The bit assignments for the ICE module status register are shown in the following register display. The settings for the bits are described in Table 24-6.

ICEMSR ICE Module Status Register **0x(FF)FFFD0E**

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

Table 24-6. ICE Module Status Register Description

Name	Description	Setting
Reserved Bits 15–4	Reserved	These bits are reserved and should be set to 0.
EMUEN Bit 3	**Emulation Enable**—When set this bit enables ICE mode.	0 = Normal mode. 1 = ICE mode.
BBIRQ Bit 2	**Bus Break Interrupt Detected**—This bit is set when a bus breakpoint is hit. Writing a 1 to this bit clears it.	0 = Bus breakpoint has not occurred. 1 = Bus breakpoint has occurred.
BRKIRQ Bit 1	**Line Vector Fetch Detected**—This bit is set when a program breakpoint is hit. Writing a 1 to this bit clears it.	0 = Program breakpoint has not occurred. 1 = Program breakpoint has occurred.
EMIRQ Bit 0	**$\overline{\text{EMUIRQ}}$ Falling Edge Detected**—This bit is set when the $\overline{\text{EMUIRQ}}$ pin is going from high to low. Writing a 1 to this bit clears it.	See description.

24.3 Typical Design Programming Example

Figure 24-2 illustrates an example of a typical emulator design. It is a simple and low-cost design that uses the MC68SZ328 as the processor to be emulated. Other functional units include the host control to the PC or workstation via an RS-232 or a dedicated parallel interface, an optional address comparator for extra breakpoint expansion, optional map FPGA for emulation memory remapping, a data bus MUX for hardware breakpoint insertion, and a MC68SZ328 pin-out extension to connect to the solder-on emulator pod. The entire MC68SZ328 bus should be buffered using level-shifting buffers when the emulator is designed in 5 V and the processor is running at 3.3 V.

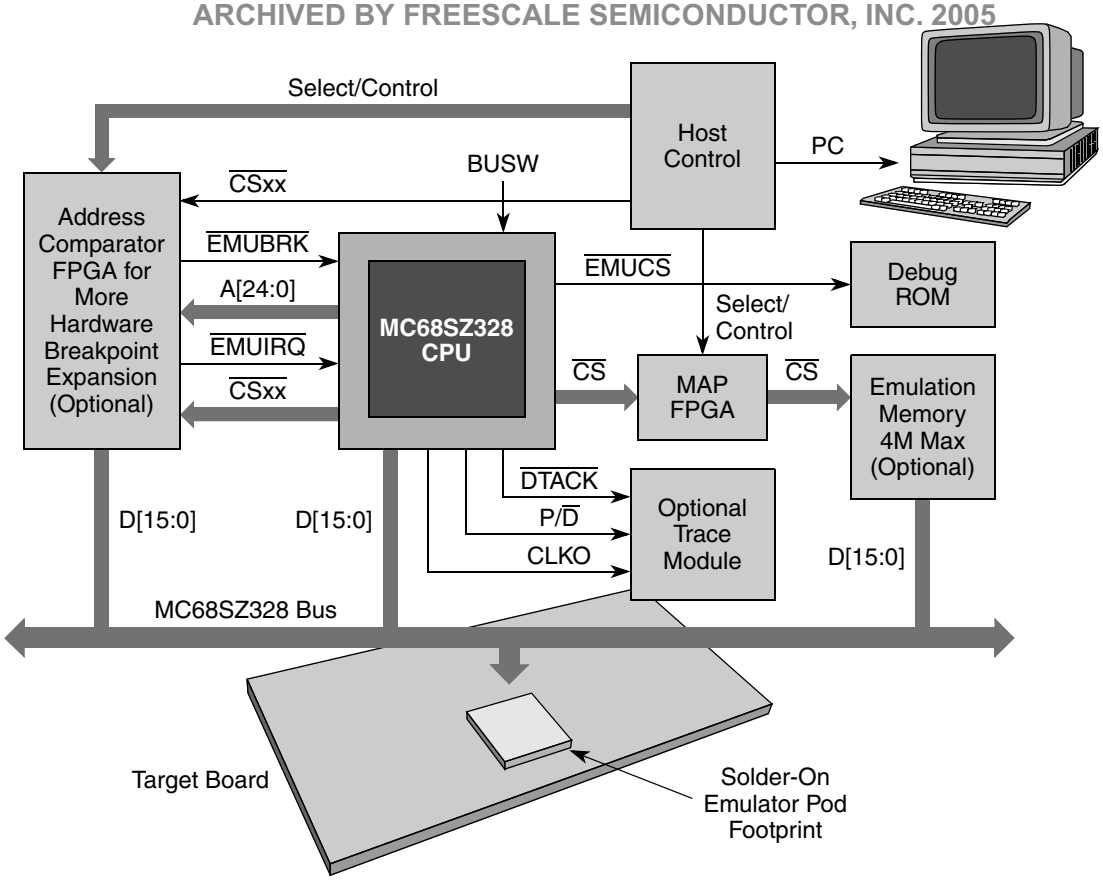


Figure 24-2. Typical Emulator Design Example

24.3.1 Host Interface

The host interface can be a processor-based or state-machine-based circuit that is used to coordinate the activities between the emulation processor and the PC host. The interface can be an RS-232 or printer parallel I/O. The interface runs on the PC, and it will translate its requests to low-level commands and send them to the emulator's controller if there is one.

24.3.2 Dedicated Debug Monitor Memory

When a breakpoint is matched, the CPU must report its status and grab the necessary contents, such as internal registers, in the system. This information is then transmitted to the host control processor to be translated before it is passed to the interface on the PC. The monitor program is located in ROM at 0xFFFC0000–0xFFFCFFFF and is enabled or disabled by the $\overline{\text{EMUCS}}$ signal.

24.3.3 Emulation Memory Mapping FPGA and Emulation Memory

Because the memory on the target board may not be fully built or debugged, it is necessary to have some memory that replaces the target memory for debugging at the initial stage. In some cases, ROM codes are downloaded to a shadowed RAM area for debugging purposes. The map FPGA will work with those chip-select signals to map them to the emulation memory, instead of going directly to the target board.

24.3.4 Optional Extra Hardware Breakpoint

The FPGA address comparator can be added to enhance the number of hardware breakpoints in the emulator. As discussed in Section 24.1.2, “Detecting Breakpoints,” on page 24-2, in multiple breakpoint mode the external FPGA address comparator compares the lower address, the internal comparator compares the upper hidden address line, and then a $\overline{\text{EMUIRQ}}$ signal is generated to tell the in-circuit emulation module to generate a breakpoint.

24.3.5 Optional Trace Module

A trace module may also be added to enhance the function of the emulator. Trace captures the bus signals of all of the cycles, so that when a stop is encountered, the interface software can report all the cycle traces back for that breakpoint. This action is based on the timebase of the CLKO signal, P/$\overline{\text{D}}$ signal, and $\overline{\text{DTACK}}$ signal to decide whether the trace capture is a program or data fetch.

24.4 Plug-in Emulator Design Example

Figure 24-3 on page 24-10 displays an example of a plug-in emulator design. The design is simple and low-cost, and it creates a very basic debugging environment.

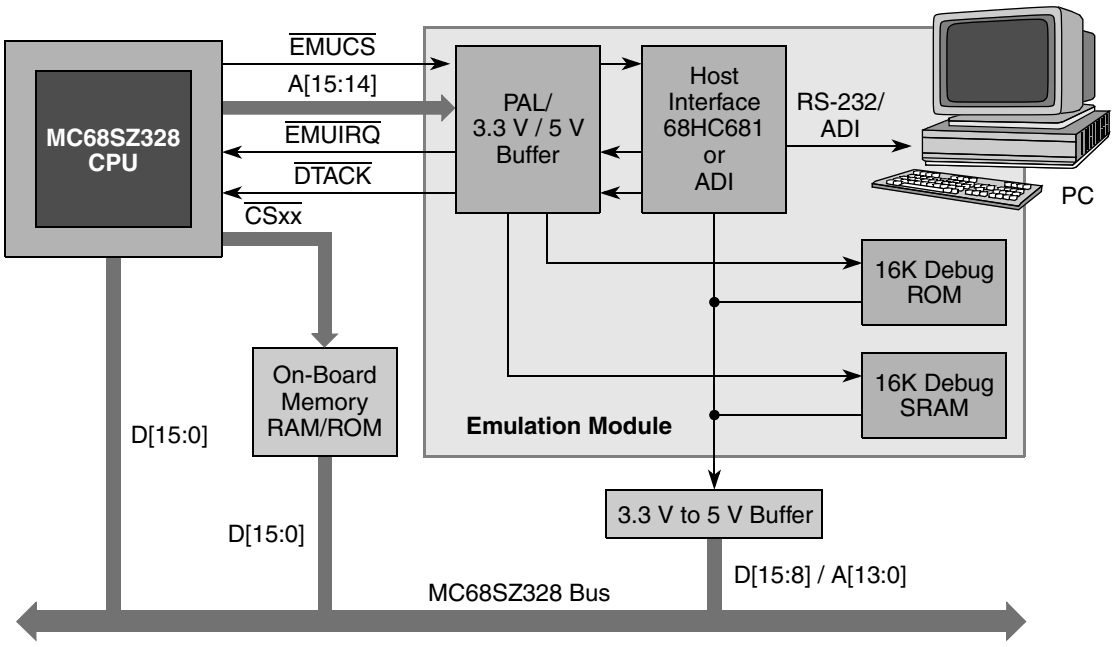


Figure 24-3. Plug-in Emulator Design Example

Although there is only one hardware breakpoint in this design, all other software breakpoints can be generated by replacing the memory content of the A0 instruction. The $\overline{\text{EMUCS}}$ is decoded by a PAL to generate chip-select signals to the UART (68HC681) or ADI interface and the debug RAM or ROM or both RAM and ROM. The emulation module is buffered with 3.3 V to 5.0 V buffers so that it can communicate with the PC without causing any problems.

The entire emulation module only uses 29 pins, including a ground signal. A very low-cost cable can be built to ship with the software debugger package. These pins can remain on the production version of the system board for production testing, as well as diagnostic and failure analysis.

24.5 Application Development Design Example

Figure 24-4 displays an example of an application development system design. This example is for initial start-up designs and software development that occurs after the target hardware system is completed.

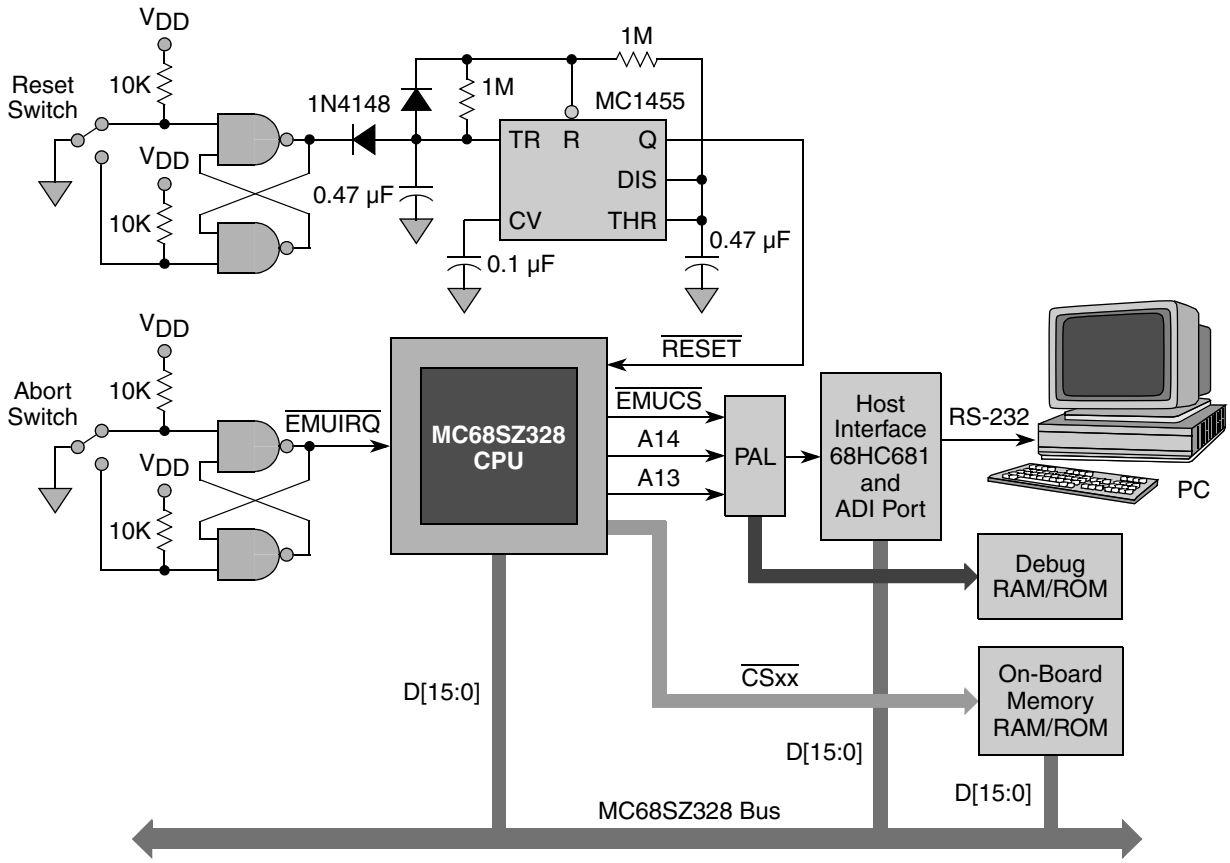


Figure 24-4. Application Development System Design Example

There is one reset switch and one abort switch. The abort switch is debounced and connected to the $\overline{\text{EMUIRQ}}$ signal. The $\overline{\text{RESET}}$ signal is generated by the MC1455 monostable timer. The host interface port is selected by the PAL decoding the $\overline{\text{EMUCS}}$, A13, and A14 signals. The board also provides optional SRAM and ROM plug-in sockets for expansion.

Freescale Semiconductor, Inc.

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Chapter 25
Reset Module

The reset module processes of all of the system reset signals required by the MC68SZ328. This chapter gives a detailed description of the reset module, associated timing, and signals.

25.1 Design Concern: External Oscillator Startup

Startup times vary from crystal to crystal and manufacturer to manufacturer. Low frequency crystals (20 kHz to 65 kHz) may require startup times of up to 3 seconds. A watch crystal may typically take from 500 milliseconds to 1.2 seconds to start up with the MC68SZ328. For this reason, an external RC timing circuit is required, as it is with the MC68VZ328 processor. The improved startup time results from an internal reset hold counter that provides a stable reset to the chip.

25.2 Module Design

Figure 25-1 is a block diagram of the reset module.

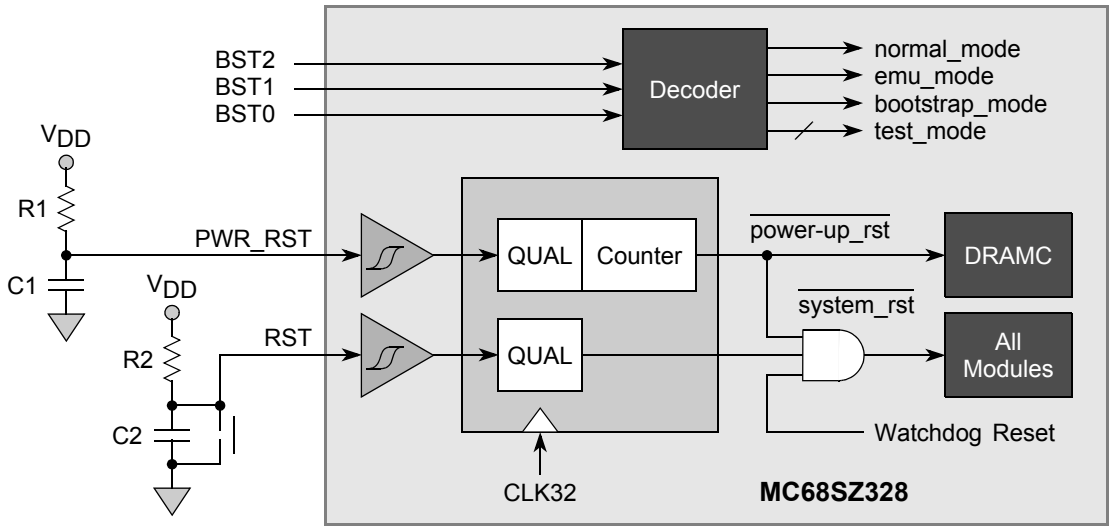


Figure 25-1. Reset Module Block Diagram

25.2.1 External Input Signals

The reset module receives two signals:

- PWR_RST—The PWR_RST signal is externally provided by the user to the MC68SZ328's $\overline{\text{PWR\_RST}}$ pin. This asynchronous signal is generated by a power-up reset. Although the pin is labeled $\overline{\text{PWR\_RST}}$, internally the signal is pwr_rst.
- RST—This signal is externally provided by the user to the MC68SZ328's $\overline{\text{RST}}$ pin. This asynchronous signal is generated by a normal reset. Although the pin is labeled $\overline{\text{RST}}$, internally the signal is named rst.

25.2.2 Internal Signals

The reset module outputs four signals:

- $\overline{\text{system\_rst}}$—This is an active low signal that signals the MC68SZ328 reset subsystem to initiate the reset for all internal modules.
- $\overline{\text{power-up\_rst}}$—This is an active low signal that signals the DRAM controller that the current reset is a power-up reset.
- Watchdog Reset—This is an reset signal generated by the watchdog timer.
- CLK32—This is the 32 kHz clock from the internal oscillator circuit. The internal oscillator circuit is driven by the user's external crystal.

25.3 Operation

Two types of resets affect the states of the MC68SZ328's I/O ports: normal resets and power-up resets.

25.3.1 Power-up Resets

A power-up reset occurs the first time power is supplied to the MC68SZ328. A power-up reset is also called a "cold start" or a "power on reset" (POR). Refer to Figure 25-2.

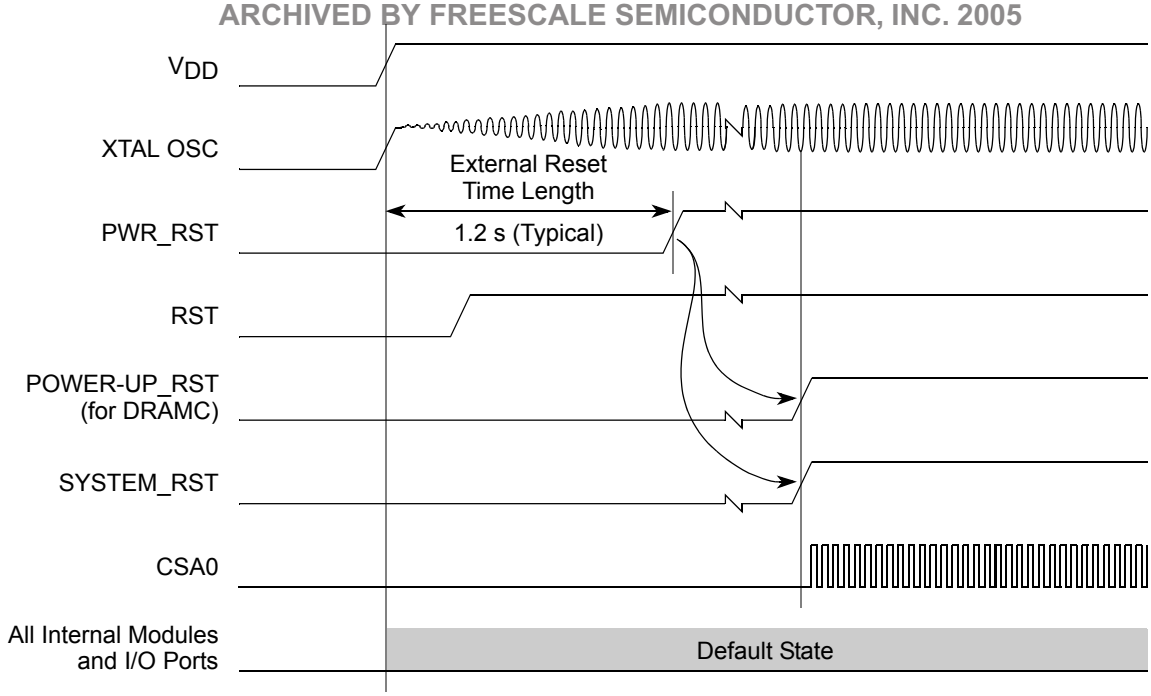


Figure 25-2. Power-up Reset

25.3.2 Normal Resets

A normal reset, also known as a "warm reset," refers to any reset initiated while power to the processor remains uninterrupted. A normal reset includes both external user resets and internal resets generated by the watchdog timer.

The sequence of events leading to the assertion of the internal reset pulse signal is as follows (refer to Figure 25-3):

1. The external reset signal is asserted.
2. The first falling edge of 32.768 kHz occurs.
3. When $\overline{\text{system\_rst}}$ asserts and $\overline{\text{power-up\_rst}}$ maintains a high state, the QUAL counter is reset.
4. The external reset signal is negated.
5. RST remains high.
6. After three falling edges of 32.768 kHz, the internal QUAL counter negates the internal $\overline{\text{system\_rst}}$ signal.

