MC68VZ328 Integrated Processor

User's Manual

MC68VZ328UM/D
Rev. 0, 02/2000

MOTOROLA

MFAX and DragonBall are trademarks of Motorola, Inc.

This document contains information on a new product. Specifications and information herein are subject to change without notice.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part.

Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

All other tradenames, trademarks, and registered trademarks are the property of their respective owners.

How to reach us:

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution; P.O. Box 5405, Denver, Colorado, 80217
1-303-675-2140 or 1-800-441-2447

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, 2 Tai Po, N.T., Hong Kong. 852-26668334

Customer Focus Center: 1-800-521-6274

Mfax™: RMFAX0@email.sps.mot.com
–TOUCHTONE 1-602-244-6609
–US & Canada ONLY 1-800-774-184
–http://sps.motorola.com/mfax/

HOME PAGE: http://motorola.com/sps

Motorola DragonBall Products Home Page: http://www.motorola.com/dragonball

Contents

About This Book

Chapter 1
Introduction

Chapter 2
Signal Descriptions

Chapter 3
Memory Map

Chapter 4
Clock Generation Module and Power Control Module

Chapter 5
System Control

Chapter 6
Chip-Select Logic

Chapter 7
DRAM Controller

Chapter 8
LCD Controller

Chapter 9
Interrupt Controller

Chapter 10
I/O Ports

Chapter 11
Real-Time Clock

Chapter 12
General-Purpose Timers

Chapter 13
Serial Peripheral Interface 1 and 2

Chapter 14
Universal Asynchronous Receiver/Transmitter 1 and 2

Chapter 15
Pulse-Width Modulator 1 and 2

Chapter 16
In-Circuit Emulation

Chapter 17
Bootstrap Mode

Chapter 18
Application Guide

Chapter 19
Electrical Characteristics

Chapter 20
Mechanical Data and Ordering Information

Index

List of Figures

List of Tables

List of Examples

About This Book

This user’s manual describes the features and operation of the MC68VZ328 (DragonBall™ VZ) microprocessor, the third generation of the DragonBall family of products. It provides the details of how to initialize, configure, and program the MC68VZ328. The manual presumes basic knowledge of 68000 architecture.

Audience

The MC68VZ328 user’s manual is intended to provide a design engineer with the necessary data to successfully integrate the MC68VZ328 into a wide variety of applications. It is assumed that the reader has a good working knowledge of the 68000 CPU. For programming information about the 68000, see the documents listed in the Suggested Reading section of this preface.

Organization

The MC68VZ328 user’s manual is organized into 20 chapters that cover the operation and programming of the DragonBall VZ device. Summaries of the chapters follow.

Chapter 1 **Introduction**: This chapter contains a device overview, system block diagrams, and an operational overview of 68000 CPU operation.

Chapter 2 **Signal Descriptions:** This chapter contains listings of the MC68VZ328 input and output signals, organized into functional groups.

Chapter 3 **Memory Map**: This chapter summarizes the memory organization, programming information, and registers’ addresses and reset values.

Chapter 4 **Clock Generation Module and Power Control Module:** This chapter provides detailed information about the operation and programming of the clock generation module as well as the recommended circuit schematics for external clock circuits. It also describes and provides programming information about the operation of the power control module and the system power states.

Chapter 5 **System Control**: This chapter describes the operation of and programming models for the system control, peripheral control, ID, and I/O drive control registers.

Chapter 6 **Chip-Select Logic:** This chapter 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 7 **DRAM Controller**: The operation and programming of the DRAM controller is described in this chapter. This module provides a glueless interface to 8-bit or 16-bit DRAM supporting EDO RAM, Fast Page Mode, and synchronous DRAM.

Chapter 8 **LCD Controller:** This chapter describes the operation and programming of the LCD controller, which provides display data for external LCD drivers or for an LCD panel.

Chapter 9 **Interrupt Controller:** This chapter provides a description and operational considerations for interrupt controller operation. It includes a description of the vector generator and pen and keyboard interrupts.

Chapter 10 **I/O Ports**: This chapter covers all 76 GPIO lines found in the MC68VZ328. Because each pin is individually configurable, a detailed description of the operation of and programming information for each pin is provided.

Chapter 11 **Real-Time Clock:** This chapter 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 12 **General-Purpose Timers**: This chapter describes 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 **Serial Peripheral Interface 1 and 2**: This chapter describes the features of the DragonBall VZ's two serial peripheral interfaces and how they are used to communicate with external devices.

Chapter 14 **Universal Asynchronous Receiver/Transmitter 1 and 2:** The two universal asynchronous receiver/transmitter (UART) ports allow the incorporation of serial communication in existing and new designs. This section describes how data is transported in character blocks using the standard "start-stop" format. It also discusses how to configure and program the UART modules.

Chapter 15 **Pulse-Width Modulator 1 and 2:** This chapter describes both pulse-width modulators. Programming information is also provided.

Chapter 16 **In-Circuit Emulation:** This chapter describes the in-circuit emulation (ICE) module and how it is used to support low-cost emulator designs for the MC68VZ328 microprocessor.

Chapter 17 **Bootstrap Mode**: The operation of bootstrap models is described in detail in this chapter. This chapter 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 1 or UART 2 controller.

Chapter 18 **Application Guide:** This chapter contains information that will assist during the integration of the MC68VZ328 into an existing or a new design. It includes a design checklist and instructions for using the MC68VZ328 Application Development System (ADS) board to get started with the design process.

Chapter 19 **Electrical Characteristics:** This chapter describes the electrical characteristics of the MC68VZ328 integrated processor.

Chapter 20 **Mechanical Data and Ordering Information:** This chapter provides mechanical data, including illustrations, and ordering information.

Suggested Reading

The following documents are required for a complete description of the MC68VZ328 and are necessary to design properly with the part. Especially for those not familiar with the 68000 CPU, 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 MC68EZ328UMA/D)

MC68VZ328 Product Brief (order number MC68VZ328P/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 user's 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

This chapter describes the overall system architecture of the MC68VZ328 (DragonBall™ VZ) integrated processor. It provides an overview of the 68000 CPU and the operational blocks of the MC68VZ328 at a system level.

The MC68VZ328 builds on the success of the earlier DragonBall processors and features a synthesizable 68000 core that utilizes an advanced process technology. Thus, the DragonBall VZ can provide system designers with more performance—the capability of running at higher speed while achieving lower power consumption using a true static core. Additionally, the new DragonBall VZ integrates the logic needed to support color LCD panels on-chip. The DragonBall VZ is the integrated processor of choice for some of the most popular PDA designs, and it can be used in a wide variety of other applications including exercise monitors, games, smart toys, depth finders, navigation systems, and smart phones.

All these features combine to make the MC68VZ328 microprocessor attractive to many system designers. Its functionality and glue logic are all optimally connected and timed with the same clock. Also, only the essential signals are brought out to the pins, allowing the MC68VZ328's primary packages (TQFP and MAPBGA) to occupy the smallest possible footprint on the circuit board.

To improve total system throughput and reduce component count, board size, and the cost of system implementation, the MC68VZ328 combines a powerful FLX68000 processor with intelligent peripheral modules and typical system interface logic. The architecture of the MC68VZ328, shown in Figure 1-1 on page 1-2, consists of the following blocks:

- FLX68000 CPU
- Chip-select logic and 8-/16-bit bus interface
- Clock generation module (CGM) and power control
- Interrupt controller
- 76 GPIO lines grouped into 10 ports
- Two pulse-width modulators (PWM 1 and PWM 2)
- Two general-purpose timers
- Two serial peripheral interfaces (SPI 1 and SPI 2)
- Two UARTs (UART 1 and UART 2) and infrared communication support
- LCD controller
- Real-time clock
- DRAM controller that supports EDO RAM, Fast Page Mode, and SDRAM
- In-circuit emulation module
- Bootstrap mode

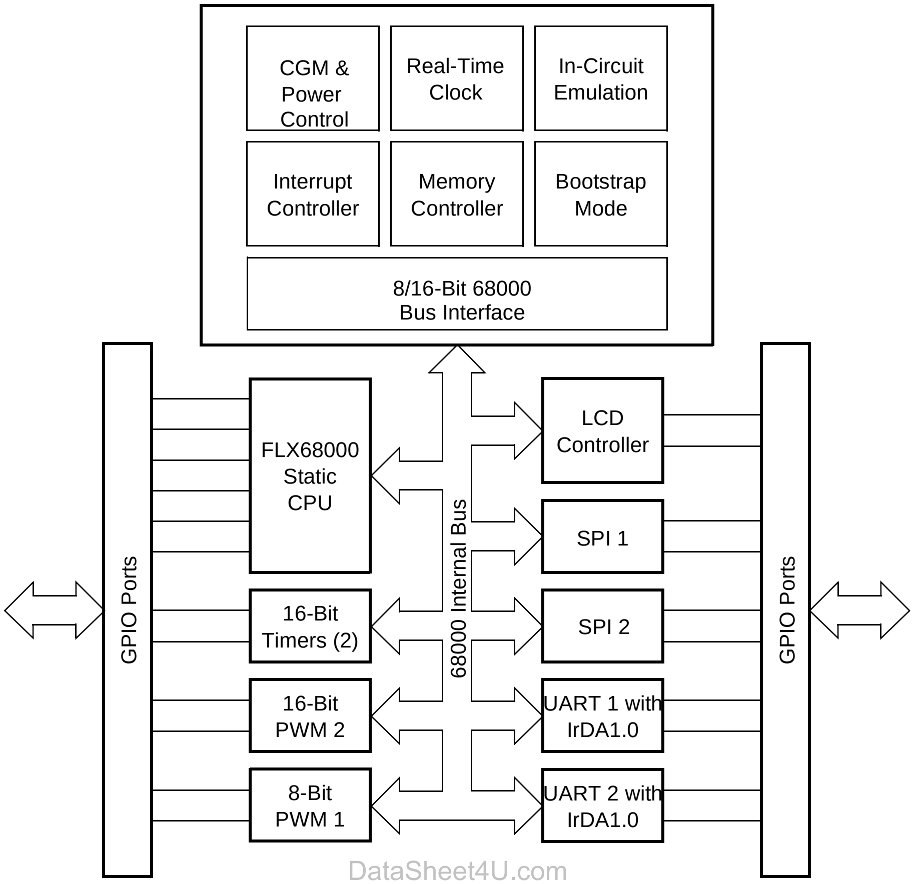


Figure 1-1. MC68VZ328 Block Diagram

1.1 Features of the MC68VZ328

The features of the DragonBall VZ include the following:

- Static FLX68000 CPU—identical to the MC68EC000 microprocessor
  - Full compatibility with MC68000 and MC68EC000
  - 32-bit internal address bus
  - Static design that allows processor clock to be stopped to provide power savings
  - 5.4 MIPS performance at 33 MHz processor clock
  - External M68000 bus interface with selectable bus sizing for 8-bit and 16-bit data ports
- System integration module (SIM) that incorporates many functions typically related to external array logic, reducing parts counts in design, with functions that include the following:
  - System configuration and programmable address mapping
  - Glueless interface to SRAM, DRAM, SDRAM, EPROM, and flash memory
  - Eight programmable chip-selects with wait-state generation logic
  - Four programmable interrupt I/Os, with keyboard interrupt capability

  - Five general-purpose, programmable edge/level/polarity interrupt IRQs
  - Other programmable I/O, multiplexed with peripheral functions of up to 76 GPIO lines
  - Programmable interrupt vector response for on-chip peripheral modules
  - Low-power mode control
- DRAM controller
  - Support for $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh cycles and self-refresh mode DRAM
  - Support for 8-bit and 16-bit port DRAM and synchronous DRAM
  - EDO RAM or automatic Fast Page Mode for LCD access
  - Programmable refresh rate
  - Support for up to two banks of DRAM and EDO RAM
  - Programmable column address size
- 76 GPIO lines grouped into 10 ports
- Two UART ports
- Two serial peripheral interface (SPI) ports
- Two 16-bit general-purpose counters/timers
  - Automatic interrupt generation
  - 30 ns resolution at 33 MHz system clock
  - Timer input/output pin
- Real-time clock/sampling timer
  - Separate power supply for the RTC
  - One programmable alarm
  - Capability of counting up to 512 days
  - Sampling timer with selectable frequency (4 Hz, 8 Hz, 16 Hz, 32 Hz, 64 Hz, 256 Hz, 512 Hz, 1 kHz)
  - Interrupt generation for digitizer sampling or keyboard debouncing
- LCD controller
  - Software-programmable screen size (up to 640 × 512) to support single (nonsplit) monochrome and color STN panels
  - Capability of directly driving popular LCD drivers and modules from Motorola, Sharp, Hitachi, Toshiba, and numerous other manufacturers
  - Support for up to 16 gray levels out of a palette of 16 density levels
  - Utilization of system memory as display memory
  - LCD contrast control using 8-bit PWM
- Two pulse-width modulator (PWM) modules
  - Audio effects support
  - 16- and 8-bit resolution
  - 5-byte FIFO that provides more flexibility on performance
  - Sound and melody generation

- Built-in emulation function
  - — Dedicated memory space for emulator debug monitor with chip-select
  - — Dedicated interrupt (interrupt level 7) for in-circuit emulation (ICE)
  - — One address-signal comparator and one control-signal comparator, with masking to support single or multiple hardware execution
  - — Breakpoint
  - — One breakpoint instruction insertion unit
- Bootstrap mode function
  - — Capability to initialize system and download programs and data to system memory through UART
  - — Acceptance of execution command to run program stored in system memory
  - — 8-byte-long instruction buffer for 68000 instruction storage and execution
- Power management
  - — Fully static HCMOS technology
  - — Programmable clock synthesizer using 32.768 kHz or 38.4 kHz external crystal for full frequency control
  - — Low-power stop capabilities
  - — Modules that can be individually shut down
  - — Operation from DC to 33 MHz (processor clock)
  - — Operating voltage of 2.7 V to 3.3 V
  - — Compact 144-lead thin quad flat pack (TQFP) and MAPBGA

1.2 CPU

The FLX68000 CPU in the MC68VZ328 is an updated implementation of the 68000 32-bit microprocessor architecture. The main features of the CPU are the following:

- 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 CPU is completely code compatible with other members of the M68000 families, which means it has access to a broad base of established real-time kernels, operating systems, languages, applications, and development tools.

1.2.1 CPU Programming Model

The CPU has 32-bit registers and a 32-bit program counter, which are shown in Figure 1-2. 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 being 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.

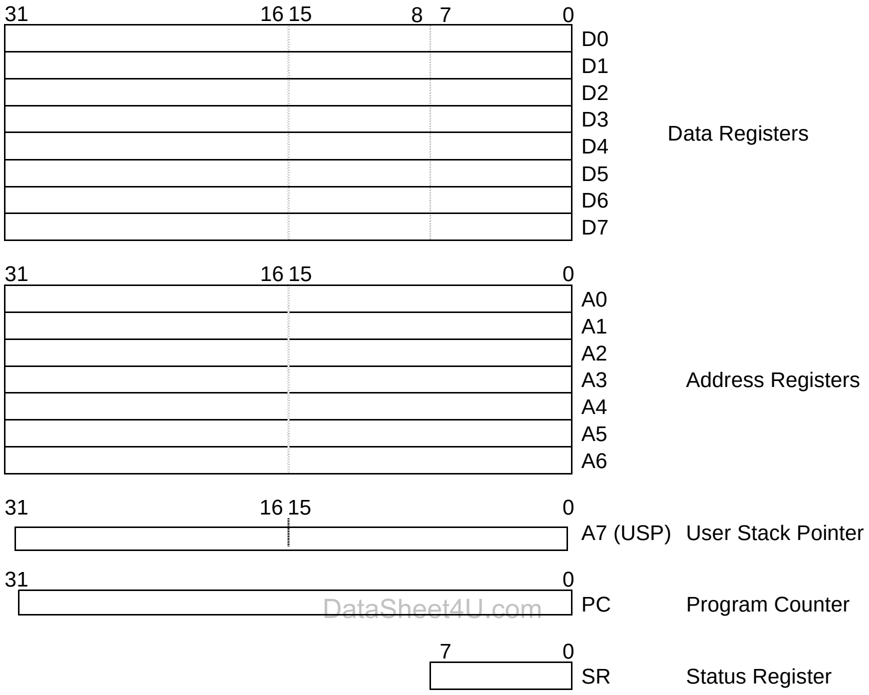


Figure 1-2. User Programming Model

In supervisor mode, the upper byte of the status register and the supervisor stack pointer (SSP) can also be programmed, as shown in Figure 1-3.

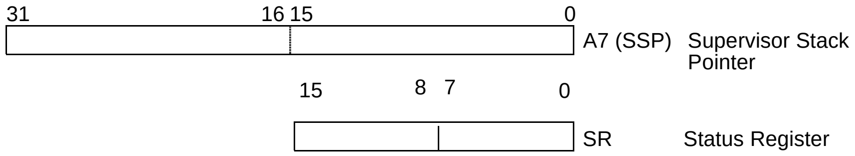


Figure 1-3. Supervisor Programming Model Supplement

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

1.2.2 Data and Address Mode Types

The CPU 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 1-1.

Table 1-1. Address Modes

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

Note:
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	

1.2.3 FLX68000 Instruction Set

The FLX68000 CPU 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 1-2 on page 1-7, include signed and unsigned multiply and divide, quick arithmetic operations, binary-coded decimal (BCD) arithmetic, and expanded operations (through traps).

Table 1-2. Instruction Set

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

Table 1-2. Instruction Set (Continued)

Mnemonic	Description	Mnemonic	Description
EORI	Exclusive OR immediate	STOP	Stop
EORI to CCR	Exclusive OR immediate to condition codes	SUB	Subtract
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		

1.3 Modules of the MC68VZ328

In addition to the powerful 68000 processor, the DragonBall VZ contains a wide variety of peripheral interface and control modules. The following subsections provide brief descriptions of these modules and how they operate.

1.3.1 Memory Controller

The memory controller provides a glueless interface to most memory chips on the market. It supports flash, ROM, SRAM, different DRAM types (EDO RAM and Fast Page Mode), and synchronous DRAM. Either one or two banks of DRAM may be used, and each bank can be a maximum of 32 Mbyte. For a more complete explanation of how memory is configured and controlled, see Chapter 3, “Memory Map.”

1.3.2 Clock Generation Module and Power Control Module

The module containing the clock synthesizer operates with either an external crystal or an external oscillator to provide a stable clock source for the internal clock generation module (CGM). The output frequency can be adjusted by writing to the CGM frequency select register. The CGM can be disabled to shut down the system clock divider chain for maximum power saving, while the real-time clock (RTC) and DRAM controller remain active. The power control module can be configured to control the CPU cycles to optimize power consumption. The power control module offers three power-saving modes: normal, doze, and sleep. When in sleep mode, the CGM wakes up automatically when any unmasked external or internal interrupt occurs. See Chapter 4, “Clock Generation Module and Power Control Module,” for more detailed information.

1.3.3 System Control

The primary function of the system control module is to provide configuration control of several other modules in the DragonBall VZ. These registers grant permission for access to many of the internal peripheral registers. In addition, the module controls address space of the internal peripheral registers and the bus time-out control and status (bus error generator). System control also is used to program the drive current of the GPIO lines. See Chapter 5, “System Control,” for more information.

1.3.4 Chip-Select Logic

The MC68VZ328 provides eight programmable general-purpose chip-select signals to allow the selection of a wide variety of memory or external peripherals. Each chip-select signal provides a write-protect option, internal and external DTACK generation, and 8-bit and 16-bit data port size selection. For more detailed information about using the chip-select logic, see Chapter 6, “Chip-Select Logic.”

1.3.5 DRAM Controller

The DRAM controller provides a glueless interface for either 8-bit or 16-bit DRAM. It supports EDO RAM, Fast Page Mode, and synchronous DRAM. The DRAM controller provides row address strobe ($\overline{\text{RAS}}$) and column address strobe ($\overline{\text{CAS}}$) signals for up to a maximum of two banks of DRAM. In addition to controlling DRAM, the DRAM controller supports access for LCD controller burst accesses. See Chapter 7, “DRAM Controller,” for more information about this module.

1.3.6 LCD Controller

The LCD controller provides display data for external LCD drivers or for an LCD panel. The LCD controller fetches display data directly from system memory through periodic DMA transfer cycles. For this reason, an understanding of the DRAM controller is recommended. For more information, please refer to Chapter 7, “DRAM Controller,” as well as Chapter 8, “LCD Controller.”

1.3.7 Interrupt Controller

The interrupt controller prioritizes internal and external interrupt requests and generates a vector number during the CPU interrupt-acknowledge cycle. Interrupt nesting is also provided so that an interrupt service routine of a lower-priority interrupt may be suspended by a higher-priority interrupt request. The on-chip interrupt controller features prioritized interrupts, a fully nested interrupt environment, programmable vector generation, and unique vector number generation for each interrupt level. For additional information about this module, see Chapter 9, “Interrupt Controller.”

1.3.8 General-Purpose I/O (GPIO) Lines

The MC68VZ328 supports a maximum of 76 GPIO lines grouped together in ports A–G, J, K, and M. These ports can be configured as GPIO pins or dedicated peripheral interface pins. Each pin can be independently programmed as a GPIO pin even when other pins related to that on-chip peripheral are used as dedicated pins. For detailed information about programming these GPIO lines, see Chapter 10, "I/O Ports."

1.3.9 Real-Time Clock

A real-time clock provides the time of day with 1-second resolution. Using an external crystal (either 32.768 kHz or 38.4 kHz) as a clock source, it keeps time as long as power is applied to the chip, even when it is in sleep or doze mode. The watchdog clock timer protects against system failures by providing a way of escape from unexpected input conditions, external events, or programming errors. Once started, this timer must be cleared by software on a regular basis so that it never reaches its time-out value. When it does reach its time-out value, the watchdog timer assumes that a system failure has occurred and the software watchdog logic resets or interrupts the CPU. For detailed information about configuring and programming this module, refer to Chapter 11, "Real-Time Clock."

1.3.10 General-Purpose Timer

The MC68VZ328 has two 16-bit timers that can be used in various modes to capture the timer value with an external event, to trigger an external event or interrupt when the timer reaches a set value, or to count external events. Each timer has an 8-bit prescaler to allow a programmable clock input frequency to be derived from the system clock. The two timers can also be cascaded together as one 32-bit timer. This module is described in detail in Chapter 12, "General-Purpose Timers."

1.3.11 Serial Peripheral Interfaces (SPI)

The MC68VZ328 contains two serial peripheral interface (SPI) modules, SPI 1 and SPI 2. The serial peripheral interfaces are mainly used for controlling external peripherals. The passed data is synchronized with the SPI clock, and it is transmitted and received with the same SPI clock. One SPI module (SPI 2) only operates in master mode, which initiates SPI transfers from the MC68VZ328 to the peripheral. The other SPI (SPI 1) may be configured as either master or slave. Chapter 13, "Serial Peripheral Interface 1 and 2," provides detailed information about the configuration and operation of the SPIs.

1.3.12 Universal Asynchronous Receiver/Transmitter (UART) Modules

The two UART ports in the MC68VZ328 may be used to communicate with external serial devices. UART 1 is identical to the UART in the DragonBall EZ processor, while UART 2 represents an enhanced version of UART 1. One of the enhancements to the UART 2 design consists of an enlarged RxFIFO and TxFIFO to reduce the number of software interrupts. An improvement to both UARTs is the system clock input frequency, which is 33.16 MHz, doubling the 16.58 MHz frequency of the MC68EZ328. For a 33.16 MHz system clock, software written for the MC68EZ328 version of the chip is not compatible unless the divider and prescaler are adjusted to compensate for the increased clock speed. For more information about the programming and configuration of these two modules, see Chapter 14, "Universal Asynchronous Receiver/Transmitter 1 and 2."

1.3.13 Pulse-Width Modulators (PWM)

The MC68VZ328 has two pulse-width modulators (PWMs). Each of the pulse-width modulators has three modes of operation—playback, tone, and digital-to-analog (D/A) conversion. Using these three modes, the PWM can be used to play back high-quality digital sounds, produce simple tones, or convert digital data into analog waveforms. The 8-bit PWM contains a 5-byte FIFO that enhances the system performance by reducing the number of interrupts to the CPU. The 16-bit PWM provides higher resolution for better sound quality. Users can enable both PWMs at the same time to generate a mixed PWMO signal. See Chapter 15, “Pulse-Width Modulator 1 and 2,” for more detailed information about the configuration and operation of these devices.

1.3.14 In-Circuit Emulation Module

The in-circuit emulation module is designed for low-cost emulator development purposes. System memory space, which is 0xFFFC0000 to 0xFFFCFFFF, is covered by the $\overline{\text{EMUCS}}$ signal and primarily dedicated to the emulator debug monitor. However, the $\overline{\text{EMUCS}}$ signal can be used to select the monitor ROM or system I/O port. Keep in mind that if the monitor ROM is selected, the system must boot up in emulator mode. Refer to Chapter 16, “In-Circuit Emulation,” for more details.

1.3.15 Bootstrap Mode

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 UART 2 controller. See Chapter 14, “Universal Asynchronous Receiver/Transmitter 1 and 2,” for information on operating and programming the UART controllers. Once a program is downloaded to the MC68VZ328, it can be executed, providing a simple debugging environment for failure analysis and a channel to update programs stored in flash memory. Simple hardware debug functions may be performed on the target system using the bootstrap utility program BBUGV.EXE, which is available on the following World Wide Web site: http://www.Motorola.com/DragonBall. See Chapter 17, “Bootstrap Mode,” for more information about this mode.

Chapter 2
Signal Descriptions

This chapter describes the MC68VZ328's input and output signals, which are organized into functional groups, as illustrated in Figure 2-1 on page 2-2. The MC68VZ328 uses a standard M68000 bus to communicate with on-chip and external peripherals. This single continuous bus exists both on and off the chip. CPU read cycles to internal memory-mapped registers of the device are invisible on the external bus, but write cycles to internal or external memory-mapped locations are visible.

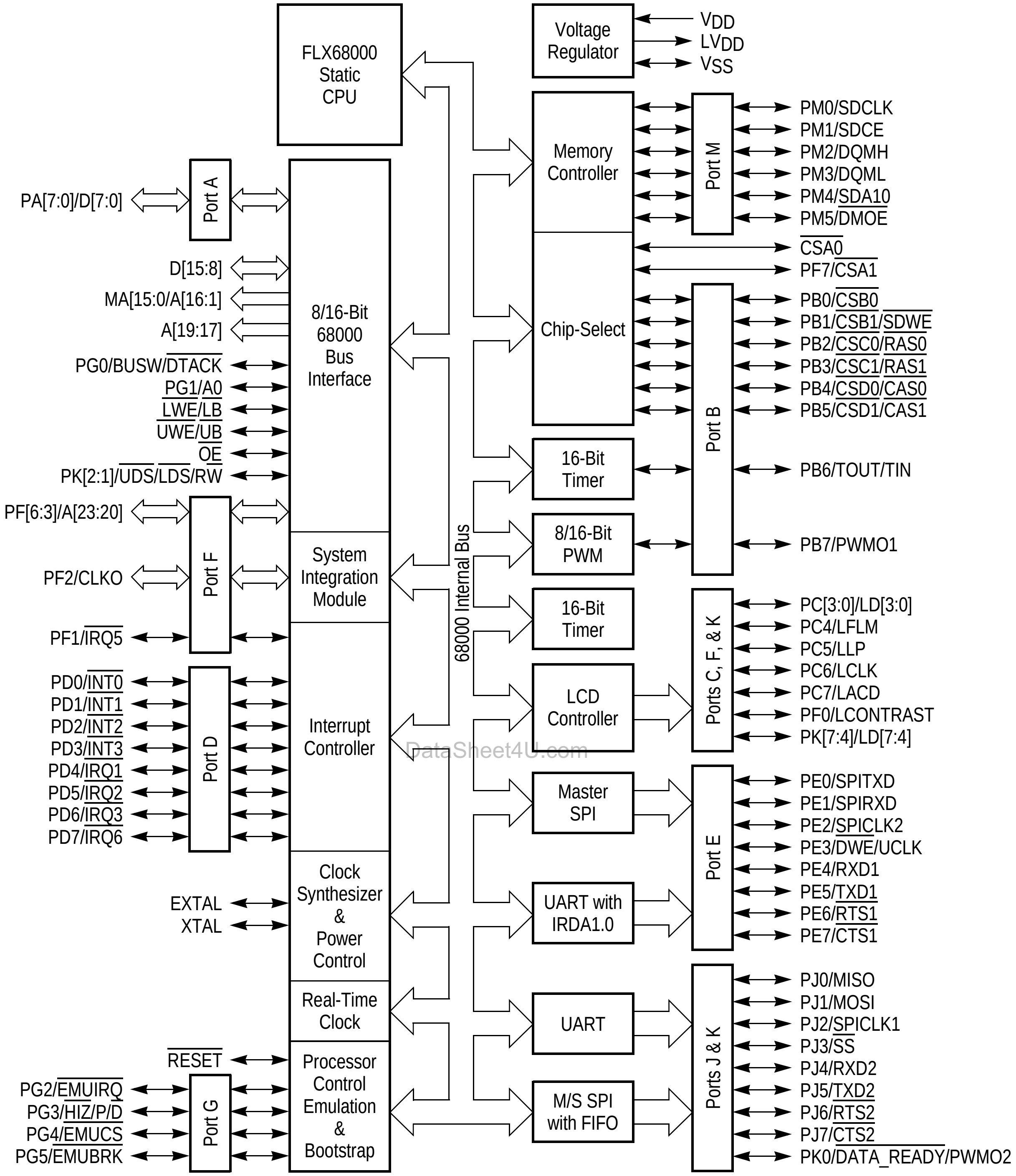


Figure 2-1. Signals Grouped by Function

2.1 Signals Grouped by Function

Table 2-1 on page 2-3 groups the MC68VZ328 signals according to their function.

Table 2-1. Signal Function Groups

Function Group	Signals	Number of Pins	
		TQFP	PBGA
Power	V_{DD}	9	5
Ground	V_{SS}	16	28
Regulator output	LV_{DD}	5	1
Clocks/PCIO	XTAL, EXTAL, CLKO/PF2	3	3
System control	RESET	1	1
Address bus/PFIO	PF[3:6]/A[23:20], A[19:14], A0/PG1, MA[15:0]/A[16:1]	24	24
Lower data bus/PAIO	PA[7:0]/D[7:0]	8	8
Upper data bus	D[15:8]	8	8
Bus control/PCIO/PEIO/ PKIO	BUSW/DTACK/PG0, OE, LWE/LB, UWE/UB, PE3/DWE/UCLK, PK2/LDS, PK3/UDS, PK1/RW	8	8
Interrupt controller/PMIO	INT0/PD0, INT1/PD1, INT2/PD2, INT3/PD3, IRQ1/PD4, IRQ2/PD5, IRQ3/PD6, IRQ6/PD7, IRQ5/PF1	9	9
LCD controller/PCIO	LACD/PC7, LCLK/PC6, LLP/PC5, LFLM/PC4, LD[7:4]/PK[7:4], LD[3:0]/PC[3:0], LCONTRAST/PF0	13	13
UART1/PEIO, UART2/PJIO	PE4/RXD1, PE5/TXD1, PE6/RTS1, PE7/CTS1, PJ4/RXD2, PJ5/TXD2, PJ6/RTS2, PJ7/CTS2	8	8
Timer/PBIO	TOUT/TIN/PB6	1	1
Pulse-width modulator/PBIO	PWMO1/PB7 (PM5/DATA_READY/PWMO2)	1	1
Master SPI/PEIO, configurable SPI/PJIO/PKIO	SPITXD/PE0, SPIRXD/PE1, SPICLK2/PE2, PJ0/MOSI, PJ1/MISO, PJ2/SPICLK1, PJ3/SS, PK0/DATA_READY/PWMO2	8	8
Chip-select, EDO RAM/PBIO, PMIO	CSA[1:0]/PF7, CSB[1:0]/PB[1:0]/SDWE, CSC[1:0]/PB[3:2]/RAS[1:0], CSD[1:0]/PB[5:4]/CAS[1:0], PM5/DMOE	9	9
SDRAM/PMIO	PM0/SDCLK, PM1/SDCE, PM2/DQMH, PM3/DQML, PM4/SDA10, (SDWE, SDCAS[1:0], SDRAS[1:0])—multiplexed with chip-select signals	5	5
Emulator pins	EMUIRQ/PG2, EMUBRK/PG5, HIZ/P/D/PG3, EMUCS/PG4	4	4
No connect pins	NC	4	0

2.2 Power and Ground Signals

The MC68VZ328 microprocessor has three types of power pins. They are V_{DD}, V_{SS}, and LV_{DD}.

- V_{DD}—External power supply to drive all I/O pins and for the internal voltage regulator. It is recommended to place a 0.1 µF bypass capacitor close to each of these pins.
- V_{SS}—Signal return pin for both digital and analog circuits.
- LV_{DD}—Internal voltage regulator output signal that is used by the internal circuitry. The LV_{DD} pins should not be used as an external circuit power supply due to current supply limitations. Each package has unique bypass capacitor requirements. The TQFP package requires that an external bypass capacitor circuit of 0.01 µF and 0.0001 µF (in parallel) be placed close to each of the LV_{DD} pins, except pin 35, which requires a 270 nF and a 0.0001 µF bypass capacitor. The PBGA has a single LV_{DD} pin (M1) requiring only a 270 nF and a 0.0001 µF bypass capacitor.

NOTE:

For maximum noise immunity, ensure that external bypass capacitors are placed as close to the pins as possible.

2.3 Clock and System Control Signals

There are four clock and system control signals.

- EXTAL—External Clock/Crystal. This input signal connects to the external low frequency crystal. The MC68VZ328 microprocessor supports both a 32.768 kHz and a 38.4 kHz crystal frequency. For a 32.768 kHz input, the internal phase-locked loop generates a PLLCLK signal that passes through two prescalers, and the resulting output (DMACLK and SYSCLK) clock is 16.58 MHz. Figure 2-2 illustrates how a crystal is usually connected to the MC68VZ328. For specific circuit design values, see Figure 4-2 on page 4-4.

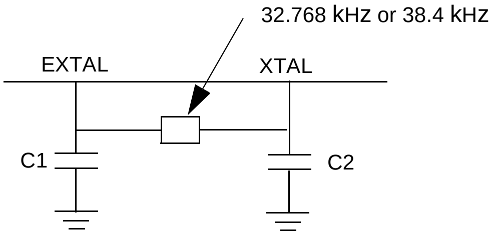


Figure 2-2. Typical Crystal Connection

- XTAL—Crystal. This output signal connects the on-chip oscillator output to an external crystal.
- CLKO/PF2—Clock Out or bit 2 of Port F. This output clock signal is derived from the on-chip clock oscillator and is internally connected to the clock output of the internal CGM. This signal is provided for external reference. The output can be disabled in the PLL control register to reduce power consumption and electromagnetic emission. See Section 4.4.1, “PLL Control Register,” on page 4-8 for more information. The CLKO/PF2 signal defaults to the Port F pin 2 input signal. For detailed information, refer to Section 10.4.7.3, “Port F Dedicated I/O Functions,” on page 10-26.
- $\overline{\text{RESET}}$—Reset. This active low, Schmitt trigger input signal resets the entire MC68VZ328 processor (CPU and peripherals). The threshold of this Schmitt trigger device is 1.2 V high and 0.8 V low. After the MC68VZ328 powers up, this reset input signal should be driven low for at least

1.2 s before its voltage is higher than 1.2 V to ensure that the crystal oscillator starts and stabilizes. See Section 4.3.1, “CLK32 Clock Signal,” on page 4-4 for details about selecting circuit values. This signal is inactive while the CPU is executing the RESET instruction.

NOTE:

When an R/C circuit is being used to generate the $\overline{\text{RESET}}$ signal to the MC68VZ328, the R/C circuit must be placed as close to the chip as possible.

2.4 Address Bus Signals

The address bus pins A[23:0] are the address lines driven by the CPU or LCD controller for panel refresh DMA. In sleep mode, all address signals are in an active state of the last bus cycle. Refer to Section 4.5.1.4, “Sleep Mode,” on page 4-12 for more detailed information.

- A0/PG1—Address 0 or Port G bit 1. After system reset, this signal defaults to A0.
- MA[15:0]/A[16:1]—Multiplexed DRAM bits 15–0 or Address bits 16–1. These address output lines are multiplexed with the DRAM row and column address signals. The MA signal is selected on DRAM access cycles.
- A[19:17]—Address lines 19–17.
- A[23:20]/PF[6:3]—Address bits 23–20 or Port F bits 6–3. These address lines are multiplexed with Port F. These signals default to address functions after reset.

2.5 Data Bus Signals

The flexible data bus interface design of the MC68VZ328 microprocessor allows programming of the lower byte of the data bus (in an 8-bit-only system) to operate as general-purpose I/O signals. In sleep mode, all of the data bus pins (D15–D0) are individually pulled up with 1-megaohm resistors. Refer to Section 4.5.1.4, “Sleep Mode,” on page 4-12 for more detailed information.

- D[15:8]—Data bits 15–8. The upper byte of the data bus is not multiplexed with any other signal. In pure 8-bit systems, this is the data bus. In mixed 8- and 16-bit systems, 8-bit memory blocks or peripherals should be connected to this bus.
- D[7:0]/PA[7:0]—Data bits 7–0 or Port A bits 7–0. This bus is the lower data byte or general-purpose I/O. In pure 8-bit systems, this bus can serve as a general-purpose I/O. The WDTH8 bit in the SCR register (0xFFF000) should be set to 1 by software before the port can be used. See Section 5.2.1, “System Control Register,” on page 5-2 for details on setting this bit. In 16-bit or mixed 8- and 16-bit systems, these pins must function as the lower data byte.

2.6 Bus Control Signals

The bus control signals are used for both the configuration and operation of the MC68VZ328 bus. The following descriptions provide detailed information about programming the signals and their use.

- $\overline{\text{LWE}}$/$\overline{\text{LB}}$, $\overline{\text{UWE}}$/$\overline{\text{UB}}$**—**Lower Byte Write-Enable and Upper Byte Write-Enable, or Lower Byte and Upper Byte data strobes. For all chip-select cycles except $\overline{\text{CSB}}$[1:0], these two pins are $\overline{\text{LWE}}$ and $\overline{\text{UWE}}$. They are used as lower and upper write-enable signals to a 16-bit port. If the chip-select is set to 8-bit port (the BSW bit is clear), use only the $\overline{\text{UWE}}$ signal for write-enable control. $\overline{\text{UWE}}$ can be used as a DRAM write-enable if DRAM refresh does not require that $\overline{\text{UWE}}$ stay high. Otherwise, $\overline{\text{DWE}}$ should be used. For $\overline{\text{CSB}}$[1:0] cycles, if the SR16 bit is clear in the CSCTRL1 register, these two pins are $\overline{\text{LWE}}$ and $\overline{\text{UWE}}$ and function as previously described. If the SR16 bit is set, these two pins are $\overline{\text{UB}}$ and $\overline{\text{LB}}$. These two data strobe signals are normally used to connect to $\overline{\text{UDS}}$ and $\overline{\text{LDS}}$ of the 16-bit memory chip.
- $\overline{\text{DWE}}$/UCLK/PE3**—**DRAM Write-Enable, UART Clock, or Port E bit 3. Use the $\overline{\text{DWE}}$ signal with DRAM, which requires an independent write-enable signal rather than one that is shared with $\overline{\text{UWE}}$. This signal stays high during refresh cycles. This pin defaults to a PE3 input signal. To select the $\overline{\text{DWE}}$ function, program Port E to $\overline{\text{DWE}}$ and enable the $\overline{\text{DWE}}$ signal by writing a 1 to the DWE bit of the DRAMC register, which is described in Section 7.3.2, "DRAM Control Register," on page 7-14. If this bit is not enabled, the UCLK signal function is selected, which is an input clock to the UART module. For a description of the UCLK signal, refer to Section 14.2.3, "Serial Interface Signals," on page 14-3. This pin defaults to GPIO input pulled high.
- BUSW/$\overline{\text{DTACK}}$/PG0**—**Bus Width, Data Transfer Acknowledge, or Port G bit 0. 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 MC68VZ328 microprocessor will latch the BUSW signal at the rising edge of the $\overline{\text{RESET}}$ signal. Its mode will determine the default bus width for $\overline{\text{CSA0}}$. For example, a logic low of BUSW on reset means that $\overline{\text{CSA0}}$ connects to an 8-bit memory device, and a logic high of BUSW on reset means that $\overline{\text{CSA0}}$ connects to a 16-bit memory device. After reset, this pin defaults to the $\overline{\text{DTACK}}$ input signal. $\overline{\text{DTACK}}$ can be configured as output by programming the Port G DIR register. If it is input, only those chip-select cycles using external $\overline{\text{DTACK}}$ will be affected. Chip-select cycles of internal $\overline{\text{DTACK}}$ will ignore the input status. This pin can be configured to GPIO after system reset. For a 16-bit $\overline{\text{CSA0}}$-selected memory device, it is recommended that this signal be pulled up, externally.
- $\overline{\text{OE}}$**—**Output Enable. This active low signal is asserted during a read cycle of the MC68VZ328 microprocessor, which enables the output of either ROM or SRAM.
- $\overline{\text{UDS}}$/PK3, $\overline{\text{LDS}}$/PK2—Data strobes or GPIO. $\overline{\text{UDS}}$ and $\overline{\text{LDS}}$ are 68000 CPU data strobe signals. These pins default to GPIO input pulled high.
- R$\overline{\text{W}}$/PK1—Read/Write or Port K bit 1. R$\overline{\text{W}}$ is the 68000 CPU read/write signal. This pin defaults to GPIO input pulled high.

2.7 Interrupt Controller Signals

This section describes signals that are used by the MC68VZ328 interrupt controller.

- $\overline{\text{INT}}$[3:0], $\overline{\text{IRQ}}$[3:1], $\overline{\text{IRQ6}}$/PD[7:0]**—**Interrupt bits 3–0, Interrupt Request bits 3-1, or Port D bits 7–0. $\overline{\text{INT}}$[3:0], $\overline{\text{IRQ}}$[3:1], and $\overline{\text{IRQ6}}$ can be configured as edge or level trigger interrupt signals. To support keyboard applications, the I/O function can be used with interrupt capabilities, which are described in Chapter 9, "Interrupt Controller." These pins default to GPIO input pulled high.

- $\overline{\text{IRQ5}}$/PF1—Interrupt Request 5 or Port F bit 1. This signal can be programmed as GPIO or as an interrupt input. When configured as an interrupt input, the signal may be programmed as a level high or level low trigger interrupt. This pin defaults to GPIO input pulled high.
- EMIQ—Emulator Interrupt Status. This bit indicates that the in-circuit emulation module or $\overline{\text{EMUIRQ}}$ pin is requesting a level 7 interrupt. 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, read the ICEMSR register to identify the interrupt source and write a 1 to the corresponding bit in the ICEMSR. See Section 9.6.4, “Interrupt Status Register,” on page 9-12 for more information.

2.8 LCD Controller Signals

The MC68VZ328 contains all necessary circuitry to support an external LCD display panel. This section describes the signals used by the LCD controller. It also provides some programming information about the use of these signals.

- LD[3:0]/PC[3:0], LD[7:4]/PK[7:4]—LCD Data Bus bits 7–0, or Port C bits 3–0 and Port K bits 7–4. LD signals output bus transfers of pixel data to the LCD panel to which it will be displayed. The pixel data is arranged to accommodate the programmable panel mode data width selection. Panel interfaces of 1, 2, 4, or 8 bits are supported.

NOTE:

The MC68VZ328’s LCD interface data bus uses the LSB (LD0) to display pixel 0,0. Some LCD panel manufacturers program their LCD panel data bus so that the MSB of the panel displays pixel 0,0. For these panels, the connection between the MC68VZ328’s LCD data bus and the LCD panel’s data bus may have a reversed bit significance. For a 4-bit LCD panel of this type, connect the MC68VZ328’s LD0 signal to the LCD panel’s data bit 3, and then connect LD1 to LCD data 2, LD2 to LCD data 1, and LD3 to LCD data 0. The four pins can also be programmed as I/O ports from Port C. These signals default as GPIO input with Port C being pulled low and Port K pulled high.

- LFLM/PC4—First Line Marker or Port C bit 4. This signal indicates the start of a new display frame. LFLM becomes active after the first line pulse of the frame and remains active until the next line pulse, at which point it deasserts and remains inactive until the next frame. LFLM can be programmed to be an active high or an active low signal. It can also be programmed as an I/O port. This pin defaults to GPIO input pulled low.
- LLP/PC5—LCD Line Pulse or Port C bit 5. The LLP signal is used to latch a line of shifted data onto an LCD panel. The LLP can be programmed to be an active high or active low signal in software. See Section 8.3.10, “LCD Polarity Configuration Register,” on page 8-16 for more information.
- LCLK/PC6—LCD Shift Clock or Port C bit 6. This is the clock output to which the output data to the LCD panel is synchronized. LCLK can be programmed to be either an active high or an active low signal. This pin can also be programmed as an I/O port. This pin defaults to GPIO input pulled low.
- LACD/PC7—LCD Alternate Crystal Direction or Port C bit 7. This output is toggled to alternate the crystal polarization on the panel. This signal can be programmed to toggle at a period of 1 to 128 frames or lines. This pin also can also be programmed as an I/O port. This pin defaults to GPIO input pulled low.

- LCONTRAST/PF0—LCD Contrast and Port F bit 0. This output is generated by the pulse-width modulator (PWM) inside the LCD controller to adjust the supply voltage to the LCD panel. This pin can also be programmed as an I/O port. This pin defaults to GPIO input pulled high.

2.9 UART 1 and UART 2 Controller Signals

There are two Universal Asynchronous Receive Transmit (UART) modules in the MC68VZ328. This section describes the signals that are used to interface with external serial devices.

- RXD1/PE4, RXD2/PJ4—UART 1 and UART 2 Receive Data or Port E bit 4 and Port J bit 4. RXD is the receiver serial input. During normal operation, NRZ data is expected, but in IrDA mode, a narrow pulse of 1.6 µs minimum is expected for each zero bit received. External circuitry must be used to convert the IrDA signal to an electrical signal. RS-232 applications need an external RS-232 receiver to convert voltage levels. These pins default to GPIO input pulled high.
- TXD1/PE5, TXD2/PJ5—UART 1 and UART 2 Transmit Data or Port E bit 5 and Port J bit 5. TXD is the transmitter serial output. During normal operation, they output NRZ data signals. In IrDA mode, they output a selectable pulse width of three-sixteenths bit period or 1.6 µs minimum bit period for each zero bit transmitted. For RS-232 applications, this pin must be connected to an RS-232 transmitter. For IrDA applications, this pin can directly drive an IrDA LED. These pins default to GPIO input pulled high.
- $\overline{\text{RTS1}}$/PE6, $\overline{\text{RTS2}}$/PJ6—UART 1 and UART 2 Request to Send or Port E bit 6 and Port J bit 6. $\overline{\text{RTS}}$ indicates that it is ready to receive data by asserting this pin (low). This pin would be connected to the far-end transmitter's $\overline{\text{CTS}}$ pin. When the receiver detects a pending overrun, it negates this pin. These pins default to GPIO input pulled high.
- $\overline{\text{CTS1}}$/PE7, $\overline{\text{CTS2}}$/PJ7—UART 1 and UART 2 Clear to Send or Port E bit 7 and Port J bit 7. $\overline{\text{CTS}}$ controls the transmitter. Normally, the transmitter waits until this signal is active (low) before a character is transmitted. If the NOCTSx bit is set in the UTX register, the transmitter sends a character whenever a character is ready to transmit. These pins default to GPIO input pulled high.

2.10 Timer Signals

There are several external timer and clock signal functions available using the MC68VZ328. This section describes the signals and how they are programmed.

- TOUT/TIN/PB6—Timer 1 Output, Timer 1 Input, or Port B bit 6. TOUT can be programmed to toggle or generate a pulse of 1-system-clock duration when the timer/counter reaches a reference value. TIN is used as the external clock source of Timer 1 or used as a capture function. This pin defaults to GPIO input pulled high.
- UCLK/$\overline{\text{DWE}}$/PE3—UART Clock input/output, DRAM Write-Enable, or Port E bit 3. The UCLK function is selected when $\overline{\text{DWE}}$ is disabled and PESEL3 is written 0. The direction of UCLK is controlled by the UCLKDIR bit of UART 1 and UART 2. For UCLK output, the UCLK bit of peripheral control register selects the clock output signal from UART 1 or UART 2. This pin defaults to GPIO input pulled high.

2.11 Pulse-Width Modulator Signals

There are two pulse-width modulator (PWM) modules in the MC68VZ328. This section describes the signals available to communicate with these PWM modules.

- PWMO1/PB7—Pulse-Width Modulator Output 1 or Port B bit 7. PWMO1 is an output signal from the logical operation (AND or OR) of both the PWM 1 and PWM 2 modules. This pin defaults to GPIO input pulled high.
- PWMO2/$\overline{\text{DATA\_READY}}$/PK0—Pulse-Width Modulator Output 2, SPI Data Ready, or Port K bit 0. PWMO2 is an output signal from the PWM 2 module. If this pin is configured for dedicated I/O function and PKDIR0 is 1, the PWMO2 signal is selected. If PKDIR0 is 0, SPI Data Ready ($\overline{\text{DATA\_READY}}$) is selected. This pin defaults to GPIO input pulled high.

2.12 Serial Peripheral Interface 1 Signals

There are two serial peripheral interface (SPI) modules in the MC68VZ328. This section describes the signals that are used with SPI 1 to interface with external devices.

- MOSI/PJ0—SPI Transmit Data or Port J bit 0. MOSI is the master output/slave input signal for the SPI shift register. This pin defaults to GPIO input pulled high.
- MISO/PJ1—SPI Receive Data or Port J bit 1. MISO is the master input/slave output signal for the SPI shift register. This pin defaults to GPIO input pulled high.
- SPICLK1/PJ2—SPI Clock or Port J bit 2. SPICLK1 is the master clock output/slave clock input signal for SPI. In polarity = 0 mode, this signal is low while the serial peripheral interface master is idle. In polarity = 1 mode, this signal is high during idle. This pin defaults to GPIO input pulled high.
- $\overline{\text{SS}}$/PJ3—SPI Slave Select or Port J bit 3. $\overline{\text{SS}}$ is the master output/slave input chip-select signal. This pin defaults to GPIO input pulled high.
- $\overline{\text{DATA\_READY}}$/PWMO2/PK0—SPI Data Ready or Port K bit 0. $\overline{\text{DATA\_READY}}$ can be used in master mode to signal the SPI master to clock out data. To select the $\overline{\text{DATA\_READY}}$ function, the PKDIR0 and PKSEL0 bits are written 0. This pin defaults to GPIO input pulled high.

2.13 Serial Peripheral Interface 2 Signals

This section describes the signals that are used with SPI 2, the second serial peripheral interface (SPI) module in the MC68VZ328, to interface with external devices.

- SPITXD/PE0—SPI Master Transmit Data or Port E bit 0. SPITXD is the master SPI shift register output signal. This pin defaults to GPIO input pulled high.
- SPIRXD/PE1—SPI Master Receive Data or Port E bit 1. SPIRXD is the input to the master SPI shift register. This pin defaults to GPIO input pulled high.
- SPICLK2/PE2—SPI Master Clock or Port E bit 2. SPICLK2 is the clock output when the serial peripheral interface master is enabled. In polarity = 0 mode, this signal is low while the serial peripheral interface master is idle. In polarity = 1 mode, this signal is high during idle. This pin defaults to GPIO input pulled high.

2.14 Chip-Select and EDO RAM Interface Signals

Chip-select logic is used to provide maximum compatibility with a wide variety of memory logic. This section and Section 2.15, “SDRAM Interface Signals,” describe the signals used to interface with RAM, SDRAM, and EDO RAM.

- $\overline{\text{CSA0}}$**—**Chip-Select A bit 0. $\overline{\text{CSA0}}$ is a default chip-select signal after reset. It is set to 6 wait states and decodes all address ranges, except internal register address space, emulator space, and bootstrap space (0xFFFC0000–0xFFFFFFFF). It can be reprogrammed during the boot sequence to another address range or different wait states. The default data bus width for $\overline{\text{CSA0}}$ is determined by the state of the BUSW signal.
- $\overline{\text{CSA1}}$/PF7, $\overline{\text{CSB}}$[1:0]/PB[1:0], $\overline{\text{CSC}}$[1:0]/PB[3:2]/$\overline{\text{RAS}}$[1:0], $\overline{\text{CSD}}$[1:0]/PB[5:4]/$\overline{\text{CAS}}$[1:0]**—**Chip-Select A, B, C, and D bits 0 and 1, Port F bit 7, Port B bits 5–0, or row and column select signals. These pins comprise the remainder of the Group A, B, C, and D chip-selects and are individually programmable. Pins that are not needed as chip-selects can be programmed as general-purpose I/O. In addition, $\overline{\text{CSC}}$[1:0] and $\overline{\text{CSD}}$[1:0] are designed to support DRAM as $\overline{\text{CAS}}$ and $\overline{\text{RAS}}$ signals. These pins default to GPIO input pulled high.
- PM5/$\overline{\text{DMOE}}$**—**Port M bit 5 or DRAM Continuous Page Mode Output Enable. $\overline{\text{DMOE}}$ is similar to the $\overline{\text{OE}}$ signal. However, $\overline{\text{DMOE}}$ only goes active on DRAM read cycles, while $\overline{\text{OE}}$ is active for all memory read cycles. In continuous page mode, $\overline{\text{RAS}}$ is held low until a page-miss, refresh required, or $\overline{\text{RAS}}$ duration time out. During an $\overline{\text{RAS}}$ low period there may be other memory access cycles, and if $\overline{\text{OE}}$ is used to enable the DRAM data output, DRAM will drive data, producing bus contention. Therefore, a dedicated output enable, $\overline{\text{DMOE}}$, is required, connecting to DRAM if continuous page mode is enabled. Using this mode will minimize the number of clocks per DRAM access. This pin defaults to GPIO input pulled high.

2.15 SDRAM Interface Signals

- $\overline{\text{CSD0}}$, $\overline{\text{CSD1}}$**—**These two signals are multiplexed with SDRAM $\overline{\text{CS0}}$ and $\overline{\text{CS1}}$. When SDRAM is enabled, $\overline{\text{CSD0}}$ and $\overline{\text{CSD1}}$ are SDRAM bank 1 and bank 2 chip-select signals. Also see Chapter 6, “Chip-Select Logic,” for more details.
- $\overline{\text{CSC0}}$**—**This signal is multiplexed with SDRAM $\overline{\text{RAS}}$. When SDRAM is enabled, this signal becomes an SDRAM $\overline{\text{RAS}}$ signal. For additional information about this subject, see Chapter 6, “Chip-Select Logic.”
- $\overline{\text{CSC1}}$**—**This signal is multiplexed with SDRAM $\overline{\text{CAS}}$. When SDRAM is enabled, this signal becomes an SDRAM $\overline{\text{RAS}}$ signal. For more details, see Chapter 6, “Chip-Select Logic.”
- $\overline{\text{SDWE}}$**—**SDRAM $\overline{\text{WE}}$. When SDRAM is enabled, this signal becomes an SDRAM Write-Enable signal. There is additional programming information about this subject in Chapter 6, “Chip-Select Logic.”
- PM0/SDCLK**—**Port M bit 0 or SDRAM Clock. This pin defaults to GPIO input pulled low.
- PM1/SDCE**—**Port M bit 1 or SDRAM Clock Enable. This pin defaults to GPIO pulled low.
- PM2/DQMH, PM3/DQML**—**Port M bits 2–3 or SDRAM input/output mask. These pins default to GPIO pulled low.
- PM4/SDA10**—**Port M bit 4 or SDRAM Address A10. This pin defaults to GPIO input pulled low.

2.16 In-Circuit Emulation (ICE) Signals

The ICE module is designed to support low-cost emulator designs using the MC68VZ328 microprocessor. There are four interface signals that are extended to external pins.

- $\overline{\text{HIZ}}$/P/$\overline{\text{D}}$/PG3—High Impedance, Program/Data, or Port G bit 3. During system reset, a logic low of this input signal will put the MC68VZ328 into Hi-Z mode, in which all MC68VZ328 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 GPIO input pulled high, but can be programmed as the P/$\overline{\text{D}}$ function. P/$\overline{\text{D}}$ is a status signal that shows whether the current bus cycle is in program space or in data space during emulation mode.
- $\overline{\text{EMUIRQ}}$/PG2—Emulator Interrupt Request or Port G bit 2. During system reset, a logic low of this input signal will put the MC68VZ328 into emulation mode, which is described in Chapter 16, "In-Circuit Emulation." For normal operation, this pin must be pulled high during system reset or left unconnected. After system reset, this pin defaults to an $\overline{\text{EMUIRQ}}$ function in normal or emulation mode. $\overline{\text{EMUIRQ}}$ is an active low, level 7 interrupt input signal.
- $\overline{\text{EMUBRK}}$/PG5—Emulator Breakpoint or Port G bit 5. During system reset, a logic low of this input signal will put the MC68VZ328 into bootstrap mode, which is described in Chapter 17, "Bootstrap Mode." For normal operation, this pin must be pulled high during system reset or left unconnected. After system reset, this pin defaults to the $\overline{\text{EMUBRK}}$ function, which is an I/O signal used in emulation mode for breakpoint control.
- $\overline{\text{EMUCS}}$/PG4—Emulator Chip-Select or Port G bit 4. $\overline{\text{EMUCS}}$ is an 8-bit data bus width chip-select signal that selects the dedicated memory space from 0xFFFC0000 to 0xFFFDFFFF. It cannot be used to select 16-bit data bus memory devices. $\overline{\text{EMUCS}}$ is not only activated in emulation mode, but in normal and bootstrap modes as well. See Chapter 16, "In-Circuit Emulation," for more information about $\overline{\text{EMUCS}}$ operation. This pin defaults to an $\overline{\text{EMUCS}}$ signal.

Chapter 3
Memory Map

The memory map is a guide to all on-chip resources. When you configure your chip, refer to Figure 3-1 and either Table 3-1 on page 3-2, which is sorted by address, or Table 3-2 on page 3-8, which is sorted alphabetically by register name.

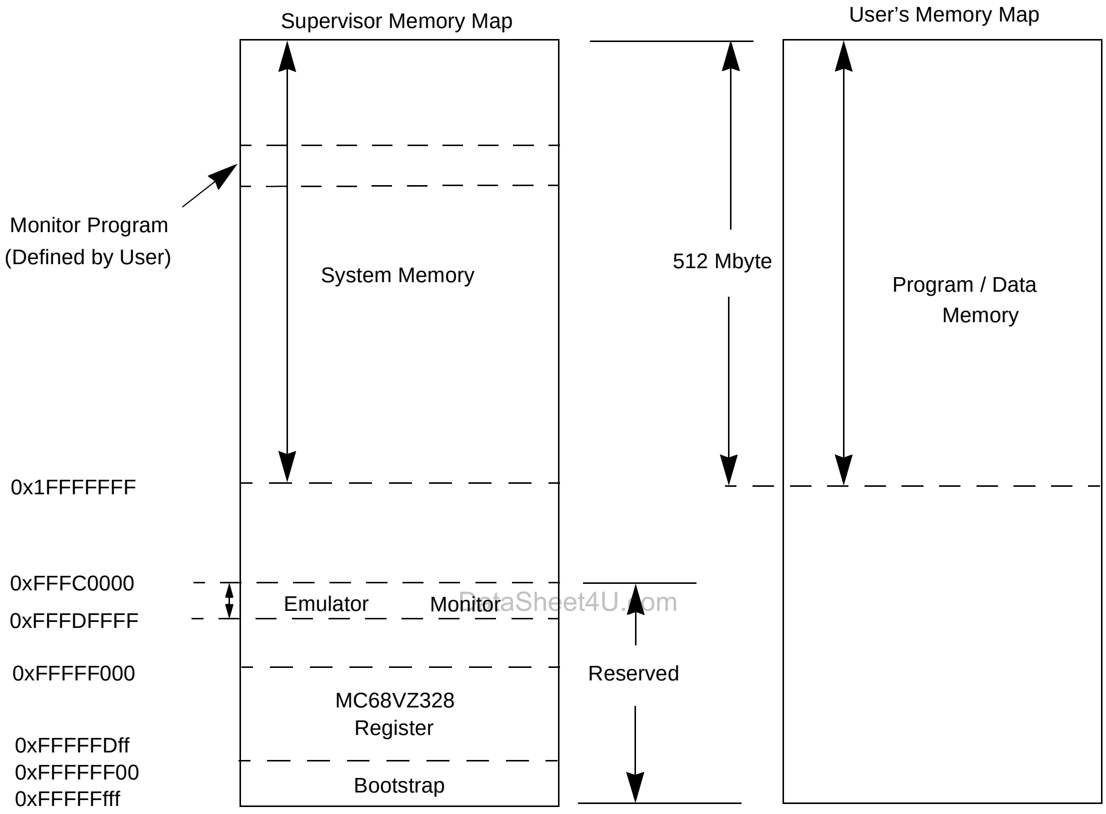


Figure 3-1. MC68VZ328 System Memory Map

3.1 Programmer’s Memory Map

On reset the base address used in the table is 0xFFFFF000 (or 0xXXFFF000, where XX is “don’t care”). If a double-mapped bit is cleared in the system control register, then the base address is 0xFFFFF000 only. Unpredictable results occur if you write to any 4K register space not documented in Table 3-1 or Table 3-2 on page 3-8.

