MOTOROLA

MC68328 (DragonBall)™ Integrated Processor User’s Manual

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 can and do vary in different applications. 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 or sustain 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 μ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

68K FAX-IT

Documentation Comments

FAX 512-891-8593—Documentation Comments Only

The Motorola High-Performance Embedded Systems Technical Communications Department provides a fax number for you to submit any questions or comments about this document or how to order other documents. We welcome your suggestions for improving our documentation. Please do not fax technical questions.

Please provide the part number and revision number (located in upper right-hand corner of the cover) and the title of the document. When referring to items in the manual, please reference by the page number, paragraph number, figure number, table number, and line number if needed.

When sending a fax, please provide your name, company, fax number, and phone number including area code.

Applications and Technical Information

For questions or comments pertaining to technical information, questions, and applications, please contact one of the following sales offices nearest you.

— Sales Offices —

Field Applications Engineering Available Through All Sales Offices

UNITED STATES

ALABAMA, Huntsville	(205) 464-6800
ARIZONA, Tempe	(602) 897-5056
CALIFORNIA, Agoura Hills	(818) 706-1929
CALIFORNIA, Los Angeles	(310) 417-8848
CALIFORNIA, Irvine	(714) 753-7360
CALIFORNIA, Rosevllle	(916) 922-7152
CALIFORNIA, San Diego	(619) 541-2163
CALIFORNIA, Sunnyvale	(408) 749-0510
COLORADO, Colorado Springs	(719) 599-7497
COLORADO, Denver	(303) 337-3434
CONNECTICUT, Wallingford	(203) 949-4100
FLORIDA, Maitland	(407) 628-2636
FLORIDA, Pompano Beach/ Fort Lauderdale	(305) 486-9776
FLORIDA, Clearwater	(813) 538-7750
GEORGIA, Atlanta	(404) 729-7100
IDAHO, Boise	(208) 323-9413
ILLINOIS, Chicago/Hoffman Estates	(708) 490-9500
INDIANA, Fort Wayne	(219) 436-5818
INDIANA, Indianapolis	(317) 571-0400
INDIANA, Kokomo	(317) 457-6634
IOWA, Cedar Rapids	(319) 373-1328
KANSAS, Kansas City/Mission	(913) 451-8555
MARYLAND, Columbia	(410) 381-1570
MASSACHUSETTS, Marborough	(508) 481-8100
MASSACHUSETTS, Woburn	(617) 932-9700
MICHIGAN, Detroit	(313) 347-6800
MINNESOTA, Minnetonka	(612) 932-1500
MISSOURI, St. Louis	(314) 275-7380
NEW JERSEY, Fairfield	(201) 808-2400
NEW YORK, Fairport	(716) 425-4000
NEW YORK, Hauppauge	(516) 361-7000
NEW YORK, Poughkeepsie/Fishkill	(914) 473-8102
NORTH CAROLINA, Raleigh	(919) 870-4355
OHIO, Cleveland	(216) 349-3100
OHIO, Columbus/Worthington	(614) 431-8492
OHIO, Dayton	(513) 495-6800
OKLAHOMA, Tulsa	(800) 544-9496
OREGON, Portland	(503) 641-3681
PENNSYLVANIA, Colmar	(215) 997-1020
Philadelphia/Horsham	(215) 957-4100
TENNESSEE, Knoxville	(615) 690-5593
TEXAS, Austin	(512) 873-2000
TEXAS, Houston	(800) 343-2692
TEXAS, Plano	(214) 516-5100
VIRGINIA, Richmond	(804) 285-2100
WASHINGTON, Bellevue	(206) 454-4160
Seattle Access	(206) 622-9960
WISCONSIN, Milwaukee/Brookfield	(414) 792-0122

CANADA

BRITISH COLUMBIA, Vancouver	(604) 293-7605
ONTARIO, Toronto	(416) 497-8181
ONTARIO, Ottawa	(613) 226-3491
QUEBEC, Montreal	(514) 731-6881

INTERNATIONAL

AUSTRALIA, Melbourne	(61-3)887-0711
AUSTRALIA, Sydney	(61(2)906-3855
BRAZIL, Sao Paulo	55(11)815-4200
CHINA, Beijing	86 505-2180
FINLAND, Helsinki	358-0-35161191
Car Phone	358(49)211501
FRANCE, Paris/Vanves	33(1)40 955 900
GERMANY, Langenhagen/ Hanover	49(511)789911
GERMANY, Munich	49 89 92103-0
GERMANY, Nuremberg	49 911 64-3044
GERMANY, Sindelfingen	49 7031 69 910
GERMANY, Wiesbaden	49 611 761921
HONG KONG, Kwai Fong	852-4808333
Tai Po	852-6668333
INDIA, Bangalore	(91-812)627094
ISRAEL, Tel Aviv	972(3)753-8222
ITALY, Milan	39(2)82201
JAPAN, Aizu	81(241)272231
JAPAN, Atsugi	81(0462)23-0761
JAPAN, Kumagaya	81(0485)26-2600
JAPAN, Kyushu	81(092)771-4212
JAPAN, Mito	81(0292)26-2340
JAPAN, Nagoya	81(052)232-1621
JAPAN, Osaka	81(06)305-1801
JAPAN, Sendai	81(22)268-4333
JAPAN, Tachikawa	81(0425)23-6700
JAPAN, Tokyo	81(03)3440-3311
JAPAN, Yokohama	81(045)472-2751
KOREA, Pusan	82(51)4635-035
KOREA, Seoul	82(2)554-5188
MALAYSIA, Penang	60(4)374514
MEXICO, Mexico City	52(5)282-2864
MEXICO, Guadalajara	52(36)21-8977
Marketing	52(36)21-9023
Customer Service	52(36)669-9160
NETHERLANDS, Best	(31)49988 612 11
PUERTO RICO, San Juan	(809)793-2170
SINGAPORE	(65)2945438
SPAIN, Madrid	34(1)457-8204
or	34(1)457-8254
SWEDEN, Solna	46(8)734-8800
SWITZERLAND, Geneva	41(22)7991111
SWITZERLAND, Zurich	41(1)730 4074
TAIWAN, Taipei	886(2)717-7089
THAILAND, Bangkok	(66-2)254-4910
UNITED KINGDOM, Aylesbury	44(296)395-252

FULL LINE REPRESENTATIVES

COLORADO, Grand Junction	
Cheryl Lee Whltely	(303) 243-9658
KANSAS, Wichita	
Melinda Shores/Kelly Greiving	(316) 838 0190
NEVADA, Reno	
Galena Technology Group	(702) 746 0642
NEW MEXICO, Albuquerque	
S&S Technologies, Inc.	(505) 298-7177
UTAH, Salt Lake City	
Utah Component Sales, Inc.	(801) 561-5099
WASHINGTON, Spokane	
Doug Kenley	(509) 924-2322
ARGENTINA, Buenos Aires	
Argonics, S.A.	(541) 343-1787

HYBRID COMPONENTS RESELLERS

Elmo Semiconductor	(818) 768-7400
Minco Technology Labs Inc.	(512) 834-2022
Semi Dice Inc.	(310) 594-4631

PREFACE

The *MC68328 Integrated Portable System Processor User's Manual* describes the programming, capabilities, and operation of the M68328; the *MC68000 Family Programmer's Reference Manual* provides instruction details for the EC000 core; and the *Integrated Portable System Process: DragonBall Product Brief* provides an overview of the M68328.

The organization of this manual is as follows:

Section 1	Overview
Section 2	System Integration Module
Section 3	Phase-Locked Loop and Power Control
Section 4	LCD Controller Module
Section 5	Real-Time Clock Module
Section 6	Timer Module
Section 7	Parallel Ports
Section 8	Universal Asynchronous Receiver Transmitter (UART) Module
Section 9	Serial Peripheral Interfact—Slave (SPIS)
Section 10	Serial Peripheral Interface—Master (SPIM)
Section 11	Pulse-Width Modulator
Section 12	Pin Assignment
Section 13	Electrical Characteristics
Index	

TRADEMARKS

All trademarks reside with their respective owners.

TABLE OF CONTENTS

Section 1
MC68328 Processor Overview

Section 2
System Integration Module

Section 3
Phase-Locked Loop and Power Control

Section 4
LCD Controller Module

Section 5
Real-Time Clock Module

Section 6
Timer

Section 7
Parallel Ports

Section 8
Universal Asynchronous Receiver/ Transmitter (UART)

Section 9
Serial Peripheral Interface— SLAVE (SPIS)

Section 10
Serial Peripheral Interface— Master (SPIM)

Section 11
Pulse-Width Modulator

Section 12
Pin Assignment

Section 13
Electrical Characteristics

SECTION 1
MC68328 PROCESSOR OVERVIEW

As the consumer market for portable devices expands, system requirements have become more demanding. Minimum number of components, smaller board space, lower power consumption, and lower system cost are a few of the criteria necessary for a successful product. In response to these consumer needs, Motorola has designed a new processor. The MC68328 DragonBall™ integrated portable system processor provides a 3.3V, fully static operation in an efficient package. The MC68328 processor delivers cost-effective performance to satisfy the extensive requirements of today's consumer market for portable devices.

The MC68328 processor provides key features that are suitable for many portable applications. Modules like a real-time clock (RTC), an LCD controller, pulse-width modulator (PWM), timers, master and slave serial peripheral interface (SPI), universal asynchronous receiver/transmitter (UART), and the system integration module (SIM28) give the system engineer more flexibility and resources to design efficient and innovative products.

1.1 KEY FEATURES

The primary features of the MC68328 processor, illustrated in Figure 1, are as follows:

- MC68EC000 Static Core Processor
- 100% Compatibility with MC68000 And MC68EC000 Processors
  —24-Bit External and 32-Bit Internal Address Bus
  —Optional A31-A24 Capable of 4 Gbytes of Address Space
  —16-Bit On-Chip Data Bus for MC68EC000 Bus Operations
  —Static Design Allows Processor Clock to Be Stopped for Power Savings
  —2.7 MIPS Performance at 16.67-MHz Processor Clock
- Dynamic Bus Sizing Support for Connections to 8-Bit and 16-Bit Devices
- System Integration Module (SIM28) Supporting Glueless System Design:
  —System Configuration, Programmable Address Mapping
  —Memory Interface for SRAM, EPROM and FLASH Memory
  —16 Programmable Peripheral Chip-Selects with Wait-State Generation Logic
  —Interrupt Controller with 13 Flexible Inputs
  —Programmable Interrupt Vector Generator
  —Hardware Watchdog Timer
  —Software Watchdog Timer
  —Low-Power Mode Control
  —Up to 77 Individually Programmable Parallel Port Signals
  —PCMCIA 1.0 Support