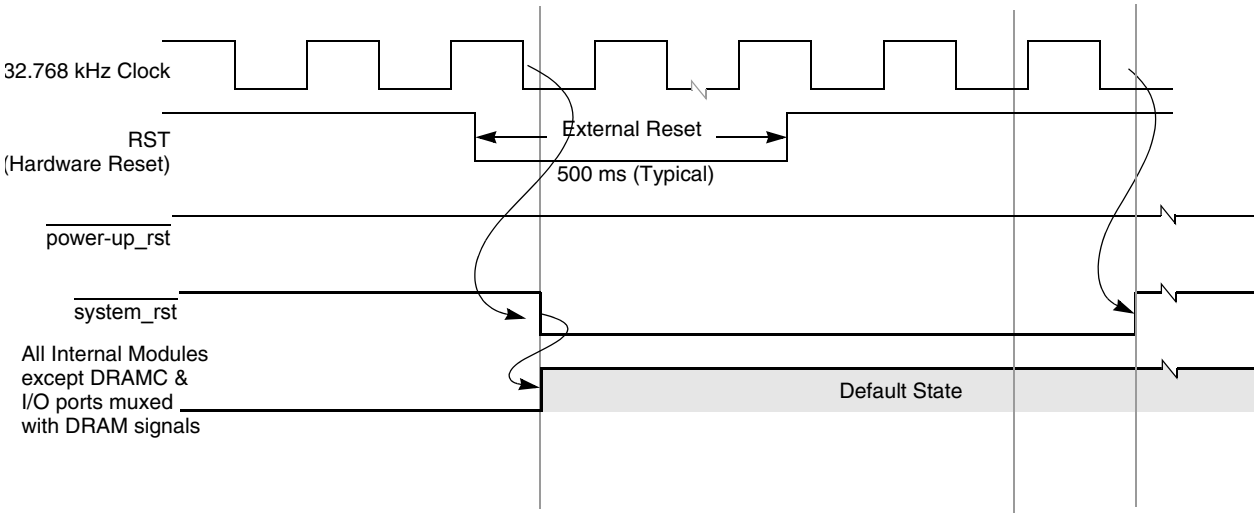


Figure 25-3. Normal Reset

25.4 Boot and Test Mode Selection

The MC68SZ328 provides eight boot and test modes selected by the BST[2:0] pins. The boot and test modes appear in Table 25-1.

Table 25-1. Boot and Test Modes

Boot Mode	BST2	BST1	BST0
Normal mode	0	0	0
Emulation mode	0	0	1
Bootstrap mode	0	1	0
Proprietary test mode 1	0	1	1
Proprietary test mode 2	1	0	0
Proprietary test mode 3	1	0	1
Proprietary test mode 4	1	1	0
High impedance mode	1	1	1

Chapter 26
Electrical Characteristics

This chapter documents electrical characteristics and provides timing information necessary to design systems using the MC68SZ328 microprocessor. Section 26.3, “DC Electrical Characteristics,” provides detailed information about both maximum and minimum DC characteristics of the MC68SZ328. Section 26.4, “AC Electrical Characteristics,” consists of output delays, input setup and hold times, and signal skew times. It also contains timing information for working with SRAM, flash, EDO DRAM, and SDRAM under different bus masters (68K, LCDC, or DMAC) and timing information for working with CSPI, I^2C, and USB peripheral devices.

26.1 Maximum Ratings

Table 26-1 provides information on maximum ratings for the MC68SZ328.

Table 26-1. Maximum Ratings

Rating	Symbol	Value	Unit
Supply voltage	V_{CC}	2.7 to 3.3	V
Input voltage	V_{IN}	0.0 to 3.3	V
Maximum operating temperature range	T_A	T_L to T_H 0 to 70	°C
Storage temperature	Test	–55 to 150	°C

26.2 Voltage Regulator Operating Specifications

A voltage regulator is used to step down the V_{DD} (3.0 V ± 10%) to 1.9 V ± 0.1 V. Table 26-2 provides specifications for the voltage regulator.

Table 26-2. Voltage Regulator Specifications

Parameters	Minimum	Typical	Maximum	Units
External V_{DD} Supply	2.7	3.0	3.3	V
Output Voltage	1.80	—	2.00	V
Regulation Current	—	—	120	mA
External Bypass Capacitor	—	680	—	nF

26.3 DC Electrical Characteristics

Table 26-3 contains both maximum and minimum DC characteristics of the MC68SZ328.

Table 26-3. Maximum and Minimum DC Characteristics

Number or Symbol	Characteristic	(3.0 ± 0.3) V			Unit
		Minimum	Typical	Maximum	
1	Full running operating current at 66 MHz[1]	—	70	80	mA
2	Standby current[2]	—	34	80	µA
V_{IH}	Input high voltage	0.7(V_{DD})	—	—	V
V_{IL}	Input low voltage	—	—	0.4	V
V_{OH}	Output high voltage (I_{OH} = 2.0 mA)	0.7(V_{DD})	—	—	V
V_{OL}	Output low voltage (I_{OL} = -2.5 mA)	—	—	0.4	V
I_{IL}	Input low leakage current (V_{IN} = GND, no pull-up or pull-down)	—	—	±1	µA
I_{IH}	Input high leakage current (V_{IN} = V_{DD}, no pull-up or pull-down)	—	—	±1	µA
I_{OH}	Output high current (V_{OH} = 0.8 V_{DD}, V_{DD} = 2.9 V)	4.0	—	—	mA
I_{OL}	Output low current (V_{OL} = 0.4V, V_{DD} = 2.9 V)	—	—	-4.0	mA
I_{OZ}	Output leakage current (V_{out} = V_{DD}, output is three-stated)	—	—	±5	µA

1. Full running operating current is measured with the following conditions:
   ESRAM, SDRAM Controller, LCD Controller and UART1 Enabled
   Test Program is running on ESRAM.
   LCD Display Data is fetched from SDRAM.
2. Standby current is measured with the real-time clock running.

26.4 AC Electrical Characteristics

The AC characteristics consist of output delays, input setup and hold times, and signal skew times. All signals are specified relative to an appropriate edge of other signals. All timing specifications are specified at an operating frequency from 0 MHz to 66 MHz with an operating supply voltage from $V_{DD\ min}$ to $V_{DD\ max}$ under an operating temperature from T_L to T_H. All timing is measured at 30 pf loading.

26.4.1 SYSCLK Reference to Chip-Select Signals Timing

Figure 26-1 compares the chip-select signal time referenced with the SYSCLK signal. Note that WS is the number of wait states in the current memory access cycle. The signal values and units of measure for this figure are found in Table 26-4. For detailed information about the individual signals, see Chapter 7, “Chip-Select Module.”

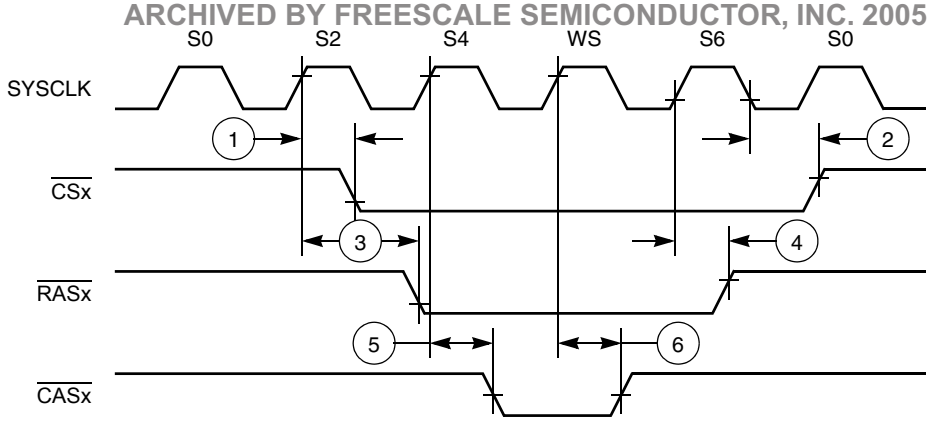


Figure 26-1. SYSCLK Reference to Chip-Select Signals Timing Diagram

Table 26-4. SYSCLK Reference to Chip-Select Signals Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	SYSCLK high to $\overline{\text{CSx}}$ asserted	—	5	ns
2	SYSCLK low to $\overline{\text{CSx}}$ negated	—	6	ns
3	SYSCLK high to $\overline{\text{RASx}}$ asserted	—	5	ns
4	SYSCLK high to $\overline{\text{RASx}}$ negated	—	6	ns
5	SYSCLK high to $\overline{\text{CASx}}$ asserted	—	5	ns
6	SYSCLK high to $\overline{\text{CASx}}$ negated	—	6	ns

26.4.2 SRAM/Flash Operation

26.4.2.1 68K Bus Master Mode Operation

26.4.2.1.1 68K Read Cycle

Figure 26-2 shows the read cycle timing used by chip-select. The signal values and units of measure for this figure are found in Table 26-5. For detailed information about the individual signals, see Chapter 7, “Chip-Select Module.”

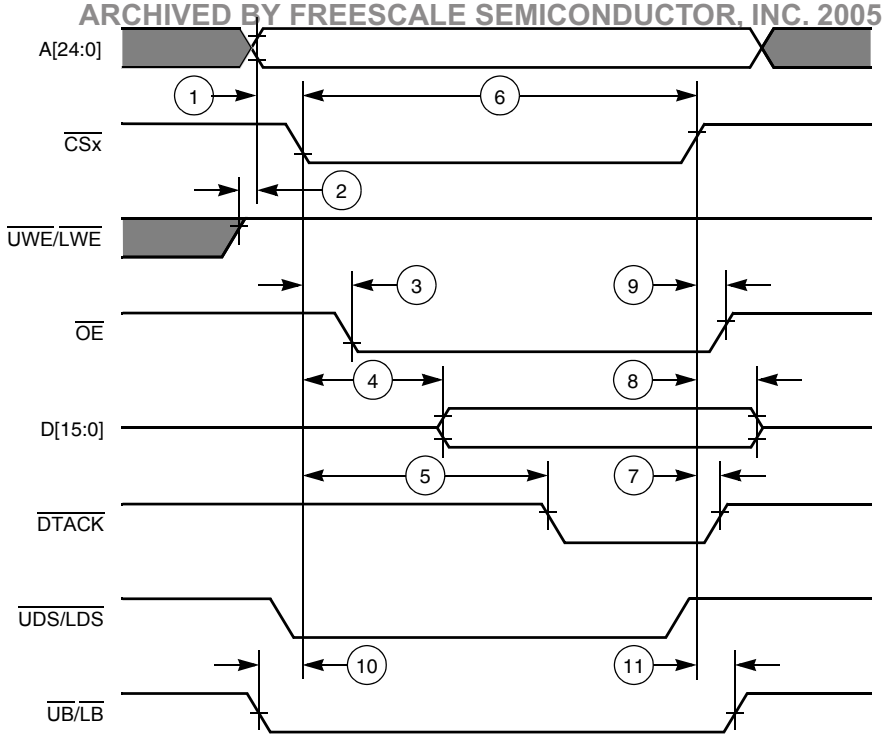


Figure 26-2. 68K Read Cycle Timing Diagram

Table 26-5. 68K Read Cycle Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Address valid to $\overline{CSx}$ asserted (bit LCS = 0, bit LCS = 1)	8, 8 + T/2	—	ns
2	$\overline{UWE}/\overline{LWE}$ negated before row address valid	0	—	ns
3	$\overline{CSx}$ asserted to $\overline{OE}$ asserted	—	5	ns
4	Data-in valid from $\overline{CSx}$ asserted	—	17 + nT	ns
5	External $\overline{DTACK}$ input setup from $\overline{CSx}$ asserted	—	8 + nT	ns
6	$\overline{CSx}$ pulse width	30 + nT	—	ns
7	External $\overline{DTACK}$ input hold after $\overline{CSx}$ is negated	0	—	ns
8	Data-in hold after $\overline{CSx}$ is negated	0	—	ns
9	$\overline{OE}$ negated after $\overline{CSx}$ is negated	0	10	ns
10	$\overline{UB}/\overline{LB}$ asserted to $\overline{CSx}$ asserted (16-bit SRAM)	10	—	ns
11	$\overline{CSx}$ negated to $\overline{UB}/\overline{LB}$ negated (16-bit SRAM)	10	—	ns

Table 26-5. 68K Read Cycle Timing Parameters (Continued)

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	

Note:
1. n is the number of wait states in the current memory access cycle and T is the system clock period.
2. The external $\overline{\text{DTACK}}$ input requirement is eliminated when $\overline{\text{CSx}}$ is programmed to use internal $\overline{\text{DTACK}}$.
3. $\overline{\text{CSx}}$ stands for $\overline{\text{CSA0}}$, $\overline{\text{CSA1}}$, $\overline{\text{CSB0}}$, $\overline{\text{CSB1}}$, $\overline{\text{CSC0}}$, $\overline{\text{CSC1}}$, $\overline{\text{CSD0}}$, $\overline{\text{CSD1}}$, $\overline{\text{CSE}}$, $\overline{\text{CSF}}$, and $\overline{\text{CSG}}$.
4. A value in parentheses is used when early cycle detection is turned on.

26.4.2.1.2 68K Write Cycle

Figure 26-3 shows the write cycle timing used by chip-selects. The $\overline{\text{WE}}$ negates a half CPU clock earlier before the negation of $\overline{\text{CSX}}$ at (a). The chip-select control register 3 setting for this is that the EWE bit is set (by default) and the WPEXT is clear (by default). If the WPEXT is set, the negation of WE is pushed forward by one clock at (b). The signal values and units of measure for this figure are found in Table 26-6. For detailed information about the individual signals, see Chapter 7, "Chip-Select Module."

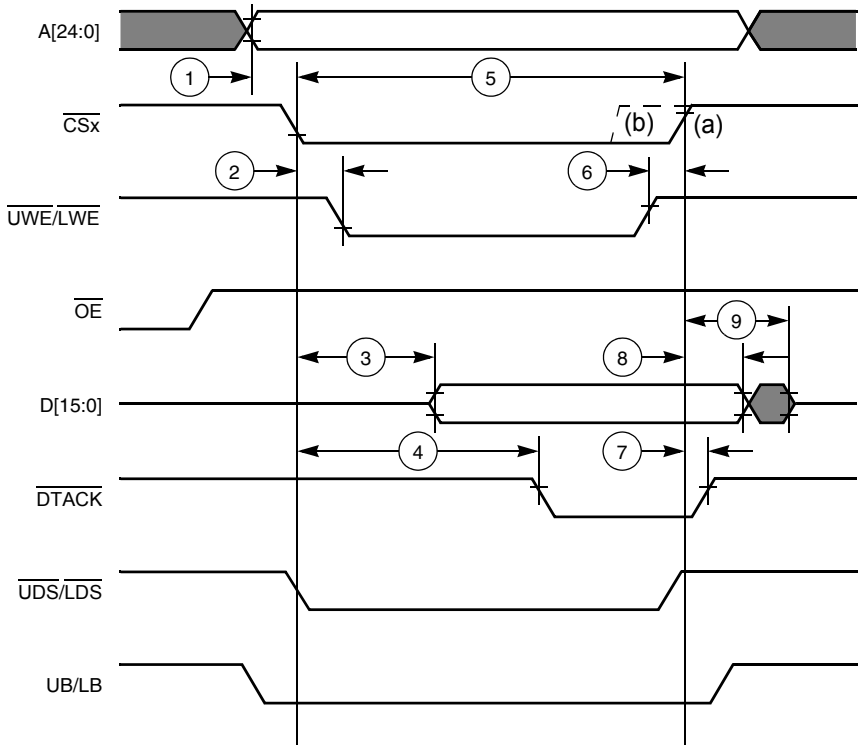


Figure 26-3. 68K Write Cycle Timing Diagram

Table 26-6. 68K Write Cycle Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Address valid to $\overline{CSx}$ asserted (bit LCS = 0, bit LCS = 1)	8, 8 + T/2	—	ns
2	$\overline{CSx}$ asserted to $\overline{UWE}/\overline{LWE}$ asserted (bit Flash= -, bit Flash = 1)	0, 10	2, 20	ns
3	$\overline{CSx}$ asserted to data-out valid	—	15	ns
4	External $\overline{DTACK}$ input setup from $\overline{CSx}$ asserted	—	8 + nT	ns
5	$\overline{CSx}$ pulse width	30 + nT	—	ns
6	$\overline{UWE}/\overline{LWE}$ negated before $\overline{CSx}$ is negated (Bit EWE=0)	—	0	ns
	(Bit EWE=1, Bit WPEXT=0)	5	10	ns
	(Bit EWE=1, Bit WPEXT=1)	20	25	ns
7	External $\overline{DTACK}$ input hold after $\overline{CSx}$ is negated	0	—	ns
8	Data-out hold after $\overline{CSx}$ is negated (Bit EWE=1, WPEXT=0)	4	—	ns
	(Bit EWE=1, WPEXT=1)	19	—	ns
9	$\overline{CSx}$ negated to data-out in Hi-Z	—	9	ns

Note:
1. n is the number of wait-states in the current memory access cycle and T is the system clock period.
2. The external $\overline{DTACK}$ input requirement is eliminated when $\overline{CSx}$ is programmed to use the internal $\overline{DTACK}$.
3. $\overline{CSx}$ stands for $\overline{CSA0}$, $\overline{CSA1}$, $\overline{CSB0}$, $\overline{CSB1}$, $\overline{CSC0}$, $\overline{CSC1}$, $\overline{CSD0}$, $\overline{CSD1}$, $\overline{CSE}$, $\overline{CSF}$, and $\overline{CSG}$.
4. A value in parentheses is used when early detection is turned on.

26.4.2.2 DMAC/LCDC Bus Mode Operation

26.4.2.2.1 Burst Read Cycle (Burst Length = 6; Wait State = 0)

Figure 26-4 shows the DMAC/LCDC burst read timing diagram for 16-bit access (0 wait state). The $\overline{CSx}$ does not toggle. In Figure 26-5, the $\overline{CSx}$ does toggle. Note that WS is the number of wait states in the current memory access cycle. The timing option is later CS via the LCS bit for CSB only. This bit is cleared by default. When it is cleared, the CSx assertion will be at (1). When it is set, one clock cycle is added and the CSx assertion will be at (2). The signal values and units of measure for this figure are found in Table 26-7. Detailed information about the operation of individual signals can be found in Chapter 7, “Chip-Select Module.” Chapter 9, “DMA Controller,” and Chapter 10, “LCD Controller.”

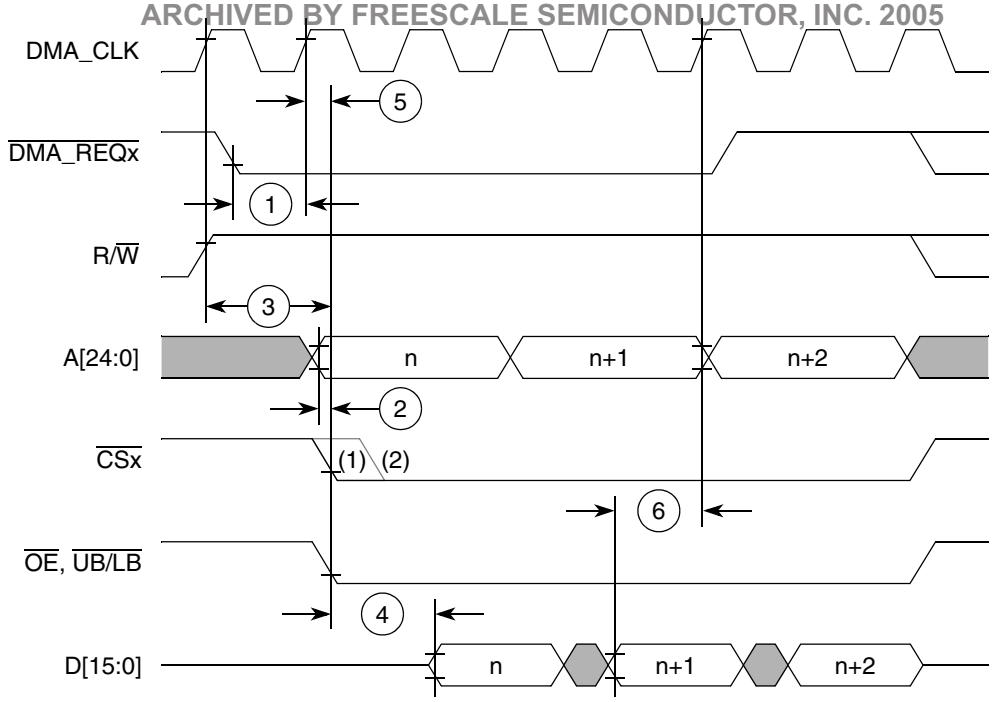


Figure 26-4. DMAC/LCDC Burst Read*

*The DMA read shown in Figure 26-4 has 3 burst. The WS field in the chip-select register is set to 0 wait states, and the LCWS bit in chip-select control register 3 is cleared.

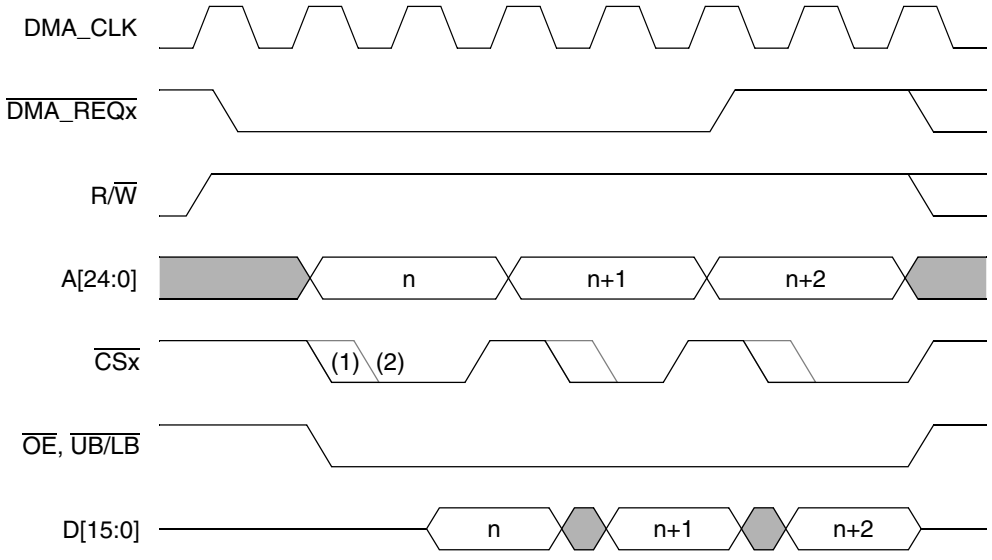


Figure 26-5. DMA Burst Read*

* Figure 26-5 presents the same timing option as Figure 26-4 but with CST = 1 so that CSx goes inactive after every word access. The wait state setting is the same as in Figure 26-4, but the actual DMA wait state is 2 on average.