Table 3-1. Programmer’s Memory Map (Sorted by Address)

Address	Name	Width	Description	Reset Value	Page Number
0xFFFFF000	SCR	8	System control register	0x1C	5-2
0xFFFFF003	PCR	8	Peripheral control register	0x00	5-4
0xFFFFF004	IDR	32	Silicon ID register	0x56000000	5-5
0xFFFFF008	IODCR	16	I/O drive control register	0x1FFF	5-6
0xFFFFF100	CSGBA	16	Chip-select group A base register	0x0000	6-4
0xFFFFF102	CSGBB	16	Chip-select group B base register	0x0000	6-4
0xFFFFF104	CSGBC	16	Chip-select group C base register	0x0000	6-4
0xFFFFF106	CSGBD	16	Chip-select group D base register	0x0000	6-4
0xFFFFF108	CSUGBA	16	Chip-select upper group address register	0x0000	6-6
0xFFFFF10A	CSCR	16	Chip-select control register	0x0000	6-16
0xFFFFF110	CSA	16	Group A chip-select register	0x00B0	6-8
0xFFFFF112	CSB	16	Group B chip-select register	0x0000	6-8
0xFFFFF114	CSC	16	Group C chip-select register	0x0000	6-8
0xFFFFF116	CSD	16	Group D chip-select register	0x0200	6-8
0xFFFFF118	EMUCS	16	Emulation chip-select register	0x0060	6-16
0xFFFFF200	PLLCR	16	PLL control register	0x24B3	4-8
0xFFFFF202	PLLFSR	16	PLL frequency select register	0x0347	4-10
0xFFFFF204	RES	—	Reserved	—	—
0xFFFFF207	PCTLR	8	Power control register	0x1F	4-14
0xFFFFF300	IVR	8	Interrupt vector register	0x00	9-7
0xFFFFF302	ICR	16	Interrupt control register	0x0000	9-8
0xFFFFF304	IMR	32	Interrupt mask register	0x00FFFFFF	9-10
0xFFFFF308	RES	32	Reserved	—	—
0xFFFFF30C	ISR	32	Interrupt status register	0x00000000	9-12

Table 3-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page Number
0xFFFFF310	IPR	32	Interrupt pending register	0x00000000	9-16
0xFFFFF314	ILCR	16	Interrupt level control register	0x6533	9-19
0xFFFFF400	PADIR	8	Port A direction register	0x00	10-6
0xFFFFF401	PADATA	8	Port A data register	0xFF	10-6
0xFFFFF402	PAPUEN	8	Port A pull-up enable register	0xFF	10-6
0xFFFFF403	RES	8	Reserved	—	—
0xFFFFF408	PBDIR	8	Port B direction register	0x00	10-8
0xFFFFF409	PBDATA	8	Port B data register	0xFF	10-8
0xFFFFF40A	PBPUEN	8	Port B pull-up enable register	0xFF	10-8
0xFFFFF40B	PBSEL	8	Port B select register	0xFF	10-8
0xFFFFF410	PCDIR	8	Port C direction register	0x00	10-11
0xFFFFF411	PCDATA	8	Port C data register	0x00	10-11
0xFFFFF412	PCPDEN	8	Port C pull-down enable register	0xFF	10-11
0xFFFFF413	PCSEL	8	Port C select register	0xFF	10-11
0xFFFFF418	PDDIR	8	Port D direction register	0x00	10-16
0xFFFFF419	PDDATA	8	Port D data register	0xFF	10-16
0xFFFFF41A	PDPUEN	8	Port D pull-up enable register	0xFF	10-16
0xFFFFF41B	PDSEL	8	Port D select register	0xF0	10-16
0xFFFFF41C	PDPOL	8	Port D polarity register	0x00	10-16
0xFFFFF41D	PDIRQEN	8	Port D interrupt request enable register	0x00	10-16
0xFFFFF41E	PDKBEN	8	Port D keyboard enable register	0x00	10-16
0xFFFFF41F	PDIRQEG	8	Port D interrupt request edge register	0x00	10-16
0xFFFFF420	PEDIR	8	Port E direction register	0x00	10-21
0xFFFFF421	PEDATA	8	Port E data register	0xFF	10-21
0xFFFFF422	PEPUEN	8	Port E pull-up enable register	0xFF	10-21
0xFFFFF423	PESEL	8	Port E select register	0xFF	10-21
0xFFFFF428	PFDIR	8	Port F direction register	0x00	10-24
0xFFFFF429	PFDATA	8	Port F data register	0xFF	10-25

Table 3-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page Number
0xFFFFF42A	PFPUEN	8	Port F pull-up/pull-down enable register	0xFF	10-27
0xFFFFF42B	PFSEL	8	Port F select register	0x87	10-27
0xFFFFF430	PGDIR	8	Port G direction register	0x00	10-28
0xFFFFF431	PGDATA	8	Port G data register	0x3F	10-28
0xFFFFF432	PGPUEN	8	Port G pull-up enable register	0x3D	10-30
0xFFFFF433	PGSEL	8	Port G select register	0x08	10-31
0xFFFFF438	PJDIR	8	Port J direction register	0x00	10-31
0xFFFFF439	PJDATA	8	Port J data register	0xFF	10-32
0xFFFFF43A	PJPUEN	8	Port J pull-up enable register	0xFF	10-33
0xFFFFF43B	PJSEL	8	Port J select register	0xEF	10-33
0xFFFFF440	PKDIR	8	Port K direction register	0x00	10-34
0xFFFFF441	PKDATA	8	Port K data register	0x0F	10-35
0xFFFFF442	PKPUEN	8	Port K pull-up/pull-down enable register	0xFF	10-36
0xFFFFF443	PKSEL	8	Port K select register	0xFF	10-36
0xFFFFF448	PMDIR	8	Port M direction register	0x00	10-37
0xFFFFF449	PMDATA	8	Port M data register	0x20	10-38
0xFFFFF44A	PMPUEN	8	Port M pull-up/pull-down enable register	0x3F	10-39
0xFFFFF44B	PMSEL	8	Port M select register	0x3F	10-40
0xFFFFF500	PWMC1	16	PWM unit 1 control register	0x0020	15-4
0xFFFFF502	PWMS1	16	PWM unit 1 sample register	0xxxxx	15-6
0xFFFFF504	PWMP1	8	PWM unit 1 period register	0xFE	15-7
0xFFFFF505	PWMCNT1	8	PWM unit 1 counter register	0x00	15-7
0xFFFFF506	RES	16	Reserved	—	—
0xFFFFF510	PWMC2	16	PWM unit 2 control register	0x0000	15-8
0xFFFFF512	PWMP2	16	PWM unit 2 period register	0x0000	15-9
0xFFFFF514	PWMW2	16	PWM unit 2 width register	0x0000	15-10
0xFFFFF516	PWMCNT2	16	PWM unit 2 counter register	0x0000	15-10

Table 3-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page Number
0xFFFFF600	TCTL1	16	Timer unit 1 control register	0x0000	12-6
0xFFFFF602	TPRER1	16	Timer unit 1 prescaler register	0x0000	12-8
0xFFFFF604	TCMP1	16	Timer unit 1 compare register	0xFFFF	12-9
0xFFFFF606	TCR1	16	Timer unit 1 capture register	0x0000	12-10
0xFFFFF608	TCN1	16	Timer unit 1 counter register	0x0000	12-11
0xFFFFF60A	TSTAT1	16	Timer unit 1 status register	0x0000	12-12
0xFFFFF610	TCTL2	16	Timer unit 2 control register	0x0000	12-6
0xFFFFF612	TPRER2	16	Timer unit 2 prescaler register	0x0000	12-8
0xFFFFF614	TCMP2	16	Timer unit 2 compare register	0xFFFF	12-9
0xFFFFF616	TCR2	16	Timer unit 2 capture register	0x0000	12-10
0xFFFFF618	TCN2	16	Timer unit 2 counter register	0x0000	12-10
0xFFFFF61A	TSTAT2	16	Timer unit 2 status register	0x0000	12-12
0xFFFFF700	SPIRXD	16	SPI unit 1 receive data register	0x0000	13-4
0xFFFFF702	SPITXD	16	SPI unit 1 transmit data register	0x0000	13-5
0xFFFFF704	SPICONT1	16	SPI unit 1 control/status register	0x0000	13-6
0xFFFFF706	SPIINTCS	16	SPI unit 1 interrupt control/status register	0x0000	13-8
0xFFFFF708	SPITEST	16	SPI unit 1 test register	0x0000	13-10
0xFFFFF70A	SPISPC	16	SPI unit 1 sample period control register	0x0000	13-11
0xFFFFF800	SPIDATA2	16	SPI unit 2 data register	0x0000	13-14
0xFFFFF802	SPICONT2	16	SPI unit 2 control/status register	0x0000	13-15
0xFFFFF900	USTCNT1	16	UART unit 1 status/control register	0x0000	14-10
0xFFFFF902	UBAUD1	16	UART unit 1 baud control register	0x003F	14-12
0xFFFFF904	URX1	16	UART unit 1 receiver register	0x0000	14-13
0xFFFFF906	UTX1	16	UART unit 1 transmitter register	0x0000	14-14
0xFFFFF908	UMISC1	16	UART unit 1 miscellaneous register	0x0000	14-16
0xFFFFF90A	NIPR1	16	UART unit 1 non-integer prescaler register	0x0000	14-18
0xFFFFF910	USTCNT2	16	UART unit 2 status/control register	0x0000	14-10

Table 3-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page Number
0xFFFFF912	UBAUD2	16	UART unit 2 baud control register	0x003F	14-12
0xFFFFF914	URX2	16	UART unit 2 receiver register	0x0000	14-13
0xFFFFF916	UTX2	16	UART unit 2 transmitter register	0x0000	14-14
0xFFFFF918	UMISC2	16	UART unit 2 miscellaneous register	0x0000	14-16
0xFFFFF91A	NIPR2	16	UART unit 2 non-integer prescaler register	0x0000	14-18
0xFFFFF91C	HMARK	16	UART unit 2 FIFO half mark register	0x0102	14-29
0xFFFFFA00	LSSA	32	LCD screen starting address register	0x00000000	8-10
0xFFFFFA05	LVPW	8	LCD virtual page width register	0xFF	8-11
0xFFFFFA08	LXMAX	16	LCD screen width register	0x03F0	8-12
0xFFFFFA0A	LYMAX	16	LCD screen height register	0x01FF	8-12
0xFFFFFA18	LCXP	16	LCD cursor X position register	0x0000	8-12
0xFFFFFA1A	LCYP	16	LCD cursor Y position register	0x0000	8-13
0xFFFFFA1C	LCWCH	16	LCD cursor width and height register	0x0101	8-14
0xFFFFFA1F	LBLKC	8	LCD blink control register	0x7F	8-14
0xFFFFFA20	LPICF	8	LCD panel interface configuration register	0x00	8-15
0xFFFFFA21	LPOLCF	8	LCD polarity configuration register	0x00	8-16
0xFFFFFA23	LACDRC	8	LACD rate control register	0x00	8-16
0xFFFFFA25	LPXCD	8	LCD pixel clock divider register	0x00	8-17
0xFFFFFA27	LCKCON	8	LCD clocking control register	0x00	8-18
0xFFFFFA29	LRRA	8	LCD refresh rate adjustment register	0xFF	8-18
0xFFFFFA2B	RES	8	Reserved	—	—
0xFFFFFA2D	LPOSR	8	LCD panning offset register	0x00	8-19
0xFFFFFA31	LFRCM	8	LCD frame rate control modulation register	0x00	8-19
0xFFFFFA33	LGPMR	8	LCD gray palette mapping register	0x84	8-20
0xFFFFFA36	PWMR	16	PWM contrast control register	0x0000	8-20
0xFFFFFA38	RMCR	8	Refresh mode control register	0x00	8-21
0xFFFFFA39	DMACR	8	DMA control register	0x62	8-22

Table 3-1. Programmer's Memory Map (Sorted by Address) (Continued)

Address	Name	Width	Description	Reset Value	Page Number
0xFFFFFB00	RTCTIME	32	RTC time of day register	0xXXXX00XX	11-3
0xFFFFFB04	RTCALRM	32	RTC alarm register	0x00000000	11-3
0xFFFFFB0A	WATCHDOG	16	Watchdog timer register	0x0001	11-4
0xFFFFFB0C	RTCCTL	8	RTC control register	0x0080	11-10
0xFFFFFB0E	RTCISR	16	RTC interrupt status register	0x0000	11-10
0xFFFFFB10	RTCIENR	16	RTC interrupt enable register	0x0000	11-12
0xFFFFFB12	STPWCH	8	Stopwatch minutes register	0x003F	11-14
0xFFFFFB1A	DAYR	16	RTC day count register	0x0xxx	11-6
0xFFFFFB1C	DAYALARM	16	RTC day alarm register	0x0000	11-8
0xFFFFFC00	DRAMMC	16	DRAM memory configuration register	0x0000	7-12
0xFFFFFC02	DRAMC	16	DRAM control register	0x0000	7-14
0xFFFFFC04	SDCTRL	16	SDRAM control register	0x003C	7-16
0xFFFFFC06	SDPWDN	16	SDRAM power down register	0x0000	7-18
0xFFFFFC80	RES	—	Reserved	—	—
0xFFFFFD00	ICEMACR	32	ICEM address compare register	0x00000000	16-4
0xFFFFFD04	ICEMAMR	32	ICEM address mask register	0x00000000	16-4
0xFFFFFD08	ICEMCCR	16	ICEM control compare register	0x0000	16-6
0xFFFFFD0A	ICEMCMR	16	ICEM control mask register	0x0000	16-6
0xFFFFFD0C	ICEMCR	16	ICEM control register	0x0000	16-8
0xFFFFFD0E	ICEMSR	16	ICEM status register	0x0000	16-10
0xFFFFFExx	Bootloader	—	Bootloader microcode space	—	—

Table 3-2. Programmer's Memory Map (Sorted by Register Name)

Name	Address	Width	Description	Reset Value	Page Number
Bootloader	0xFFFFFExx	—	Bootloader microcode space	—	—
CSA	0xFFFFF110	16	Group A chip-select register	0x00B0	6-8
CSB	0xFFFFF112	16	Group B chip-select register	0x0000	6-8
CSC	0xFFFFF114	16	Group C chip-select register	0x0000	6-8
CSCR	0xFFFFF10A	16	Chip-select control register	0x0000	6-16
CSD	0xFFFFF116	16	Group D chip-select register	0x0200	6-8
CSGBA	0xFFFFF100	16	Chip-select group A base register	0x0000	6-4
CSGBB	0xFFFFF102	16	Chip-select group B base register	0x0000	6-4
CSGBC	0xFFFFF104	16	Chip-select group C base register	0x0000	6-4
CSGBD	0xFFFFF106	16	Chip-select group D base register	0x0000	6-4
CSUGBA	0xFFFFF108	16	Chip-select upper group address register	0x0000	6-6
DAYALARM	0xFFFFFB1C	16	RTC day alarm register	0x0000	11-8
DAYR	0xFFFFFB1A	16	RTC day count register	0x0xxx	11-6
DMACR	0xFFFFFA39	8	DMA control register	0x62	8-22
DRAMC	0xFFFFFC02	16	DRAM control register	0x0000	7-14
DRAMMC	0xFFFFFC00	16	DRAM memory configuration register	0x0000	7-12
EMUCS	0xFFFFF118	16	Emulation chip-select register	0x0060	6-16
HMARK	0xFFFFF91C	16	UART unit 2 FIFO half mark register	0x0102	14-29
ICEMACR	0xFFFFFD00	32	ICEM address compare register	0x00000000	16-4
ICEMAMR	0xFFFFFD04	32	ICEM address mask register	0x00000000	16-4
ICEMCCR	0xFFFFFD08	16	ICEM control compare register	0x0000	16-6
ICEMCMR	0xFFFFFD0A	16	ICEM control mask register	0x0000	16-6
ICEMCR	0xFFFFFD0C	16	ICEM control register	0x0000	16-8
ICEMSR	0xFFFFFD0E	16	ICEM status register	0x0000	16-10
ICR	0xFFFFF302	16	Interrupt control register	0x0000	9-8
IDR	0xFFFFF004	32	Silicon ID register	0x56000000	5-5
ILCR	0xFFFFF314	16	Interrupt level control register	0x6533	9-19
IMR	0xFFFFF304	32	Interrupt mask register	0x00FFFFFF	9-10

Table 3-2. Programmer's Memory Map (Sorted by Register Name) (Continued)

Name	Address	Width	Description	Reset Value	Page Number
IODCR	0xFFFFF008	16	I/O drive control register	0x1FFF	5-6
IPR	0xFFFFF310	32	Interrupt pending register	0x00000000	9-16
ISR	0xFFFFF30C	32	Interrupt status register	0x00000000	9-12
IVR	0xFFFFF300	8	Interrupt vector register	0x00	9-7
LACDRC	0xFFFFFA23	8	LACD rate control register	0x00	8-16
LBLKC	0xFFFFFA1F	8	LCD blink control register	0x7F	8-14
LCKCON	0xFFFFFA27	8	LCD clocking control register	0x00	8-18
LCWCH	0xFFFFFA1C	16	LCD cursor width and height register	0x0101	8-14
LCXP	0xFFFFFA18	16	LCD cursor X position register	0x0000	8-12
LCYP	0xFFFFFA1A	16	LCD cursor Y position register	0x0000	8-13
LFRCM	0xFFFFFA31	8	LCD frame rate control modulation register	0x00	8-19
LGPMR	0xFFFFFA33	8	LCD gray palette mapping register	0x84	8-20
LPICF	0xFFFFFA20	8	LCD panel interface configuration register	0x00	8-15
LPOLCF	0xFFFFFA21	8	LCD polarity configuration register	0x00	8-16
LPOSR	0xFFFFFA2D	8	LCD panning offset register	0x00	8-19
LPXCD	0xFFFFFA25	8	LCD pixel clock divider register	0x00	8-17
LRRA	0xFFFFFA29	8	LCD refresh rate adjustment register	0xFF	8-18
LSSA	0xFFFFFA00	32	LCD screen starting address register	0x00000000	8-10
LVPW	0xFFFFFA05	8	LCD virtual page width register	0xFF	8-11
LXMAX	0xFFFFFA08	16	LCD screen width register	0x03F0	8-12
LYMAX	0xFFFFFA0A	16	LCD screen height register	0x01FF	8-12
NIPR1	0xFFFFF90A	16	UART unit 1 non-integer prescaler register	0x0000	14-18
NIPR2	0xFFFFF91A	16	UART unit 2 non-integer prescaler register	0x0000	14-18
PADATA	0xFFFFF401	8	Port A data register	0xFF	10-6
PADIR	0xFFFFF400	8	Port A direction register	0x00	10-6
PAPUEN	0xFFFFF402	8	Port A pull-up enable register	0xFF	10-6
PBDATA	0xFFFFF409	8	Port B data register	0xFF	10-8

Table 3-2. Programmer's Memory Map (Sorted by Register Name) (Continued)

Name	Address	Width	Description	Reset Value	Page Number
PBDIR	0xFFFFF408	8	Port B direction register	0x00	10-8
PBPUEN	0xFFFFF40A	8	Port B pull-up enable register	0xFF	10-8
PBSEL	0xFFFFF40B	8	Port B select register	0xFF	10-8
PCDATA	0xFFFFF411	8	Port C data register	0x00	10-11
PCDIR	0xFFFFF410	8	Port C direction register	0x00	10-11
PCPDEN	0xFFFFF412	8	Port C pull-down enable register	0xFF	10-11
PCR	0xFFFFF003	8	Peripheral control register	0x00	5-4
PCSEL	0xFFFFF413	8	Port C select register	0xFF	10-11
PCTLR	0xFFFFF207	8	Power control register	0x1F	4-14
PDDATA	0xFFFFF419	8	Port D data register	0xFF	10-16
PDDIR	0xFFFFF418	8	Port D direction register	0x00	10-16
PDIRQEG	0xFFFFF41F	8	Port D interrupt request edge register	0x00	10-16
PDIRQEN	0xFFFFF41D	8	Port D interrupt request enable register	0x00	10-16
PDKBEN	0xFFFFF41E	8	Port D keyboard enable register	0x00	10-16
PDPOL	0xFFFFF41C	8	Port D polarity register	0x00	10-16
PDPUEN	0xFFFFF41A	8	Port D pull-up enable register	0xFF	10-16
PDSEL	0xFFFFF41B	8	Port D select register	0xF0	10-16
PEDATA	0xFFFFF421	8	Port E data register	0xFF	10-21
PEDIR	0xFFFFF420	8	Port E direction register	0x00	10-21
PEPUEN	0xFFFFF422	8	Port E pull-up enable register	0xFF	10-21
PESEL	0xFFFFF423	8	Port E select register	0xFF	10-21
PFDATA	0xFFFFF429	8	Port F data register	0xFF	10-25
PFDIR	0xFFFFF428	8	Port F direction register	0x00	10-24
PFPUEN	0xFFFFF42A	8	Port F pull-up/pull-down enable register	0xFF	10-27
PFSEL	0xFFFFF42B	8	Port F select register	0x87	10-27
PGDATA	0xFFFFF431	8	Port G data register	0x3F	10-28
PGDIR	0xFFFFF430	8	Port G direction register	0x00	10-28
PGPUEN	0xFFFFF432	8	Port G pull-up enable register	0x3D	10-30

Table 3-2. Programmer's Memory Map (Sorted by Register Name) (Continued)

Name	Address	Width	Description	Reset Value	Page Number
PGSEL	0xFFFFF433	8	Port G select register	0x08	10-31
PJDATA	0xFFFFF439	8	Port J data register	0xFF	10-32
PJDIR	0xFFFFF438	8	Port J direction register	0x00	10-31
PJPUEN	0xFFFFF43A	8	Port J pull-up enable register	0xFF	10-33
PJSEL	0xFFFFF43B	8	Port J select register	0xEF	10-33
PKDATA	0xFFFFF441	8	Port K data register	0x0F	10-35
PKDIR	0xFFFFF440	8	Port K direction register	0x00	10-34
PKPUEN	0xFFFFF442	8	Port K pull-up/pull-down enable register	0xFF	10-36
PKSEL	0xFFFFF443	8	Port K select register	0xFF	10-36
PLLCR	0xFFFFF200	16	PLL control register	0x24B3	4-8
PLLFSR	0xFFFFF202	16	PLL frequency select register	0x0347	4-10
PMDATA	0xFFFFF449	8	Port M data register	0x20	10-38
PMDIR	0xFFFFF448	8	Port M direction register	0x00	10-37
PMPUEN	0xFFFFF44A	8	Port M pull-up/pull-down enable register	0x3F	10-39
PMSEL	0xFFFFF44B	8	Port M select register	0x3F	10-40
PWMC1	0xFFFFF500	16	PWM unit 1 control register	0x0020	15-4
PWMC2	0xFFFFF510	16	PWM unit 2 control register	0x0000	15-8
PWMCNT1	0xFFFFF505	8	PWM unit 1 counter register	0x00	15-7
PWMCNT2	0xFFFFF516	16	PWM unit 2 counter register	0x0000	15-10
PWMP1	0xFFFFF504	8	PWM unit 1 period register	0xFE	15-7
PWMP2	0xFFFFF512	16	PWM unit 2 period register	0x0000	15-9
PWMR	0xFFFFFA36	16	PWM contrast control register	0x0000	8-20
PWMS1	0xFFFFF502	16	PWM unit 1 sample register	0xxxxx	15-6
PWMW2	0xFFFFF514	16	PWM unit 2 width register	0x0000	15-10
RES	0xFFFFF204	—	Reserved	—	—
RES	0xFFFFF308	32	Reserved	—	—
RES	0xFFFFF403	8	Reserved	—	—
RES	0xFFFFF506	16	Reserved	—	—

Table 3-2. Programmer's Memory Map (Sorted by Register Name) (Continued)

Name	Address	Width	Description	Reset Value	Page Number
RES	0xFFFFFA2B	8	Reserved	—	—
RES	0xFFFFFC80	—	Reserved	—	—
RMCR	0xFFFFFA38	8	Refresh mode control register	0x00	8-21
RTCALRM	0xFFFFFB04	32	RTC alarm register	0x00000000	11-3
RTCCTL	0xFFFFFB0C	8	RTC control register	0x0080	11-10
RTCIENR	0xFFFFFB10	16	RTC interrupt enable register	0x0000	11-12
RTCISR	0xFFFFFB0E	16	RTC interrupt status register	0x0000	11-10
RTCTIME	0xFFFFFB00	32	RTC time of day register	0xXXXX00XX	11-3
SCR	0xFFFFF000	8	System control register	0x1C	5-2
SDCTRL	0xFFFFFC04	16	SDRAM control register	0x003C	7-16
SDPWDN	0xFFFFFC06	16	SDRAM power down register	0x0000	7-18
SPICONT1	0xFFFFF704	16	SPI unit 1 control/status register	0x0000	13-6
SPICONT2	0xFFFFF802	16	SPI unit 2 control/status register	0x0000	13-15
SPIDATA2	0xFFFFF800	16	SPI unit 2 data register	0x0000	13-14
SPIINTCS	0xFFFFF706	16	SPI unit 1 interrupt control/status register	0x0000	13-8
SPIRXD	0xFFFFF700	16	SPI unit 1 receive data register	0x0000	13-4
SPISPC	0xFFFFF70A	16	SPI unit 1 sample period control register	0x0000	13-11
SPITEST	0xFFFFF708	16	SPI unit 1 test register	0x0000	13-10
SPITXD	0xFFFFF702	16	SPI unit 1 transmit data register	0x0000	13-5
STPWCH	0xFFFFFB12	8	Stopwatch minutes register	0x003F	11-14
TCMP1	0xFFFFF604	16	Timer unit 1 compare register	0xFFFF	12-9
TCMP2	0xFFFFF614	16	Timer unit 2 compare register	0xFFFF	12-9
TCN1	0xFFFFF608	16	Timer unit 1 counter register	0x0000	12-11
TCN2	0xFFFFF618	16	Timer unit 2 counter register	0x0000	12-10
TCR1	0xFFFFF606	16	Timer unit 1 capture register	0x0000	12-10
TCR2	0xFFFFF616	16	Timer unit 2 capture register	0x0000	12-10
TCTL1	0xFFFFF600	16	Timer unit 1 control register	0x0000	12-6
TCTL2	0xFFFFF610	16	Timer unit 2 control register	0x0000	12-6

Table 3-2. Programmer's Memory Map (Sorted by Register Name) (Continued)

Name	Address	Width	Description	Reset Value	Page Number
TPRER1	0xFFFFF602	16	Timer unit 1 prescaler register	0x0000	12-8
TPRER2	0xFFFFF612	16	Timer unit 2 prescaler register	0x0000	12-8
TSTAT1	0xFFFFF60A	16	Timer unit 1 status register	0x0000	12-12
TSTAT2	0xFFFFF61A	16	Timer unit 2 status register	0x0000	12-12
UBAUD1	0xFFFFF902	16	UART unit 1 baud control register	0x003F	14-12
UBAUD2	0xFFFFF912	16	UART unit 2 baud control register	0x003F	14-12
UMISC1	0xFFFFF908	16	UART unit 1 miscellaneous register	0x0000	14-16
UMISC2	0xFFFFF918	16	UART unit 2 miscellaneous register	0x0000	14-16
URX1	0xFFFFF904	16	UART unit 1 receiver register	0x0000	14-13
URX2	0xFFFFF914	16	UART unit 2 receiver register	0x0000	14-13
USTCNT1	0xFFFFF900	16	UART unit 1 status/control register	0x0000	14-10
USTCNT2	0xFFFFF910	16	UART unit 2 status/control register	0x0000	14-10
UTX1	0xFFFFF906	16	UART unit 1 transmitter register	0x0000	14-14
UTX2	0xFFFFF916	16	UART unit 2 transmitter register	0x0000	14-14
WATCHDOG	0xFFFFFB0A	16	Watchdog timer register	0x0001	11-4

Chapter 4
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 comprises 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 MC68VZ328 and its associated peripherals.

The CGM uses a low-frequency oscillator in conjunction with a multiplier/divider chain to produce the clock signals used throughout the MC68VZ328 integrated processor. The frequency of all clock signals (except the low-frequency reference) are individually selectable through software control. The MC68VZ328 has four different power modes to provide optimum power efficiency.

The PCM controls the power consumption of the CPU by applying clock signals to the CPU at reduced burst widths. For maximum power savings, the MC68VZ328 can be placed in sleep mode in which all clocks (except for the low-frequency clock) are disabled.

NOTE:

The CGM module is designated as the PLL module in earlier versions of the DragonBall family. The nomenclature changed from PLL to CGM to be consistent with Motorola naming and standards conventions. The term PLL is used only to describe the actual PLL circuit within the CGM.

4.1 Introduction to the Clock Generation Module

The CGM produces four clock signals:

- CLK32—A low-frequency reference clock used by almost every module
- DMACLK—Used to create the remaining two clocks, and serves as DMA clock for the LCD controller
- SYSCLK—Used by most modules, including the CPU
- LCDCLK—Used as reference by the LCD

The distribution of the clock signals generated by the CGM is shown in Table 4-1. With the exception of the CLK32 signal, the frequency of the clock signals can be individually programmed.

Table 4-1. CGM Clock Signal Distribution

Used by or Available To	CLK32	SYSCLK	DMACLK	LCDCLK
CLKO/PF2 pin		X		
DRAM controller	X	X	X	
LCD controller			X	X
PCM	X			
PWM	X	X		
RTC	X			
SPIs		X		
Timers	X	X		
UARTs		X		

4.2 CGM Operational Overview

The CGM consists of six major parts, as shown in the simplified block diagram in Figure 4-1. The clock source for the CGM is a crystal oscillator that is comprised of an external crystal connected to the internal XTAL oscillator circuit. The output of the XTAL oscillator is the CLK32 signal, whose frequency is determined by the frequency of the external crystal. The CLK32 clock signal serves as a source for the PLL and many other modules within the MC68VZ328.

The output frequency of the PLL (PLLCLK) is determined by the frequency of CLK32 and by the values of the PC and QC fields of the PLL frequency select register (PLLFSR). The output of the PLL is applied to a divider chain composed of two prescalers. The PLLCLK clock is first input into prescaler 1. Its output frequency is selected by the prescaler select 1 (PRESC1) bit in the PLLCR. The output of the prescaler 1 (PR1CLK) is applied to prescaler 2, whose output frequency (DMACLK) is controlled by the prescaler select 2 (PRESC2) bit in the PLLCR. The DMACLK signal is applied to the LCD controller in the MC68VZ328 and also serves as the clock source for the LCD clock divider and the SYSCLK divider.

The output of the LCD clock divider is LCDCLK, whose frequency is controlled by the LCD clock selection (LCDCLK) field in the PLLCR. The LCDCLK signal is only used by the LCD controller. The SYSCLK divider produces a SYSCLK clock signal that is used throughout the MC68VZ328. SYSCLK is also used as the CPU clock signal (CPUCLK) by the internal FLX68000 CPU. SYSCLK is the only CGM-generated clock signal that can be made available to external devices via the buffered output of the clock out/Port F bit 2 pin (CLKO/PF2). See Section 10.4.7.3, “Port F Dedicated I/O Functions,” on page 10-26 for more information. The output is available when the clock enable bit of the PLLCR is enabled and bit 2 in the Port F select register (PFSEL) is cleared.

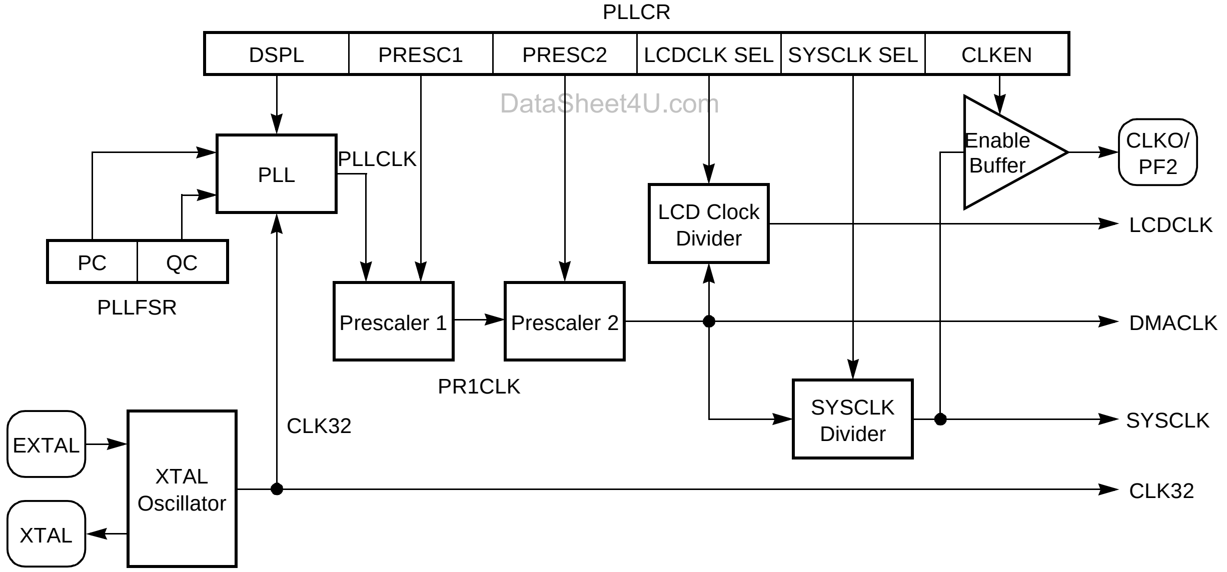


Figure 4-1. Clock Generation Module (CGM) Simplified Block Diagram

4.3 Detailed CGM Clock Descriptions

Section 4.3.1, “CLK32 Clock Signal,” and Section 4.3.2, “PLLCLK Clock Signal,” describe in detail the operation of each clock signal produced by the CGM.

4.3.1 CLK32 Clock Signal

The low-frequency output of the XTAL oscillator (CLK32) is available within a few hundred milliseconds after initial power is applied to the circuit. The frequency of the CLK32 signal is determined by the frequency of the external crystal. The CGM supports either a 32.768 kHz or a 38.4 kHz crystal.

NOTE:

> Regardless of the crystal frequency used, the output is always labeled CLK32.

Figure 4-2 represents a suggestion of how a crystal may be connected to the MC68VZ328. The values of C1 and C2 in Figure 4-2 are determined by using the crystal load capacitance (CL), PCB stray capacitance, Cstray (measured or approximated), and DragonBall input capacitance (Cdbvz << 1.0 pf) according to the following formula:

$$CL = Cstray + Cdbvz + (C1 * C2) / (C1 + C2) \qquad \textit{Eqn. 4-1}$$

Typical design values are C1 = C2 = 20 pf. The user should consult the crystal manufacturer for appropriate circuit layout and circuit values.

The CLK32 clock signal is unique in that while the other clock sources are disabled when the MC68VZ328 is placed in sleep mode, the CLK32 clock is available as long as power is applied. See Section 4.5.1.4, “Sleep Mode,” for detailed information on sleep mode.

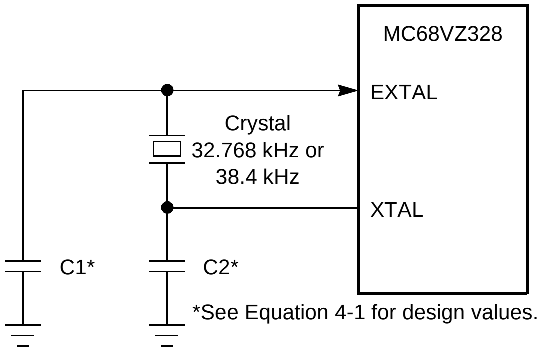


Figure 4-2. Example of External Crystal Connection

4.3.2 PLLCLK Clock Signal

The PLL output frequency, PLL clock (PLLCLK), is determined by a combination of the CLK32 signal’s input frequency and the values in the PC and QC fields of the PLLFSR. Section 4.3.2.2, “PLL Frequency Selection,” describes the procedure for frequency selection.

4.3.2.1 PLLCLK Initial Power-up Sequence

Refer to Figure 4-3 for a graphical representation of the following power-up sequence description. When power is initially applied to the MC68VZ328, the XTAL oscillator begins to oscillate. Due to the low-power design on the oscillator pads, the $\overline{\text{RESET}}$ signal must be asserted (low) for at least 1.2 s to ensure that the crystal oscillator starts and stabilizes. This is a significant change from the 250 ms required with the previous DragonBall and DragonBall EZ processors. The length of the delay (1.2 s) is an approximate value and should only be used as a starting point. The $\overline{\text{RESET}}$ pin (input) is a Schmitt trigger device with a threshold of 1.4 V high and 1.0 V low.

NOTE:

> On power up, the $\overline{\text{RESET}}$ signal should be deasserted after the crystal has energized and its output has stabilized, as shown in Figure 4-3. While most crystal oscillators typically operate with a value of 1.2 seconds, the optimum value will be determined experimentally. Due to the inherent nature of crystals, refer to manufacturers documentation for optimum circuit design information.

After $\overline{\text{RESET}}$ is deasserted, the PLLCLK signal is available to the divider chain, resulting in the availability of DMACLK from prescaler 2.

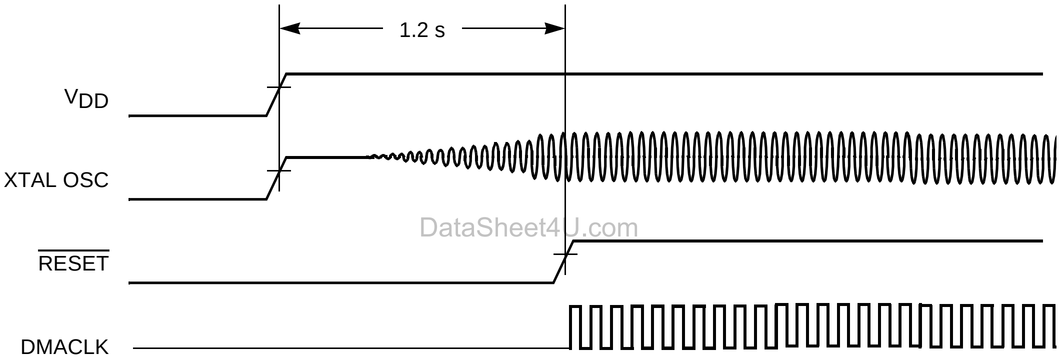


Figure 4-3. Initial Power-up Sequence Timing

4.3.2.2 PLL Frequency Selection

Using the default settings for the PC and QC fields of the PLLFSR and a CLK32 input frequency of 32.768 kHz produces a PLLCLK output of 66.322 MHz. For a 38.400 kHz crystal, the same default settings produce a 77.722 MHz PLLCLK. The PLLCLK clock is phase locked to the CLK32 clock input signal.

WARNING:

> The value of prescaler 1 must always be set to divide-by-two to prevent DMACLK and SYSCLK from operating beyond their design limits.

The PLL uses a dual-modulus counter to multiply the CLK32 frequency before it is input to the prescaler and the rest of the divider chain. Dual-modulus counters operate differently from other counters in that the overall multiplication ratio depends on two separate values, PC and QC.

In the following equation, the value of Q is defined as $1 \leq Q \leq 14$, and the value of P is defined as $P \geq Q + 1$.

$$\text{Multiplier} = 2(14(P+1)+Q+1) \qquad \textit{Eqn. 4-2}$$

For example, if Q = 3 and P = 71, then the following equations obtain:

Multiplier = 2 * (14 (71 + 1) + 3 + 1) = 2 * (1008 + 4) = 2024

2024 * 32.768 kHz = 66.322432 MHz

The default multiplier value is 2024. Using any multiplier equal to or greater than 794 (decimal) allows changing the PLLCLK in 32.768 kHz or 38.4 kHz steps. The minimum PC and QC values are P = 0x1B and Q = 0x04 (which produce a multiplier of 794 decimal).

4.3.2.3 PLLCLK Frequency Selection Programming Example

Example 4-1 on page 4-7 demonstrates the recommended sequence of events to change the PLLCLK frequency. The assumptions are:

- All peripherals have been disabled using chip-select. See Chapter 6, “Chip-Select Logic,” for details.
- SYSCLK is operating at the highest possible frequency (SYSCLK SEL = 100).

In Example 4-1, the variable NEWFREQ is the new frequency value (P and Q values) to be programmed. The MC68VZ328 is placed in sleep mode before the stop command. See Section 4.5.1.4, “Sleep Mode,” for detailed information about sleep modes. This routine enables the timer to wake up the PLL after 96 CLK32 periods. When the PLL wakes up, it will be at the new frequency. The interrupt service routine for the temporary timer interrupt should clear the timer interrupt and then return. In addition, the PLLCLK should only be changed during an early phase of the boot-up sequence.

NOTE:

> Example 4-1 is designed for clarity, and is not necessarily efficient.

Example 4-1. Configuring the PLLCLK Frequency

```
NEWFREQ equ somevalue                  ;P and Q value of new frequency
PLLCONTROL equ $FFFFF200               ;PLL Control Register
PLLFREQ equ $FFFFF202                  ;PLL Frequency Control Register
TCOMPARE equ $FFFFF604                 ;Timer Compare Value Register
TCONTROL equ $FFFFF600                 ;Timer Control Register
IMR equ $FFFFF304                      ;Interrupt Mask Register

             move.l IMR,-(SP)          ;save the Interrupt Mask register
             move.l #$fffffffd,IMR     ;enable ONLY Timer interrupt
             move.w #$0001,TCOMPARE    ;set compare value to 2
             move.w #$0119,TCONTROL    ;enable Timer 2 with CLK32 source
SYNC1        btst.b #$7,PLLFREQ        ;synchronize to CLK32 high level
             beq.s SYNC1               ;CLK32 is still not high, go back
SYNC2        btst.b #$7,PLLFREQ        ;synchronize to CLK32 low level
             bne.s SYNC2               ;CLK32 is still not low, go back
             move.w #NEWFREQ,PLLFREQ   ;load the new frequency
             ori.b #$8,PLLCONTROL+1    ;disable the PLL (in 30 clocks)-sleep mode
             stop #$2000               ;stop, enable all interrupts
; the PLL shuts down here and waits for the Timer interrupt
; interrupt service for Timer occurs here
             move.w (SP)+,IMR          ;restore the Interrupt Mask Register
             rts                       ;PLL is now at the new frequency
; The PLL has reacquired lock and SYSCLK is stable
```

4.3.2.4 Programming Considerations When Changing Frequencies

The following information is provided to assist the user in programming the MC68VZ328.

- When programming the SYSCLK frequency, ensure that it does not exceed 33.161216 MHz at any time.
- Since the PRESC1 and PRESC2 bits are set to %1 by default, the DMACLK output is approximately 16 MHz.
- Because most of the modules—such as the UART, SPI, general-purpose timers, and PWM—use the SYSCLK for bit-rate generation, changing the PLLCLK frequency will also change SYSCLK and overall system timing (except for CLK32). Therefore, once a PLLCLK frequency is selected, it should not be changed during system operation.
- To reduce power consumption, the output of the PLL can be disabled using the DISPLL bit in the PLL control register, which places the chip in sleep mode. See Section 4.5.1.4, “Sleep Mode,” for more details. When the MC68VZ328 is awakened from sleep mode by a wake-up event, the PLL output (PLLCLK) is available after a delay determined by the setting in the WKSEL field of PLLCR. Unlike the initial power-up sequence, the crystal oscillator is already on, so the crystal startup time is not a factor.

4.4 CGM Programming Model

This section describes the two registers that enable and control the frequency of the CGM clocks.

4.4.1 PLL Control Register

The PLL control register (PLLCR) controls the frequency selection of the LCDCLK, SYSCLK, and DMACLK. It also enables the output of the PLL and clock out/Port F pin 2 (CLKO/PF2). The settings for each bit and field in the register are described in Table 4-2.

PLLCR PLL Control Register **0xFFFFF200**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
			LCDCLK SEL			SYSCLK SEL			PRESC1		PRESC2	CLKEN	DISPLL		WKSEL	
TYPE			rw	rw	rw	rw	rw	rw	rw		rw	rw	rw		rw	rw
RESET	0	0	1	0	0	1	0	0	1	0	1	1	0	0	1	1

0x24B3

Table 4-2. PLL Control Register Description

Name	Description	Setting
Reserved Bits 15–14	Reserved	These bits are reserved and should be set to 0.
LCDCLK SEL Bits 13–11	**LCD Clock Select**—This field controls the divide ratio used by the LCD clock divider to convert DMACLK to LCDCLK. This field can be changed at any time.	000 = DMACLK ÷ 2. 001 = DMACLK ÷ 4. 010 = DMACLK ÷ 8. 011 = DMACLK ÷ 16. 1xx = DMACLK ÷ 1 (%100 after reset).
SYSCLK SEL Bits 10–8	**System Clock Select**—This field controls the divide ratio used by the SYSCLK divider to convert DMACLK to SYSCLK. This field can be changed at any time.	000 = DMACLK ÷ 2. 001 = DMACLK ÷ 4. 010 = DMACLK ÷ 8. 011 = DMACLK ÷ 16. 1xx = DMACLK ÷ 1 (%100 after reset).
PRESC1 Bit 7	**Prescaler 1 Select**—This bit selects the divide ratio of the prescaler 1.	0 = PLLCLK ÷ 1. 1 = PLLCLK ÷ 2 (default).
Reserved Bit 6	Reserved	This bit is reserved and should be set to 0.
PRESC2 Bit 5	**Prescaler 2 Select**—This bit selects the divide ratio used by the prescaler 2 to divide the output of prescaler 1, producing DMACLK. This field can be changed at any time.	0 = PR1CLK ÷ 1. 1 = PR1CLK ÷ 2 (default).
CLKEN Bit 4	**Clock Enable**—This bit enables the buffered output of the SYSCLK at the CLKO/PF2 pin when bit 2 of the PFSEL register is also cleared.	0 = CLKO enabled. 1 = CLKO disabled (default).

Table 4-2. PLL Control Register Description (Continued)

Name	Description	Setting
DISPLL Bit 3	**Disable PLL**—This bit, when set, disables the output of the PLL, placing the chip in sleep mode, its lowest power state.	0 = PLL enabled (default). 1 = PLL disabled.
Reserved Bit 2	Reserved	This bit is reserved and should be set to 0.
WKSEL Bits 1–0	**Wake-up Clock Select**—This field selects the delay of the PLL output from the initiation of the wake up until an output is available. Since the delay time is calculated by counting CLK32 cycles, the frequency of the crystal oscillator will determine the amount of delay that each setting produces.	See Table 4-3 for delay settings.

Table 4-3. WKSEL Field (PLLCR) Delay Settings

Bits 1–0	CLK32 Periods	Delay in Milliseconds (32.768 kHz)	Delay in Milliseconds (38.4 kHz)
00	32	0.976	0.833
01	48	1.465	1.250
10	64	1.953	1.667
11	96	2.93 (default)	2.500 (default)

4.4.2 PLL Frequency Select Register

The PLL frequency select register (PLLFSR) controls the two dividers of the dual-modulus counter. It also contains the write-protect bit for the QC and PC counters and the CLK32 status bit. Although PLLFSR register can be accessed in bytes, it should always be written as a 16-bit word. The settings for each bit and field in the register is described in Table 4-4.

PLLFSR PLL Frequency Select Register **0x(FF)FFF202**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	CLK32	PROT			QC				PC							
TYPE	r	rw*			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1	1
							0x0347									

*This bit can be set by software but is cleared only by reset.

Table 4-4. PLL Frequency Select Register Settings

Name	Description	Setting
CLK32 Bit 15	**Clock32 Status**—This read-only bit indicates the status of the CLK32 clock signal. The bit switches with each cycle of the CLK32 clock.	0 = CLK32 low. 1 = CLK32 high.
PROT Bit 14	**Protect Bit**—This bit write protects the QC and PC fields of the PLLFSR. After this bit is set by software, the register is write protected until a reset clears this bit.	0 = PLLFSR is not protected. 1 = PLLFSR is write protected.
Reserved Bits 13–12	Reserved	These bits are reserved and must remain at their default value.
QC Bits 11–8	**Q Counter**—This field contains the Q value that is used by the PLL to produce the PLLCLK.	Field value range is 1 $\leq$ Q $\leq$ 14.
PC Bits 7–0	**P Counter**—This field contains the P value that is used by the PLL to produce the PLLCLK.	Field value range is P $\geq$ Q + 1.

4.5 Introduction to the Power Control Module

The purpose of the power control module (PCM) is to optimize the power consumption of the FLX68000 CPU by turning the CPU off for a programmed number of clock pulses. The CPU consumes more power than any component in the MC68VZ328, so to conserve power while the CPU is relatively idle, the PCM can disable the CPU clock or apply the clock in bursts. When the MC68VZ328 is in one of these reduced-power modes, it is restored to normal operation by a wake-up event. When this occurs, the clock is immediately enabled, allowing the CPU to service the request. The DMA controller is not affected by the PCM having full access to the bus while the CPU is idle, keeping the LCD screen refreshed.

4.5.1 Operating the PCM

The power control module has four modes of operation: normal, burst, doze and sleep. In normal mode, the PCM is off. The MC68VZ328 enters burst mode when the PCM is enabled. In burst mode, the PCM controls the burst width of the CPUCLK signal to the CPU. If the burst width of the CPU clock is reduced to zero, CPUCLK is disabled and the MC68VZ328 is in doze mode. The lowest power mode setting is sleep mode. It is entered by setting the disable PLL (DISPLL) bit in the PLLCR, which disables the PLL and thus disables every clock signal in the CGM except CLK32. Section 4.5.1.1, “Normal Mode,” through Section 4.5.1.4, “Sleep Mode,” give detailed information about each of the four power modes.

4.5.1.1 Normal Mode

After reset, the PCM is disabled, the CPU clock runs continuously, and the MC68VZ328 consumes maximum power. This is normal mode.

4.5.1.2 Burst Mode

Setting the PCEN bit in the power control register (PCTRL) enables the PCM, causing the clock burst width of the CPU clock to be under the control of the PCTLR WIDTH settings in increments of 3 percent (one thirty-first of a cycle). Initially, the burst width is set to 100 percent. Software can then change the burst width to a lower value, and the clock is applied to the CPU in bursts. The burst-width register can be programmed for burst widths of any value between zero thirty-firsts and thirty-one thirty-firsts. This effectively produces a system clock with a variable burst width (and power dissipation) between 3 percent and 100 percent in incremental steps of 3 percent.

When the PCM is enabled, if a wake-up event is received, the PCM is immediately disabled, restoring the continuous CPU clock. It is the responsibility of the wake-up service routine to reenable the PCM.

4.5.1.3 Doze Mode

Setting the width field of PCTLR to %00000 reduces the burst width of the CPU clock to zero, causing the MC68VZ328 to enter doze mode. As with burst mode, the CPUCLK is immediately enabled when it receives a wake-up event. At the end of the service routine, the PCM can be reenabled with a width of %00000, putting the CPU back into doze mode. Once the CPU is placed in doze mode, only a wake-up event or hardware reset will reenable it.

NOTE:

The most effective power-control strategy is to run the CPU in normal mode until CPU action is not needed and then to enter doze mode by writing 0x80 into the PCTLR. This disables the CPU clock at the earliest possible moment, but allows the CPU to immediately respond to wake-up events. The peripheral devices, including the LCD controller, are not affected by the PCM.

4.5.1.4 Sleep Mode

Unlike burst or doze mode, sleep mode disables all of the clocks in the MC68VZ328 with the exception of the CLK32. The output of the PLL in the CGM is disabled in sleep mode through setting the DISPLL bit in the PLLCR register. Only the 32 kHz clock works to keep the real-time clock operational. Wake-up events activate the PLL, and the system clock starts operating after a delay determined by the WKSEL setting in the PLLCR.

Other events that occur during sleep mode include:

- All Address Bus signals are in the active state of the last bus cycle.
- All data bus pins (D15–D0) are individually pulled up with 1-megaohm resistors.
- If CLK32 is selected as the clock source, the general-purpose timer operates even while the PLL is in sleep mode.
- The RTC interrupt status register can post interrupts while the system clock is in doze or sleep mode.

4.5.2 CGM Operation During Sleep Mode

Shutting down the PLL to place the system in sleep mode is similar to the process used to change the frequency. The difference is that the system can be awakened only by a wake-up event or reset. Before shutting the PLL down, make sure that all peripheral devices are prepared for shutdown. The PLL shuts down 30 clock cycles of SYSCLK after the DISPLL bit is set in the PLLCR, allowing sufficient time to execute the stop instruction. When a wake-up event occurs, the PLL is enabled, and after a delay determined by the WKSEL setting in the PLLCR, the PLLCLK begins, as do as the rest of the clocks in the divider chain of the CGM. The CPU executes an interrupt service routine for the level of the wake-up event.

After the rte instruction in the wake-up service routine, the CPU returns and starts execution on the instruction following the stop instruction. Example 4-2 illustrates a typical shutdown sequence. It assumes that all peripherals have been shut down before the PLL is stopped.

Example 4-2. Shutdown Example

```
IRQMASK equ wake-up_mask_level

ori.b #$8,PLLCONTROL+1           ;disable the PLL (in 30 clocks)
stop #IRQMASK                    ;stop, enable wake-up events

;the PLL shuts down here
;The PLL has reacquired lock and SYSCLK is stable
;interrupt service occurs here

rts                              ;the system is operating
```

4.5.3 Burst Mode Operation

Figure 4-4 on page 4-13 shows a simplified block diagram of the PCM. When operating at 100 percent, the SYSCLK input is unaffected by burst-width control appearing as CPUCLK from the clock control. When a value has been placed in the width field of the PCTLR, the burst-width control allows the SYSCLK signal through to the clock control until the CPU clock's time slot has expired and is to be disabled. At that time the clock control requests the bus from the CPU. After the bus is granted, the CPUCLK stops. A bus grant to the DMA controller is asserted, allowing the DMA controller complete access to the bus.

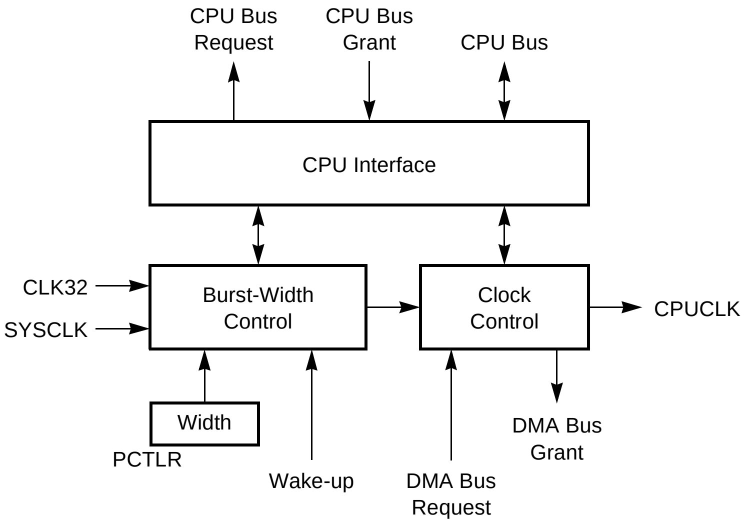


Figure 4-4. Power Control Module Block Diagram

If a wake-up event occurs while CPUCLK is disabled, the PCM is disabled and CPUCLK is immediately restored, allowing the CPU to process the event. The DMA controller always has priority, so if a DMA access is in progress, the CPU will wait until the DMA controller has completed its access before servicing the wake-up routine. Note that the LCD DMA controller has access to the bus at all times and the SYSCLK (master clock to all peripherals) is continuously active.

Figure 4-5 illustrates how the PCM operates. As described previously, a width setting of %11111 represents 31 periods of CLK32, or approximately 1 ms. In this example, the width setting in the PCTLR is 00011. The clock bursts are applied at a burst width of three thirty-firsts, or approximately at 10 percent on time, making the CPU active about 10 percent of the time. The remainder of the time, the CPU is in doze mode. When a wake-up event occurs, CPUCLK immediately returns to 100 percent so the CPU can service the wake-up event interrupt.

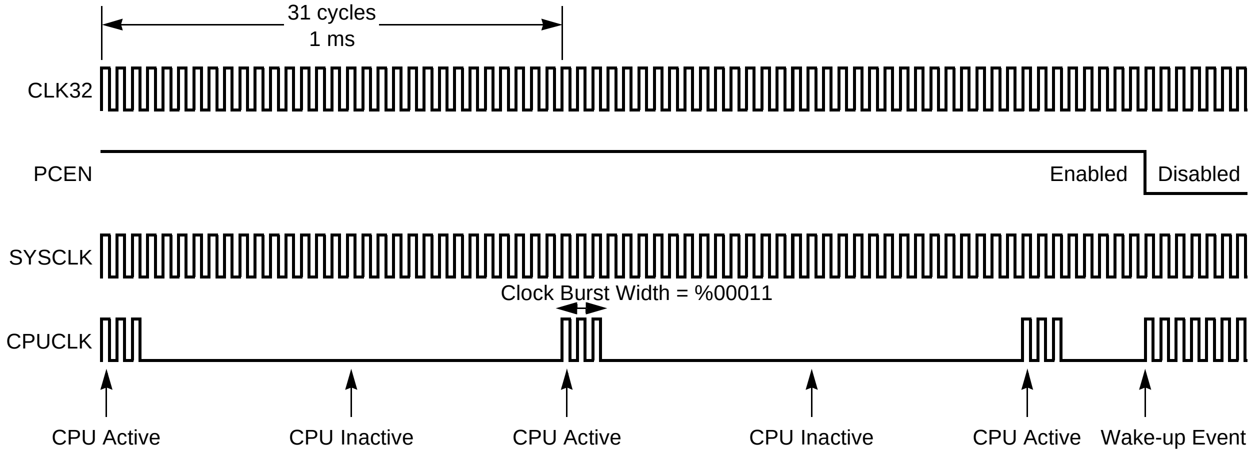


Figure 4-5. Power Control Operation in Burst Mode

4.5.4 Power Control Register

The power control register (PCTLR) enables the power control module and determines when the CPUCLK signal is applied to the CPU. The settings for each bit and field in the register are described in Table 4-5.

PCTLR 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 4-5. Power Control Register Description

Name	Description	Setting
PCEN Bit 7	**Power Control Enable**—This bit controls the operation of the power control module. While 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 power control module.	0 = Power control is disabled (default). 1 = Power control is enabled.
Reserved Bits 6–5	Reserved	These bits are reserved and should remain set to 0.
WIDTH Bits 4–0	**Width**—This field controls the width of the CPU clock bursts in increments of one thirty-first. While this bit is set to 1 and the PCM is enabled, the clock is applied to the CPU in burst widths of one thirty-first (3 percent). When the width field is 0x1F, the clock is always on, and when it is 0, the clock is always off. You can immediately wake it up again without waiting for the PLL to reacquire lock. The contents of this field are not affected by the PCEN bit. When an interrupt disables the power control module, these bits are not changed.	00000 = 0/31 clock burst width. 00001 = 1/31 clock burst width. 00010 = 2/31 clock burst width. . . . 11111 = 31/31 clock burst width.

Chapter 5
System Control

This chapter describes the system control register of the MC68VZ328 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)

5.1 System Control Operation

The on-chip resources use a reserved 4,096-byte block of address space for their registers. This block is mapped beginning at location 0xFFFFF000 (32-bit) or 0xXXFFF000 (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 0xFFFFF000.

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.

5.1.1 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 MC68VZ328 if it is enabled and not cleared or disabled before reaching terminal count. The watchdog timer is enabled at reset.

5.2 Programming Model

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

5.2.1 System Control Register

The 8-bit read/write system control register (SCR) resides at the address 0xFFFFF000 or 0xXXFFF000 (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 5-1.

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

	BIT 7	6	5	4	3	2	1	BIT 0
	BETO	WPV	PRV	BETEN	SO	DMAP		WDTH8
TYPE	rw	rw	rw	rw	rw	rw		rw
RESET	0	0	0	1	1	1	0	0
	0x1C							

Table 5-1. 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 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 5-1. 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 0xFFFFF000–0xFFFFFFFF. 1 = The on-chip registers are mapped at 0xFFFFF000–0xFFFFFFFF and 0xXXFFF000–0xXXFFFFF (XX = “don’t care”).
Reserved Bit 1	Reserved	This bit is reserved and reads 0.
WDTH8 Bit 0	**8-Bit Width Select**—This control bit allows the D[7:0] pins to be used for Port A input/output.	0 = Not an 8-bit system. 1 = 8-bit system.

5.2.2 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 5-2.

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 5-2. 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.
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 12.1.4, "TOUT/TIN/PB6 Pin," on page 12-3.	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).

5.2.3 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 5-3.

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	1	1	0	0	0	0	0	0	0	0	0
	0x5600															

	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 5-3. 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.	See description
SWID Bits 15–0	**Software ID**—This field contains the custom software ID. It is normally “0000.”	See description

5.2.4 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 2 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 5-4.

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
				AB	DB	CB	PM	PK	PJ	PG	PF	PE	PD	PC	PB	PA
TYPE				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
							0x1FFF									

Table 5-4. I/O Drive Control Register Description

Name	Description	Setting
Reserved Bits 15–13	Reserved	Do not use these bits.
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 2 mA. 1 = I/O drive current for each pin is 4 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 2 mA. 1 = I/O drive current for each pin is 4 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 2 mA. 1 = I/O drive current for each pin is 4 mA.
PM–PA Bits 9–0	**Port M to Port A 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 2 mA. 1 = I/O drive current for each pin is 4 mA.

Chapter 6
Chip-Select Logic

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

6.1 Overview of the CSL

The MC68VZ328 microprocessor contains eight general-purpose, programmable chip-select signals, which are used to select external devices on the address and data bus. The signals are arranged in four groups of two—$\overline{\text{CSA}}$[1:0], $\overline{\text{CSB}}$[1:0], $\overline{\text{CSC}}$[1:0], and $\overline{\text{CSD}}$[1:0].

$\overline{\text{CSA0}}$ 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 C ($\overline{\text{CSC0}}$/$\overline{\text{CSC1}}$) and Group D ($\overline{\text{CSD0}}$/$\overline{\text{CSD1}}$) chip-selects are unique in that they can also be programmed as row address strobe ($\overline{\text{RAS0}}$/$\overline{\text{RAS1}}$) and column address strobe ($\overline{\text{CAS0}}$/$\overline{\text{CAS1}}$) for the DRAM interface. For details, refer to Section 7.3.2, "DRAM Control Register," on page 7-14 and Section 6.3.3, "Chip-Select Registers," in this chapter.

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, Fast Page Mode, or synchronous) are supported, as shown in Table 6-1 on page 6-2.

Table 6-1. Chip-Select and Memory Types

Chip-Select Signal	Memory Supported
$\overline{\text{CSA0}}$	ROM, SRAM, flash memory chip
$\overline{\text{CSA1}}$	ROM, SRAM, flash memory chip
$\overline{\text{CSB0}}$	ROM, SRAM, flash memory chip
$\overline{\text{CSB1}}$	ROM, SRAM, flash memory chip
$\overline{\text{CSC0}}/\overline{\text{RAS0}}$	DRAM, ROM, SRAM, flash memory chip-select
$\overline{\text{CSC1}}/\overline{\text{RAS1}}$	DRAM, ROM, SRAM, flash memory chip-select
$\overline{\text{CSD0}}/\overline{\text{CAS0}}$	DRAM, ROM, SRAM, flash memory chip-select
$\overline{\text{CSD1}}/\overline{\text{CAS1}}$	DRAM, ROM, SRAM, flash memory chip-select

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 ($\overline{\text{AS}}$) 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 (32K, 64K, 128K, 256K, 512K, 1 Mbyte, 2 Mbyte, 4 Mbyte, 8 Mbyte, or 16 Mbyte). These memory ranges represent the most popular memory sizes available on the market and apply to the registers CSB, CSC, and CSD. The CSA register primarily supports ROM, which is usually 128K to 16 Mbyte. 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 number of wait states (from 0 to 13), data bus size selection, and whether a $\overline{\text{DTACK}}$ signal is automatically generated for the chip-select logic.

6.2 Chip-Select Operation

A chip-select output signal is asserted when an address is matched and after the $\overline{\text{AS}}$ 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, an 8K block size must begin on an 8K boundary, and a 64K block size can only begin on a 64K 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 during interrupt acknowledge (Function Code 7) cycles.

6.2.1 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 6-1. This area can be defined by programming the UPSIZ bits in the CSB, CSC, and CSD registers to between 32K and the entire chip-select area.

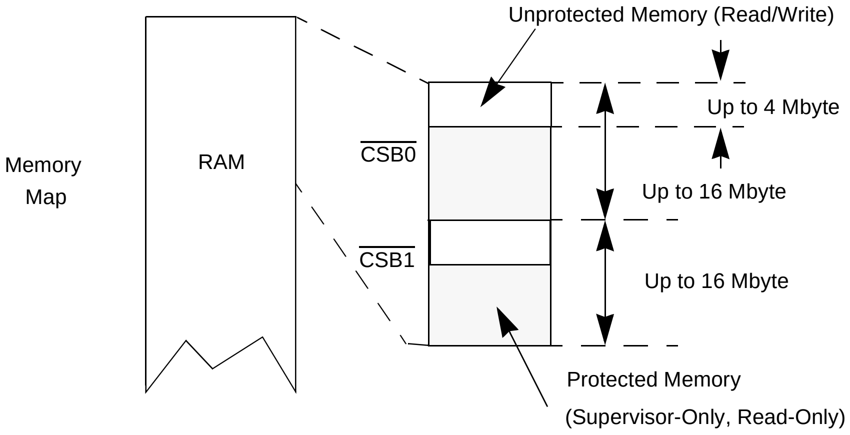


Figure 6-1. Size Selection and Memory Protection for $\overline{\text{CSB0}}$ and $\overline{\text{CSB1}}$

6.2.2 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 $\overline{\text{CSA0}}$ and $\overline{\text{EMUCS}}$, 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 (0xFFFFF000 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 nonvalid, so they will not assert until they are programmed and the EN bit is set in the chip-select registers.

6.2.3 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.

6.3 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.

6.3.1 Chip-Select Group Base Address Registers

The upper 15 bits of each base address register selects 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
	GB A28	GB A27	GB A26	GB A25	GB A24	GB A23	GB A22	GB A21	GB A20	GB A19	GB A18	GB A17	GB A16	GB A15	GB A14	
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 6-2. Chip-Select Group A Base Address Register Description

Name	Description	Setting
GBAx Bits 15–1	**Group A Base Address**—These bits select the high-order bits (28–14) 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 Bit 0	Reserved	This bit is 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
	GB B28	GB B27	GB B26	GB B25	GB B24	GB B23	GB B22	GB B21	GB B20	GB B19	GB B18	GB B17	GB B16	GB B15	GB B14	
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 6-3. Chip-Select Group B Base Address Register Description

Name	Description	Setting
GBBx Bits 15–1	**Group B Base Address**—These bits select the high-order bits (28–14) 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 Bit 0	Reserved	This bit is 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
	GB C2 8	GB C2 7	GB C2 6	GB C2 5	GB C2 4	GB C2 3	GB C2 2	GB C2 1	GB C2 0	GB C1 9	GB C1 8	GB C1 7	GB C1 6	GB C1 5	GB C1 4	
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. Chip-Select Group C Base Address Register Description

Name	Description	Setting
GBCx Bits 15–1	**Group C Base Address**—These bits select the high-order bits (28–14) 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 Bit 0	Reserved	This bit is 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
	GBD28	GBD27	GBD26	GBD25	GBD24	GBD23	GBD22	GBD21	GBD20	GBD19	GBD18	GBD17	GBD16	GBD15	GBD14	
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 6-5. Chip-Select Group D Base Address Register Description

Name	Description	Setting
GBDx Bits 15–1	**Group D Base Address**—These bits select the high-order bits (28–14) 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 Bit 0	Reserved	This bit is reserved and should be set to 0.

6.3.2 Chip-Select Upper Group Base Address Register

The default setting for chip-select decoding limits addressing to A28. When the full address decode enable (UGEN) bit is set, it allows full address decoding. Full address decoding is enabled for all four of the chip-select registers by the UGEN bit in the chip-select upper group base address register (CSUGBA). The bit value of the MSB for each of the four chip-select registers can be written into each of the four MSB fields in this register. The settings for this register are shown in Table 6-6.

CSUGBA Chip-Select Upper Group Base Address Register 0x(FF)FFF108

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	UGEN	AGBA[31:29]				BGBA[31:29]				CGBA[31:29]				DGBA[31:29]		
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 6-6. Chip-Select Upper Group Base Address Register Description

Name	Description	Setting
UGEN Bit 15	**Full Address Decode Enable**—This bit enables full address range decoding for all chip-select registers.	0 = Ignores A31, A30, and A29. 1 = Decoding includes A31, A30, and A29.
AGBA[31:29] Bits 14–12	**MSB for Chip-Select A**—The upper most significant bits for chip-select group A base address. The value will be ignored if UGEN is disabled.	Enter value for bits 31–29 of chip-select register A.

Table 6-6. Chip-Select Upper Group Base Address Register Description (Continued)

Name	Description	Setting
Reserved Bit 11	Reserved	This bit is reserved and should be set to 0.
BGBA[31:29] Bits 10–8	**MSB for Chip-Select B**—The upper most significant bits for chip-select group B base address. The value will be ignored if UGEN is disabled.	Enter value for bits 31–29 of chip-select register B.
Reserved Bit 7	Reserved	This bit is reserved and should be set to 0.
CGBA[31:29] Bits 6–4	**MSB for Chip-Select C**—The upper most significant bits for chip-select group C base address. The value will be ignored if UGEN is disabled.	Enter value for bits 31–29 of chip-select register C.
Reserved Bit 3	Reserved	This bit is reserved and should be set to 0.
DGBA[31:29] Bits 2–0	**MSB for Chip-Select D**—The upper most significant bits for chip-select group D base address. The value will be ignored if UGEN is disabled.	Enter value for bits 31–29 of chip-select register D.