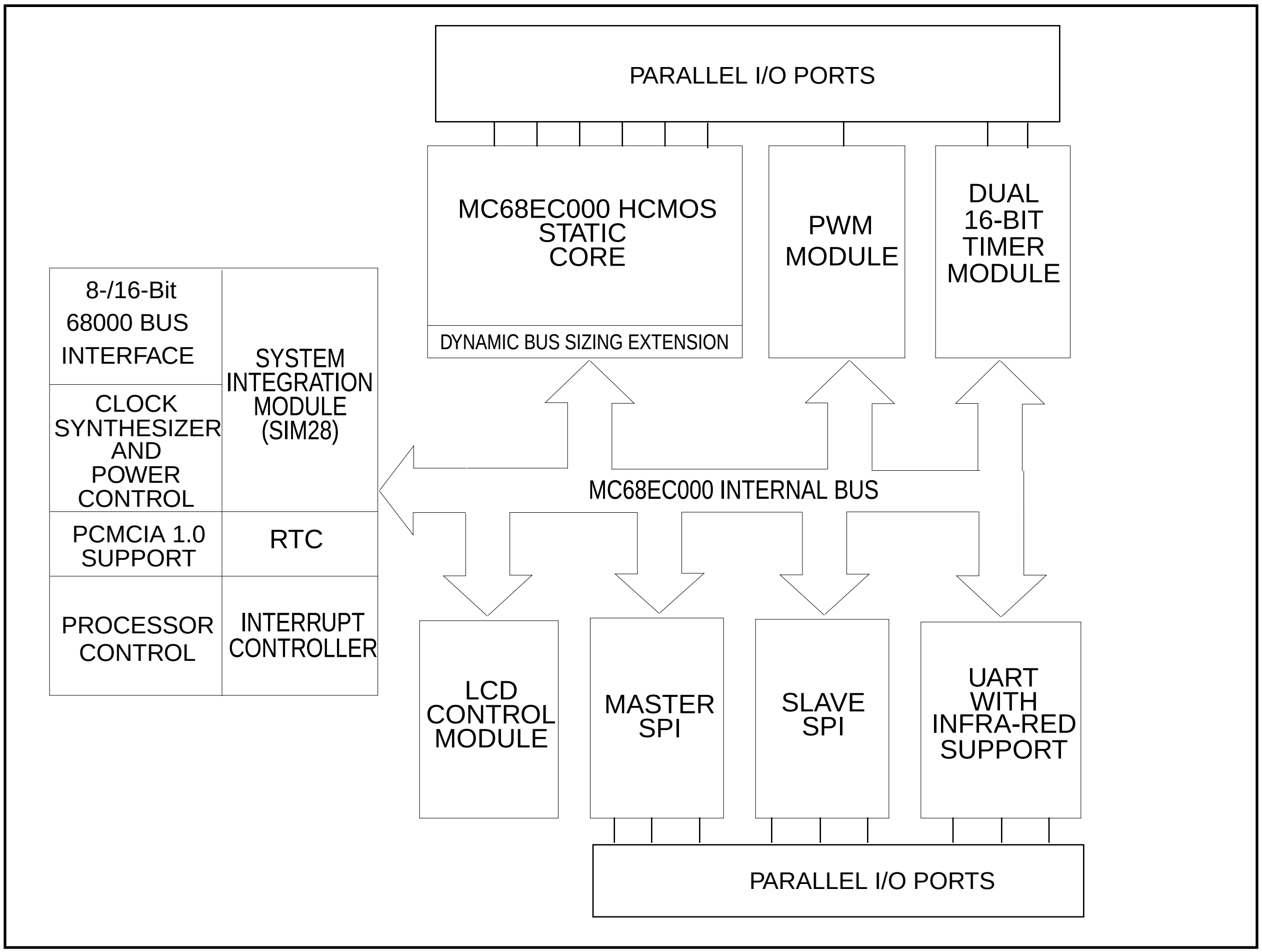


Figure 1. MC68328 Block Diagram

- UART
  —Supports IrDA-Compliant Physical-Layer Protocol
  —8-Byte FIFOs for Rx and Tx
- Two Separate Serial Peripheral Interface Ports (Master and Slave)
  —Support For External POCSAG Decoder (Slave)
  —Support for Digitizer For A/D Input or EEPROM (Master)
- Dual Channel 16-Bit General-Purpose Counter/Timer
  —Multimode Operation, Independent Capture/Compare Registers
  —Automatic Interrupt Generation
  —240ns Resolution at 16.67-MHz System Clock
  —Each Timer Has an Input and an Output Pin for Capture and Compare
- Pulse-Width Modulation Output for Sound Generation
  —Programmable Frame Rate
  —16-Bit Programmable
  —Supports Motor Control
- Real-Time Clock
  —24-Hour Time
  —1 Programmable Alarm

- Power Management
  —3.3 V Operation
  —Fully Static HCMOS Technology
  —Programmable Clock Synthesizer for Full Frequency Control
  —Low Power Stop Capabilities
  —Individual Module Shut Down Capability
  —Lowest Power-Mode Control (Shut Down CPU and Peripherals)
- LCD Control Module
  —Software Programmable Screen Size to Support Single (Non-Split) Monochrome/ STN Panels
  —Direct Drive Capability of Common LCD Drivers/Modules from Motorola and Other LCD Drive Manufacturers
  —Support as Many as 4 Grey Levels
  —Use System Memory as Display Memory
- IEEE 1149.1 Boundary Scan Test Access Port (JTAG)
- Operation From DC To 16.67 MHz (Processor Clock)
- Operating Voltage of 3.3V ± 0.3V
- Compact 144-Lead Thin-Quad-Flat-Pack (TQFP) Package

1.1.1 ORGANIZATION

The M68300 family of integrated processors and controllers is built on an M68000 core processor and a selection of intelligent peripherals appropriate for a set of applications. Common system glue logic such as address decoding, wait-state insertion, interrupt prioritization, and watchdog timing is also included.

Each member of the M68300 family is distinguished from the others by its selection of on-chip peripherals. Peripherals are chosen to address specific applications but are often useful in a variety of applications. The peripherals may be highly sophisticated timing or protocol engines that have their own processors, or they may be more traditional peripheral functions such as UARTs and timers.

1.1.2 ADVANTAGES

The many features incorporated into a single M68300 Family chip help system designers realize significant savings in design time, power consumption, cost, board space, pin count, and programming. The equivalent functionality can easily require 20 separate components. Each component might have 16–64 pins, totalling over 350 connections. Most of these connections require interconnects or are duplications. Each connection: (1) is a candidate for a bad solder joint or misrouted trace, (2) is another part to qualify, purchase, inventory, and maintain. Each component (1) requires a share of the printed circuit board, (2) draws power, which often drives large buffers to get the signal to another chip. The cumulative power consumption of all the components must be available from the power supply. The signals between the central processor unit (CPU) and a peripheral might not be compatible nor run from the same clock, which could require time delays or other special design considerations.

In an M68300 family component, the major functions and glue logic are all properly connected internally, timed with the same clock, fully tested, and uniformly documented. Only essential signals are brought out to pins. The primary package is the surface-mount plastic QFP for the smallest possible footprint.

1.2 MC68328 ARCHITECTURE

To improve total system throughput and reduce component count, board size, and cost of system implementation, the MC68328 processor integrates a powerful MC68EC000 processor, intelligent peripheral modules, and typical system interface logic. These functions include the system integration module (SIM28), timers, LCD controller, and more.

1.2.1 EC000 STATIC CORE

The EC000 core is a core implementation of the M68000 32-bit microprocessor architecture. The features of the EC000 core processor include:

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

The EC000 core is completely upward user code-compatible with all other members of the M68000 microprocessor families and thus has access to a broad base of established real-time kernels, operating systems, languages, applications, and development tools.

1.2.1.1 EC000 CORE PROGRAMMING MODEL. The EC000 core offers sixteen 32-bit registers and a 32-bit program counter (see Figure 1-1). The first 8 registers (D7–D0) serve as data registers for byte (8-bit), word (16-bit) and long-word (32-bit) operations. Because using data registers will affect the condition-code register (which indicates negative number, carry, and overflow conditions), they (the data registers) are used primarily for data manipulation. The second set of 7 registers (A6–A0) and the user stack pointer (USP) may function as software stack pointers and base-address registers. These registers can be used for word and long-word operations and do not affect the condition-code register. All of the registers (D7–D0 and A6–A0) may serve as index registers.

In supervisor mode, the upper byte of the status register (SR) and the supervisor stack pointer (SSP) are also available to programmers. These registers are shown in Figure 1-3.

The SR (refer to Figure 1-3) contains the interrupt mask (7 levels available) as well as these condition codes: extend (X), negative (N), zero (Z), overflow (V), and carry (C). Additional status bits indicate whether the processor is in trace mode (T-bit) or in supervisor/ user state (S-bit).

1.2.1.2 DATA TYPES AND ADDRESS MODES. Five basic data types are supported:

1. Bits
2. Binary-coded decimal (BCD) digits (4 bits)

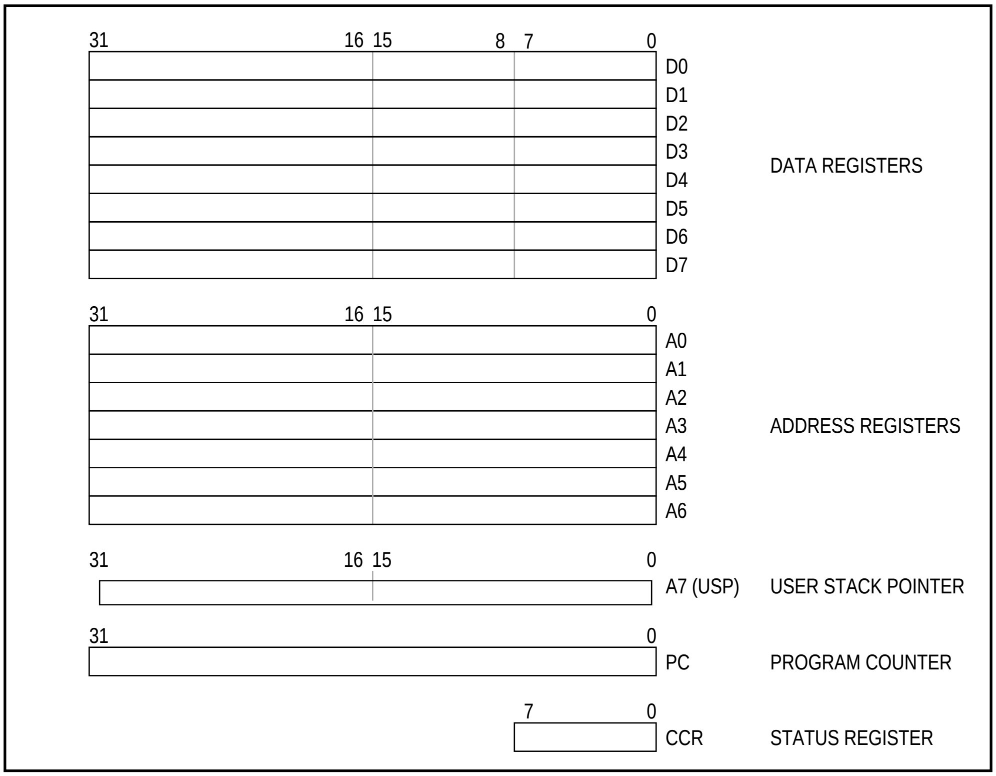


Figure 1-1. User Programming Model

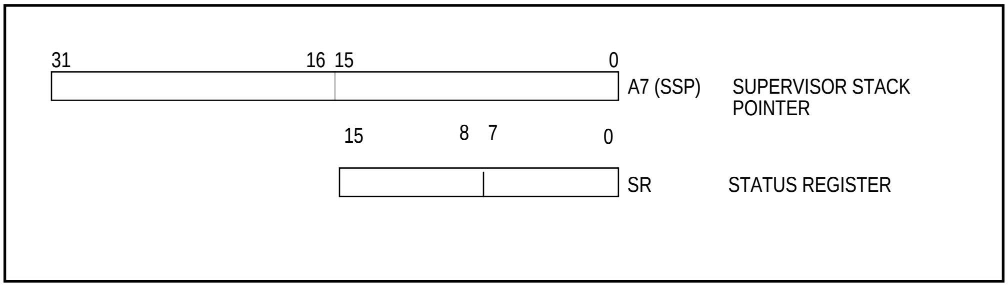


Figure 1-2. Supervisor Programmer's Model Supplement

3. Bytes (8 bits)
4. Words (16 bits)
5. Long words (32 bits)

In addition, operations on other data types such as memory addresses, status word data, etc. are provided in the instruction set.

The 14 address modes listed in Table 1-1 include six basic types:

1. Register direct
2. Register indirect
3. Absolute
4. Program counter relative

5. Immediate
6. Implied

The register-indirect address modes can perform post-increment, pre-decrement, offset, and index operations. The program-counter relative mode can be modified through index and offset operations.