Table 26-7. DMAC/LCDC Burst Read of 3 Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	$\overline{\text{DMA\_REQx}}$ setup time	0.5	—	ns
2	Address valid to $\overline{\text{CSx}}$ asserted	13	—	ns
3	$\overline{\text{RW}}$ to $\overline{\text{CSx}}$ asserted	—	—	ns
4	$\overline{\text{CSx}}$ asserted to data-out valid	—	20	ns
5	DMA_CLK high to $\overline{\text{CSx}}$	6	—	ns
6	Data setup time	13	—	ns

Note:
1. $\overline{\text{DMA\_REQx}}$ stands for $\overline{\text{DMA\_REQ1}}$ and $\overline{\text{DMA\_REQ2}}$. The $\overline{\text{DMA\_REQx}}$ signal is requested only if the external $\overline{\text{DMA\_REQx}}$ function is enabled.
2. $\overline{\text{CSx}}$ stands for $\overline{\text{CSA0}}$, $\overline{\text{CSA1}}$, $\overline{\text{CSB0}}$, $\overline{\text{CSB1}}$, $\overline{\text{CSC0}}$, $\overline{\text{CSC1}}$, $\overline{\text{CSD0}}$, $\overline{\text{CSD1}}$, $\overline{\text{CSE}}$, $\overline{\text{CSF}}$, and $\overline{\text{CSG}}$.

26.4.2.2.2 Single Write Cycle

Figure 26-6 is a single write cycle in DMA mode. If the CPU wait state setting is 0, in DMA mode, a 3-clock-cycle CSx is generated with LCWS set. If LCWS is clear, a 2-clock-cycle CSx is generated instead. When the WPEXT bit in chip-select control register 3 is clear, WE negates at (1)—that is, 1 clock earlier than the CSx negation. When the bit is set, WE negates at (2), 1 clock earlier than (1).

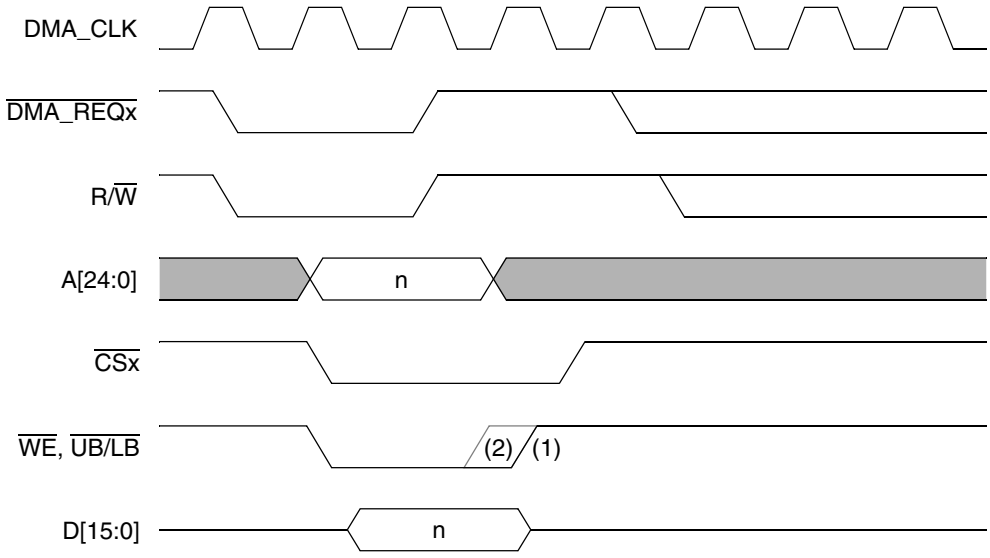


Figure 26-6. DMA Single Write*

* The WS field (wait state setting) in the chip-select register is set to 0 wait states, and the LCWS bit in chip-select control register 3 is set.

26.4.2.2.3 Burst Write Cycle (DMAC only; Burst Length = 6; Wait State = 0)

Figure 26-7 is the timing diagram for the DMAC burst write of 3 for 16-bit access (0 wait state). LCWS is clear so that each datum has an access time of 2 clocks. As shown in the diagram, the SRAM address, may extend over 2 clocks in each read. EWE is set and EPEXT is clear so that the WE pulses are just 1 clock

wide. EPEXT cannot be set in 2 clocks of access time. The minimum access time setting for EPEXT to take effect is 3 clocks. In Figure 26-7, note that WS is the number of wait states in the current memory access cycle. The signal values and units of measure for this figure are found in Table 26-8. Detailed information about the operation of individual signals can be found in Chapter 7, "Chip-Select Module." Chapter 9, "DMA Controller," and Chapter 10, "LCD Controller."

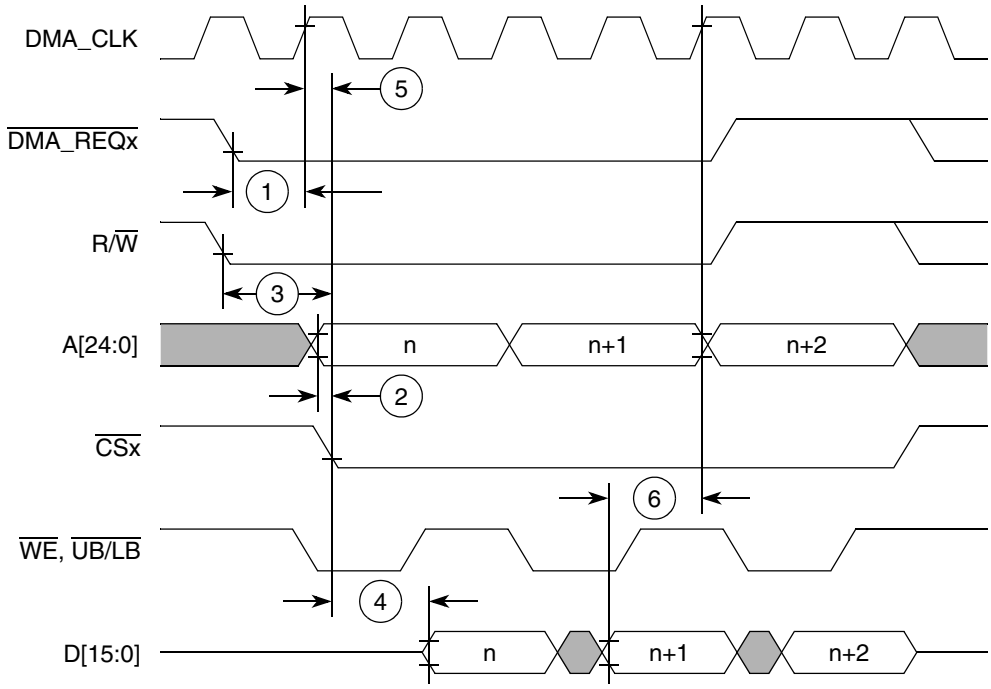


Figure 26-7. DMA Burst Write*

*DMA burst write of 3. The WS field in the chip-select register is set to 0 wait states, and the LCWS bit in chip-select control register 3 is cleared.

Table 26-8. DMAC Burst Write of 3 Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	DMA_REQx setup time	0.5	—	ns
2	Address valid to CSx asserted	13	—	ns
3	R/W to CSx asserted	1	—	ns
4	CSx asserted to data-out valid	0	—	ns
5	DMA_CLK high to CSx	6	—	ns
6	Data setup time	13	—	ns

Note:
1. DMA_REQx stands for DMA_REQ1, and DMA_REQ2. The DMA_REQx signal is requested only if the external DMA_REQx function is enabled.
2. CSx stands for CSA0, CSA1, CSB0, CSB1, CSC0, CSC1, CSD0, CSD1, CSE, CSF, and CSG.

26.4.3 SDRAM Operation

26.4.3.1 SDRAM Command Encodings

Table 26-9 summarizes the command encodings utilized by the SDRAM controller. These commands represent a subset of the commands defined by the JEDEC standard. Note that the SDRAM Auto/Self-Refresh Load Command Register commands share the same encoding.

Table 26-9. SDRAM Command Encodings

Function	Symbol	CKE n-1	CKE n	CS	RAS	CAS	WE	A10	BA[1:0]	A[9:0]
Deselect	DSEL	H	X	H	X	X	X	X	X	X
No Operation	NOP	H	X	L	H	H	H	X	X	X
Read	READ	H	X	L	H	L	H	L	V	V
Write	WRIT	H	X	L	H	L	L	L	V	V
Bank Activate	ACT	H	X	L	L	H	H	V	V	V
Burst Terminate	TBST	H	X	L	H	H	L	X	V	X
Precharge Select Bank	PRE	H	X	L	L	H	L	L	V	X
Precharge All Banks	PALL	H	X	L	L	H	L	H	X	X
Auto-Refresh	CBR	H	X	L	L	L	H	X	X	X
Self Refresh Entry	SLFRSH	H	L	L	L	L	H	X	X	X
Self Refresh Exit	SLFRSHX	L	H	H	X	X	X	X	X	X
Power-Down Entry	PWRDN	H	L	X	X	X	X	X	X	X
Power-Down Exit	PWRDNX	L	H	H	X	X	X	X	X	X
Mode Register Set	MRS	H	X	L	L	L	L	V	V	V

26.4.3.2 Normal Read/Write Mode Operation

26.4.3.2.1 68K Off-Page Read Cycle (16-Bit Memory)

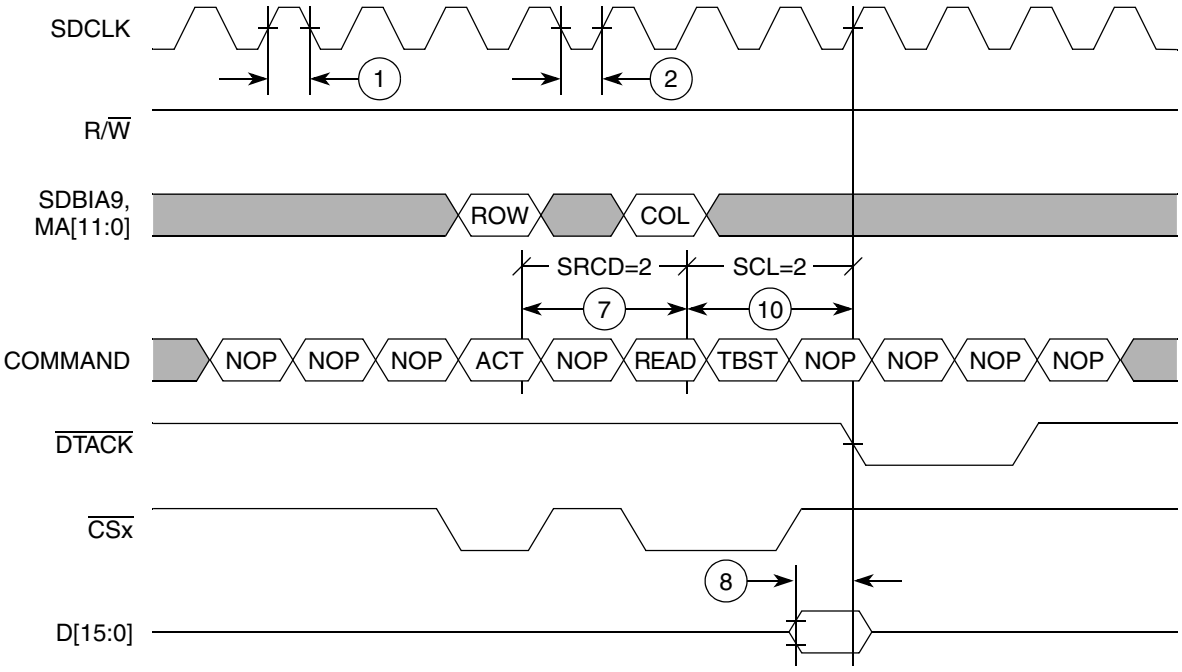


Figure 26-8. 68K Off-Page Read Cycle (16-Bit Memory)

26.4.3.2.2 68K On-Page Read Cycle (16-Bit Memory)

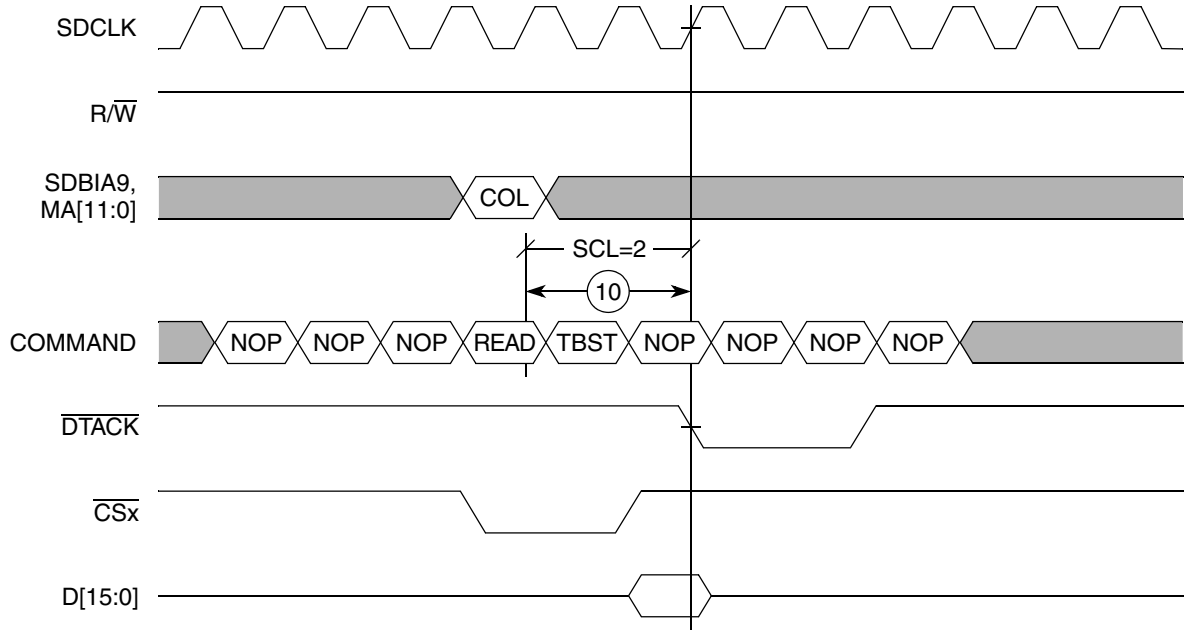


Figure 26-9. 68K On-Page Read Cycle (16-Bit Memory)

26.4.3.2.3 68K Miss-Page Read Cycle (16-Bit Memory)

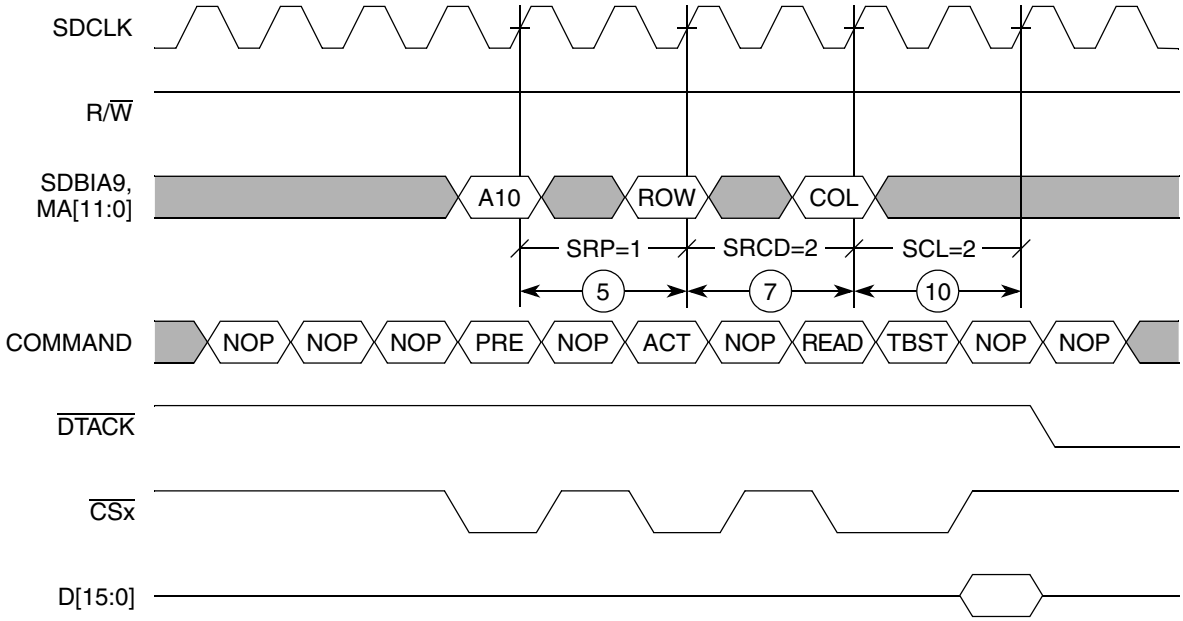


Figure 26-10. 68K Miss-Page Read Cycle (16-Bit Memory)

26.4.3.2.4 68K Off-Page Write Followed by On-Page Write Cycle

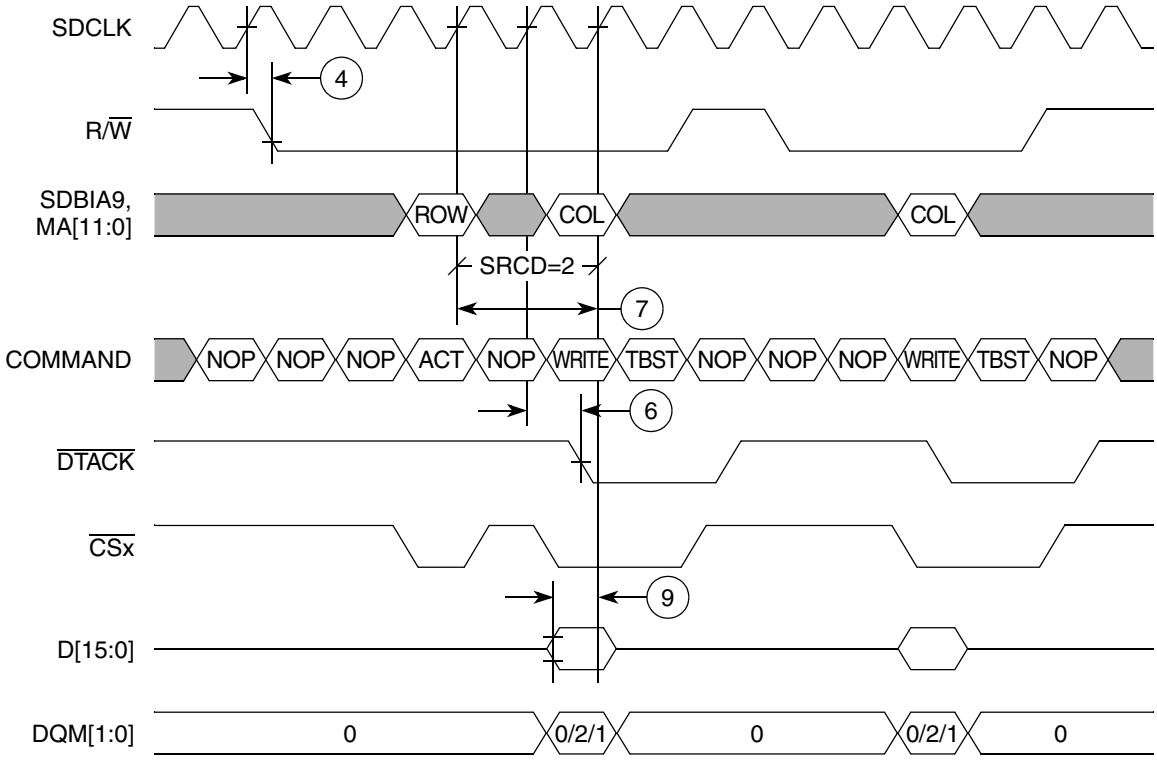


Figure 26-11. 68K Off-Page Write Followed by On-Page Write Cycle

26.4.3.2.5 Precharge Command Mode

The Precharge Command Mode (SMODE = 001) is used during SDRAM device initialization and to manually deactivate one or more active banks. While in this mode, an access (either read or write) to the SDRAM address space will generate a precharge command cycle.

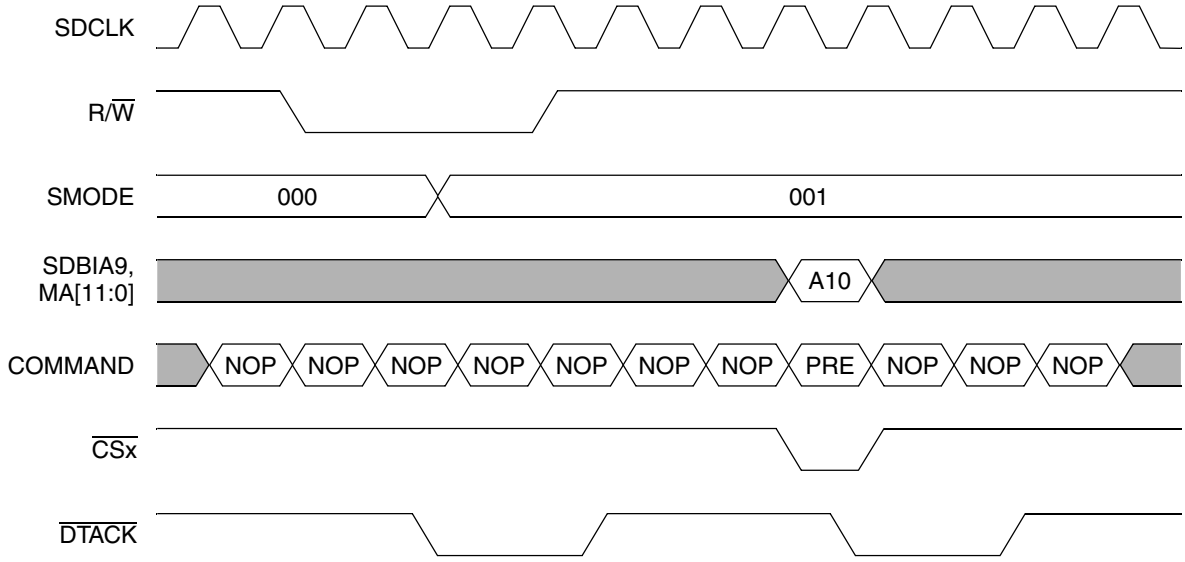


Figure 26-12. Software-Activate Precharge Command

26.4.3.2.6 Auto-Refresh Mode

The Auto-Refresh Mode (SMODE = 010) is used to manually request SDRAM refresh cycles and 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; therefore the address supplied during the refresh command need only specify the correct SDRAM device. The lower address lines are “don’t care.” Either a read or write cycle may be used.