6.3.3 Chip-Select Registers

There are four 16-bit chip-select (CSA, CSB, CSC, and CSD) 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 DTACK signal. The settings for the registers are described in Table 6-7 through Table 6-10 on page 6-14.

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	w
RESET	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0
								0x00B0								

Table 6-7. 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 5.2.1, "System Control Register," on page 5-2 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 LWE/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 LWE/UWE signals go active at the same clock edge. 1 = The chip-select signal goes low 1 clock before LWE/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 DTACK signal is returned for this chip-select. **Note:** When using the external DTACK signal, you must configure the BUSW/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 DTACK. When using the external DTACK signal, you must select DTACK function in Port G. WS0 is the DWS0, CWS0, BWS0, or AWS0 bit in the CSCTRL1 register.

Table 6-7. Chip-Select Register A Description (Continued)

Name	Description	Setting
SIZ Bits 3–1	**Chip-Select Size**—This field determines the memory range of the chip-select. For CSAx and CSBx, the chip-select size is between 128K and 16 Mbyte. For CSCx and CSDx, the chip-select size is between 32K and 16 Mbyte.	000 = 128K (32K or 8 Mbyte* for CSCx and CSDx). 001 = 256K (64K or 16 Mbyte* for CSCx and CSDx). 010 = 512K (128K for CSCx and CSDx). 011 = 1 Mbyte (256K for CSCx and CSDx). 100 = 2 Mbyte (512K for CSCx and CSDx). 101 = 4 Mbyte (1 Mbyte for CSCx and CSDx). 110 = 8 Mbyte (2 Mbyte for CSCx and CSDx). 111 = 16 Mbyte (4 Mbyte for CSCx and CSDx). *** Note:** Large DRAM size selection requires the DSIZ3 bit in the chip-select control register to be set.
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	WS3–1			SIZ			EN
TYPE	rw	rw	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
								0x0000								

Table 6-8. 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 5.2.1, “System Control Register,” on page 5-2 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 5.2.1, “System Control Register,” on page 5-2 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 6-8. Chip-Select Register B Description (Continued)

Name	Description	Setting
WS3–1 Bits 6–4	**Wait State**—This field determines the number of wait states added before an internal DTACK signal is returned for this chip-select. **Note:** When using the external DTACK signal, you must configure the BUSW/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 DTACK. When using the external DTACK signal, you must select DTACK function in Port G. WS0 is the 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. For CSAx and CSBx, the chip-select size is between 128K and 16 Mbyte. For CSCx and CSDx, the chip-select size is between 32K and 16 Mbyte.	000 = 128K (32K or 8 Mbyte* for CSCx and CSDx). 001 = 256K (64K or 16 Mbyte* for CSCx and CSDx). 010 = 512K (128K for CSCx and CSDx). 011 = 1 Mbyte (256K for CSCx and CSDx). 100 = 2 Mbyte (512K for CSCx and CSDx). 101 = 4 Mbyte (1 Mbyte for CSCx and CSDx). 110 = 8 Mbyte (2 Mbyte for CSCx and CSDx). 111 = 16 Mbyte (4 Mbyte for CSCx and CSDx). *** Note:** Large DRAM size selection requires the DSIZ3 bit in the chip-select control register to be set.
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	w
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
								0x0000								

Table 6-9. 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 5.2.1, “System Control Register,” on page 5-2 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 5.2.1, “System Control Register,” on page 5-2 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 $\overline{LWE}$/$\overline{UWE}$ 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.

Table 6-9. 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 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. For $\overline{\text{CSAx}}$ and $\overline{\text{CSBx}}$, the chip-select size is between 128K and 16 Mbyte. For $\overline{\text{CSCx}}$ and $\overline{\text{CSDx}}$, the chip-select size is between 32K and 16 Mbyte.	000 = 128K (32K or 8 Mbyte* for $\overline{\text{CSCx}}$ and $\overline{\text{CSDx}}$). 001 = 256K (64K or 16 Mbyte* for $\overline{\text{CSCx}}$ and $\overline{\text{CSDx}}$). 010 = 512K (128K for $\overline{\text{CSCx}}$ and $\overline{\text{CSDx}}$). 011 = 1 Mbyte (256K for $\overline{\text{CSCx}}$ and $\overline{\text{CSDx}}$). 100 = 2 Mbyte (512K for $\overline{\text{CSCx}}$ and $\overline{\text{CSDx}}$). 101 = 4 Mbyte (1 Mbyte for $\overline{\text{CSCx}}$ and $\overline{\text{CSDx}}$). 110 = 8 Mbyte (2 Mbyte for $\overline{\text{CSCx}}$ and $\overline{\text{CSDx}}$). 111 = 16 Mbyte (4 Mbyte for $\overline{\text{CSCx}}$ and $\overline{\text{CSDx}}$). *** Note:** Large DRAM size selection requires the DSIZ3 bit in the chip-select control register to be set.
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		COMB	DRAM	FLASH	BSW	WS3–1			SIZ			EN
TYPE	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w
RESET	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	0x0200															

Table 6-10. 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 5.2.1, “System Control Register,” on page 5-2 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 5.2.1, “System Control Register,” on page 5-2 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.
COMB Bit 10	**Combining**—This bit controls combining $\overline{RAS0}$ and $\overline{RAS1}$ memory space to generate $\overline{RAS0}$. When this bit is set to 1, $\overline{RAS1}$ can be used as a general-purpose I/O signal.	0 = $\overline{RAS0}$ to $\overline{RAS0}$ memory space. 1 = $\overline{RAS0}$ covers both $\overline{RAS0}$ and $\overline{RAS1}$ memory space B.
DRAM Bit 9	**DRAM Selection**—This bit is used to enable $\overline{RAS}$ and $\overline{CAS}$ signals. Configuring the CSC register as a non-DRAM memory type requires clearing the DRAM bit of the CSD register. **Note:** The DRAM bit overrides the flash bit.	0 = Select $\overline{CSC}$[1:0] and $\overline{CSD}$[1:0]. 1 = Select $\overline{CAS}$ and $\overline{RAS}$.

Table 6-10. Chip-Select Register D Description (Continued)

Name	Description	Setting
FLASH Bit 8	**Flash Memory Support**—When enabled, this bit provides support for flash memory by forcing the LWE/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 LWE/UWE signals go active at the same clock edge. 1 = The chip-select signal goes low 1 clock before LWE/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 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 DTACK. When using the external DTACK signal, you must select DTACK function in Port G. WS0 is the 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. For CSAx and CSBx, the chip-select size is between 128K and 16 Mbyte. For CSCx and CSDx, the chip-select size is between 32K and 16 Mbyte.	000 = 128K (32K or 8 Mbyte* for CSCx and CSDx). 001 = 256K (64K or 16 Mbyte* for CSCx and CSDx). 010 = 512K (128K for CSCx and CSDx). 011 = 1 Mbyte (256K for CSCx and CSDx). 100 = 2 Mbyte (512K for CSCx and CSDx). 101 = 4 Mbyte (1 Mbyte for CSCx and CSDx). 110 = 8 Mbyte (2 Mbyte for CSCx and CSDx). 111 = 16 Mbyte (4 Mbyte for CSCx and CSDx). *** Note:** Large DRAM size selection requires the DSIZ3 bit in the chip-select control register to be set.
EN Bit 0	**Chip-Select Enable**—This write-only bit enables each chip-select.	0 = Disabled. 1 = Enabled.

6.3.4 Emulation Chip-Select Register

In addition to the eight general-purpose chip-select signals, the MC68VZ328 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 ($\overline{\text{DTACK}}$) 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 6-11. 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 $\overline{\text{DTACK}}$ 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 EWS0 bit in the CSCTRL1 register.
Reserved Bits 3–0	Reserved	These bits are reserved and should be set to 0.

6.3.5 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 6-12 on page 6-17.

CSCTRL1 Chip-Select Control Register 1 0x(FF)FFF10A

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
		EUP EN	SR 16	EW S0	DW S0	CW S0	BW S0	AW S0		DSI Z3		DUP S2		CUP S2		BUP S2
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									

Table 6-12. Chip-Select Control Register 1 Description

Name	Description	Setting
Reserved Bit 15	Reserved	This bit is reserved and should be set to 0.
EUPEN Bit 14	**Extra UPSIZ Bit Enable**—This bit enables the BUPS2, CUPS2, and DUPS2 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.
EWSO Bit 12	**Emulation Chip-Select Wait State Bit 0**—This bit is the lowest significant bit of the EMU wait state register.	Refer to Table 6-11 on page 6-16 on the emulation chip-select register for the wait state setting.
DWSO Bit 11	**CSD Wait State Bit 0**—This bit is the lowest significant bit of the CSD wait state register.	Refer to Table 6-10 on page 6-14 on the chip-select register D for the wait state setting.
CWSO Bit 10	**CSC Wait State Bit 0**—This bit is the lowest significant bit of the CSC wait state register.	Refer to Table 6-9 on page 6-12 on the chip-select register C for the wait state setting.
BWSO Bit 9	**CSB Wait State Bit 0**—This bit is the lowest significant bit of the CSB wait state register.	Refer to Table 6-8 on page 6-10 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 6-7 on page 6-8 on the chip-select register A for the wait state setting.
Reserved Bit 7	Reserved	This bit is reserved and should be set to 0.
DSIZ3 Bit 6	**Size Bit 3 for DRAM Chip-Select Addressing Space**—When set, this bit extends the DRAM size.	If SIZ[2:0] = 000, the $\overline{CSD0}$ and $\overline{CSD1}$ spaces are each 8 Mbyte. For 001, each space is 16 Mbyte. Only valid when the DRAM bit of the CSD register is set.
Reserved Bit 5	Reserved	This bit is reserved and should be set to 0.
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 6-1 on page 6-18.

Table 6-12. Chip-Select Control Register 1 Description (Continued)

Name	Description	Setting
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 6-1.
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 6-1.

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

Example 6-1. 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:

$4\text{ Mbyte} / 2^{(7-3)} = 256\text{K}$

6.3.6 Chip-Select Control Register 2

This register controls early cycle detection for both static and dynamic types 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	ECDS	ECDT	EASP			EASDLY[1:0]									
TYPE	rw	rw	rw	rw			rw	rw								
RESET	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
							0x1000									

Table 6-13. 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.

Table 6-13. Chip-Select Control Register 2 Description (Continued)

Name	Description	Setting
ECDS Bit 14	**Early Cycle Detection for Static Memory**—This bit advances the chip-select signals for SRAM, ROM, or flash memory. It allows more setup time for slow memory without adding CPU wait states.	0 = Disabled. 1 = Enabled.
ECDT Bit 13	**Early Cycle Detection Type**—When the master enable for early cycle detection is on (that is, ECDD = 1), this bit selects what signal from the CPU is used to trigger the bus cycle.	0 = Use the early ASB from the CPU as the triggering signal for early cycle detection. 1 = Use the TSCAE from the CPU as the triggering signal for early cycle detection.
EASP Bit 12	**Early ASB Delay Processing for Static Memory Early Cycle Detection**—To prevent the early ASB signal from the CPU from being asserted before a valid address is present from the CPU, the early ASB can be programmed so it is delayed before going to the chip-select generator. This bit must be programmed appropriately when early ASB is chosen as the early cycle detection signal.	0 = Use selectable delay chain as the delay processing method. 1 = Use negative CPU edge synchronization as the delay processing method (default setting).
Reserved Bits 11–10	Reserved	These bits are reserved and should be set to 0.
EASDLY[1:0] Bits 9–8	**Early ASB Delay Value**—When delay chain is chosen as the delay processing method for early ASB (that is, the EASP bit is clear), these bits select the level of the delay element for the early ASB to get through.	00 = No delay. 01 = 1 level. 10 = 2 levels. 11 = 3 levels.
Reserved Bits 7–0	Reserved	These bits are reserved and should be set to 0.

6.3.7 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										
TYPE	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 6-14. 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.
LCWS Bit 13	**Wait State Trim for LCD-SRAM Access**—When this bit is set, one additional wait state is added to the LCD-SRAM access cycle. For example, if the wait state is set to zero, all CPU accesses require 4 cycles to complete, the chip-select signal to SRAM lasts 2.5 CPU clock cycles, and 2 cycles are used for LCD access. When LCWS is enabled, the LCD access is delayed; the 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.
Reserved Bits 9–0	Reserved	These bits are reserved and should be set to 0.

Example 6-2 on page 6-21 demonstrates how to initialize the chip-select with a particular memory configuration.

Example 6-2. Programming Example

```
************************************************
*   Chip-Select registers
************************************************
REGSBASE     equu    0xFFFFF000internal registers base address
BASEA      equ   REGSBASE+0x100   group A base register
BASEB      equ   REGSBASE+0x102   group B base register
BASEC      equ   REGSBASE+0x104   group C base register
BASED      equ   REGSBASE+0x106   group D base register
CSA        equ   REGSBASE+0x110   group A chip-select register
CSB        equ   REGSBASE+0x112   group B chip-select register
CSC        equ   REGSBASE+0x114   group C chip-select register
CSD        equ   REGSBASE+0x116   group D chip-select register
************************************************
*   PORT control registers
************************************************
PORTBASE     equ    REGSBASE+0x400 port B registers base address
PBDir        equ    PORTBASE+0x08 port B direction register
PBData       equ    PORTBASE+0x09 port B data register
PBPU         equ    PORTBASE+0x0A port B pullup enable register
PBSel        equ    PORTBASE+0x0B port B select register

************************************************
*   Initialization
************************************************
START        move.b   #0x00,PBSel      disable PortB, select chip-selects

             move.w   #0x0000,BASEA   set base address 0x0000000
             move.w   #0x8081,CSA     read-only,16-bit,0 wait state,128K

             move.w   #0x2000,BASEB   set base address 0x4000000
             move.w   #0x0093,CSB     read/write,16-bit,1 wait state,256K

             move.w   #0x2040,BASEC   set base addrs 0x4080000
             move.w   #0x0191,CSC     read/write,flash,16-bit,1 ws,32K

             move.w   #0x0000,CSD     config CSC,CSD as non-DRAM memory type


* The preceding initialization will configure the CSA and CSB chip-selects as
* follows :
*
*   CSA0 0x0000000-0x001ffff,read-only, 16-bit,0 wait state,128K
*   CSA1 0x0020000-0x003ffff,read-only, 16-bit,0 wait state,128K
*   CSB0 0x4000000-0x403ffff,read/write,16-bit,1 wait state,256K
*   CSB1 0x4040000-0x407ffff,read/write,16-bit,1 wait state,256K
*   CSC0 0x4080000-0x4087fff,read/write,flash,16-bit,1 wait state, 32K
*   CSC1 0x4088000-0x408ffff,read/write,flash,16-bit,1 wait state, 32K
*   CSD0 disabled
*   CSD1 disabled
```

Chapter 7
DRAM Controller

This chapter describes the DRAM controller for the MC68VZ328. The operation of the DRAM controller is closely linked to the chip-select logic. Please refer to Chapter 6, “Chip-Select Logic,” for more details.

7.1 Introduction to the DRAM Controller

The DRAM controller provides a glueless interface for either 8-bit or 16-bit DRAM. It supports EDO RAM, Fast Page Mode, and synchronous DRAM. The DRAM controller provides Row Address Strobe ($\overline{\text{RAS}}$) and Column Address Strobe ($\overline{\text{CAS}}$) signals for up to a maximum of two banks of DRAM. In addition to controlling DRAM, the DRAM controller provides support for LCD controller burst accesses.

The DRAM controller has the following features:

- 68000 CPU zero wait-state operation support
- $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh cycles and self-refresh mode DRAM support
- 8- and 16-bit port DRAM support
- Fast Page Mode and EDO RAM modes or synchronous burst for LCD DMA access cycles
- Programmable refresh rate
- Support for a maximum of two banks of DRAM
- Programmable row and column address size with symmetrical or asymmetrical addressing
- Support for up to 16 Mbyte × 16 or 32 Mbyte × 8 DRAM or SDRAM

A block diagram of the DRAM controller appears in Figure 7-1 on page 7-2.

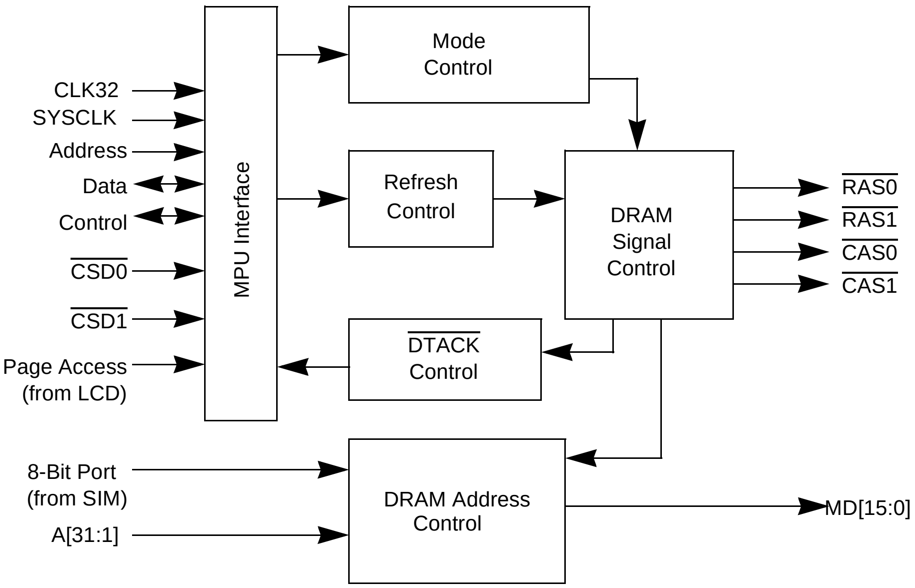


Figure 7-1. DRAM Controller Block Diagram

7.2 DRAM Controller Operation

This section describes the DRAM controller's operation.

7.2.1 Address Multiplexing

The address multiplexer can support a wide variety of memory devices in either 8- or 16-bit mode. The upper internal address lines from the CPU or LCD controller provide the row address, and the lower internal address lines are used as the column address. This scheme enables the use of Fast Page Mode or EDO RAM mode read accesses to the DRAM during LCD DMA cycles. The DRAM multiplexer also supports different row and column configurations, depending on the arrangement of the DRAM rows and columns and the data port size (8 or 16 bit) of the DRAM.

For 4 Mbyte (512K × 8) DRAM, there are usually only 10 row addresses and 9 column addresses. For this configuration in 8-bit mode, the internal address bus PA[8:0] is used for column addresses, and PA[18:9] is used for row addresses. Similarly, if we use 16-bit DRAM with the same number of row and column addresses, the column addresses require PA[9:1], and PA[19:10] is used for the row addresses.

The address multiplexing options are provided in Table 7-1 on page 7-4. The MC68VZ328's DRAM controller uses PA[8:1] as the column addresses for MD[7:0] and then allows software to select either PA0 or PA9 for column address MD8. Similar address selection options are provided for MD9 and MD10 column addresses, the MD0 row address, and the row addresses MD8 through MD12 .

The MD[12:0] signals share the same address pins that output as nonmultiplexed addresses A[13:1] for non-DRAM external accesses. Since the internal addresses (PA[13:1]) are present as the column address selection from the DRAM address multiplexer, these addresses may be used as the nonmultiplexed addresses A[13:1] for non-DRAM external accesses. This simplifies the overall multiplexing scheme for the MC68VZ328.

NOTE:

The A0 signal is not used as a DRAM address pin connection.

Table 7-1 on page 7-4 contains the address multiplexing options for the VZ pins listed. All the options are programmed in the DRAM memory configuration (DRAMMC) register except as noted in the table. The row labeled "Column Address Options" is used for Fast Page Mode and EDO RAM and is enabled when the SDEN bit (bit 15) in the SDRAM control register (0xFFFFFC04) is 0. The row labeled "Column Address Options Specific for SDRAM" is used for SDRAM and is enabled when the SDEN bit in the SDRAM control register is 1.

Table 7-1. DRAM Address Multiplexing Options

	A1/MD0	A2/MD1	A3/MD2	A4/MD3	A5/MD4	A6/MD5	A7/MD6	A8/MD7
Row Address Options	PA23 PA22 PA11	PA12	PA13	PA14	PA15	PA16	PA17	PA18
Column Address Options	PA1	PA2	PA3	PA4	PA5	PA6	PA7	PA8
Column Address Options for SDRAM	PA1[1] PA0	PA2	PA3	PA4	PA5	PA6	PA7	PA8
MD Address	MD0	MD1	MD2	MD3	MD4	MD5	MD6	MD7

	A9/MD8	A10/MD9	A11/MD10	A12/MD11	A13/MD12	A14/MD13	A15/MD14	A16/MD15
Row Address Options	PA10 PA20	PA9 PA19	PA19 PA21	PA20 PA22	PA10 PA21 PA23	PA22	PA23	PA24
Column Address Options	PA0 (PA1) PA9	PA0 PA10	PA0 PA11	PA12	PA13	PA22	PA23	PA24
Column Address Options for SDRAM	PA1[2] PA9	PA1	0	PA20[3] PA22	PA10[4] PA21 PA23	PA22	PA23	PA24
MD Address	MD8	MD9	MD10	MD11	MD12	MD13	MD14	MD15

1.Pin A1/MD0 has column address options of PA0 and PA1 for SDRAM. The SCOL bit (bit 6) of the SDRAM control register (0xFFFFFC04) determines the selection. When SCOL = 0, PA1 is selected. When SCOL = 1, PA0 is selected.

2.Pin A9/MD8 has column address options of PA1 and PA9 for SDRAM. The COL8 bit (bit 5) of the DRAM memory configuration register (0xFFFFFC00) determines the selection. When COL8 = 0, PA9 is selected. When COL8 = 1, PA1 is selected.

3.Pin A12/MD11 has column address options of PA20 and PA22 for SDRAM. The ROW11 bit (bit 11) of the DRAM memory configuration register (0xFFFFFC00) determines the selection. When ROW11 = 0, PA20 is selected. When ROW11 = 1, PA22 is selected.

4.Pin A13/MD12 has column address options of PA10, PA21, and PA23 for SDRAM. The ROW12 field (bits 15–14) of the DRAM memory configuration register (0xFFFFFC00) determines the selection. When ROW12 = 00, PA10 is selected. When ROW12 = 01, PA21 is selected. When ROW12 = 10, PA23 is selected.

Table 7-2 through Table 7-5 on page 7-6 provide recommendations for MC68VZ328–to–SDRAM connections and for selecting multiplexing options for different types of SDRAM.

Table 7-2. 16 Mbit SDRAM—256 (16-Bit) and 512 (8-Bit) Page Size

SDRAM Pins	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	BS
VZ Pins	A1/MD0	A2/MD1	A3/MD2	A4/MD3	A5/MD4	A6/MD5	A7/MD6	A8/MD7	A9/MD8	A10/MD9	A11/MD10	A12/MD11
Row Address Options	PA11	PA12	PA13	PA14	PA15	PA16	PA17	PA18	PA10	PA9	PA19	PA20
Column Address Options (16-Bit)	PA1	PA2	PA3	PA4	PA5	PA6	PA7	PA8	X	X	0	PA20
Column Address Options (8-Bit)	PA0	PA2	PA3	PA4	PA5	PA6	PA7	PA8	PA1	X	0	PA20

Note: X = “don’t care”

Table 7-3. 64 Mbit SDRAM—256 (16-Bit) and 512 (8-Bit) Page Size

SDRAM Pins	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	BS0	BS1
VZ Pins	A1/MD0	A2/MD1	A3/MD2	A4/MD3	A5/MD4	A6/MD5	A7/MD6	A8/MD7	A9/MD8	A10/MD9	A11/MD10	A12/MD11	A13/MD12	A14/MD13
Row Address Options	PA11	PA12	PA13	PA14	PA15	PA16	PA17	PA18	PA10	PA9	PA19	PA20	PA21	PA22
Column Address Options (16-Bit)	PA1	PA2	PA3	PA4	PA5	PA6	PA7	PA8	X	X	0	X	PA21	PA22
Column Address Options (8-Bit)	PA0	PA2	PA3	PA4	PA5	PA6	PA7	PA8	PA1	X	0	X	PA21	PA22

Note: X = “don’t care”

Table 7-4. 128 Mbit SDRAM—512 (16-Bit) and 1024 (8-Bit) Page Size

SDRAM Pins	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	BS0	A11	BS1
VZ Pins	A1/ MD 0	A2/ MD 1	A3/ MD 2	A4/ MD 3	A5/ MD 4	A6/ MD 5	A7/ MD 6	A8/ MD 7	A9/ MD 8	A10 /MD 9	A11 /MD 10	A12/ MD 11	A13 /MD 12	A15/ MD 14
Row Address Options	PA 11	PA 12	PA 13	PA 14	PA 15	PA 16	PA 17	PA 18	PA 20	PA 19	PA 21	PA22	PA 10	PA23
Column Address Options (16-Bit)	PA 1	PA 2	PA 3	PA 4	PA 5	PA 6	PA 7	PA 8	PA 9	X	0	PA22	X	PA23
Column Address Options (8-Bit)	PA 0	PA 2	PA 3	PA 4	PA 5	PA 6	PA 7	PA 8	PA 9	PA 1	0	PA22	X	PA23

Note: X = "don't care"

Table 7-5. 256 Mbit SDRAM—512 (16-Bit) and 1024 (8-Bit) Page Size

SDRAM Pins	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	BS 0	BS 1
VZ Pins	A1/ MD 0	A2/ MD 1	A3/ MD 2	A4/ MD 3	A5/ MD 4	A6/ MD 5	A7/ MD 6	A8/ MD 7	A9/ MD 8	A10 /MD 9	A11 /MD 10	A12 /MD 11	A13 /MD 12	A15 /MD 14	A16 /MD 15
Row Address Options	PA 11	PA 12	PA 13	PA 14	PA 15	PA 16	PA 17	PA 18	PA 20	PA 19	PA 21	PA 22	PA 10	PA 23	PA 24
Column Address Options (16-Bit)	PA 1	PA 2	PA 3	PA 4	PA 5	PA 6	PA 7	PA 8	PA 9	X	0	X	X	PA 23	PA 24
Column Address Options (8-Bit)	PA 0	PA 2	PA 3	PA 4	PA 5	PA 6	PA 7	PA 8	PA 9	PA 1	0	X	X	PA 23	PA 24

Note: X = "don't care"

7.2.2 $\overline{\text{DTACK}}$ Generation

In a 16 MHz system frequency, 60 ns DRAM can support a zero wait state (4 clocks per access) for CPU bus cycles. Therefore, $\overline{\text{DTACK}}$ is only delayed for refresh operations that occur before a read/write access cycle. The value of N clocks (N is the number of system clock cycles required for refresh) will be inserted into a read or write cycle when the CPU cycle collides with a refresh cycle. Refresh, in this case, has a higher priority.

NOTE:

> The value of N can be 1–4 clocks, depending on the collision overlap of the refresh cycle and CPU bus cycle.

7.2.3 Refresh Control

During normal operation, the MC68VZ328 DRAM cycles are distributed evenly over the refresh period. The DRAM refresh rate requirement may vary between different DRAM chips. Users can program the REF field in the DRAM configuration register (DRAMMC) to select the required refresh frequency.

The following examples demonstrate refresh values using two different settings and clock sources:

- When CLK32 = 32.768 kHz:
  - CLK = 0
  - DRAMMC register value (REF) = 0
  - refresh period = 15.2 µs
- If SYSCLK = 16.58 MHz:
  - CLK = 1
  - DRAMMC register value (REF) = 7
  - refresh period = 15.44 µs

7.2.4 LCD Interface

Figure 7-2 illustrates the LCD controller and DRAM controller interface. The DRAM controller supports page bursting accesses. When the PAGE_ACCESS signal is active and $\overline{\text{CSD}}$[1:0] is active, Fast Page Mode or EDO RAM mode will be initiated.

In Fast Page Mode mode, the first access will always be 4 clocks. Additional clocks may be added to the access cycle for the second and subsequent access cycles using the BC0 and BC1 bits of the DRAMC register. One, two, three, and four additional clocks are supported by the DRAM controller. The notation for the additional clock cycles is to display the first three numbers, separated by hyphens, followed by an ellipsis and the final number: first clock-second clock-third clock-...-last clock. For example, the notation 4-2-2-...-2 represents 4 clocks for the first transfer and 2 clocks for the second and subsequent transfers. The first access is always 4 clocks.

Single clocks and transfers are only supported in EDO RAM mode, allowing the fastest LCD DMA transfers. However, in EDO RAM mode, the BC0 and BC1 bits are ignored by the DRAM controller. For additional information about operation using an LCD display, see Chapter 8, “LCD Controller.”

When an LCD controller cycle and a refresh request collide before the LCD controller cycle starts, refresh will go first, and N more clocks will be added to the first access (N is the number of system clock cycles required for refresh). Therefore, in EDO RAM mode, for a 4-1-1-...-1 cycle, the access will become (4+N)-1-1-...-1.

When consecutive LCD controller burst accesses cross a memory page boundary, the DRAM controller will hold the LCD controller that is negating the internal $\overline{\text{DTACK}}$ signal to change the row address and wait for a precharge time. When a refresh request occurs in the middle of an LCD controller cycle transfer, refresh will be deferred until the end of the LCD controller cycle. Since the LCD controller cycle only lasts for 8 cycles, the deferred refresh cycle will not overlap with the next refresh request.

The $\overline{\text{DTACK}}$ signal is used to hold the LCD controller after the address changes on each word of an LCD transfer. If $\overline{\text{DTACK}}$ is asserted, the LCD controller will assume a fixed wait-state transfer per the setup within the LCD controller. The LCD controller will hold as long as $\overline{\text{DTACK}}$ is not asserted.

The PAGE_ACCESS signal from the LCD controller indicates to the DRAM controller and system integration module that an LCD DMA burst transfer is about to begin. The associated chip-select signal will hold active throughout the LCD controller’s access cycle. In this mode, the DRAM controller supports page accesses.

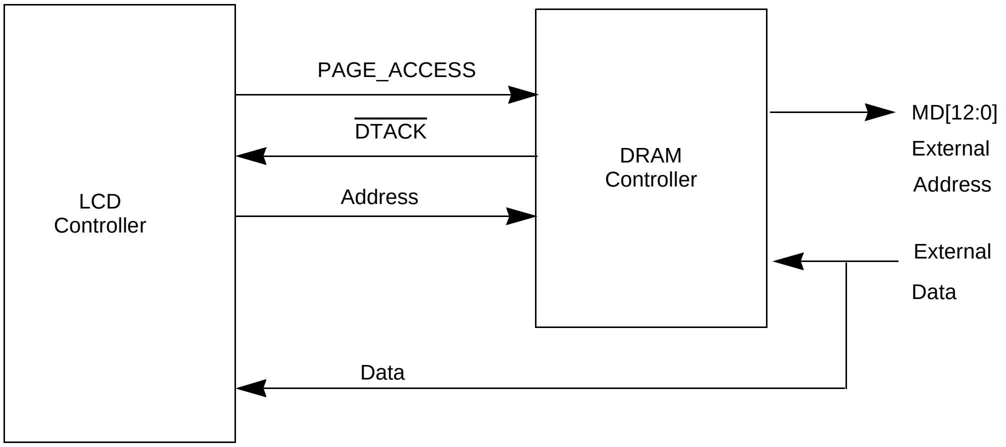


Figure 7-2. LCD Controller and DRAM Controller Interface

7.2.5 8-Bit Mode

From the system integration module (SIM), 8-bit operation on the fly can be selected using the signal 8-bit port. If one of the $\overline{\text{CSDx}}$ signals is programmed as 8-bit mode, the 8-bit mode signal will be active at the same time that $\overline{\text{CSDx}}$ is active. In 8-bit mode, the DRAM address multiplexer will use PA0 instead of PA1 as the least significant multiplexed address, and the remainder of the multiplexed address lines will be adjusted to fit the 8-bit operation of the selected DRAM device. $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and refresh signal functions will remain the same. Depending on the DRAM type used, the system software may need to adjust the address multiplexer options in the DRAMMC register.

7.2.6 Low-Power Standby Mode

If DRAM that supports self-refresh mode is being used, the RM bit in the DRAMC register can be programmed to self-refresh mode before entering sleep mode. The DRAM controller will generate one $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ cycle, negate $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ for the required precharge time, then assert $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$, and continue to assert them until the mode is changed in the RM bit. DRAMs that support self-refresh mode will enter self-refresh typically 100 µs after $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ are held in the asserted state. After a wake up, one $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh cycle will occur, and then normal-mode operation will continue.

For DRAMs without self-refresh mode, ensure that the LPR bit in the DRAMC register is set for $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh mode to continue while the processor is shut down and all other modules are disabled.

7.2.7 Data Retention During Reset

DRAM needs to retain data during reset, whether it is an external reset or an internal watchdog reset. The DRAM controller itself has a special design to support this feature. Figure 7-3 illustrates the timing for data retention.

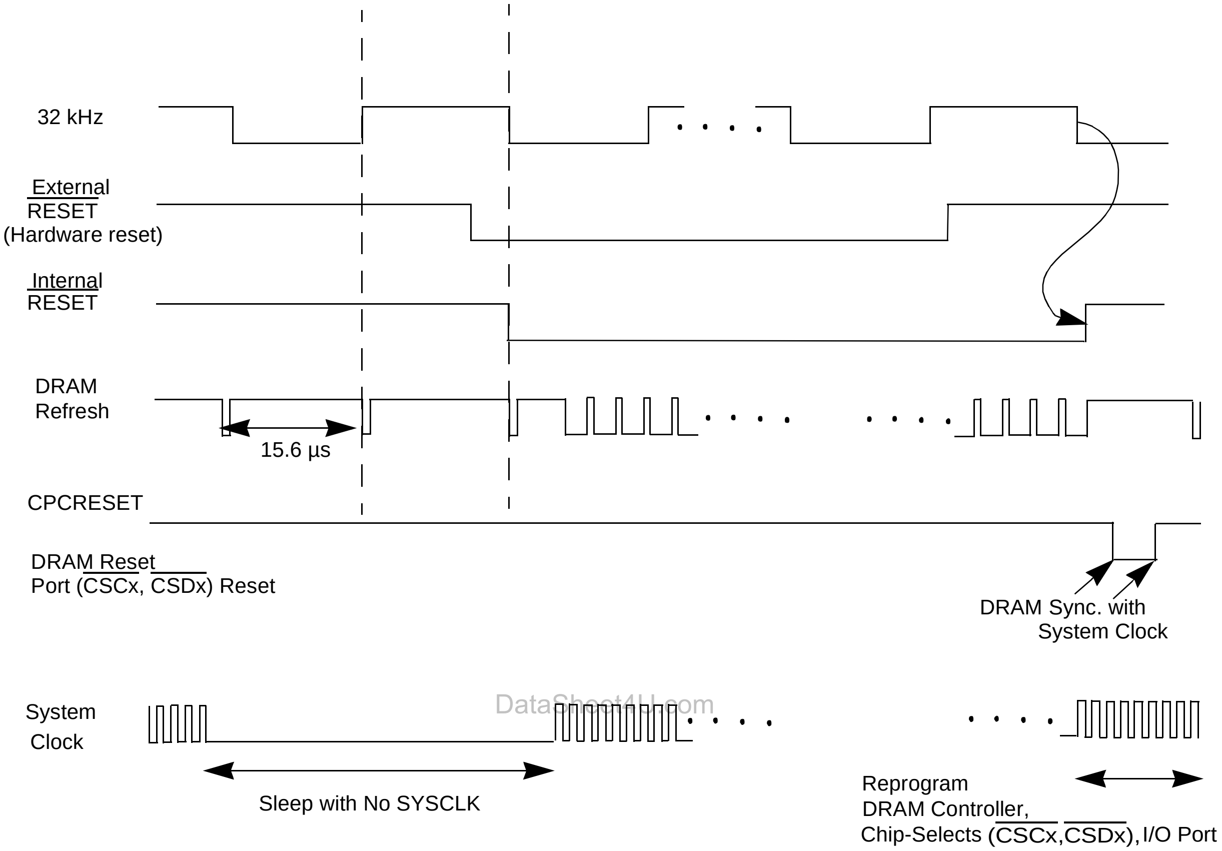

32 kHz
External RESET (Hardware reset)
Internal RESET
DRAM Refresh
15.6 µs
CPCRESET
DRAM Reset Port (CSCx, CSDx) Reset
DRAM Sync. with System Clock
System Clock
Sleep with No SYSCLK
Reprogram DRAM Controller, Chip-Selects (CSCx,CSDx), I/O Port

Figure 7-3. Data Retention for the Reset Cycle

7.2.8 Data Retention Sequence

Data is retained in the following sequence:

1. The external $\overline{\mathrm{RESET}}$ signal is sent to the MC68VZ328.
2. The internal $\overline{\mathrm{RESET}}$ signal is generated by synchronizing the external $\overline{\mathrm{RESET}}$ signal with the CLK32 signal.
3. When the internal $\overline{\mathrm{RESET}}$ is asserted, the DRAM controller will stop the current refresh operation and enter burst refresh mode, which is a consecutive $\overline{\mathrm{CAS}}$-before-$\overline{\mathrm{RAS}}$ refresh cycle.
4. The external $\overline{\mathrm{RESET}}$ signal continues asserting.
5. The external $\overline{\mathrm{RESET}}$ signal is negated.
6. The internal $\overline{\mathrm{RESET}}$ signal is negated.
7. The DRAM controller terminates the burst $\overline{\mathrm{CAS}}$-before-$\overline{\mathrm{RAS}}$ refresh cycle.
8. The internal CPCRESET signal is generated for 16 clocks to reset the DRAM controller and the $\overline{\mathrm{CSCx}}$ and $\overline{\mathrm{CSDx}}$ port signals.
9. The chip is now reset.
10. The core processor programs the DRAM controller and the port pins after this reset to resume DRAM controller operation.

NOTE:

> The initialization code should program or initialize the DRAM controller and the general-purpose I/O port signals within the DRAM’s specified refresh time.

7.3 Programming Model

This section describes the programming model for the DRAM controller.

7.3.1 DRAM Memory Configuration Register

The DRAM memory configuration register (DRAMMC) is used to set the DRAM refresh interval and configure the address multiplexer for the specific memory device being used. The bit position and values are shown in the following register display. The details about the register settings are described in Table 7-6.

DRAMMC DRAM Memory Configuration Register **0x(FF)FFFC00**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	ROW12		ROW0		ROW 11	ROW 10	ROW 9	ROW 8	COL 10	COL 9	COL 8	REF				
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	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. DRAM Memory Configuration Register Description

Name	Description	Setting
ROW12 Bits 15–14	**Row Address MD12**—This field selects the row address bit for multiplexed address MD12.	00 = PA10 01 = PA21 10 = PA23 11 = Not valid
ROW0 Bits 13–12	**Row Address MD0**—This field selects the row address bit for multiplexed address MD0.	00 = PA11 01 = PA22 10 = PA23 11 = Not valid
ROW11 Bit 11	**Row Address MD11**—This bit selects the row address bit for multiplexed address MD11.	0 = PA20 1 = PA22
ROW10 Bit 10	**Row Address MD10**—This bit selects the row address bit for multiplexed address MD10.	0 = PA19 1 = PA21
ROW9 Bit 9	**Row Address MD9**—This bit selects the row address bit for multiplexed address MD9.	0 = PA9 1 = PA19
ROW8 Bit 8	**Row Address MD8**—This bit selects the row address bit for multiplexed address MD8.	0 = PA10 1 = PA20
COL10 Bit 7	**Column Address MD10**—This bit selects the column address bit for multiplexed address MD10.	0 = PA11 1 = PA0
COL9 Bit 6	**Column Address MD9**—This bit selects the column address bit for multiplexed address MD9.	0 = PA10 1 = PA0
COL8 Bit 5	**Column Address MD8**—This bit selects the column address bit for multiplexed address MD8.	0 = PA9 1 = PA0

Table 7-6. DRAM Memory Configuration Register Description (Continued)

Name	Description	Setting
REF Bits 4–0	**Refresh Cycle**—This value determines the refresh rate for the DRAM controller. The refresh rate can be calculated using the equation shown in Example 7-1.	See description

The REF value is the time of 1 refresh cycle.

Example 7-1. Calculating REF Field Values for Refresh Times

When CLK = 0, 32 kHz (or 34.8 kHz) is used for refresh control.

- If REF = 0, the refresh rate = 2 × 32 kHz.
- If REF = 1, the refresh rate = 32 kHz.
- If REF = (2 to 15), the refresh rate = 32 kHz / (REF + 1).

When CLK = 1, the system clock is used for refresh control.

- The refresh rate = SYSCLK / (32 × (REF + 1)).

7.3.2 DRAM Control Register

The DRAM control (DRAMC) register is used to control the operation of the DRAM controller. The bit position and values are shown in the following register display. The details about the register settings are described in Table 7-7.

DRAMC DRAM Control Register **0x(FF)FFFC02**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	EN	RM	BC1–0		CLK	EDO	PGSZ				MSW	LSP	SLW	LPR	RST	DWE
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	1	0
								0x0000								

Table 7-7. DRAM Control Register Description

Name	Description	Setting
EN Bit 15	**Master DRAM Controller Enable**—This bit enables and disables the DRAM controller.	0 = Disable the DRAM controller. 1 = Enable the DRAM controller.
RM Bit 14	**Refresh Mode**—This bit sets the refresh mode.	0 = $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh mode. 1 = Self-refresh mode.
BC1–0 Bit 13–12	**Page Access Clock Cycle (Fast Page Mode)**—These bits determine the number of additional clocks for the second and subsequent accesses within a Fast Page Mode read cycle after the first data word.[1]	00 = 1 additional clock (2 clocks/transfer). 01 = 2 additional clocks (3 clocks/transfer). 10 = 3 additional clocks (4 clocks/transfer). 11 = 4 additional clocks (5 clocks/transfer).
CLK Bit 11	**Clock**—This bit selects the clock that is provided to the refresh timer.	0 = CLK32 (Period A) is selected. 1 = System clock (Period B) is selected.
EDO Bit 10	**Extended Data Out**—This bit selects the page access mode for LCD DMA DRAM accesses. This bit should only be set if the DRAM supports EDO RAM transfers. When the EDO bit is set, BC0 and BC1 do not affect the number of clocks for LCD DMA DRAM accesses. EDO RAM mode is the fastest LCD DMA transfer mode.	0 = Fast Page Mode mode is selected. 1 = EDO enables 1 clock for each LCD DMA data word transfer after the first word transfer. Bits BC1–0 are ignored.
PGSZ Bits 9–8	**Page Size**—This field determines the page size in the word for Fast Page Mode mode access.	00 = 256 01 = 512 10 = 1,024 11 = 2,048
Reserved Bits 7–6	Reserved	These bits are reserved and should be set to 0.
MSW Bit 5	**Slow Multiplexing**—Setting this bit adds a system clock for DRAM address multiplexing, which allows for a heavily loaded A/DMA bus. Setting this bit causes an additional wait state for all core accesses and the first LCD DMA word access.	0 = Normal address multiplexing. 1 = Slower address multiplexing.

Table 7-7. DRAM Control Register Description (Continued)

Name	Description	Setting
LSP Bit 4	**Light Sleep**—Setting this bit enables the core or LCD controller to access the DRAM when the RM bit is set (DRAM is in self-refresh mode). Self-refresh mode is temporarily interrupted for the DRAM access and automatically returns to self-refresh mode once the transfer is complete. Transfers in this mode are much slower than normal. Therefore, it is best to clear the RM bit if the DRAM is to be awake for extended periods of time. If this bit is clear, DRAM accesses will not occur when RM is set, and attempts will cause the bus to time out.	0 = Self-refresh is interrupted only by clearing the RM bit. 1 = Self-refresh is temporarily interrupted by core or LCD controller accesses to DRAM.
SLW Bit 3	**Slow RAM**—Setting this bit extends the $\overline{\text{RAS}}$ precharge period for slower DRAM devices. This bit should be set if the $\overline{\text{RAS}}$ precharge time requirement for the device being used is greater than 60 ns (33 MHz system clock) or 120 ns (16.58 MHz system clock).	0 = Normal $\overline{\text{RAS}}$ precharge (2 system clocks). 1 = Extended $\overline{\text{RAS}}$ precharge for slower DRAM devices (4 system clocks).
LPR Bit 2	**Low-Power Refresh Enable**—This bit is used to control the refresh during low-power modes.	0 = Disable low-power refresh mode. 1 = Enable low-power refresh mode.
RST Bit 1	**Reset Burst Refresh Enable**—This bit controls the refresh type during $\overline{\text{RESET}}$ assertion.	0 = Normal distributed refresh operation during DRAM reset function. 1 = Continuous burst refresh operation during DRAM reset function.
DWE Bit 0	**DRAM Write-Enable**—This bit is used to enable the $\overline{\text{DWE}}$ signal, which can be employed when a DRAM is being used that needs an independent write-enable signal, rather than sharing one with the $\overline{\text{UWE}}$ signal.	0 = Disable $\overline{\text{DWE}}$. 1 = Enable $\overline{\text{DWE}}$.

[1] The first Fast Page Mode access will always be 4 clocks. When an LCD controller cycle and a refresh request collide before the LCD controller cycle starts, refresh will go first, and N more clocks will be added to the first access (N is the number of system clock cycles required for refresh).

7.3.3 SDRAM Control Register

This register controls operation when SDRAM is being used. The bit position and values are shown in the following register display. The details about the register settings are described in Table 7-8.

SDCTRL SDRAM Control Register **0x(FF)FFFC04**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	SDEN	CPM		RE	IP	MR				SCOL	BNKADDH		BNKADDL		CL	RACL
TYPE	rw	rw		rw	rw	rw				rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0

0x003C

Table 7-8. SDRAM Control Register Description

Name	Description	Setting
SDEN Bit 15	**SDRAM Enable**—When this bit is set, together with the DRAM enable bit (bit 9 of the CSD register) being set and the EDO bit (DRAMC register bit 10) being cleared, the SDRAM operation is enabled.	0 = SDRAM disable. 1 = SDRAM enable (see description for other bits that must be set).
CPM Bit 14	**Continuous Page Mode**—This bit enables the DRAM to operate in continuous page mode. DRAM will only be precharged during a page-miss condition.	0 = SDRAM not in continuous page mode. 1 = SDRAM in continuous page mode.
Reserved Bit 13	Reserved	This bit is reserved and must be set to 0.
RE Bit 12	**Refresh Enable**—This bit enables the refresh cycle for SDRAM.	0 = SDRAM Refresh cycle not enabled. 1 = SDRAM refresh cycle enabled.
IP Bit 11	**Initiate All Bank Precharge Command**—Setting this bit triggers the precharge command for all banks of SDRAM.	0 = IP command to SDRAM disabled. 1 = IP command to SDRAM enabled.
MR Bit 10	**Initiate Mode Register Set Command**—Setting this bit triggers the load mode register command to SDRAM.	0 = MR command to SDRAM disabled. 1 = MR command to SDRAM enabled.
Reserved Bits 9–7	Reserved	These bits are reserved and should be set to 0.
SCOL Bit 6	**SDRAM Column Option**—This bit selects the SDRAM column address MD0.	0 = PA1 (normally for 16-bit SDRAM). 1 = PA0 (normally for 8-bit SDRAM).
BNKADDH Bits 5–4	**SDRAM High Order Bank Address Line Selection**—A 2-bit bank register selection address is generated by selecting the appropriate CPU address line. This register bit allows selection of the high order bit.	00 = PA20. 01 = PA22. 10 = PA24. 11 = Force this bank address line to 0. See Table 7-9 on page 7-17 for programming examples.

Table 7-8. SDRAM Control Register Description (Continued)

Name	Description	Setting
BNKADDL Bits 3–2	**SDRAM Low Order Bank Address Line Selection**—A 2-bit bank register selection address is generated by selecting the appropriate CPU address line. This register bit allows selection of the low order bit.	00 = PA19. 01 = PA21. 10 = PA23. 11 = Force this bank address line to 0. See Table 7-9 for programming examples.
CL Bit 1	**CAS Latency**—This bit selects the CAS latency for the SDRAM cycle. The bit must be programmed before the initialization sequence.	0 = CAS latency is 1 clock count. 1 = CAS latency is 2 clock counts.
RACL Bit 0	**Refresh to Active Command Latency**—This bit selects the latency for SDRAM from refresh to active cycle.	0 = 3 Clock counts. 1 = 6 Clock counts.

Table 7-9. SDRAM Bank Address Programming Examples

Application	BNKADDH	BNKADDL	Remarks
Make all SDRAM appear as one single bank	11	11	None
Two banks of SDRAM—for example, 16 Mbyte	00	11	Choose PA20 as bank selection address
Four banks of SDRAM—for example, 64 Mbyte	01	10	Choose PA22 and PA21 as bank selection address
Four banks of SDRAM—for example, 128 Mbyte	01	10	Choose PA22 and PA23 as bank selection address
Four banks of SDRAM—for example, 256 Mbyte	10	10	Choose PA24 and PA23 as bank selection address
Note: These bits are all set in EDO RAM or Fast Page Mode, allowing the use of only one page register.			

7.3.4 SDRAM Power-down Register

This register controls how the SDRAM and the MC68VZ328 operate during a power-down operation. The bit position and values are shown in the following register display. The details about the register settings are described in Table 7-10.

SDPWDN SDRAM Power-down Register **0x(FF)FFFC06**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	APEN	PDEN			PDTOUT[3:0]											
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 7-10. SDRAM Power-down Register Description

Name	Description	Settings
APEN Bit 15	**SDRAM Active Power-down Enable**—The bit is set to make the SDRAM Chip Enable signal go low immediately when the DRAM controller is not sending a command, writing data, or reading data with the SDRAM.	0 = APEN disabled. 1 = APEN enabled.
PDEN Bit 14	**SDRAM Precharged Power-down Enable**—The bit is set to make the SDRAM Chip Enable signal go low when the DRAM controller is not sending a command after the SDRAM is precharged for a certain time. The time depends on the value in PDTOUT[3:0].	0 = PDEN disabled. 1 = PDEN enabled.
Reserved Bits 13–12	Reserved	These bits are reserved and should be set to 0.
PDTOUT [3:0] Bits 11–8	**SDRAM Precharged Power-down Time Out**—The bit is set to make the SDRAM Chip Enable signal go low when a time out occurs when the PDEN bit is set. Each binary unit represents a maximum of 128 clocks. When in power-down mode, SDRAM can be woken by a CPU or LCD access.	See the description.
Reserved Bits 7–0	Reserved	These bits are reserved and should be set to 0.

Chapter 8
LCD Controller

This chapter describes the operation of the liquid crystal display (LCD) controller and supplies the programming information necessary to implement it in design projects. The LCD controller provides display data for external LCD drivers or for an LCD panel. The LCD controller fetches display data directly from system memory through periodic DMA transfer cycles. For this reason, an understanding of the DRAM controller is recommended. For more information, please refer to Chapter 7, “DRAM Controller.” The LCD controller uses very little bus bandwidth, giving the core sufficient processing time.

8.1 LCD Controller Features

The following list describes the features of the LCD controller.

- Both system and display memory that is shared, so that dedicated video memory is not required
- Standard panel interface for industry-standard LCD drivers
- Support for single (nonsplit) monochrome screen and color STN LCD panels through preprocessing of image data with software
- Fast fly-by-type, 16-bit-wide, burst DMA screen-refresh transfers from system memory
- Maximum display size of 640 × 512 pixels
- Panel interface for 8-, 4-, 2-, and 1-bit-wide LCD data bus
- 16 simultaneous grayscale levels from a palette of 16 density levels
- Hardware blinking cursor that is programmable to a maximum of 31 × 31 pixels
- Hardware panning (soft horizontal and vertical scrolling)
- 8-bit PWM for software contrast control
- New FRC algorithm that improves the flickering effect found in 4- and 16-grayscale LCD panels
- Support for self-refresh-type LCD panels

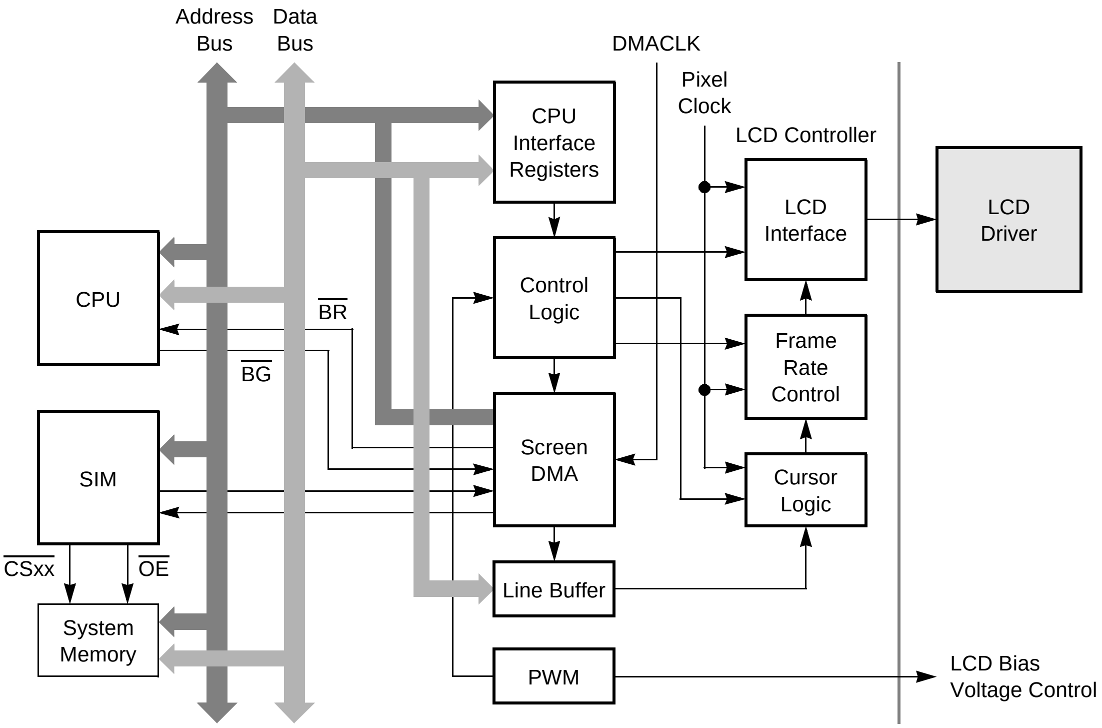


Figure 8-1. LCD Controller Block Diagram

8.2 LCD Controller Operation

The LCD controller consists of CPU interface registers, control logic, a screen DMA controller, a line buffer, cursor logic, frame rate control, and an LCD panel interface. Figure 8-1 illustrates how these blocks are organized.

The CPU interface registers provide control of different features of the LCD controller. Connected to the CPU bus, the control logic provides the internal control and counting signals for other blocks in the LCD controller. The DMA generates a bus request ($\overline{\text{BR}}$) signal to the core, and when the bus is granted, it performs a few memory bursts to fill up the line buffer. The number of DMA clock cycles in each burst is the programmable number of clocks per transfer, which makes it easier to support a system with memory with different speed grades.

The line buffer collects display data from system memory during DMA cycles and outputs it to the cursor logic block. The input is synchronized with the fast DMA clock, while the output is synchronized to the relatively slow LCD pixel clock. The cursor control logic, when enabled, is used to generate a block-shaped cursor on the display screen. The height and width of the cursor can be changed, as long as a number between 1 and 31 is used. The cursor may also be completely black or reversed video, and the blinking rate is adjustable when the BKEN bit in the LCD blink control (LBLKC) register is set.

Frame rate control is mainly used for grayscale displays and generates a maximum of 16 grayscale levels out of 16 density levels, as shown in Table 8-1 on page 8-7. The density level corresponds to the number of times that a pixel is turned on when the display is refreshing. Since crystal formulations and driving voltage may vary, the quality of the grayscale can be fine-tuned by programming the LCD gray palette mapping register (LGPMR).

The LCD interface logic is used to pack the display data into the correct size and output it to the LCD panel's data bus. The polarity of the LFLM, LP, and LCLK signals and pixel data can all be programmed to suit different LCD panel requirements.

8.2.1 Connecting the LCD Controller to an LCD Panel

The following signals are used to connect the LCD controller to an LCD panel:

- LD[7:0]—The LCD Data bus lines transfer pixel data to the LCD panel so that it can be displayed. Data is arranged differently on the bus, depending on which LCD panel mode is selected. The output pixel data can be negated through programming. See Section 8.3.10, "LCD Polarity Configuration Register," for more information. The LCD controller is initially configured to drive single-screen monochrome LCD panels. The data bus size for an LCD panel can be configured to 1, 2, 4, or 8 bits by programming the LPICF register.
- LFLM—The LCD Frame Marker signal indicates the start of a new display frame. LFLM becomes active after the last line pulse of the frame and remains active until the next line pulse, at which point it deasserts and remains inactive until the next frame. The LFLM can be programmed to be an active high or active low signal in software. See Section 8.3.10, "LCD Polarity Configuration Register," for more information.
- LLP—The LCD Line Pulse signal is used to latch a line of shifted data onto an LCD panel. The LLP can be programmed to be an active high or active low signal in software. See Section 8.3.10, "LCD Polarity Configuration Register," for more information.
- LCLK—The LCD Shift Clock signal is the clock output to which the output data to the LCD panel is synchronized. The LCLK can be programmed to be an active high or active low signal in software. See Section 8.3.10, "LCD Polarity Configuration Register," for more information.
- LACD—The LCD Alternate Crystal Direction output signal is toggled to alternate the crystal polarization on the panel. This signal can be programmed to toggle for a period of 1 to 16 frames. The LACD signal will toggle after a preprogrammed number of FLM or LP pulses. It can be programmed so that the LACD will toggle once every 1 to N frames or LLP pulse. The targeted number N is equal to the alternation code's 7-bit value plus one. The default value for LACDRC is 0, which enables the LACD signal to toggle on every frame. See Section 8.3.11, "LACD Rate Control Register," for more information.

8.2.1.1 Panel Interface Timing

The LCD controller continuously passes the pixel data into the LCD panel via the LCD data bus. The bus is timed by the LCLK, LLP, and LFLM signals. The LCLK signal clocks the pixel data into the display drivers' internal shift register. The LLP signal latches the shifted pixel data into a wide latch at the end of a line, while the LFLM signal marks the first line of the displayed page.

The LCD controller is designed to support most monochrome LCD panels. Figure 8-2 on page 8-4 illustrates the LCD interface timing for 1-, 2-, and 4-bit LCD data bus operation. The LLP signal signifies the end of the current line of serial data. The LLP signal enclosed by the LFLM signal marks the end of the first line of the current frame.

Some LCD panels can use an active low LFLM, LLP, or LCLK signal and reversed pixel data. To change the polarity of these signals, set the FLMPOL, LPPOL, LCKPOL, and PIXPOL bits in the LCD polarity configuration (LPOLCF) register to 1. In addition to the interface timing pins, the LACD pin will toggle after a preprogrammed number of LFLM pulses. The purpose of this pin is to prevent the crystal in the LCD panel from degrading.

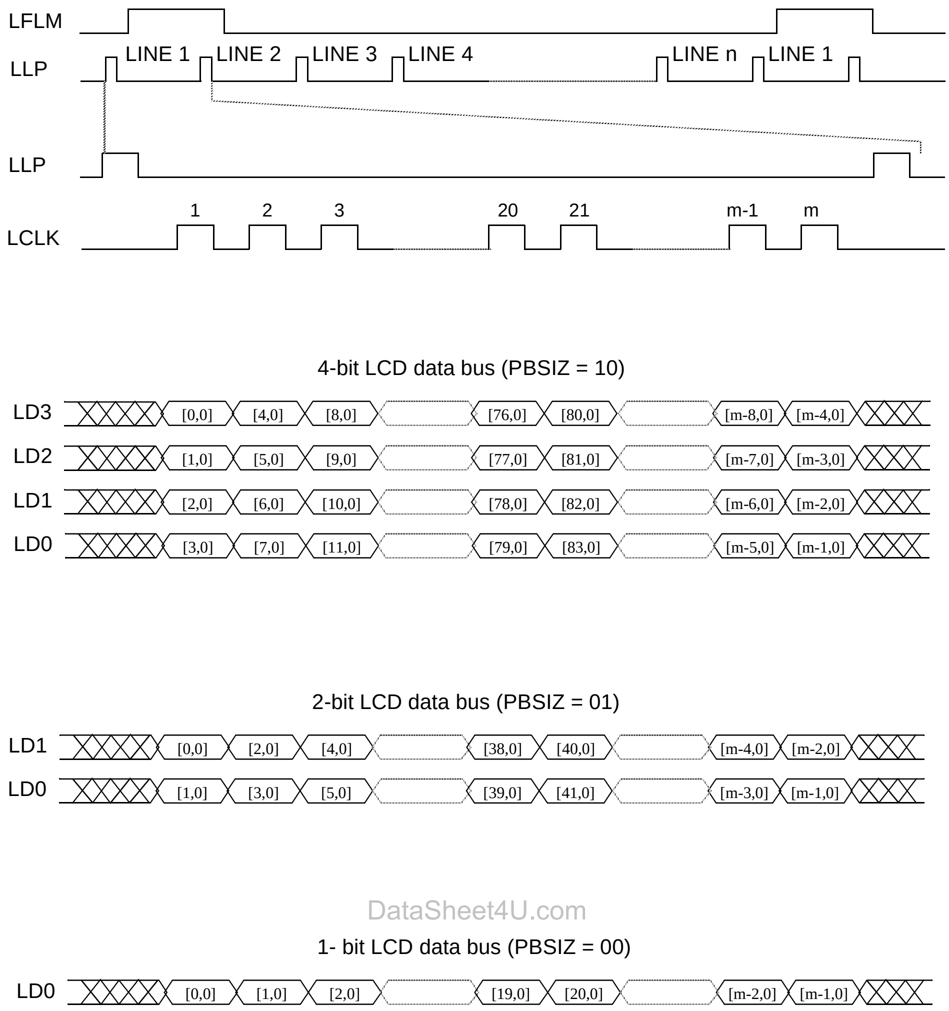


Figure 8-2. LCD Interface Timing for 4-, 2-, and 1-Bit Data Widths

8.2.2 Controlling the Display

The LCD controller is designed to drive single-screen monochrome STN LCD panels with up to 640 × 512 pixels in black-and-white display and 320 × 240 pixels in gray level display. A screen size larger than 320 × 240 for gray level display may cause flickering due to a slow refresh rate. The best efficiency is achieved when the screen width is a multiple of the DMA controller’s 16-bit bus width.

8.2.2.1 Format of the LCD Screen

The screen width and height of the LCD panel are programmable through software. Figure 8-3 on page 8-5 illustrates the relationship between the portion of a large graphics file displayed on the screen and the actual page. The units in the figure are measured in pixel counts.

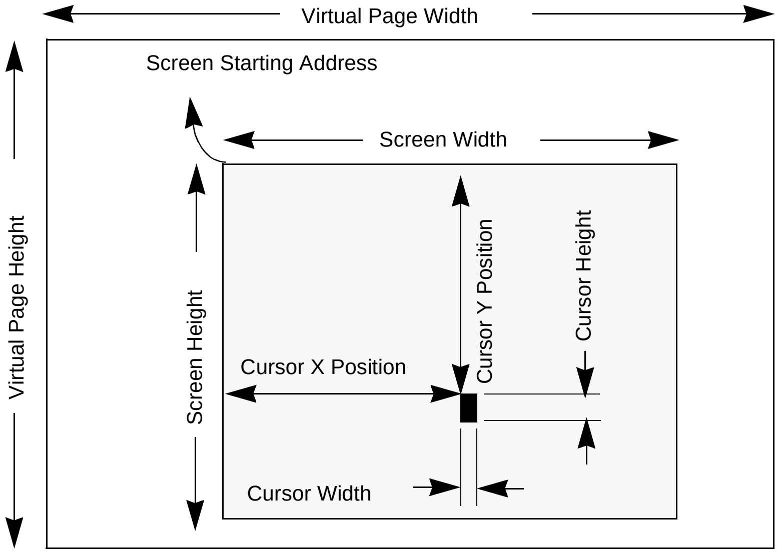


Figure 8-3. LCD Screen Format

The LCD screen width (LXMAX) and LCD screen height (LYMAX) registers are where the size of the LCD panel is specified. The LCD controller will start scanning the display memory at the location pointed to by the LCD screen starting address (LSSA) register. Therefore, the shaded area in Figure 8-3 will be displayed on the LCD panel.

The maximum page width and page height are specified by the LCD virtual page width (LVPW) and LCD virtual page height parameters. By changing the LSSA register, a screen-sized window can be vertically or horizontally scrolled (panned) anywhere inside the virtual page boundaries. However, it is up to the programmer, through software, to position the starting address so that the scanning logic's system memory pointer does not stretch beyond the virtual page width or height. Otherwise, strange objects will appear on the screen. The LVPH parameter shows the bottom of the page, but it is not used by the LCD controller.

8.2.2.2 Format of the Cursor

To define the position of the hardware cursor, the LCD controller maintains a vertical line counter (YCNT) to keep track of the current pixel's vertical position. YCNT, in conjunction with XCNT (the horizontal pixel counter), specifies the screen position of the pixel data being processed. When the pixel falls within a window specified by the cursor's reference position, cursor width, and cursor height, the original pixel bits can be shown with different properties. These properties can be transparent (cursor is disabled), full (black cursor), reversed video, full (white cursor), or blinking. The hardware cursor blink can be made to blink by setting the BKEN bit in the LBLKC register to 1, which alternates the original signal and cursor periodically. The speed at which the cursor blinks may be controlled by selecting the BDx bit in the LBLKC register. The half-period may be as long as 2 seconds.