Table 1-1. Address Modes

Address modes	Syntax
Register direct address	
Data register direct	Dn
Address register direct	An
Absolute data address	
Absolute short	xxx.W
Absolute long	xxx.L
Program counter relative address	
Relative with offset	d_{16}(PC)
Relative with index offset	d_8(PC, Xn)
Register indirect address register	
Register indirect	(An)
Postincrement register indirect	(An)+
Predecrement register indirect	–(An)
Register indirect with offset	d_{16}(An)
Indexed register indirect with offset	d_8(An, Xn)
Immediate data address	
Immediate	#xxx
Quick immediate	#1–#8
Implied address	
Implied register	SR/USP/SP/PC

Legend:
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	

1.2.1.3 INSTRUCTION SET OVERVIEW. Table 1-2 lists the EC000 core instruction set. The instruction set supports high-level languages that facilitate programming. Each instruction, with few exceptions, operates on bytes, words, and long-words, and most instructions can use any of the 14 address modes. A combination of instruction types, data types, and address modes provides over 1000 useful instructions. These instructions include signed and unsigned, multiply and divide, quick arithmetic operations, BCD arithmetic, and expanded operations (through traps).

1.2.2 SYSTEM INTEGRATION MODULE

The MC68328 SIM28 consists of several functions that control the system startup, initialization, configuration, and the external bus with a minimum of external devices. The memory

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	Ones 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 cond, decrement and branch	RTR	Return and restore
DIVS	Signed divide	RTS	Return from subroutine
DIVU	Unsigned divide	SBCD	Subtract decimal with extend
EOR	Exclusive OR	Scc	Set conditional
EORI	Exclusive OR immediate	STOP	Stop
EORI to CCR	Exclusive OR immediate to condition codes	SUB	Subtract
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	—	—

interface lets users connect gluelessly with the popular SRAM, EPROM as well as PCMCIA 1.0 memory cards. With the assistance of chip-select logic, wait states can be programmable. The hardware and software watchdog timers help users perform system protections. The interrupt controller accepts and resolves the priority from internal modules and externally generated interrupts and also handles the mask and wake-up selection control for power control. The low-power logic can control the CPU power dissipation by a frequency change or stopping it altogether. The SIM28 can also configure the pin to let users select either dedicated I/O or parallel I/O. This feature helps increase the number of available I/O ports by reclamation when the dedicated function is not in use.

1.2.2.1 SYSTEM CONFIGURATION. The MC68328 processor system configuration logic consists of a system control register (SCR) that lets users configure these major function operations:

- System status and control logic
- Register double mapping
- Bus error generation control
- Module control registers protection from access by user programs

1.2.2.2 VCO/PLL CLOCK SYNTHESIZER. The clock synthesizer can operate with either an external crystal or an external oscillator for reference, using the internal phase-locked loop (PLL). The other option is for an external clock to directly drive the clock signal at the operational frequency.

1.2.2.3 CHIP-SELECT LOGIC. The MC68328 processor provides 16 programmable, general-purpose, chip-select signals. For a given chip-select block, users may choose: (1) whether the chip-select allows read-only or both read and write accesses, (2) whether a $\overline{\text{DTACK}}$ is automatically generated for this chip-select, and (3) the number of wait states (from zero to six) until the $\overline{\text{DTACK}}$ will be generated.

1.2.2.4 EXTERNAL BUS INTERFACE. The external bus interface handles the transfer of information between the internal MC68EC000 core and the memory, peripherals, or other processing elements in the external address space. It consists of a 16-bit 68000 data bus interface for internal-only devices and a programmable 8-bit or 16-bit data bus interface to external devices.

1.2.2.5 INTERRUPT CONTROLLER. The interrupt controller accepts and prioritizes both 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 has these major features:

- Prioritized interrupt sources (internal and external)
- Fully nested interrupt environment
- Programmable vector generation
- Unique vector number generated for each interrupt level
- Interrupt masking
- Wakeup interrupt masking

1.2.2.6 PARALLEL GENERAL-PURPOSE I/O PORTS. The MC68328 processor supports up to 77 bit general-purpose I/O ports that can be configured as general-purpose I/O pins or as dedicated peripheral-interface pins of the on-chip modules.

Each port pin can be independently programmed as a general-purpose I/O pin even when other pins related to that on-chip peripheral are used as dedicated pins. If all the pins for a particular peripheral are configured as general-purpose I/O, the peripheral will still operate normally, although this is useful only with the RTC and timer modules.

1.2.2.7 SOFTWARE WATCHDOG. A software watchdog timer protects against system failures by providing a means of escape from unexpected input conditions, external events, or programming errors. Once started, the software watchdog timer must be cleared by software on a regular basis so that it never reaches its time-out value. When it reaches the time-out value, the watchdog timer assumes that a system failure has occurred, and the software watchdog logic resets or interrupts the MC68EC000 core.

1.2.2.8 LOW POWER STOP LOGIC. Various power-save options are available: turn off unused peripherals, reduce processor clock speed, disable the processor altogether, or a combination of these.

A wake-up-from-low-power mode can be achieved by an interrupt at the interrupt controller logic that runs throughout the period of processor low-power. Programmable interrupt sources can serve as events to wake up the EC000 core.

The on-chip peripherals can initiate a wake-up; for example, the timer can be set to wake-up after a certain elapsed time or number of external events.

1.2.3 LCD Controller

- Interfaces with monochrome STN LCD modules
- Up to 4 levels of gray scale through frame rate control
- Use system RAM for display memory
- Screen refresh through DMA

1.2.4 UART and Infrared Communication Support

The UART supports standard asynchronous serial communications at normal baud rates and is compatible with IrDA Physical Communication Protocol

1.2.5 Real-Time Clock

A 32.76kHz or 38.4kHz crystal (the same as the clock synthesizer clock source) drives the real-time clock in the MC68328 processor and provides an alarm interrupt.

1.2.6 JTAG Test Access Port

To assist in system diagnostics, the MC68328 processor includes dedicated user-accessible test logic that is fully compliant with the IEEE 1149.1 standard for boundary-scan testability, often referred to as JTAG (Joint Test Action Group).

1.2.7 SIM28 Programming Model

The SIM28 programming model is listed in Table 1-3.

Table 1-3. Programmer's Memory Map

Address	Name	Width	Block	Description	Reset Value(hex)
Base+$000	SCR	8	SIM	System Control Register	$0C
Base+$100	GRPBASEA	16	CS	Chip Select Group A Base Register	$0000
Base+$102	GRPBASEB	16	CS	Chip Select Group B Base Register	$0000
Base+$104	GRPBASEC	16	CS	Chip Select Group C Base Register	$0000
Base+$106	GRPBASED	16	CS	Chip Select Group D Base Register	$0000
Base+$108	GRPMASKA	16	CS	Chip Select Group A Mask Register	$0000
Base+$10A	GRPMASKB	16	CS	Chip Select Group B Mask Register	$0000
Base+$10C	GRPMASKC	16	CS	Chip Select Group C Mask Register	$0000
Base+$10E	GRPMASKD	16	CS	Chip Select Group D Mask Register	$0000
Base+$110	CSA0	32	CS	Group A Chip Select 0 Register	$00010006
Base+$114	CSA1	32	CS	Group A Chip Select 1 Register	$00010006
Base+$118	CSA2	32	CS	Group A Chip Select 2 Register	$00010006
Base+$11C	CSA3	32	CS	Group A Chip Select 3 Register	$00010006
Base+$120	CSB0	32	CS	Group B Chip Select 0 Register	$00010006
Base+$124	CSB1	32	CS	Group B Chip Select 1 Register	$00010006
Base+$128	CSB2	32	CS	Group B Chip Select 2 Register	$00010006
Base+$12C	CSB3	32	CS	Group B Chip Select 3 Register	$00010006
Base+$130	CSC0	32	CS	Group C Chip Select 0 Register	$00010006
Base+$134	CSC1	32	CS	Group C Chip Select 1 Register	$00010006
Base+$138	CSC2	32	CS	Group C Chip Select 2 Register	$00010006
Base+$13C	CSC3	32	CS	Group C Chip Select 3 Register	$00010006
Base+$140	CSD0	32	CS	Group D Chip Select 0 Register	$00010006
Base+$144	CSD1	32	CS	Group D Chip Select 1 Register	$00010006
Base+$148	CSD2	32	CS	Group D Chip Select 2 Register	$00010006
Base+$14C	CSD3	32	CS	Group D Chip Select 3 Register	$00010006
Base+$200	PLLCR	16	PLL	PLL Control Register	$2400
Base+$202	PLLFSR	16	PLL	PLL Frequency Select Register	$0123
Base+$204	Reserved	-	PLL	Do Not Access	-
Base+$207	PCTLR	8	PCTL	Power Control Register	$1F
Base+$300	IVR	8	INTR	Interrupt Vector Register	$00
Base+$302	ICR	16	INTR	Interrupt Control Register	$0000
Base+$304	IMR	32	INTR	Interrupt Mask Register	$00FFFFFF
Base+$308	IWR	32	INTR	Interrupt Wakeup Enable Register	$00FFFFFF
Base+$30C	ISR	32	INTR	Interrupt Status Register	$00000000
Base+$310	IPR	32	INTR	Interrupt Pending Register	-
Base+$400	PADIR	8	PIO	Port A Direction Register	$00
Base+$401	PADATA	8	PIO	Port A Data Register	$00
Base+$403	PASEL	8	PIO	Port A Select Register	$00
Base+$408	PBDIR	8	PIO	Port B Direction Register	$00
Base+$409	PBDATA	8	PIO	Port B Data Register	$00
Base+$40B	PBSEL	8	PIO	Port B Select Register	$00
Base+$410	PCDIR	8	PIO	Port C Direction Register	$00
Base+$411	PCDATA	8	PIO	Port C Data Register	$00
Base+$413	PCSEL	8	PIO	Port C Select Register	$00
Base+$418	PDDIR	8	PIO	Port D Direction Register	$00
Base+$419	PDDATA	8	PIO	Port D Data Register	$00
Base+$41A	PDPUEN	8	PIO	Port D Pullup Enable Register	$FF
Base+$41C	PDPOL	8	PIO	Port D Polarity Register	$00
Base+$41D	PDIRQEN	8	PIO	Port D IRQ Enable Register	$00

Table 1-3. Programmer’s Memory Map (Continued)