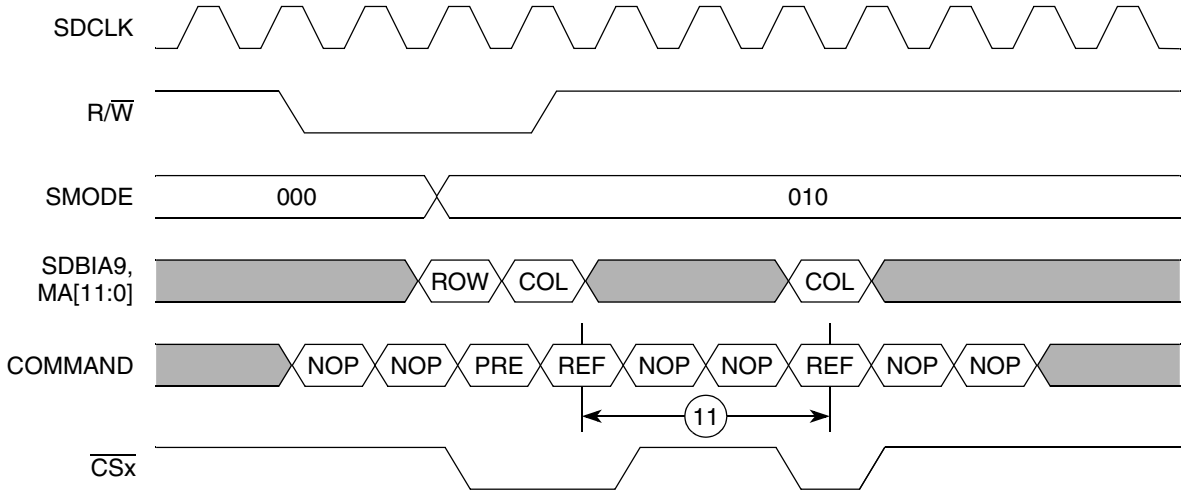


Figure 26-13. Software Initiated Auto-Refresh Timing Diagram

26.4.3.2.7 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.

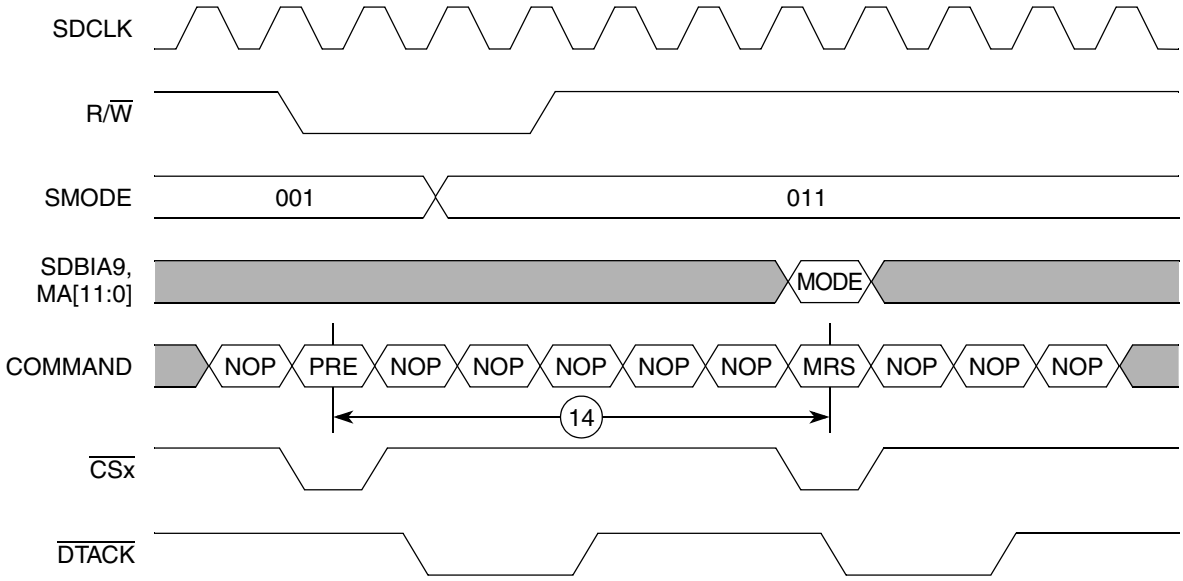


Figure 26-14. Software-Activate Set Mode Register Command

26.4.3.2.8 Self-Refresh Entry

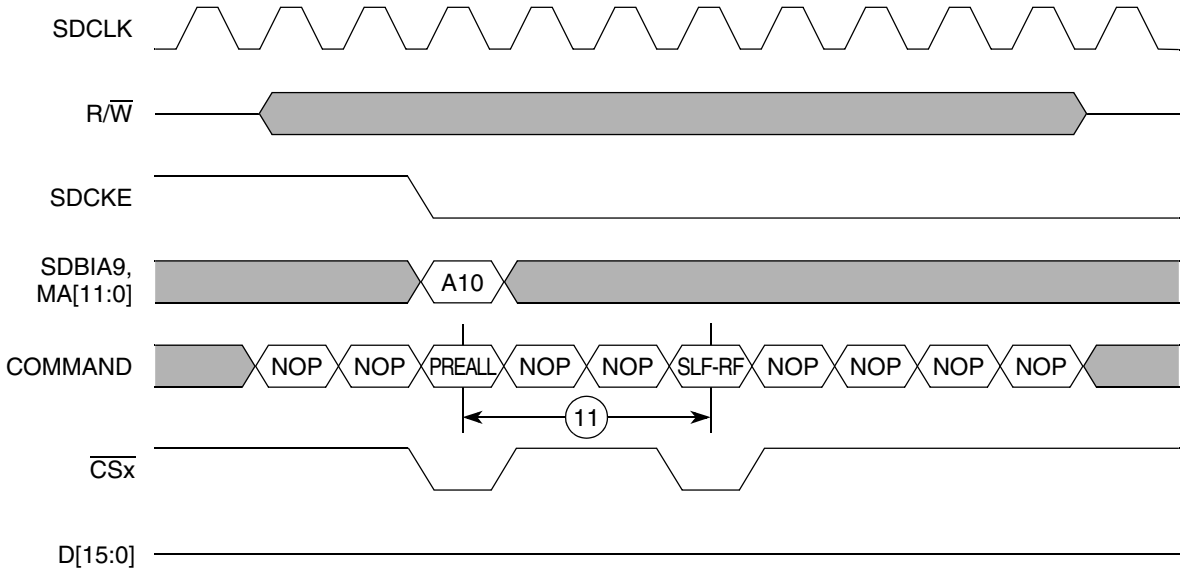


Figure 26-15. Self-Refresh Entry Timing Diagram

26.4.3.2.9 Self-Refresh Exit

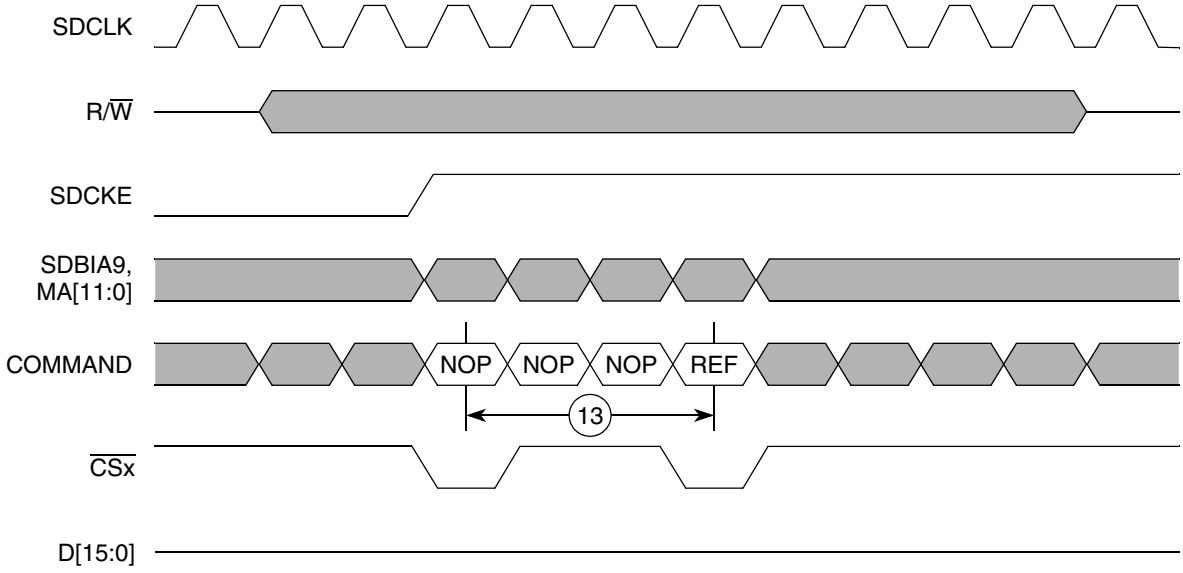


Figure 26-16. Self-Refresh Exit Timing Diagram

26.4.3.2.10 Power-Down Mode Entry

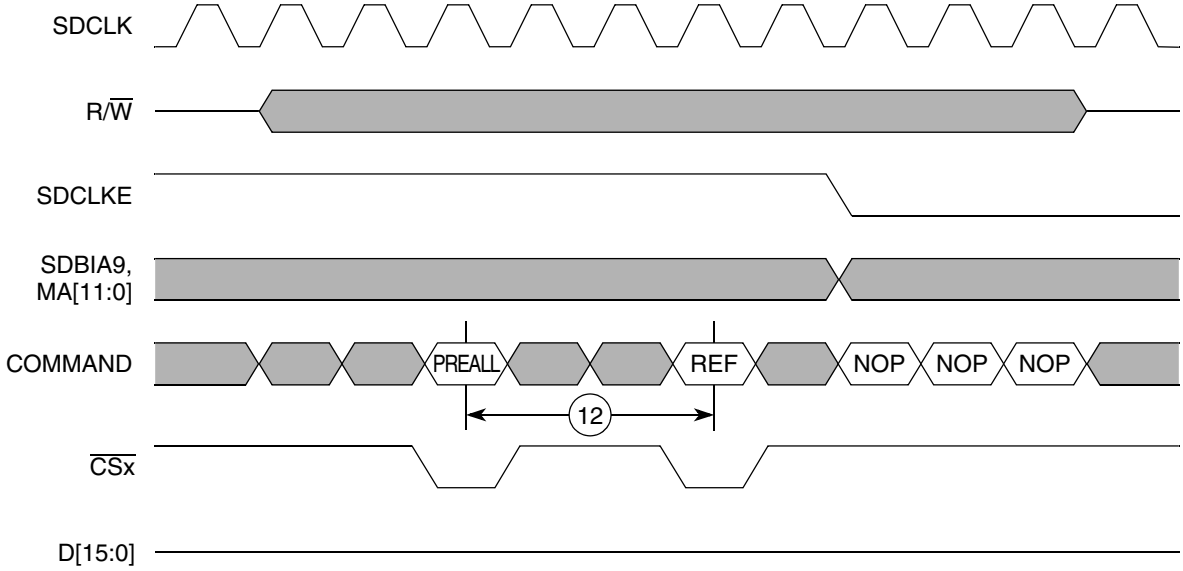


Figure 26-17. Power-Down Mode Entry Timing Diagram

26.4.3.2.11 Power-Down Mode Exit

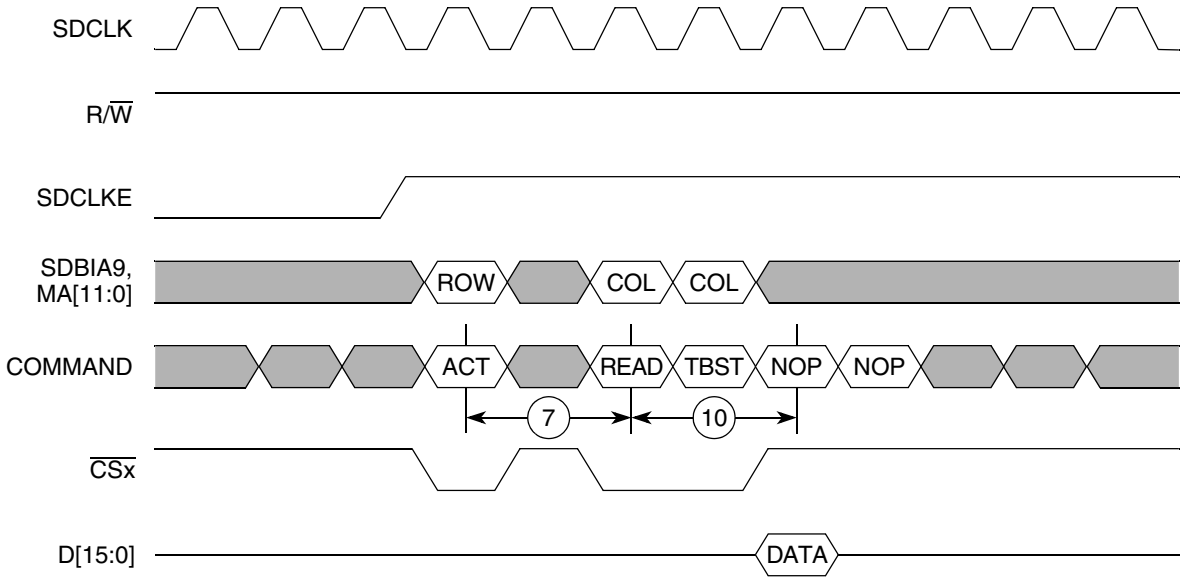


Figure 26-18. Power-Down Exit Timing Diagram

Table 26-10. Timing Parameters for Figure 26-8 Through Figure 26-18

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Clock high pulse time	7	—	ns
2	Clock low pulse time	7	—	ns
3	Clock high to address valid	3	13	ns
4	Clock high to R/W low	3	12	ns
5	Precharge command to active command (bit SRP = 1, 0)	30, 45	—	ns
6	Clock high to DTACK asserted	10	—	ns
7	Active command to read/write command (bits SRCD[1:0] = 00, 01, 10, 11)	60, 15, 30, 45	—	ns
8	Data setup time	13	—	ns
9	Data valid to clock high	10	—	ns
10	Read tp data sample latency (bits SCL[1:0] = 01, 10, 11)	15, 30, 45	—	—
11	Delay of between refresh cycles.	Refresh rate	—	—
12	Precharge command to refresh command (bit SRP = 1, 0)	30, 45	—	—
13	nop command to refresh command	15	—	ns
14	Precharge command to set mode command	90	—	ns