8.2.2.3 Mapping the Display Data

The LCD controller supports 1 or 2 bits per pixel graphics mode. In the 1-bit 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. In 2-bit mode, each pixel is represented by two bits of display memory. To map the data to the LCD panel, program the appropriate bit in the corresponding address of the display memory. Figure 8-4 illustrates how the system memory data in both modes are mapped.

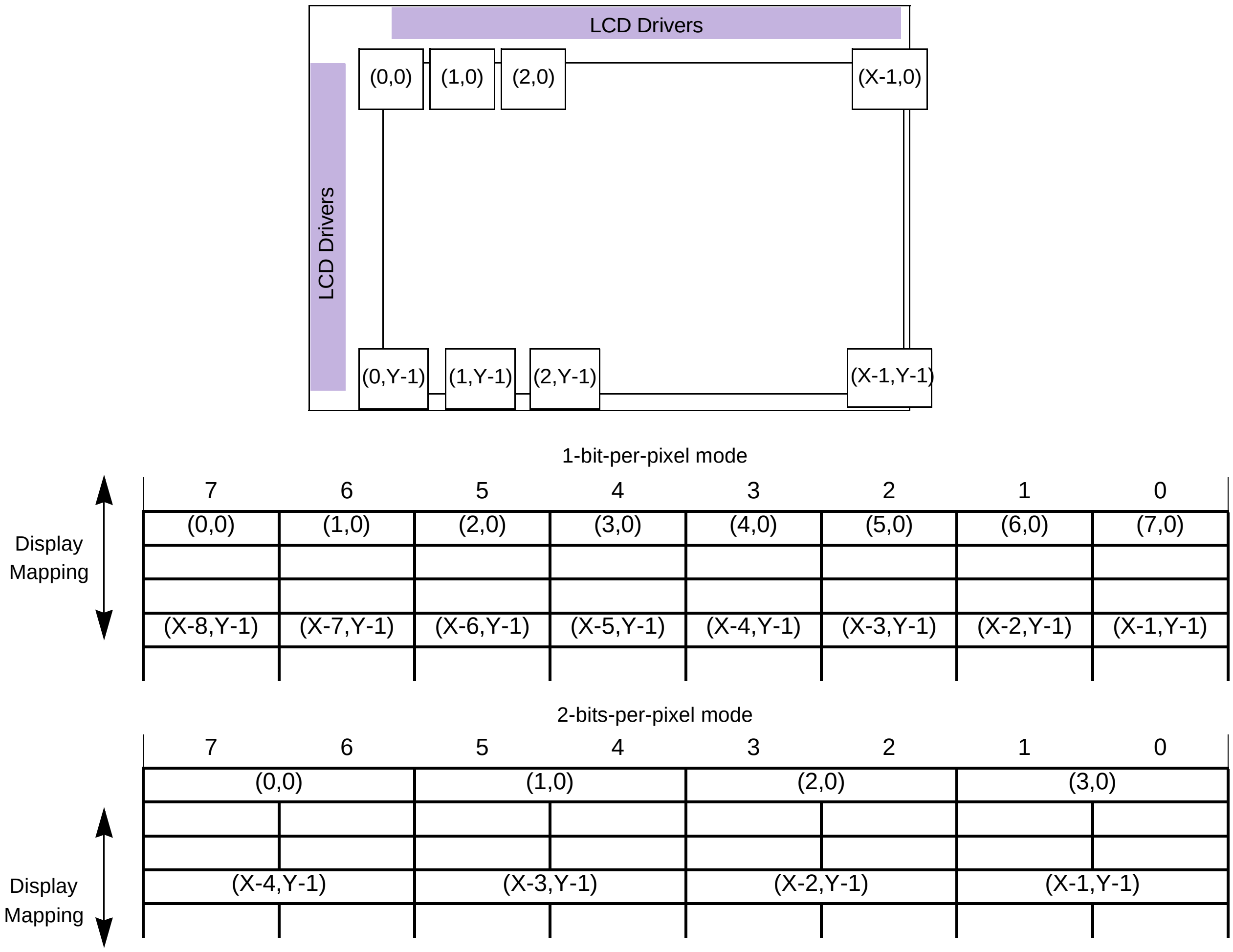


Figure 8-4. Mapping Memory Data on the Screen

8.2.2.4 Generating Grayscale Tones

In 2-bits-per-pixel mode, circuitry inside the LCD controller generates intermediate grayscale tones on the LCD panel by adjusting the density of ones and zeroes that appear over the frames. The LCD controller can generate 16 simultaneous grayscale levels out of a palette consisting of 16 shades. The two levels between black and white can be selected using the information in Table 8-1 on page 8-7. Use the LGPMR registers to program the grayscale level.

Table 8-1.   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

Since crystal formulations and driving voltages vary, the visual grayscale effect may or may not be linearly related to the frame rate. For certain types of graphics, a logarithmic scale like zero, one-fourth, one-half, and one might be more visually pleasing than a linearly spaced scale like zero, five-sixteenths, eleven-sixteenths, and one. This flexible mapping scheme allows optimizing the visual effect for the specific panel or application during a four-level grayscale display mode.

NOTE:

The Controlling Frame Rate Modulation function available in previous versions of the DragonBall integrated processor is not available in the MC68VZ328.

8.2.3 Using Low-Power Mode

Some panels may have a PANEL_OFF signal, which is used to turn off the panel for low-power mode. In an MC68VZ328 system, this signal is not supported, but can be easily implemented using a parallel I/O pin. The software can be programmed to achieve PANEL_OFF by using parallel I/O in the following sequence:

1. Drive the LCD bias voltage to 0 V.
2. Set the LCDON bit to 0 in the LCD clocking control (LCKCON) register, turning off the LCD controller.

To turn the LCD controller back on, follow the following steps:

1. Set the LCDON bit to 1 in the LCKCON register, which turns on the LCD controller.
2. Pause for 1 or 2 ms.
3. Drive the LCD bias voltage to +15 V or -15 V.

When setting the LCDON bit in the CLKCON register to 0, the LCD controller will enter low-power mode by stopping its own pixel clock prior to the next line buffer fill DMA. Further screen DMA and display refresh operations will then be halted in this mode. When the LCD controller is turned back on, DMA and screen refresh activities will resume synchronously.

8.2.4 Using the DMA Controller

The LCD DMA controller is a fly-by-type, 16-bit-wide, fast data transfer device. Since the LCD screen has to be continuously refreshed at a rate of 50 Hz to 70 Hz, the pixel bits in the memory will be read and transferred to the corresponding pixels on the screen. To minimize bus obstruction, a burst type and fly-by transfer is required. Each cycle is evenly distributed across the time frame. Every time the internal line buffer needs data, it asserts the $\overline{BR}$ signal to request the bus from the core. Once the core grants the bus ($\overline{BG}$ is asserted), the DMA controller gets control of the bus signal and issues a number of words read from memory. The read data is then internally passed to the internal pixel buffer. During the LCD access cycles, output enable and chip-select signals for the corresponding system memory chip are asserted by the chip-select logic inside the system integration module. It is possible to minimize bus bandwidth obstruction by using zero LCD access wait-states (one clock per access).

8.2.4.1 Bus Bandwidth Calculation Example

Since LCD screen refresh occurs periodically, the load that the LCD controller puts on the host data bus becomes an important consideration to the high-performance handheld system designer. There are many issues involved in estimating bandwidth overhead to the data bus. Consider a typical scenario:

- Screen size: 320 × 240 pixels
- Bits per pixel: 2 bits per pixel
- Screen refresh rate: 60 Hz
- System clock: 16.58 MHz
- Host bus size: 16 bit
- DMA access cycle: 2 cycles per 16-bit word

The following T_l period is used by the LCD controller to update one line of the screen:

$$
\begin{aligned}
T_l &= \frac{1}{60\ \text{Hz}} \times \frac{1}{240\ \text{lines}} \\
&= 69.4\ \mu\text{s}
\end{aligned}
$$

During the same period, the line buffer must be filled. The following T_{DMA} duration is how long the DMA cycle will hold up the bus:

$$
\begin{aligned}
T_{DMA} &= \frac{320\ \text{pixels} \times 2\ \text{bits per pixel} \times 2\ \text{clocks}}{16.67\ \text{MHz} \times 16\text{-bit bus}} \\
&= 4.8\ \mu\text{s}
\end{aligned}
$$

Thus, the percentage of host bus time taken up by the LCD controller’s DMA is P_{DMA}:

$$
\begin{aligned}
P_{DMA} &= \frac{4.8\text{‘}\ \mu\text{s}}{69.4\text{‘}\ \mu\text{s}} \\
&= 6.92\text{‘}\%
\end{aligned}
$$

8.2.5 Self-Refresh Mode

The LCD driver from Epson was used as a reference for the design of the refresh mode. In self-refresh mode, the LCD module will update the screen periodically from internal RAM using the LP and FRM pulse.

8.2.5.1 Entering Self-Refresh Mode

Setting the self-refresh register bit 7 to 1 means that the LSCLK and LD will remain 0 when the end of the frame is reached. The LP and FRM pulse continue as in normal mode, but there are no pulses on either the LSCLK or LD.

8.2.5.2 Canceling Self-Refresh Mode

Setting the self-refresh register bit 7 to 0 means that the normal mode is entered when the end of the frame is reached. On entering normal mode, data is sent out from the beginning of the page.

8.3 Programming Model

The remaining sections of this chapter provide detailed descriptions of the registers, their settings, and sample programming examples.

8.3.1 LCD Screen Starting Address Register

The LCD screen starting address (LSSA) register is used to inform the LCD panel where to fetch the data to be displayed. 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 8-2.

LSSA LCD Screen Starting Address Register **0x(FF)FFFA00**

	BIT 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	BIT 16
	SSA 31	SS A30	SS A29	SS A28	SS A27	SS A26	SS A25	SS A24	SS A23	SS A22	SS A21	SS A20	SS A19	SS A18	SS A17	SSA 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

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SSA 15	SS A14	SS A13	SS A12	SS A11	SS A10	SS A9	SS A8	SS A7	SS A6	SS A5	SS A4	SS A3	SS A2	SS A1	
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 8-2. LCD Screen Starting Address Register Description

Name	Description	Setting
SSAx Bits 31–1	**Screen Starting Address 31–1**—This field is the screen starting address of the LCD panel. The LCD controller will start fetching pixel data from system memory at this address. This field must start at a location that will enable a complete picture to be stored in 1 Mbyte memory boundary (A[19:00]). In other words, A[31:20] has a fixed value for a picture's image.	See description.
Reserved Bit 0	Reserved	This bit is reserved and should be set to 0.

8.3.2 LCD Virtual Page Width Register

The LCD virtual page width (LVPW) register contains the width of the displayed image. 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 8-3.

LVPW LCD Virtual Page Width Register **0x(FF)FFFA05**

	BIT 7	6	5	4	3	2	1	BIT 0
	VP8	VP7	VP6	VP5	VP4	VP3	VP2	VP1
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
				0xFF				

Table 8-3. LCD Virtual Page Width Register Description

Name	Description	Setting
VPx Bits 7–0	**Virtual Page Width 8–1**—These bits specify the virtual page width of the LCD panel in terms of word count. The virtual page width is the virtual width in pixels divided by 16 for a black-and-white display, by 8 for a 4-grayscale display, and by 4 for a 16-grayscale display.	See description

8.3.3 LCD Screen Width Register

The LCD screen width register (LXMAX) is used to specify the width of the LCD panel's screen in pixels. This register must be a multiple of 16. 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 8-4.

LXMAX LCD Screen Width Register **0x(FF)FFFA08**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
							XM9	XM8	XM7	XM6	XM5	XM4				
TYPE							rw	rw	rw	rw	rw	rw				
RESET	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0
								0x03F0								

Table 8-4. LCD Screen Width Register Description

Name	Description	Setting
Reserved Bits 15–10	Reserved	These bits are reserved and should be set to 0.
XMx Bits 9–4	**Maximum Width 9–4**—These bits represent the width of the LCD panel in the number of pixels.	See description.
Reserved Bits 3–0	Reserved	These bits are reserved and should be set to 0.

8.3.4 LCD Screen Height Register

The LCD screen height register (LYMAX) is used to define the height of the LCD panel's screen in pixels. 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 8-5.

LYMAX LCD Screen Height Register **0x(FF)FFFA0A**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
								YM8	YM7	YM6	YM5	YM4	YM3	YM2	YM1	YM0
TYPE								rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1

0x01FF

Table 8-5. LCD Screen Height Register Description

Name	Description	Setting
Reserved Bits 15–9	Reserved	These bits are reserved and should be set to 0.
YMx Bits 8–0	**Maximum Height 8–0**—These bits represent the height of the LCD panel in the number of pixels, which is equal to YMAX + 1.	See description.

8.3.5 LCD Cursor X Position Register

The LCD cursor X position (LCXP) register is used to determine the horizontal pixel position of the cursor on the LCD panel. 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 8-6.

LCXP LCD Cursor X Position Register **0x(FF)FFFA18**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	CC1	CC0					CXP9	CXP8	CXP7	CXP6	CXP5	CXP4	CXP3	CXP2	CXP1	CXP0
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 8-6. LCD Cursor X Position Register Description

Name	Description	Setting
CCx Bits 15–14	**Cursor Control 1 and 0**—These bits are used to control the format of the cursor.	00 = Transparent, cursor is disabled. 01 = Full (black) cursor. 10 = Reversed video. 11 = Full (white) cursor.
Reserved Bits 13–10	Reserved	These bits are reserved and should be set to 0.

Table 8-6. LCD Cursor X Position Register Description (Continued)

Name	Description	Setting
CXPx Bits 9–0	**Cursor X Position 9–0**—These bits represent the cursor's horizontal starting position, X, in terms of pixel count (from 0 to XMAX).	See description.

8.3.6 LCD Cursor Y Position Register

The LCD cursor Y position (LCYP) register is used to determine the vertical pixel position of the cursor on the LCD panel. 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 8-7.

LCYP LCD Cursor Y Position Register **0x(FF)FFFA1A**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
								CYP 8	CYP 7	CYP 6	CYP 5	CYP 4	CYP 3	CYP 2	CYP 1	CYP0
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 8-7. LCD Cursor Y Position Register Description

Name	Description	Setting
Reserved Bits 15–9	Reserved	These bits are reserved and should be set to 0.
CYPx Bits 8–0	**Cursor Vertical Y Pixel 8–0**—These bits represent the cursor's vertical starting position, Y, in terms of pixel count (from 0 to YMAX).	See description.

8.3.7 LCD Cursor Width and Height Register

The LCD cursor width and height (LCWCH) register is used to determine the width and height of the cursor, in screen pixels. 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 8-8.

LCWCH LCD Cursor Width and Height Register **0x(FF)FFFA1C**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
				CW4	CW3	CW2	CW1	CW0				CH4	CH3	CH2	CH1	CH0
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	0	1

0x0101

Table 8-8. LCD Cursor Width and Height Register Description

Name	Description	Setting
Reserved Bits 15–13	Reserved	These bits are reserved and should be set to 0.
CWx Bits 12–8	**Cursor Width 4–0**—These bits specify the width of the hardware cursor in pixel count (from 1 to 31).	See description.
Reserved Bits 7–5	Reserved	These bits are reserved and should be set to 0.
CHx Bits 4–0	**Cursor Height 4–0**—These bits specify the height of the hardware cursor in pixel count (from 1 to 31).	See description.
Note: The cursor is disabled if the CWx or CHx bits are set to 0.		

8.3.8 LCD Blink Control Register

The LCD blink control register (LBLKC) is used to control how the cursor blinks. 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 8-9 on page 8-15.

LBLKC LCD Blink Control Register **0x(FF)FFFA1F**

	BIT 7	6	5	4	3	2	1	BIT 0
	BKEN	BD6	BD5	BD4	BD3	BD2	BD1	BD0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	1	1	1	1	1	1	1
	0x7F							

Table 8-9. LCD Blink Control Register Description

Name	Description	Setting
BKEN Bit 7	**Blink Enable**—This bit determines if the cursor will blink or remain steady.	1 = Blink is enabled 0 = Blink is disabled (default)
BDx Bits 6–0	**Blink Divisor 6–0**—These bits determine if the cursor will toggle once per a specified number of internal frame pulses plus one. The half-period may be as long as 2 seconds.	See description

8.3.9 LCD Panel Interface Configuration Register

The LCD panel interface configuration (LPICF) register is used to determine the data bus width of the LCD panel and to determine if it is a black-and-white or grayscale display. 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 8-10.

LPICF LCD Panel Interface Configuration Register **0x(FF)FFFA20**

	BIT 7	6	5	4	3	2	1	BIT 0
					PBSIZ1–0		GS1–0	
TYPE					rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 8-10. LCD Panel Interface Configuration Register Description

Name	Description	Setting
Reserved Bits 7–4	Reserved	These bits are reserved and should be set to 0.
PBSIZ1–0 Bits 3–2	**Panel Bus Width 1–0**—These bits specify the bus width of the LCD panel.	00 = 1 bit. 01 = 2 bit. 10 = 4 bit. 11 = 8 bit.
GS1–0 Bits 1–0	**Grayscale Mode Selection 1–0**—These bits determine the mode of operation of the grayscale display device.	00 = Black-and-white mode. 01 = Four-level grayscale mode. 10 = Sixteen-level grayscale mode. 11 = Reserved.

8.3.10 LCD Polarity Configuration Register

The LCD polarity configuration (LPOLCF) register controls the polarity of the interface signal that goes to the LCD panel. 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 8-11.

LPOLCF LCD Polarity Configuration Register **0x(FF)FFFA21**

	BIT 7	6	5	4	3	2	1	BIT 0
					LCKPOL	FLMPOL	LPPOL	PIXPOL
TYPE					rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 8-11. LCD Polarity Configuration Register Description

Name	Description	Setting
Reserved Bits 7–4	Reserved	These bits are reserved and should be set to 0.
LCKPOL Bit 3	**LCD Shift Clock Polarity**—This bit controls the polarity of the active edge of the LCD shift clock.	0 = Active negative edge of LCLK. 1 = Active positive edge of LCLK.
FLMPOL Bit 2	**Frame Marker Polarity**—This bit controls the polarity of the frame marker.	0 = Frame marker is active high. 1 = Frame marker is active low.
LPPOL Bit 1	**Line Pulse Polarity**—This bit controls the polarity of the line pulse.	0 = Line pulse is active high. 1 = Line pulse is active low.
PIXPOL Bit 0	**Pixel Polarity**—This bit controls the polarity of the pixels.	0 = Pixel polarity is active high. 1 = Pixel polarity is active low.

8.3.11 LACD Rate Control Register

The LCD alternate crystal direction rate control (LACDRC) register is used to control the alternate rates of the liquid crystal direction. 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 8-12 on page 8-17.

LACDRC LACD Rate Control Register **0x(FF)FFFA23**

	BIT 7	6	5	4	3	2	1	BIT 0
	ACDSLT	ACD6	ACD5	ACD4	ACD3	ACD2	ACD1	ACD0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 8-12. LACD Rate Control Register Description

Name	Description	Setting
ACDSLT Bit 7	**Clock Source Select**—This bit selects the clock source for the internal counter that generates an LACD signal.	0 = Select frame pulse as input clock 1 = Select line pulse as input clock
ACDx Bits 6–0	**Alternate Crystal Direction Control 6–0**—These bits represent the ACD toggle rate control code. The LACD signal will toggle once every 1 to 128 FLM/LP cycles based on the value specified in this register. The actual number of FLM cycles is the value programmed plus one. Shorter cycles tend to give better results.	See description

8.3.12 LCD Pixel Clock Divider Register

The LCD pixel clock divider (LPXCD) register is used to program the divider, which generates the pixel clock. 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 8-13.

LPXCD LCD Pixel Clock Divider Register **0x(FF)FFFA25**

	BIT 7	6	5	4	3	2	1	BIT 0
			PCD5	PCD4	PCD3	PCD2	PCD1	PCD0
TYPE			rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 8-13. LCD Pixel Clock Divider Register Description

Name	Description	Setting
Reserved Bits 7–6	Reserved	These bits are reserved and should be set to 0.
PCDx Bits 5–0	**Pixel Clock Divider 5–0**—These bits represent the pixel clock divisor. The LCDCLK signal from the PLL is divided by N (PCD5–0 + 1) to yield the actual pixel clock. Values of 1–63 will yield N = 2 to N = 64. If these bits are set to 0 (N = 1), the PIX clock will be used directly, bypassing the divider circuit. Refer to Chapter 4, “Clock Generation Module and Power Control Module,” for more information.	See description.

8.3.13 LCD Clocking Control Register

The LCD clocking control (LCKCON) register is used to enable the LCD controller and control the LCD memory cycle. 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 8-14.

LCKCON LCD Clocking Control Register **0x(FF)FFFA27**

	BIT 7	6	5	4	3	2	1	BIT 0
	LCDON	Unused						
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 8-14. LCD Clocking Control Register Description

Name	Description	Setting
LCDON Bit 7	**LCD Control**—This bit enables the LCD controller. Default is off.	0 = Disable the LCD controller 1 = Enable the LCD controller
Unused Bits 6–0	These bits are not used by the chip and may be used for temporary storage. At reset these bits are cleared.	See description

8.3.14 LCD Refresh Rate Adjustment Register

The LCD refresh rate adjustment (LRRA) register is used to fine-tune the display refresh rate by introducing an idle interval between alternate LCD DMA and display cycles. 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 8-15.

LRRA LCD Refresh Rate Adjustment Register **0x(FF)FFFA28**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
							RRA[9:0]									
TYPE							rw	rw	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 8-15. LCD Refresh Rate Adjustment Register Description

Name	Description	Setting
Reserved Bits 15–10	Reserved	These bits are reserved and should be set to 0.

Table 8-15. LCD Refresh Rate Adjustment Register Description (Continued)

Name	Description	Setting
RRAx Bits 9–0	**Refresh Rate 9–0**—These bits contain the frame period, which can be calculated as follows: FRAME PERIOD = (12 + XMAX + RRA) x YMAX x (PXCD + 1) x LCDCLK_PERIOD where: Frame period = time for each screen update XMAX = screen width in number of pixels RRA = hexadecimal value stored in the LRRA register YMAX = screen height in number of pixels PXCD = hexadecimal value stored in the LPXCD register LCDCLK_PERIOD: refer to Section 4.4.1, “PLL Control Register,” on page 4-8 for setting LCDCLK period	See description.

8.3.15 LCD Panning Offset Register

The LCD panning offset register (LPOSR) is used to control how many pixels the picture is shifted to the left. 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 8-16.

LPOSR LCD Panning Offset Register **0x(FF)FFFA2D**

	BIT 7	6	5	4	3	2	1	BIT 0
					POS3	POS2	POS1	POS0
TYPE					rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 8-16. LCD Panning Offset Register Description

Name	Description	Setting
Reserved Bits 7–4	Reserved	These bits are reserved and should be set to 0.
POSx Bits 3–0	**Pixel Offset Code**—These bits specify the number of pixels being shifted to the left of the display panel. This is independent of the black-and-white or gray mode.	0001 = Picture is shifted 1 pixel to the left. 0010 = Picture is shifted 2 pixels to the left. . . . 1111 = Picture is shifted 15 pixels to the left.
Note: When the LOPSR register is being modified, the software must adjust the cursor’s reference position.		

8.3.16 LCD Frame Rate Control Modulation Register

This register of address space 0x(FF)FFFA31 is used for frame rate modulation control in the MC68EZ328, but it is unused in the MC68VZ328. This register is removed and not available for the temporary storage of data.

8.3.17 LCD Gray Palette Mapping Register

For four-level grayscale displays, full black and full white are the two predefined display levels. The other two intermediate grayscale shading densities can be adjusted in the LCD gray palette mapping register (LGPMR). 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 8-17.

LGPMR LCD Gray Palette Mapping Register **0x(FF)FFFA33**

	BIT 7	6	5	4	3	2	1	BIT 0
	G23	G22	G21	G20	G13	G12	G11	G10
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	0	0	0	0	1	0	0
				0x84				

Table 8-17. LCD Gray Palette Mapping Register Description

Name	Description	Setting
G23–G20 Bits 7–4	**Grayscale 23–20**—These bits represent one of the two grayscale shading densities.	See description
G13–G10 Bits 3–0	**Grayscale 13–10**—These bits represent the other grayscale shading density.	See description

8.3.18 PWM Contrast Control Register

The pulse-width modulator contrast control register (PWMR) is used to control the PWMO signal, which adjusts the contrast of the LCD panel. 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 8-18.

PWMR PWM Contrast Control Register **0x(FF)FFFA36**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
						SRC1–0		CCPEN	PW7	PW6	PW5	PW4	PW3	PW2	PW1	PW0
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								

Table 8-18. PWM Contrast Control Register Description

Name	Description	Setting
Reserved Bits 15–11	Reserved	These bits are reserved and should be set to 0.
SRC1–0 Bits 10–9	**Source 1–0**—These bits select the input clock source for the PWM counter. The PWM output frequency is equal to the frequency of the input clock divided by 256.	00 = Line pulse. 01 = Pixel clock. 10 = LCD clock. 11 = Reserved.

Table 8-18. PWM Contrast Control Register Description (Continued)

Name	Description	Setting
CCPEN Bit 8	**Contrast Control Enable**—This bit is used to enable or disable the contrast control function.	0 = Contrast control is off. 1 = Contrast control is on.
PWx Bits 7–0	**Pulse Width 7–0**—This bit controls the pulse-width of the built-in pulse-width modulator, which controls the contrast of the LCD screen. See Chapter 15, “Pulse-Width Modulator 1 and 2,” for more information.	See description.

8.3.19 Refresh Mode Control Register

Only a single bit in this register is used to enable or disable LCD self-refresh mode. The remaining bits are reserved. The bit assignment for the register is shown in the following register display. The settings for the bit in the register is listed in Table 8-19.

RMCR Refresh Mode Control Register **0x(FF)FFFA38**

	BIT 7	6	5	4	3	2	1	BIT 0
	REF_ON							
TYPE	rw							
RESET	0	0	0	0	0	0	0	0
				0x00				

Table 8-19. Refresh Mode Control Register Description

Name	Description	Setting
REF_ON Bit 7	**Self-Refresh On**—Setting this bit enables the self-refresh mode of operation with the LCD panel.	0 = Disable self-refresh mode. 1 = Enter self-refresh mode.
Reserved Bits 6–0	Reserved	These bits are reserved and should be set to 0.
Note: On entering self-refresh mode, the LSCLK and LD[7:0] signals stay low. FRM and LP work as normal.		

8.3.20 DMA Control Register

The LCD controller contains an 8 × 16 pixel buffer, which stores DMA-in data from system memory. This data is then passed to the LCD for display. When enough data has been removed from the buffer that it needs to be refilled, a new DMA transfer must be initiated. The DMA control register controls when the buffer should be refilled and the DMA burst length used when refilling. The bit assignments for the register are shown in the following register display. The settings for the bits are listed in Table 8-20.

DMACR DMA Control Register **0x(FF)FFFA39**

	BIT 7	6	5	4	3	2	1	BIT 0
	DMABL[3:0]					DMATM[2:0]		
TYPE	rw	rw	rw	rw		rw	rw	rw
RESET	0	1	1	0	0	0	1	0
					0x62			

Table 8-20. DMA Control Register Description

Name	Description	Setting
DMABL[3:0] Bits 7–4	**DMA Burst Length**—This field sets the number of words to be loaded to the pixel buffer in each DMA cycle.	See description and table footnote.
Reserved Bit 3	Reserved	This bit is reserved and should be set to 0.
DMATM[2:0] Bits 2–0	**DMA Trigger Mark**—This field sets the low-level mark in the pixel buffer to trigger a DMA request. The low-level mark equals to the number of words left in the pixel buffer.	See description and table footnote.
Note: Since the FIFO size is 8 × 16, DMABL and DMATM must be programmed based on the following criteria: F_HI + F_LO <= 8 1 <= F_HI <= 8 1<= F_LO <= 6		

8.4 Programming Example

The following is an example of how to program the related registers to properly configure an LCD panel with a resolution of 240 × 160 pixels, 4 levels of grayscale, and a 4-bit LCD data interface. The virtual image is 320 pixels wide and panned by 3 pixels.

Example 8-1. Programming Example

```
LCDINT        move.l #$A80000,#$FFFA00   ;display data address starts at $A80000
              move.w #240,#$FFFA08       ;LCD horizontal size is 240
              move.w #159,#$FFFA0A       ;LCD vertical size is 160
              move.b #40,#$FFFA05        ;4 level gray and 320 pixels wide image
              move.b #$09,#$FFFA20       ;LCD panel data bus is 4 bits,4 level gray
              move.b #3,#$FFFA25         ;pixel clock rate equal 1/4 of LCDCLK from PLL
              move.b #10,#$FFFA29        ;refresh rate adjustment
              move.b #$03,#$FFFA2D       ;shift picture by 3 pixels
              move.b #$82,#$FFFA27       ;switch on LCDC, 2 wait state for memory cycle
```

Chapter 9
Interrupt Controller

This chapter describes the interrupt controller and all of the signals associated with it. The interrupt controller of the MC68VZ328 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)
- Timer unit 1 (level 6)
- Timer unit 2 (configurable from level 1 to 6)
- Pulse-width modulator unit 1 (level 6)
- Pulse-width modulator unit 2 (configurable from level 1 to 6)
- $\overline{\text{IRQ5}}$ external interrupt—pen (level 5)
- Serial peripheral interface unit 1 (configurable from level 1 to 6)
- Serial peripheral interface unit 2 (level 4)
- UART unit 1 (level 4)
- UART unit 2 (configurable from level 1 to 6)
- Software watchdog timer interrupt (level 4)
- Real-time clock (level 4)
- Real-time interrupt (level 4)
- Keyboard interrupt (level 4)
- General-purpose interrupt $\overline{\text{INT}}$[3:0] (level 4)—these pins can be used as keyboard interrupts
- $\overline{\text{IRQ3}}$ external interrupt (level 3)
- $\overline{\text{IRQ2}}$ external interrupt (level 2)
- $\overline{\text{IRQ1}}$ external interrupt (level 1)

9.1 Interrupt Processing

Interrupts on the MC68VZ328 are processed as illustrated in the flowchart shown in Figure 9-1. 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.

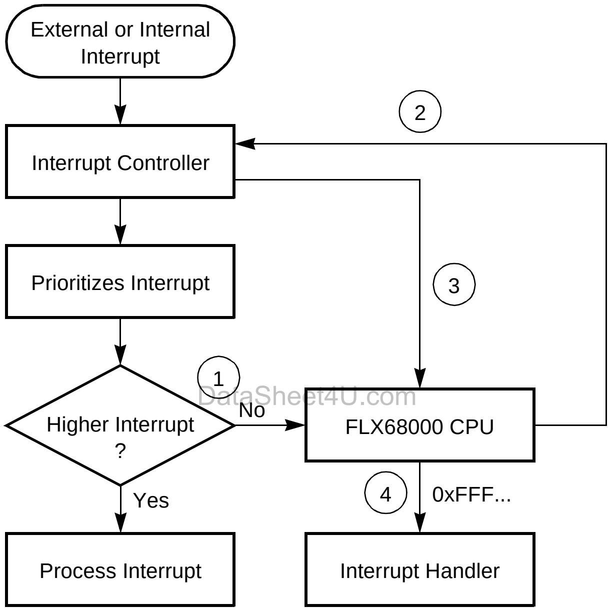


Figure 9-1. Interrupt Processing Flowchart

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 MC68VZ328, 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 MC68VZ328 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.

9.2 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 9-1. User interrupts are part of the exception processing on the MC68VZ328, and the vector numbers for user interrupts are configurable. For additional information regarding exception processing, see the *M68000 Family Programmer's Reference Manual*.

Table 9-1. 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 9-1. 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 MC68VZ328 does not provide autovector interrupts. At system startup, the user interrupt vector must be programmed, thereby allowing the processor to handle interrupts properly.

9.3 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 ($\overline{\text{CSA0}}$) is asserted and all other chip-selects are negated. The $\overline{\text{CSA0}}$ 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 MC68VZ328 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 MC68VZ328's $\overline{\text{RESET}}$ signal should be held low for at least 1.2 s after V_{DD} is applied. See Section 4.3.2.1, "PLLCLK Initial Power-up Sequence," on page 4-5 for detailed information about selecting the optimum $\overline{\text{RESET}}$ delay. After reset, all peripheral function signals and parallel I/O signals appear as inputs with pull-up resistors turned on, unless otherwise specified. The multiplexed, parallel I/O D[7:0]/PA[7:0] function is controlled by the WDTH8 bit in the system control register. If the value of WDTH8 is 0, it is D[7:0]. If WIDTH8 is 1, it is PA[7:0].

9.3.1 Operation Mode Selection During Reset

The MC68VZ328 supports three modes of operation: normal mode, emulation mode, and bootstrap mode. The selection of the modes is controlled by the $\overline{\text{EMUIRQ}}$, $\overline{\text{EMUBRK}}$, and $\overline{\text{HIZ}}$ signals during system reset, so special attention should be paid when using these signals. Refer to Chapter 2, "Signal Descriptions," for more information.

9.3.2 Data Bus Width for Boot Device Operation

The word size of the boot device (ROM/EPROM/FLASH) is determined by the BUSW signal. If it is high during the rising edge of the $\overline{\text{RESET}}$ signal, the 16-bit boot device will be configured. Otherwise, it will be configured as an 8-bit boot device.

9.4 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.

9.4.1 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 9.4, "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.

9.4.2 Interrupt Vectors

The MC68VZ328 provides one interrupt vector for each of the seven user interrupt levels. These interrupt vectors form the user interrupt vector section of Table 9-1 on page 9-3. 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).

9.5 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 9-1 on page 9-3. The coding for the vector numbers is provided in Table 9-2.

Table 9-2. 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.	

9.6 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 ($\overline{\text{INT}}$[3:0], $\overline{\text{IRQ1}}$, $\overline{\text{IRQ2}}$, $\overline{\text{IRQ3}}$, and $\overline{\text{IRQ6}}$) can be cleared by writing a 1 to the corresponding status bit in the 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.

9.6.1 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 MC68VZ328'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 9-3.

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 9-3. 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.

9.6.2 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 9-4.

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	POL5							
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 9-4. 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 9-4. 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.
POL5 Bit 7	**Polarity Control 5**—This bit controls interrupt polarity for the $\overline{\text{IRQ5}}$ 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.
Reserved Bits 6–0	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.

9.6.3 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
									ME MIQ	MR TI	MS PI1	MIR Q5	MIR Q6	MIR Q3	MIR Q2	MIR Q1
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

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
			MP WM 2	MU AR T2	MI NT 3	MI NT 2	MI NT 1	MI NT 0	MP WM 1	MK B	MT MR 2	MR TC	MW DT	MU AR T1	MT MR 1	MS PI2
TYPE			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

0xFFFF

Table 9-5. Interrupt Mask Register Description

Name	Description	Settings
Reserved Bits 31–24	Reserved	These bits are reserved and should be set to 0.
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.
MSPI1 Bit 21	**Mask SPI1 Interrupt**—When set, this bit indicates that the SPI 1 interrupt is masked. It is set to 1 after reset.	0 = Enable SPI 1 interrupt. 1 = Mask SPI 1 interrupt.
MIRQ5 Bit 20	**Mask $\overline{\text{IRQ5}}$ Interrupt**—When set, this bit indicates that $\overline{\text{IRQ5}}$ is masked. It is set to 1 after reset.	0 = Enable $\overline{\text{IRQ5}}$ interrupt. 1 = Mask $\overline{\text{IRQ5}}$ 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.

Table 9-5. Interrupt Mask Register Description (Continued)

Name	Description	Settings
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.
Reserved Bits 15–14	Reserved	These bits are reserved and should be set to 0.
MPWM2 Bit 13	**Mask PWM 2 Interrupt**—When set, this bit indicates that PWM 2 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 UART 2 is masked. It is set to 1 after reset.	0 = Enable UART 2 interrupt. 1 = Mask UART 2 interrupt.
MINT3 Bit 11	**Mask External $\overline{\text{INT3}}$ Interrupt**—Setting this bit masks the $\overline{\text{INT3}}$ interrupt. It is set to 1 after reset.	0 = Enable $\overline{\text{INT3}}$ interrupt. 1 = Mask $\overline{\text{INT3}}$ interrupt.
MINT2 Bit 10	**Mask External $\overline{\text{INT2}}$ Interrupt**—Setting this bit masks the $\overline{\text{INT2}}$ interrupt. It is set to 1 after reset.	0 = Enable $\overline{\text{INT2}}$ interrupt. 1 = Mask $\overline{\text{INT2}}$ interrupt.
MINT1 Bit 9	**Mask External $\overline{\text{INT1}}$ Interrupt**—Setting this bit masks the $\overline{\text{INT1}}$ interrupt. It is set to 1 after reset.	0 = Enable $\overline{\text{INT1}}$ interrupt. 1 = Mask $\overline{\text{INT1}}$ interrupt.
MINT0 Bit 8	**Mask External $\overline{\text{INT0}}$ Interrupt**—Setting this bit masks the $\overline{\text{INT0}}$ interrupt. It is set to 1 after reset.	0 = Enable $\overline{\text{INT0}}$ interrupt. 1 = Mask $\overline{\text{INT0}}$ interrupt.
MPWM1 Bit 7	**Mask PWM 1 Interrupt**—Setting this bit masks the PWM 1 interrupt. It is set to 1 after reset.	0 = Enable pulse-width modulator 1 interrupt. 1 = Mask pulse-width modulator 1 interrupt.
MKB Bit 6	**Mask Keyboard Interrupt**—Setting this bit masks the keyboard interrupt. It is set to 1 after reset.	0 = Enable keyboard interrupt. 1 = Mask keyboard 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 UART 1 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.
MSPI2 Bit 0	**Mask SPI 2 Interrupt**—When set, this bit indicates that the SPI 2 interrupt is masked. It is set to 1 after reset.	0 = Enable SPI 2 interrupt. 1 = Mask SPI 2 interrupt.

9.6.4 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 $\overline{\text{INT}}$[3:0], $\overline{\text{IRQ1}}$, $\overline{\text{IRQ2}}$, $\overline{\text{IRQ3}}$, 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
									EMIQ	RTI	SPI1	IRQ5	1RQ6	IRQ3	IRQ2	IRQ1
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

0x00000000

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
			PWM2	UART2	INT3	INT2	INT1	INT0	PWM1	KB	TMR2	RTC	WDT	UART1	TMR1	SPI2
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

0x00000000

Table 9-6. Interrupt Status Register Description

Name	Description	Settings
Reserved Bits 31–24	Reserved	These bits are reserved and should be set to 0.
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, “In-Circuit Emulation Module Status Register,” on page 16-10 for more information.	0 = No emulator interrupt is pending. 1 = An emulator interrupt is pending.

Table 9-6. Interrupt Status Register Description (Continued)

Name	Description	Settings
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.
SPI1 Bit 21	**SPI 1 Interrupt Status**—When set, this bit indicates an interrupt event from SPI unit 1.	0 = No SPI 1 interrupt is pending. 1 = An SPI 1 interrupt is pending.
IRQ5 Bit 20	**Interrupt Request Level 5**—This bit, when set, indicates that an external device is requesting an interrupt on level 5. If the $\overline{\text{IRQ5}}$ signal is set to be a level-sensitive interrupt, the source of the interrupt must first be cleared.	0 = No level 5 interrupt is pending. 1 = A level 5 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.
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.
Reserved Bits 15–14	Reserved	These bits are reserved and should be set to 0.
PWM2 Bit 13	**Pulse-Width Modulator 2 Interrupt**—This bit indicates that an interrupt event from PWM unit 2 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 PWM 2 interrupt is pending. 1 = A PWM 2 interrupt is pending.

Table 9-6. Interrupt Status Register Description (Continued)

Name	Description	Settings
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.
INT3 Bit 11	**External $\overline{\text{INT3}}$ Interrupt**—This bit, when set, indicates that a level 4 interrupt has occurred. It is usually for a keyboard interface. When it is programmed as edge-triggered, it can only be cleared by writing a 1 to the port D register. See Section 10.4.5, “Port D Registers,” on page 10-16 for details.	0 = No $\overline{\text{INT3}}$ interrupt is pending. 1 = An $\overline{\text{INT3}}$ interrupt is pending.
INT2 Bit 10	**External $\overline{\text{INT2}}$ Interrupt**—This bit, when set, indicates that a level 4 interrupt has occurred. It is usually for a keyboard interface. When it is programmed as edge-triggered, it can only be cleared by writing a 1 to the port D register. See Section 10.4.5, “Port D Registers,” on page 10-16 for details.	0 = No $\overline{\text{INT2}}$ interrupt is pending. 1 = An $\overline{\text{INT2}}$ interrupt is pending.
INT1 Bit 9	**External $\overline{\text{INT1}}$ Interrupt**—This bit, when set, indicates that a level 4 interrupt has occurred. It is usually for a keyboard interface. When it is programmed as edge-triggered, it can only be cleared by writing a 1 to the port D register. See Section 10.4.5, “Port D Registers,” on page 10-16 for details.	0 = No $\overline{\text{INT1}}$ interrupt is pending. 1 = An $\overline{\text{INT1}}$ interrupt is pending.
INT0 Bit 8	**External $\overline{\text{INT0}}$ Interrupt**—This bit, when set, indicates that a level 4 interrupt has occurred. It is usually for a keyboard interface. When it is programmed as edge-triggered, it can only be cleared by writing a 1 to the port D register. See Section 10.4.5, “Port D Registers,” on page 10-16 for details.	0 = No $\overline{\text{INT0}}$ interrupt is pending. 1 = An $\overline{\text{INT0}}$ interrupt is pending.
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 PWM 1 interrupt is pending. 1 = A PWM 1 interrupt is pending.
KB Bit 6	**Keyboard Interrupt Request**—This bit, when set, indicates that there is a level 4 interrupt event from a keyboard pending.	0 = No keyboard interrupt is pending. 1 = A keyboard 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.

Table 9-6. Interrupt Status Register Description (Continued)

Name	Description	Settings
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.
SPI2 Bit 0	**SPI Unit 2 Interrupt Status**—When set, this bit indicates an interrupt event from SPI unit 2.	0 = No SPI 2 interrupt is pending. 1 = An SPI 2 interrupt is pending.

9.6.5 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
									EMIQ	RTI	SPI1	IRQ5	IRQ6	IRQ3	IRQ2	IRQ1
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

0x00000000

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
			PWM2	UART2	INT3	INT2	INT1	INT0	PWM1	KB	TMR2	RTC	WDT	UART1	TMR1	SPI2
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

0x00000000

Table 9-7. Interrupt Pending Register Description

Name	Description	Settings
Reserved Bits 31–24	Reserved	These bits are reserved and should be set to 0.
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 16-10 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.
SPI1 Bit 21	**SPI 1 Interrupt Pending**—When set, this bit indicates an interrupt event from SPI unit 1.	0 = No SPI 1 interrupt is pending. 1 = An SPI 1 interrupt is pending.

Table 9-7. Interrupt Pending Register Description (Continued)

Name	Description	Settings
IRQ5 Bit 20	**Interrupt Request Level 5**—This bit, when set, indicates that an external device is requesting an interrupt on level 5. If the $\overline{\text{IRQ5}}$ signal is set to be a level-sensitive interrupt, the source of the interrupt must first be cleared.	0 = No level 5 interrupt is pending. 1 = A level 5 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.
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.
Reserved Bits 15–14	Reserved	These bits are reserved and should be set to 0.
PWM2 Bit 13	**Pulse-Width Modulator 2 Interrupt**—This bit indicates an interrupt event from PWM unit 2 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 PWM 2 interrupt. 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.
INT3 Bit 11	**External $\overline{\text{INT3}}$ Interrupt**—This bit, when set, indicates that a level 4 interrupt has occurred. It is usually for a keyboard interface. When it is programmed as edge-triggered, it can only be cleared by writing a 1 to the port D register. See Section 10.4.5, "Port D Registers," on page 10-16 for details.	0 = No $\overline{\text{INT3}}$ interrupt is pending. 1 = An $\overline{\text{INT3}}$ interrupt is pending.

Table 9-7. Interrupt Pending Register Description (Continued)

Name	Description	Settings
INT2 Bit 10	**External $\overline{\text{INT2}}$ Interrupt**—This bit, when set, indicates that a level 4 interrupt has occurred. It is usually for a keyboard interface. When it is programmed as edge-triggered, it can only be cleared by writing a 1 to the port D register. See Section 10.4.5, “Port D Registers,” on page 10-16 for details.	0 = No $\overline{\text{INT2}}$ interrupt is pending. 1 = An $\overline{\text{INT2}}$ interrupt is pending.
INT1 Bit 9	**External $\overline{\text{INT1}}$ Interrupt**—This bit, when set, indicates that a level 4 interrupt has occurred. It is usually for a keyboard interface. When it is programmed as edge-triggered, it can only be cleared by writing a 1 to the port D register. See Section 10.4.5, “Port D Registers,” on page 10-16 for details.	0 = No $\overline{\text{INT1}}$ interrupt is pending. 1 = An $\overline{\text{INT1}}$ interrupt is pending.
INT0 Bit 8	**External $\overline{\text{INT0}}$ Interrupt**—This bit, when set, indicates that a level 4 interrupt has occurred. It is usually for a keyboard interface. When it is programmed as edge-triggered, it can only be cleared by writing a 1 to the port D register. See Section 10.4.5, “Port D Registers,” on page 10-16 for details.	0 = No $\overline{\text{INT0}}$ interrupt is pending. 1 = An $\overline{\text{INT0}}$ interrupt is pending.
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 PWM 1interrupt. 1 = A PWM 1 interrupt is pending.
KB Bit 6	**Keyboard Interrupt Request**—This bit, when set, indicates that there is a level 4 interrupt event from a keyboard pending.	0 = No keyboard interrupt is pending. 1 = A keyboard 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.
SPI2 Bit 0	**SPI Unit 2 Interrupt Pending**—When set, this bit indicates an interrupt event from SPI unit 2.	0 = No SPI 2 interrupt is pending. 1 = An SPI 2 interrupt is pending.

9.6.6 Interrupt Level Register

TIMER 2, UART 2, PWM 2, and SPI 1 are new modules to the MC68VZ328 compared to the previous version, MC68EZ328. Interrupts generated from these modules are level configurable. The interrupt level control register (ILCR) controls the interrupt level for these interrupts.

ILCR Interrupt Level Register **0x(FF)FFF314**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
		SPI1_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															

Programming register bits 14–12, 10–8, 6–4, and 2–0 with the values shown in Table 9-8 causes the corresponding interrupt source to generate different interrupt levels.

Table 9-8. 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 four interrupts is set to the default level indicated:

- $\overline{\text{TIMER2IRQ}}$ (level 3)
- $\overline{\text{UART2IRQ}}$ (level 5)
- $\overline{\text{PWM2IRQ}}$ (level 3)
- $\overline{\text{SPI2IRQ}}$ (level 6)

9.7 Keyboard Interrupts

Keyboard interrupt features provide a smart power-management capability. The CPU core can be put to sleep when no key is being pressed. Once a key is pressed, however, the core wakes up to service the request. This event-driven approach significantly reduces power consumption. $\overline{\text{KB0}}$ to $\overline{\text{KB7}}$ (multiplexed with $\overline{\text{INT}}$[3:0], $\overline{\text{IRQ1}}$, $\overline{\text{IRQ2}}$, $\overline{\text{IRQ3}}$, and $\overline{\text{IRQ6}}$) are input pins for the keyboard interface. They are internally ORed together and generate an interrupt that indicates to the core that a key has been pressed.

9.8 Pen Interrupts

The MC68VZ328 is designed to support pen and touch panel inputs. In most of these systems, the setup involves a touch panel connected to an analog-to-digital (A/D) converter and the microprocessor. To achieve low power consumption and system performance, the A/D is usually connected to an interrupt of the microprocessor. When the touch panel is touched, the CPU is activated through the interrupt and the A/D starts collecting data. On the MC68VZ328, $\overline{\text{IRQ5}}$ is a level 5 interrupt with pull-up properties that is normally used as a pen interrupt. Connecting the $\overline{\text{IRQ5}}$ to a transistor network with the A/D, a pen-down interrupt can be implemented with the MC68VZ328 system. With the special design circuitry inside, this pen interrupt supports both pen-down and pen-up interrupts. The polarity of the pen interrupt can be set by programming the POL5 bit of the interrupt control register.

Chapter 10
I/O Ports

This chapter describes the 10 multipurpose ports of the MC68VZ328. It also describes how to use the ports for external I/O control and to determine the status of the external signals. All 10 ports (A–G, J, K, and M) 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. Although each port consists of a group of five to eight signal lines, all commands and actions occur at the pin level because each pin of a port is individually configured. 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.

When pins are programmed as GPIO, the direction of individual pins (input or output) can be configured, and pull-up resistors (or pull-down resistors in some ports) can be enabled or disabled. When pins are programmed as dedicated I/O, a pin's direction cannot be controlled. *A few exceptions to this rule are noted in the programming information about the specific ports.*

10.1 Port Configuration

With the exception of Port A, every port is multiplexed with at least one other dedicated I/O function. Several ports have pins that can be configured for one of several dedicated I/O functions. Table 10-1 on page 10-2 shows the I/O functions available for each port.

Ports are programmed by four dedicated 8-bit registers: direction, data, pull-up enable, and select. The exceptions are Port A and Port D. Port A does not have a select register since it can only be used as a GPIO. 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 MC68VZ328. Port D is unique in that it is used for handling external interrupts. It has four dedicated interrupt control registers in addition to the previously referenced four registers.

The I/O drive control register (IOCR) in system control controls the drive strength (in mA) of all I/O signals, including all of the ports. By default, all I/O pins on the MC68VZ328 default to a 4 mA driving current. After reset, it is recommended the user select 2 mA drive strength for those signals not requiring high current to ensure maximum power savings.

Table 10-1. Dedicated I/O Functions of Ports

Port	Dedicated I/O Module	Dedicated I/O Module	Dedicated I/O Module	Dedicated I/O Module
A	Lower byte of data bus			
B	Chip-select	DRAM controller	GP timers	PWM output
C	LCD controller			
D	Interrupt controller			
E	SPI	DRAM controller	UART	Bus control
F	DRAM controller	CGM	Address bits 23–20	Interrupt request 5
	LCD contrast	Chip-select		
G	Bus control	In-circuit emulation	Address bit 0	
J	UART	SPI		
K	Bus control	LCD controller	SPI	
M	DRAM controller			

10.2 Status of I/O Ports During Reset

Two types of resets affect the states of the MC68VZ328's I/O ports: warm reset and power-up reset. A warm reset refers to any reset initiated while power to the processor remains uninterrupted. A power-up reset occurs the first time power is supplied to the MC68VZ328. Power-up resets are also called cold start resets.

10.2.1 Warm Reset

Figure 10-1 on page 10-3 details timing during a warm reset. All I/O ports, except Ports B and M, reset to their default states on assertion of the reset signal and remain at their default states during the time period labeled Reset Assertion Time Length. The port default state is determined by the register reset values of the I/O port registers. Register reset values are found in Table 3-1 on page 3-2 and Table 3-2 on page 3-8. Ports B and M maintain their previous programmed states on reset assertion and retain their states during the Reset Assertion Time Length. The previous states of Ports B and M before reset assertion are, for the purposes of the figure, assumed.

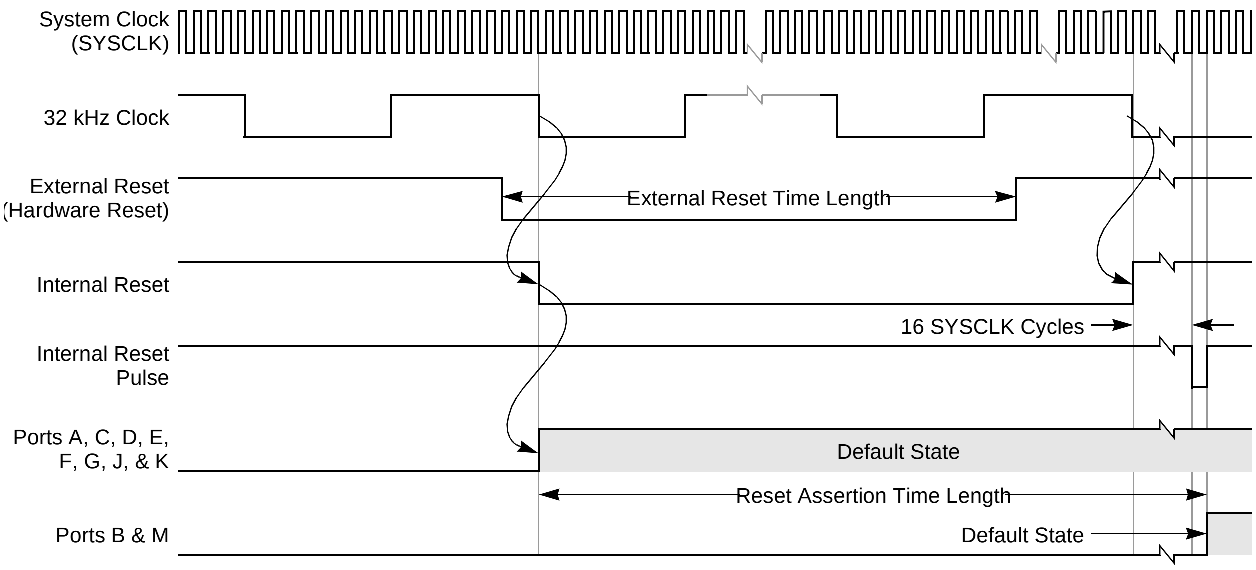


Figure 10-1. I/O Port Warm Reset Timing

As shown in Figure 10-1, resets for Ports A, C–G, J, and K are triggered by the assertion of the internal reset signal. The internal reset signal is synchronized with the first falling edge of the 32 kHz clock after the external reset has been asserted. The resets for Ports B and M are triggered by the negation of the internal reset pulse signal. The sequence of events (as shown in Figure 10-1) leading to the assertion of the internal reset pulse signal are as follows:

1. The external reset signal is negated.
2. The first falling edge of 32 kHz occurs.
3. After 16 cycles of SYSCLK, the internal reset pulse, whose width is 1 SYSCLK cycle, is generated.

Port B and Port M are designed to maintain or hold their previous states during the Reset Assertion Time Length to support the “data retention during reset” feature of the DRAM controller. Holding the previous states of Port B and Port M allows multiplexed DRAM control signals to remain active during the system Reset Assertion Time Length. This feature allows the DRAM controller to maintain the refresh cycles for DRAM during unpredictable reset time lengths, thereby preserving DRAM data after reset negation. More details appear in Chapter 7, “DRAM Controller.”

10.2.2 Power-up Reset

The power-up reset sequence of events is the same as for a warm reset, except that the I/O states of Port B and Port M are unknown during the Reset Assertion Time Length. Because Port B and Port M do not reset until the negation of the internal reset pulse signal, they do not have a previous state on a power-up reset.

While preliminary testing indicates that, on power-up reset, Ports B and M are configured as inputs with internal resistors enabled, this cannot be guaranteed. For any external device that may be sensitive to the brief unknown states of Port B or Port M on power-up resets, it is recommended that the device be connected to other available ports whose state can be ascertained.

10.2.3 Summary of Port Behavior During Reset

Table 10-2 summarizes the behavior of all MC68VZ328 I/O ports during the Reset Assertion Time Length (see Figure 10-1 on page 10-3) for power-up resets and warm resets.

Table 10-2. MC68VZ328 I/O Port Status During the Reset Assertion Time Length

I/O Ports	Warm Reset	Power-up Reset
A	Resets to default state	Resets to default state
B	Maintains previous state	Unknown 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
G	Resets to default state	Resets to default state
J	Resets to default state	Resets to default state
K	Resets to default state	Resets to default state
M	Maintains previous state	Unknown state

Note: The default state is defined by the reset values of the corresponding I/O port's registers. Please refer to Table 3-1 on page 3-2 and Table 3-2 on page 3-8 for details.

10.3 I/O Port Operation

The following subsections describe details of the I/O ports' operation.

10.3.1 Data Flow from the I/O Module

The operation of a port connected to another module in the MC68VZ328 is illustrated in Figure 10-2 on page 10-5.

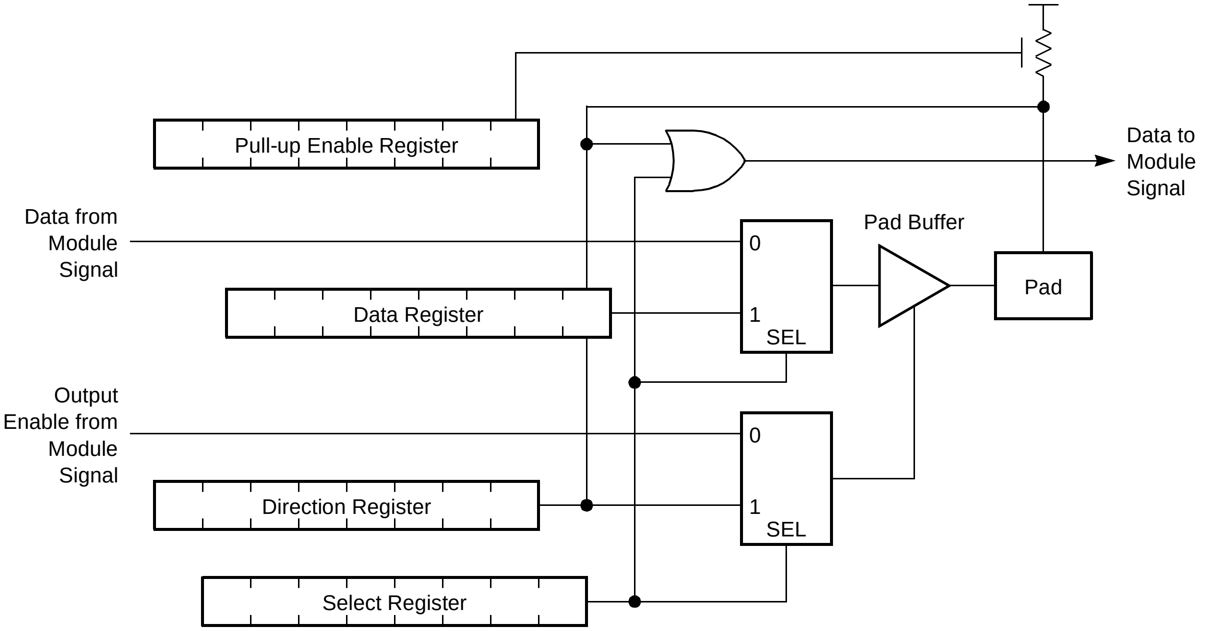


Figure 10-2. I/O Port Operation

For example, if Figure 10-2 represents the D0 bit of Port E, when the SEL0 in the select register is cleared, the “data from module” line is connected to the serial peripheral interface module’s TXD signal (SPITXD). Because SPITXD is output-only, the MC68VZ328 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 SPI module pin function is enabled. Bit D0 of Port E is the master SPMTXD signal. The SPI module controls the direction of data flow for the pin, which is always output. 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.

10.3.2 Data Flow to the I/O Module

An example of data flow to the I/O module is the D1 bit of Port E. This signal’s function is the SPI’s RXD (SPIRXD) signal. In this case, SPIRXD is input-only; thus, the chip negates the “output enable from module” line, and the “data from module” line is not disabled (see Figure 10-2). The “data to module” signal is connected to the SPIRXD input of the SPI.

10.3.3 Operating a Port as GPIO

While 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. While 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.

10.3.4 Port Pull-up and Pull-down Resistors

The pull-up and pull-down resistors 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 A–F, J, K, and M default to the I/O function with internal pull-up or pull-down enabled. Resistor assignments for individual ports is shown in Table 10-3. Meanwhile, Port G defaults to the dedicated function, except for the $\overline{\text{HIZ}}$/P/$\overline{\text{D}}$/PG3 pin, which defaults to the PG3 function.

Table 10-3. Pull-up and Pull-down Resistors by Port

Port	Pull-up	Pull-down
A, B, D, E, G, and J	All bits	None
C	None	All bits
F	Bits 7, 2–0	Bits 6–3
K	Bits 3–0	Bits 7–4
M	Bit 5	Bits 4–0

10.4 Programming Model

The chapter's remaining sections provide programming information about individual ports.

10.4.1 Port A Registers

The Port A registers are general-purpose 8-bit I/O registers. They consist of the following:

- Port A direction register (PADIR)
- Port A data register (PADATA)
- Port A pull-up enable register (PAPUEN)

Port A functions either as a GPIO (PA[7:0]) or the lower data byte of the data bus (D[7:0]). Port A can be used as PA[7:0] only when the MC68VZ328 is operating as an 8-bit system by setting the WDTH8 bit in the system control register (0xFFFFF000). If the MC68VZ328 is operating in either 16-bit or mixed 8- and 16-bit systems, the pins only function as D[7:0].

At reset the WDTH8 bit of the SCR is cleared, resulting in Port A becoming the lower data byte of the data bus (D[7:0]) with internal pull-up resistors enabled.

In sleep mode, all of the data bus pins (D[15:0]) are individually pulled up with 1 MΩ resistors.

10.4.1.1 Port A Direction Register

The Port A direction register controls the direction (input or output) of the line associated with the PADATA bit position. The settings for the bit positions are shown in Table 10-4.

PADIR Port A Direction Register **0x(FF)FFF400**

	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 10-4. Port A Direction Register Description

Name	Description	Setting
DIRx Bits 7–0	**Direction**—These bits control the direction of the pins in an 8-bit system.	0 = Input 1 = Output

10.4.1.2 Port A Data Register

The eight PADATA bits control or report the data on the pins while the associated SELx bits are high. While the DIRx bits are high (output), the Dx bits control the pins. While 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 will accept the data, but the data written to each cannot be accessed until the respective 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. The settings for the bit positions are shown in Table 10-5.

PADATA Port A Data Register **0x(FF)FFF401**

	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 10-5. Port A Data Register Description

Name	Description	Setting
Dx Bits 7–0	**Data**—These bits reflect 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

10.4.1.3 Port A Pull-up Enable Register

The Port A pull-up enable register (PAPUEN) controls the pull-up resistors for each line in Port A. The settings for the bit positions are shown in Table 10-6.

PAPUEN Port A Pull-up Enable Register **0x(FF)FFF402**

	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 10-6. Port A Pull-up Enable Register Description

Name	Description	Setting
PUx Bits 7–0	**Pull-up**—These bits enable the pull-up resistors on the port.	0 = Pull-up resistors are disabled 1 = Pull-up resistors are enabled

10.4.2 Port B Registers

Port B is made up 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.

10.4.2.1 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 10-7 on page 10-9.

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 10-7. Port B Direction Register Description

Name	Description	Setting
DIRx Bits 7–0	**Direction**—These bits control 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

10.4.2.2 Port B Data Register

The settings for the PBDATA bit positions are shown in Table 10-8.

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
RESET	1	1	1	1	1	1	1	1

0xFF*

*Actual bit value depends on external circuits connected to pin.

Table 10-8. Port B Data Register Description

Name	Description	Setting
Dx Bits 7–0	**Data**—These bits reflect 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, DRAM control, TIN/TOUT, and PWM dedicated I/O signals. These pins can be programmed as GPIO when these other assignments are not used.

These bits control or report the data on the pins while the associated SELx bits are high. While the DIRx bits are high (output), the Dx bits control the pins. While 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 will 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.

10.4.2.3 Port B Dedicated I/O Functions

The eight PBDATA lines are multiplexed with the chip-select, DRAM control, TIN/TOUT, and PWM dedicated I/O signals whose assignments are shown in Table 10-9.

Table 10-9. Port B Dedicated Function Assignments

Bit	GPIO Function	Dedicated I/O Functions
0	Data bit 0	$\overline{\text{CSB0}}$
1	Data bit 1	$\overline{\text{CSB1}}$/$\overline{\text{SDWE}}$
2	Data bit 2	$\overline{\text{CSC0}}$/$\overline{\text{RAS0}}$
3	Data bit 3	$\overline{\text{CSC1}}$/$\overline{\text{RAS1}}$
4	Data bit 4	$\overline{\text{CSD0}}$/$\overline{\text{CAS0}}$
5	Data bit 5	$\overline{\text{CSD1}}$/$\overline{\text{CAS1}}$
6	Data bit 6	TIN/TOUT
7	Data bit 7	PWMO1

Bits 1–5 operate as chip-select signals or DRAM signals. Signal selection is controlled by bit 9 (DRAM) in the chip-select D (CSD) register, which is described in Section 6.3.3, “Chip-Select Registers,” on page 6-8. Bit 0 is used only as D0 or $\overline{\text{CSB0}}$. No additional programming is required.

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 Section 12.1.4, “TOUT/TIN/PB6 Pin,” on page 12-3 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 Chapter 15, “Pulse-Width Modulator 1 and 2,” for additional information.

10.4.2.4 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 10-10 on page 10-11.

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 10-10. Port B Pull-up Enable Register Description

Name	Description	Setting
PUx Bits 7–0	**Pull-up**—These bits enable the pull-up resistors on the port.	0 = Pull-up resistors are disabled 1 = Pull-up resistors are enabled

10.4.2.5 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 10-11.

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 10-11. Port B Select Register Description

Name	Description	Setting
SELx Bits 7–0	**Select** —These bits select 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.