Address	Name	Width	Block	Description	Reset Value(hex)
Base+$41F	PDIRQEDGE	8	PIO	Port D IRQ Edge Register	$00
Base+$420	PEDIR	8	PIO	Port E Direction Register	$00
Base+$421	PEDATA	8	PIO	Port E Data Register	$00
Base+$422	PEPUEN	8	PIO	Port E Pullup Enable Register	$80
Base+$423	PESEL	8	PIO	Port E Select Register	$80
Base+$428	PFDIR	8	PIO	Port F Direction Register	$00
Base+$429	PFDATA	8	PIO	Port F Data Register	$00
Base+$42A	PFPUEN	8	PIO	Port F Pullup Enable Register	$FF
Base+$42B	PFSEL	8	PIO	Port F Select Register	$FF
Base+$430	PGDIR	8	PIO	Port G Direction Register	$00
Base+$431	PGDATA	8	PIO	Port G Data Register	$00
Base+$432	PGPUEN	8	PIO	Port G Pullup Enable Register	$FF
Base+$433	PGSEL	8	PIO	Port G Select Register	$FF
Base+$438	PJDIR	8	PIO	Port J Direction Register	$00
Base+$439	PJDATA	8	PIO	Port J Data Register	$00
Base+$43B	PJSEL	8	PIO	Port J Select Register	$00
Base+$440	PKDIR	8	PIO	Port K Direction Register	$00
Base+$441	PKDATA	8	PIO	Port K Data Register	$00
Base+$442	PKPUEN	8	PIO	Port K Pullup Enable Register	$FF
Base+$443	PKSEL	8	PIO	Port K Select Register	$FF
Base+$448	PMDIR	8	PIO	Port M Direction Register	$00
Base+$449	PMDATA	8	PIO	Port M Data Register	$00
Base+$44A	PMPUEN	8	PIO	Port M Pullup Enable Register	$FF
Base+$44B	PMSEL	8	PIO	Port M Select Register	$FF
Base+$500	PWMC	16	PWM	PWM Control Register	$0000
Base+$502	PWMP	16	PWM	PWM Period Register	$0000
Base+$504	PWMW	16	PWM	PWM Width Register	$0000
Base+$506	PWMCNT	16	PWM	PWM Counter	$0000
Base+$600	TCTL1	16	Timer	Timer Unit 1 Control Register	$0000
Base+$602	TPRER1	16	Timer	Timer Unit 1 Prescalar Register	$0000
Base+$604	TCMP1	16	Timer	Timer Unit 1 Compare Register	$FFFF
Base+$606	TCR1	16	Timer	Timer Unit 1 Capture Register	$0000
Base+$608	TCN1	16	Timer	Timer Unit 1 Counter	$0000
Base+$60A	TSTAT1	16	Timer	Timer Unit 1 Status Register	$0000
Base+$60C	TCTL2	16	Timer	Timer Unit 2 Control Register	$0000
Base+$60E	TPREP2	16	Timer	Timer Unit 2 Prescaler Register	$0000
Base+$610	TCMP2	16	Timer	Timer Unit 2 Compare Register	$FFFF
Base+$612	TCR2	16	Timer	Timer Unit 2 Capture Register	$0000
Base+$614	TCN2	16	Timer	Timer Unit 2 Counter	$0000
Base+$616	TSTAT2	16	Timer	Timer Unit Status Register	$0000
Base+$618	WCR	16	WD	Watchdog Control Register	$0000
Base+$61A	WCR	16	WD	Watchdog Compare Register	$FFFF
Base+$61C	WCN	16	WD	Watchdog Counter	$0000
Base+$700	SPISR	16	SPIS	SPIS Register	$0000
Base+$800	SPIMDATA	16	SPIM	SPIM Data Register	$0000
Base+$802	SPIMCONT	16	SPIM	SPIM Control/Status Register	$0000
Base+$900	USTCNT	16	UART	UART Status/Control Register	$0000
Base+$902	UBAUD	16	UART	UART Baud Control Register	$003F
Base+$904	URX	16	UART	UART RX Register	$0000

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

Address	Name	Width	Block	Description	Reset Value(hex)
Base+$906	UTX	16	UART	UART TX Register	$0000
Base+$908	UMISC	16	UART	UART Misc Register	$0000
Base+$A00	LSSA	32	LCDC	Screen Starting Address Register	$00000000
Base+$A05	LVPW	8	LCDC	Virtual Page Width Register	$FF
Base+$A08	LXMAX	16	LCDC	Screen Width Register	$03FF
Base+$A0A	LYMAX	16	LCDC	Screen Height Register	$01FF
Base+$A18	LCXP	16	LCDC	Cursor X Position	$0000
Base+$A1A	LCYP	16	LCDC	Cursor Y Position	$0000
Base+$A1C	LCWCH	16	LCDC	Cursor Width & Height Register	$0101
Base+$A1F	LBLKC	8	LCDC	Blink Control Register	$7F
Base+$A20	LPICF	8	LCDC	Panel Interface Config Register	$00
Base+$A21	LPOLCF	8	LCDC	Polarity Config Register	$00
Base+$A23	LACDRC	8	LCDC	ACD (M) Rate Control Register	$00
Base+$A25	LPXCD	8	LCDC	Pixel Clock Divider Register	$00
Base+$A27	LCKCON	8	LCDC	Clocking Control Register	$40
Base+$A29	LLBAR	8	LCDC	Last Buffer Address Register	$3E
Base+$A2B	LOTCR	8	LCDC	Octet Terminal Count Register	$3F
Base+$A2D	LPOSR	8	LCDC	Panning Offset Register	$00
Base+$A31	LFRCM	8	LCDC	Frame Rate Control Modulation Register	$B9
Base+$A32	LGPMR	16	LCDC	Gray Palette Mapping Register	$1073
					-
Base+$B00	HMSR	32	RTC	RTC Hours Minutes Seconds Register	$00000000
Base+$B04	ALARM	32	RTC	RTC Alarm Register	$00000000
					-
Base+$B0C	CTL	8	RTC	RTC Control Register	$00
Base+$B0E	ISR	8	RTC	RTC Interrupt Status Register	$00
Base+$B10	IENR	8	RTC	RTC Interrupt Enable Register	$00
Base+$B12	STPWCH	8	RTC	Stopwatch Minutes	$00

Note

The base is $FFFFF000 and $FFF000 from reset. If the double-mapped bit is cleared in the SCR, then the base is $FFFFF000. Do not access any space within the 4K register space that is not defined in the above table. Unpredictable results may occur.

SECTION 2
SYSTEM INTEGRATION MODULE

The MC68328 processor system integration module (SIM28) consists of several functions that control the system startup, initialization, configuration, and the external bus with minimum glue logic. The SIM28 contains the following functions:

- System configuration
- Chip-selects and wait states
- External bus interfaces
- Interrupt configuration / response
- PCMCIA V1.0 memory card support

2.1 MODULE OPERATION

The various SIM28 internal function blocks and their operation are described here along with methods and recommendations to program the various register locations that allow users to configure the MC68328 processor for their target systems.

2.1.1 MC68328 Processor System Configuration

The MC68328 processor system configuration logic consists of a system control register (SCR) that lets users configure operation of the following functions:

- Access permission of internal peripheral registers
- Address space of internal peripheral registers
- Bus timeout control and status (bus-error generator)

The on-chip peripherals occupy a reserved 4096-byte block of address space for their registers. This block is located at $FFFFF000 and $FFF000 from reset. There is a double-map control bit in the SCR to disable this double-mapping feature. If the bit is cleared, the on-chip peripheral registers appear at the top 4K of the 4 Gbyte address range starting from $FFFFF000. The on-chip peripherals address-decode logic block diagram is shown in Figure 2-1.

2.1.1.1 SYSTEM CONTROL REGISTER FUNCTIONS. The SCR allows for various settings that influence system operation, such as power-down and oscillator control logic, bus interface, and hardware watchdog protection. It also includes status bits that allow exception-handler code to monitor the cause of exceptions and resets.

While most of the functions controlled by these register bits are described in detail elsewhere, they include those listed below.

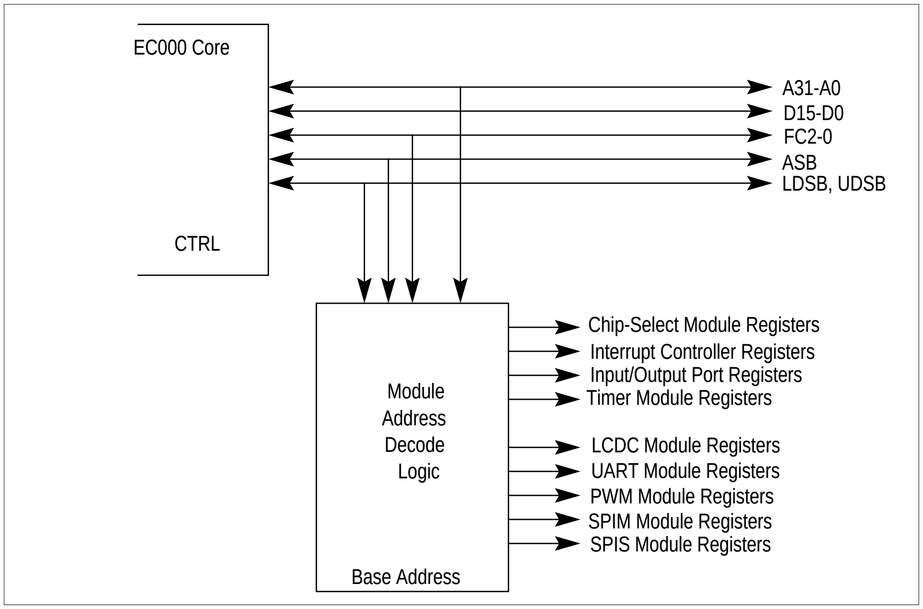


Figure 2-1. Chip-Select Logic

2.1.1.2 SYSTEM PROTECTION FUNCTIONS. The hardware watchdog (bus timeout monitor) and the software watchdog timer provide system protections.

The hardware watchdog provides a bus monitor that causes a bus error (assert $\overline{\text{BERR}}$ internally) when a bus cycle is not terminated by $\overline{\text{DTACK}}$ after 128 clock cycles have elapsed. The bus-error timeout (BETO) status bit in the SCR will also be set so that a bus-error exception handler can determine the cause of the bus error.

The bus-error timeout logic consists of a watchdog counter that, when enabled, begins to count clock cycles as $\overline{\text{AS}}$ is asserted (for internal or external bus accesses). The negation of $\overline{\text{AS}}$ normally terminates the count; however, if the count reaches terminal count before $\overline{\text{AS}}$ is negated, $\overline{\text{BERR}}$ is asserted until $\overline{\text{AS}}$ is negated. The bus-error timeout logic uses one control bit and one status bit in the SCR.

To operate the software watchdog timer, refer to the timer section.

2.1.2 Chip-Select and Wait-State Logic

The MC68328 processor provides a set of 16 general-purpose, programmable, chip-select signals arranged as 4 groups of 4 that includes two special-purpose chip-select signals. Each of the general-purpose signals can address an entire 32-bit address range and has a common set of features. The $\overline{\text{CSA0}}$ is special in the sense that it is also a boot device chip-select. From reset, all the addresses are mapped to device $\overline{\text{CSA0}}$ until the group-base address A is programmed and the valid bit is set. From that point on, $\overline{\text{CSA0}}$ does not decode globally and it is only asserted when decoded from the programming information in the group-base address register, group-base address mask register, chip-select base register, and chip-select option register. The $\overline{\text{CSD3}}$ is special in the sense that it is associated with the PCMCIA support logic. When the $\overline{\text{CSD3}}$ is asserted, the PCMCIA control signals are asserted. For each memory area, users also may define an internally generated cycle-termination signal ($\overline{\text{DTACK}}$) with a programmable number of wait states. This feature eliminates board space that would otherwise be necessary for cycle-termination logic.

The 16 general-purpose chip-selects allow up to 4 different classes of devices/memory for use in a system without external decode or wait-state generation logic. For example, to address a 16 Mbyte ROM space, 4 pairs of 16-Mbit ROM chips are used ($\overline{\text{CSA0}}$ through $\overline{\text{CSA3}}$ are connected to those 4 pairs of ROMs). Another typical configuration could be an 8-bit EPROM, a fast 16-bit SRAM, up to 4 simple I/O peripherals, and a nonvolatile flash memory.

The chip-select block diagram is shown in Figure 2-2.

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 $\overline{\text{AS}}$ goes low. The address match is described in terms of a base address and an address mask. Thus, the byte size of the matching block must be a power of 2, and the base address must be an integer multiple of this size. Therefore, an 8-Kbyte block size must begin on an 8-Kbyte boundary, and a 64-Kbyte block size can only begin on a 64-Kbyte boundary, etc.

Each chip-select is programmable and the registers have read-write capability so that the values programmed can be read back.

For a given chip-select, users may also choose: (1) whether the chip-select allows read-only or read/write accesses, (2) whether a $\overline{\text{DTACK}}$ is automatically generated for this chip-select, and (3) the number of wait states (from zero to six).

2.1.2.1 PROGRAMMABLE DATA-BUS SIZE. Each of the chip-selects includes the facility of a data-bus port-sizing extension to the basic M68000 bus. This allows the system designer to mix 16-bit and 8-bit contiguous address memory devices (RAM, ROM) on a 16-bit data-bus system. If the CPU core performs a 16-bit data transfer in an 8-bit memory space, then two 8-bit cycles will occur. However, the address strobe and data strobe remain asserted until the end of the second 8-bit cycle. In this case, only the external MC68EC000 data bus upper byte (D15-D8) is used; the least significant bit of 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/written.