26.4.3.3 DMAC/LCDC Burst Read Cycle

26.4.3.3.1 On-Page Burst Read Cycle

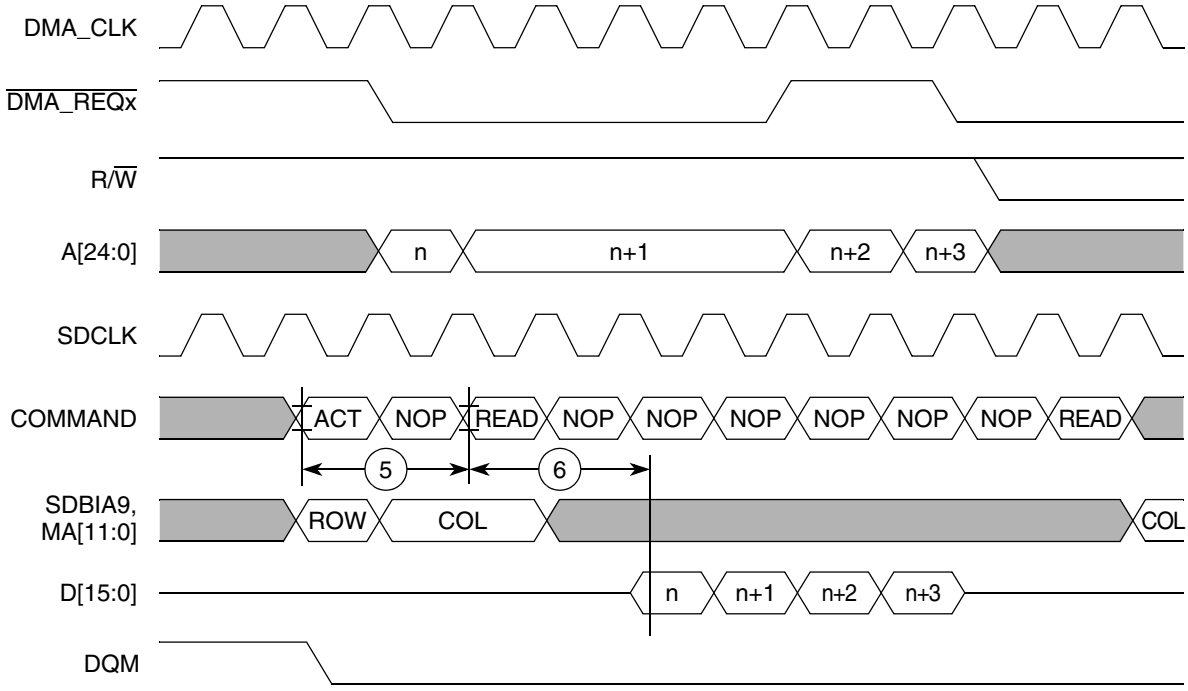


Figure 26-19. On-Page Burst Read Timing Diagram

26.4.3.3.2 Off-Page Burst Read Cycle

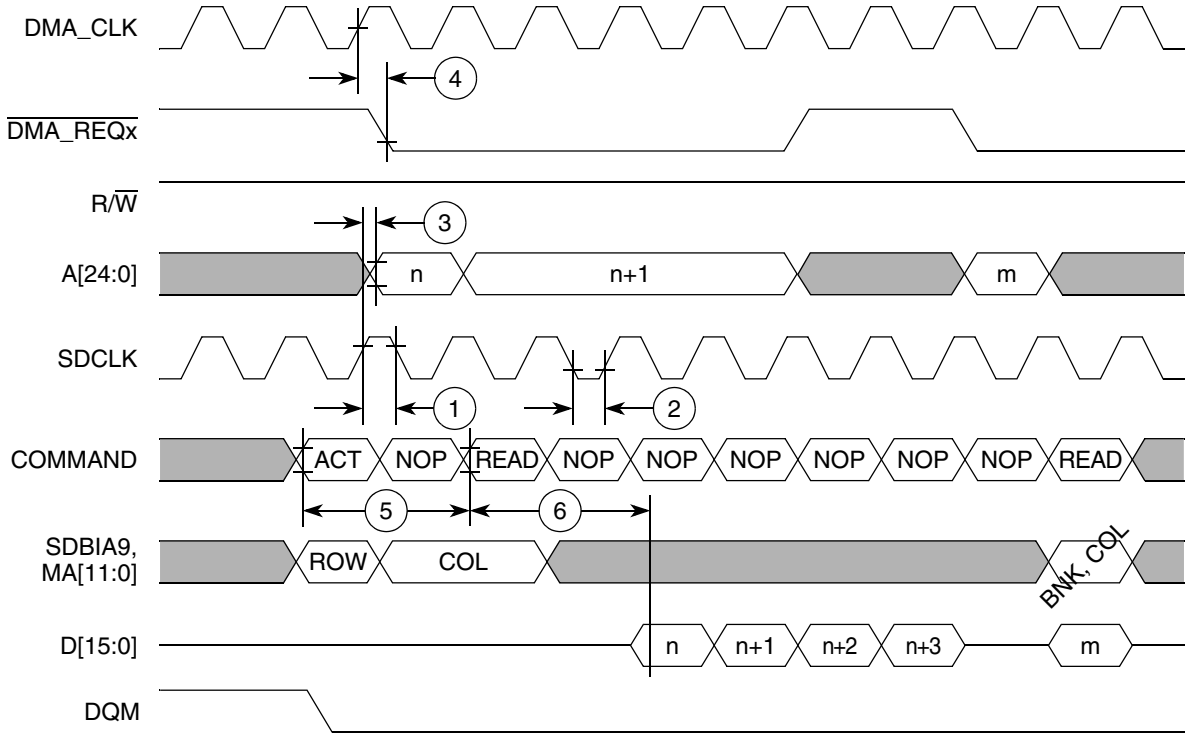


Figure 26-20. Off-Page Burst Read Timing Diagram

26.4.3.4 DMAC Burst Write Cycle

26.4.3.4.1 On-Page Burst Write Cycle

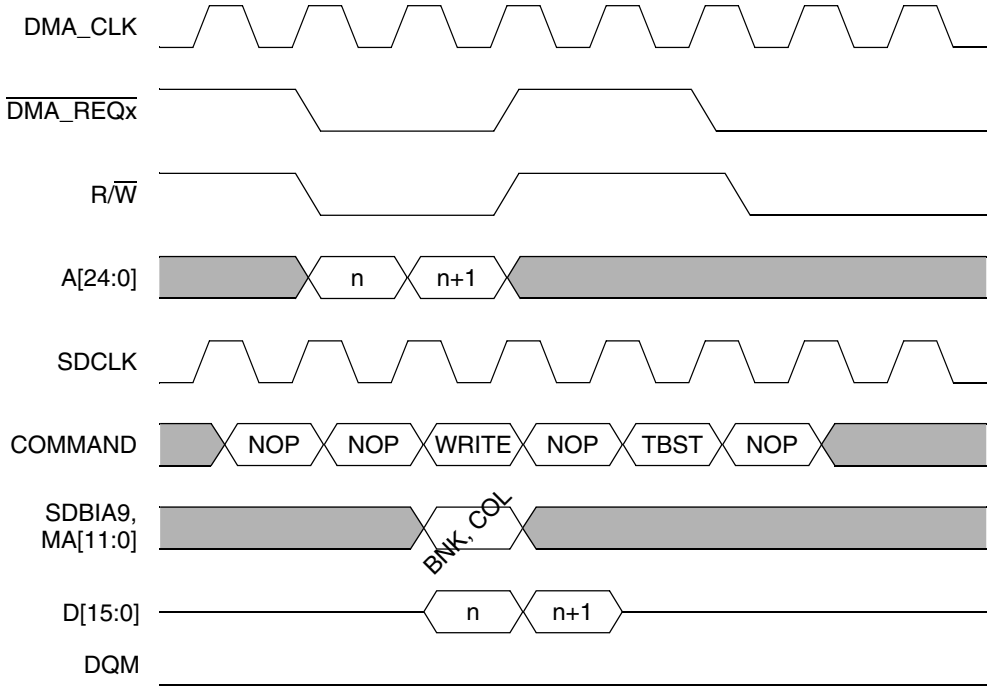


Figure 26-21. On-Page Burst Write Timing Diagram

26.4.3.4.2 Off-Page Burst Write Cycle

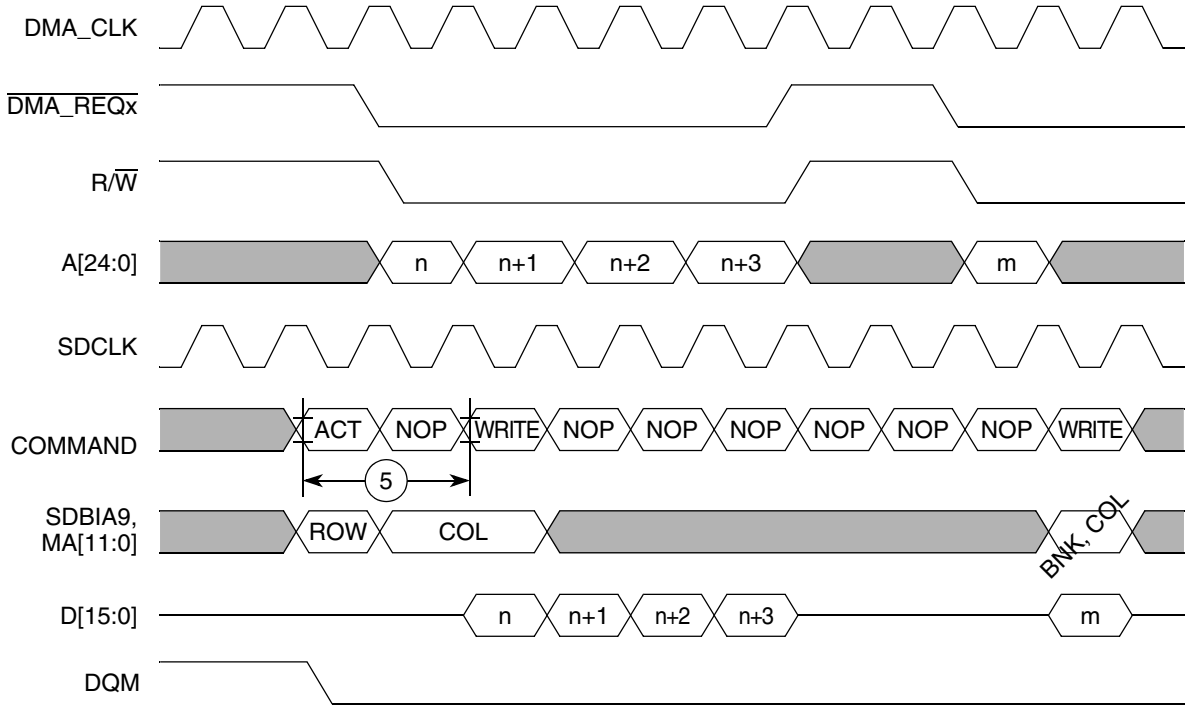


Figure 26-22. Off-Page Burst Write Timing Diagram

26.4.3.4.3 Burst Write Followed by On-Page Read

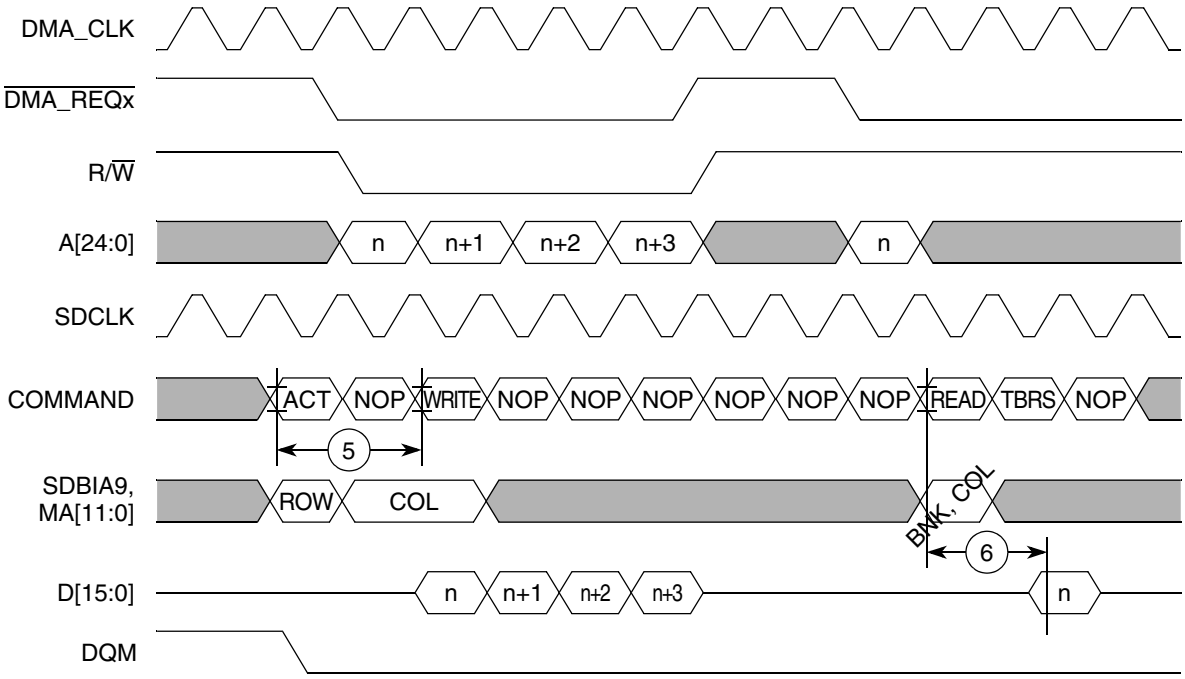


Figure 26-23. Burst Write Followed by On-Page Read Timing Diagram

26.4.3.4.4 Burst Read Followed by On-Page Write

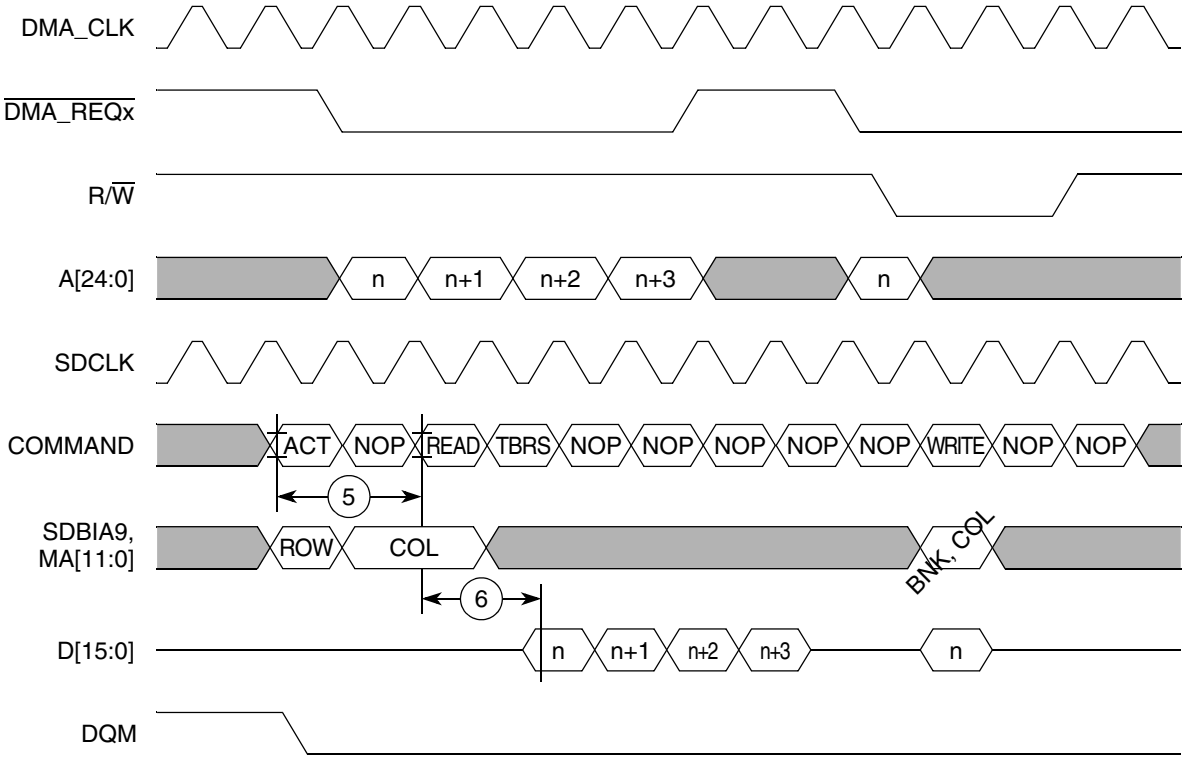


Figure 26-24. Burst Read Followed by On-Page Write Timing Diagram

Table 26-11. Timing Parameters for Figure 26-19 Through Figure 26-24

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Clock high pulse time	7	—	ns
2	Clock low pulse time	7	—	ns
3	Clock high to address valid	3	13	ns
4	Clock high to $\overline{\text{DMA\_REQx}}$ asserted	3	12	ns
5	Active command to read/write command (bits SRCD[1:0] = 00, 01, 10, 11)	60, 15, 30, 45	—	ns
6	Read tp data sample latency (bits SCL[1:0] = 01,10,11)	15, 30, 45	—	ns

26.4.4 EDO DRAM Operation

EDO RAM has no block-crossing problem, and its bank is always precharged after access. These features make the EDO controller much simpler than the SDRAM controller. However, the design should take care of page crossing during burst access in DMAC/LCDC access.

26.4.4.1 68K Mode Operation

In read operation, the EDO controller responds with the falling edge of $\overline{\text{RASx}}$. Please note that the CPU clock, CPUCLK, could be lower than or equal to the DMA clock, DMA_CLK. If the CPU clock is set as equal to the DMA clock, as in the timing diagram Figure 26-25, the EDO controller will assert $\overline{\text{RASx}}$ at s4. Keeping $\overline{\text{RASx}}$ active ensures the data's validity throughout the CPU data sample state, falling edge of s6. Keeping $\overline{\text{DTACK}}$ active ensures that the CPU in a lower CPU clock can recognize the $\overline{\text{DTACK}}$.

26.4.4.1.1 68K Read Cycle

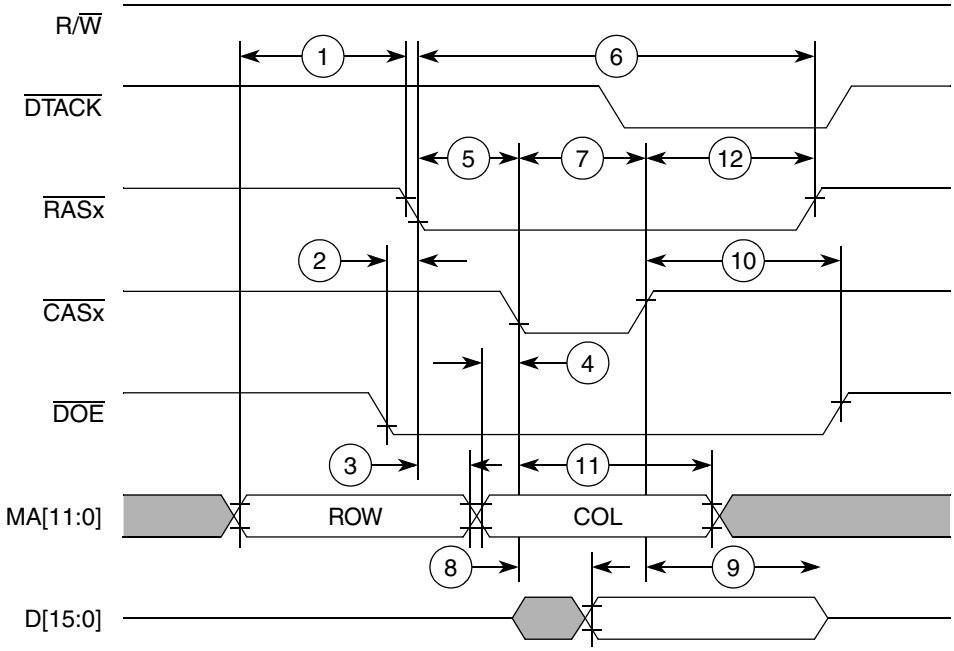


Figure 26-25. 68K Read Cycle

Table 26-12. 68K Read Cycle 16-Bit Access Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Row address valid to RASx asserted	11	—	ns
2	RASx asserted before DOE is asserted (ETRC=00, 01, 10, 11)	60, 15, 30, 45	—	ns
3	RASx asserted before row address invalid	6	13	ns
4	Column address valid to CASx asserted (ETRC = 00, 01, 10, 11)	46, 9, 16, 31	—	ns
5	RASx asserted to CASx asserted	ETRC	—	ns
6	RASx pulse width	ETRC + ETC + 4	—	ns
7	CASx pulse width	ETC	—	ns
8	CASx asserted to data-in valid (ETC[1:0] = 00, 01, 10, 11)	—	60, 14, 30, 45	ns
9	Data-in hold after CASx is negated	29	—	ns
10	DOE negated after CASx is negated	14	—	ns
11	CASx asserted before column address invalid (ETC = 00, 01, 10, 11)	90, 45, 60, 75	—	ns
12	RASx negated after CASx is negated	14	—	ns

26.4.4.1.2 68K Write Cycle

Figure 26-26 shows the 16-bit CPU write timing. Since it is write access and the data is already presented at the data bus early, the CPU write cycle can be shortened and no wait state is inserted.

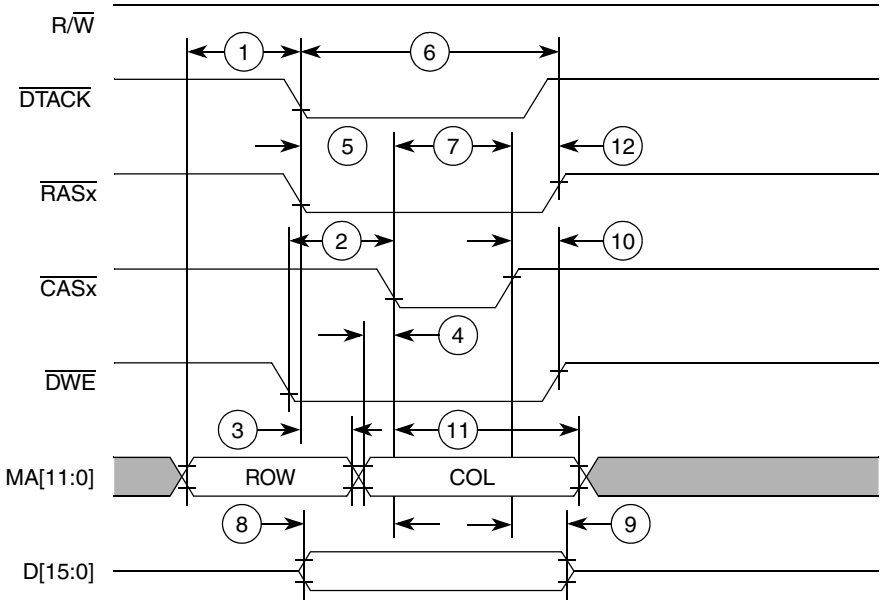


Figure 26-26. 68K 16-Bit Write Cycle

Table 26-13. 68K Write Cycle 16-Bit Access Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Row address valid to $\overline{RASx}$ asserted	11	—	ns
2	$\overline{DWE}$ asserted before $\overline{CASx}$ asserted (ETRC = 00, 01, 10, 11)	59, 14, 30, 44	—	ns
3	$\overline{RASx}$ asserted before row address invalid (ETRC = 00, 01, 10, 11)	13, 6, 13, 13	—	ns
4	Column address valid to $\overline{CASx}$ asserted (ETRC = 00, 01, 10, 11)	49, 9, 16, 31	—	ns
5	$\overline{RASx}$ asserted to $\overline{CASx}$ asserted	ETRC	—	ns
6	$\overline{RASx}$ pulse width	ETRC + ETC + 4	—	ns
7	$\overline{CASx}$ pulse width	ETC	—	ns
8	Data-out valid before $\overline{CASx}$ asserted	31	—	ns
9	Data-out hold after $\overline{CASx}$ negated	13	—	ns
10	$\overline{DWE}$ negated after $\overline{CASx}$ negated	15	—	ns
11	$\overline{CASx}$ asserted before column address invalid (ETC=00,01,10,11)	70, 25, 40, 55	—	ns
12	$\overline{RASx}$ negated after $\overline{CASx}$ negated	14	—	ns

26.4.4.2 DMAC/LCDC Mode Operation

26.4.4.2.1 Burst Read Cycle

Figure 26-27 shows a DMAC/LCDC burst read of 2. The $\overline{\text{RASx}}$ will not go high between data within a burst. $\overline{\text{DOE}}$ asserts at the same time as $\overline{\text{CASx}}$, as shown at (1). It continuously applies until the end and negates at the same time as $\overline{\text{RASx}}$, as shown at (2). See Table 26-14 for timing parameters.

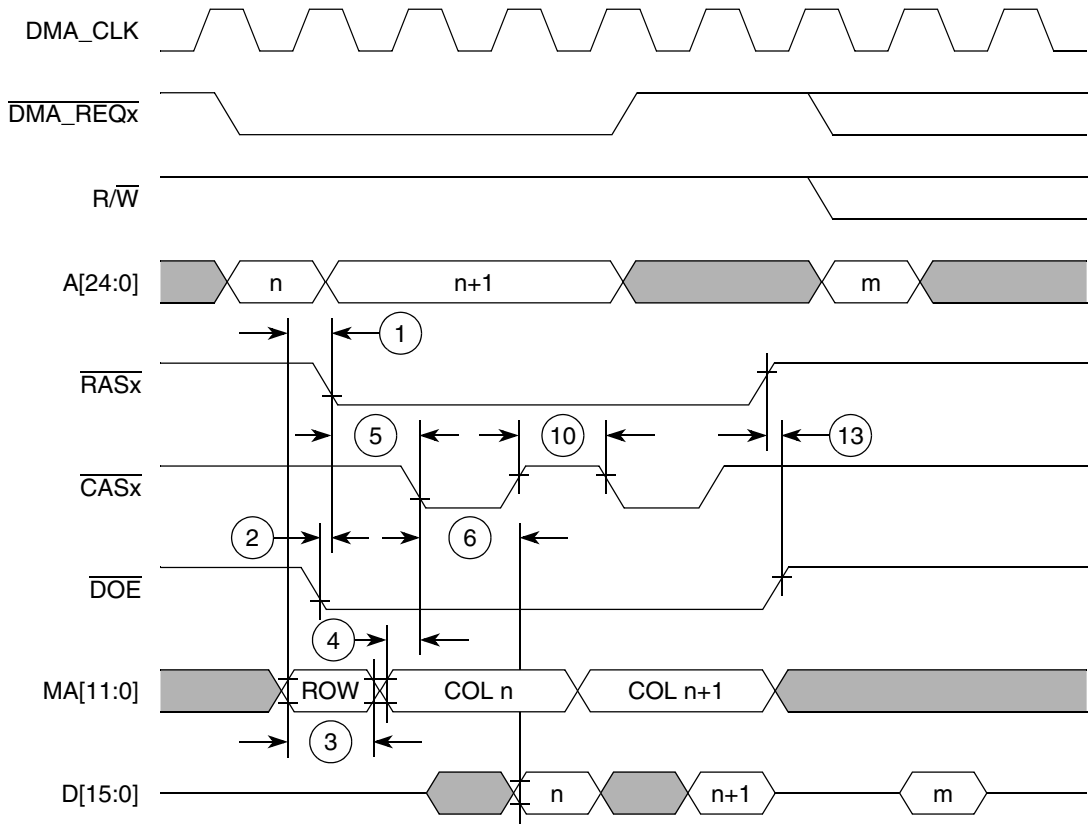


Figure 26-27. DMAC/LCDC Burst Read*

*Burst read of 2 followed by a possible write. There is a 1-clock bus turnaround margin.

26.4.4.2.2 Burst Read Cycle (Miss-Page in Middle)

The difference between Figure 26-27 and Figure 26-28 is that the second data in the burst is miss-page. The controller's page hit module detects it and tells the state machine to assert wait state at (1). Also, $\overline{\text{RASx}}$ needs to go high to precharge the DRAM in order to activate the page where the second data is located. In this example, EPTR is 1; and then $\overline{\text{RASx}}$ asserts at (2) after 2 clocks of high.