10.4.3 Port C Registers

Port C is composed of the following 8-bit general-purpose I/O registers:

- Port C direction register (PCDIR)
- Port C data register (PCDATA)
- Port C pull-down 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.

10.4.3.1 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 10-12.

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 10-12. Port C Direction Register Description

Name	Description	Setting
DIRx Bits 7–0	**Direction**—These bits control the direction of the pins in an 8-bit system. They reset to 0.	0 = Input 1 = Output

10.4.3.2 Port C Data Register

The settings for the PCDATA bit positions are shown in Table 10-13.

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
RESET	0	0	0	0	0	0	0	0
				0x00*				

*Actual bit value depends on external circuits connected to pin.

Table 10-13. Port C Data Register Description

Name	Description	Setting
Dx Bits 7–0	**Data**—These bits reflect 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 Section 8.2.1, “Connecting the LCD Controller to an LCD Panel,” on page 8-3 for more detailed information.

These bits control or report the data on the pins while the associated SELx bits are high. While the DIRx bits are high (output), the Dx bits control the pins. While 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 will 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.

10.4.3.3 Port C Dedicated I/O Functions

The eight PCDATA lines are multiplexed with the LCD controller dedicated I/O signals whose assignments are shown in Table 10-14.

Table 10-14. 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	LFLM
5	Data bit 5	LLP
6	Data bit 6	LCLK
7	Data bit 7	LACD

10.4.3.4 Port C Pull-down Enable Register

The Port C pull-down enable register (PCPDEN) controls the pull-down resistors for each line in Port C. The settings for the bit positions are shown in Table 10-15.

PCPDEN Port C Pull-down Enable Register **0x(FF)FFF412**

	BIT 7	6	5	4	3	2	1	BIT 0
	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
				0xFF				

Table 10-15. Port C Pull-down Enable Register Description

Name	Description	Setting
PDx Bits 7–0	**Pull-down**—These bits enable the pull-down resistors on the port.	0 = Pull-down resistors are disabled 1 = Pull-down resistors are enabled

10.4.3.5 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 10-16.

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 10-16. Port C Select Register Description

Name	Description	Setting
SELx Bits 7–0	**Select**—These bits select 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.

10.4.4 Port D Operation

Port D has the same functionality as other GPIO ports, except that it also has interrupt capabilities. It should be used as either a general-purpose, interrupt-generating port or as a keyboard input port. Figure 10-3 illustrates how this type of port operates.

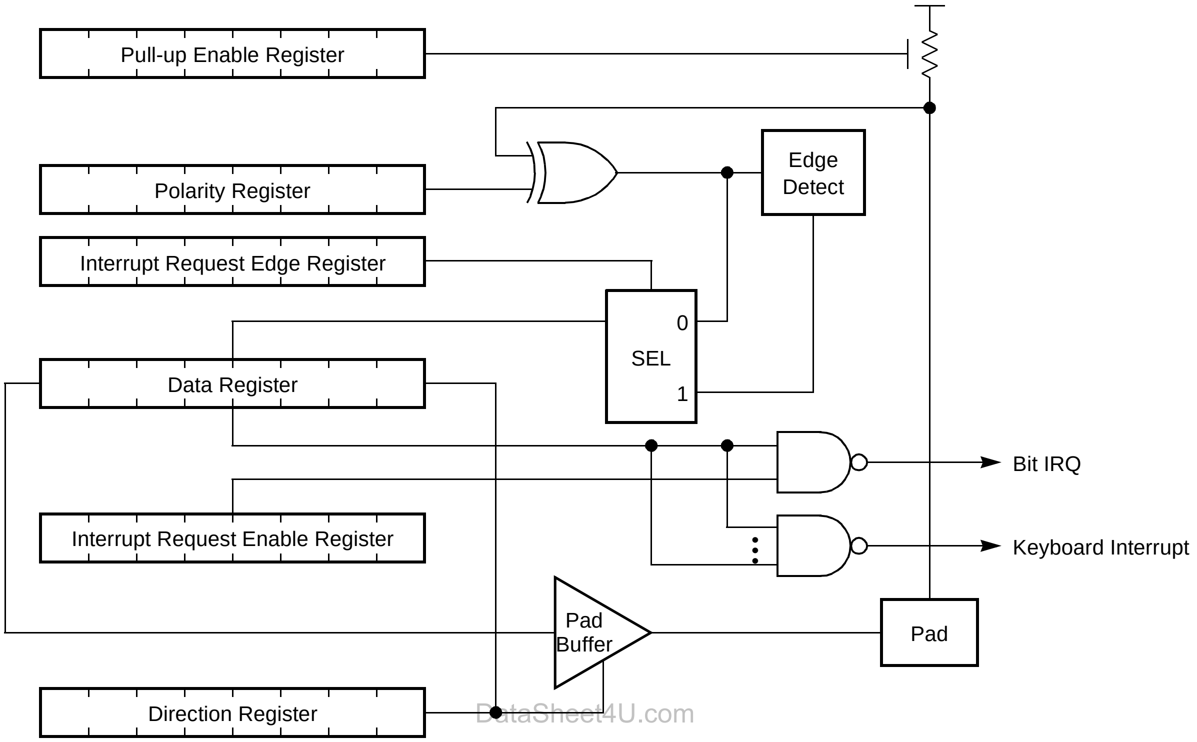


Figure 10-3. Interrupt Port Operation

Port D generates nine interrupt signals. Eight of these interrupts are generated by the bits of each port. One bit is the logical OR result of all eight bits, which is applied to the MC68VZ328 interrupt controller as a level 4 keyboard interrupt (KB) in the interrupt status register. See Section 9.6.4, “Interrupt Status Register,” on page 9-12 for more details.

10.4.5 Port D Registers

Unlike the other ports, Port D is unique in that it is comprised of eight 8-bit I/O registers. They consist of the following:

- Port D direction register (PDDIR)
- Port D data register (PDDATA)
- Port D pull-up enable register (PDPUEN)
- Port D select register (PDSEL)
- Port D polarity register (PDPOL)
- Port D interrupt request enable register (PDIRQEN)
- Port D keyboard enable register (PDKBEN)
- Port D interrupt request edge register (PDIRQEG)

10.4.5.1 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 10-17.

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 10-17. Port D Direction Register Description

Name	Description	Setting
DIRx Bits 7–0	**Direction**—These bits control the direction of the pins in an 8-bit system. They reset to 0.	0 = Input 1 = Output

10.4.5.2 Port D Data Register

The settings for the PDDATA bit positions are shown in Table 10-18.

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 10-18. Port D Data Register Description

Name	Description	Setting
Dx Bits 7–0	**Data**—These bits reflect 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 $\overline{\text{INT}}$ and $\overline{\text{IRQ}}$ dedicated I/O signals whose assignments are shown in Table 10-19. Port D signals can be programmed as GPIO when not used for handling external interrupts.

These bits control or report the data on the pins while the associated SELx bits are high. While the DIRx bits are high (output), the Dx bits control the pins. While 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 will 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.

Table 10-19. Port D Dedicated Function Assignments

Bit	GPIO Function	Dedicated I/O Function
0		$\overline{\text{INT0}}$
1		$\overline{\text{INT1}}$
2		$\overline{\text{INT2}}$
3		$\overline{\text{INT3}}$
4	Data bit 4	$\overline{\text{IRQ1}}$
5	Data bit 5	$\overline{\text{IRQ2}}$
6	Data bit 6	$\overline{\text{IRQ3}}$
7	Data bit 7	$\overline{\text{IRQ6}}$

10.4.5.3 Port D Interrupt Options

Interrupt bits 3–0 ($\overline{\text{INT}}$[3:0]), interrupt request bits 3–1 ($\overline{\text{IRQ}}$[3:1]), interrupt request bit 6 ($\overline{\text{IRQ6}}$), or Port D bits 7–0 can be configured as edge- or level-triggered interrupt signals.

NOTE:

> When external interrupts $\overline{\text{INT}}$[3:0], $\overline{\text{IRQ1}}$, $\overline{\text{IRQ2}}$, $\overline{\text{IRQ3}}$, and $\overline{\text{IRQ6}}$ are programmed as edge-triggered interrupts, they can be cleared by writing a 1 to the corresponding status bit in the interrupt status register in the interrupt controller. When programmed as level-triggered interrupts, these interrupts are cleared at the requesting sources.

To support keyboard applications, the I/O function can be used with interrupt capabilities, which are described in Chapter 9, “Interrupt Controller.”

The individual interrupt bits can be masked on a bit-by-bit basis. The KB is enabled or disabled by the KBENx bits of the PDKBEN register. Individual interrupts can be configured as either edge- or level-sensitive by asserting or clearing the IQEGx bits of the PDIRQEG register. Likewise, the polarity of the interrupt is determined by the POLx bits of the PDPOL register.

All of the interrupt signals in the table can be used as system wake-up interrupts, except for the edge interrupt on $\overline{\text{INT}}$[3:0]. Edge interrupts on $\overline{\text{INT}}$[3:0] can only interrupt the CPU when the system is awake. The $\overline{\text{INT}}$[3:0] signals are all level 4 interrupts, but $\overline{\text{IRQx}}$ has its own level. Any combination of Port D signals and OR (negative logic) can be selected to generate keyboard (KB) interrupts to the CPU. The $\overline{\text{KBx}}$ signal is an active low, level-sensitive interrupt of the selected pins. Like the other ports, each pin can be configured as an input or output on a bit-by-bit basis. When they are configured as inputs, each pin can generate a CPU interrupt.

10.4.5.4 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 10-20.

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 10-20. Port D Pull-up Enable Register Description

Name	Description	Setting
PUx Bits 7–0	**Pull-up**—These bits enable the pull-up resistors on the port.	0 = Pull-up resistors are disabled 1 = Pull-up resistors are enabled

10.4.5.5 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 10-21.

PDSEL Port D Select Register **0x(FF)FFF41B**

	BIT 7	6	5	4	3	2	1	BIT 0
	SEL7	SEL6	SEL5	SEL4				
TYPE	rw	rw	rw	rw				
RESET	1	1	1	1	0	0	0	0

0xF0

Table 10-21. Port D Select Register Description

Name	Description	Setting
SELx Bits 7–4	**Select**—These bits select 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 Bits 3–0	Reserved	These bits are reserved and should be set to 0.

10.4.5.6 Port D Polarity Register

These bits select the input signal polarity of $\overline{\text{INT}}$[3:0]. The polarity of the rising or falling edge is selected by the POLx bits. Interrupts are active high (or rising edge) when these bits are low. Interrupts are active low (or falling edge) while these bits are high. The settings for the bit positions of PDPOL are shown in Table 10-22.

PDPOL Port D Polarity Register **0x(FF)FFF41C**

	BIT 7	6	5	4	3	2	1	BIT 0
					POL3	POL2	POL1	POL0
TYPE					rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0

0x00

Table 10-22. Port D Polarity Register Description

Name	Description	Setting
Reserved Bits 7–4	Reserved	These bits are reserved and should be set to 0.
POLx Bits 3–0	**Polarity**—These bits determine the input signal polarity of $\overline{\text{INT}}$[3:0] interrupts.	0 = Data is unchanged. 1 = The input data is inverted before being presented to the holding register.

10.4.5.7 Port D Interrupt Request Enable Register

The interrupt enable bits (IQEN[3:0]) determine which $\overline{\text{INT}}$[3:0] will generate an interrupt to the interrupt controller module. The settings for the bit positions of PDIRQEN are shown in Table 10-23.

PDIRQEN Port D Interrupt Request Enable Register **0x(FF)FFF41D**

	BIT 7	6	5	4	3	2	1	BIT 0
					IQEN3	IQEN2	IQEN1	IQEN0
TYPE					rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 10-23. Port D Interrupt Request Enable Register Description

Name	Description	Setting
Reserved Bits 7–4	Reserved	These bits are reserved and should be set to 0.
IQENx Bits 3–0	**Interrupt Enable—**These bits select the $\overline{\text{INT}}$[3:0] pins that are presented to the interrupt controller.	0 = Interrupt disabled. 1 = Interrupt enabled.

10.4.5.8 Port D Keyboard Enable Register

All the selected signals are active low in reference to the external pins, and those that are asserted will generate a keyboard interrupt to the interrupt controller. When a KBENx bit is selected, the DIRx bits need to be configured as an input. The SELx, POLx, IQENx, and IQEGx bits have no effect on the functionality of KBENx. Deasserting the interrupt source is the only way to clear a keyboard interrupt. The settings for the bit positions of PDKBEN are shown in Table 10-24.

PDKBEN Port D Keyboard Enable Register **0x(FF)FFF41E**

	BIT 7	6	5	4	3	2	1	BIT 0
	KBEN7	KBEN6	KBEN5	KBEN4	KBEN3	KBEN2	KBEN1	KBEN0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 10-24. Port D Keyboard Enable Register Description

Name	Description	Setting
KBENx Bits 7–0	**Keyboard Enable—**These bits select the $\overline{\text{INT}}$[3:0] pins that are presented to the interrupt controller.	0 = The keyboard interrupt is disabled. 1 = The keyboard interrupt is enabled.

10.4.5.9 Port D Interrupt Request Edge Register

The polarity of the rising or falling edge is selected by the POLx bits. It should be noted that the edge-level interrupt for $\overline{\text{INT}}$[3:0] cannot be used for system wake up. The level-sensitive interrupt should be used. The settings for the bit positions of PDIRQEG are shown in Table 10-25 on page 10-21.

PDIRQEG Port D Interrupt Request Edge Register **0x(FF)FFF41F**

	BIT 7	6	5	4	3	2	1	BIT 0
					IQEG3	IQEG2	IQEG1	IQEG0
TYPE					rw	rw	rw	rw
RESET	0	0	0	0	0	0	0	0
					0x00			

Table 10-25. Port D Interrupt Request Edge Register Description

Name	Description	Setting
Reserved Bits 7–4	Reserved	These bits are reserved and should be set to 0.
IQEGx Bits 3–0	**Edge Enable**—The polarity of the rising or falling edge is selected by the POLx bits.	0 = Level-sensitive interrupts are selected. 1 = $\overline{\text{INT}}$[3:0] edge-sensitive interrupts are selected.

10.4.6 Port E Registers

Port E is composed 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)

Each signal in the PEDATA register connects to an external pin. As with the other ports, each bit on Port E is individually configured. Port E is multiplexed with the serial peripheral interface (SPI) and UART signals.

10.4.6.1 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 10-26.

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 10-26. Port E Direction Register Description

Name	Description	Setting
DIRx Bits 7–0	**Direction—**These bits control the direction of the pins in an 8-bit system. They reset to 0.	0 = Input 1 = Output

10.4.6.2 Port E Data Register

The settings for the bit positions of the PEDATA register are shown in Table 10-27.

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 10-27. Port E Data Register Description

Name	Description	Setting
Dx Bits 7–0	**Data**—These bits reflect 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 serial peripheral interface (SPI), UART, and bus control signals. These pins can be programmed as GPIO when the SPI, UART, and bus control features are not used. See Chapter 13, “Serial Peripheral Interface 1 and 2,” and Section 2.6, “Bus Control Signals,” on page 2-6 for more detailed information.

These bits control or report the data on the pins while the associated SELx bits are high. While the DIRx bits are high (output), the Dx bits control the pins. While 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 will 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.

10.4.6.3 Port E Dedicated I/O Functions

The eight PEDATA lines are multiplexed with the SPI and UART dedicated I/O signals whose assignments are shown in Table 10-28.

Table 10-28. Port E Dedicated Function Assignments

Bit	GPIO Function	Dedicated I/O Function
0	Data bit 0	SPITXD
1	Data bit 1	SPIRXD
2	Data bit 2	SPICLK2
3	Data bit 3	$\overline{\text{DWE}}$/$\overline{\text{UCLK}}$
4	Data bit 4	RXD1
5	Data bit 5	TXD1

Table 10-28. Port E Dedicated Function Assignments (Continued)

Bit	GPIO Function	Dedicated I/O Function
6	Data bit 6	$\overline{\text{RTS1}}$
7	Data bit 7	$\overline{\text{CTS1}}$

10.4.6.4 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 10-29.

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 10-29. Port E Pull-up Enable Register Description

Name	Description	Setting
PUx Bits 7–0	**Pull-up**—These bits enable the pull-up resistors on the port	0 = Pull-up resistors are disabled 1 = Pull-up resistors are enabled

10.4.6.5 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 10-30.

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 10-30. Port E Select Register Description

Name	Description	Setting
SELx Bits 7–0	**Select**—These bits select 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.

10.4.7 Port F Registers

Port F is composed 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)

Each signal in the PFDATA register connects to an external pin. As on the other ports, each bit on Port F is individually configured.

10.4.7.1 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 10-31.

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 10-31. Port F Direction Register Description

Name	Description	Setting
DIRx Bits 7–0	**Direction**—These bits control the direction of the pins in an 8-bit system. They reset to 0.	0 = Input 1 = Output

10.4.7.2 Port F Data Register

The settings for the bit positions of the PFDATA register are shown in Table 10-32.

PFDATA Port F Data Register **0x(FF)FFF429**

	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 10-32. Port F Data Register Description

Name	Description	Setting
Dx Bits 7–0	**Data**—These bits reflect 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.

These bits control or report the data on the pins while the associated SELx bits are high. While the DIRx bits are high (output), the Dx bits control the pins. While 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 will 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.

10.4.7.3 Port F Dedicated I/O Functions

The eight PFDATA lines are multiplexed with the dedicated I/O signals whose assignments are shown in Table 10-33.

Table 10-33. Port F Dedicated I/O Function Assignments

Bit	GPIO Function	Dedicated I/O Function
0	Data bit 0	LCONTRAST
1	Data bit 1	$\overline{\text{IRQ5}}$
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. Bit 1 can be programmed as $\overline{\text{IRQ5}}$, an external level 5 interrupt.

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 Section 4.2, “CGM Operational Overview,” on page 4-3 for more information about this signal.

Bit 7 is used for the chip-select signal $\overline{\text{CSA1}}$. See Section 6.2, “Chip-Select Operation,” on page 6-2 for detailed information.

10.4.7.4 Port F Pull-up/Pull-down Enable Register

The Port F pull-up/pull-down 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 10-34.

PFPUEN Port F Pull-up/Pull-down Enable Register **0x(FF)FFF42A**

	BIT 7	6	5	4	3	2	1	BIT 0
	PU7	PD6	PD5	PD4	PD3	PU2	PU1	PU0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
				0xFF				

Table 10-34. Port F Pull-up/Pull-down Enable Register Description

Name	Description	Setting
PU7 Bit 7	**Pull-up**—This bit enables the pull-up resistor on the port.	0 = Pull-up resistor is disabled 1 = Pull-up resistor is enabled
PDx Bits 6–3	**Pull-down**—These bits enable the pull-down resistors on the port.	0 = Pull-down resistors are disabled 1 = Pull-down resistors are enabled
PUx Bits 2–0	**Pull-up**—These bits enable the pull-up resistors on the port.	0 = Pull-up resistors are disabled 1 = Pull-up resistors are enabled

10.4.7.5 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 10-35.

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 10-35. Port F Select Register Description

Name	Description	Setting
SELx Bits 7–0	**Select**—These bits select 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.

10.4.8 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)

Each signal in the PGDATA register connects to an external pin. It should be noted that pins 6 and 7 are not connected to external pins. Port G provides a total of six pins, and each bit is individually configured.

10.4.8.1 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 10-36.

PGDIR Port G Direction Register **0x(FF)FFF430**

	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 10-36. Port G 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**—These bits control the direction of the pins in an 8-bit system. They reset to 0.	0 = Input 1 = Output

10.4.8.2 Port G Data Register

The settings for the bit positions of the PGDATA register are shown in Table 10-37 on page 10-29.

PGDATA Port G Data Register 0x(FF)FFF431

	BIT 7	6	5	4	3	2	1	BIT 0
			D5	D4	D3	D2	D1	D0
TYPE			rw	rw	rw	rw	rw	rw
	0	0	1	1	1	1	1	1
RESET	0x3F*							

*Actual bit value depends on external circuits connected to pin.

Table 10-37. Port G 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**—These bits reflect 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 while the associated SELx bits are high. While the DIRx bits are high (output), the Dx bits control the pins. While 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 will 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 10-36 on page 10-28 for information about setting the bits in the PGDIR register.

10.4.8.3 Port G Dedicated I/O Functions

The six PGDATA lines are multiplexed with the dedicated I/O signals whose assignments are shown in Table 10-38.

Table 10-38. 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
2	Data bit 2	$\overline{\text{EMUIRQ}}$
3	Data bit 3	$\overline{\text{HIZ}}$/P/$\overline{\text{D}}$
4	Data bit 4	$\overline{\text{EMUCS}}$
5	Data bit 5	$\overline{\text{EMUBRK}}$
6		
7		

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 MC68VZ328 microprocessor will latch the BUSW signal at the rising edge of the Reset signal. Its mode will determine the default bus width for $\overline{\text{CSA0}}$. Bit 1 is Address 0. After system reset, this signal defaults to A0.

Bit 3 is $\overline{\text{HIZ}}$/P/$\overline{\text{D}}$ (High Impedance or Program/Data). During system reset, a logic low of this input signal will put the MC68VZ328 into Hi-Z mode, in which all MC68VZ328 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 Chapter 16, "In-Circuit Emulation," for detailed information on their operation.

10.4.8.4 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.

10.4.8.5 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 10-39 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
			PU5	PU4	PU3	PU2	PU1	PU0
TYPE			rw	rw	rw	rw	rw	rw
RESET	0	0	1	1	1	1	0	1
	0x3D							

Table 10-39. Port G Pull-up 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**—These bits enable the pull-up resistors on the port.	0 = Pull-up resistors are disabled 1 = Pull-up resistors are enabled

10.4.8.6 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 10-40 on page 10-31 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
			SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
TYPE			rw	rw	rw	rw	rw	rw
RESET	0	0	0	0	1	0	0	0
				0x08				

Table 10-40. Port G 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**—These bits select 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.

10.4.9 Port J Registers

Port J is composed of the following four 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)

Each signal in the PJDATA register connects to an external pin. As on the other ports, each bit on Port J is individually configured.

10.4.9.1 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 10-41.

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 10-41. Port J Direction Register Description

Name	Description	Setting
DIRx Bits 7–0	**Direction**—These bits control the direction of the pins in an 8-bit system. They reset to 0.	0 = Input 1 = Output

10.4.9.2 Port J Data Register

The bit settings for the PJDATA register are shown in Table 10-42.

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 10-42. Port J Data Register Description

Name	Description	Setting
Dx Bits 7–0	**Data**—These bits reflect 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 SPI (with internal FIFO) and UART 2 signals. These pins can be programmed as GPIO when the dedicated I/O signals are not in use.

These bits control or report the data on the pins while the associated SELx bits are high. While the DIRx bits are high (output), the Dx bits control the pins. While 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 will 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.

10.4.9.3 Port J Dedicated I/O Functions

The eight PJDATA lines are multiplexed with the dedicated I/O signals whose assignments are shown in Table 10-43.

Table 10-43. 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	SPICLK1
3	Data bit 3	$\overline{\text{SS}}$
4	Data bit 4	RXD2
5	Data bit 5	TXD2
6	Data bit 6	$\overline{\text{RTS2}}$

Table 10-43. Port J Dedicated I/O Function Assignments (Continued)

Bit	GPIO Function	Dedicated I/O Function
7	Data bit 7	$\overline{\text{CTS2}}$

Bits 0–3 are control signals connected to SPI 1. Their operation is detailed in Section 13.2.4, “SPI 1 Signals,” on page 13-3. The remaining 4 bits are control signals for UART 2; more information appears in Section 14.2.3, “Serial Interface Signals,” on page 14-3.

10.4.9.4 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 10-44.

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 10-44. Port J Pull-up Enable Register Description

Name	Description	Setting
PUx Bits 7–0	**Pull-up**—These bits enable the pull-up resistors on the port	0 = Pull-up resistors are disabled 1 = Pull-up resistors are enabled

10.4.9.5 Port J Select Register

The 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 10-45.

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
				0xEF				

Table 10-45. Port J Select Register Description

Name	Description	Setting
SELx Bits 7–0	**Select**—These bits select 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.

10.4.10 Port K Registers

Port K is composed of the following 8-bit general-purpose I/O registers:

- Port K direction register (PKDIR)
- Port K data register (PKDATA)
- Port K pull-up/-down enable register (PKPUEN)
- Port K select register (PKSEL)

Each signal in the PKDATA register connects to an external pin. As on the other ports, each bit on Port K is individually configured.

10.4.10.1 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 10-46.

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 10-46. Port K Direction Register Description

Name	Description	Setting
DIRx Bits 7–0	**Direction**—These bits control the direction of the pins in an 8-bit system. They reset to 0.	0 = The pins are inputs. 1 = The pins are outputs.

10.4.10.2 Port K Data Register

The settings for the PKDATA register bit positions are shown in Table 10-47 on page 10-35.

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
	0	0	0	0	1	1	1	1
RESET	0x0F*							

*Actual bit value depends on external circuits connected to pin.

Table 10-47. Port K Data Register Description

Name	Description	Setting
Dx Bits 7–0	**Data**—These bits reflect 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 K is multiplexed with the IrDA, SPI, and LCD controller signals. These pins can be programmed as GPIO when the dedicated I/O signals are not in use.

These bits control or report the data on the pins while the associated SELx bits are high. While the DIRx bits are high (output), the Dx bits control the pins. While 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 will 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.

10.4.10.3 Port K Dedicated I/O Functions

The eight PKDATA lines are multiplexed with the dedicated I/O signals whose assignments are shown in Table 10-48.

Table 10-48. Port K Dedicated I/O Function Assignments

Bit	GPIO Function	Dedicated I/O Function
0	Data bit 0	$\overline{\text{DATA\_READY}}$/PWM2
1	Data bit 1	R$\overline{\text{W}}$
2	Data bit 2	$\overline{\text{LDS}}$
3	Data bit 3	$\overline{\text{UDS}}$
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 SPI 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.

When selected bit 1 (R$\overline{\text{W}}$) is connected to the 68000 CPU Read/Write signal, this pin defaults to Port K bit 1, GPIO input, pulled high.

The remaining bits are involved with bus control. See Section 2.6, “Bus Control Signals,” on page 2-6 for more detailed information.

10.4.10.4 Port K Pull-up/Pull-down Enable Register

The pull-up/pull-down enable register (PKPUEN) controls the pull-up and the pull-down resistors for each line in Port K. The settings for the PKPUEN register bit positions are shown in Table 10-49.

PKPUEN Port K Pull-up/Pull-down Enable Register **0x(FF)FFF442**

	BIT 7	6	5	4	3	2	1	BIT 0
	PD7	PD6	PD5	PD4	PU3	PU2	PU1	PU0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw
RESET	1	1	1	1	1	1	1	1
				0xFF				

Table 10-49. Port K Pull-up/Pull-down Enable Register Description

Name	Description	Setting
PUx Bits 7–0	**Pull-up/Pull-down Enable**—These bits enable 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

10.4.10.5 Port K Select Register

The 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 10-50.

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	1	1	1	1	1	1	1	1
				0xFF				

Table 10-50. Port K Select Register Description

Name	Description	Setting
SELx Bits 7–0	**Select**—These bits select 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.

10.4.11 Port M Registers

Port M is composed of the following four 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)

Each signal in the PMDATA register connects to an external pin. It should be noted that pins 6 and 7 are not connected to external pins.

10.4.11.1 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 10-51.

PMDIR Port M Direction Register **0x(FF)FFF448**

	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 10-51. Port M 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**—These bits control the direction of the pins in an 8-bit system. They reset to 0.	0 = The pins are inputs. 1 = The pins are outputs.

10.4.11.2 Port M Data Register

The settings for the PMDATA register bit positions are shown in Table 10-52.

PMDATA Port M Data Register **0x(FF)FFF449**

	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 10-52. Port M 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**—These bits reflect 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 M is multiplexed with the SDRAM controller signals. These pins can be programmed as GPIO when the SDRAM I/O signals are not in use.

These bits control or report the data on the pins while the associated SELx bits are high. While the DIRx bits are high (output), the Dx bits control the pins. While 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 will 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.

10.4.11.3 Port M Dedicated I/O Functions

The six PMDATA lines are multiplexed with the dedicated I/O signals whose assignments are shown in Table 10-53.

Table 10-53. Port M Dedicated I/O Function Assignments

Bit	GPIO Function	Dedicated I/O Function
0	Data bit 0	SDCLK
1	Data bit 1	SDCE
2	Data bit 2	DQMH
3	Data bit 3	DQML
4	Data bit 4	SDA10
5	Data bit 5	$\overline{\text{DMOE}}$
6		
7		

All of the dedicated I/O functions are involved in the operation of the DRAM controller. See Chapter 7, “DRAM Controller,” for more details.

10.4.11.4 Port M Pull-up/Pull-down Enable Register

The pull-up/pull-down enable register (PMPUEN) controls the pull-up and pull-down resistors for each line in Port M. The settings for the PMPUEN register bit positions are shown in Table 10-54.

PMPUEN Port M Pull-up/Pull-down Enable Register **0x(FF)FFF44A**

	BIT 7	6	5	4	3	2	1	BIT 0
			PU5	PD4	PD3	PD2	PD1	PD0
TYPE			rw	rw	rw	rw	rw	rw
RESET	0	0	1	1	1	1	1	1
				0x3F				

Table 10-54. Port M 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**—These bits enable 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

10.4.11.5 Port M Select Register

The 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 10-55.

PMSEL Port M Select Register **0x(FF)FFF44B**

	BIT 7	6	5	4	3	2	1	BIT 0
			SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
TYPE			rw	rw	rw	rw	rw	rw
RESET	0	0	1	1	1	1	1	1
				0x3F				

Table 10-55. Port M 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**—These bits select 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.

Chapter 11
Real-Time Clock

This chapter describes the real-time clock (RTC) module, which is composed of six blocks as shown in Figure 11-1: the 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. The RTC module can generate three different level 4 interrupts to the interrupt controller. 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 MC68VZ328.

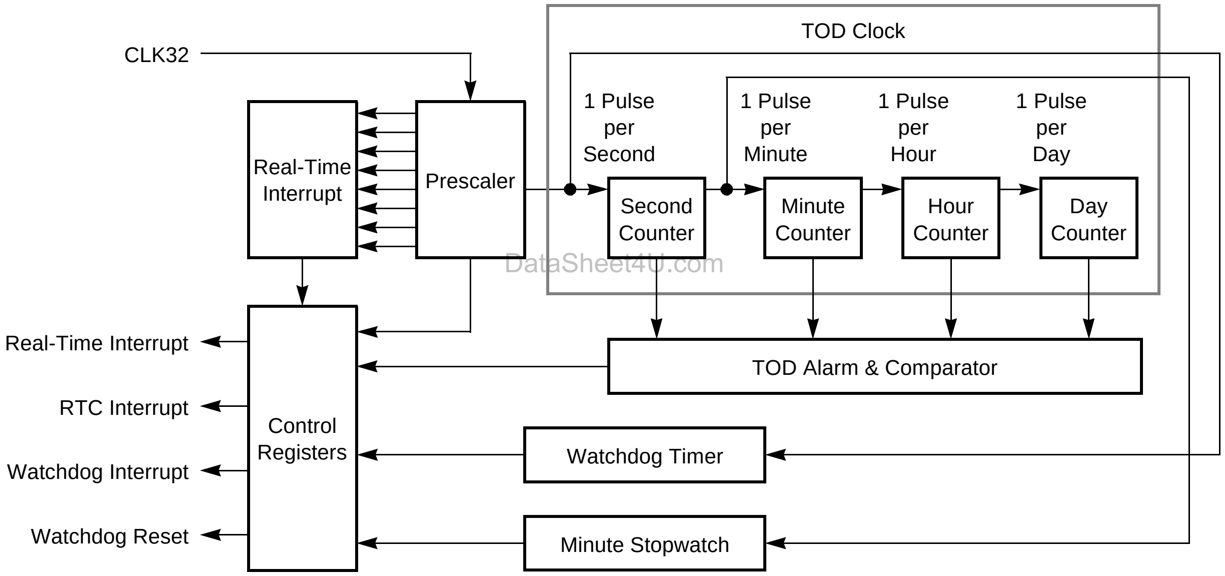


Figure 11-1. Real-Time Clock Module Simplified Block Diagram

11.1 RTC Overview

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 11-1 on page 11-2. 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. If 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 level 4 interrupts 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 11-1.

Table 11-1. RTC Interrupt Mapping

Internal Name	Interrupt Controller	Resolution
Real-time interrupt	Real-time interrupt	Eight different rates
Stopwatch	Real-time clock	Minutes
1HZ	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.

11.1.1 Prescaler

The prescaler divides the CLK32 reference clock down to 1 pulse per second, resulting in a signal labeled 1HZ. After an initial power up, the CLK32 signal is always available, even when the unit is in a reduced power mode. See Section 4.3.1, “CLK32 Clock Signal,” on page 4-4 for more information about the CLK32; see Section 4.5, “Introduction to the Power Control Module,” on page 4-10 for detailed information on the power modes of the MC68VZ328. The actual frequency of the CLK32 is determined by the external crystal used as the crystal oscillator. The MC68VZ328 supports either a 32.768 kHz or a 38.4 kHz frequency crystal.

NOTE:

If a 38.4 kHz crystal is used as the crystal oscillator, the REFREQ bit in the real-time control register (RTCCTL) must be set. Failure to set this bit will make the RTC timing incorrect.

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.

11.1.2 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 in day counts.

NOTE:

> To allow maximum flexibility in design, each of the four counters in the TOD clock can accept values that exceed their valid range. The MC68VZ328 does not check for range validity. If an out-of-range value is entered, the counter will reset to zero the next time it is incremented. For example, if 26 is written to the hours counter, the counter will remain 26 until incremented by the minutes counters. When incremented, the hours counter will return to zero. It is the responsibility of the user to ensure the range validity of data in the TOD clock.

Each of the four counters may be enabled to produce an interrupt when it rolls over. Upon reaching 59, the seconds and minutes counters each produce an MIN or HR interrupt (if 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.

11.1.3 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 AL 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.

11.1.4 Watchdog Timer

The watchdog timer is an added check that a program is running and sequencing properly. When the 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.

11.1.5 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 11-9 on page 11-12. 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 a level 4 interrupt to the MC68VZ328 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.

11.1.6 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.

11.1.6.1 Minute Stopwatch Application Example

The minute stopwatch can be used to turn off the LCD controller after 5 minutes of inactivity. To accomplish this, the minute stopwatch is programmed with a value of 5 minutes, and then the stopwatch interrupt (SW bit) in the RTCIENR is enabled. At consecutive minute increments, the minute stopwatch value is decremented. An SW interrupt is generated when the counter counts to -1. The stopwatch interrupt (SW bit) in the RTCISR occurs after 5 minutes. In addition to the 5 minutes of the stopwatch, there is an unknown number of seconds from the time the stopwatch is set until the first minute.

11.2 Programming Model

Section 11.2.1, “RTC Time Register,” through Section 11.2.9, “Stopwatch Minutes Register,” provide programming information on the real-time clock.

11.2.1 RTC Time 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 11-2.

RTCTIME RTC Hours, Minutes, and Seconds 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	X	X	X	X	X	0	0	X	X	X	X	X	X
	0xXXXX															

	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	X	X	X	X	X	X
	0x00XX															

Table 11-2. RTC Hours, Minutes, and Seconds 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.
Reserved Bits 15–6	Reserved	These bits are reserved and should be set to 0.
SECONDS Bit 5–0	**Seconds**—These bits indicate the current second.	The bits can be set to any value between 0 and 59.

11.2.2 RTC Day Count Register

The real-time clock day count 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 11-3.

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	?	?	?	?	?	?	?	?	?
								0x0XXX								

Table 11-3. 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.

11.2.3 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 RTCTIME register are described in Table 11-4.

RTCALRM RTC Alarm Register **0x(ff)FFFB04**

	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	0	0	0	0	0	0	0	0	0	0	0

0x00000000

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

0x00000000

Table 11-4. 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**—These bits indicate 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**—These bits indicate 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 Bit 5–0	**Seconds**—These bits indicate 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.

11.2.4 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 11-5.

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	0
								0x0000								

Table 11-5. 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.

11.2.5 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 11-6.

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	0	1
	0x0001															

Table 11-6. 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**—These bits represent 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).
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).

11.2.6 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 11-7.

RTCCTL RTC Control Register **0x(ff)FFFB0C**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
									RTCEN		REFREQ					
TYPE									rw		rw					
RESET	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
									0x0080							

Table 11-7. 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**—This bit, when set, enables the real-time clock.	0 = Disable the real-time clock 1 = Enable the real-time clock (default)
Reserved Bit 6	Reserved	This bit is reserved and should be set to 0.
REFREQ Bit 5	**Reference Frequency**—This bit is set to the frequency of the crystal oscillator.	0 = Reference frequency is 32.768 kHz (default). 1 = Reference frequency is 38.4 kHz.
Reserved Bits 4–0	Reserved	These bits are reserved and should be set to 0.

11.2.7 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 11-8 on page 11-11. For more information about the frequency of the RTC interrupts, refer to Table 11-9 on page 11-12.

RTCISR RTC Interrupt Status Register 0x(ff)FFFB0E

	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 11-8. 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**—If 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**—If 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**—If 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 11-8. RTC Interrupt Status Register Description (Continued)

Name	Description	Setting
MIN Bit 1	**Minute Flag**—If 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**—If enabled, the stopwatch flag is set when the stopwatch minute count-down times out.	0 = The stopwatch did not time out. 1 = The stopwatch timed out.

Table 11-9. Real-Time Interrupt Frequency Settings

Real-Time Interrupt Frequency	32.768 kHz Reference Clock		38.4 kHz Reference Clock	
RFE7	512 Hz	1.9531 ms	600 Hz	1.6666 ms
RFE6	256 Hz	3.9062 ms	300 Hz	3.3333 ms
RFE5	128 Hz	7.8125 ms	150 Hz	6.6666 ms
RFE4	64 Hz	15.625 ms	75 Hz	13.3333 ms
RFE3	32 Hz	31.25 ms	37.5 Hz	26.6666 ms
RFE2	16 Hz	62.5 ms	18.75 Hz	53.3333 ms
RFE1	8 Hz	125 ms	9.375 Hz	106.6666 ms
RFE0	4 Hz	250 ms	4.6875 Hz	213.3333 ms

11.2.8 RTC Interrupt Enable Register

The RTC interrupt enable register (RTCIENR) is used to enable the interrupts in the RTCSIR if the corresponding bit is set. The settings for the RTCIENR register are described in Table 11-10 on page 11-13. For information about the frequency of the real-time interrupts, refer to Table 11-9.

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 11-10. 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 11-9 on page 11-12.	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 11-9 on page 11-12.	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 11-9 on page 11-12.	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 11-9 on page 11-12.	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 11-9 on page 11-12.	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 11-9 on page 11-12.	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 11-9 on page 11-12.	0 = RIE1 interrupt is disabled. 1 = RIE1 interrupt is enabled.
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 11-9 on page 11-12.	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.

Table 11-10. RTC Interrupt Enable Register Description (Continued)

Name	Description	Setting
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 = 1-minute interrupt is disabled. 1 = 1-minute interrupt is enabled.

11.2.9 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 11-11.

NOTE:

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**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
											CNT					
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 11-11. 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 non-zero value, which is less than 63 minutes, is written to this register.

Chapter 12
General-Purpose Timers

This chapter describes in detail the operation of the general-purpose timer modules of the MC68VZ328. The GP timers consist of two general-purpose 16-bit timers, a prescaler, 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 either the leading or trailing edges of an input pulse. The timer can also 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. The two timers may also be cascaded together to operate as a single 32-bit timer.

12.1 GP Timer Overview

The two 16-bit timers (Timer 1 and Timer 2) that make up the general-purpose timers are identical. Figure 12-1 illustrates the general-purpose timer block diagram. The following sections describe the operation of the GP timers in detail.

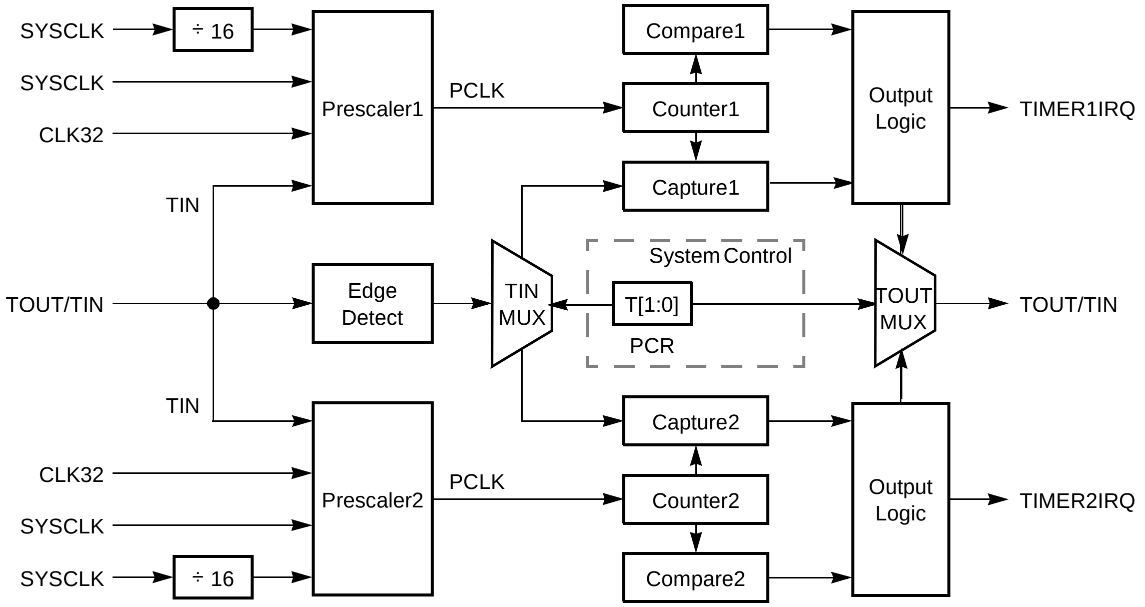


Figure 12-1. General-Purpose Timer Block Diagram

12.1.1 Clock Source and Prescaler

The clock source for each timer is individually selectable through software. The selected clock source is fed to a prescaler that acts as a divider with a programmable division ratio ranging from 1 to 256. The output of each prescaler drives its respective counter.

The clock sources are SYSCLK, SYSCLK/16, CLK32, and an external clock from the timer I/O pin (TIO). The clock input source is selected by the CLKSOURCE field of the timer control registers (TCTLx). The PRESCALER field of the timer prescaler register (TPRERx) selects the divide ratio of the input clock (PCLK) that drives the counter. The prescaler divides the input clock by a value between 1 and 256. The programmable prescaler allows a maximum period of 512 seconds when using a 32.768 kHz crystal oscillator or 436 seconds using a 38.4 kHz oscillator.

Of the four clock sources, only CLK32 continues to operate while the MC68VZ328 is in sleep mode. See Section 4.5.2, “CGM Operation During Sleep Mode,” on page 4-12 for more information on CLK32 operation during sleep mode.

NOTE:

Ensure that the timer is disabled by clearing the TEN bit in the TCTLx register before changing either the clock source or prescaler setting.

12.1.2 Timer Events and Modes of Operation

There are two types of events that produce interrupts: compare events and capture events. Compare events occur when the value in the counter matches the contents of the compare register. Capture events occur when a defined transition of the TOUT/TIN pin is detected.

The counter can be programmed to run in one of two modes: restart or free-running. The free-running/restart bit in the TCTLx register only controls how the counter operates after a compare event occurs. It does not affect counter operation following capture events. A description of each mode follows.

12.1.2.1 Restart Mode

In restart mode, the following actions occur when the compare value in the timer compare register (TCMPx) matches the value in the timer counter register (TCNx):

1. The counter resets to 0x0000.
2. The compare event (COMP) bit of the timer status register (TSTATx) is set.
3. The TIMERx interrupt is issued to the interrupt controller if the IRQEN bit of the TCTLx register is set.
4. The timer counter resumes counting.

This mode is useful when you need to generate periodic events or, when it is used with the timer output signals, audio tones.

12.1.2.2 Free-Running Mode

Free-running mode is similar in operation to restart mode, except that when a compare event occurs, the counter continues counting without resetting to 0x0000. When 0xFFFF is reached, the counter rolls over to 0x0000 and continues counting.

12.1.3 Timer Capture Register

Each timer has a 16-bit capture register that takes a “snapshot” of the timer counter when a defined transition of the signal applied to the TIN pin is detected by the capture edge detector. There are three transitions of the TIN that can trigger a capture event:

- Capture on rising edge
- Capture on falling edge
- Capture on rising or falling edge

The type of transition that triggers the capture is selected by the CAP field of the TCTLx register. Pulses that produce the capture edge can be as short as 20 ns. The minimum time between pulses is two PCLK periods.

When a capture event occurs, the CAPT status bit is set in the TSTATx register. A TIMERx interrupt is sent to the MC68VZ328 interrupt controller if the capture function is enabled and the IRQEN bit of the TCTLx register is set. The timer is disabled at reset.

12.1.4 TOUT/TIN/PB6 Pin

The TOUT/TIN pins are multiplexed with bit 6 of the Port B registers. The Port B registers determine if the pin is assigned to the GP timers or to pin 6 of Port B (the default setting), as described in Section 10.4.2.3, “Port B Dedicated I/O Functions,” on page 10-10. Because the TOUT/TIN/PB6 is a bidirectional pin, the direction of the pin is also controlled in the Port B registers.

NOTE:

Unlike other port register pins, the TOUT/TIN/PB6 pin direction is still controlled by the DIR6 bit in the Port B direction register even though the pin is assigned to the GP timers.

When the in direction is selected, the pin (TIN) is available as a clock input to the timer or as the input trigger to the edge-detect circuit for the capture registers. The T[1:0] field in the peripheral control register (PCR) switches the TIN input between capture register 1 and capture register 2. When T = 0x00 the TIN is connected to Timer 1, and when T = 0x01 the TIN is connected to Timer 2.

When the out direction is enabled, the pin (TOUT) is used to toggle or output a pulse when a timer compare event occurs.

12.1.5 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 5.2.2, “Peripheral Control Register,” on page 5-4 for more details.

Table 12-1 shows the two possible configurations of cascaded timers. When T[1:0] = 0x10, Timer 1 and Timer 2 are cascaded together. 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. Cascade 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

12.1.5.1 Compare and Capture Using Cascaded Timers

When the timers are cascaded, the associated compare and capture registers are not. The flow diagram in Figure 12-2 on page 12-5 suggests one method for 32-bit compares using a cascaded timer. Captures can also 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.

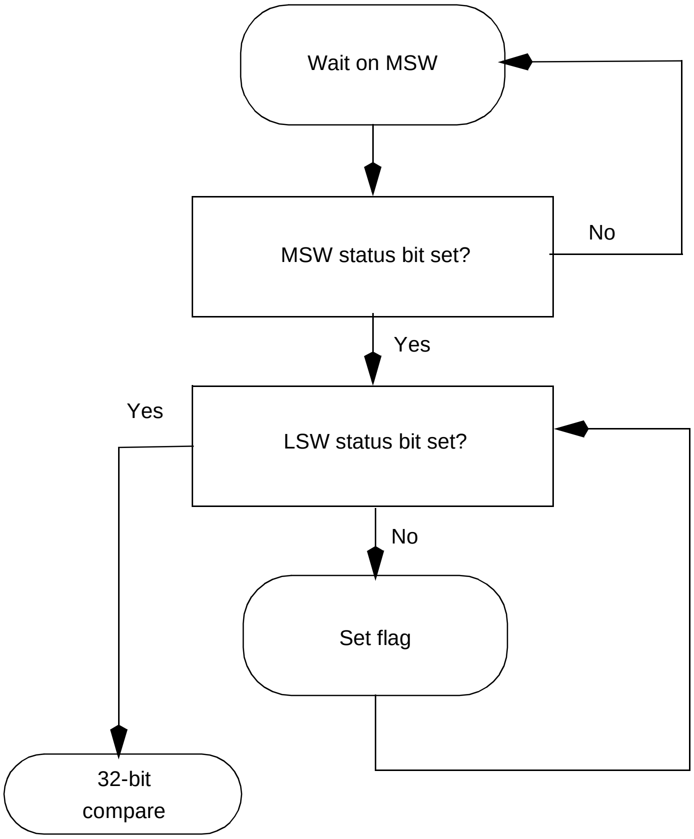


Figure 12-2. Compare Routine for 32-Bit Cascaded Timers

12.2 Programming Model

The following sections provide programming information about the settings of the two 16-bit timers in the GP timers module. Because the two timers are identical, the register description and the associated table describing the register settings apply to both registers.

12.2.1 Timer Control Registers 1 and 2

Each timer control (TCTLx) register controls the overall operation of its corresponding GP timer. The settings for the registers are described in Table 12-2. The TCTL registers control the following:

- Selecting the free-running or restart mode after a compare event
- Selecting the capture trigger event
- Controlling the output compare mode
- Enabling the compare event interrupt
- Selecting the prescaler clock source
- Enabling and disabling 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.

Table 12-2. Timer Control Register Description (Continued)

Name	Description	Setting
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. The output appears for one SYSCLK period.	0 = Active-low pulse (default). 1 = Toggle output.
IRQEN Bit 4	**Interrupt Request Enable**—This bit enables an interrupt on a compare event.	00 = Disable the compare interrupt (default). 01 = Enable the compare interrupt.
CLKSOURCE Bit 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.

12.2.2 Timer Prescaler Registers 1 and 2

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

12.2.3 Timer Compare Registers 1 and 2

Each timer compare (TCMPx) register 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 0x0000 to 0xFFFF.

12.2.4 Timer Capture Registers 1 and 2

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 0x0000 to 0xFFFF.

12.2.5 Timer Counter Registers 1 and 2

Each read-only timer counter (TCNx) register 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 0x0000 to 0xFFFF.

12.2.6 Timer Status Registers 1 and 2

Each timer status (TSTATx) register 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**—This status bit, when set, indicates that a capture event occurred.	0 = No capture event occurred. 1 = A capture event has occurred.
COMP Bit 0	**Compare Event**—This status bit, when set, indicates when a compare event occurs.	0 = No compare event occurred. 1 = A compare event has occurred.

Chapter 13
Serial Peripheral Interface 1 and 2

The MC68VZ328 contains two serial peripheral interface (SPI) modules, SPI 1 and SPI 2. This chapter describes the operation and programming of both SPI modules.

While SPI 2 operates as a master-mode-only SPI module, SPI 1 represents an enhanced version of the SPI 2 design. Equipped with a data FIFO, SPI 1 may operate as a master- or slave-configurable SPI interface module, allowing the MC68VZ328 to interface with either an external SPI master or an SPI slave device.

13.1 SPI 1 Overview

This section discusses how SPI 1 may be used to communicate with external devices. SPI 1 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 13-1 illustrates the configurable serial peripheral interface block diagram.

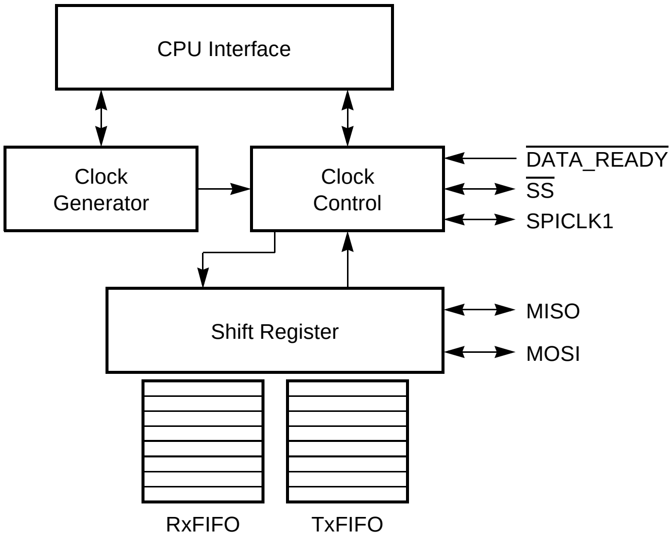


Figure 13-1. SPI 1 Block Diagram

13.2 SPI 1 Operation

The SPI 1 signal pins are multiplexed with bit 0 ($\overline{\text{DATA\_READY}}$) of the Port K register and bits 3–0 (MOSI, MISO, and SPICLK1) of the Port J register. Therefore, before SPI 1 is used, it is necessary to write 0 to these bits in the PKSEL and PJSEL registers, respectively. See Section 10.4.9.5, “Port J Select Register,” on page 10-33 and Section 10.4.10.5, “Port K Select Register,” on page 10-36 for detailed information.

13.2.1 Using SPI 1 as Master

If SPI 1 is configured as master, it uses a serial link to transfer data between the MC68VZ328 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, the SPI master’s output port can be ignored and used for other purposes. In order to utilize the internal TxD and RxD data FIFOs, two auxiliary output signals, $\overline{\text{SS}}$ and $\overline{\text{DATA\_READY}}$, are used for data transfer rate control. The user may also program the sample period control register to a fixed data transfer rate.

13.2.2 Using SPI 1 as Slave

If SPI 1 is configured as slave, the SPI 1 control register can be configured to match the external SPI master’s timing. $\overline{\text{SS}}$ 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 13-2 shows the generic SPI timing.

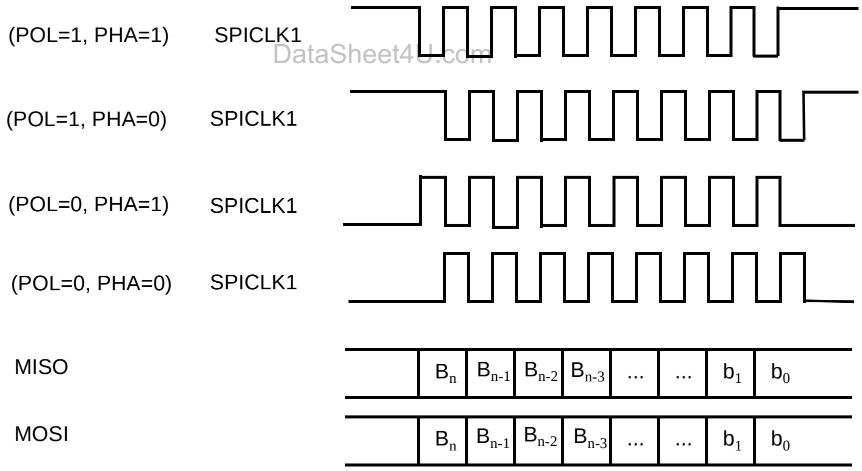


Figure 13-2. SPI 1 Generic Timing

NOTE:

SPI 1 does not consume any power when it is disabled.

13.2.3 SPI 1 Phase and Polarity Configurations

When SPI 1 is used as master, the SPICLK1 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 SPICLK1 edge. The polarity of SPICLK1 may be configured (to invert the SPICLK1 signal), but it does not change the edge-triggered events that are internal to the SPI 1. This flexibility allows it to operate with most serial peripheral devices available in the marketplace.

13.2.4 SPI 1 Signals

The following signals are used to control SPI 1:

- MOSI—Master Out/Slave In bidirectional signal, which is multiplexed with PJ0, 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, which is multiplexed with PJ1, 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.
- SPICLK1—SPI Clock bidirectional signal, which is multiplexed with PJ2, is the SPI clock output in master mode. In slave mode it is the input SPI clock signal.
- $\overline{\text{SS}}$—Slave Select bidirectional signal, which is multiplexed with PJ3, is output in master mode and input in slave mode.
- $\overline{\text{DATA\_READY}}$—SPI 1 Data Ready input signal is used only in master mode. It is multiplexed with PK0 and will edge- or level-trigger an SPI burst if used.

13.3 SPI 1 Programming Model

This section provides information for programming SPI 1.

13.3.1 SPI 1 Receive Data Register

This read-only register holds the top of the 8 × 16 RxFIFO, which receives data from an external SPI device during 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 13-1.

SPIRXD SPI 1 Receive Data Register **0x(FF)FFF700**

	BIT 7	6	5	4	3	2	1	BIT 0
	DATA							
TYPE	r	r	r	r	r	r	r	r
RESET	0	0	0	0	0	0	0	0
	0x0000							

Table 13-1. SPI 1 Receive Data Register Description

Name	Description	Setting
DATA Bits 7–0	**Data**—Top of SPI 1's RxFIFO (8 × 16)	The data in this register has no meaning if the RR bit in the interrupt control/status register is cleared.

13.3.2 SPI 1 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 XCH bit is set. For example, a user may write to TxFIFO during the SPI data exchange process. In either master or slave mode, a maximum of 8 data words are loaded. Data written to this register can be of 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 SPI 1 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 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, zeros are shifted out serially as the TxD signal. Writes to this register are ignored while the SPIEN bit in the SPI 1 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 13-2.

SPITXD SPI 1 Transmit Data Register **0x(FF)FFF702**

	BIT 7	6	5	4	3	2	1	BIT 0
	DATA							
TYPE	w	w	w	w	w	w	w	w
RESET	0	0	0	0	0	0	0	0
	0x00							

Table 13-2. SPI 1 Transmit Data Register Description

Name	Description	Setting
DATA Bits 7–0	**Data**—Top SPI data to be loaded to the 8 × 16 TxFIFO	See description

13.3.3 SPI 1 Control/Status Register

This register controls the configuration and operation of the SPI 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 13-3.

SPICONT1 SPI 1 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 13-3. SPI 1 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 SPI 1 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	**DATA_READY Control**—In master mode, these 2 bits select the waveform of the DATA_READY input signal. In slave mode, they have no effect.	00 = Don't care DATA_READY 01 = Falling edge trigger input 10 = Active low level trigger input 11 = RSV
MODE Bit 10	**SPI 1 Mode Select**—This bit selects the mode of SPI 1.	0 = SPI 1 is slave mode 1 = SPI 1 is master mode
SPIEN Bit 9	**SPI 1 Enable**—This bit enables SPI 1. This bit 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 SPI 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	**SS Polarity Select**—In both master and slave modes, this bit selects the polarity of SS signal.	0 = Active low 1 = Active high

Table 13-3. SPI 1 Control/Status Register Description (Continued)

Name	Description	Setting
SSCTL Bit 6	**$\overline{\text{SS}}$ Waveform Select**—In master mode, this bit selects the output wave form for the $\overline{\text{SS}}$ signal. In slave mode, this bit controls RxFIFO advancement.	Master Mode: 0 = $\overline{\text{SS}}$ stays low between SPI 1 bursts 1 = Insert pulse between SPI 1 bursts Slave Mode: 0 = RxFIFO advanced by Bit Count 1 = RxFIFO advanced by $\overline{\text{SS}}$ 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

13.3.4 SPI 1 Interrupt Control/Status Register

This register is used to provide interrupt control and status of various operations in SPI 1. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 13-4.

SPIINTCS SPI 1 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 13-4. SPI 1 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 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 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 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 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 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 13-4. SPI 1 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 TE bit is set.	0 = Disable TxFIFO empty interrupt. 1 = Enable TxFIFO empty interrupt.
BO Bit 7	**Bit Count Overflow**—This bit is set when the SPI is in "slave SPI 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 four 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.

13.3.5 SPI 1 Test Register

The configurable SPI test (SPITEST) register indicates the state machine status of SPI 1 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 13-5.

SPITEST SPI 1 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 13-5. SPI 1 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.

13.3.6 SPI 1 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 (SPICLK1) 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 13-6 on page 13-11.

SPISPC SPI 1 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 13-6. SPI 1 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 = SPICLK1 clock 1 = CLK32 (32.68 kHz normal crystal used)
WAIT Bits 14–0	**Wait**—Number of clock periods inserted between data transactions in master mode	0000 = 0 clocks 0001 = 1 clock 0002 = 2 clocks . . . 7FFF = 32767 clocks (approximately 1 second)

13.4 SPI 2 Overview

This section discusses how SPI 2 can be used to communicate with external devices, such as EEPROMs, analog-to-digital converters, and other peripherals. The SPI 2 module is a 3- or 4-wire system, depending on whether you are using unidirectional or bidirectional communication mode. It provides the clock for data transfer and can only function as a master device. It is fully compatible with the serial peripheral interface on Motorola’s 68HC05 and 68HC11 microprocessors. Figure 13-3 shows the SPI 2 block diagram.

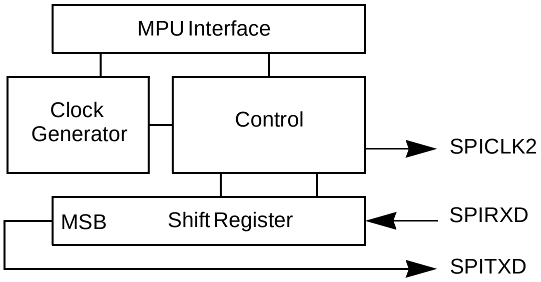


Figure 13-3. SPI 2 Block Diagram

13.5 SPI 2 Operation

The serial peripheral interface 2 operates as a master-mode-only SPI module using a serial link to transfer data between the MC68VZ328 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, SPI 2's output port is freed to be used for other purposes. See Figure 13-4.

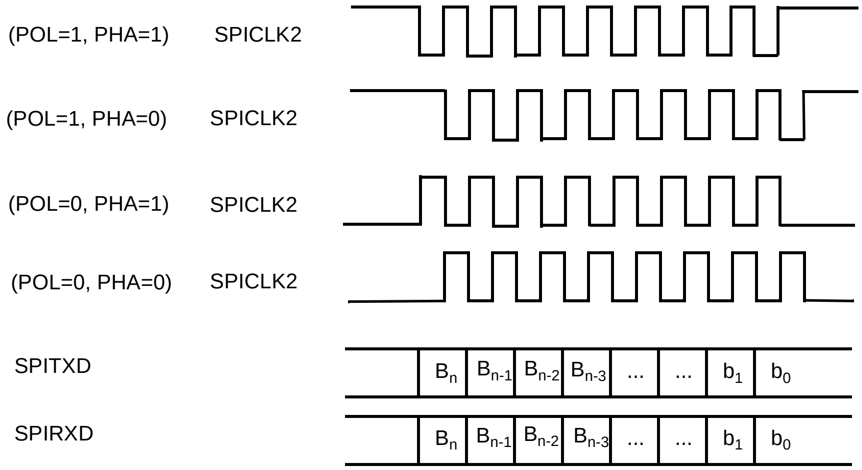


Figure 13-4. SPI 2 Generic Timing

The SPI 2 pins are multiplexed with bits 2–0 of the Port E registers, so when you use SPI 2, you must write 000 to these bits in the PESEL register. See Section 10.4.6, "Port E Registers," on page 10-21 for more information.

NOTE:

The SPI 2 module does not consume any power when it is disabled.

You must enable the ENABLE bit in the SPICONT2 register before you can change any other bits. To perform a serial data transfer, set the ENABLE bit; then, in a separate write cycle, set the appropriate control bits. The SPI 2 module is then ready to accept data into the SPIDATA2 register, which cannot be written while the SPI 2 module is disabled or busy. Once the data is loaded, the XCH bit is set in the SPICONT2 register, which triggers an exchange. The XCH bit remains set until the transfer is complete. If you clear the MSPI bit in the interrupt mask register before you trigger an exchange, an interrupt will be posted when the exchange is complete. See Section 9.6.3, "Interrupt Mask Register," on page 9-10 for more information. You can discover the status of the interrupt in the IRQ bit of the SPICONT2 register, and you can clear this bit by writing a 0 to it.

For systems that need more than 16 clocks to transfer data, the ENABLE bit can remain asserted between exchanges. The enable signal required by some SPI slave devices should be provided by an I/O port pin.

13.5.1 SPI 2 Phase and Polarity Configurations

The SPI 2 module uses the SPICLK2 signal to transfer data in and out of the shift register. Data is clocked using any one of four programmable clock phase and polarity variations. In 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 SPICLK2 edge. Polarity inverts SPICLK2, but does not change the edge-triggered events that are internal to the SPI 2 module. This flexibility allows it to operate with most serial peripheral devices on the market.

13.5.2 SPI 2 Signals

The following signals are used to control the SPI 2 module:

- SPITXD—The Transmit Data pin, which is multiplexed with PE0, is the output of the shift register. A new data bit is presented, but it depends on whether you have selected phase or polarity.
- SPIRXD—The Receive Data pin, which is multiplexed with PE1, is the input to the shift register. A new bit is shifted in, but it depends on whether you have selected phase or polarity.
- SPICLK2—The SPI 2 master Clock output pin is multiplexed with PE2. When the SPI 2 module is triggered, the selected number of clock pulses are issued. In polarity 0 mode, this signal is low while the SPI 2 module is idle, and it is high in polarity 1 mode.

NOTE:

A chip-select signal may be required by the external device. A GPIO pin may be assigned to this function.

13.6 SPI 2 Programming Model

This section provides information for programming SPI 2.

13.6.1 SPI 2 Data Register

The SPI 2 data (SPIDATA2) register exchanges data with external slave devices. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 13-7.

SPIDATA2 SPI 2 Data Register **0x(FF)FFF800**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	DATA															
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	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 13-7. SPI 2 Data Register Description

Name	Description	Setting
DATA Bits 15–0	**Data**—Top of SPI 2's RxFIFO (8 × 16)	The data in this register has no meaning if the RR bit in the interrupt control/status register is clear.

13.6.2 SPI 2 Data Register Timing

The data bits are exchanged with the external device. The data must be loaded before the XCH bit in the SPICONT2 register is set. In phase 0, data is presented on the SPITXD pin when this register is written. In phase 1, the first data bit is presented on the first SPICLK2 edge. At the end of the exchange, data from the peripheral is present in this register and bit 0 is the least significant bit. As data is shifted MSB first, outgoing data is automatically MSB justified. For example, if the exchange length is 10 bits, the first bit presented to the external device will be bit 9, followed by the remaining bits.