Note that a 16-bit data bus is always used internally for access to peripheral registers, regardless of any mode settings for the external bus. Where internal peripheral registers are 16-bits wide, they can be read or written only in one bus cycle. This eliminates possible conflicts and reading of inaccurate values where 16-bit-wide register contents are volatile (timer counter registers, for example) or where the whole 16-bit value affects some aspect of system operation (chip-select base address, for example).

It is recommended that any external peripheral that needs only an 8-bit data-bus interface but does not require contiguous address locations: (1) use a chip-select configured as 16-bit data-bus width, and (2) connect to D7-D0. This balances more evenly the load on the two halves of the data bus in an 8-bit system.

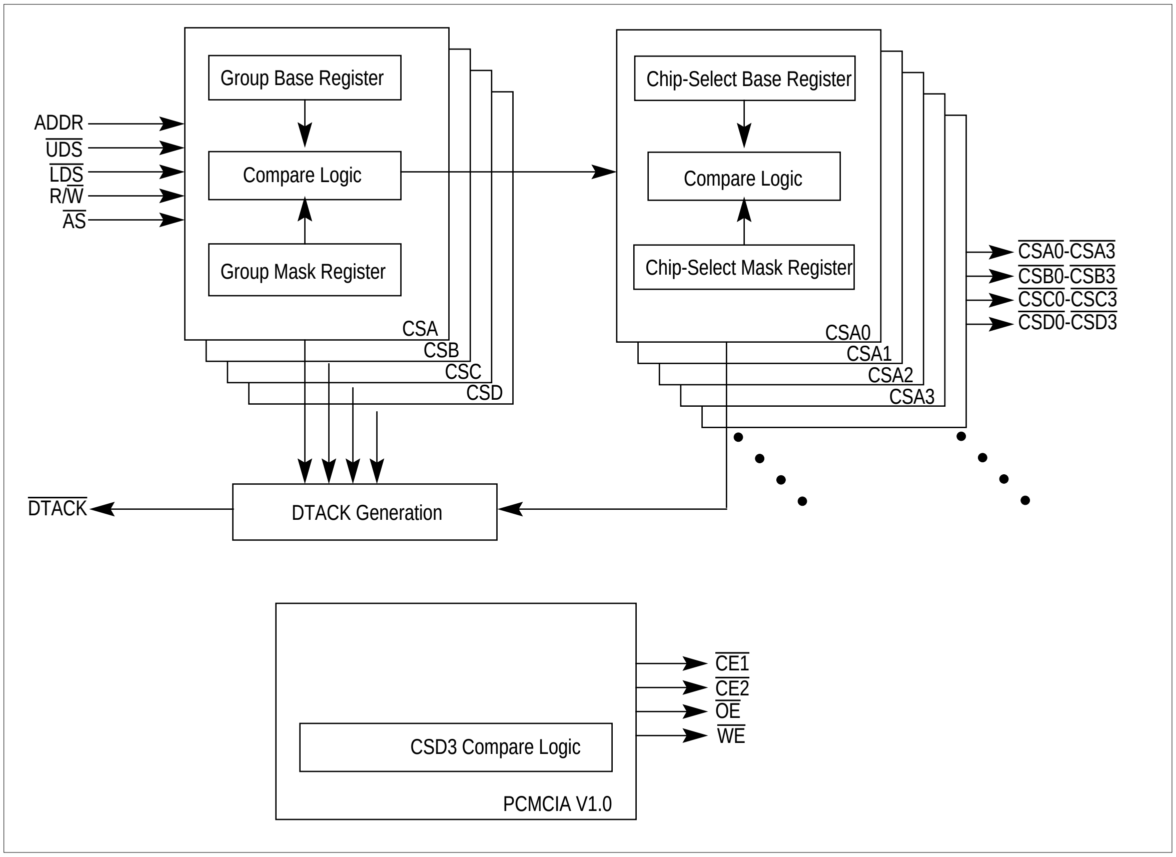


Figure 2-2. Chip Select Block

The default for each chip-select is a 16-bit data-bus width. The BUSW bits in the chip-select option registers enable 16-bit/8-bit data-bus width for each of the 16 chip-select ranges. The initial bus width for the boot chip-select is selected by placing on the BBUSW pin at reset a logic 0 or 1 to specify 8-bit or 16-bit wide data bus, respectively. This allows a boot EPROM of either data-bus width to be used in any given system.

All external accesses that do not match one of the chip-select address ranges will be assumed to be a 16-bit device. That is just one access performed for a 16-bit transfer. It can also be a 8-bit port but accessed at every other byte.

The boot chip-select is initialized from cold reset to assert in response to any address except the on-chip module register space (i.e. $FFFFF000 to $FFFFFFFF or $FFF000 to $FFFFFF). This ensures a chip-select to the boot ROM or EPROM to fetch the reset vector and execute the initialization code, which should set up the SCR and the chip-select ranges early on in that initialization sequence.

The data-bus port size for CSA0 on reset, and hence the data width of the boot ROM device, is programmed by placing on the BBUSW pin during reset logic 0 or 1 for 8-bit and 16-bit wide data bus, respectively.

The other chip-selects are initialized to be nonvalid, and so will not assert until they are programmed and the valid bits set.

2.1.2.2 OVERLAP IN CHIP-SELECT RANGES. Users should take care when programming the group address and chip-select registers. If they are programmed to overlap, the CS signals will overlap.

NOTE

Unused chip-selects must be programmed to 0 wait states and 16-bit width. Map them to dummy space, if possible.

When the CPU attempts to write to a read-only location, as programmed by users when setting up the chip-selects, the chip-select signal is not asserted, no DTACK is asserted and BERR is asserted if the bus-error timer is enabled.

NOTE

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

The programming of the chip-select module is discussed later in this chapter. For additional information, refer to **2.5 Chip-Select Registers**.

2.2 PROGRAMMING MODEL

The various modules in the MC68328 processor, including the SIM28, contain registers that control the modules and provide status information from the modules. All of these registers reside in the top 4096-byte range ($FFFFF000 to $FFFFFFFF) of addresses in the memory map of the MC68EC000 core processor. It is also doubly mapped at $FFF000 to $FFFFFF from reset.

2.2.1 System Control Register (SCR)

The SCR can be read or written at any time by 8-bit or 16-bit transfers. An 8-bit read/write location, it resides at address hex $FFF000 or $FFFFF000 in supervisor data space. The SCR cannot be accessed in user data space if the supervisor-only bit (SO) is set. The SO bit is set to 1 after reset. The register consists of 3 status bits and 4 system-control bits. The bus-error timeout (BETO) status bit is normally 0 and is set to 1 by a bus timeout event in the system. Writing a 0 to these bits has no effect; writing a 1 clears the status bit.

7	6	5	4	3	2	1	0
BETO	WPV	PRV	BETEN	SO	DMAP	RSVD	WDTH8

Address: $(FF)FFF000 Reset Value: $0C

Figure 2-3. System Control Register

BETO Bus-Error Timeout

This status bit is set when the bus-error timer times out. Writing a 1 to this bit clears it while writing a 0 has no effect.

WPV Write-Protect Violation

This status bit is set when a write-protection violation occurs; that is, when there is a write to a write-protected area. Writing a 1 to this location clears the status bit while writing a 0 has no effect.

PRV Privilege Violation

This status bit is set when accessing a supervisor-only area in user mode. Writing a 1 to this bit clears the status bit while writing a 0 has no effect.

BETEN Bus-Error Timeout Enable

The bus-error timeout function is enabled when this control bit is set. BERR is asserted on the bus cycle if the bus timer reaches the terminal count and the $\overline{\text{DTACK}}$ is not asserted.

SO Supervisor Only

This supervisor-only control bit limits the on-chip register accesses to supervisor only. Writing a 1 to this location sets to supervisor-only mode. Clearing this bit allows both supervisor and user to access the on-chip registers.

DMAP Double Map

If this bit is set, the register is mapped at $FFFFF000-$FFFFFFFF and $FFF000-$FFFFFF. If the bit is cleared, the register is mapped at $FFFFF000-$FFFFFFFF only. This bit is set to 1 after reset.

RSVD Reserved

Reserved bit. Should be 0 in normal operating mode.

WDTH8 8-Bit Width Select

This control bit should be set to 1 when the system is an 8-bit only system. This allows D7-D0 pins to be used for port B I/O.

2.3 INTERRUPT CONTROLLER BLOCK

The interrupt controller supports a variety of interrupts, both internal and external. This block prioritizes and encodes pending interrupts. It also generates vectors during the interrupt-acknowledge cycle.

2.3.1 Interrupt Controller Overview

The interrupt-controller block supports 23 interrupts. Both edge- and level-sensitive interrupts are supported. A programmable vector can be generated for each interrupt level. Interrupt sources include the following:

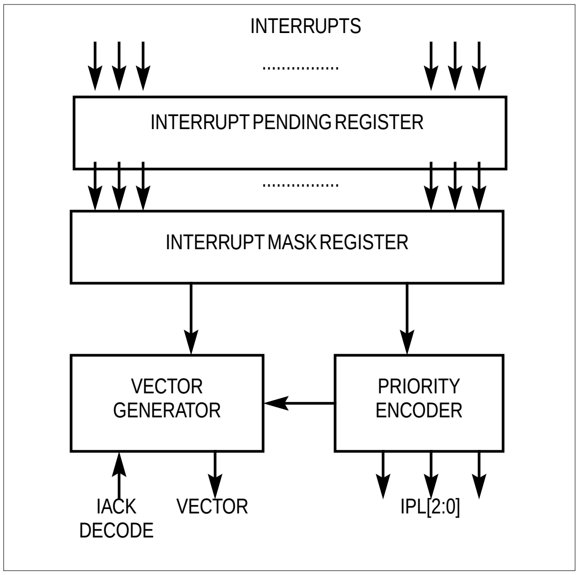


Figure 2-4. Interrupt Controller Block Diagram

- $\overline{\text{IRQ7}}$ (level 7)
- SPI slave needs service (level 6)
- Timer 1 event (level 6)
- $\overline{\text{IRQ6}}$ external interrupt (level 6)
- $\overline{\text{PENIRQ}}$ (level 5)
- SPI master needs service (level 4)
- Timer 2 event (level 4)
- UART needs service (transmit or receive) (level 4)
- Watchdog timer interrupt (level 4)
- Real-time clock interrupt (level 4)
- Keyboard interrupt (level 4)
- PWM period rollover (level 4)
- General purpose interrupt $\overline{\text{INT0}}$-$\overline{\text{INT7}}$ (level 4) - these pins are muxed with keyboard
- $\overline{\text{IRQ3}}$ external interrupt (level 3)
- $\overline{\text{IRQ2}}$ external interrupt (level 2)
- $\overline{\text{IRQ1}}$ external interrupt (level1)

Users have access to the upper five bits of the vector that allow placement of the interrupt-vector table anywhere in the vector space.

The interrupt controller gathers, prioritizes, and posts interrupts to the M68000 core. Full compatibility with normal M68000 operation is maintained.

Here’s a typical scenario: when an interrupt is received, it is prioritized. Assuming there are no higher interrupts pending, it is posted to the M68000 core. The core responds with an interrupt-acknowledge cycle. The interrupt controller generates a vector and presents it to the core, which then jumps to the interrupt-service routine. If the core is in an interrupt-service routine and a higher priority interrupt is posted, the process repeats and the higher priority is serviced. Interrupt priority is based on the interrupt level. Interrupts with the same interrupt level are prioritized by the software during the execution of the interrupt-service routine. The MC68328 processor provides one interrupt vector for each interrupt level. The most significant 5 bits of the interrupt vector are user-programmable while the lower 3 bits reflect the interrupt level 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, its status is accessible in the interrupt-pending register.

2.3.2 Programmer's Model

This section describes the bits and registers that control the interrupt module. All unused bits may be written with no effect. When they are read, they indicate 0.

2.3.2.1 INTERRUPT VECTOR REGISTER (IVR). This register programs the upper 5 bits of the interrupt vector.During the interrupt-acknowledge cycle, the lower 3 bits encode the interrupt level.