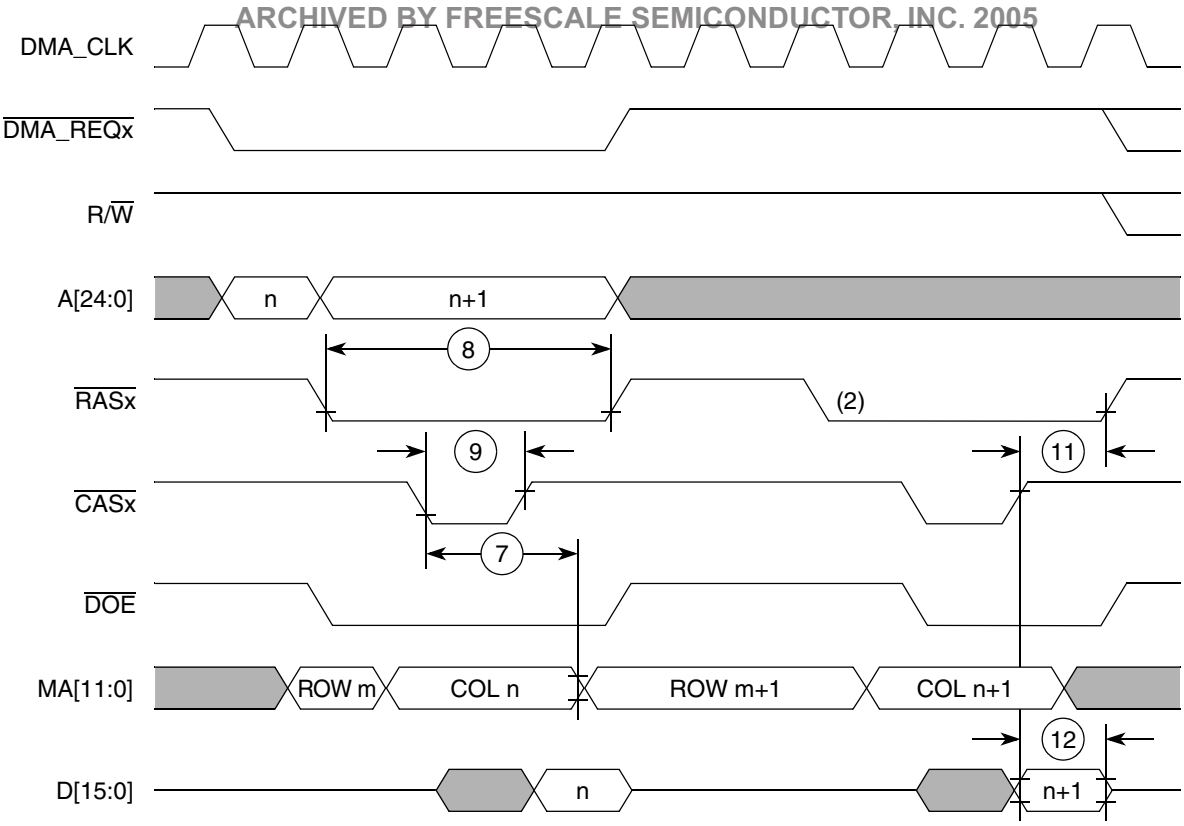


Figure 26-28. DMAC/LCDC Burst Read (Miss-Page in Middle)*

*Burst read of 2 with the second read data located on the next page. Row precharge and activation occurs by rising edge and falling edge of RASx signal.

Table 26-14. DMAC/LCDC Read Cycle 16-Bit Access Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Row address valid to RASx asserted	15	—	ns
2	RASx asserted before DOE asserted (ETRC=00, 01, 10, 11)	60, 15, 30, 45	—	ns
3	RASx asserted before row address invalid (ETRC = 00, 01, 10, 11))	13, 6, 13, 13	—	ns
4	Column address valid to CASx asserted (ETRC = 00, 01, 10, 11)	45, 8, 16, 30	—	ns
5	RASx asserted to CASx asserted	ETRC	—	ns
6	CASx asserted to data-in valid (ETRC = 00, 01, 10, 11)	—	60, 15, 30, 45	ns
7	CASx asserted before column address invalid (ETC = 00, 01, 10, 11)	60, 15, 28, 45	—	ns
8	RASX pulse width	ETRC + ETC + 4	—	ns

Table 26-14. DMAC/LCDC Read Cycle 16-Bit Access Timing Parameters (Continued)

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
9	$\overline{\text{CASx}}$ pulse width	ETC	—	ns
10	$\overline{\text{CASx}}$ precharge time	14	—	ns
11	$\overline{\text{CASx}}$ negated to $\overline{\text{RASx}}$ negated	14	—	ns
12	Data-in hold after $\overline{\text{CASx}}$ negated	29	—	ns
13	$\overline{\text{DOE}}$ negated after $\overline{\text{CASx}}$ negated	14	—	ns

26.4.4.2.3 Burst Write Cycle (DMAC Only)

An on-page write burst of 2 is shown in Figure 26-29. Normally the fastest EDO cycle needs 2 clocks per data. The basic element is a combination of 1 clock of $\overline{\text{CASx}}$ and 2 clocks of $\overline{\text{RASx}}$. A burst consists of many of these elements, and $\overline{\text{RASx}}$ does not negate in the middle of the burst because the fastest way it operates is using page mode. However, when a page miss occurs, $\overline{\text{RASx}}$ will negate for precharging the DRAM. The $\overline{\text{RASx}}$ must negate at the burst end. Typically, at the end of the burst, $\overline{\text{RASx}}$ negates 1 clock after $\overline{\text{CASx}}$ negates. See Table 26-15 for timing parameters.

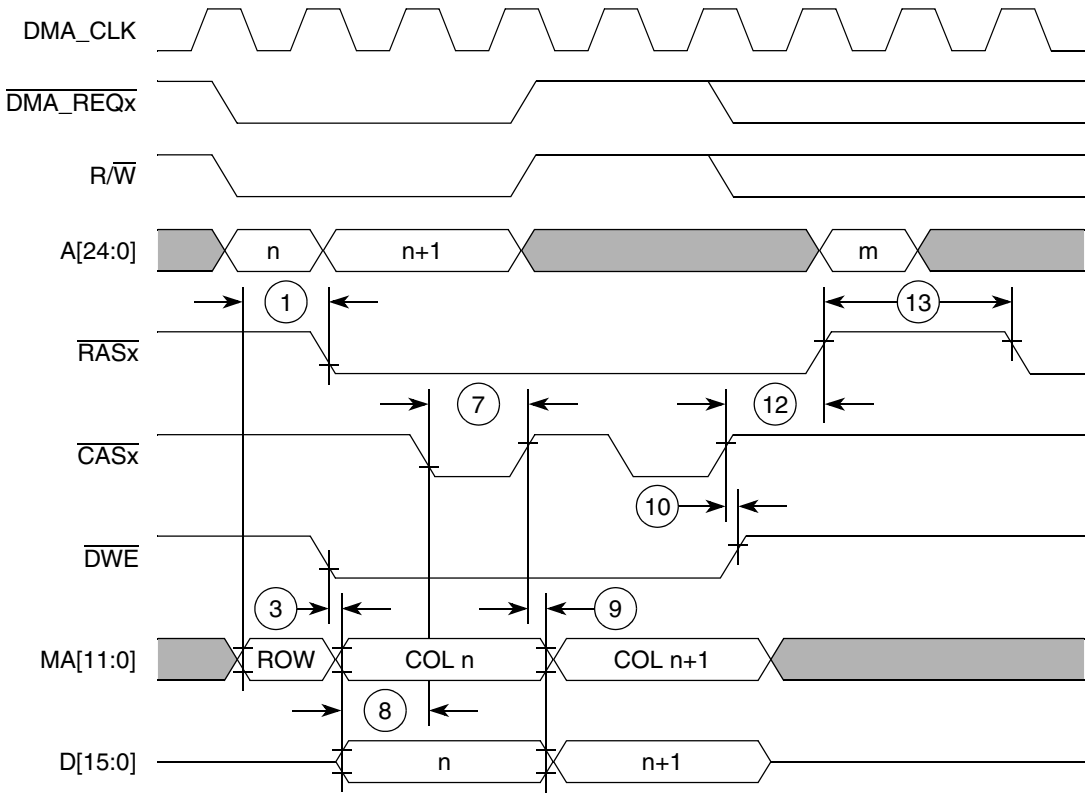


Figure 26-29. DMA Burst of 2 Data Write Followed by Possible Write

26.4.4.2.4 Burst Write Cycle (Miss-Page in the Middle) (DMAC Only)

Figure 26-30 shows a write burst of 2 with the second data located in the other page. $\overline{\text{RASx}}$ goes high to precharge the DRAM. It retutns to low to activate the page and start the cycle for read, as in the first data. Similar to a read operation, the row address for the second data must be issued when the corresponding page is being activated.

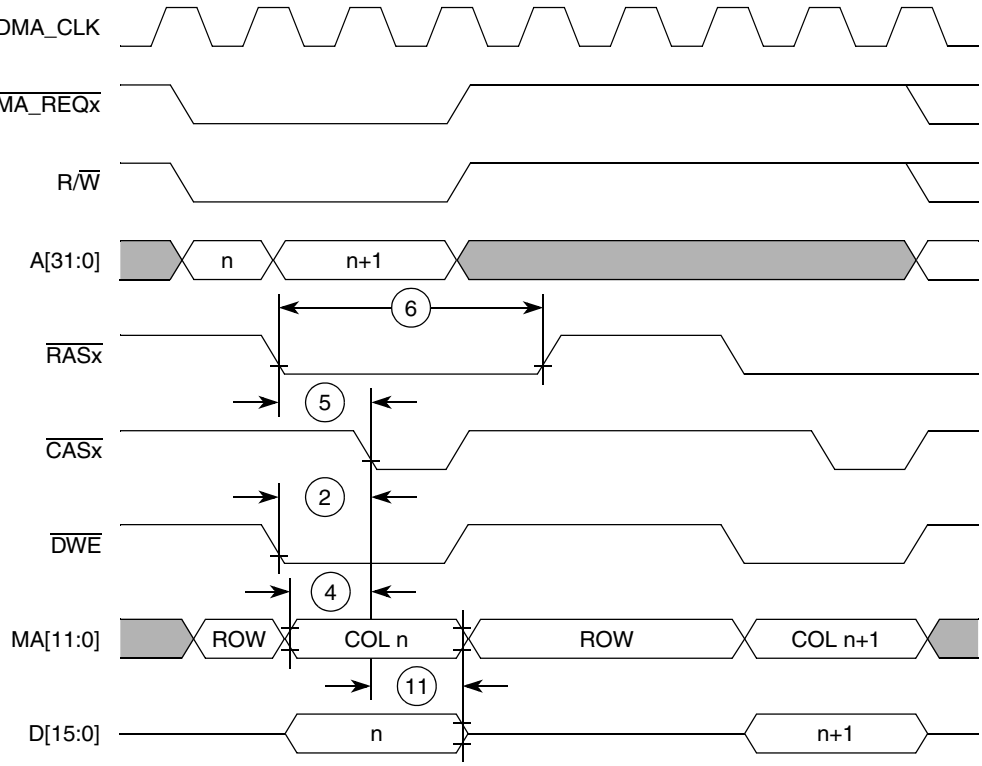


Figure 26-30. DMA Write Burst of 2 with Page-Miss at Second Data

Table 26-15. DMAC Write Cycle 16-Bit Access Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Row address valid to $\overline{\text{RASx}}$ asserted	15	—	ns
2	$\overline{\text{DWE}}$ asserted before $\overline{\text{CASx}}$ asserted (ETRC = 00, 01, 10, 11)	60, 14, 29, 45	—	ns
3	$\overline{\text{RASx}}$ asserted before row address invalid (ETRC = 00, 01, 10, 11)	13, 6, 13, 13	—	ns
4	Column address valid to $\overline{\text{CASx}}$ asserted (ETRC = 00, 01, 10, 11)	45, 7, 15, 30	—	ns
5	$\overline{\text{RASx}}$ asserted to $\overline{\text{CASx}}$ asserted	ETRC	—	ns
6	$\overline{\text{RASx}}$ pulse width	ETRC + ETC + 4	—	ns
7	$\overline{\text{CASx}}$ pulse width	ETC	—	ns
8	Data-out valid before $\overline{\text{CASx}}$ asserted	11	—	ns

Table 26-15. DMAC Write Cycle 16-Bit Access Timing Parameters (Continued)

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
9	Data-out hold after $\overline{\text{CASx}}$ negated	3	—	ns
10	$\overline{\text{DWE}}$ negated after $\overline{\text{CASx}}$ negated	15	—	ns
11	$\overline{\text{CASx}}$ asserted before column address invalid (ETC = 00,01,10,11)	58, 13, 28, 43	—	ns
12	$\overline{\text{RASx}}$ negated after $\overline{\text{CASx}}$ negated	14	—	ns
13	$\overline{\text{RASx}}$ precharge time	ETPR	—	ns

26.4.4.3 EDO DRAM Hidden Refresh Cycle

26.4.4.3.1 Hidden Refresh

Figure 26-31 shows the EDO DRAM hidden refresh cycle timing for power saving mode. The signal values and units of measure for this figure are found in Table 26-16. Detailed information about the operation of individual signals can be found in Chapter 8, “DRAM Controller.”

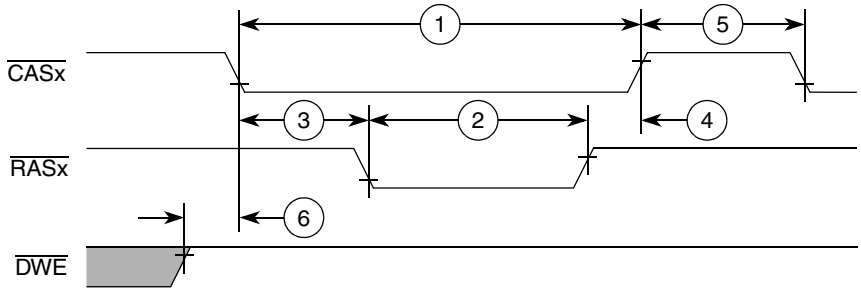


Figure 26-31. DRAM Hidden Refresh Cycle Timing Diagram

Table 26-16. DRAM Hidden Refresh Cycle Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	$\overline{\text{CASx}}$ pulse width (ETRAS = 00,01,10,11)	120, 45, 60, 90	—	ns
2	$\overline{\text{RASx}}$ pulse width	ETRAS	—	ns
3	$\overline{\text{CASx}}$ asserted to $\overline{\text{RASx}}$ asserted	14	—	ns
4	$\overline{\text{CASx}}$ negated to $\overline{\text{RASx}}$ negated	14	—	ns
5	$\overline{\text{CASx}}$ negated to next $\overline{\text{CASx}}$ asserted	ETPR + ETRC	—	ns
6	$\overline{\text{DWE}}$ negated before $\overline{\text{CASx}}$ asserted	59	—	ns

Note: $\overline{\text{RASx}}$ stands for $\overline{\text{RAS0}}$ and $\overline{\text{RAS1}}$. $\overline{\text{CASx}}$ stands for $\overline{\text{CAS0}}$ and $\overline{\text{CAS1}}$.

26.4.5 LCD Controller Timing

Figure 26-23 shows the 4/8/16 bits/pixel TFT color mode panel timing. Detailed information about the operation of individual signals can be found in Chapter 10, “LCD Controller.”

26.4.5.1 4/8/16 Bits/Pixel TFT Color Mode

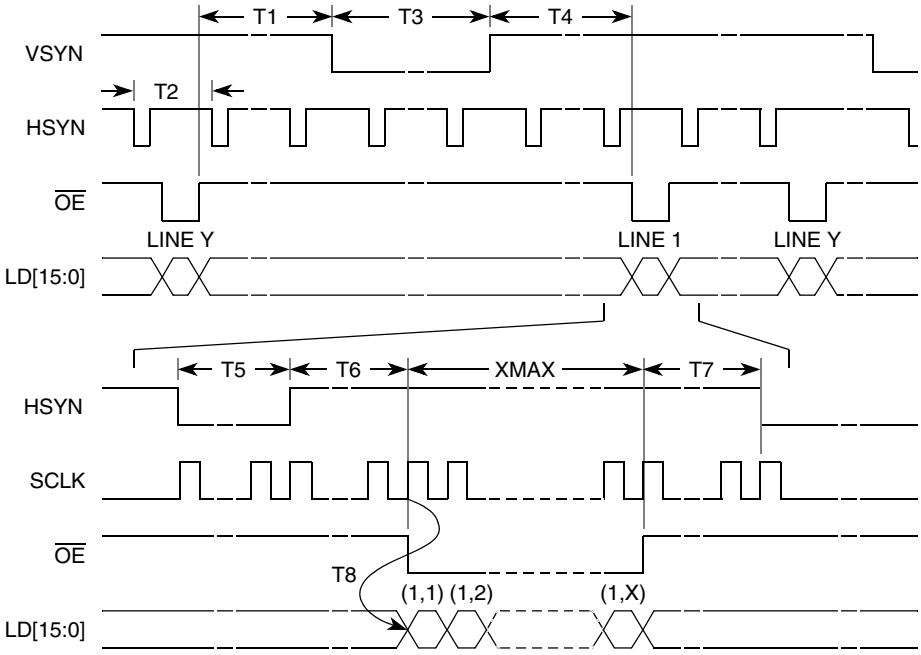


Figure 26-32. 4/8/16 Bits/Pixel TFT Color Mode Panel Timing Diagram

Table 26-17. 4/8/16 Bits/Pixel TFT Color Mode Panel Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
T1	End of OE to beginning of VSYN	5	VWAIT1*T2+HWAIT1+HWIDTH+HWAIT2	Ts
T2	HSYN period (T5+T6+XMAX+T7)	XMAX+5	HWAIT2+HWIDTH+XMAX+HWAIT1	Ts
T3	VSYN pulse width	T2	VWIDTH*T2	Ts
T4	End of VSYN to beginning of OE	2	VWAIT2*T2+2	Ts
T5	HSYN pulse width	2	HWIDTH	Ts
T6	End of HSYN to beginning of OE	2	HWAIT2	Ts
T7	End of OE to beginning of HSYN	1	HWAIT1	Ts
T8	SCLK to valid LD data	0	3	ns

Note:
- Ts is the SCLK period which equals LCDC_CLK / (PCD + 1). Normally LCDC_CLK = 15ns.
- VSYN, HSYN and OE can be programmed as active high or active low. In the timing diagram, all these 3 signals are active low.
- The polarity of SCLK and LD[15:0] can also be programmed.
- SCLK can be programmed to be deactivated in VSYN pulse or OE deasserted period. In the above timing diagram, SCLK is always active.

26.4.5.2 8 Bit/Pixel CSTN Panel Mode

Figure 26-33 shows the 8 bit/pixel CSTN color mode panel timing diagram. Detailed information about the operation of individual signals can be found in Chapter 10, "LCD Controller."

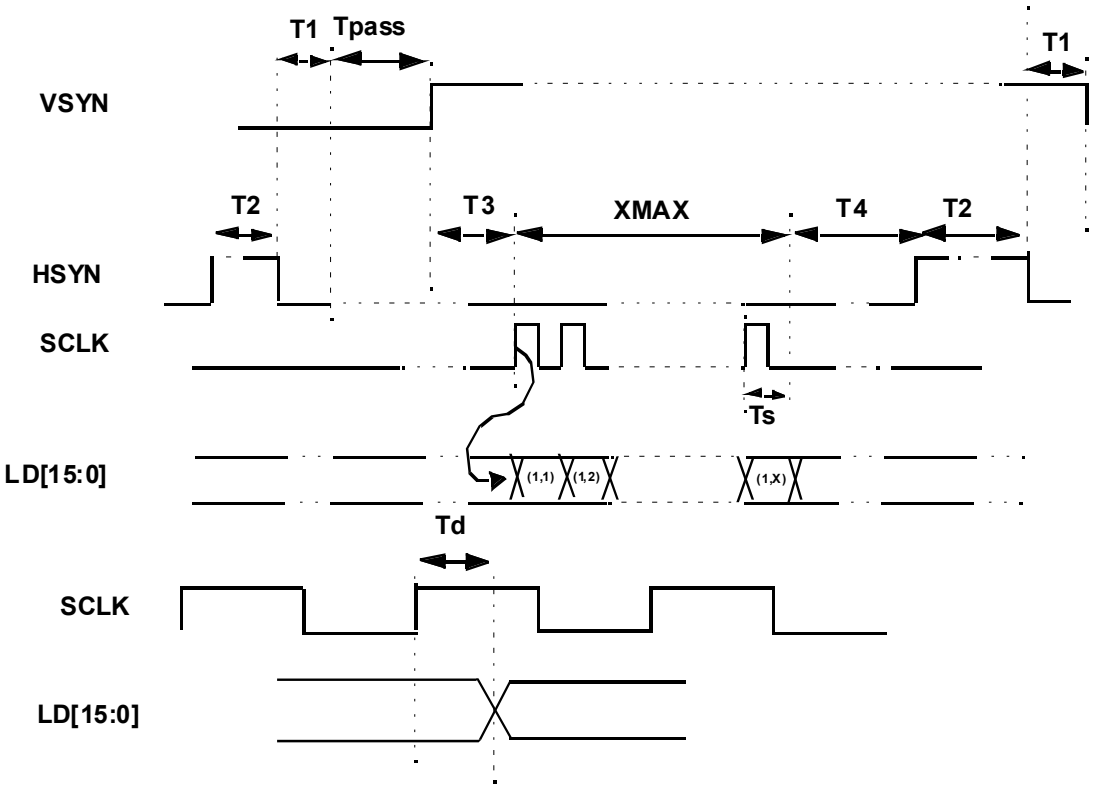


Figure 26-33. 8 Bit/Pixel CSTN Color Mode Panel Timing Diagram

Table 26-18. 8 Bits/Pixel CSTN Color Mode Panel Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
T1	HSYN to VSYN falling edge delay	2	HWAIT2+2	Tpix
T2	HSYN pulse width	1	HWIDTH+1	Tpix
T3	VSYN to SCLK	4	4	Tpix
T4	SCLK to HSYN	2	HWAIT1+1	Tpix
Td	SCLK to valid LD data	—	3	ns
Tpass	Last Line HSYN to VSYN	2	VWAIT1*Tvclk+Tol+2TPix	—

Note:
- VSYN, HSYN and SCLK can be programmed as active high or active low. In the timing diagram, all 3 signals are active high.
- Ts is the shift clock period.
- Tpix is the pixel period which equals (LCDC_CLK period * (PCD=1)).
- Ts = Tpix * pbsize, (panel data bus width in number of bits)
- Tvclk = Ts * (pass_div +1)
- Tol = var*Tpix, where var equals ((T1+T2+T4-1) modulus 8), i.e. max is one Ts.

26.4.5.3 1/2/4 Bit/Pixel STN Panel Mode

Figure 26-34 shows the 1/2/4 bit/pixel STN color mode panel timing diagram. Detailed information about the operation of individual pixels can be found in Chapter 10, "LCD Controller."