NOTE:

Writes to this field are ignored while the ENABLE bit is clear or while the XCH bit is set. This field contains unknown data if it is read while the XCH bit is set.

13.6.3 SPI 2 Control/Status Register

The SPI 2 control/status (SPICONT2) register controls how the SPI 2 module operates and reports its status. The bit position assignments for this register are shown in the following register display. The settings for this register are described in Table 13-8.

SPICONT2 SPI 2 Control/Status Register **0x(FF)FFF802**

	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	DATA RATE						ENABLE	XCH	IRQ	IRQEN	PHA	POL	BIT COUNT			
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 13-8. SPI 2 Control/Status Register Description

Name	Description	Setting
DATA RATE Bits 15–13	**Data Rate**—This field selects the bit rate of the SPICLK2 signal based on the division of the system clock. The master clock for the SPI 2 module 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.
Reserved Bits 12–10	Reserved	These bits are reserved and should be set to 0.
ENABLE Bit 9	**Enable**—This bit enables the SPI 2 module. This bit must be asserted before initiating an exchange and should be deasserted after the exchange is complete.	0 = The SPI 2 module is disabled. 1 = The SPI 2 module is enabled.
XCH Bit 8	**Exchange**—This bit triggers a data exchange and remains set while the exchange is in progress. During the busy period, the SPIDATA2 register cannot be written.	0 = Idle. 1 = Initiate an exchange (write) or busy (read).
IRQ Bit 7	**Interrupt Request**—This bit is set when an exchange is finished. If the IRQEN bit is set, an interrupt is generated. The MSPI bit of the interrupt mask register must be cleared for the interrupt to be posted to the core. See Section 9.6.3, “Interrupt Mask Register,” on page 9-10 for more information. This bit remains asserted until it is cleared by writing a 0. You can write a 1 to this bit to generate an interrupt request for system debugging.	0 = An exchange is in progress or idle. 1 = The exchange is complete.

Table 13-8. SPI 2 Control/Status Register Description (Continued)

Name	Description	Setting
IRQEN Bit 6	**Interrupt Request Enable**—This bit enables an interrupt to be generated when an SPI 2 module exchange is finished. This bit does not affect the operation of the IRQ bit; it only affects the interrupt signal to the interrupt controller.	0 = Disable interrupt generation. 1 = Allow interrupt generation.
PHA Bit 5	**Phase**—This bit controls the clock and 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 the 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 the 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.

Chapter 14
Universal Asynchronous Receiver/Transmitter 1 and 2

This chapter describes both UARTs in the DragonBall VZ integrated processor. The two UART ports in the MC68VZ328 may be used to communicate with external serial devices. UART 1 in the DragonBall VZ processor is identical to the UART in the DragonBall EZ processor, while UART 2 represents an enhanced version of UART 1. One of the enhancements in the UART 2 design is an enlarged RxFIFO and TxFIFO to reduce the number of software interrupts. An improvement to both UARTs is the system clock input frequency, which is 33.16 MHz, doubling the 16.58 MHz frequency of the MC68EZ328. For the 33.16 MHz system clock, software written for the MC68EZ328 version of the chip is not compatible unless the divider and prescaler are adjusted accordingly to compensate for the increased clock speed.

Because the two UART modules are nearly identical, the signal nomenclature throughout this chapter uses an *x* suffix to represent either 1 or 2. For example, TXDx represents either TXD1 or TXD2 depending on which UART is being used.

14.1 Introduction to the UARTs

This section describes how data is transported in character blocks using the standard “start-stop” format. It also discusses how to configure and program the UART modules, which have the following features:

- Full-duplex operation
- Flexible 5-wire serial interface
- Direct “glueless” support of IrDA physical layer protocol
- Robust receiver data sampling with noise filtering
- 12-byte FIFO for receive, 8-byte FIFO for transmit (UART 1)
- “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 230.4 kbps with 16x sample clock
- External 1x clock for high-speed synchronous communication
- Eight maskable interrupts
- Low-power idle model

The UART 2 module is an enhanced version of the UART 1. The features listed above are enhanced by the following modifications in the UART 2 module:

- The size of the RxFIFO and TxFIFO is increased to 64 bytes each.
- Both the RxFIFO and TxFIFO half mark levels are user selectable.
- The RTS signal can be triggered by either a near RxFIFO full condition or at the level defined by the RxFIFO level marker, rather than the RxFIFO half-full bit as is UART 1.

Both the UART 1 and UART 2 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 those applications that need other bit rates, a 1x clock mode is available providing a data-bit clock. Figure 14-1 illustrates a high-level block diagram of both UART modules.

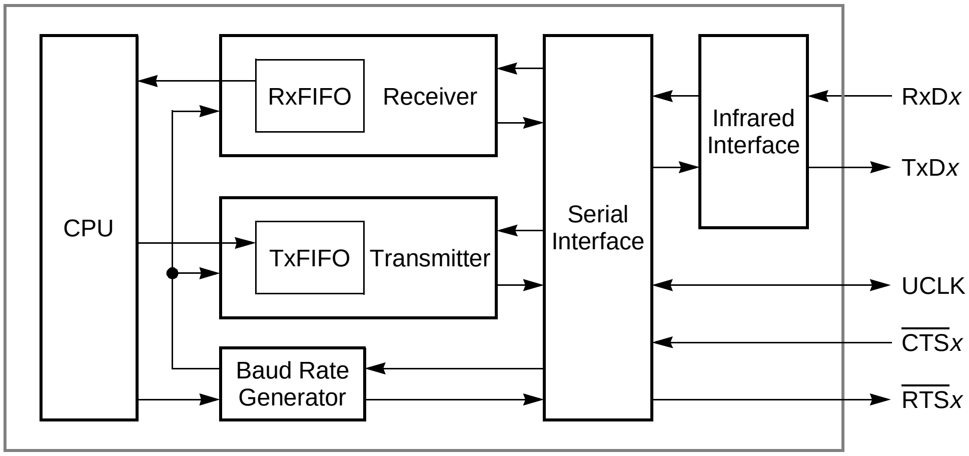


Figure 14-1. UART Simplified Block Diagram

14.2 Serial Operation

The UART modules have two modes of operation—NRZ and IrDA. Section 14.2.1, “NRZ Mode,” and Section 14.2.2, “IrDA Mode,” describe these two modes of operation.

14.2.1 NRZ Mode

The nonreturn to zero (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 14-2 on page 14-3 illustrates a character in NRZ mode.

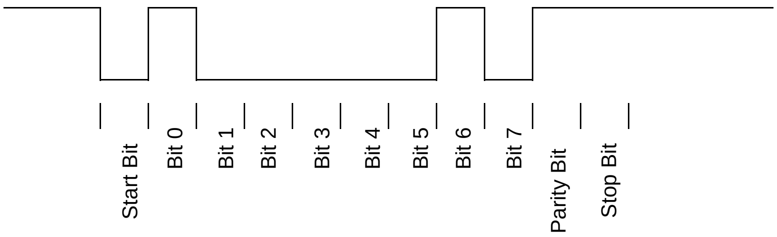


Figure 14-2. NRZ ASCII “A” Character with Odd Parity

14.2.2 IrDA Mode

Infrared (IrDA) mode uses character frames as NRZ mode does, but, 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 uses active low pulses. Figure 14-3 illustrates a character in IrDA mode.

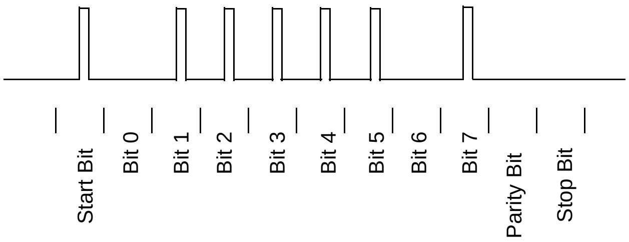


Figure 14-3. IrDA ASCII “A” Character with Odd Parity

14.2.3 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, which is 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. While 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, which is a configurable narrow pulse, is output for each zero bit that is transmitted.
- $\overline{\text{CTS1}}$/$\overline{\text{CTS2}}$—The Clear to Send signal, which is 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 while 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) register.

NOTE:

> If the NOCTSx bit of the UTX register is set, the transmitter sends a character whenever a character is ready to be transmitted. The $\overline{\text{CTSx}}$ 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, which is multiplexed with PE4 (PJ4 in UART 2), is the receiver serial input. As for the TXDx pin, while 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, which is 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 serve 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 5.2.2, "Peripheral Control Register," on page 5-4 for more details.

14.3 UART Operation

Both UART modules consist of three sub-blocks:

- Transmitter
- Receiver
- Baud rate generator

Section 14.3.1, "Transmitter Operation," through Section 14.3.3, "Baud Rate Generator Operation," discuss these sub-blocks in detail.

14.3.1 Transmitter Operation

The transmitter accepts a character (byte) from the CPU bus and transmits it serially. While 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.

14.3.1.1 TxFIFO 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 8-byte FIFO (64-byte FIFO in UART 2), the FIFO EMPTY interrupt should be enabled. The interrupt service routine should load data until the TX AVAIL bit in the UTX 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, the transmitter generates an interrupt when the FIFO has fewer than 4 bytes remaining. Because UART 2 has a larger FIFO buffer, 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 while the TX AVAIL bit is clear is ignored.

14.3.1.2 $\overline{\text{CTS}}$ Signal Operation

$\overline{\text{CTSx}}$ is used for hardware flow control. If $\overline{\text{CTSx}}$ is negated (high), the transmitter finishes sending the character in progress (if any) and then waits for $\overline{\text{CTSx}}$ to become asserted (low) again before starting the next character. The current state of the $\overline{\text{CTSx}}$ pin is sampled by the bit clock and can be monitored by reading the CTSx STAT bit of the UTX register. An interrupt can be generated when the $\overline{\text{CTSx}}$ pin changes state. The CTSx DELTA bit of the UTX register goes high when the $\overline{\text{CTSx}}$ pin toggles. For applications that do not need hardware flow control, such as IrDA, the NOCTSx bit of the UTX register should be set. While 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 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 register is set.
2. Wait until the BUSY bit goes low.
3. Clear the TXEN bit in the USTCNT register, which flushes the FIFO.
4. Wait until the BUSY bit goes low.
5. Set the TXEN bit.
6. Set the SEND BREAK bit in the UTX 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 should be empty and the transmitter should be idle and waiting for the next character.

If the TXEN bit of the USTCNT register is negated while a character is being transmitted, the character will be completed before the transmitter returns to IDLE. The transmit FIFO is immediately flushed when the TXEN bit is cleared. When the message has been completely sent and the UART is to be disabled, monitor the BUSY bit to determine when the transmitter has actually completed 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.

14.3.2 Receiver Operation

The receiver block of the UART accepts a serial data stream, converting 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, RXDx 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 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 register are valid only when read as a 16-bit word with the received character byte.

14.3.2.1 Rx FIFO Buffer Operation

As with the transmitter, the receiver FIFO is flexible. If the software being used has short interrupt latency time, the FIFO FULL interrupt in the URX register can be enabled. 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 can then be 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 4 empty bytes remain in the FIFO. If the 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 at a minimum) pulses for each zero bit received. Otherwise, normal NRZ is expected. An infrared transceiver directly connected to the RXDx pin transforms the infrared signal into an electrical signal. Polarity is programmable so that RXDx can be connected directly to an external IrDA transceiver.

14.3.3 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 14-4 on page 14-7 illustrates a block diagram of the baud rate generator.

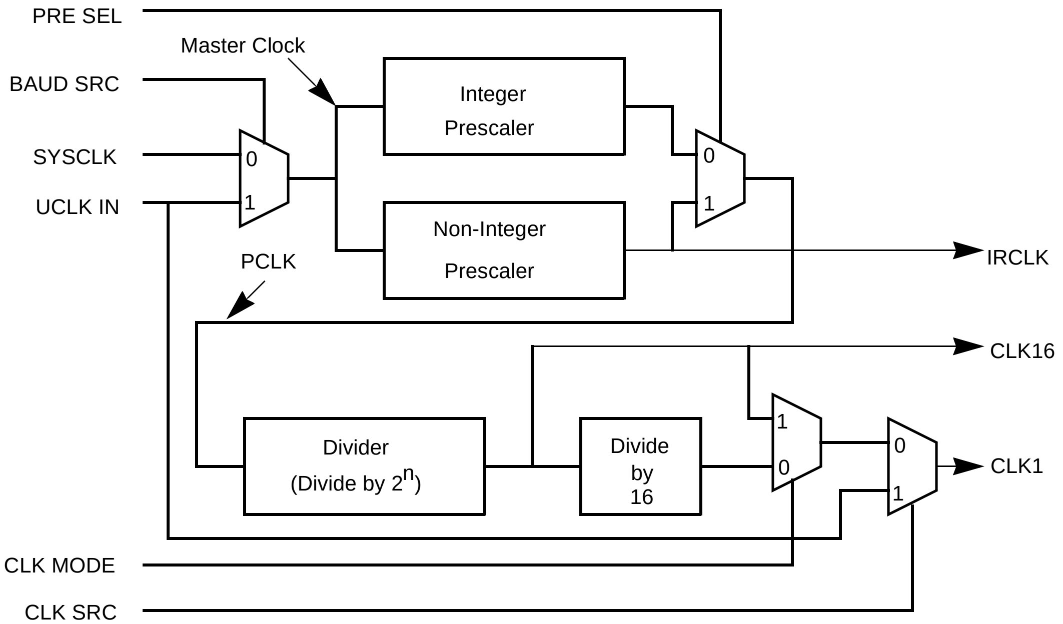


Figure 14-4. Baud Rate Generator Block Diagram

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.

14.3.3.1 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. While the CLKM bit of the USTCNT register is high, CLK1 is directly sourced by the CLK16 signal.

14.3.3.2 Non-Integer Prescaler

The non-integer prescaler is used to generate special, nonstandard baud frequencies. When IrDA mode is enabled, zeros are transmitted as three-sixteenth bit-time pulses.

NOTE:

> If the integer prescaler is used in IrDA operation, the baud rate will be determined by the integer prescaler. The non-integer prescaler will then be used for controlling the pulse width, but it must be less than or equal to three-sixteenths of bit time.

For example, in IrDA mode, the non-integer prescaler provides a clock at 1.843200 MHz (115.200 kHz × 16). This clock is used to generate transmit pulses, which are three-sixteenths of a 115.200 kHz bit time.

Table 14-1 on page 14-8 contains the values to use for IrDA operation.

Table 14-1. 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	—	—	—

Example 14-1 provides a sample divisor calculation.

Example 14-1. Sample Divisor Calculation

33.16 MHz sysclk / 1.8432 MHz for IrDA bit time = 18.0

18.0 = 16 + ($20 × 1/16)

Where:

16 = minimum divisor

$20 = step value

1/16 = step size

Table 14-2 contains the values to program into the non-integer prescaler register for IrDA operation.

Table 14-2. Non-Integer Prescaler Settings

Mode	Select (Binary)	Step Value (Hex)
IrDA	011	0x20

14.3.3.3 Integer Prescaler

The baud rate generator can provide standard baud rates from many system clock frequencies. Table 14-3 contains the values that should be used in the UBAUD register for a default 33.16 MHz system clock frequency.

Table 14-3. Selected Baud Rate Settings

Baud Rate	Divider	Prescaler (Hex)
230400	0	0x38
115200	1	0x38
57600	2	0x38
28800	3	0x38
14400	4	0x38
38400	1	0x26
19200	2	0x26
9600	3	0x26
4800	4	0x26
2400	5	0x26
1200	6	0x26
600	7	0x26

14.4 Programming Model

Section 14.4.1, “UART 1 Status/Control Register,” through Section 14.4.14, “FIFO Level Marker Interrupt Register,” describe the UART registers and detailed information about their settings. The UART 1 registers are described first.

14.4.1 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 14-4.

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	RX EN	TX EN	CL KM	PE N	O DD	ST OP	8/7	OD EN	CT SD	RX FE	RX HE	RX RE	TX EE	TX HE	TX AE
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	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 14-4. 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

Table 14-4. UART 1 Status/Control Register Description (Continued)

Name	Description	Setting
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	**$\overline{\text{CTS1}}$ 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

14.4.2 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 14-5.

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
			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	1	1	1	1	1	1
							0x003F									

Table 14-5. 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 SELx bit in the Port E registers must be 0. See Section 10.4.6, “Port E Registers,” on page 10-21 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**—These bits control 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**—These bits control the division value of the baud generator prescaler. The division value is determined by the following formula: Prescaler division value = 65 (decimal) – PRESCALER	See description.

14.4.3 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 14-6.

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	OV RU N	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 14-6. 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.	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 while 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

Table 14-6. UART 1 Receiver Register Description (Continued)

Name	Description	Setting
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. While 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

14.4.4 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 14-7.

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	rw	rw	rw	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 14-7. 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.	0 = Transmitter FIFO is not empty 1 = Transmitter FIFO is empty

Table 14-7. UART 1 Transmitter Register Description (Continued)

Name	Description	Setting
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 14.3.1.2, “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 while the $\overline{\text{CTS1}}$ signal is asserted 1 = Ignore the $\overline{\text{CTS1}}$ signal
BUSY Bit 10	**Busy (Tx Status)**—When this bit is high, it indicates that the transmitter is busy sending a character. This bit is asserted while 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)**—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. While the NOCTS1 bit is high, this bit can serve as a general-purpose input.	0 = $\overline{\text{CTS1}}$ signal is low 1 = $\overline{\text{CTS1}}$ signal is high
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

14.4.5 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 14-8.

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
	BAUDTEST	CLKSRC	FORCEPERR	LOOP	BAUDRESET	IRTEST			RTS1CONT	RTS1	IRDAEN	IRDALOOP	RXPOL	TXPOL		
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 14-8. 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.
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.	0 = Normal operation. 1 = IrDA test mode.
Reserved Bits 9–8	Reserved	These bits are reserved and should be set to 0.

Table 14-8. UART 1 Miscellaneous Register Description (Continued)

Name	Description	Setting
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. When no more than four slots are available, $\overline{\text{RTS1}}$ is negated.
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).
Reserved Bits 1–0	Reserved	These bits are reserved and should be set to 0.

14.4.6 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 14-9.

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 14-9. 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 14-4 on page 14-7 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 = Disable the non-integer prescaler.
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.

14.4.7 Non-Integer Prescaler Programming Example

The following steps show how to generate a 3.072 MHz clock frequency from a 16.580608 MHz clock source.

1. Calculate the divisor:
   divisor = 16.580608 MHz ÷ 3.072000 MHz = 5.397333
2. Find the value for the SELECT field in the NIPR. The divisor is between four and eight, so Table 14-1 on page 14-8 indicates that the SELECT field is 001. The divisor step size for the selected range is one sixty-fourth.
3. Find 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 = 89.42). 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).

14.4.8 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 14-10.

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	RX EN	TX EN	CL KM	PE N	OD D	ST OP	8/7	OD EN	CT SD	RX FE	RX HE	RX RE	TX EE	TX HE	TX AE
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	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 14-10. 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

Table 14-10. UART 2 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	**$\overline{\text{CTS2}}$ 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
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

14.4.9 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 14-11.

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
			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	1	1	1	1	1	1
	0x003F															

Table 14-11. 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 SELx bit in the Port E registers must be 0. See Section 10.4.6, "Port E Registers," on page 10-21 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**—These bits control 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**—These bits control the division value of the baud generator prescaler. The division value is determined by the following formula: Prescaler division value = 65 (decimal) – PRESCALER	See description.

14.4.10 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 14-12.

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 14-12. 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.	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 while 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

Table 14-12. UART 2 Receiver Register Description (Continued)

Name	Description	Setting
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. While 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

14.4.11 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 14-13.

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	rw	rw	rw	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 14-13. 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.	0 = Transmitter FIFO is not empty 1 = Transmitter FIFO is empty

Table 14-13. UART 2 Transmitter Register Description (Continued)

Name	Description	Setting
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 14.3.1.2, “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 while the $\overline{\text{CTS2}}$ signal is asserted 1 = Ignore the $\overline{\text{CTS2}}$ signal
BUSY Bit 10	**Busy (Tx Status)**—When this bit is high, it indicates that the transmitter is busy sending a character. This bit is asserted while 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)**—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. While the NOCTS2 bit is high, this bit can serve as a general-purpose input.	0 = $\overline{\text{CTS2}}$ signal is low 1 = $\overline{\text{CTS2}}$ signal is high
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

14.4.12 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 14-14.

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
	BAUDTEST	CLKSRC	FORCEPERR	LOOP	BAUDRESET	IRTEST			RTS2CONT	RTS2	IRDAEN	IRDALOOP	RXPOL	TXPOL		
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 14-14. 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.
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.	0 = Normal operation. 1 = IrDA test mode.
Reserved Bits 9–8	Reserved	These bits are reserved and should be set to 0.

Table 14-14. UART 2 Miscellaneous Register Description (Continued)

Name	Description	Setting
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. When no more than four slots are available, $\overline{\text{RTS2}}$ is negated.
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).
Reserved Bits 1–0	Reserved	These bits are reserved and should be set to 0.

14.4.13 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 14-15.

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 14-15. 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 14-4 on page 14-7 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 = Disable the non-integer prescaler.
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.

14.4.14 FIFO Level Marker Interrupt Register

The UART FIFO level marker register configures the level at which either the RxFIFO or the TxFIFO 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 14-16.

HMARK 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
					TXFIFO LEVEL MARKER								RXFIFO LEVEL MARKER			
TYPE					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 14-16. FIFO Level Marker Interrupt Register Description

Name	Description	Setting
Reserved Bits 15–12	Reserved	These bits are reserved and should be set to 0.
TXFIFO LEVEL MARKER Bits 11–8	**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 14-17 on page 14-30 for settings.
Reserved Bits 7–4	Reserved	These bits are reserved and should be set to 0.
RXFIFO LEVEL MARKER Bits 3–0	**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 14-17 on page 14-30 for settings.

Table 14-17. FIFO Level Marker Settings

Tx FIFO Level Marker	Number of Slots Empty
0000	Disable
0001	>= 4
0010	>= 8
0011	>= 12
0100	>= 16
0101	>= 20
0110	>= 24
0111	>= 28
1000	>= 32
1001	>= 36
1010	>= 40
1011	>= 44
1100	>= 48
1101	>= 52
1110	>= 56
1111	>= 60

Rx FIFO Level Marker	Number of Bytes Received
0000	Disable
0001	>= 4
0010	>= 8
0011	>= 12
0100	>= 16
0101	>= 20
0110	>= 24
0111	>= 28
1000	>= 32
1001	>= 36
1010	>= 40
1011	>= 44
1100	>= 48
1101	>= 52
1110	>= 56
1111	>= 60

Chapter 15
Pulse-Width Modulator 1 and 2

This chapter describes the DragonBall VZ’s two pulse-width modulators (PWMs). Each of the pulse-width modulators has 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.

15.1 Introduction to PWM Operation

PWM 1 uses 8-bit resolution, which is compatible with the MC68EZ328 (DragonBall EZ). PWM 2 uses 16-bit resolution, which is compatible with the MC68328 (the original DragonBall). 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 15-1.

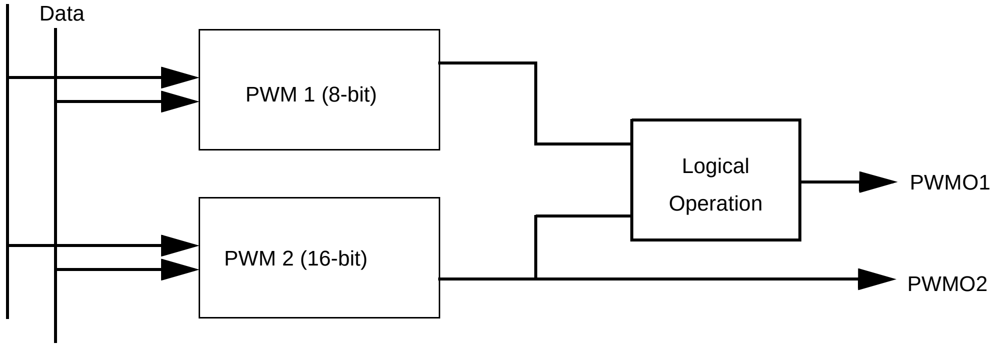


Figure 15-1. PWM 1 and PWM 2 System Configuration Diagram

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 5.2.2, “Peripheral Control Register,” on page 5-4 for details about the settings of these bits.

15.1.1 PWM Clock Signals

Figure 15-2 shows a simplified block diagram of PWM 1. 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 the pulse width modulator is made by the clock source (CLKSRC) bit in the PWM 1 control register.

The CLKSEL (clock selection) field in the 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 factor from 1 to 128, respectively.

15.2 PWM 1

PWM 1 is an 8-bit PWM module that is 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 15-2 illustrates the block diagram of the pulse-width modulator unit 1.

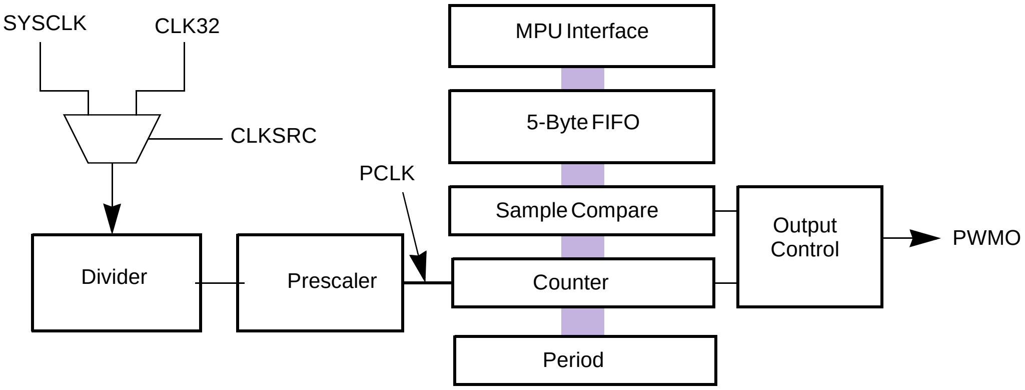


Figure 15-2. PWM 1 Block Diagram

15.3 PWM Operation

The pulse-width modulator has three modes of operation—playback, tone, and D/A.

15.3.1 Playback Mode

In playback mode, the pulse-width modulator uses 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 15-3 illustrates how variable-width pulses affect an audio waveform.

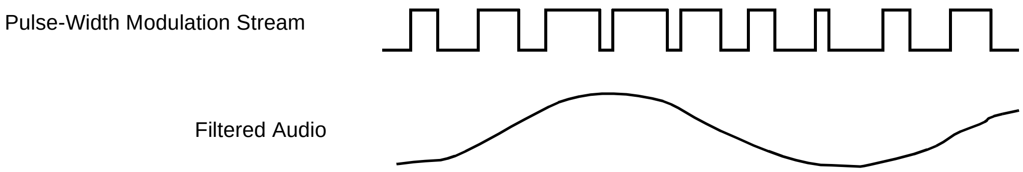


Figure 15-3. Audio Waveform Generation

Digital sample values can be loaded into the pulse-width modulator 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.

15.3.1.1 Tone Mode

In tone mode, the pulse-width modulator generates a continuous tone at a single frequency when the PWM registers are programmed. The lowest frequency that can be generated is 0.25 Hz.

15.3.1.2 D/A Mode

The pulse-width modulator 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, it becomes a D/A converter.

15.4 Programming Model

This section contains programming information about both PWM 1 and PWM 2.

15.4.1 PWM 1 Control Register

This register controls the operation of the pulse-width modulator, 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 15-1.

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	w	w	rw	rw
RESET	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
									0x0020							

Table 15-1. PWM 1 Control Register Description

Name	Description	Setting
CLKSRC Bit 15	**Clock Source**—This bit is used to select the clock source to the pulse-width modulator.	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. If a 38.4 kHz crystal is used, 38.4 kHz is selected.
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 PWMS 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 one 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 15-1. 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 the pulse-width modulator. If this bit is not enabled, writing to other pulse-width modulator registers is ignored.	0 = Disabled* 1 = Enabled**
REPEAT Bits 3–2	**Sample Repeats**—These write-only bits select 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 the pulse-width modulator 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 the pulse-width modulator is enabled, it begins a new period, and the following events occur:

- The output pin is set to start a new period.
- The prescaler and counter are released and begin counting.
- The IRQ bit is set, thus indicating that the FIFO is empty.

15.4.2 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, high byte is first placed into the 8-bit FIFO, and then low byte. When individual sample bytes are being written, they must be written to the low byte (SAMPLE1) only. The pulse-width modulator 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 15-2.

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 15-2. 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 the pulse-width modulator 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 the pulse-width modulator after the SAMPLE0 field. When used with single 8-bit samples, data must be written to this byte.	None

15.4.3 PWM 1 Period Register

This register controls the pulse-width modulator period. When the counter value matches PERIOD + 1, the counter is reset to start another period. Therefore, the following equation applies:

$$\text{PWMO (Hz)} = \text{PCLK (Hz)} / (\text{PERIOD} + 2) \qquad \textit{Eqn. 15-1}$$

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 15-3.

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 15-3. PWM 1 Period Register Description

Name	Description	Setting
PERIOD Bits 7–0	**Period**—This field represents the pulse-width modulator's period control value.	None

15.4.4 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 15-4.

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 15-4. PWM 1 Counter Register Description

Name	Description	Setting
COUNT Bits 7–0	**Count**—This field represents the value of the current count.	None

15.5 PWM 2

PWM 2 is a 16-bit PWM module that is compatible with the one used in the original DragonBall processor, MC68328. 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 15-4 illustrates the block diagram of the pulse-width modulator unit 2.

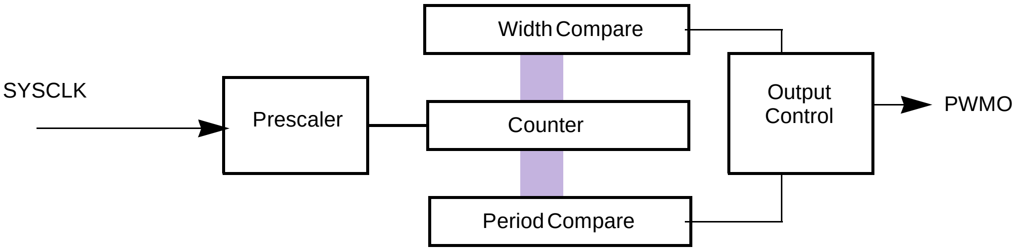


Figure 15-4. PWM 2 Block Diagram

15.5.1 PWM 2 Control Register

This register controls how the overall pulse-width modulator 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 15-5.

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 15-5. 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.

Table 15-5. PWM 2 Control Register Description (Continued)

Name	Description	Setting
PIN Bit 7	**Pin Status Indicator**—This bit indicates the current status of the PWM.	0 = PWM output is high. 1 = PWM output is low.
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**—These bits select 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.

15.5.2 PWM 2 Period Register

This register controls the period of PWM 2. When the counter value matches the 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 15-6.

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 15-6. PWM 2 Period Register Description

Name	Description	Setting
PERIOD Bits 15–0	**Period**—This field represents the pulse-width modulator's period control value.	None

NOTE:

There is an special case: when the register is set to $00, the output will never go high. The pulse signal duty cycle will be 0 percent.

15.5.3 PWM 2 Pulse Width 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 15-7.

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 15-7. 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.

15.5.4 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 15-8.

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	rw	rw	rw	rw	rw	rw	rw	rw	rw	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 15-8. PWM 2 Counter Register Description

Name	Description	Setting
COUNT Bits 15–0	**Count**—Indicates the current counter value.	None

Chapter 16
In-Circuit Emulation

This chapter describes the in-circuit emulation (ICE) module of the MC68VZ328 and provides detailed information about its operation and registers. The ICE module is designed to support low-cost emulator designs using the MC68VZ328 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 16-1 illustrates the block diagram of the in-circuit emulation module.

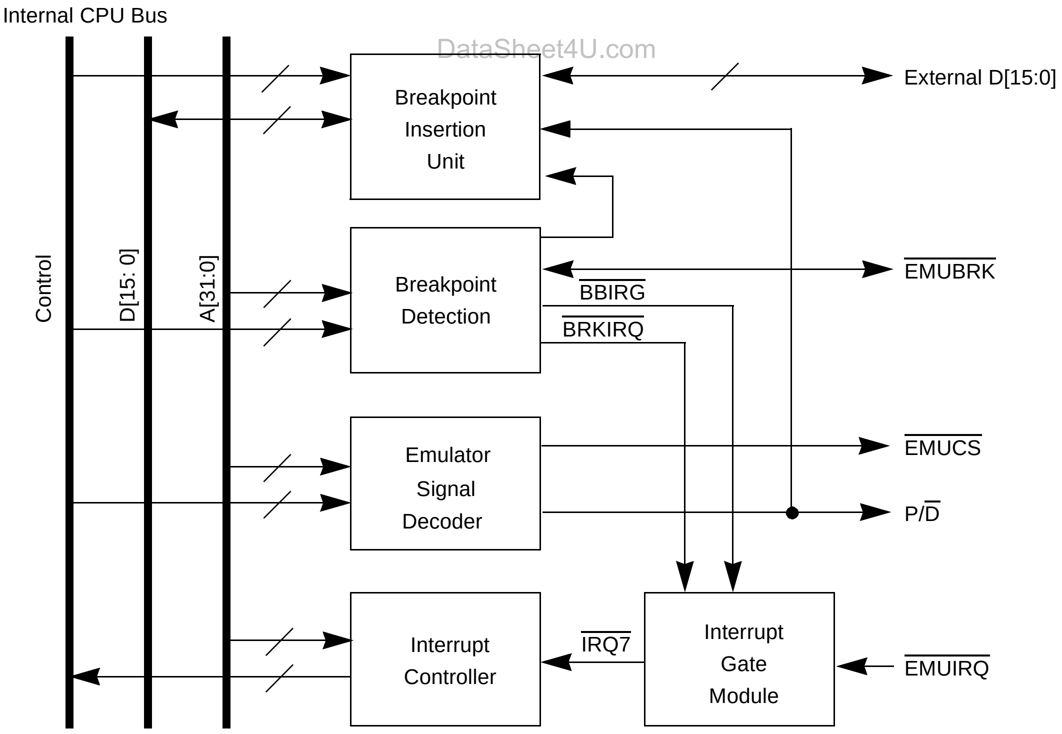


Figure 16-1. In-Circuit Emulation Module Block Diagram

16.1 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

16.1.1 Entering Emulation Mode

The in-circuit emulation module latches the state of the $\overline{\text{EMUIRQ}}$ signal on the rising edge of the $\overline{\text{RESET}}$ signal. To put the MC68VZ328 in emulation mode, the $\overline{\text{EMUIRQ}}$ signal must be driven low (externally) during system reset. After system reset, $\overline{\text{EMUIRQ}}$ becomes a falling edge trigger signal, which generates a level 7 interrupt when active. For emulation mode, the $\overline{\text{CSA0}}$ signal is not asserted for reset fetch, since it is 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 0xFFFCFFFF, 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 MC68VZ328 begins another operation mode.

16.1.2 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.

Since 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 16.1.5, “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.

16.1.2.1 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.

16.1.3 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 0xFFFC0000–0xFFFCFFFF (64K), 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.

16.1.4 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 9, “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.

16.1.5 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.

16.2 Programming Model

This section contains information about the ICE registers and programming information about their settings.

16.2.1 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 16-1 on page 16-5.

ICEMACR ICE Module Address Compare Register 0x(FF)FFFFD00

	BIT 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	BIT 16
	AC31	AC30	AC29	AC28	AC27	AC26	AC25	AC24	AC23	AC22	AC21	AC20	AC19	AC18	AC17	AC16
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	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
	AC15	AC14	AC13	AC12	AC11	AC10	AC9	AC8	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	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)FFFFD04

	BIT 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	BIT 16
	AM31	AM30	AM29	AM28	AM27	AM26	AM25	AM24	AM23	AM22	AM21	AM20	AM19	AM18	AM17	AM16
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	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
	AM15	AM14	AM13	AM12	AM11	AM10	AM9	AM8	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
TYPE	rw	rw	rw	rw	rw	rw	rw	rw	rw	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 16-1. ICE Module Address Compare and Mask Registers Description

Name	Description	Setting
ACx Bits 31–0	**Address Compare 31–0**—These bits represent 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.
AMx Bits 31–0	**Address Mask 31–0**—These bits mask 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.

16.2.2 In-Circuit Emulation Module Control Compare and Mask Register

The in-circuit emulation module control compare (ICEMCCR) register 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 16-2 and Table 16-3.

ICEMCCR ICE Module Control Compare Register **0x(FF)FFFFFD08**

	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 16-2. 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 Control Mask Register **0x(FF)FFFFFD0A**

	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 16-3. ICE Control Mask Register Description

Name	Description	Setting
Reserved Bits 15–2	Reserved	These bits are reserved and should be set to 0.

Table 16-3. ICE Control Mask Register Description (Continued)

Name	Description	Setting
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.

16.2.3 In-Circuit Emulation 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 16-4.

ICEMCR ICE Module Control Register **0x(FF)FFFFD0C**

	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 16-4. ICE Module Control Register Description

Name	Description	Setting
Reserved Bits 15–7	Reserved	These bits are reserved and should be set to 0.
SWEN Bit 6	**Software Enable EMU Module**—In normal mode, writing to this bit enables the breakpoint function.	0 = Disable breakpoint function. 1 = Enable breakpoint function.
Reserved Bit 5	Reserved	This bit is reserved and should be set to 0.
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.
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 in Table 16-5 on page 16-9. 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 16-5 on page 16-9.
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.

Table 16-4. ICE Module Control Register Description (Continued)

Name	Description	Setting
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 16-5. 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

16.2.4 In-Circuit Emulation 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 16-6.

ICEMSR ICE Module Status Register **0x(FF)FFFFD0E**

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

0x0000

Table 16-6. 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**—This bit, when set, 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.

16.3 Typical Design Programming Example

Figure 16-2 on page 16-11 illustrates an example of a typical emulator design. It is a simple and low-cost design that uses the MC68VZ328 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 MC68VZ328 pin-out extension to connect to the solder-on emulator pod. The entire MC68VZ328 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.

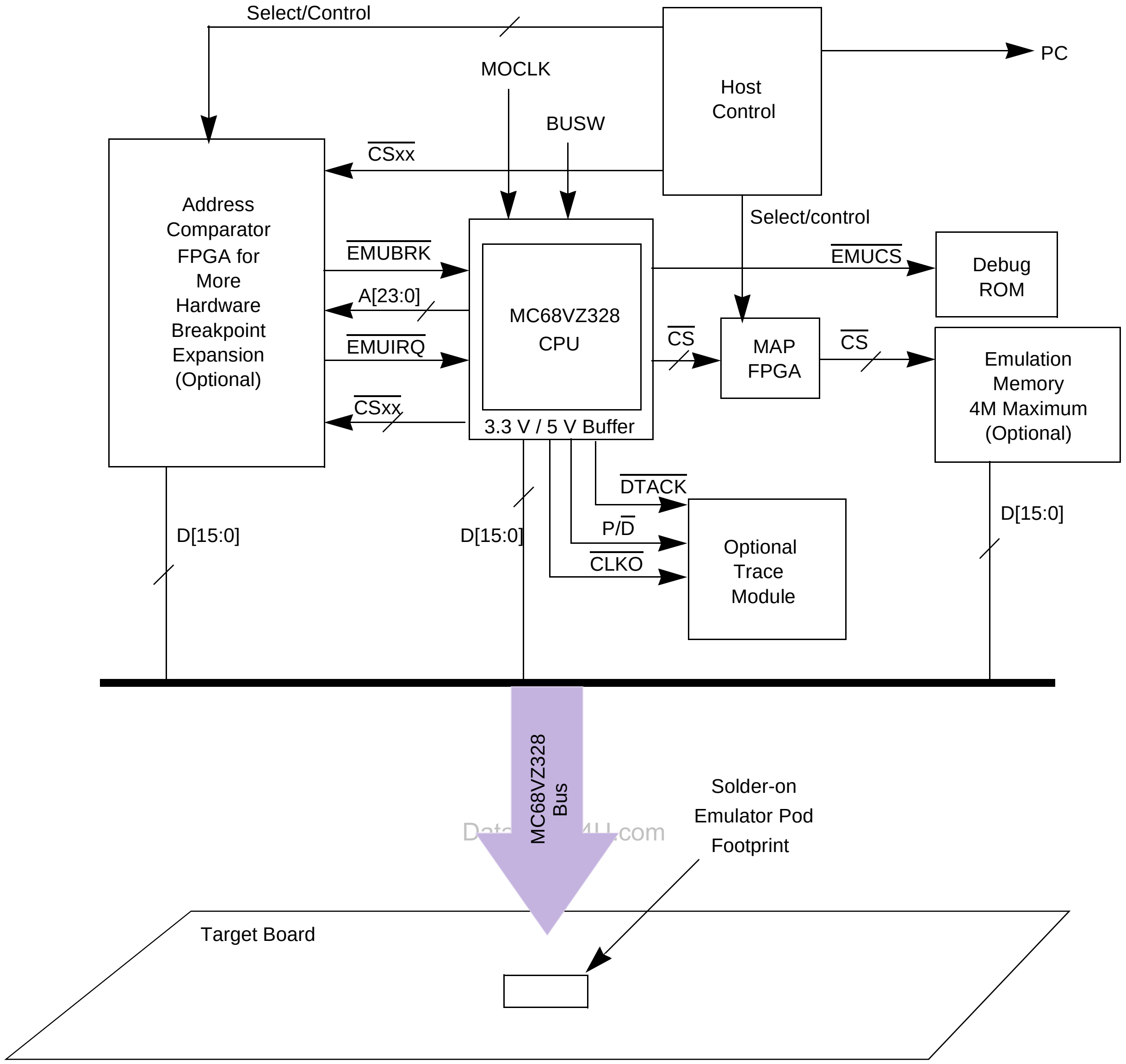


Figure 16-2. Typical Emulator Design Example

16.3.1 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.

16.3.2 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.

16.3.3 Emulation Memory Mapping FPGA and Emulation Memory

Since 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.

16.3.4 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 16.1.2, “Detecting Breakpoints,” 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.

16.3.5 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.

16.4 Plug-in Emulator Design Example

Figure 16-3 on page 16-13 displays an example of a plug-in emulator design. The design is simple and low-cost, and it creates a very basic debugging environment.

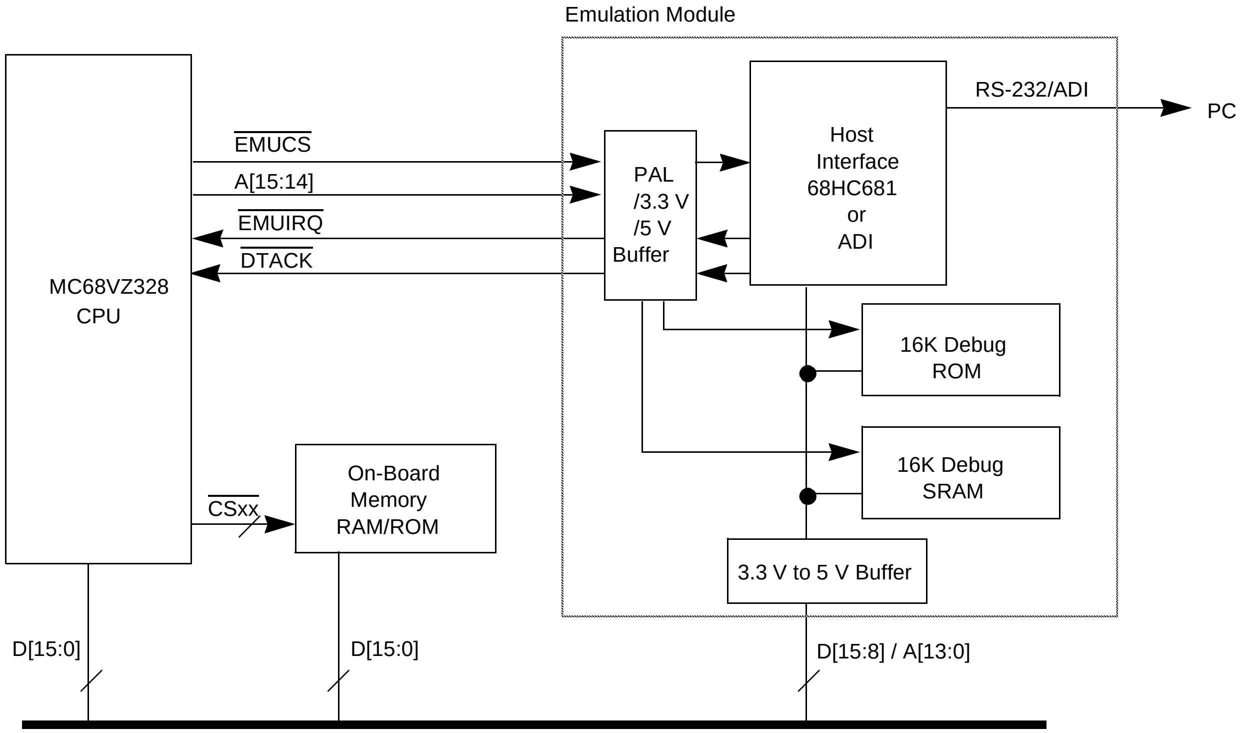


Figure 16-3. Plug-in Emulator Design Example

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.

16.5 Application Development Design Example

Figure 16-4 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.

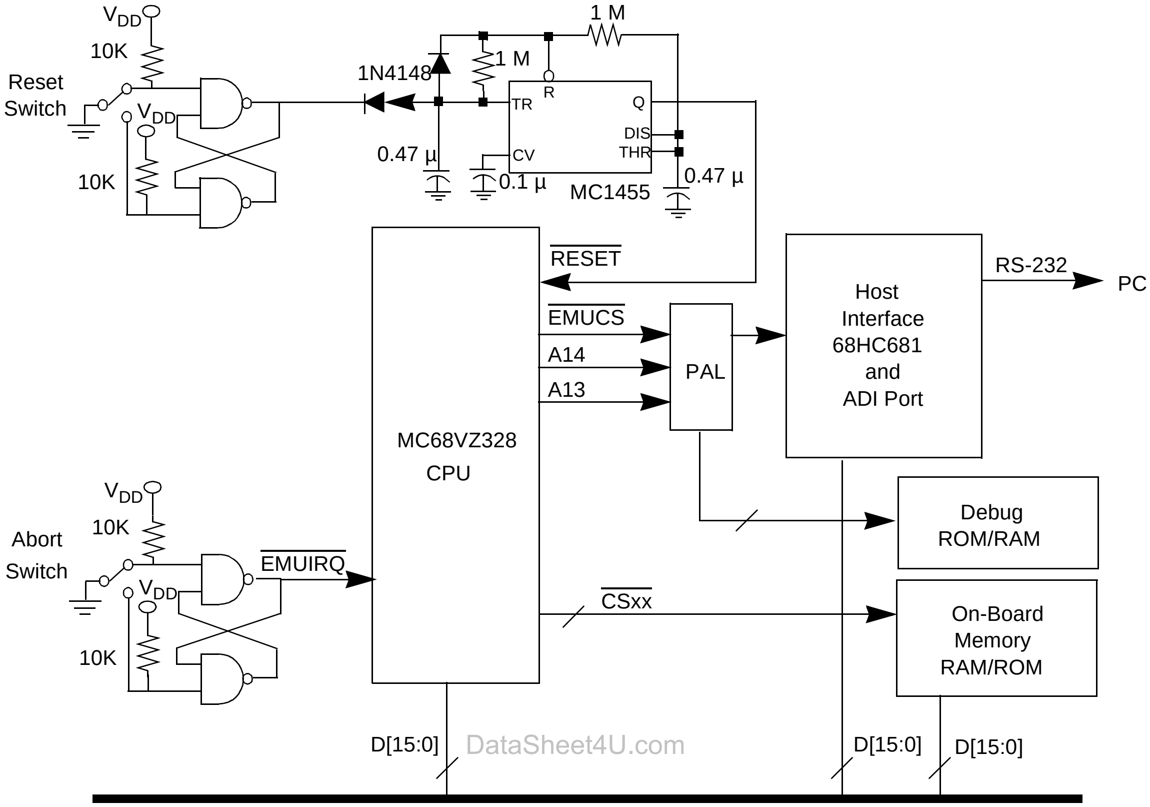


Figure 16-4. Application Development System Design Example

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.

Chapter 17
Bootstrap Mode

This chapter describes the operation and programming information of the bootstrap mode of the MC68VZ328. 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 UART 2 controller. See Chapter 14, “Universal Asynchronous Receiver/Transmitter 1 and 2,” for information on operating and programming the UART controllers. Once a program is downloaded to the MC68VZ328, 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

17.1 Bootstrap Mode Operation

In bootstrap mode, the MC68VZ328’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 MC68VZ328 whether the PLL input clock is 32.768 kHz or 38.4 kHz crystal, as well as 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 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 MC68VZ328’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 MC68VZ328 is operating in a CPU standalone system. The instruction buffer starts at 0xFFFFC0. 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 record type determines what occurs.

17.1.1 Entering Bootstrap Mode

Bootstrap mode is one of the three operation modes (normal, emulation, and bootstrap) of the MC68VZ328. Of the three modes, bootstrap has the highest priority. To enter bootstrap mode, the $\overline{\text{EMUBRK}}$ signal must be driven low and a system reset must be performed. After reset, bootstrap reset vectors are internally generated for reset vector fetch cycles. Figure 17-1 illustrates 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.

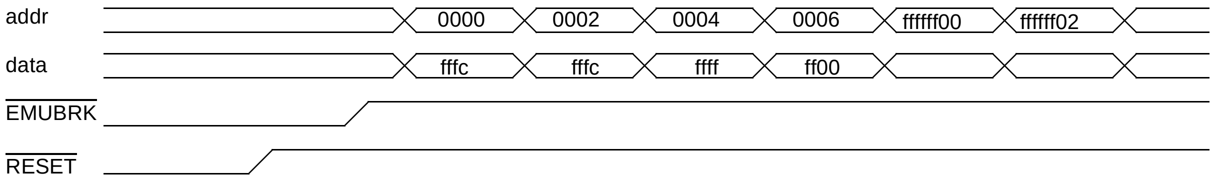


Figure 17-1. Bootstrap Mode Reset Timing

17.1.2 Bootstrap Record Format

Bootstrap mode data transfers will only accept bootstrap records (b-records) whose format is shown in Table 17-1. The two major attributes of b-records are that they are in uppercase and end with a carriage return.

Table 17-1. Bootstrap Record Format

4-Byte	1-Byte	N(Count)-Byte
Address	Count	Data

17.1.2.1 Data B-Record Format

There are two types of b-records that use the same format. The data b-record contains data to be transferred. The 4-byte address field indicates where the data will be stored, and this address could be any MC68VZ328 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.

17.1.2.2 Execution B-Record Format

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:

- After a program is downloaded to system RAM, issuing an execution b-record initiates program execution. In this case, the address field of the b-record will be the start address of the program.
- When loading a 68000 instruction into the instruction buffer and filling the remainder of the unused buffer space with `nop--$4e71`, issuing an execution b-record executes the 68000 instruction that is stored in IBUFF and returns to bootloader mode. In this case, the address field of the b-record will be the start address of IBUFF.

17.1.3 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 match the 32.768 kHz or 38.400 kHz crystal by reading the first received character. 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.”

17.1.4 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 14.4.2, “UART 1 Baud Control Register,” on page 14-12. 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 (the default). 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.

17.1.5 System Initialization Programming Example

Before downloading a program to system memory, the target system may need to be initialized using the internal registers. An init file can be built using a text editor. Example 17-1 is an initialization file for the MC68VZ328ADS board.

Example 17-1. System Initialization Programming Example

```
*************************************************
* init.b -- Init ADS to default monitor config
* date: 04/20/98
*
*************************************************

FFFFF1180130  emucs init
FFFFF000011C  SCR init
FFFFFB0A0100  Disable WD
FFFFF42B0183  enable clko
FFFFF40B0100  enable chip select
FFFFFD0D0108  disable hardmap
FFFFFD0E0107  clear level 7 interrupt
FFFFF100020100 CSA 2M - 4M
FFFFF1100201A7
FFFFF102020000 CSB 0 - 256K
FFFFF112020091
FFFFFC00028F00 DRAM Config
FFFFFC02029667 DRAM Control
FFFFF106020200 CSD init -- RAS0 4M-6M, RAS1 6M-8M
FFFFF11602029D enable DRAM cs
FFFFF3000140   IVR
FFFFF30404007FFFFF IMR
```

NOTE:

The bootloader starts receiving a new b-record when a nonhexadecimal digit is received. Therefore, comments can be made in the b-record file as long as it contains no more than eight consecutive hexadecimal digits.

17.1.6 Application Programming Example

The code shown in Example 17-2 can be used to calculate a CRC value. The example demonstrates how assembly code is assembled and downloaded to system RAM.

Example 17-2. 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 17-2, 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.

```
0000400010428142423C30200032C6548154420C42
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 $FFFFFF5A` 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 MC68VZ328 will remain in bootstrap mode.

17.1.7 Example of Instruction Buffer Usage

Example 17-3 demonstrates how to run a 68000 instruction using the instruction buffer.

Example 17-3. Using Instruction Buffers

```
ORG.L           $FFFFFFC0                       ; instruction buffer location
                move.w #$55,D0                  ; 4-byte long instruction(303C0055)
                nop                             ; fill the rest of IBUFF
                nop
nop
nop
                end
```

After the data is assembled and converted to b-record format, it appears as in the following lines (where FFFFFFC0 is the IBUFF address location):