7	6	5	4	3	2	1	0
VECTOR					0	0	0

Address: $(FF)FFF300 Reset Value: $00

Figure 2-5. Interrupt Control Register

NOTE

If the interrupt-vector register is never programmed and an interrupt occurs, the interrupt vector $0F is returned to indicate an uninitialized interrupt.

2.3.2.2 VECTOR GENERATOR. The interrupt controller block provides a vector to the MC68EC000 core. Users have access to the upper 5 bits of the vector to place the vector table anywhere in the vector space. Coding for the vectors is as follows:

Table 2-1. Interrupt Vectors

Interrupt	Vector
level 7	xxxxx111
level 6	xxxxx110
level 5	xxxxx101
level 4	xxxxx100
level 3	xxxxx011
level 2	xxxxx010
level 1	xxxxx001

2.3.2.3 INTERRUPT CONTROL REGISTER (ICR). This register controls the external-interrupt inputs. It has polarity control and edge/level programmability.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ET1	ET2	ET3	ET6	POL1	POL2	POL3	POL6	UNUSED							

Address: $(FF)FFF302 Reset Value: 0000

Figure 2-6. Interrupt Control Register

ET1 $\overline{\text{IRQ1}}$ Edge Trigger Select

While this bit is set, the external $\overline{\text{IRQ1}}$ bit is an edge-triggered interrupt. Users must clear $\overline{\text{IRQ1}}$ in the interrupt-status register to clear the interrupt; that is, writing a 1 to the $\overline{\text{IRQ1}}$ bit in the interrupt-status register. While this bit is low, $\overline{\text{IRQ1}}$ is a level-sensitive interrupt. In this case, users must clear the source of the interrupt. On reset, this bit is cleared to 0 (level-sensitive interrupt).

0 = Level-sensitive interrupt
1 = Edge-sensitive interrupt

ET2 $\overline{\text{IRQ2}}$ Edge Trigger Select

While this bit is set, the external $\overline{\text{IRQ2}}$ bit is an edge-triggered interrupt. Users must clear $\overline{\text{IRQ2}}$ in the interrupt-status register to clear the interrupt. While this bit is low, $\overline{\text{IRQ2}}$ is a level-sensitive interrupt. In this case, users must clear the source of the interrupt. On reset, this bit is cleared to 0 (level-sensitive interrupt).

0 = Level-sensitive interrupt
1 = Edge-sensitive interrupt

ET3 $\overline{\text{IRQ3}}$ Edge Trigger Select

While this bit is set, the external $\overline{\text{IRQ3}}$ bit is an edge-triggered interrupt. Users must clear $\overline{\text{IRQ3}}$ in the interrupt-status register to clear the interrupt. While this bit is low, $\overline{\text{IRQ3}}$ is a level-sensitive interrupt. In this case, users must clear the source of the interrupt. On reset, this bit is cleared to 0 (level-sensitive interrupt).

0 = Level-sensitive interrupt
1 = Edge-sensitive interrupt

ET6 $\overline{\text{IRQ6}}$ Edge Trigger Select

While this bit is set, the external $\overline{\text{IRQ6}}$ bit is an edge-triggered interrupt. Users must clear $\overline{\text{IRQ6}}$ in the interrupt status register to clear the interrupt. While this bit is low, $\overline{\text{IRQ6}}$ is a level-sensitive interrupt. In this case, users must clear the source of the interrupt. On reset, this bit is clear to 0 (level-sensitive interrupt).

0 = Level-sensitive interrupt
1 = Edge-sensitive interrupt

POL1 Polarity Control for Interrupt 1

0 = Negative polarity
1 = Positive polarity

POL2 Polarity Control for Interrupt 2

0 = Negative polarity
1 = Positive polarity

POL3 Polarity Control for Interrupt 3

0 = Negative polarity
1 = Positive polarity

POL6 Polarity Control for Interrupt 6

0 = Negative polarity
1 = Positive polarity

2.3.2.4 INTERRUPT MASK REGISTER (IMR). This control register masks the interrupt if the corresponding control bit is set. There is one control bit for each interrupt source. When an interrupt is masked, it does not generate an interrupt request to the processor core, but its status can still be observed in the interrupt-pending register. From reset, all the interrupts are masked and all the bits in this register are set to 1.

The interrupt-mask bit positions corresponds to the bits in the interrupt-status register, interrupt-pending register, and wakeup-enable register. When each bit is set, its interrupt is masked (disabled).

MSPIM (Mask SPI Master Interrupt, Bit 0)

This bit, while set, indicates that the SPI master interrupt is masked. It is set to 1 after reset.

0 = Enable SPI master interrupt
1 = Mask SPI master interrupt

MTMR2 (Mask Timer 2 Interrupt, Bit1)

This bit, while set, indicates that the timer-2 interrupt is masked. It is set to 1 after reset.

0 = Enable timer-2 master interrupt
1 = Mask timer-2 master interrupt

MUART (Mask UART Interrupt, Bit2)

This bit, while set, indicates that the UART interrupt is masked. It is set to 1 after reset.

0 = Enable UART interrupt
1 = Mask UART interrupt

MWDT (Mask Watchdog Timer Interrupt, Bit 3)

This bit, while set, indicates that the watchdog timer interrupt is masked. It is set to 1 after reset.

0 = Enable WDT interrupt
1 = Mask WDT interrupt

RESERVED

This bit is not used and reads 0.

MRTC (Mask RTC Interrupt, Bit 4)

This bit, while set, indicates that the real-time clock interrupt is masked. It is set to 1 after reset.

0 = Enable RTC interrupt
1 = Mask RTC interrupt

MKB (Mask Keyboard Interrupt, Bit 6)

This bit, while set, indicates that the keyboard interrupt is masked. It is set to 1 after reset.

0 = Enable KB interrupt
1 = Mask KB interrupt

MPWM (Mask PWM Interrupt, Bit 7)

This bit, while set, indicates that the PWM interrupt is masked. It is set to 1 after reset.

0 = Enable PWM interrupt
1 = Mask PWM interrupt

MINT0 (Mask External INT0, Bit 8)

This bit, while set, indicates that the external interrupt INT0 is masked. It is set to 1 after reset.

0 = Enable INT0 interrupt
1 = Mask INT0 interrupt

MINT1 (Mask External INT1, Bit 9)

This bit, while set, indicates that the external interrupt INT1 is masked. It is set to 1 after reset.

0 = Enable INT1 interrupt
1 = Mask INT1 interrupt

MINT2 (Mask External INT2, Bit 10)

This bit, while set, indicates that the external interrupt INT2 is masked. It is set to 1 after reset.

0 = Enable INT2 interrupt
1 = Mask INT2 interrupt

MINT3 (Mask External INT3, Bit 11)

This bit, while set, indicates that the external interrupt INT3 is masked. It is set to 1 after reset.

0 = Enable INT3 interrupt
1 = Mask INT3 interrupt

MINT4 (Mask External INT4, Bit 12)

This bit, while set, indicates that the external interrupt INT4 is masked. It is set to 1 after reset.

0 = Enable INT4 interrupt
1 = Mask INT4 interrupt

MINT5 (Mask External INT5, Bit 13)

This bit, while set, indicates that the external interrupt INT5 is masked. It is set to 1 after reset.

0 = Enable INT5 interrupt
1 = Mask INT5 interrupt

MINT6 (Mask External INT6, Bit 14)

This bit, while set, indicates that the external interrupt INT6 is masked. It is set to 1 after reset.

0 = Enable INT6 interrupt
1 = Mask INT6 interrupt

MINT7 (Mask External INT7, Bit 15)

This bit, while set, indicates that the external interrupt INT7 is masked. It is set to 1 after reset.

0 = Enable INT7 interrupt
1 = Mask INT7 interrupt

MIRQ1 (Mask IRQ1 Interrupt, Bit 16)

This bit, while set, indicates that the external IRQ level-1 interrupt is masked. It is set to 1 after reset.

0 = Enable IRQ1 interrupt
1 = Mask IRQ1 interrupt

MIRQ2 (Mask IRQ2 Interrupt, Bit 17)

This bit, while set, indicates that the external IRQ level-2 interrupt is masked. It is set to 1 after reset.

0 = Enable IRQ2 interrupt
1 = Mask IRQ2 interrupt

MIRQ3 (Mask IRQ3 Interrupt, Bit 18)

This bit, while set, indicates that the external IRQ level-3 interrupt is masked. It is set to 1 after reset.

0 = Enable IRQ3 interrupt
1 = Mask IRQ3 interrupt

MIRQ6 (Mask IRQ6 Interrupt, Bit 19)

This bit, while set, indicates that the external IRQ level-6 interrupt is masked. It is set to 1 after reset.

0 = Enable IRQ6 interrupt
1 = Mask IRQ6 interrupt

MPEN (Mask Pen Interrupt, Bit 20)

This bit, while set, indicates that the pen-down interrupt is masked. It is set to 1 after reset.

0 = Enable pen-down interrupt
1 = Mask pen-down interrupt

MSPIS (Mask SPI Slave Interrupt, Bit 21)

This bit, while set, indicates that the SPI slave interrupt is masked. It is set to 1 after reset.

0 = Enable SPIS interrupt
1 = Mask SPIS interrupt

MTMR1 (Mask Timer 1 Interrupt, Bit 22)

This bit, while set, indicates that the Timer-1 interrupt is masked. It is set to 1 after reset.

0 = Enable timer-1 interrupt
1 = Mask timer-1 interrupt

MIRQ7 (Mask IRQ7 Interrupt, Bit 23)

This bit, while set, indicates that the IRQ7 interrupt is masked. It is set to 1 after reset. The IRQ7 is nonmaskable in the sense that if the interrupt level is masked all the way to level 7 in the M68000 core, the IRQ7 interrupt is still observed by the processor because the level 7 interrupt is nonmaskable for the M68000 core. It can, however, be masked by this control bit.

0 = Enable IRQ7 interrupt
1 = Mask IRQ7 interrupt

2.3.2.5 INTERRUPT WAKEUP-ENABLE REGISTER (IWR). This control register enables the corresponding interrupt source to start the power-control-wakeup sequence. While the bit is set, it enables that interrupt to cause wakeup; while the bit is clear, the function is disabled. After reset, all wakeups are enabled and all the bits in this register are set to 1.

2.3.2.6 INTERRUPT STATUS REGISTER (ISR). This register indicates which interrupts are asserting to the processor. When an interrupt vector is passed to the MC68EC000 core, the interrupt-handler routine can determine the source of interrupt by examining this status register. If there are multiple interrupt sources at that level, the software can prioritize them at that time. There is one interrupt vector for each interrupt level. The lower three bits of the interrupt vector constitute the interrupt level being acknowledged

.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNUSED								IRQ7	TMR1	SPIS	PEN	IRQ6	IRQ3	IRQ2	IRQ1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	PWM	KB	LCDC	RTC	WDT	UART	TMR2	SPIM

Address: $(FF)FFF30C Reset Value: $00000000

Figure 2-7. Interrupt Status Register

All the interrupt status bits, except IRQs 1, 2, 3, 6, and 7, reflect the interrupt request from their respective interrupt sources. A status bit is cleared when the interrupt request is cleared from the requesting module or input pin. Refer to the timer, SPIM, SPIS, RTC, and PWM sections for details on clearing an interrupt request from those modules. The $\overline{\text{IRQ7}}$ is an active low-edge triggered interrupt request. Its status is cleared by writing a 1 to the $\overline{\text{IRQ7}}$ interrupt status bit. The IRQs 1, 2, 3, and 6 are edge/level programmable. These interrupt bits are cleared by writing a 1 to the corresponding interrupt status bit if they are programmed as edge-triggered.

$\overline{\text{IRQ6}}$

This bit, while set, indicates that an external device is requesting an interrupt on level 6. If IRQ6 is set to be a level-sensitive interrupt, users must clear the source of the interrupt. If IRQ6 is set to be an edge-sensitive interrupt, users must clear the interrupt by writing a 1 to this status bit. Writing a 0 to this bit and the other bits of this register has no effect.