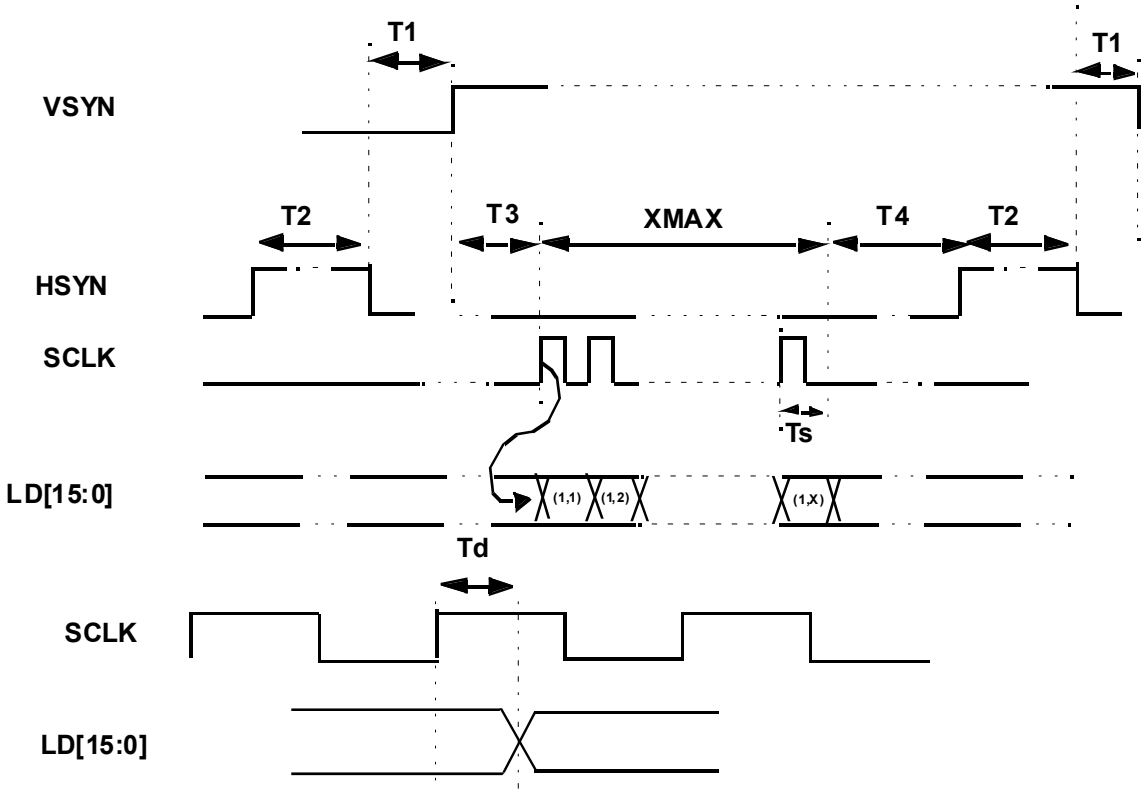


Figure 26-34. 1/2/4 Bit/Pixel STN Color Mode Panel Timing Diagram

Table 26-19. 1/2/4 Bit/Pixel STN Color Mode Panel Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
T1	HSYN to VSYN delay	2	HWAIT2+2	Tpix
T2	HSYN pulse width	1	HWIDTH+1	Tpix
T3	VSYN to SCLK	4	4	Tpix
T4	SCLK to HSYN	2	HWAIT1+1	Tpix
Td	SCLK to valid LD data	—	3	ns

Note:
1. VSYN, HSYN and SCLK can be programmed as active high or active low. In the above timing diagram, all 3 signals are active high.
2. Ts is the shift clock period.
3. Tpix is the pixel period which equals (LCDC_CLK period / (PCD + 1)).
4. Ts = Tpix * pbsize, (panel data bus width in number of bits)

26.4.6 CSPI Generic Timing

Figure 26-35 shows the timing diagram for CSPI. The signal values and units of measure for Figure 26-35 through Figure 26-40 are found in Table 26-20 on page 26-33. Detailed information about the operation of individual signals can be found in Chapter 20, "Configurable Serial Peripheral Interface."

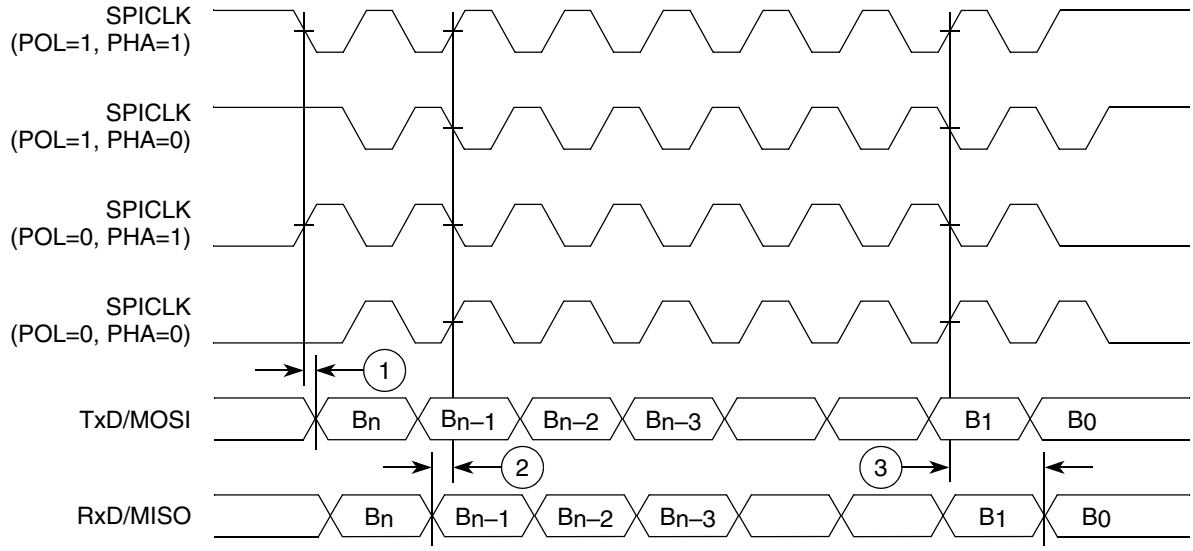


Figure 26-35. CSPI Generic Timing Diagram

26.4.7 CSPI Master Using $\overline{\text{DATA\_READY}}$ Edge Trigger

Figure 26-36 shows the timing diagram for the CSPI master using the $\overline{\text{DATA\_READY}}$ edge trigger. The signal values and units of measure for Figure 26-35 through Figure 26-40 are found in Table 26-20 on page 26-33. Detailed information about the operation of individual signals can be found in Chapter 20, "Configurable Serial Peripheral Interface."

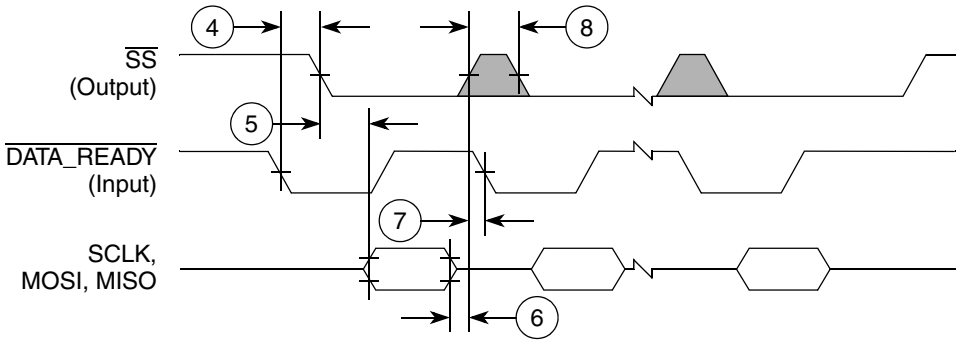


Figure 26-36. CSPI Master Using $\overline{\text{DATA\_READY}}$ Edge Trigger Timing Diagram

26.4.8 CSPI Master Using $\overline{\text{DATA\_READY}}$ Level Trigger

Figure 26-37 shows the timing diagram for the CSPI master using the $\overline{\text{DATA\_READY}}$ level trigger. The signal values and units of measure for Figure 26-35 through Figure 26-40 are found in Table 26-20 on page 26-33. Detailed information about the operation of individual signals can be found in Chapter 20, "Configurable Serial Peripheral Interface."

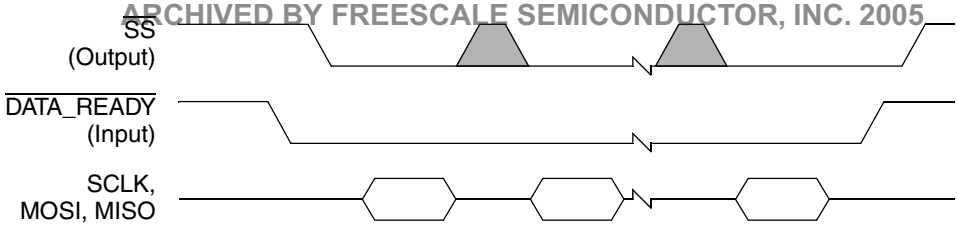


Figure 26-37. CSPI Master Using DATA_READY Level Trigger Timing Diagram

26.4.9 CSPI Master "Don't Care" DATA_READY

Figure 26-38 shows the timing diagram for the CSPI master with "don't care" DATA_READY. The signal values and units of measure for Figure 26-35 through Figure 26-40 are found in Table 26-20 on page 26-33. Detailed information about the operation of individual signals can be found in Chapter 20, "Configurable Serial Peripheral Interface."

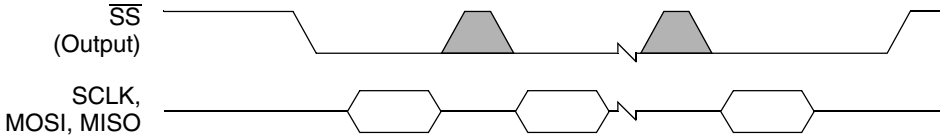


Figure 26-38. CSPI Master "Don't Care" DATA_READY Timing Diagram

26.4.10 CSPI Slave FIFO Advanced by Bit Count

Figure 26-39 shows the timing diagram for the CSPI slave FIFO advanced by bit count. The signal values and units of measure for Figure 26-35 through Figure 26-40 are found in Table 26-20 on page 26-33. Detailed information about the operation of individual signals can be found in Chapter 20, "Configurable Serial Peripheral Interface."

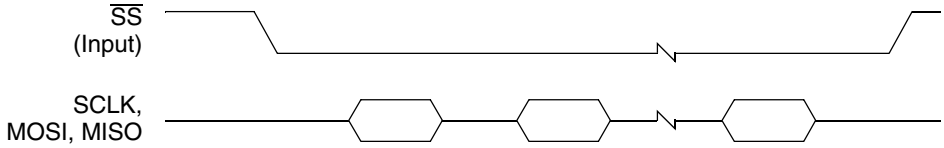


Figure 26-39. CSPI Slave FIFO Advanced by Bit Count Timing Diagram

26.4.11 CSPI 1 Slave FIFO Advanced by SS Rising Edge

Figure 26-40 shows the timing diagram for the CSPI slave FIFO advanced by SS rising edge. The signal values and units of measure for Figure 26-35 through Figure 26-40 are found in Table 26-20 on page 26-33. Detailed information about the operation of individual signals can be found in Chapter 20, "Configurable Serial Peripheral Interface."

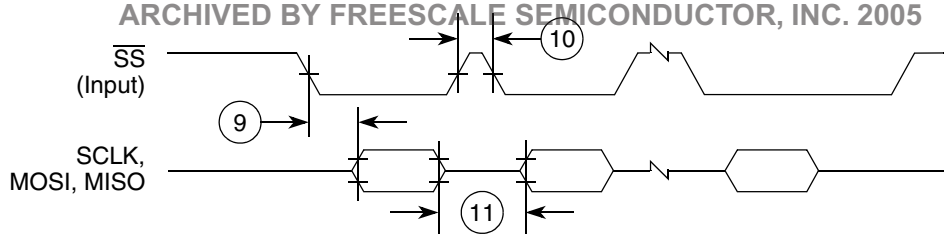


Figure 26-40. CSPI Slave FIFO Advanced by SS Rising Edge Timing Diagram

Table 26-20. Timing Parameters for Figure 26-35 through Figure 26-40

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Clock edge to TxD data ready	—	0.5	T
2	RxD data ready to clock edge	0.3	—	T
3	Clock edge to RxD data hold time	0.3	—	T
4	DATA_READY to SS output low	—	2	T
5	SS output low to first SCLK edge	1	—	T
6	Last SCLK edge to SS output high	0.5	—	T
7	SS output high to DATA_READY low	0.5	—	T
8	SS output pulse width	1	—	T
9	SS input low to first SCLK edge	1	—	T
10	SS input pulse width	0	—	T
11	Pause between data word	0	—	T

Note:
T = SPI system clock period
WAIT = Number of SYSCLK or 32.768 kHz clocks per sample period control register

26.4.12 I²C Input/Output Timing Characteristics

Table 26-21 lists specifications for the I²C input timing parameters shown in Figure 26-41.

Table 26-21. I²C Input Timing Specifications Between SCL and SDA

Parameter	Symbol	66 MHz CPUCLK		Unit
		Min.	Max.	
Start condition hold time	I_1	2	—	Bus clocks
Clock low period	I_2	6	—	Bus clocks
SCL/SDA rise time (V_{IL}=0.5V to V_{IH}=2.4V)	I_3	—	1	ms
Data hold time	I_4	0	—	ns

Table 26-21. I^2C Input Timing Specifications Between SCL and SDA (Continued)

Parameter	Symbol	66 MHz CPUCLK		Unit
		Min.	Max.	
SCL/SDA fall time (V_{IH}=2.4V to V_{IL}=0.5V)	I_5	—	1	ms
Clock high time	I_6	—	—	Bus clocks
Data setup time	I_7	11	—	ns
Start condition setup time (for repeated start condition only)	I_8	2	—	Bus clocks
Stop condition setup time	I_9	2	—	Bus clocks

Table 26-22 list specifications for the I^2C output timing parameters shown in Figure 26-41.

Table 26-22. I^2C Output Timing Specifications Between SCL and SDA

Parameter	Symbol	66 MHz CPUCLK		Unit
		Min.	Max.	
Start condition hold time	I_1[1]	11	—	Bus clocks
Clock low period	I_2[1]	11	—	Bus clocks
SCL/SDA rise time (V_{IL}=0.5V to V_{IH}=2.4V)	I_3[1]	—	—	µs
Data hold time	I_4[1]	8	—	Bus clocks
SCL/SDA fall time (V_{IH}=2.4V to V_{IL}=0.5V)	I_5[1]	—	3	ns
Clock high time	I_6[1]	11	—	Bus clocks
Data setup time	I_7[1]	3	—	Bus clocks
Start condition setup time (for repeated start condition only)	I_8[1]	21	—	Bus clocks
Stop condition setup time	I_9[1]	12	—	Bus clocks

NOTE:

1. Programming IFDR with the maximum frequency (IFDR = 0x20) results in the minimum output timings listed here. The I^2C interface is designed to scale the data transition time, moving it to the middle of the SCL low period. The actual position is affected by the prescale and division values programmed in IFDR.
2. Because SCL and SDA are open-collector-type outputs, which the processor can only actively drive low, the time SCL or SDA takes to reach a high level depends on external signal capacitance and pull-up resistor values.
3. Specified at a nominal 30 pF load.

Figure 26-41 shows timing for the values in Table 26-21 and Table 26-22.

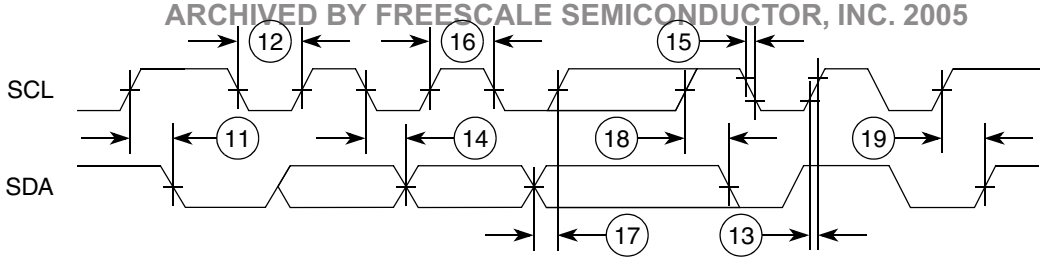


Figure 26-41. I^2C Input/Output Timings

26.4.13 USB Device Timing Characteristics

Figure 26-42 and Figure 26-43 show USB device timing. The signal values and units of measure for this figure are found in Table 26-23 and Table 26-24. Table 26-25 is the USB device truth table. Detailed information about the operation of individual signals can be found in Chapter 21, "USB Device Module."

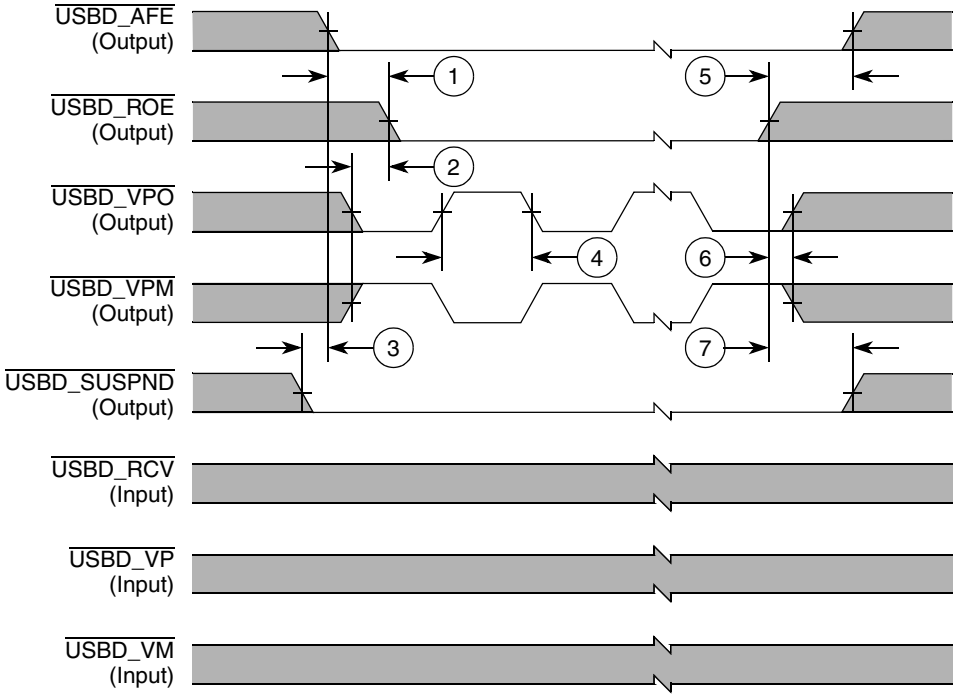


Figure 26-42. USB Device Timing for Data Transfer to USB Transceiver (TX)

Table 26-23. USB Timing Parameters for Data Transfer to USB Transceiver (TX)

Number	Characteristic	(3.0 ± 0.3) V			Unit
		Minimum	Typical	Maximum	
1	USBD_AFE assert to $\overline{\text{USBD\_ROE}}$ active	These timings depend on the USB transmitter specifications			
2	VPO stable time before ROE enable or VPO stable time after ROE disable	—	83.3	—	ns
3	ROE enable to first data transfer or last data transfer to ROE disable	—	83.3	—	ns

Table 26-23. USB Timing Parameters for Data Transfer to USB Transceiver (TX) (Continued)

Number	Characteristic	(3.0 ± 0.3) V			Unit
		Minimum	Typical	Maximum	
4	VPM stable time before ROE enable or VPM stable time after ROE disable	—	83.3	—	ns

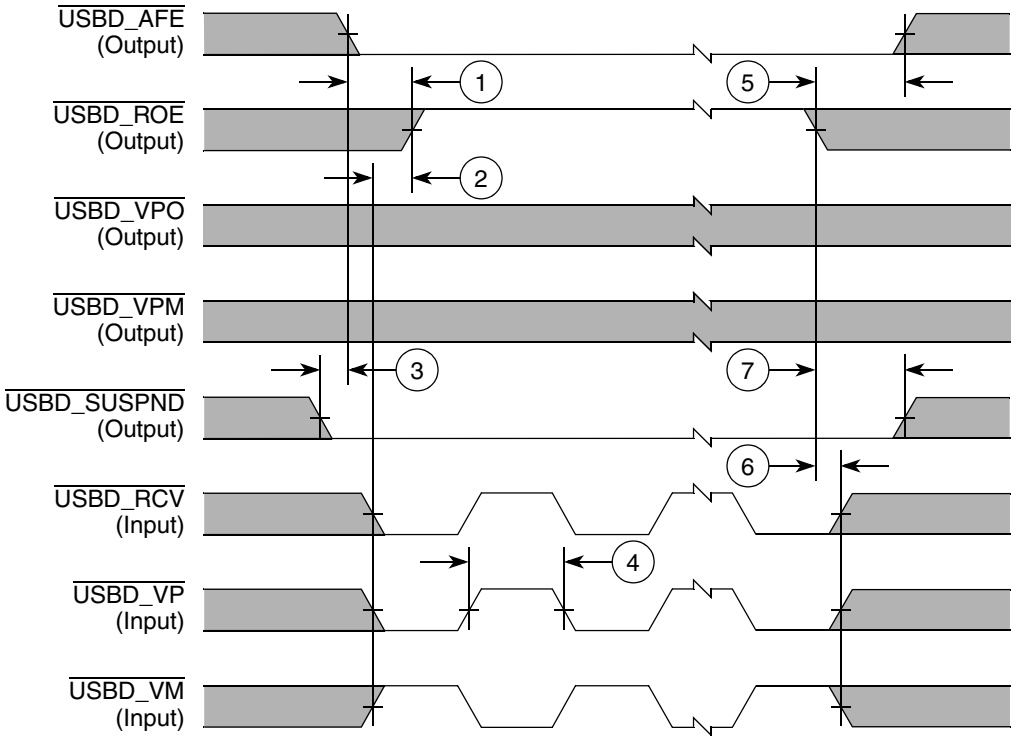


Figure 26-43. USB Device Timing for Data Transfer from USB Transceiver (RX)

Table 26-24. USB Timing Parameters for Data Transfer from USB Transceiver (RX)

Number	Characteristic	(3.0 ± 0.3) V			Unit
		Minimum	Typical	Maximum	
1	USBD_AFE assert to $\overline{\text{USBD\_ROE}}$ active	These timings depend on the USB transmiter specifications			
2	VP stable time before ROE enable or VP stable time after ROE disable	—	83.3	—	ns
3	ROE enable to first data transfer or last data transfer to ROE disable	—	83.3	—	ns
4	VM stable time before ROE enable or VM stable time after ROE disable	—	83.3	—	ns

Table 26-25. USB Device Truth Table

Mode	Signal Status				Result on USB Bus
	USBD_VP	USBD_VM	USBD_VPO	USBD_VPM	
$\overline{\text{USBD\_ROE}}$ = high (receive)	0	0	—	—	SEO
	0	1	—	—	Logic 0
	1	0	—	—	Logic 1
	1	1	—	—	Undefined
$\overline{\text{USBD\_ROE}}$ = high (transmit)	—	—	0	0	SEO
	—	—	0	1	Logic 0
	—	—	1	0	Logic 1
	—	—	1	1	Undefined

Note: Bidirectional drivers may use muxing scheme to drive VM/VP signals.