```
FFFFFFC00C303C00554E714E714E714E71
FFFFFFC000
```

The first b-record loads the instruction buffer. The second b-record tells the bootloader to run the instruction in the instruction buffer. When the execution is complete, it accepts new b-record transfers. The CPU registers D0–D6 and A0 are used by the bootloader program. Writing to these registers may corrupt the bootloader program.

17.2 Bootloader Flowchart

The following flowchart illustrates how the bootloader program operates inside the MC68VZ328. The bootloader starts when the MC68VZ328 enters bootstrap mode.

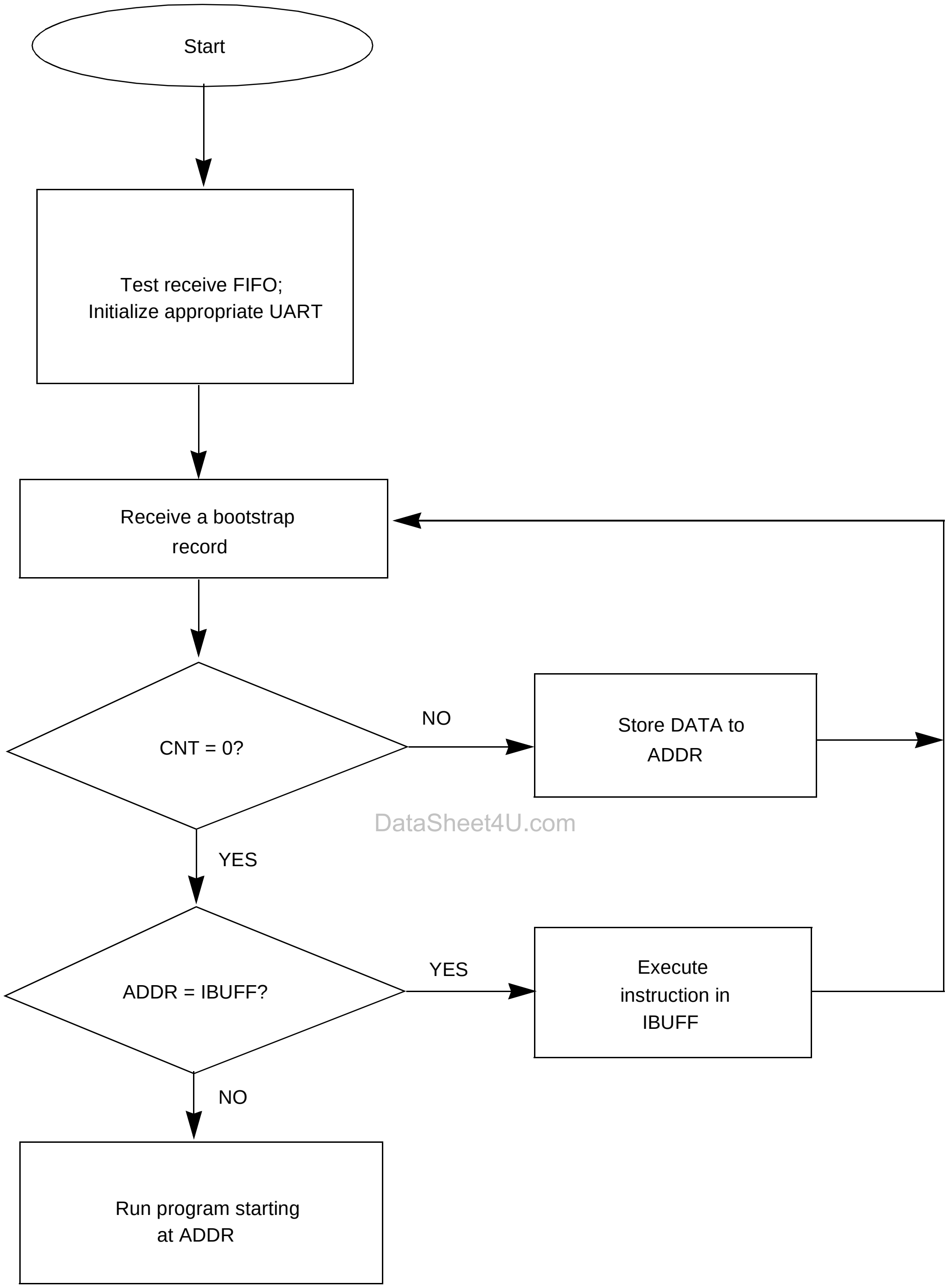


Figure 17-2. Bootloader Program Operation

17.3 Special Notes

The following information may be useful when the MC68VZ328 is in bootstrap mode.

- A b-record is a string of uppercase hex characters with optional comments that follow.
- Comments in a b-record or b-record 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 18
Application Guide

This chapter contains helpful information that will assist with integrating the MC68VZ328 into new or existing designs. It includes a design checklist and instructions for using the MC68VZ328 Application Development System (ADS) board to get the design process started as quickly as possible.

18.1 Design Checklist

When the MC68VZ328 microprocessor is being integrated into an application, the following items can be used as guides during the design process. These guidelines are the result of issues that frequently occurred during debugging or in the process of operating actual designs.

18.1.1 Determining the Chip ID and Version

Each chip has different sets of numbers etched onto it, and one of these sets is the mask and revision number for that particular chip. The mask number and the revision number are combined into one. For example, with the number 0F98S, 0 is the revision number and F98S is the mask number. This information is necessary for obtaining the correct errata information for that version of the chip, ensuring more efficient product design. Once the mask and revision numbers are known, go to the DragonBall Web site (http://www.Motorola.com/DragonBall) and look for any MC68VZ328 chip errata pertaining to those numbers. If Web access is not available, contact the local Motorola sales office.

18.1.2 8-Bit Bus Width Issues

To ensure maximum flexibility, the MC68VZ328 supports both 8- and 16-bit data bus modes. Except the chip-select group A, which carries the boot chip select signal $\overline{\text{CSA0}}$ and is normally connected to boot ROM, all the chip select signals are programmable to 8-bit or 16-bit mode after reset. The data bus width for the $\overline{\text{CSA0}}$ and $\overline{\text{CSA1}}$ signals is only controlled by the BUSW/$\overline{\text{DTACK}}$/PG0 signal. For a system with 16-bit data boot ROM, BUSW is pulled high or left unconnected during system reset. For an 8-bit data boot ROM system, BUSW must be externally driven low during system reset. The BUSW status is latched by the rising edge of the $\overline{\text{RESET}}$ signal, and the latched BUSW status is indicated by the BSW bit of the chip-select A control register. See Section 6.3.3, "Chip-Select Registers," on page 6-8 for more details. Also, after reset, the BUSW/$\overline{\text{DTACK}}$/PG0 pin can be selected as a $\overline{\text{DTACK}}$ or PG0 function, but it defaults to the $\overline{\text{DTACK}}$ function. This signal should be permanently driven low for an 8-bit system to force all bus cycles to a zero wait state until this pin is reconfigured to the PG0 function. Fortunately, the system clock is divided by two (the PRESC bit in the PLLCR register is set) after reset, which doubles the length of each bus cycle and provides ample access time to memories. Therefore, BUSW/$\overline{\text{DTACK}}$/PG0 should be programmed to the PG0 function before the system clock is configured to divide by one (the PRESC bit in the PLLCR register is cleared).

18.1.3 Clock and Layout Considerations

This section covers layout considerations affecting DragonBall timing issues during operation and also during the initial power up.

- Place the crystal within 0.5 inches of the MC68VZ328. The crystal and the capacitors must be as close to the chip as possible.
- If an RC reset circuit is being used, place the resistor and capacitor within 0.5 inches of the MC68VZ328. The $\overline{\text{RESET}}$ pin is a Schmitt trigger input signal. A simple power-up RC reset circuit can be used. Since the internal module takes time to complete the reset operation, a minimum 250 ms power-up reset pulse is required.
- Use multiple power and ground planes. It is strongly recommended to use at least one ground plane, one 3.3 V V_{DD} plane, and one 5 V V_{SS} plane (if 5 V parts exist in the system). This helps improve the power stability and enhance the noise immunity of the system.

18.1.4 Bus and I/O Considerations

Several of the items that are warned against in this section appear to be good design practice. However, experience has demonstrated that not heeding the following suggestions can lead to problems.

- Do not leave unused input pins floating. Unused inputs should be tied high or low, but not left floating. Unused inputs can be tied directly to V_{SS} or V_{DD} or through pull-ups or pull-downs to V_{SS} or V_{DD}.
- Use the port pins efficiently. When port pins are not used, they should be configured as inputs with pull-up enabled or as an output with pull-up disabled to reduce power consumption.
- Apply internal pull-ups to dedicated function pins. Many pins are mixed with a dedicated function. The internal pull-up or pull-down resistors apply to both the dedicated function and the general I/O function. For instance, when using the RXD/PE4 signal and the RXD function, the associated internal pull-up resistor can be used to pull up the RXD input signal.
- Do not rely solely on the value of internal pull-up resistors. The internal resistors are nominally 1 megaohm, but their deviation is large.
- Always provide a development interface port on your design. The MC68VZ328 has bootstrap mode and a bootstrap utility program that can be used to download programs and data to a target system and perform simple hardware debugging functions. However, bootstrap mode only uses the RXD and TXD signals of the UART port, so it is recommended that a UART port be included in the design for system debugging and flash memory updating.

Chapter 19
Electrical Characteristics

This chapter documents electrical characteristics and provides timing information necessary to design systems using the MC68VZ328 microprocessor. Section 19.2, “DC Electrical Characteristics,” provides detailed information about both maximum and minimum DC characteristics of the MC68VZ328. Section 19.3, “AC Electrical Characteristics,” consists of output delays, input setup and hold times, and signal skew times. It also contains timing information for working with RAM, DRAM, and other memory-related modules and peripherals.

19.1 Maximum Ratings

Table 19-1 provides information on maximum ratings.

Table 19-1. Maximum Ratings

Rating	Symbol	Value	Unit
Supply voltage	V_{DD}	–0.3 to 7.0	V
Input voltage	V_{IN}	–0.3 to 7.0	V
Maximum operating temperature range	T_A	T_L to T_H 0 to 70	°C
Storage temperature	Test	–55 to 150	°C

19.2 DC Electrical Characteristics

Table 19-2 contains both maximum and minimum DC characteristics of the MC68VZ328.

Table 19-2. Maximum and Minimum DC Characteristics

Number or Symbol	Characteristic	(3.0 ± 0.3) V			Unit
		Minimum	Typical	Maximum	
1	Full running operating current at 33 MHz	—	20	40	mA
2	Standby current[1]	—	35	60	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] Standby current is measured only when the real-time clock is running.

19.3 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 33 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 95 pF loading.

19.3.1 CLKO Reference to Chip-Select Signals Timing

Figure 19-1 on page 19-3 compares the chip-select signal time referenced with the CLKO 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 19-3 on page 19-3. For detailed information about the individual signals, see Chapter 6, “Chip-Select Logic.”

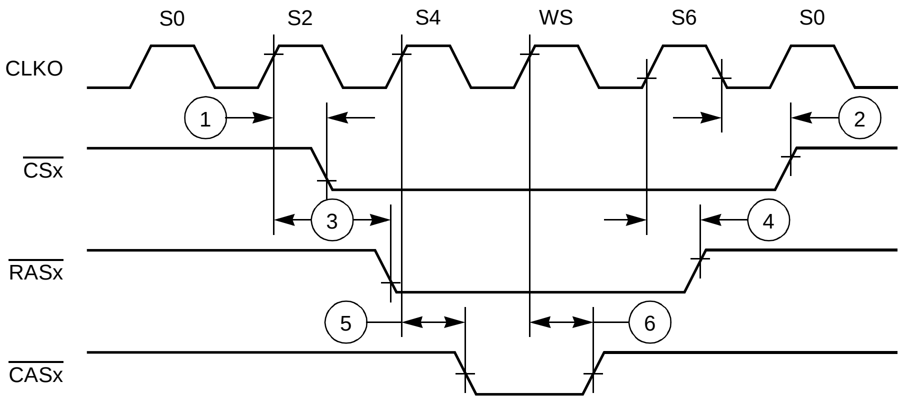


Figure 19-1. CLKO Reference to Chip-Select Signals Timing Diagram

Table 19-3. CLKO Reference to Chip-Select Signals Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	CLKO high to $\overline{\text{CSx}}$ asserted	—	10	ns
2	CLKO low to $\overline{\text{CSx}}$ negated	—	12	ns
3	CLKO high to $\overline{\text{RASx}}$ asserted	—	10	ns
4	CLKO high to $\overline{\text{RASx}}$ negated	—	12	ns
5	CLKO high to $\overline{\text{CASx}}$ asserted	—	10	ns
6	CLKO high to $\overline{\text{CASx}}$ negated	—	12	ns

19.3.2 Chip-Select Read Cycle Timing

Figure 19-2 on page 19-4 shows the read cycle timing used by chip-select. The signal values and units of measure for this figure are found in Table 19-4 on page 19-4. For detailed information about the individual signals, see Chapter 6, “Chip-Select Logic.”

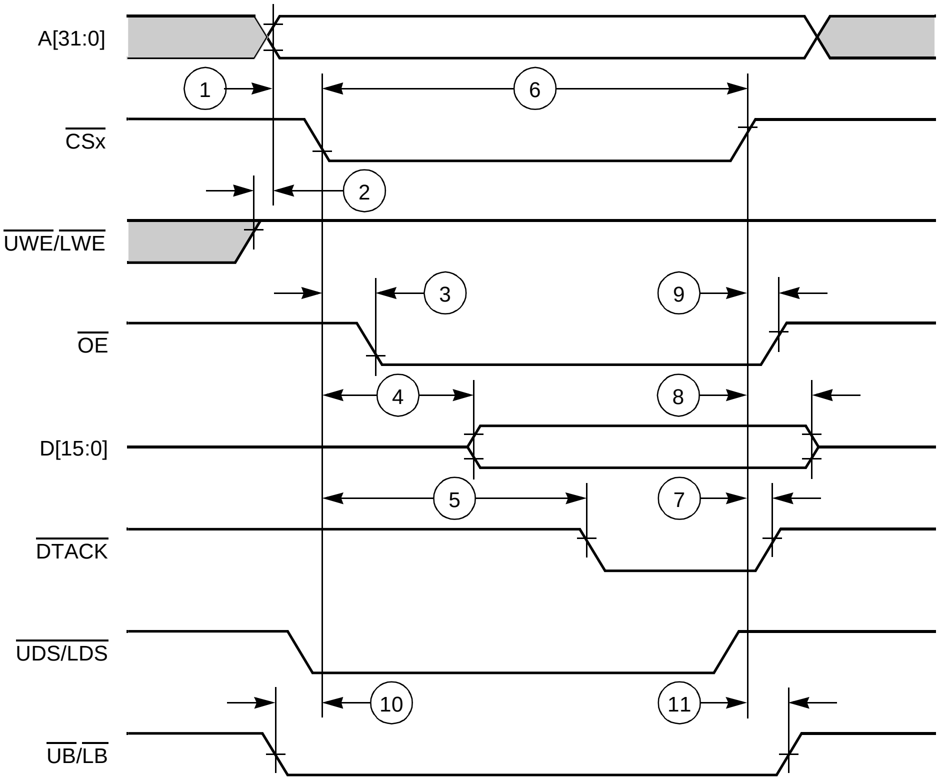


Figure 19-2. Chip-Select Read Cycle Timing Diagram

Table 19-4. Chip-Select Read Cycle Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Address valid to CSx asserted (bit ECDS = 0, bit ECDS = 1)	20, 20 - T/2	—	ns
2	UWE/LWE negated before row address valid	0	—	ns
3	CSx asserted to OE asserted	—	0	ns
4	Data-in valid from CSx asserted	—	35 + nT	ns
5	External DTACK input setup from CSx asserted	—	20 + nT	ns
6	CSx pulse width (bit ECDS = 0, bit ECDS = 1)	60 + nT, (60 + T/2) + nT	—	ns
7	External DTACK input hold after CSx is negated	0	—	ns
8	Data-in hold after CSx is negated	0	—	ns
9	OE negated after CSx is negated	0	10	ns
10	UB/LB asserted to CSx asserted (16-bit SRAM)	10	—	ns

Table 19-4. Chip-Select Read Cycle Timing Parameters (Continued)

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
11	$\overline{\mathrm{CSx}}$ negated to $\overline{\mathrm{UB/LB}}$ negated (16-bit SRAM)	10	—	ns

Note:
n is the number of wait states in the current memory access cycle.
T is the system clock period.
The external $\overline{\mathrm{DTACK}}$ input requirement is eliminated when $\overline{\mathrm{CSx}}$ is programmed to use internal $\overline{\mathrm{DTACK}}$.
$\overline{\mathrm{CSx}}$ stands for $\overline{\mathrm{CSA0}}$, $\overline{\mathrm{CSA1}}$, $\overline{\mathrm{CSB0}}$, $\overline{\mathrm{CSB1}}$, $\overline{\mathrm{CSC0}}$, $\overline{\mathrm{CSC1}}$, $\overline{\mathrm{CSD0}}$, or $\overline{\mathrm{CSD1}}$.
A value in parentheses is used when early cycle detection is turned on.

19.3.3 Chip-Select Write Cycle Timing

Figure 19-3 shows the write cycle timing used by chip-select. The signal values and units of measure for this figure are found in Table 19-5 on page 19-6. For detailed information about the individual signals, see Chapter 6, “Chip-Select Logic.”

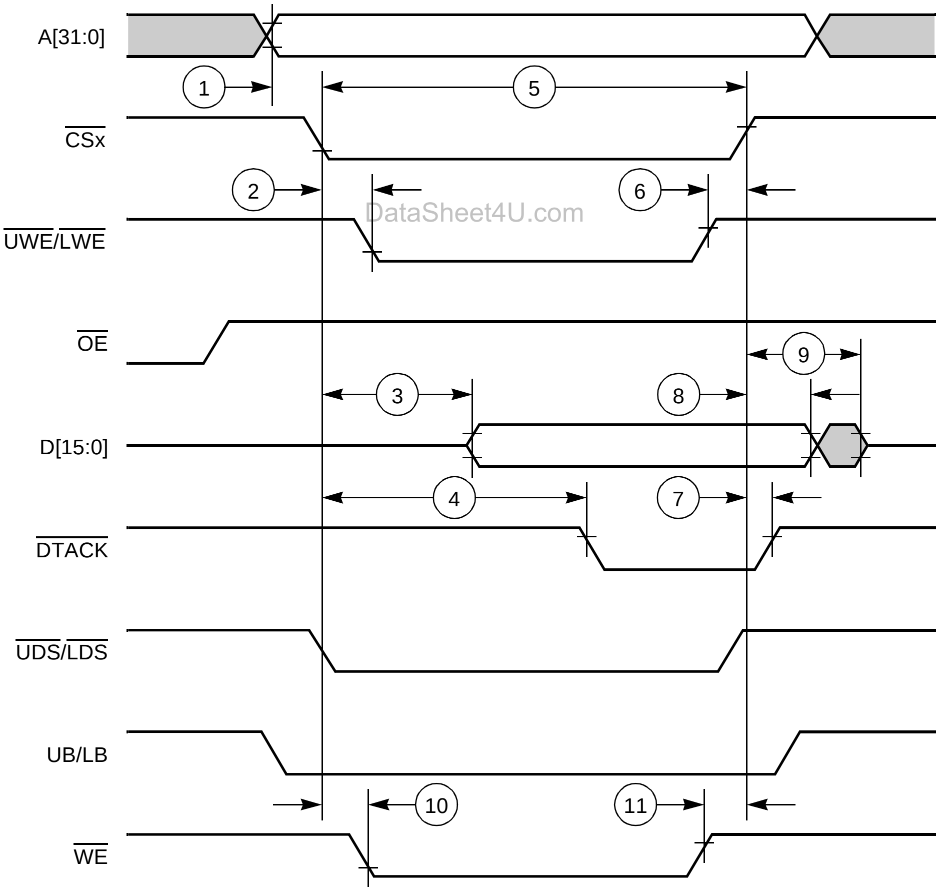


Figure 19-3. Chip-Select Write Cycle Timing Diagram

Table 19-5. Chip-Select Write Cycle Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Address valid to CSx asserted (bit ECDS = 0, bit ECDS = 1)	20, 20 - T/2	—	ns
2	CSx asserted to UWE/LWE asserted	0	4	ns
3	CSx asserted to data-out valid	—	30	ns
4	External DTACK input setup from CSx asserted	—	20 + nT	ns
5	CSx pulse width (bit ECDS = 0, bit ECDS = 1)	60 + nT, (60 + T/2) + nT	—	ns
6	UWE/LWE negated before CSx is negated	10	20	ns
7	External DTACK input hold after CSx is negated	0	—	ns
8	Data-out hold after CSx is negated	8	—	ns
9	CSx negated to data-out in Hi-Z	—	18	ns
10	CSx asserted to WE asserted (16-bit SRAM)	0	4	ns
11	WE negated before CSx is negated (16-bit SRAM)	10	20	ns

Note:
n is the number of wait-states in the current memory access cycle.
T is the system clock period.
The external DTACK input requirement is eliminated when CSx is programmed to use the internal DTACK.
CSx stands for CSA0, CSA1, CSB0, CSB1, CSC0, CSC1, CSD0, or CSD1.
A value in parentheses is used when early detection is turned on.

19.3.4 Chip-Select Flash Write Cycle Timing

Figure 19-4 on page 19-7 shows the flash write cycle timing used by chip-select. The signal values and units of measure for this figure are found in Table 19-6 on page 19-7. For detailed information about the individual signals, see Chapter 6, "Chip-Select Logic."

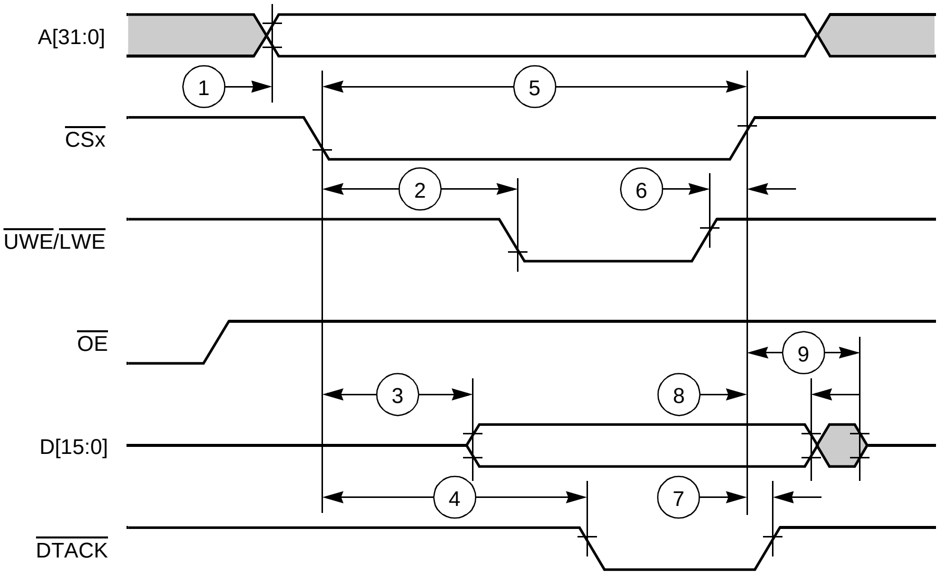


Figure 19-4. Chip-Select Flash Write Cycle Timing Diagram

Table 19-6. Chip-Select Flash Write Cycle Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Address valid to CSx asserted (bit ECDS = 0, bit ECDS = 1)	20, 20 - T/2	—	ns
2	CSx asserted to UWE/LWE asserted	20	40	ns
3	CSx asserted to data-out valid	—	30	ns
4	External DTACK input setup from CSx asserted	—	20 + nT	ns
5	CSx pulse width (bit ECDS = 0, bit ECDS = 1)	60 + nT, (60 + T/2) + nT	—	ns
6	UWE/LWE negated before CSx is negated	10	20	ns
7	External DTACK input hold after CSx is negated	0	—	ns
8	Data-out hold after CSx is negated	8	—	ns
9	CSx negated to data-out in Hi-Z	—	18	ns

Note:
n is the number of wait states in the current memory access cycle.
T is the system clock period.
The external DTACK input requirement is eliminated when CSx is programmed to use the internal DTACK.
CSx stands for CSA0, CSA1, CSB0, CSB1, CSC0, CSC1, CSD0, or CSD1.
A value in parentheses is used when early detection is turned on.

19.3.5 Chip-Select Timing Trim

Figure 19-5 shows the timing diagram for the chip-select timing trim. The signal values and units of measure for this figure are found in Table 19-7. For detailed information about the individual signals, see Chapter 6, "Chip-Select Logic."

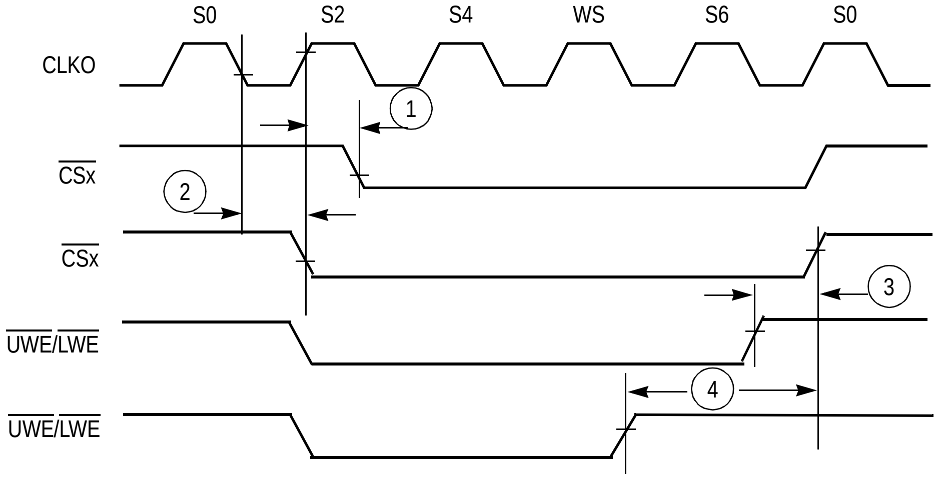


Figure 19-5. Chip-Select Timing Trim Timing Diagram

Table 19-7. Chip-Select Timing Trim Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	CLKO high to $\overline{\text{CSx}}$ asserted (bit ECDS = 0)	—	10	ns
2	CLKO low to $\overline{\text{CSx}}$ asserted (bit ECDS = 1)	—	10	ns
3	$\overline{\text{UWE}}$/$\overline{\text{LWE}}$ negated before $\overline{\text{CSx}}$ is negated (bit WPEXT = 0)	10	20	ns
4	$\overline{\text{UWE}}$/$\overline{\text{LWE}}$ negated before $\overline{\text{CSx}}$ is negated (bit WPEXT = 1)	40	50	ns

19.3.6 DRAM Read Cycle 16-Bit Access (CPU Bus Master)

Figure 19-6 on page 19-9 shows the DRAM read cycle timing diagram for 16-bit access (CPU bus master). The signal values and units of measure for this figure are found in Table 19-8 on page 19-9. Detailed information about the operation of individual signals can be found in Chapter 7, "DRAM Controller," and Chapter 6, "Chip-Select Logic."

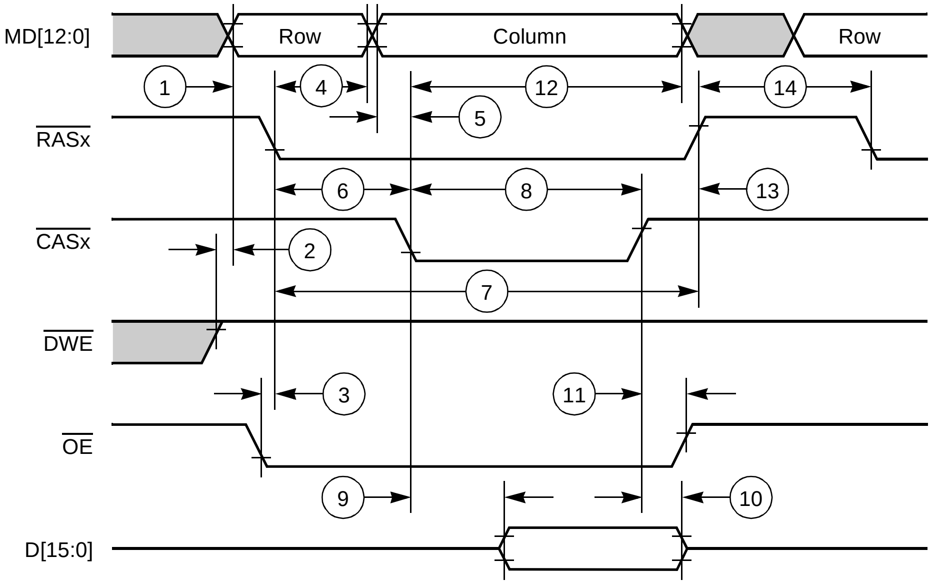


Figure 19-6. DRAM Read Cycle 16-Bit Access (CPU Bus Master) Timing Diagram

Table 19-8. DRAM Read Cycle 16-Bit Access (CPU Bus Master) Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Row address valid to RASx asserted	40	—	ns
2	DWE negated before row address valid	0	—	ns
3	OE asserted before RASx is asserted	0	—	ns
4	RASx asserted before row address invalid (MSW = 0,1)	12,27	—	ns
5	Column address valid to CASx asserted (MSW = 0,1)	10,25	—	ns
6	RASx asserted to CASx asserted (MSW = 0,1)	28,58	32	ns
7	RASx pulse width (SLW = 0,1)	90,120	—	ns
8	CASx pulse width (BC[1:0] = 00,01,10,11)	28,58,88,118	—	ns
9	CASx asserted to data-in valid (BC[1:0] = 00,01,10,11 for FPM)	—	15,45,75,105 (FPM) 20 (EDO)	ns
10	Data-in hold after CASx is negated	0 (FPM) 30 (EDO)	—	ns
11	OE negated after CASx is negated	0 (FPM) 30 (EDO)	35	ns

Table 19-8. DRAM Read Cycle 16-Bit Access (CPU Bus Master) Timing Parameters (Continued)

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
12	$\overline{\text{CASx}}$ asserted before column address invalid	50	—	ns
13	$\overline{\text{RASx}}$ negated after $\overline{\text{CASx}}$ is negated	28	—	ns
14	$\overline{\text{RASx}}$ precharge time (SLW= 0,1)	58,118	—	ns

Note: $\overline{\text{RASx}}$ stands for $\overline{\text{RAS0}}$ and $\overline{\text{RAS1}}$. $\overline{\text{CASx}}$ stands for $\overline{\text{CAS0}}$ and $\overline{\text{CAS1}}$.

Note: MSW is bit 5, SLW is bit 3, and BC[1:0] comprises bits 13–12 in the DRAMC register. When the table identifies these bits, the sequence of their listed values corresponds to the sequence of timing data provided.

19.3.7 DRAM Write Cycle 16-Bit Access (CPU Bus Master)

Figure 19-7 shows the DRAM write cycle timing diagram for 16-bit access (CPU bus master). The signal values and units of measure for this figure are found in Table 19-9 on page 19-11. Detailed information about the operation of individual signals can be found in Chapter 7, "DRAM Controller," and Chapter 6, "Chip-Select Logic."

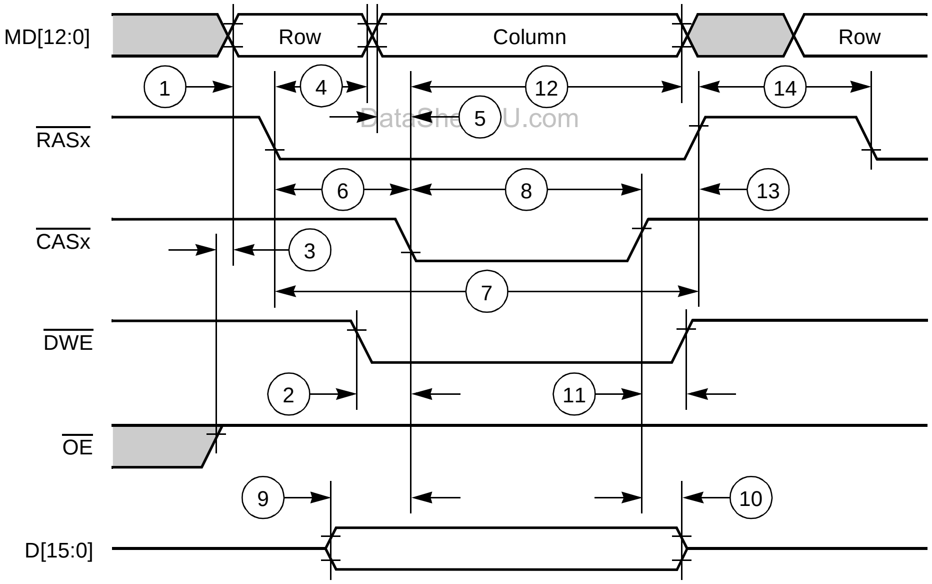


Figure 19-7. DRAM Write Cycle 16-Bit Access (CPU Bus Master) Timing Diagram

Table 19-9. DRAM Write Cycle 16-Bit Access (CPU Bus Master) Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Row address valid to $\overline{\text{RASx}}$ asserted	40	—	ns
2	$\overline{\text{DWE}}$ asserted before $\overline{\text{CASx}}$ asserted	25	—	ns
3	$\overline{\text{OE}}$ negated before $\overline{\text{RASx}}$ asserted	0	—	ns
4	$\overline{\text{RASx}}$ asserted before row address invalid (MSW = 0,1)	12,27	—	ns
5	Column address valid to $\overline{\text{CASx}}$ asserted (MSW = 0,1)	10,25	—	ns
6	$\overline{\text{RASx}}$ asserted to $\overline{\text{CASx}}$ asserted (MSW = 0,1)	28,58	—	ns
7	$\overline{\text{RASx}}$ pulse width (SLW = 0,1)	90,120	—	ns
8	$\overline{\text{CASx}}$ pulse width (BC[1:0] = 00,01,10,11)	28,58,88,118	—	ns
9	Data-out valid before $\overline{\text{CASx}}$ asserted	25	—	ns
10	Data-out hold after $\overline{\text{CASx}}$ negated	25	—	ns
11	$\overline{\text{DWE}}$ negated after $\overline{\text{CASx}}$ negated	0	—	ns
12	$\overline{\text{CASx}}$ asserted before column address invalid	50	—	ns
13	$\overline{\text{RASx}}$ negated after $\overline{\text{CASx}}$ negated	28	—	ns
14	$\overline{\text{RASx}}$ precharge time (SLW = 0,1)	50,118	—	ns

Note: $\overline{\text{RASx}}$ stands for $\overline{\text{RAS0}}$ and $\overline{\text{RAS1}}$. $\overline{\text{CASx}}$ stands for $\overline{\text{CAS0}}$ and $\overline{\text{CAS1}}$.

Note: MSW is bit 5, SLW is bit 3, and BC[1:0] comprises bits 13–12 in the DRAMC register. When the table identifies these bits, the sequence of their listed values corresponds to the sequence of timing data provided.

19.3.8 DRAM Hidden Refresh Cycle (Normal Mode)

Figure 19-8 on page 19-12 shows the DRAM hidden refresh cycle timing diagram for normal mode. The signal values and units of measure for this figure are found in Table 19-10 on page 19-12. Detailed information about the operation of individual signals can be found in Chapter 7, “DRAM Controller.”

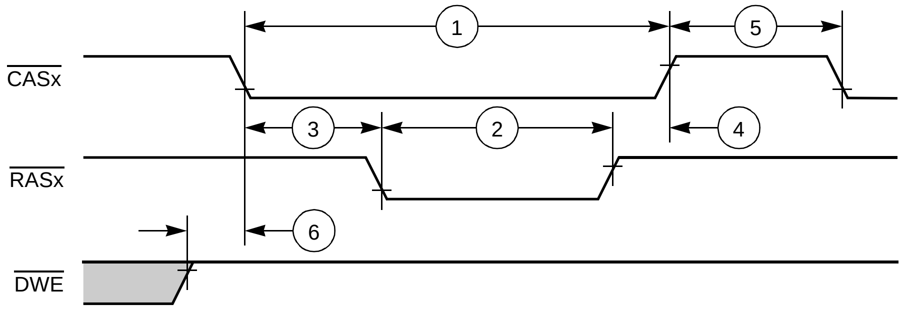


Figure 19-8. DRAM Hidden Refresh Cycle (Normal Mode) Timing Diagram

Table 19-10. DRAM Hidden Refresh Cycle (Normal Mode) Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	$\overline{\text{CASx}}$ pulse width	88	—	ns
2	$\overline{\text{RASx}}$ pulse width	88	—	ns
3	$\overline{\text{CASx}}$ asserted to $\overline{\text{RASx}}$ asserted	28	32	ns
4	$\overline{\text{RASx}}$ negated to $\overline{\text{CASx}}$ negated	-28	—	ns
5	$\overline{\text{CASx}}$ negated to next $\overline{\text{CASx}}$ asserted	88	—	ns
6	$\overline{\text{DWE}}$ negated before $\overline{\text{CASx}}$ asserted	58	—	ns
Note: $\overline{\text{RASx}}$ stands for $\overline{\text{RAS0}}$ and $\overline{\text{RAS1}}$. $\overline{\text{CASx}}$ stands for $\overline{\text{CAS0}}$ and $\overline{\text{CAS1}}$.				

19.3.9 DRAM Hidden Refresh Cycle (Low-Power Mode)

Figure 19-9 shows the DRAM hidden refresh cycle timing diagram for low-power mode. The signal values and units of measure for this figure are found in Table 19-11 on page 19-13. Detailed information about the operation of individual signals can be found in Chapter 7, “DRAM Controller.”

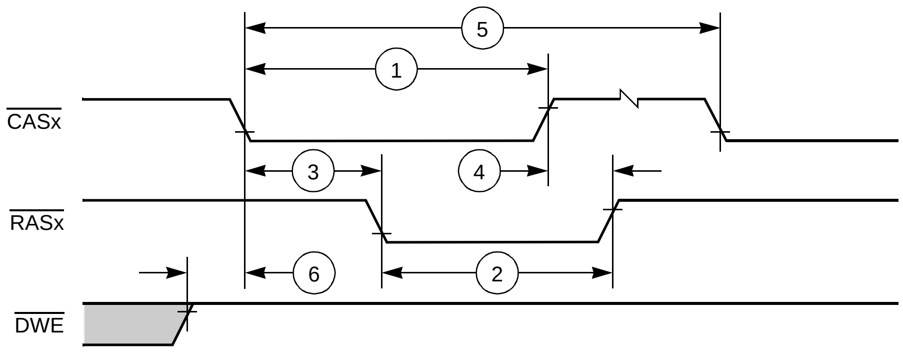


Figure 19-9. DRAM Hidden Refresh Cycle (Low-Power Mode) Timing Diagram

Table 19-11. DRAM Hidden Refresh Cycle (Low-Power Mode) Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	CASx pulse width	120	—	ns
2	RASx pulse width	120	—	ns
3	CASx asserted to RASx asserted	30	—	ns
4	CASx negated to RASx negated	30	—	ns
5	Refresh cycle (using 32.768 KHz crystal)	15	—	us
5	Refresh cycle (using 38.400 KHz crystal)	13	—	us
6	DWE negated before CASx asserted	58	—	ns
Note: RASx stands for RAS0 and RAS1. CASx stands for CAS0 and CAS1.				

19.3.10 LCD SRAM/ROM DMA Cycle 16-Bit Mode Access (1 Wait State)

Figure 19-10 shows the LCD SRAM/ROM DMA cycle timing diagram for 16-bit access (1 wait state). 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 19-12 on page 19-14. Detailed information about the operation of individual signals can be found in Chapter 7, “DRAM Controller,” and Chapter 8, “LCD Controller.”

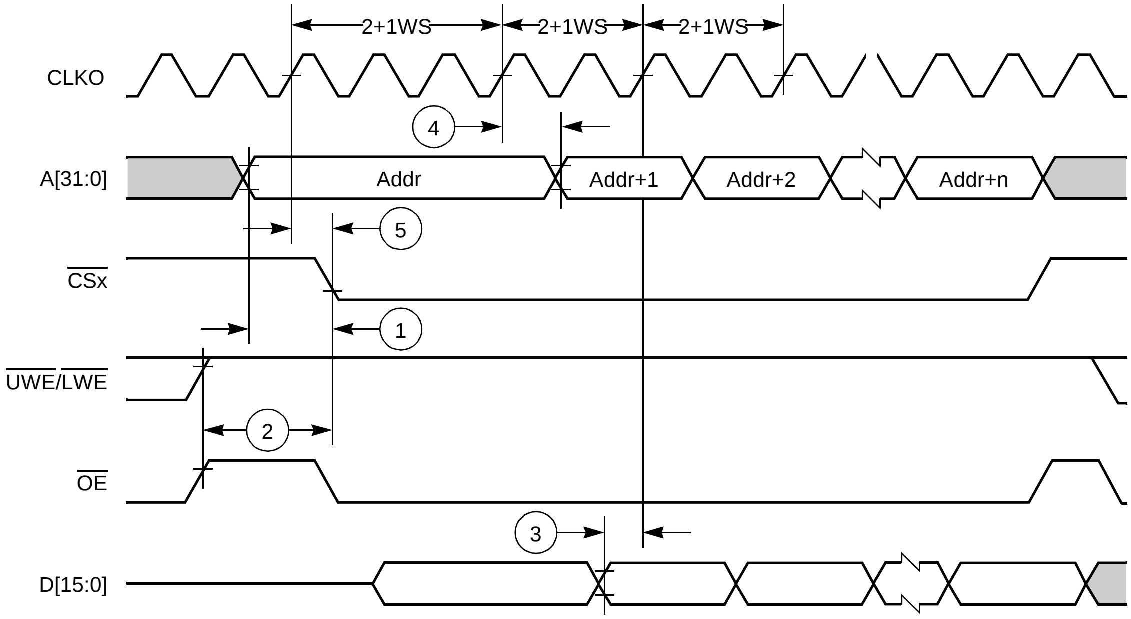


Figure 19-10. LCD SRAM/ROM DMA Cycle 16-Bit Mode Access Timing Diagram

Table 19-12. LCD SRAM/ROM DMA Cycle 16-Bit Mode Access Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Address valid to $\overline{\text{CSx}}$ asserted	20	—	ns
2	$\overline{\text{UWE}}$/$\overline{\text{LWE}}$ to $\overline{\text{CSx}}$ asserted	28	—	ns
3	Data setup time	16	—	ns
4	CLKO to address valid	—	10	ns
5	CLKO high to $\overline{\text{CSx}}$	—	10	ns

19.3.11 LCD DRAM DMA Cycle 16-Bit EDO RAM Mode Access (LCD Bus Master)

Figure 19-11 shows the timing diagram for the LCD DRAM DMA cycle for 16-bit EDO RAM mode access (LCD bus master). The signal values and units of measure for this figure are found in Table 19-13 on page 19-15. Detailed information about the operation of individual signals can be found in Chapter 7, “DRAM Controller,” and Chapter 8, “LCD Controller.”