0 = No level-6 interrupt pending
1 = Level-6 interrupt pending

UART

This bit, while set, indicates that the UART module needs service. The transmitter might need data, the receiver might have data ready to transfer to memory, or the CTS or GPIO pins might have changed state. Each of these interrupts is maskable in the UART control register. Refer to the UART description for details. This interrupt is a Level 4 interrupt.

0 = No UART service request pending
1 = UART service needed

SPIM

SPIM indicates that a data transfer completed by setting this bit high. Users must clear this interrupt in the SPI control register. Refer to the SPI section for details. This interrupt is a Level 4 interrupt.

0 = No SPI master interrupt
1 = SPI master interrupting

PWM

This bit indicates the PWM period rollover. Refer to the PWM section for details. This is a Level 4 interrupt.

0 = No PWM period rollover
1 = PWM period rolled over

TIMER1

This bit indicates that a timer-1 event occurred. Please refer to the timer section for operation details. This is a Level 6 interrupt.

0 = No timer-1 events
1 = Timer-1 events

TIMER2

This bit indicates that timer-2 timed out. Refer to the timer section for operation details. This is a Level 4 interrupt.

0 = No timer-2 events
1 = Timer-2 events

$\overline{\text{IRQ3}}$

This bit, while set, indicates that an external device requests an interrupt on Level 3. If $\overline{\text{IRQ3}}$ is set to be a level-sensitive interrupt, users must clear the source of the interrupt. If $\overline{\text{IRQ3}}$ is set to be an edge-sensitive interrupt, users must clear the interrupt by writing a 1 to this status bit. Writing a 0 to this bit has no effect.

0 = No Level 3 interrupt pending
1 = Level 3 interrupt pending

$\overline{\text{IRQ2}}$

This bit, while set, indicates that an external device requests an interrupt on Level 2. If $\overline{\text{IRQ2}}$ is set to be a level-sensitive interrupt, users must clear the source of the interrupt. If $\overline{\text{IRQ2}}$ is set to be an edge-sensitive interrupt, users must clear the interrupt by writing a 1 to this status bit. Writing a 0 to this bit and the remainder of the bits in this register has no effect.

0 = No Level 2 interrupt pending
1 = Level 2 interrupt pending

$\overline{\text{IRQ1}}$

This bit, while set, indicates that an external device requests an interrupt on Level 1. If $\overline{\text{IRQ1}}$ is set to be a level-sensitive interrupt, users must clear the source of the interrupt. If $\overline{\text{IRQ1}}$ is set to be an edge-sensitive interrupt, users must clear the interrupt by writing a 1 to this status bit. Writing a 0 to this bit and the remainder of the bits in this register has no effect.

0 = No Level 1 interrupt pending
1 = Level 1 interrupt pending

$\overline{\text{IRQ7}}$

This bit, while set, indicates that an external device requests an interrupt on Level 7. The $\overline{\text{IRQ7}}$ is an active low-edge sensitive interrupt, so users must clear the interrupt by writing a 1 to this status bit. Writing a 0 to this bit and the remainder of the bits in this register has no effect.

0 = No Level 7 interrupt pending
1 = Level 7 interrupt pending

2.3.2.7 INTERRUPT-PENDING REGISTER (IPR). This read-only register indicates which interrupts are pending. If an interrupt source requests an interrupt but that interrupt is masked by the interrupt-mask register, that interrupt bit will be set in the interrupt-pending register, but not in the interrupt-status register. If it is not masked, the interrupt bit will be the same in both registers.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNUSED								IRQ7	TMR1	SPIS	PEN	IRQ6	IRQ3	IRQ2	IRQ1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	PWM	KB	LCDC	RTC	WDT	UART	TMR2	SPIM

Address: $(FF)FFF310 Reset Value: none

Figure 2-8. Interrupt-Pending Register

2.4 KEYBOARD INTERRUPT I/O

Keyboard-interrupt features provide smart power management. The MC68EC000 can be placed in sleep mode when no keys are pressed. When a key is pressed, the MC68EC000 can wake up and serve the user's request. This event-driven approach significantly reduces power consumption. $\overline{\text{KB0}}$ to $\overline{\text{KB7}}$ are input pins for the keyboard interface. A logical OR operation is performed on these inputs to generate an interrupt, indicating to the MC68EC000 core that a key was pressed.

2.5 CHIP-SELECT REGISTERS

The following paragraphs describe the registers in the chip-select function and an example of how to program the registers. The chip-select function does not operate until the register is initialized and the valid bit is set in the corresponding group-base address registers. The only exception is the $\overline{\text{CSA0}}$, which is the boot device chip-select.

2.5.1 Group-Base Address Registers (GBR0-GBR3)

There are four 16-bit group-base address registers in the chip-select function (one for each chip-select group). The group registers (group-base address register and group-base address mask register) decode the upper address bits and the chip-select option registers decode the lower address bits. There are 4 chip-selects in each group. For example, in group A, the chip-selects are $\overline{\text{CSA0}}$, $\overline{\text{CSA1}}$, $\overline{\text{CSA2}}$, and $\overline{\text{CSA3}}$.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GROUP BASE ADDRESS (GBA31-GBA20)												UNUSED			V
GBA31	GBA30	GBA29	GBA28	GBA27	GBA26	GBA25	GBA24	GBA23	GBA22	GBA21	GBA20	0	0	0	0

Address: $(FF)FFF100, 102, 104, 106 Reset Value: $0000

Figure 2-9. Group-Base Address Registers

GROUP BASE ADDRESS (GBA31-GBA20)

The group-base address field (the upper 12 bits of each base address register) selects the starting address for the group address range. The corresponding bits GBA31–GBA20 in the group-base mask register define the block size for the group. The base address field is compared to the address on the address bus to determine if the group is decoded.

V Valid Bit

This bit indicates the validity of the contents of its base address register and address mask register pair. The programmed chip-selects do not assert until the V-bit is set. A reset clears the V-bit in each base address register.

1 = Contents are valid
0 = Contents are not valid

2.5.2 Group-Base Address Mask Registers

These registers adjust the comparison range for the group. Bits in this register set to 1 always compare true with the corresponding address line.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GROUP BASE MASK (GMA31-GMA20)												UNUSED			
GMA31	GMA30	GMA29	GMA28	GMA27	GMA26	GMA25	GMA24	GMA23	GMA22	GMA21	GMA20	0	0	0	0

Address: $(FF)FFF108, 10A, 10C, 10E Reset Value: $0000

Figure 2-10. Group-Base Address Mask Registers

GROUP BASE MASK (GMA31-GMA20)

0 = The corresponding address bit must match for a group to match
1 = The corresponding address bit compares true (don’t care)

These bits mask address A31 through A20. Usually, the group base address mask register bits select the group size. For example, if all the mask bits are cleared, A31 to A20 are compared against the value programmed in the group base address register. In this case, the group has a 1 Mbyte space. If bit 4 (GMA20 bit) is set and the remainder of the bits are clear, the group is selected if A31 to A21 are the same as the value programmed in the group-base address register. This provides 2 Mbyte of space for the group. Further decoding is performed for each chip-select by comparing lower address lines and the chip-select registers. If the devices are small, it is possible to program the groups to the same space and use the chip-select registers to decode the areas for each chip-select.

2.5.3 CHIP-SELECT REGISTERS

There are four 32-bit chip-select option registers in each chip-select group—one for each chip-select signal.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDRESS COMPARE A23-A16								UNUSED							BSW
AC23	AC22	AC21	AC20	AC19	AC18	AC17	AC16	-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS MASK 23-16								UNUSED				RO	WAIT		
AM23	AM22	AM21	AM20	AM19	AM18	AM17	AM16	-	-	-	-	-	-	-	-

Address: $(FF)FFF110, 114, 118, 11C, 120, 124, 128, 12C Reset Value: $00010006

Figure 2-11. Chip-Select Registers for Group A and B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDRESS COMPARE 23-12												UNUSED			BSW
AC23	AC22	AC21	AC20	AC19	AC18	AC17	AC16	AC15	AC14	AC13	AC12	-	-	-	-
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS MASK 23-12												RO	WAIT		
AM23	AM22	AM21	AM20	AM19	AM18	AM17	AM16	AM15	AM14	AM13	AM12	-	-	-	-

Address: $(FF)FFF130, 134, 138, 13C, 140, 144, 148, 14C Reset Value: $00010006

Figure 2-12. Chip-Select Registers for Group C and D

Chip selects in group A and B (i.e. $\overline{\text{CSA0}}$, $\overline{\text{CSA1}}$, $\overline{\text{CSA2}}$, $\overline{\text{CSA3}}$, $\overline{\text{CSB0}}$, $\overline{\text{CSB1}}$, $\overline{\text{CSB2}}$, $\overline{\text{CSB3}}$) decode address A23-A16 (minimum 64K of space). Chip selects in group C and D decode address A23-A12 (minimum 4K of space).

ADDRESS COMPARE 23-16 (Group A, B)

This bit field is the address-compare field. A group address match and a match of address bits 23-16 generate this chip-select. Notice some of the address bits overlap in the group base address/mask registers and the chip-select register. This allows for a large group to be selected and for chip-select to be finely decoded.

ADDRESS MASK 23-16

This field masks corresponding bits in the address-compare field. A ‘‘1’’ forces a true comparison (don’t care) on the corresponding bit.

ADDRESS COMPARE 23-12 (Group C, D)

This bit field is the address-compare field. A group address match and a match of address bits 23-12 generate this chip-select.

ADDRESS MASK 23-12

This field masks corresponding bits in the address compare field. A “1” forces a true comparison (don’t care) on the corresponding bit.

BSW Bus Width

This bit sets the bus width for this chip-select area.

0 = 8-bit
1 = 16-bit

RO Read Only

This bit sets the chip-select to read only. Otherwise, read and write accesses are allowed. Writes to read-only areas generate a bus error.

0 = Read/Write
1 = Read Only

WAIT

This field determines the number of wait states added before an internal $\overline{\text{DTACK}}$ is returned for this chip-select.

000 = Zero wait states
001 = One wait state
010 = Two wait states
011 = Three wait states
100 = Four wait states
101 = Five wait states
110 = Six wait states
111 = External DTACK

2.6 PCMCIA 1.0 SUPPORT

The MC68328 processor supports PCMCIA 1.0 memory card chip-selects and read / write signals. To meet the fanout requirement, use external buffers to interface to the memory card.

2.6.1 Block Diagram Overview

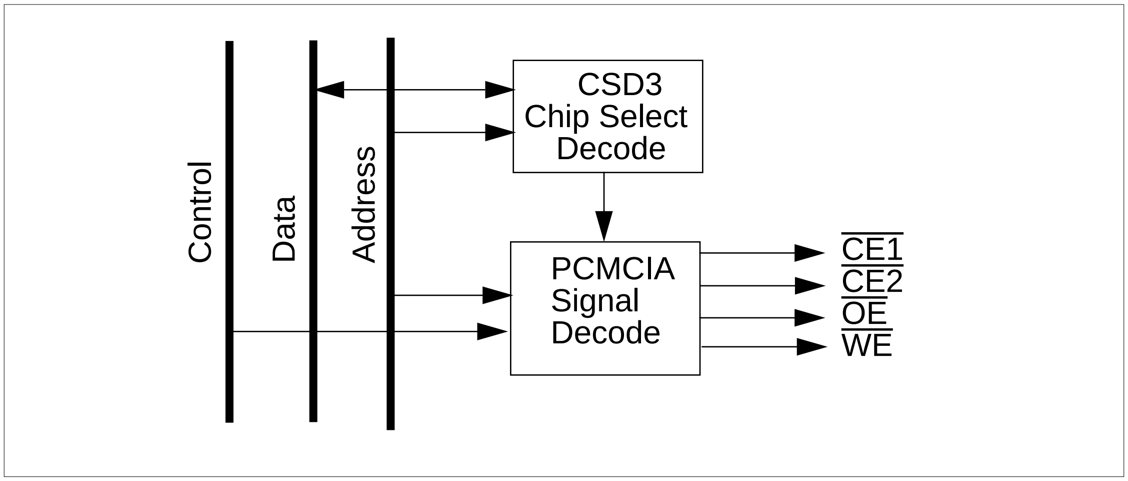


Figure 2-13. PCMCIA Block Diagram

The PCMCIA address decode is through $\overline{\text{CSD3}}$. Selecting $\overline{\text{CSD3}}$ assets the corresponding PCMCIA control signals.