26.4.14 MMC/SD Timing Characteristics

Figure 26-44 shows the MMC/SD timing diagram. The signal values and units of measure for this figure are found in Table 26-26. Detailed information about the operation of individual signals can be found in Chapter 17, “Multimedia Card/Secure Digital Host Controller.”

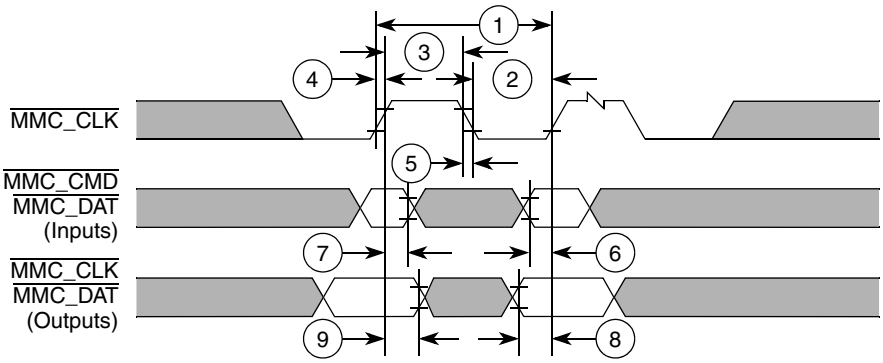


Figure 26-44. MMC/SD Timing Data

Table 26-26. MMC/SD Timing Data Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Clock Frequency Data Transfer Mode (PP) 10 cards	0	20	MHz
2	MMCSD_CLK Low time	10	—	ns
3	MMCSD_CLK high time	10	—	ns
4	MMCSD_CLK rise time	—	10	ns

Table 26-26. MMC/SD Timing Data Parameters (Continued)

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
5	MMCSD_CLK fall time	—	10	ns
6	MMCSD_CMD, MMCSD_DAT[3:0] Input Setup time	3	—	ns
7	MMCSD_CMD, MMCSD_DAT[3:0] Input Hold time	3	—	ns
8	MMCSD_CMD, MMCSD_DAT[3:0] Output Setup time	5	—	ns
9	MMCSD_CMD, MMCSD_DAT[3:0] Output Hold time	5	—	ns

26.4.15 MSHC Timing Characteristics

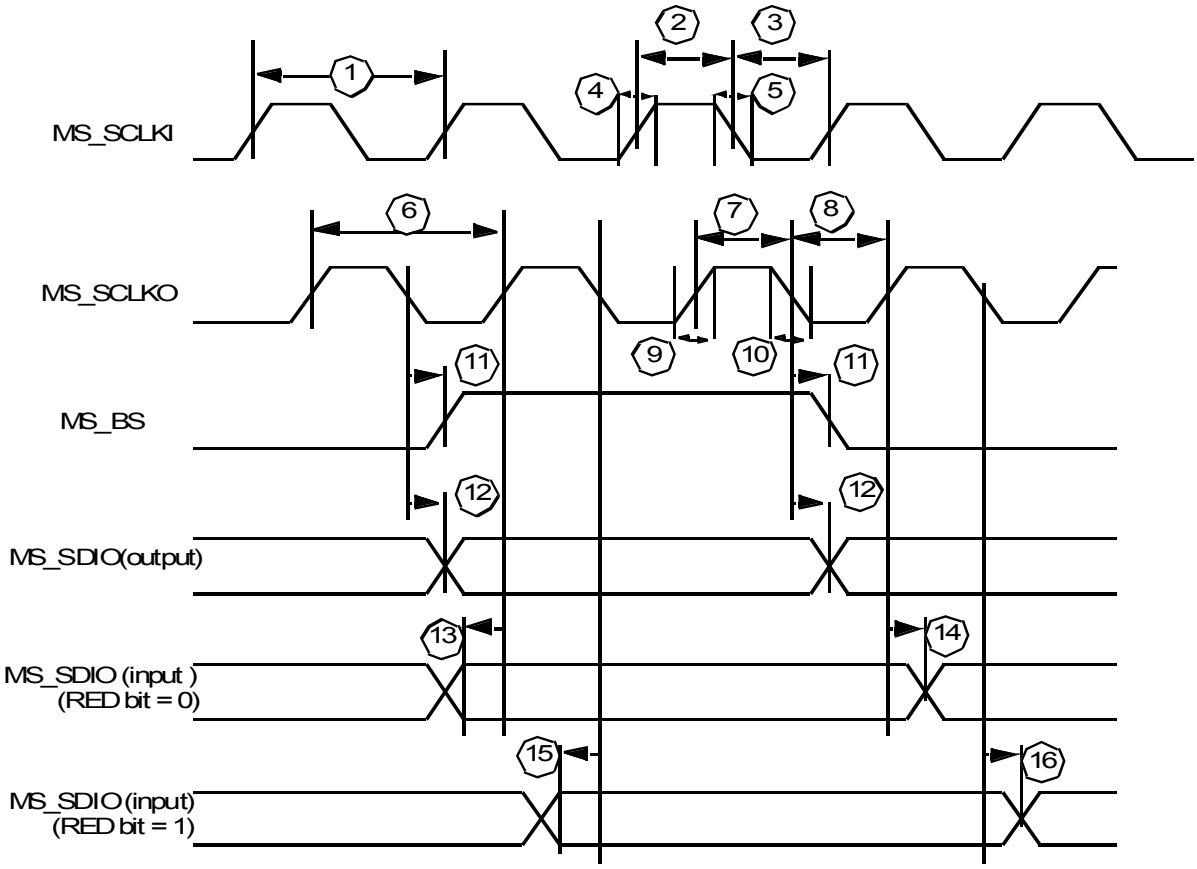


Figure 26-45. MSHC Signal Timing Diagram

Table 26-27. MSHC Timing Data Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	MS_SCLKI frequency	0	33	MHz
2	MS_SCLKI high pulse width	15	—	ns
3	MS_SCLKI low pulse width	15	—	ns
4	SCLKI rise time	—	3	ns
5	SCLKI fall time	—	3	ns
6	SCLKO frequency[1]	—	34	MHz
7	SCLKO high pulse width[1]	13	—	ns
8	SCLKO low pulse width[1]	15	—	ns
9	SCLKO rise time[1]	—	5	ns
10	SCLKO fall time[1]	—	5	ns
11	BS delay time[1]	—	3	ns
12	SDIO output delay time[1,2]	—	3	ns
13	SDIO input setup time for SCLKO rising edge (RED bit = 0)[3]	15	—	ns
14	SDIO input hold time for SCLKO rising edge (RED bit = 0)[3]	0	—	ns
15	SDIO input setup time for SCLKO falling edge (RED bit = 1)[4]	15	—	ns
16	SDIO input hold time for SCLKO falling edge (RED bit = 1)[4]	0	—	ns

1. Loading capacitor condition is less than or equal to 30pF.
2. An external resistor (100 – 200 Ohm) should be inserted for the current control in the series on MS_SDIO pin, because there is a possibility of causing signal conflict between the MS_SDIO pin and Memory Stick SDIO pin when the pin direction changes.
3. When used MSC2[RED] bit = 0, MSHC samples MS_SDIO input data at MS_SCLKO rising edge.
4. When used MSC2[RED] bit = 1, MSHC smaples MS_SDIO input data at MS_SCLKO falling edge.

26.4.16 ASP Characteristics

Table 26-28. Pen ADC Characteristics

Characteristic	(Temperature = 25°C, QVDD = 1900 mV)			Unit
	Minimum	Typical	Maximum	
Offset	—	32768	—	—
Offset Error	—	8199	—	—

Table 26-28. Pen ADC Characteristics (Continued)

Characteristic	(Temperature = 25°C, QVDD = 1900 mV)			Unit
	Minimum	Typical	Maximum	
Full-scale input span	0	—	QVDD	V
Absolute positive input voltage range	-0.2	—	1.9	V
Absolute negative input voltage range	-0.2	—	1.9	V
Absolute positive input current range	-2.5	—	9.5	µA
Absolute negative input current range	-2.5	—	9.5	µA
Resolution	—	—	8	bits
Throughput rate	—	—	9.5	kHz
Bandgap (NM)	—	200	—	mV

Chapter 27
Mechanical Data and Ordering Information

This chapter provides mechanical data, including illustrations, and ordering information.

27.1 Ordering Information

Table 27-1 provides ordering information for the 196-lead mold array process ball grid array (MAPBGA) package.

Table 27-1. MC68SZ328 Ordering Information

Package Type	Frequency (MHz)	Temperature	Order Number
196-lead MAPBGA	66	0 ^{O}C to 70 ^{O}C	MC68SZ328VH66V

27.2 MAPBGA Pin Assignments

Figure 27-1 provides a top view of the MAPBGA pin assignments and Figure 27-2 provides a bottom view.

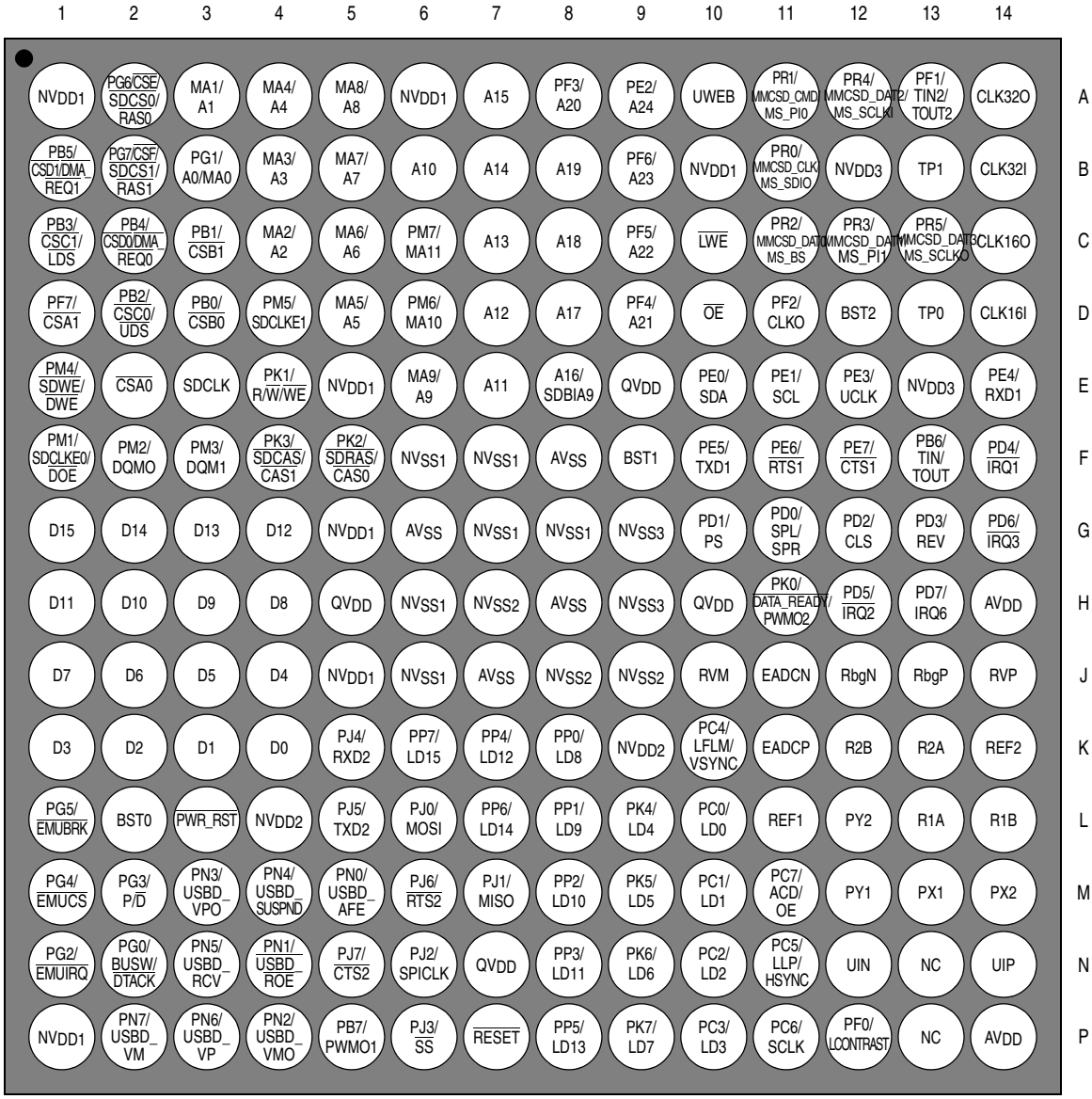


Figure 27-1. MC68SZ328 MAPBGA Pin Assignments—Top View

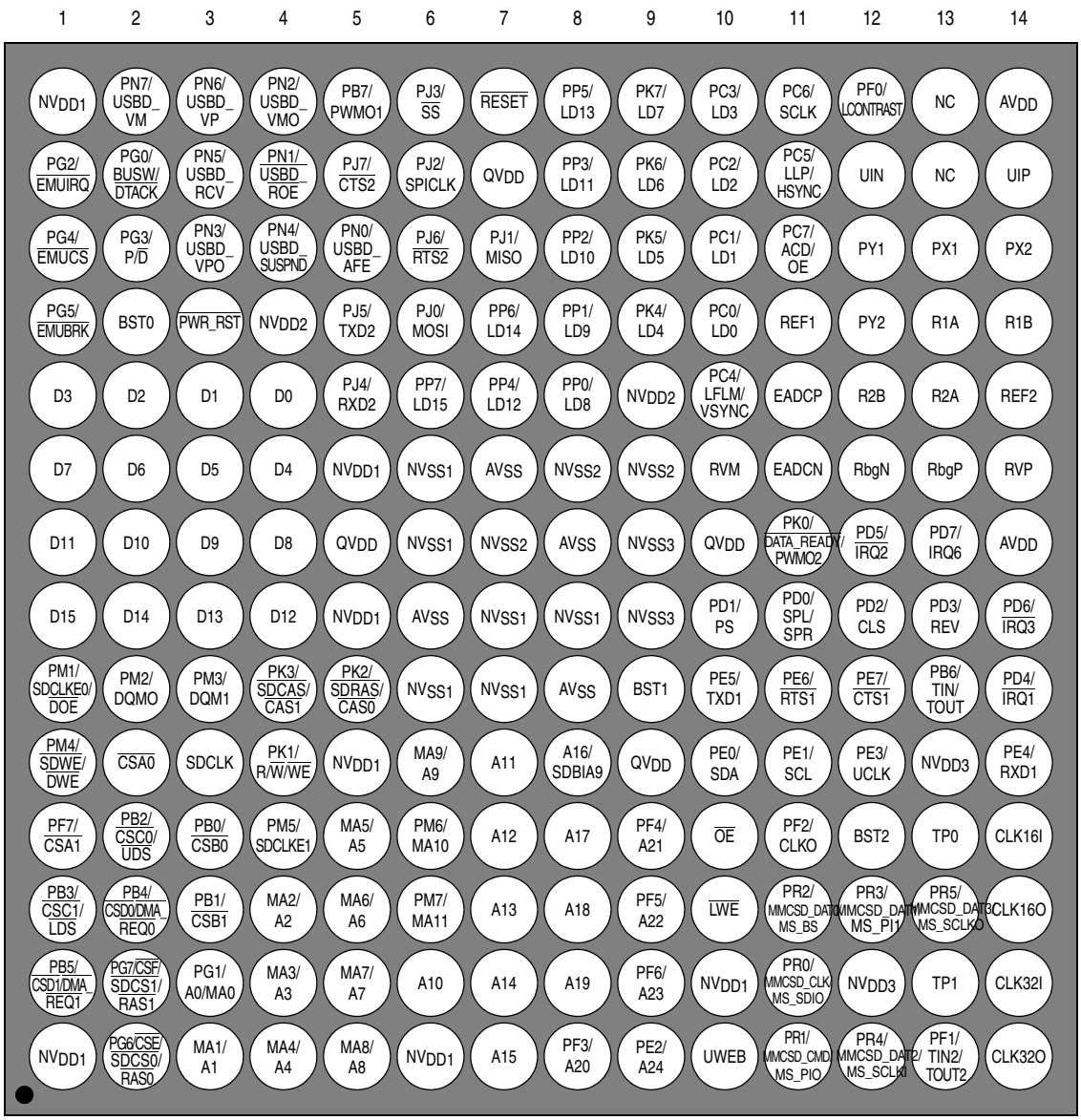
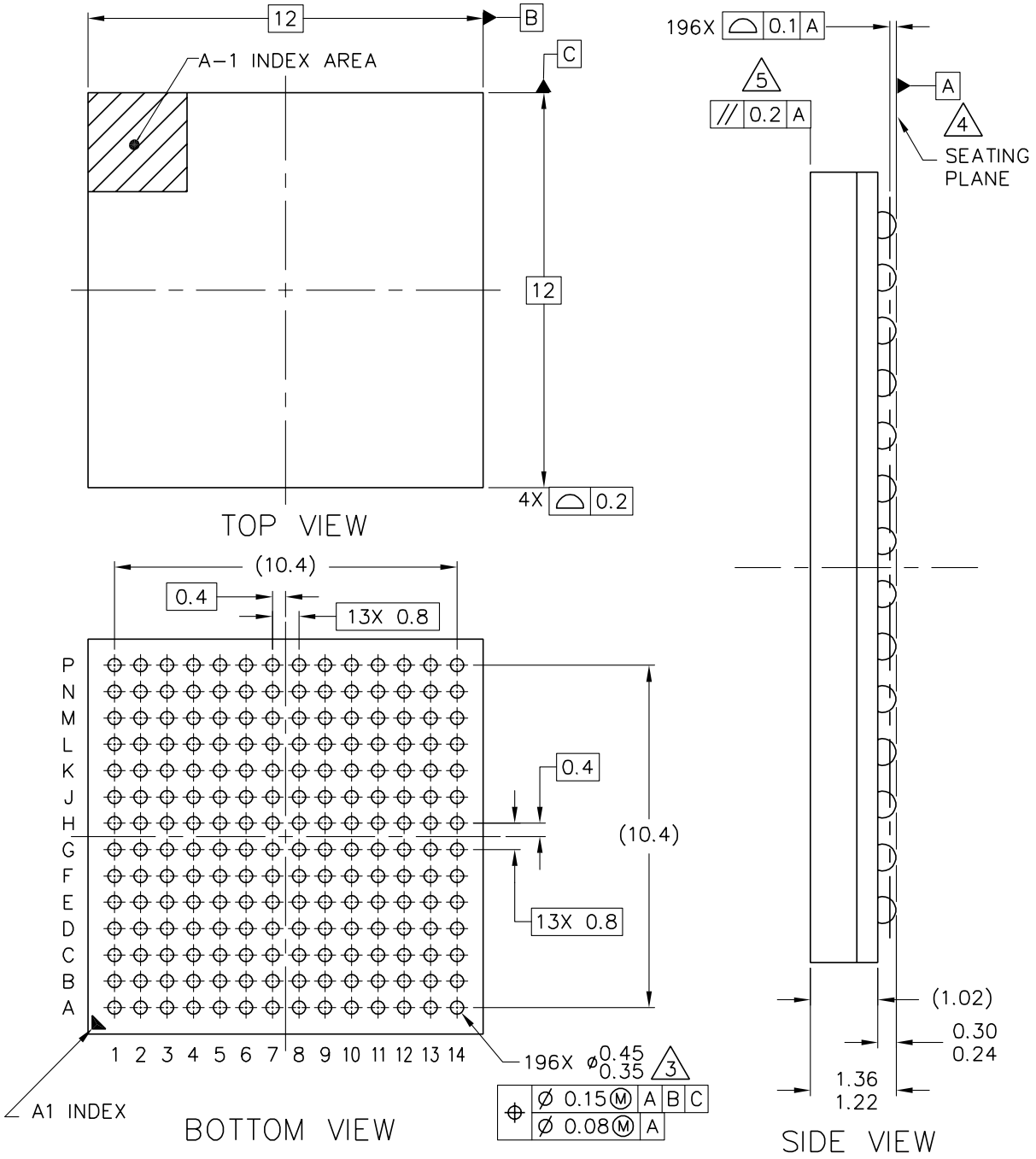


Figure 27-2. MC68SZ328 MAPBGA Pin Assignments—Bottom View

27.3 MAPBGA Package Dimensions

Figure 27-4 illustrates the MAPBGA 12 mm × 12 mm package, which has 0.8 mm spacing between the pads. The device designator for the MAPBGA package is VF.



CASE 1267A-02
REV A

Figure 27-3. MC68SZ328 MAPBGA Mechanical Drawing (Sheet 1 of 2)

NOTES:

1. ALL DIMENSIONS IN MILLIMETERS.
2. DIMENSIONING AND TOLERANCING PER ASME Y14.5M–1994.
3. MAXIMUM SOLDER BALL DIAMETER MEASURED PARALLEL TO DATUM A.
4. DATUM A, THE SEATING PLANE, IS DETERMINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
5. PARALLELISM MEASUREMENT SHALL EXCLUDE ANY EFFECT OF MARK ON TOP SURFACE OF PACKAGE.

Figure 27-4. MC68SZ328 MAPBGA Mechanical Drawing (Sheet 2 of 2)

27.4 PCB Finish Requirement

For a more reliable BGA assembly process, use HASL finish on PCB. EMNI AU finish is not recommended. When EMNI AU finish is used on PCB, brittle intermetallic fractures occasionally occur at the BGA pad–to–PCB pad solder joint.

Index

Numerics

A

B

C

D

E

F

G

H

I

L

N

O

P

R

S

T

U

V

W

X

Y

Z