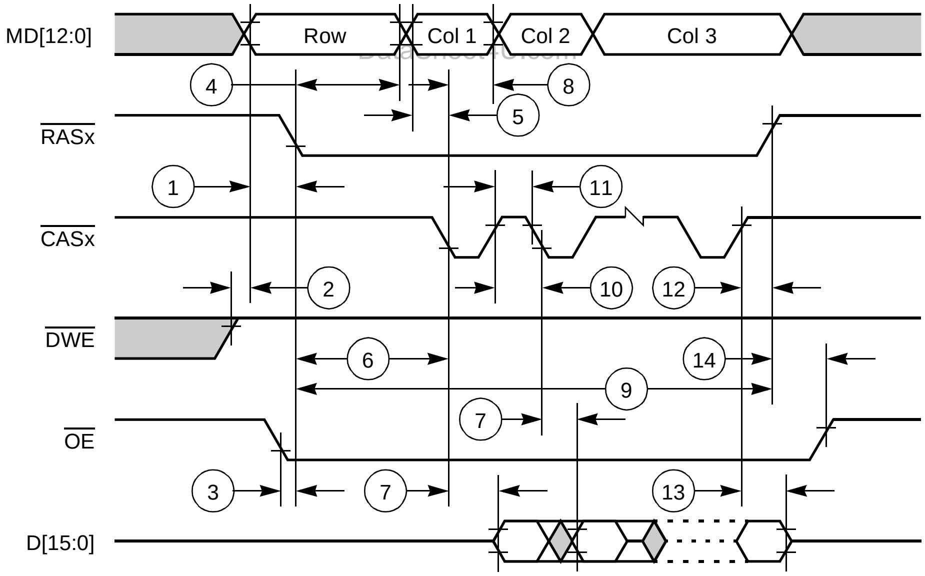


Figure 19-11. LCD DRAM DMA Cycle 16-Bit EDO RAM Mode Access (LCD Bus Master) Timing Diagram

Table 19-13. LCD DRAM DMA Cycle 16-Bit EDO RAM Mode Access (LCD Bus Master) Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Row address valid to $\overline{\text{RASx}}$ asserted	45	—	ns
2	$\overline{\text{DWE}}$ negated before row address valid	0	—	ns
3	$\overline{\text{OE}}$ asserted before $\overline{\text{RASx}}$ asserted	0	—	ns
4	$\overline{\text{RASx}}$ asserted before row address invalid (MSW = 0,1)	12,27	—	ns
5	Column address valid to $\overline{\text{CASx}}$ asserted (MSW = 0,1)	10,25	—	ns
6	$\overline{\text{RASx}}$ asserted to $\overline{\text{CASx}}$ asserted (MSW = 0,1)	28,58	—	ns
7	$\overline{\text{CASx}}$ asserted to data-in valid	—	20	ns
8	$\overline{\text{CASx}}$ asserted before column address invalid	20	—	ns
9	$\overline{\text{RASX}}$ pulse width	(2N + 1)T	—	ns
10	$\overline{\text{CASx}}$ pulse width	28	—	ns
11	$\overline{\text{CASx}}$ precharge time	26	—	ns
12	$\overline{\text{RASx}}$ negated to $\overline{\text{CASx}}$ negated	-28	—	ns
13	Data-in hold after $\overline{\text{CASx}}$ negated	30	—	ns
14	$\overline{\text{OE}}$ negated after $\overline{\text{CASx}}$ negated	28	32	ns

Note:
N is the number of words in one DMA transfer.
T is the system clock period.
$\overline{\text{RASx}}$ stands for $\overline{\text{RAS0}}$ and $\overline{\text{RAS1}}$. $\overline{\text{CASx}}$ stands for $\overline{\text{CAS0}}$ and $\overline{\text{CAS1}}$.
MSW is bit 5 in the DRAMC register. When this bit is set to 0, the first timing number applies; when it is set to 1, the second timing number applies.

19.3.12 LCD DRAM DMA Cycle 16-Bit Fast Page Mode Access (LCD Bus Master)

Figure 19-12 shows the timing diagram for the LCD DRAM DMA cycle for 16-bit Fast Page Mode mode access (LCD bus master). The signal values and units of measure for this figure are found in Table 19-14. Detailed information about the operation of individual signals can be found in Chapter 7, “DRAM Controller,” and Chapter 8, “LCD Controller.”

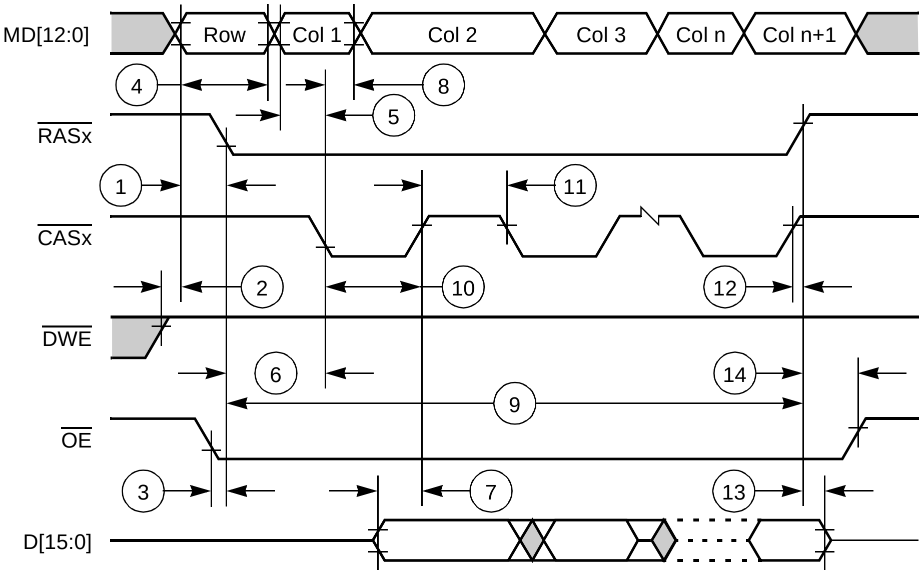


Figure 19-12. LCD DRAM DMA Cycle 16-Bit Fast Page Mode Access (LCD Bus Master) Timing Diagram

Table 19-14. LCD DRAM DMA Cycle 16-Bit Fast Page Mode Access (LCD Bus Master) Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Row address valid to RASx asserted	45	—	ns
2	DWE negated before row address valid	0	—	ns
3	OE asserted before RASx asserted	0	—	ns
4	RASx asserted before row address invalid (MSW = 0,1)	12,27	—	ns
5	Column address valid to CASx asserted (MSW = 0,1)	10,25	—	ns
6	RASx asserted to CASx asserted (MSW = 0,1)	28,58	—	ns
7	Data setup time	15	—	ns
8	CASx asserted before column address invalid	20	—	ns

Table 19-14. LCD DRAM DMA Cycle 16-Bit Fast Page Mode Access (LCD Bus Master) Timing Parameters (Continued)

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
9	$\overline{\text{RASx}}$ pulse width	(2N + 1)T	—	ns
10	$\overline{\text{CASx}}$ pulse width (BC[1:0] = 00,01,10,11 in FPM)	28,58,88,118	—	ns
11	$\overline{\text{CASx}}$ precharge time	26	—	ns
12	$\overline{\text{RASx}}$ negated to $\overline{\text{CASx}}$ negated	-28	—	ns
13	Data-in hold after $\overline{\text{CASx}}$ negated	0	—	ns
14	$\overline{\text{OE}}$ negated after $\overline{\text{CASx}}$ negated	0	2	ns

Note:
N is the number of words in one DMA transfer.
T is the system clock period.
$\overline{\text{RASx}}$ stands for $\overline{\text{RAS0}}$ and $\overline{\text{RAS1}}$. $\overline{\text{CASx}}$ stands for $\overline{\text{CAS0}}$ and $\overline{\text{CAS1}}$.
MSW is bit 5 and BC[1:0] comprises bits 13–12 in the DRAMC register. When the table identifies these bits, the sequence of their listed values corresponds to the sequence of timing data provided.

19.3.13 LCD Controller Timing

Figure 19-13 shows the LCD controller timing diagram for normal mode, and Figure 19-14 on page 19-18 displays the timing diagram for self-refresh mode. The signal values and units of measure for both figures are found in Table 19-15 on page 19-18. Detailed information about the operation of individual signals can be found in Chapter 8, “LCD Controller,” and Chapter 7, “DRAM Controller.”

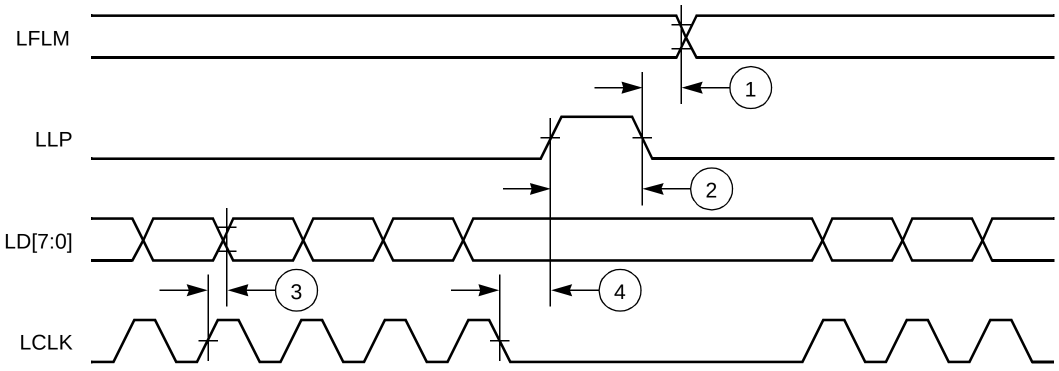


Figure 19-13. LCD Controller Timing Diagram (Normal Mode)

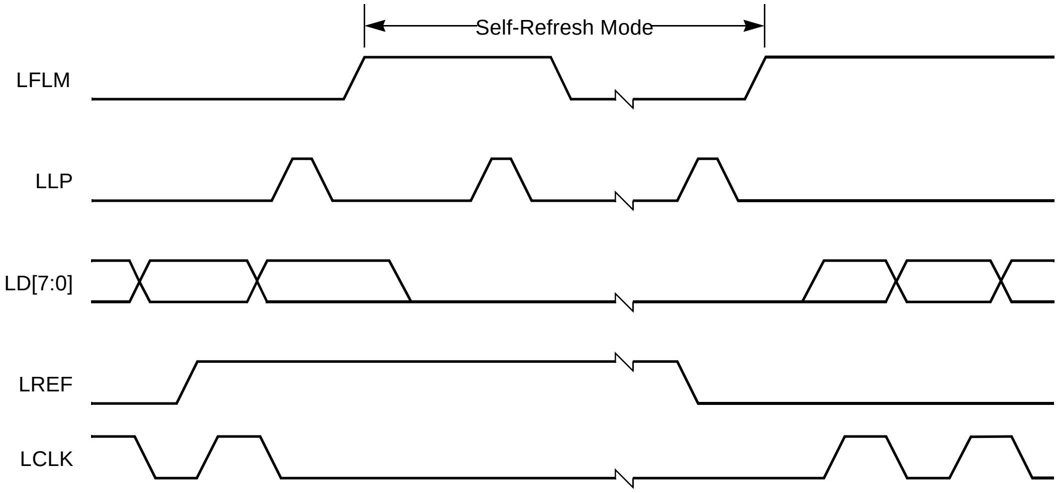


Figure 19-14. LCD Controller Timing Diagram (Self-Refresh Mode)

Table 19-15. LCD Controller Timing Parameters

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Line pulse to frame signal	(4 * pixclk) - 2	—	ns
2	Line pulse width	(4 * pixclk) - 2	—	ns
3	LCLK to data valid	-2	2	ns
4	Shift clock to line pulse	(2 * pixclk) - 2	(2 * pixclk) + 2	ns

Note:
The preceding data is measured by summing the polarity bits LFLM, LLP, and LCLK in the POLCF register.
The variable pixclk = LCD_CLK / (pcd + 1).
The self-refresh mode timing between LFRM, LSCLK, LD, and LLP are the same as in normal mode.
The self-refresh mode is entered and exited on the positive edge of LFRM.
In self-refresh mode, the LFRM and LLP waveforms are identical to the waveforms in normal mode, while LD and LCLK remain in inactive level.

19.3.14 Page-Miss SDRAM CPU Read Cycle (CAS Latency = 1)

Figure 19-15 shows the timing diagram for the page-miss SDRAM CPU read cycle. The signal values and units of measure for this figure are found in Table 19-16 on page 19-31. Detailed information about the operation of individual signals can be found in both Chapter 8, “LCD Controller,” and Chapter 7, “DRAM Controller.”

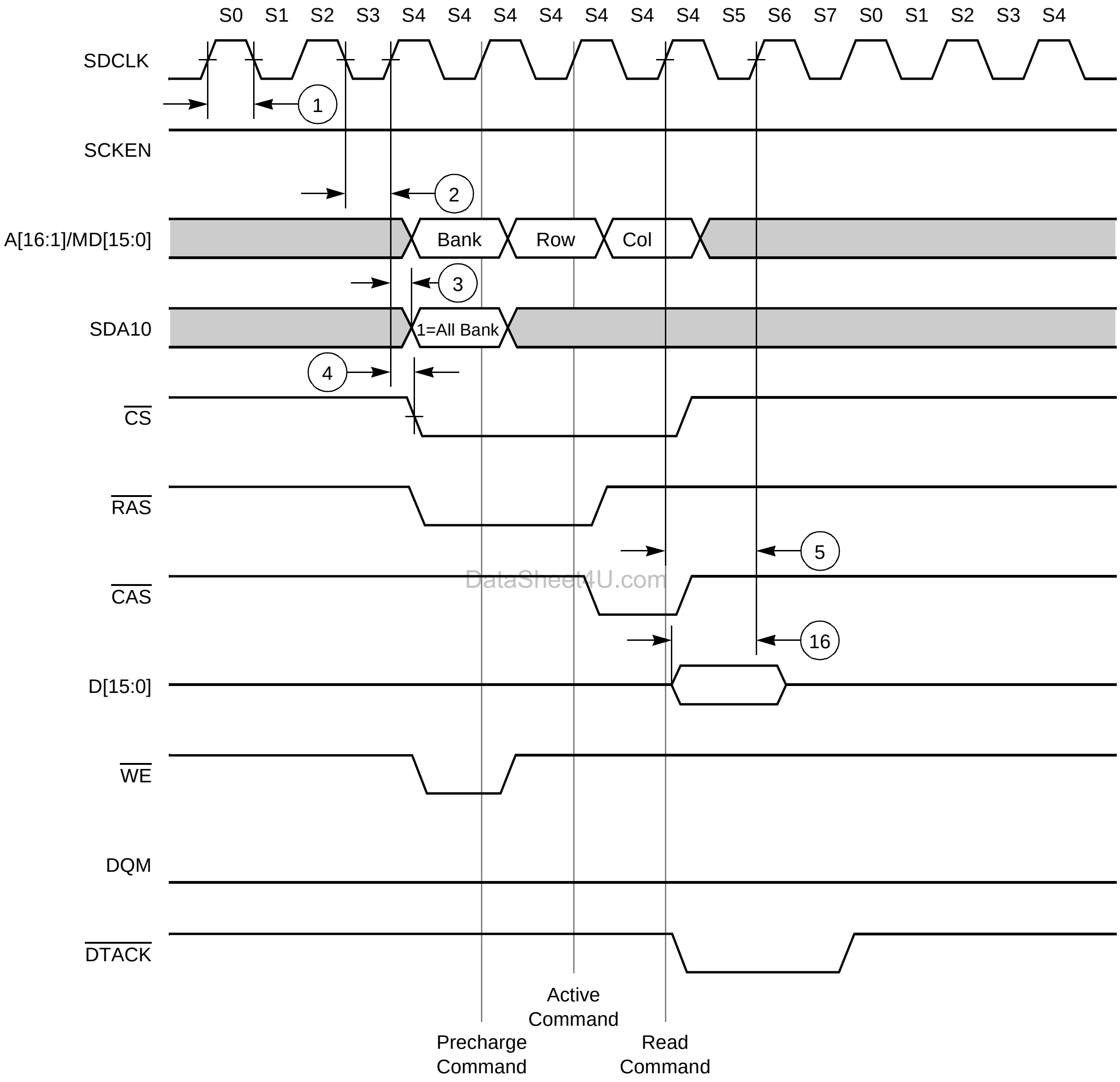


Figure 19-15. Page-Miss SDRAM CPU Read Cycle Timing Diagram

19.3.15 Page-Hit SDRAM CPU Read Cycle (CAS Latency = 1)

Figure 19-16 shows the timing diagram for the page-hit SDRAM CPU read cycle. The signal values and units of measure for this figure are found in Table 19-16 on page 19-31. Detailed information about the operation of individual signals can be found in both Chapter 8, “LCD Controller,” and Chapter 7, “DRAM Controller.”

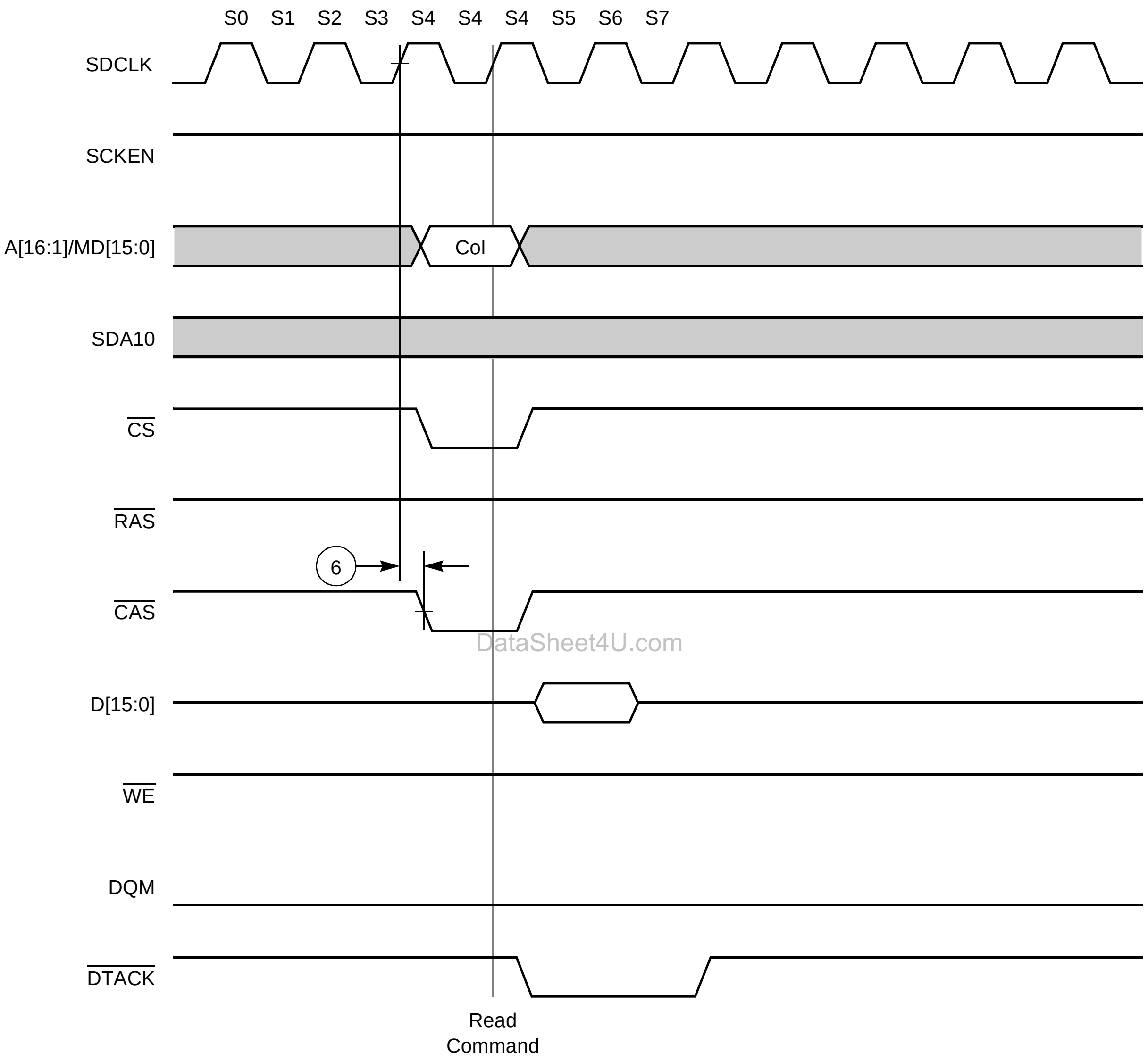


Figure 19-16. Page-Hit SDRAM CPU Read Cycle Timing Diagram

19.3.16 Page-Hit CPU Read Cycle for 8-Bit SDRAM (CAS Latency = 1)

Figure 19-17 shows the timing diagram for the page-hit CPU read cycle for 8-bit SDRAM. The signal values and units of measure for this figure are found in Table 19-16 on page 19-31. Detailed information about the operation of individual signals can be found in both Chapter 8, “LCD Controller,” and Chapter 7, “DRAM Controller.”

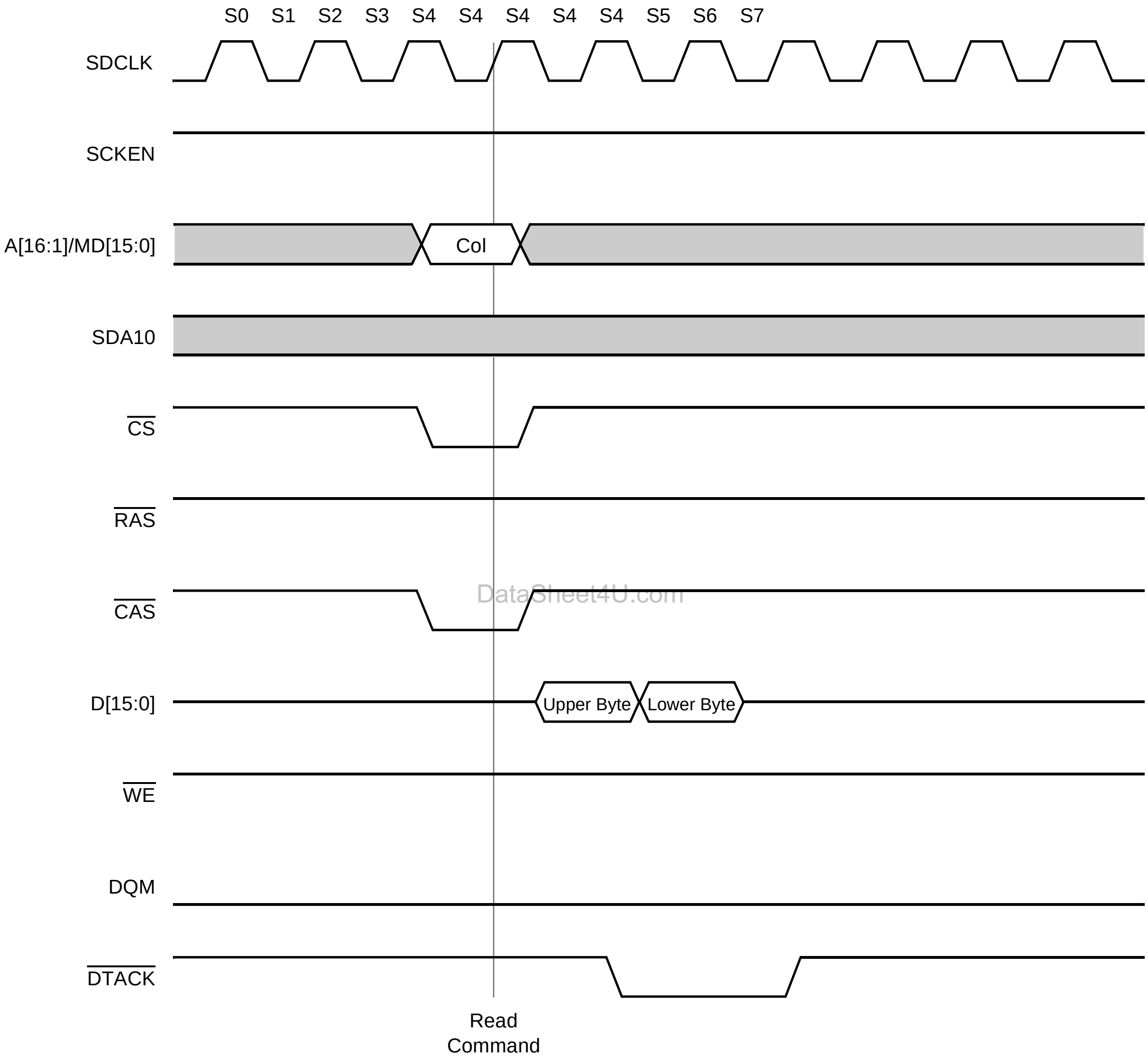


Figure 19-17. Page-Hit CPU Read Cycle for 8-Bit SDRAM Timing Diagram

19.3.17 Page-Miss SDRAM CPU Write Cycle (CAS Latency = 1)

Figure 19-18 shows the timing diagram for the page-miss SDRAM CPU write cycle for 8-bit SDRAM. The signal values and units of measure for this figure are found in Table 19-16 on page 19-31. Detailed information about the operation of individual signals can be found in both Chapter 8, “LCD Controller,” and Chapter 7, “DRAM Controller.”

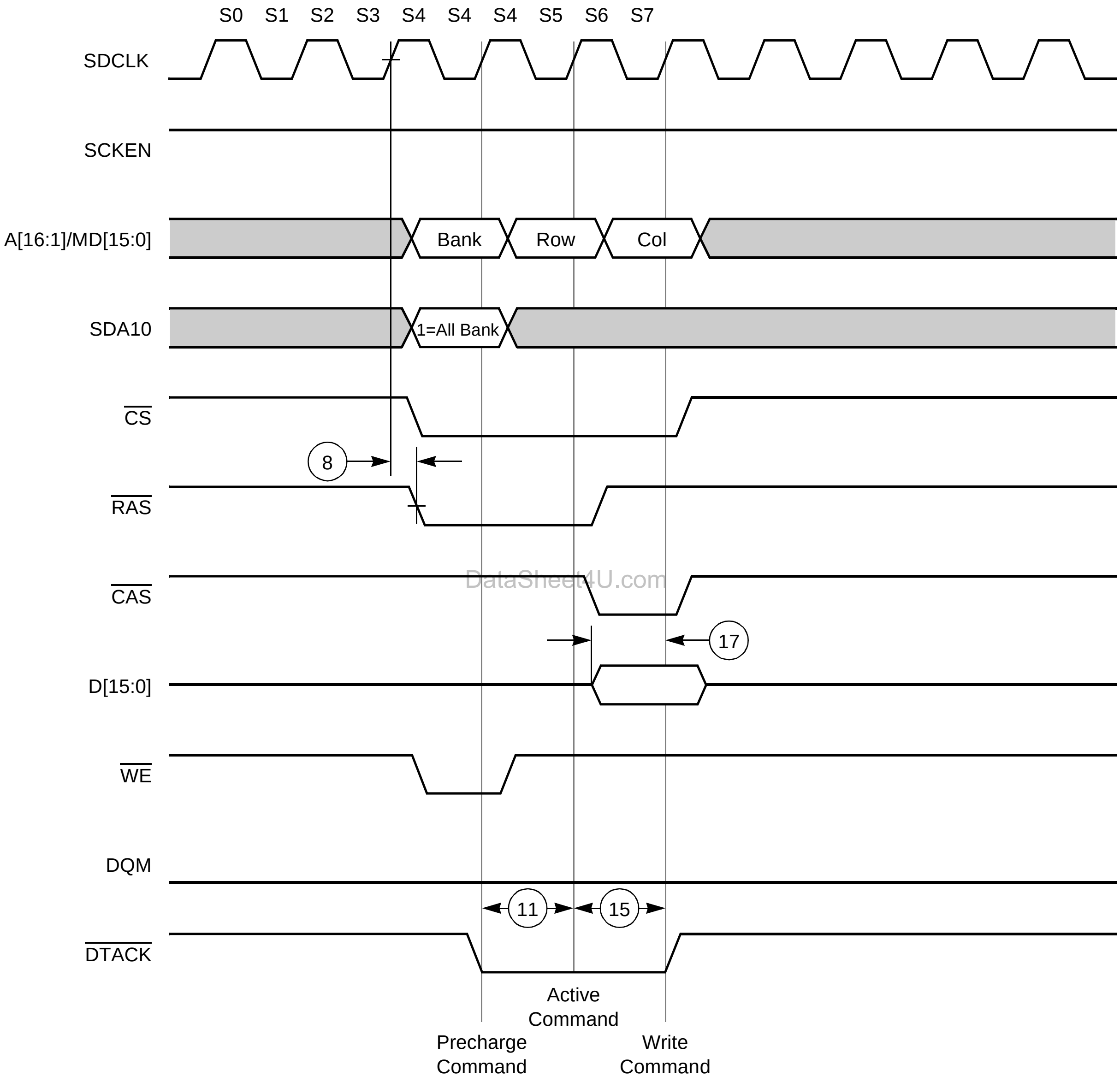


Figure 19-18. Page-Miss SDRAM CPU Write Cycle Timing Diagram

19.3.18 Page-Hit SDRAM CPU Write Cycle (CAS Latency = 1)

Figure 19-19 shows the timing diagram for the page-hit SDRAM CPU write cycle for 8-bit SDRAM. The signal values and units of measure for this figure are found in Table 19-16 on page 19-31. Detailed information about the operation of individual signals can be found in both Chapter 8, “LCD Controller,” and Chapter 7, “DRAM Controller.”

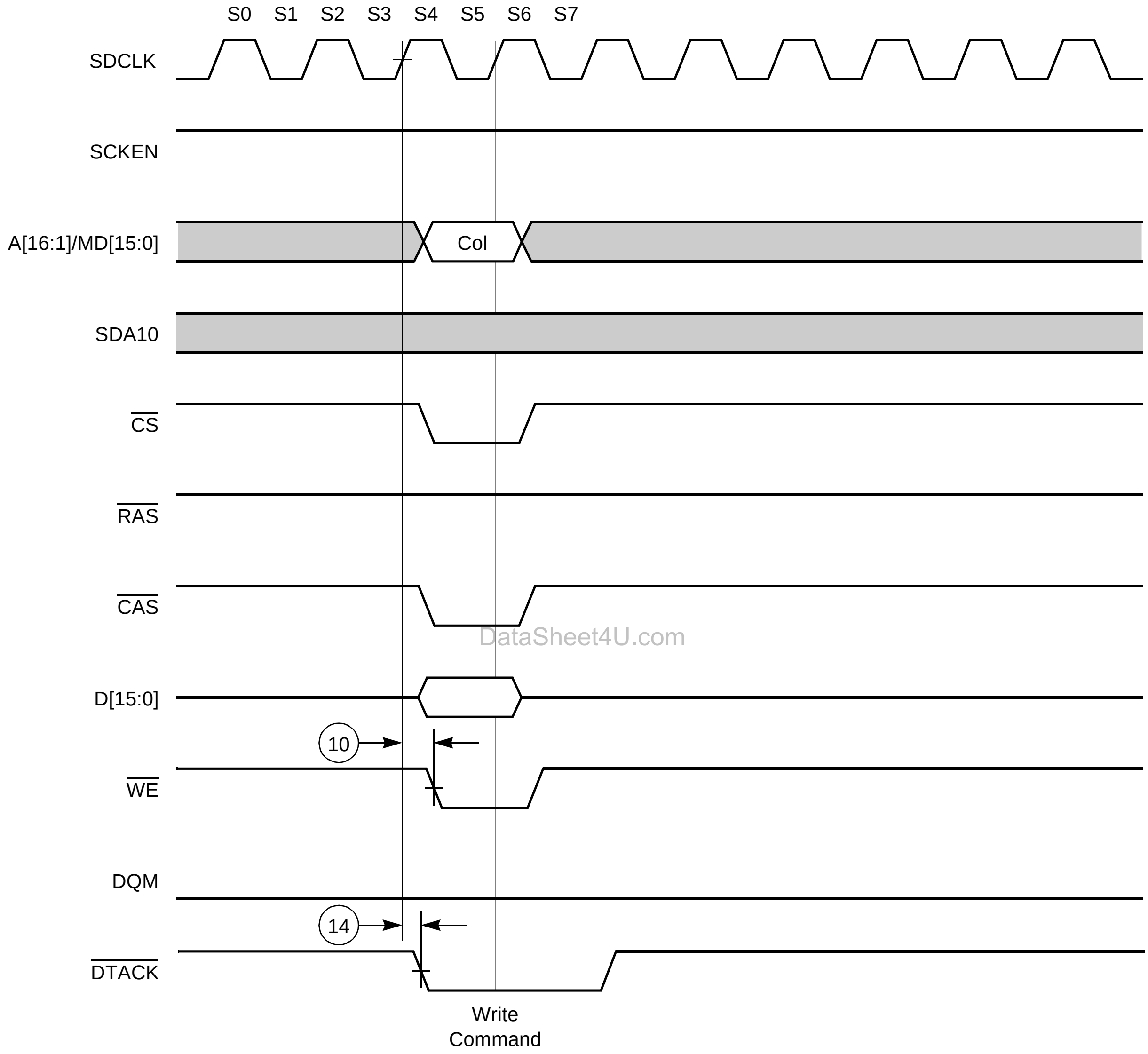


Figure 19-19. Page-Hit SDRAM CPU Write Cycle Timing Diagram

19.3.19 Page-Hit CPU Byte-Write Cycle for 8-Bit SDRAM (CAS Latency = 1)

Figure 19-20 shows the timing diagram for the page-hit SDRAM CPU byte-write cycle for 8-bit SDRAM. The signal values and units of measure for this figure are found in Table 19-16 on page 19-31. Detailed information about the operation of individual signals can be found in both Chapter 8, “LCD Controller,” and Chapter 7, “DRAM Controller.”

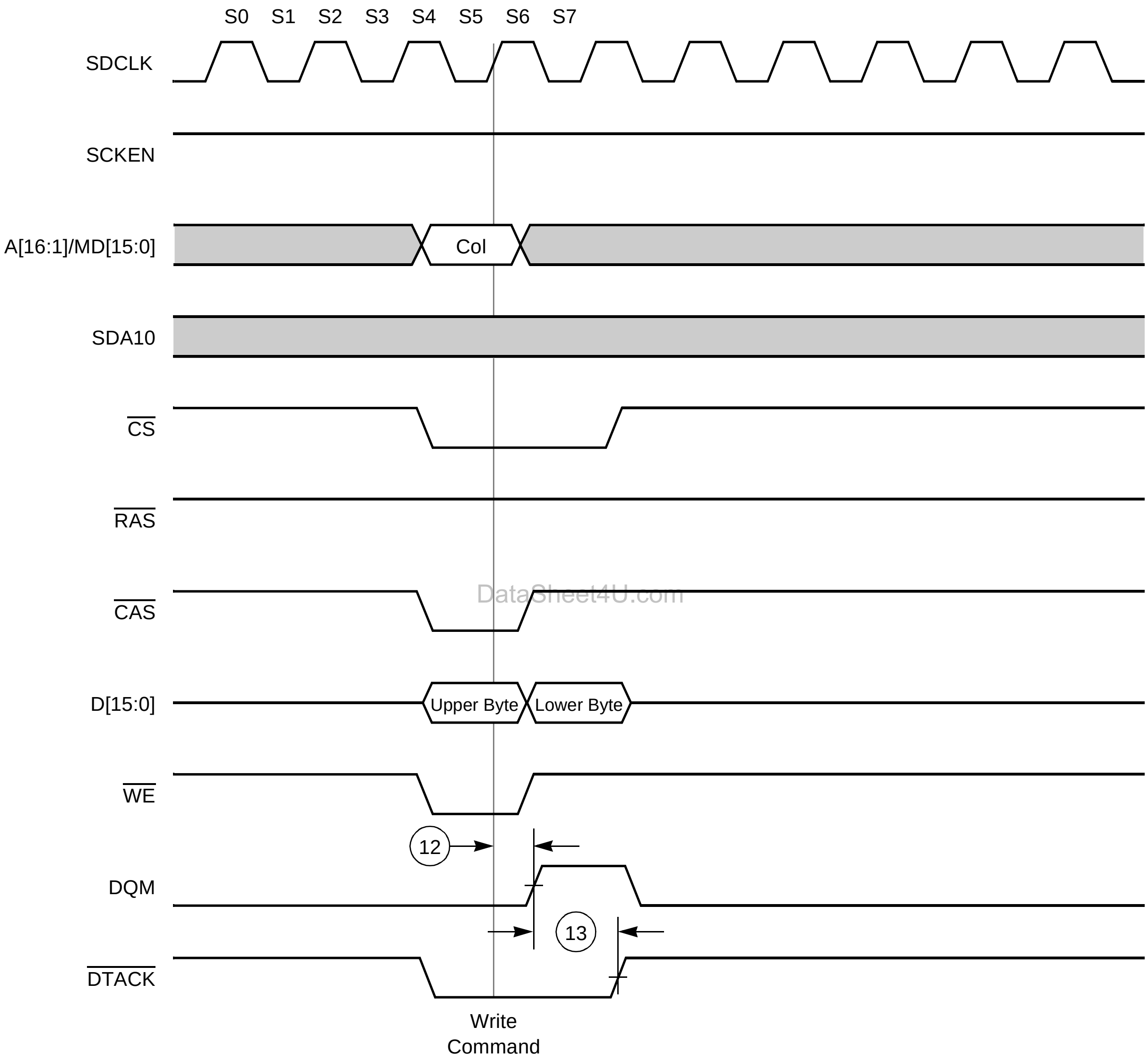


Figure 19-20. Page-Hit CPU Byte-Write Cycle for 8-Bit SDRAM Timing Diagram

19.3.20 Page-Hit CPU Read Cycle in Power-down Mode (CAS Latency = 1, Bit APEN of SDRAM Power-down Register = 1)

Figure 19-21 shows the timing diagram for the page-hit CPU read cycle in power-down mode. The signal values and units of measure for this figure are found in Table 19-16 on page 19-31. Detailed information about the operation of individual signals can be found in both Chapter 8, "LCD Controller," and Chapter 7, "DRAM Controller."

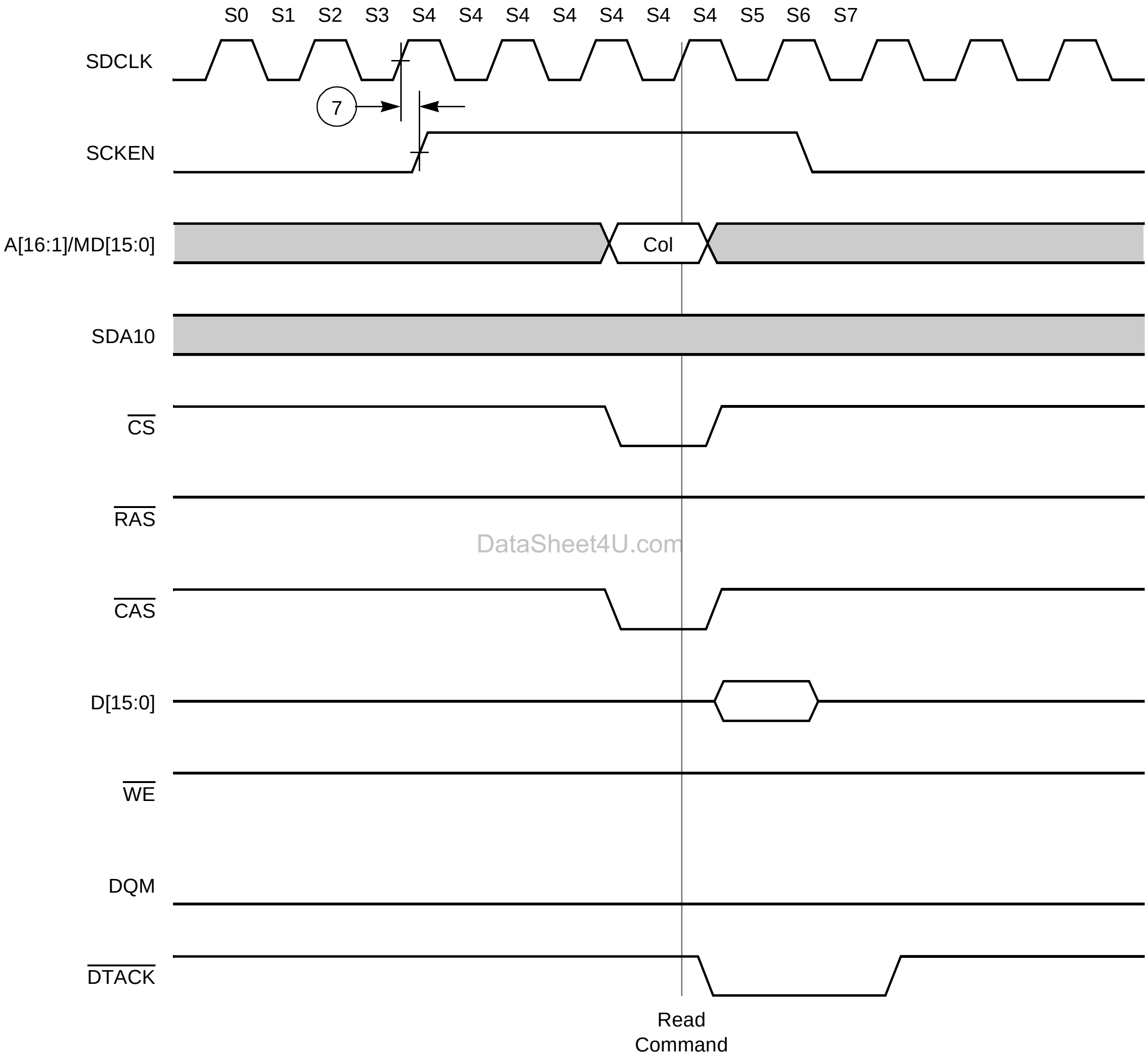


Figure 19-21. Page-Hit CPU Read Cycle in Power-down Mode Timing Diagram

19.3.21 Exit Self-Refresh Due to CPU Read Cycle (CAS Latency = 1, Bit RM of DRAM Control Register = 1)

Figure 19-22 shows the timing diagram for the exit self-refresh due to the CPU read cycle. The signal values and units of measure for this figure are found in Table 19-16 on page 19-31. Detailed information about the operation of individual signals can be found in both Chapter 8, “LCD Controller,” and Chapter 7, “DRAM Controller.”

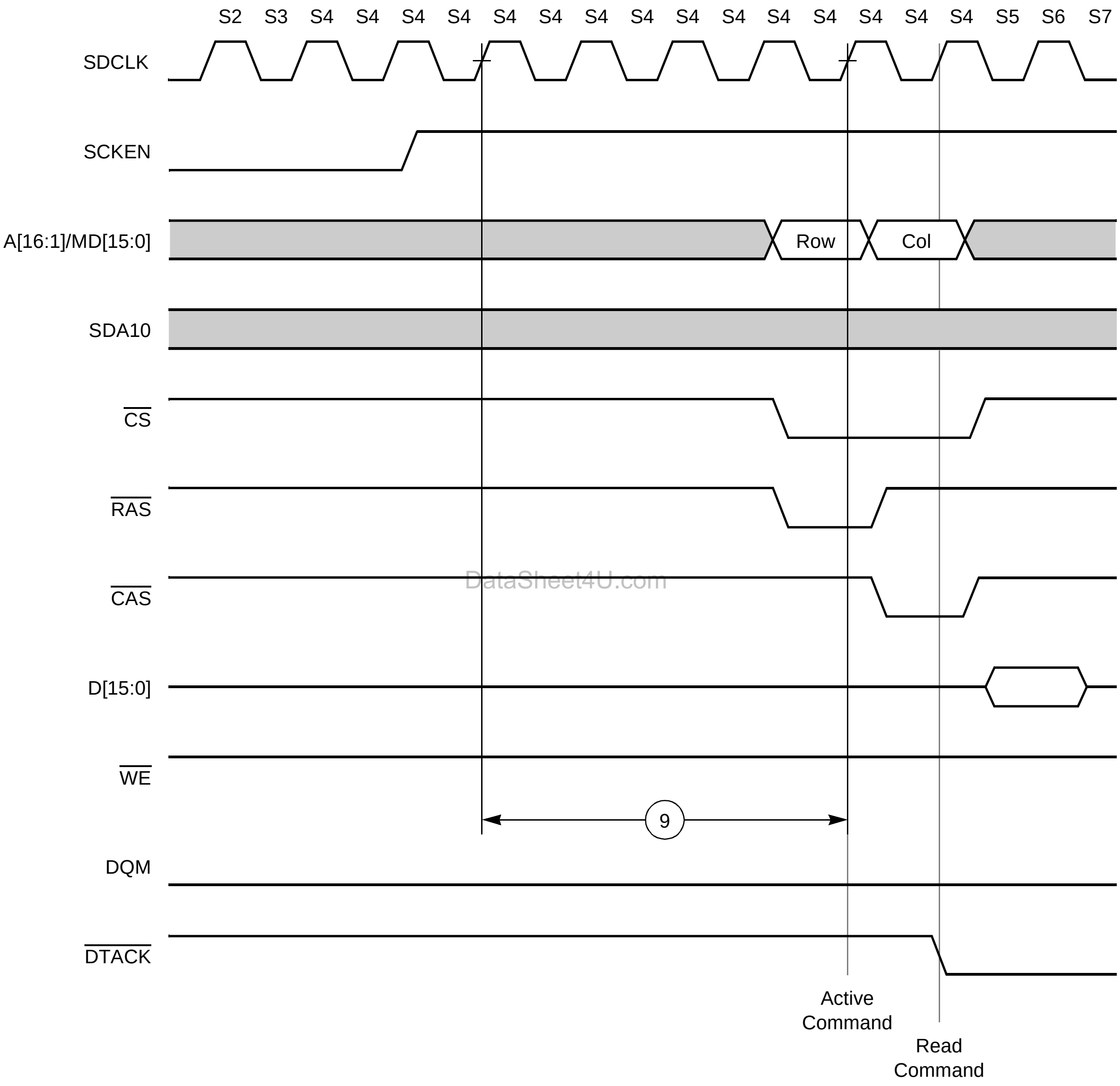


Figure 19-22. Exit Self-Refresh Due to CPU Read Cycle Timing Diagram

19.3.22 Enter Self-Refresh Due to No Activity for 64 Clocks (Bit RM of DRAM Control Register = 1)

Figure 19-23 shows the timing diagram for enter self-refresh due to no activity. The signal values and units of measure for this figure are found in Table 19-16 on page 19-31. Detailed information about the operation of individual signals can be found in both Chapter 8, "LCD Controller," and Chapter 7, "DRAM Controller."

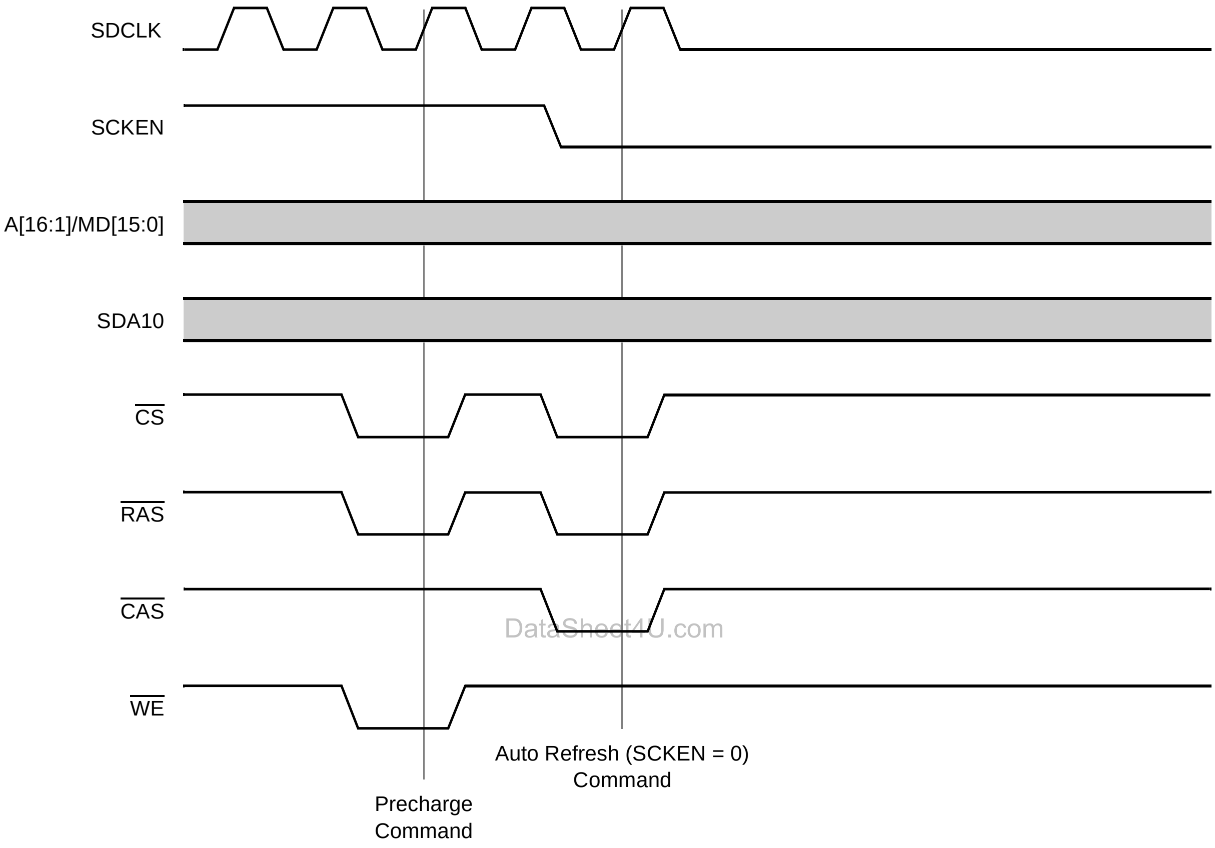


Figure 19-23. Enter Self-Refresh Due to No Activity Timing Diagram

19.3.23 Page-Miss at Starting of LCD DMA for SDRAM (CAS Latency = 1)

Figure 19-24 shows the timing diagram for the page-miss at the starting of LCD DMA for SDRAM. The signal values and units of measure for this figure are found in Table 19-16 on page 19-31. Detailed information about the operation of individual signals can be found in both Chapter 8, “LCD Controller,” and Chapter 7, “DRAM Controller.”

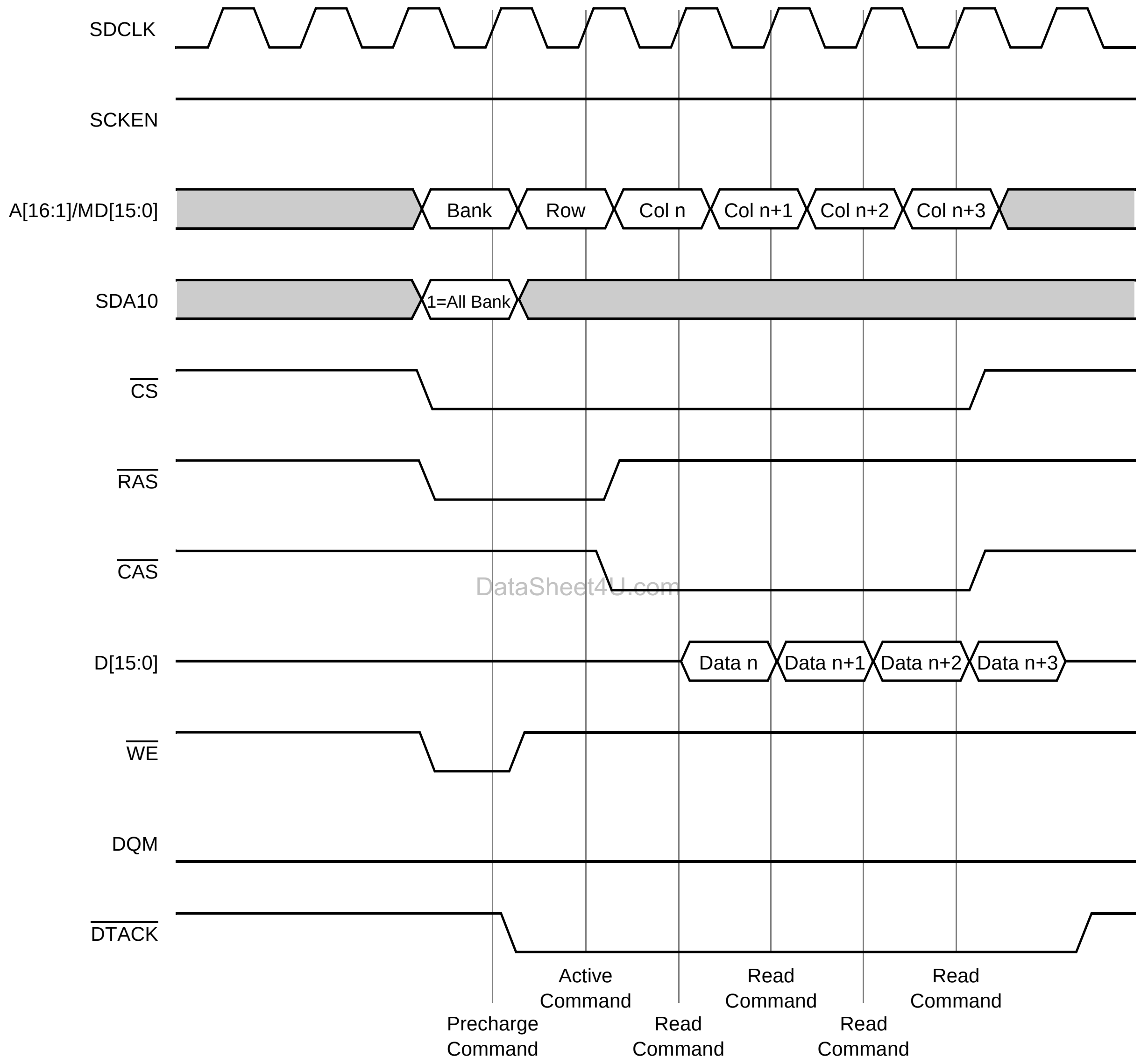


Figure 19-24. Page-Miss at Starting of LCD DMA for SDRAM Timing Diagram

19.3.24 Page-Miss at Start and in Middle of LCD DMA (CAS Latency = 1)

Figure 19-25 shows the timing diagram for the page-miss at the start and in the middle of LCD DMA. The signal values and units of measure for this figure are found in Table 19-16 on page 19-31. Detailed information about the operation of individual signals can be found in both Chapter 8, “LCD Controller,” and Chapter 7, “DRAM Controller.”

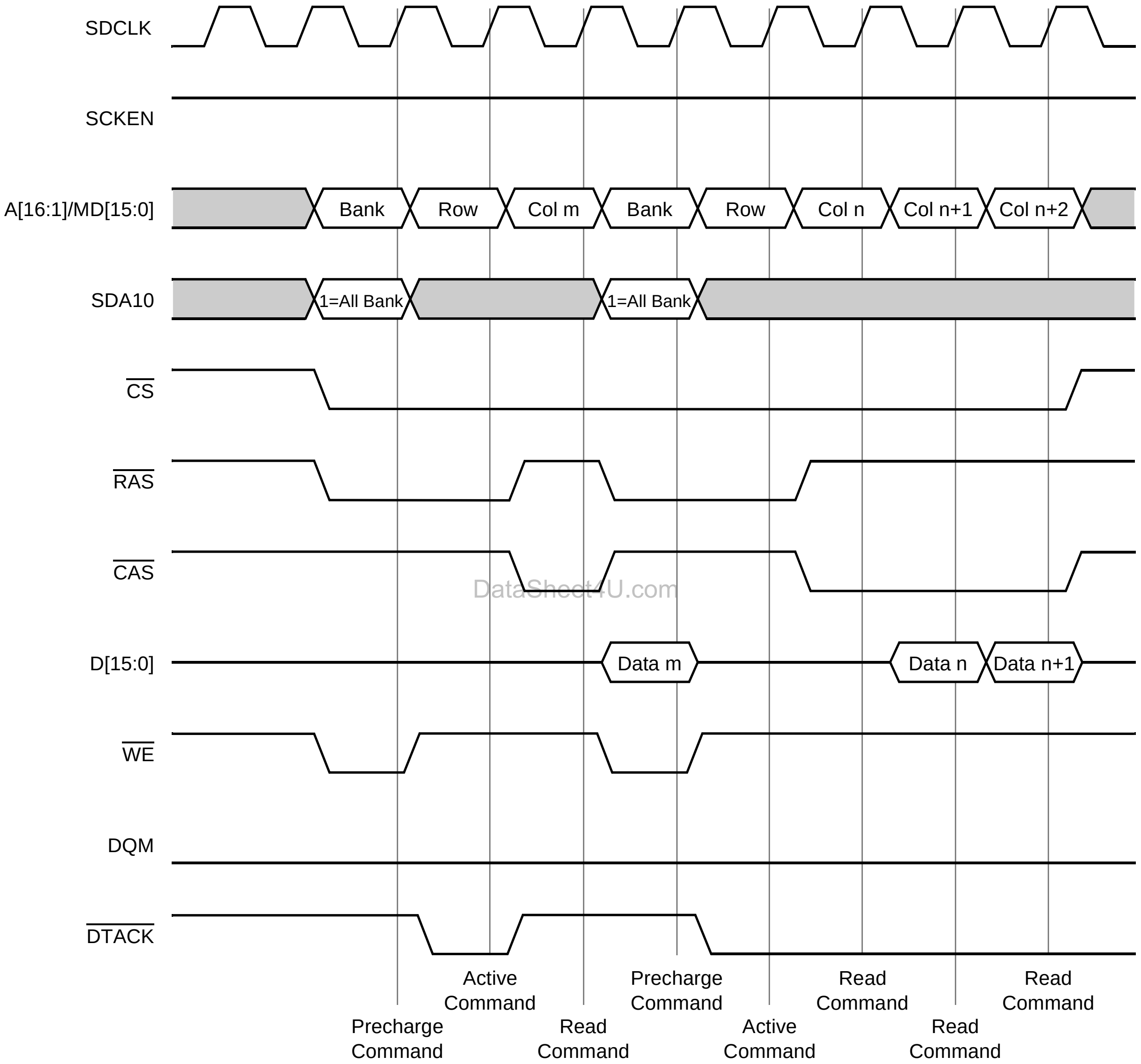


Figure 19-25. Page-Miss at Start and in Middle of LCD DMA Timing Diagram

19.3.25 Page-Hit LCD DMA Cycle for SDRAM (CAS Latency = 1)

Figure 19-26 shows the timing diagram for the page-hit LCD DMA cycle for SDRAM. The signal values and units of measure for this figure are found in Table 19-16 on page 19-31. Detailed information about the operation of individual signals can be found in both Chapter 8, “LCD Controller,” and Chapter 7, “DRAM Controller.”

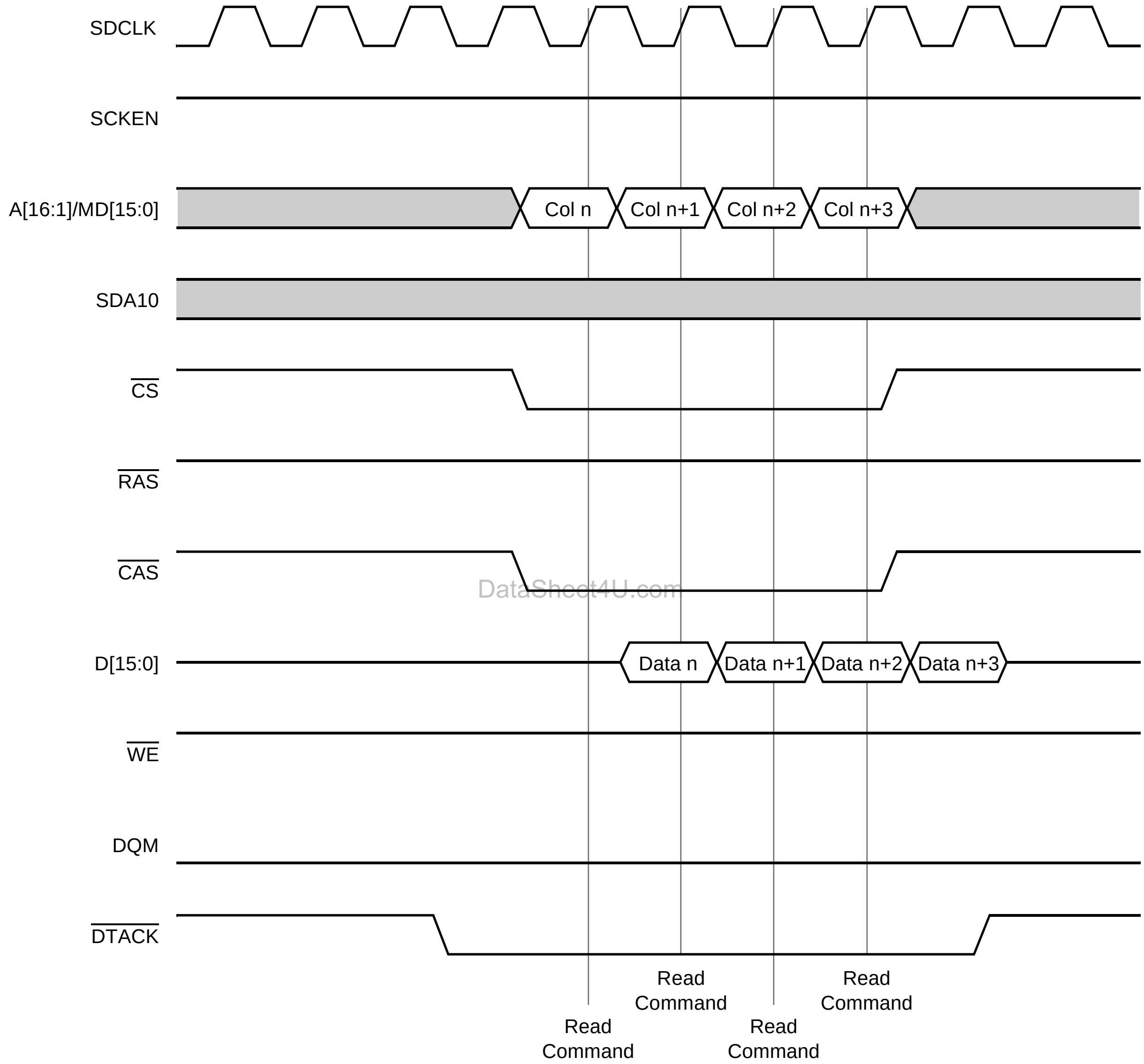


Figure 19-26. Page-Hit LCD DMA Cycle for SDRAM Timing Diagram

Table 19-16. Timing Parameters for Figure 19-15 Through Figure 19-26

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Clock high pulse time	12	—	ns
2	Clock low pulse time	13	—	ns
3	Clock high to address valid	3	13	ns
4	Clock high to chip-select	3	12	ns
5	Read to data sample latency	CAS latency	—	CLK
6	Clock high to $\overline{\text{CAS}}$ asserted	3	12	ns
7	Clock high to SCKEN asserted	8	12	ns
8	Clock high to $\overline{\text{RAS}}$ asserted	3	12	ns
9	Self-refresh exit to active command asserted	4 (7)*	—	CLK
10	Clock high to $\overline{\text{WE}}$ asserted	3	12	ns
11	Precharge command to active command	1 (2)**	—	CLK
12	Clock high to DQM asserted	3	12	ns
13	DQM width asserted	28	—	ns
14	Clock high to $\overline{\text{DTACK}}$ asserted	10	—	ns
15	Active command to read/write command	1 (2)**	—	CLK
16	Data setup time	13	—	ns
17	Data valid to clock high	10	—	ns

*** Note:** The value inside the parentheses is used if the value of bit RACL of the SDRAM control register is 1.

**** Note:** The value inside the parentheses is used if the value of bit CL of the SDRAM control register is 1.

19.3.26 SPI 1 and SPI 2 Generic Timing

Figure 19-27 shows the timing diagram for SPI 1 and SPI 2. The signal values and units of measure for Figure 19-27 through Figure 19-32 are found in Table 19-17 on page 19-34. Detailed information about the operation of individual signals can be found in Chapter 13, “Serial Peripheral Interface 1 and 2.”

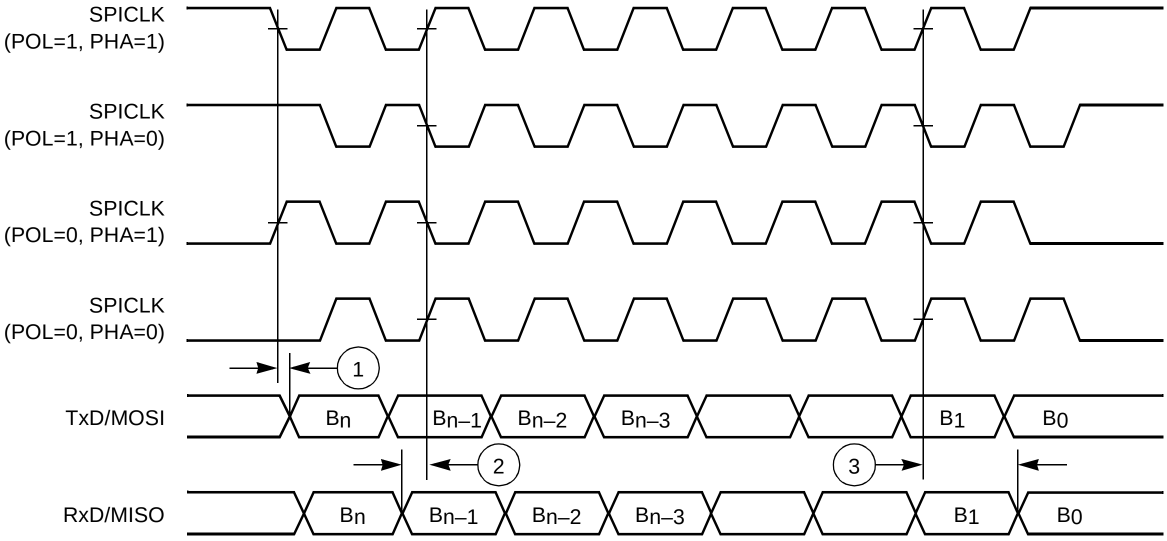


Figure 19-27. SPI 1 and SPI 2 Generic Timing Diagram

19.3.27 SPI 1 Master Using $\overline{\text{DATA\_READY}}$ Edge Trigger

Figure 19-28 shows the timing diagram for the SPI 1 master using the $\overline{\text{DATA\_READY}}$ edge trigger. The signal values and units of measure for Figure 19-27 through Figure 19-32 are found in Table 19-17 on page 19-34. Detailed information about the operation of individual signals can be found in Chapter 13, “Serial Peripheral Interface 1 and 2.”

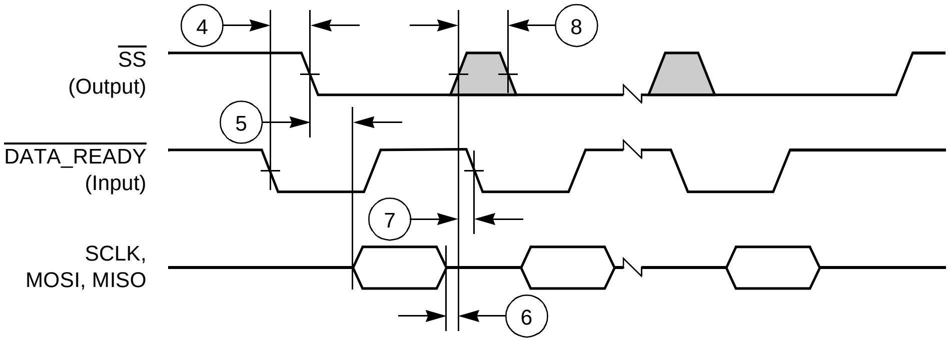


Figure 19-28. SPI 1 Master Using $\overline{\text{DATA\_READY}}$ Edge Trigger Timing Diagram

19.3.28 SPI 1 Master Using DATA_READY Level Trigger

Figure 19-29 shows the timing diagram for the SPI 1 master using the DATA_READY level trigger. The signal values and units of measure for Figure 19-27 through Figure 19-32 are found in Table 19-17 on page 19-34. Detailed information about the operation of individual signals can be found in Chapter 13, "Serial Peripheral Interface 1 and 2."

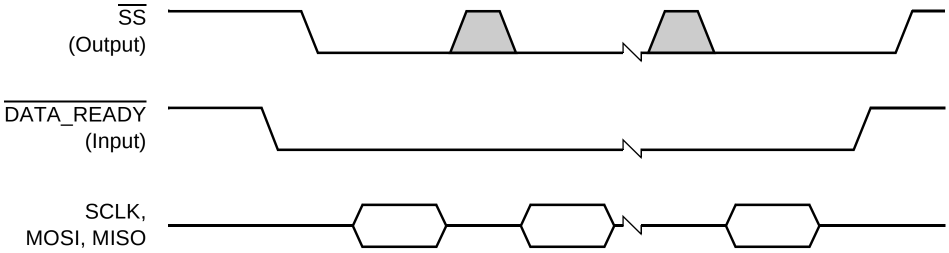


Figure 19-29. SPI 1 Master Using DATA_READY Level Trigger Timing Diagram

19.3.29 SPI 1 Master "Don't Care" DATA_READY

Figure 19-30 shows the timing diagram for the SPI 1 master with DATA_READY "don't care." The signal values and units of measure for Figure 19-27 through Figure 19-32 are found in Table 19-17 on page 19-34. Detailed information about the operation of individual signals can be found in Chapter 13, "Serial Peripheral Interface 1 and 2."

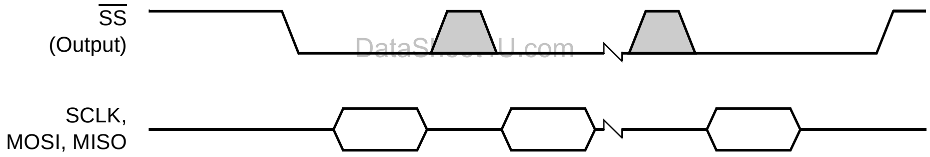


Figure 19-30. SPI 1 Master "Don't Care" DATA_READY Timing Diagram

19.3.30 SPI 1 Slave FIFO Advanced by Bit Count

Figure 19-31 shows the timing diagram for the SPI 1 slave FIFO advanced by bit count. The signal values and units of measure for Figure 19-27 through Figure 19-32 are found in Table 19-17 on page 19-34. Detailed information about the operation of individual signals can be found in Chapter 13, "Serial Peripheral Interface 1 and 2."

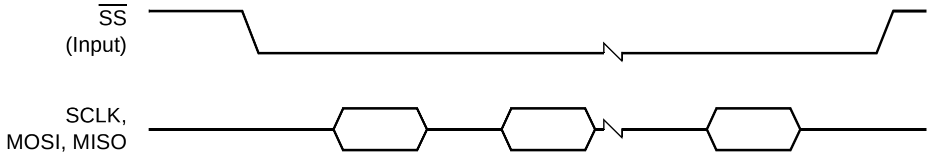


Figure 19-31. SPI 1 Slave FIFO Advanced by Bit Count Timing Diagram

19.3.31 SPI 1 Slave FIFO Advanced by $\overline{\text{SS}}$ Rising Edge

Figure 19-32 shows the timing diagram for the SPI 1 slave FIFO advanced by $\overline{\text{SS}}$ rising edge. The signal values and units of measure for Figure 19-27 through Figure 19-32 are found in Table 19-17. Detailed information about the operation of individual signals can be found in Chapter 13, "Serial Peripheral Interface 1 and 2."

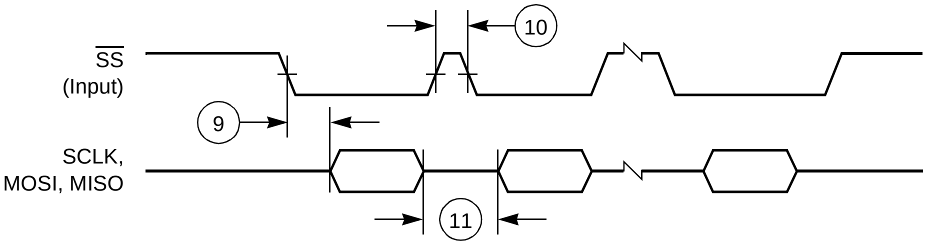


Figure 19-32. SPI 1 Slave FIFO Advanced by $\overline{\text{SS}}$ Rising Edge Timing Diagram

Table 19-17. Timing Parameters for Figure 19-27 Through Figure 19-32

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	Clock edge to TxD data ready	—	0.25T	ns
2	RxD data ready to clock edge	0.25T	—	ns
3	Clock edge to RxD data hold time	0.25T	—	ns
4	$\overline{\text{DATA\_READY}}$ to $\overline{\text{SS}}$ output low	—	2T	ns
5	$\overline{\text{SS}}$ output low to first SCLK edge	2T	—	ns
6	Last SCLK edge to $\overline{\text{SS}}$ output high	T	—	ns
7	$\overline{\text{SS}}$ output high to $\overline{\text{DATA\_READY}}$ low	T	—	ns
8	$\overline{\text{SS}}$ output pulse width	2T + WAIT	—	ns
9	$\overline{\text{SS}}$ input low to first SCLK edge	T	—	ns
10	$\overline{\text{SS}}$ input pulse width	0	—	ns
11	Pause between data word	0	—	ns

Note:
T = SPI clock period
WAIT = Number of sysclk or 32.768 KHz clocks per sample period control register

19.3.32 Normal Mode Timing

Figure 19-33 shows the timing diagram for normal mode timing of the MC68VZ328. The signal values and units of measure for Figure 19-33 through Figure 19-35 are found in Table 19-18 on page 19-36.

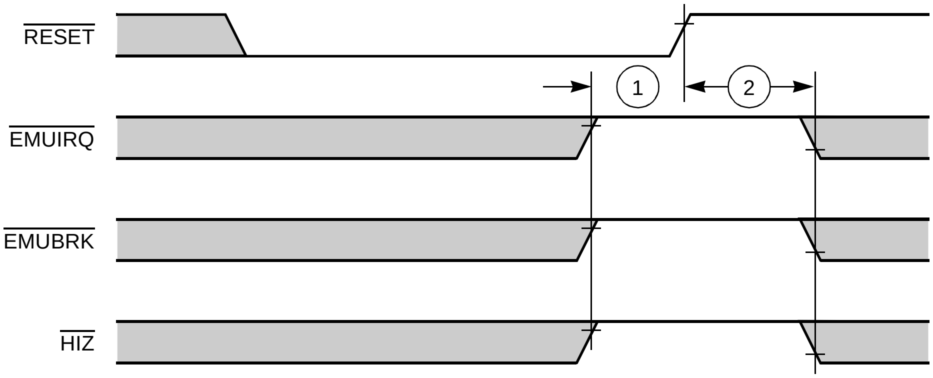


Figure 19-33. Normal Mode Timing Diagram

19.3.33 Emulation Mode Timing

Figure 19-34 shows the timing diagram for emulation mode timing of the MC68VZ328. The signal values and units of measure for Figure 19-33 through Figure 19-35 are found in Table 19-18 on page 19-36.

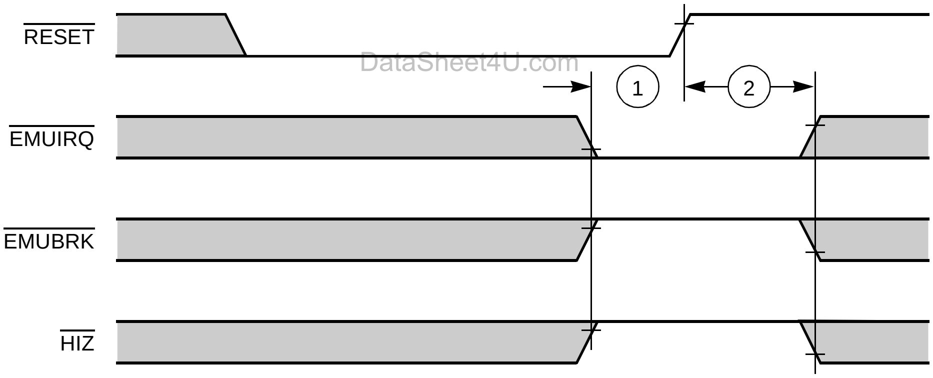


Figure 19-34. Emulation Mode Timing Diagram

19.3.34 Bootstrap Mode Timing

Figure 19-35 shows the timing diagram for bootstrap mode timing of the MC68VZ328. The signal values and units of measure for Figure 19-33 through Figure 19-35 are found in Table 19-18.

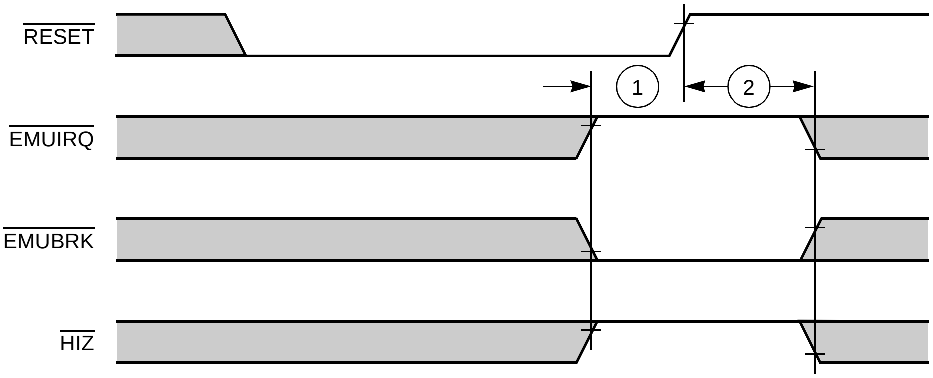

$\overline{\text{RESET}}$
1
2
$\overline{\text{EMUIRQ}}$
$\overline{\text{EMUBRK}}$
$\overline{\text{HIZ}}$

Figure 19-35. Bootstrap Mode Timing Diagram

Table 19-18. Timing Parameters for Figure 19-33 Through Figure 19-35

Number	Characteristic	(3.0 ± 0.3) V		Unit
		Minimum	Maximum	
1	$\overline{\text{EMUIRQ}}$, $\overline{\text{EMUBRK}}$, and $\overline{\text{HIZ}}$ setup time	10	—	ns
2	$\overline{\text{EMUIRQ}}$, $\overline{\text{EMUBRK}}$, and $\overline{\text{HIZ}}$ hold time	20	—	ns

Chapter 20
Mechanical Data and Ordering Information

This chapter provides mechanical data, including illustrations, and ordering information.

20.1 Ordering Information

Table 20-1 provides ordering information for the two package types: the 144-lead, plastic, thin quad flat package (TQFP) and the 144-lead mold array process ball grid array (MAPBGA) package.

Table 20-1. MC68VZ328 Ordering Information

Package Type	Frequency (MHz)	Temperature	Order Number
144-lead TQFP	33	0 °C to 70 °C	MC68VZ328PV33V
144-lead MAPBGA	33	0 °C to 70 °C	MC68VZ328VF33V
144-lead TQFP	33	-40 °C to 85 °C	MC68VZ328CPV33V
144-lead MAPBGA	33	-40 °C to 85 °C	MC68VZ328CVF33V

20.2 TQFP Pin Assignments

Figure 20-1 provides a top view of TQFP pin assignments.

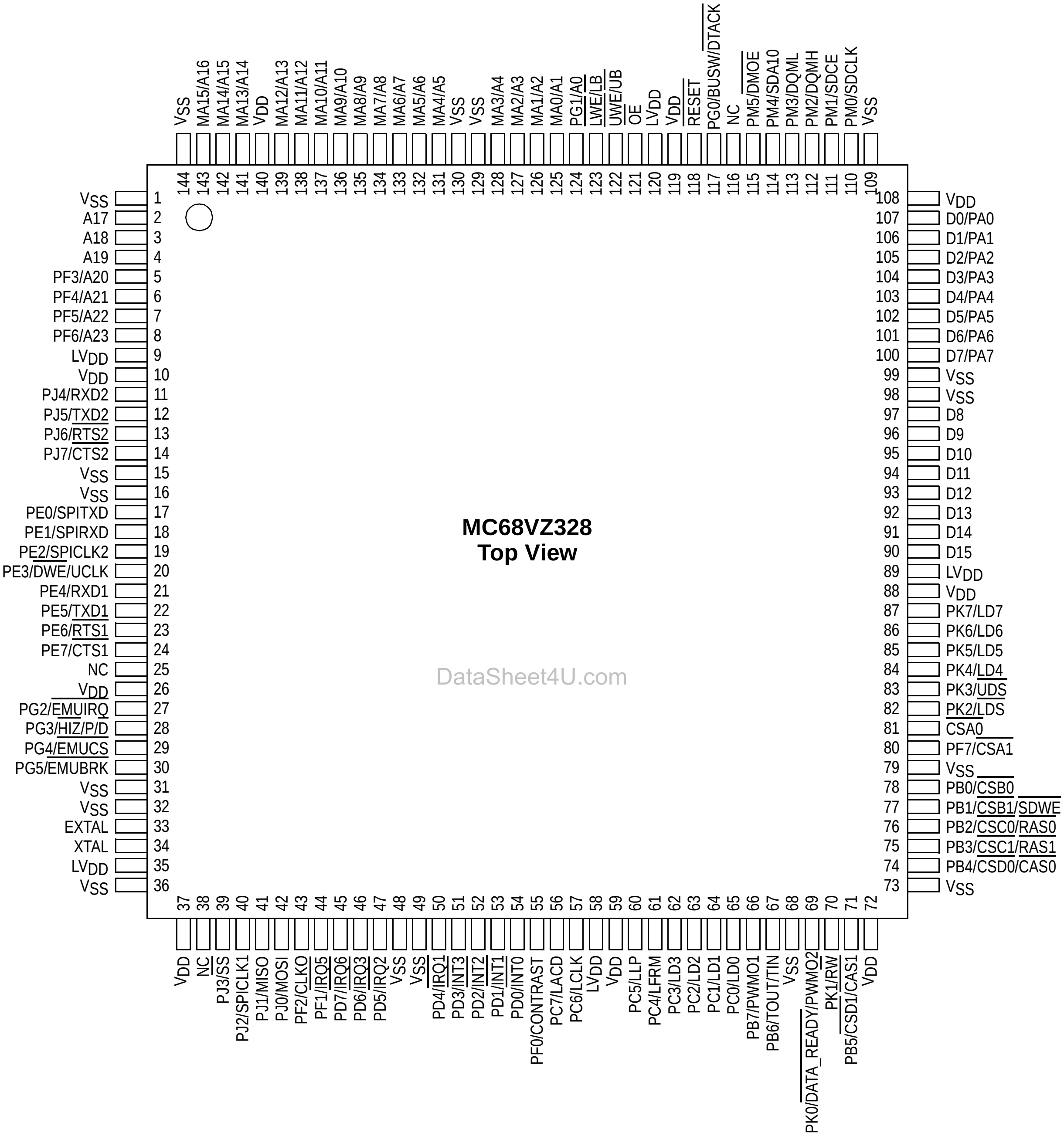
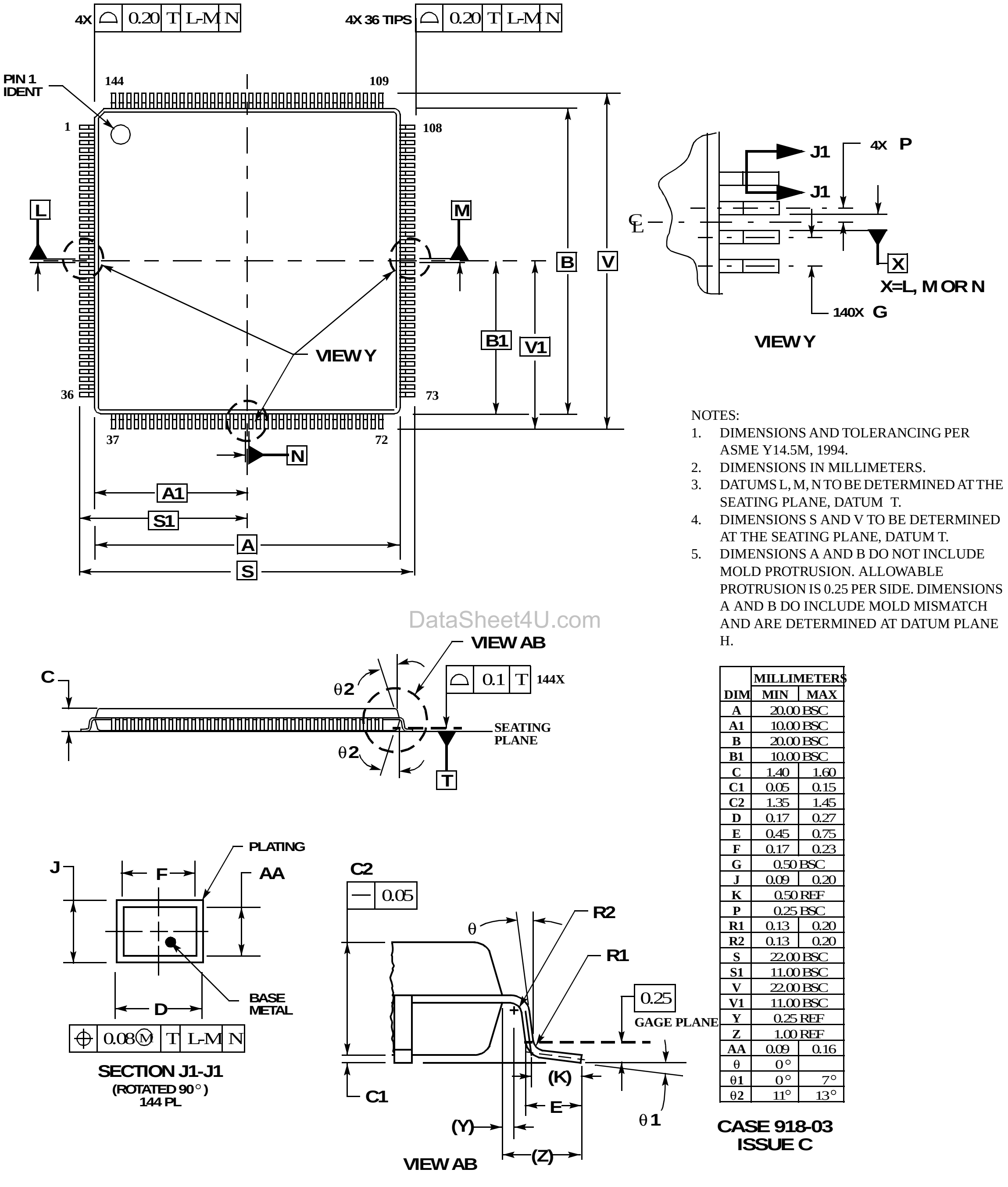


Figure 20-1. MC68VZ328 TQFP Pin Assignments—Top View

20.3 TQFP Package Dimensions

Figure 20-2 illustrates the TQFP 20 mm × 20 mm package, which has 0.5 mm spacing between the pads. The device designator for the TQFP package is PV.



NOTES:

1. DIMENSIONS AND TOLERANCING PER ASME Y14.5M, 1994.
2. DIMENSIONS IN MILLIMETERS.
3. DATUMS L, M, N TO BE DETERMINED AT THE SEATING PLANE, DATUM T.
4. DIMENSIONS S AND V TO BE DETERMINED AT THE SEATING PLANE, DATUM T.
5. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE H.

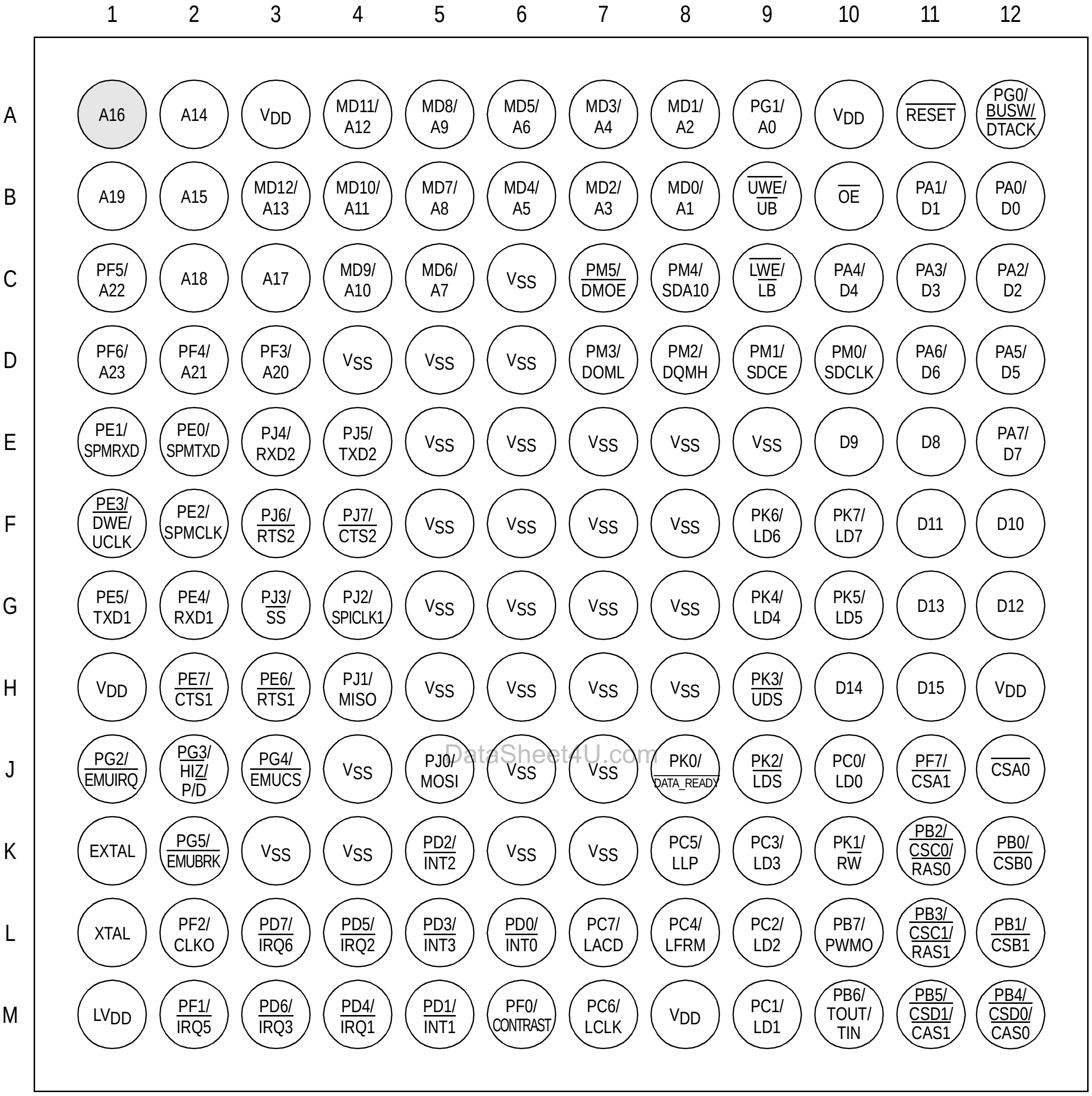
	MILLIMETERS	
DIM	MIN	MAX
A	20.00 BSC	
A1	10.00 BSC	
B	20.00 BSC	
B1	10.00 BSC	
C	1.40	1.60
C1	0.05	0.15
C2	1.35	1.45
D	0.17	0.27
E	0.45	0.75
F	0.17	0.23
G	0.50 BSC	
J	0.09	0.20
K	0.50 REF	
P	0.25 BSC	
R1	0.13	0.20
R2	0.13	0.20
S	22.00 BSC	
S1	11.00 BSC	
V	22.00 BSC	
V1	11.00 BSC	
Y	0.25 REF	
Z	1.00 REF	
AA	0.09	0.16
θ	0°	
θ1	0°	7°
θ2	11°	13°

CASE 918-03
ISSUE C

Figure 20-2. MC68VZ328 TQFP Mechanical Drawing

20.4 MAPBGA Pin Assignments

Figure 20-3 provides a top view of the MAPBGA pin assignments.



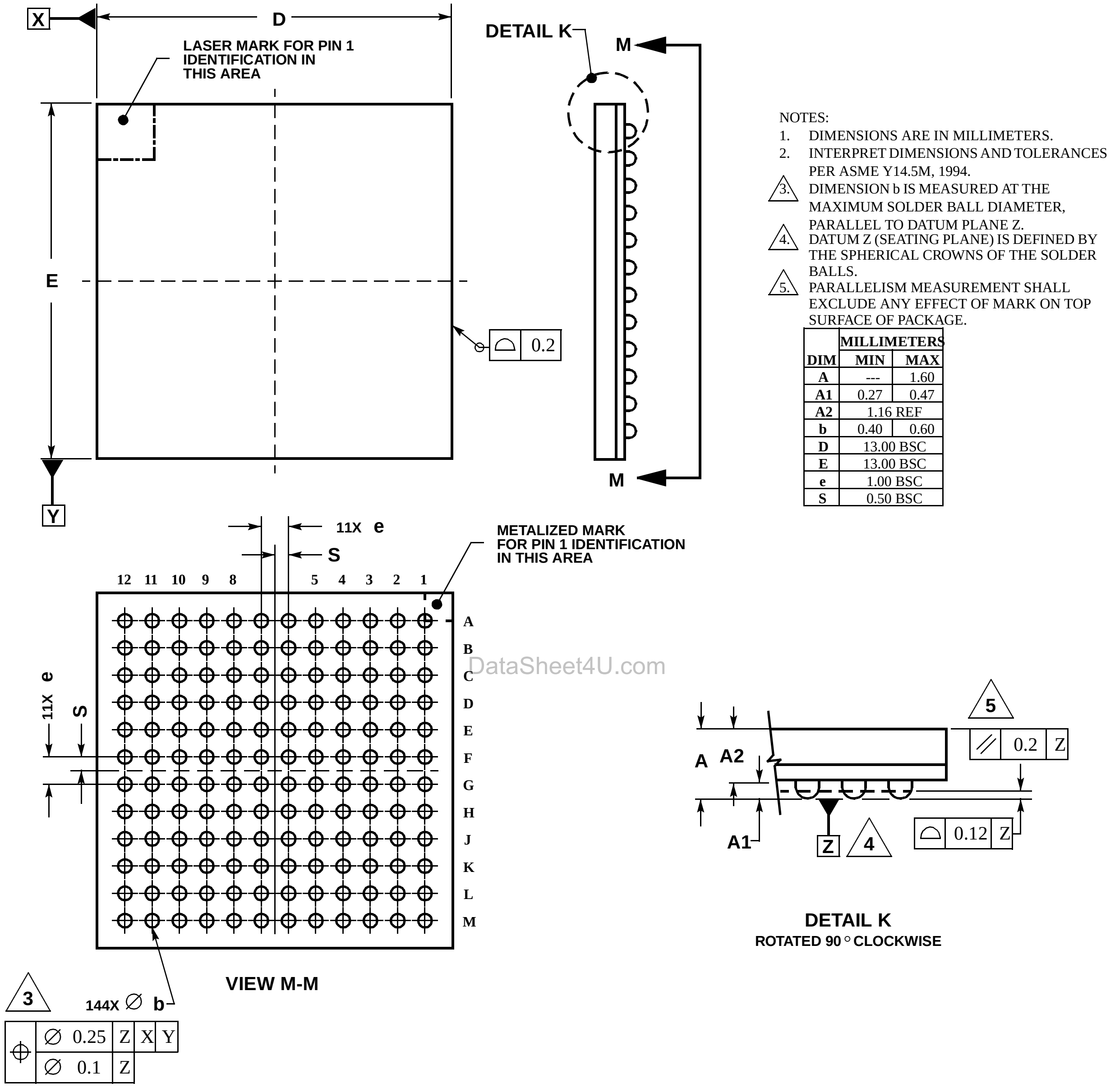
	1	2	3	4	5	6	7	8	9	10	11	12
A	A16	A14	VDD	MD11/A12	MD8/A9	MD5/A6	MD3/A4	MD1/A2	PG1/A0	VDD	RESET	PG0/BUSW/DTACK
B	A19	A15	MD12/A13	MD10/A11	MD7/A8	MD4/A5	MD2/A3	MD0/A1	UWE/UB	OE	PA1/D1	PA0/D0
C	PF5/A22	A18	A17	MD9/A10	MD6/A7	VSS	PM5/DMOE	PM4/SDA10	LWE/LB	PA4/D4	PA3/D3	PA2/D2
D	PF6/A23	PF4/A21	PF3/A20	VSS	VSS	VSS	PM3/DOML	PM2/DQMH	PM1/SDCE	PM0/SDCLK	PA6/D6	PA5/D5
E	PE1/SPMRXD	PE0/SPMTXD	PJ4/RXD2	PJ5/TXD2	VSS	VSS	VSS	VSS	VSS	D9	D8	PA7/D7
F	PE3/DWE/UCLK	PE2/SPMCLK	PJ6/RTS2	PJ7/CTS2	VSS	VSS	VSS	VSS	PK6/LD6	PK7/LD7	D11	D10
G	PE5/TXD1	PE4/RXD1	PJ3/SS	PJ2/SPICLK1	VSS	VSS	VSS	VSS	PK4/LD4	PK5/LD5	D13	D12
H	VDD	PE7/CTS1	PE6/RTS1	PJ1/MISO	VSS	VSS	VSS	VSS	PK3/UDS	D14	D15	VDD
J	PG2/EMUIRQ	PG3/HIZ/P/D	PG4/EMUCS	VSS	PJ0/MOSI	VSS	VSS	PK0/DATA_READY	PK2/LDS	PC0/LD0	PF7/CSA1	CSA0
K	EXTAL	PG5/EMUBRK	VSS	VSS	PD2/INT2	VSS	VSS	PC5/LLP	PC3/LD3	PK1/RW	PB2/CSC0/RAS0	PB0/CSB0
L	XTAL	PF2/CLKO	PD7/IRQ6	PD5/IRQ2	PD3/INT3	PD0/INT0	PC7/LACD	PC4/LFRM	PC2/LD2	PB7/PWMO	PB3/CSC1/RAS1	PB1/CSB1
M	LVDD	PF1/IRQ5	PD6/IRQ3	PD4/IRQ1	PD1/INT1	PF0/CONTRAST	PC6/LCLK	VDD	PC1/LD1	PB6/TOUT/TIN	PB5/CSD1/CAS1	PB4/CSD0/CAS0

Top View

Figure 20-3. MC68VZ328 MAPBGA Pin Assignments—Top View

20.5 MAPBGA Package Dimensions

Figure 20-4 illustrates the MAPBGA 13 mm × 13 mm package, which has 1 mm spacing between the pads. The device designator for the MAPBGA package is VF.



NOTES:
1. DIMENSIONS ARE IN MILLIMETERS.
2. INTERPRET DIMENSIONS AND TOLERANCES PER ASME Y14.5M, 1994.
3. DIMENSION b IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO DATUM PLANE Z.
4. DATUM Z (SEATING PLANE) IS DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
5. PARALLELISM MEASUREMENT SHALL EXCLUDE ANY EFFECT OF MARK ON TOP SURFACE OF PACKAGE.

	MILLIMETERS	
DIM	MIN	MAX
A	---	1.60
A1	0.27	0.47
A2	1.16 REF	
b	0.40	0.60
D	13.00 BSC	
E	13.00 BSC	
e	1.00 BSC	
S	0.50 BSC	

CASE 1242A-03
ISSUE B

Figure 20-4. MC68VZ328 MAPBGA Mechanical Drawing

20.6 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