SECTION 3
PHASE-LOCKED LOOP AND POWER CONTROL

The phase-locked-loop (PLL) block generates all clocks for the MC68328 processor. It includes a crystal oscillator for use with low-frequency (32.768 kHz) crystals. The PLL generates a high-frequency master clock phase-locked to the crystal reference.

3.1 OVERVIEW

The PLL is a flexible clock source for the MC68328. It provides a crystal-controlled master clock at frequencies from 10MHz to the maximum operational frequency in 32-kHz steps. The master clock can be divided to provide a system clock as low as 1/16th of the voltage-controlled oscillator (VCO) frequency. The low-frequency reference clock (32.768 kHz or 38.4 kHz) is always available to the real-time clock or timer. The PLL can be disabled to save power, but it can be re-enabled within 2 ms of a wake-up interrupt. This block, in conjunction with the power-control block, provides an efficient power-control mechanism for the MC68328 processor (see Figure 3-1 below).

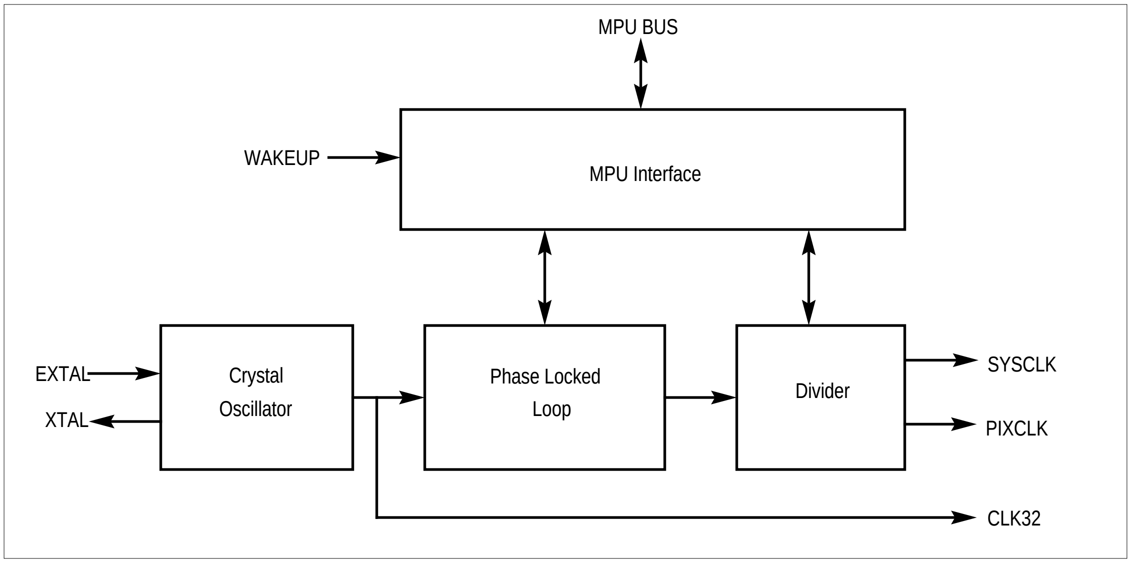


Figure 3-1. PLL Block Diagram

3.2 PROGRAMMER'S MODEL

The PLL has three registers that provide complete control and status information. Descriptions of these registers follow.

3.2.1 PLL Control Register

This register (illustrated in Figure 3-2) controls the overall PLL operation. Several bits are provided for control of the dynamic performance of the PLL. Refer to **Section 3.4.3** for operation details.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED		PIXCLK SEL			SYSCLK SEL			UNUSED			CLKEN	DISPLL	RSVD		

Address: $(FF)FFF202 Reset Value: $2410

Figure 3-2. PLL Control Register

PIXCLK SEL

These bits select the master frequency for the LCD pixel clock. The master clock is derived from the VCO frequency as shown by the list below.

000 = VCO / 2
001 = VCO / 4
010 = VCO / 8
011 = VCO / 16
1XX = VCO / 1 (binary 100 after reset)

SYSCLK SEL

These bits select the master frequency for the MC68328 processor system clock. The master clock is derived from the VCO frequency as shown by the list below.

000 = VCO / 2
001 = VCO / 4
010 = VCO / 8
011 = VCO / 16
1XX = VCO / 1 (binary 100 after reset)

These bits can be changed at any time. The VCO frequency is unaffected by changes.

CLKEN

This bit enables the CLKO pin while high.

1= CLKO enabled
0= CLKO disabled

DISPLL Disable PLL

This bit, while high, disables the PLL. The system clock is shut down and the MC68328 processor assumes its lowest power state. Only the 32 kHz clock runs. Refer to **Section 3.4.3** for a description of the preferred method for system clock shutdown. Once the PLL is disabled, only a wake-up interrupt or reset can re-enable it.

1 = PLL disabled
0 = PLL enabled

3.2.2 Frequency Select Register

This register (illustrated in Figure 3-3) controls the two dividers of the dual-modulus counter. One additional bit assists the software to protect the PLL from accidental writes that change the frequency. Another bit prepares for the VCO frequency change. While this register can be accessed as bytes, it should always be written as a 16-bit word.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLK32	PROT	UNUSED		QC				PC							

Address: $(FF)FFF202 Reset Value: $0123

Figure 3-3. Frequency-Select Register

CLK32 Clock 32

This bit indicates the current status of the CLK32 signal and synchronizes the software to the 32kHz reference clock when the VCO frequency is to be changed or the PLL is to be disabled. Refer to Section **3.3 PLL Operation** for details.

PROT Protect Bit

This bit protects the “P” and “Q” counter values from additional writes. After this bit is set by software, the frequency-select register cannot be written. Only a reset clears this bit.

QC Q Count

These bits control the “Q” counter.

PC P Count

These bits control the “P” counter.

3.3 PLL OPERATION

This section describes the operation and preferred sequences to control the PLL.

3.3.1 Initial Powerup

At initial powerup, the crystal oscillator begins oscillation within several hundred milliseconds. While reset remains asserted, the PLL begins the lockup sequence and locks within several milliseconds of the crystal oscillator startup. Once lockup occurs, the system clock is available at the default master frequency of 16.580608 MHz (assuming a 32.768 kHz crystal). To generate the master frequency, multiply the reference (32.768 kHz) by the PLL divisor. The default divisor is 506. The divisor can be changed under software control and is outlined below.

NOTE

The default divider value (506) was selected as it can directly generate standard baud frequencies at accuracies of better than 0.01%.

3.3.2 Divider

The PLL uses a dual-modulus prescaler to reduce power consumption. This approach divides the VCO frequency by 14 before it is fed to the rest of the divider chain. Dual-modulus counters operate differently from other counters in that the overall divide ratio is dependent on two separate values, P and Q. Besides the power-saving advantage above a divisor of 225 (decimal), every divisor is available to fine-tune the VCO in 32 kHz steps. The formula for the dual-modulus divider is:

Divisor = 14 (P + 1) + Q + 1

Where:

1 <= Q <= 14

P >= Q + 1

Below the value of 225, some divisors are not allowed as the P and Q relationships cannot be met.

3.3.3 Normal Startup

When the MC68328 processor is awakened from sleep mode by a system interrupt, the PLL achieves lock within a few milliseconds. The crystal oscillator is always on after initial powerup, so the crystal startup time is not a factor. The master clock starts operation after the PLL achieves lock.

3.3.4 Change of Frequency

To change the VCO frequency, use the sequence below. As the system clock is disabled while the PLL loses then reacquires lock, disable all peripheral devices before making changes to the frequency.

This fragment assumes all peripherals have been disabled and the CPU is operating at the highest possible frequency (SYSCLK SEL = 7). FREQSEL is the address of the frequency-select register. NEWFREQ is the new frequency value (P and Q values) to be programmed.

```
              lea #$FFF202,A0         point to the Freq Sel Register
              move.w #NEWFREQ,D1      prepare the new frequency (Q and P)
WAIT          move.w (A0),D0          get the contents of the register
              bpl.w WAIT              wait for CLK32 to go high
              move.w D1,(A0)          load the new frequency
WAIT1         move.w (A0),D0          the program will wait in WAIT1
              bmi.w WAIT1             or WAIT2 during the period when the
WAIT2         move.w (A0),D0          PLL loses then reacquires lock
              bpl.w WAIT2
* at this point, the PLL will have reacquired lock and SYSCLK will
* be stable at the new frequency and the program can continue
```

Normally, the master frequency will be changed only during the bootup sequence. While it is possible to dynamically control the master frequency, it is recommended that the frequency be set to its permanent value at bootup (or use the default).

3.3.5 PLL Shutdown

The procedure for PLL shut down to place the system in sleep mode is similar to changes made to the frequency. The difference is that the system can be awakened only by an interrupt or reset. While there are different approaches, the simplest is to synchronize the software to the rising edge of CLK32, write the disable bit, then execute a STOP instruction. The CPU no longer fetches instructions then waits for the clock to stop. When an interrupt awakens the system after the PLL acquires lock, the CPU executes an interrupt-service routine for the level of the pending interrupt. After the interrupt-service routine, the CPU begins execution at the instruction after the STOP instruction. The instruction sequence below illustrates the flow. It is assumed that all peripherals and the LCD controller have been shut down before the PLL stops.

```
              lea #$FFF202, AO        point to the Freq Sel Register
WAIT          move.w (A0),D0          synchronize to rising CLK32 edge
              bpl.w WAIT              wait for CLK32 to go high
              bset #3,(A0)            Disable the PLL
              stop #$2000             stop fetching and wait for any IRQ

* The system waits here for the PLL to restart after a wakeup IRQ
* After the IRQ routine, the instruction flow continues from here

              JMP STARTUP             jump to housekeeping routine
```

3.4 POWER CONTROL MODULE OVERVIEW

The power control module improves power efficiency as it allocates power (clocks) to the CPU core and other modules in the MC68328 processor under software control. Clocks can be enabled in bursts. While executing tasks that require significant CPU resources, the clock can be enabled for extended periods of time. While the CPU is relatively idle, the clock can be disabled or bursted with a low duty cycle. When a wakeup interrupt occurs, the clock is immediately enabled, allowing the CPU to service the request. The DMA controller is not affected by the power controller. It has full access to the bus while the CPU is idle, keeping the screen refreshed. The following sections describe the use and operation of the power control block.

3.4.1 Description

Figure 3-4 is a block diagram of the power control module. Following reset, the power controller is disabled and the MC68EC000 clock is continuously on. When the block is enabled, software controls the clock burst width in increments of 1/31. Initially, the duty cycle is set to 100%. Software can then change the duty cycle to a lower value and the clock begins to burst. In normal operation, the MC68EC000 does not have to operate continuously. Usually, it waits for user input. An interrupt from the keyboard, for example, disables the power controller, and the clock again becomes continuous. When the software completes its service of the task, the power controller can again be enabled to burst the clock and reduce power consumption. Clock control is in increments of approximately 3% (1/31).

When the burst-width control sub-block indicates that the CPU clock's time slot has expired and is to be disabled, clock control requests the bus from the CPU. After the bus is granted, the clock stops. Bus grant to the DMA controller is asserted and the DMA controller has complete access to the bus.

If a wakeup interrupt event occurs while the CPU clock is disabled, the clock is immediately enabled and the CPU processes the interrupt. The DMA controller always has priority, so if

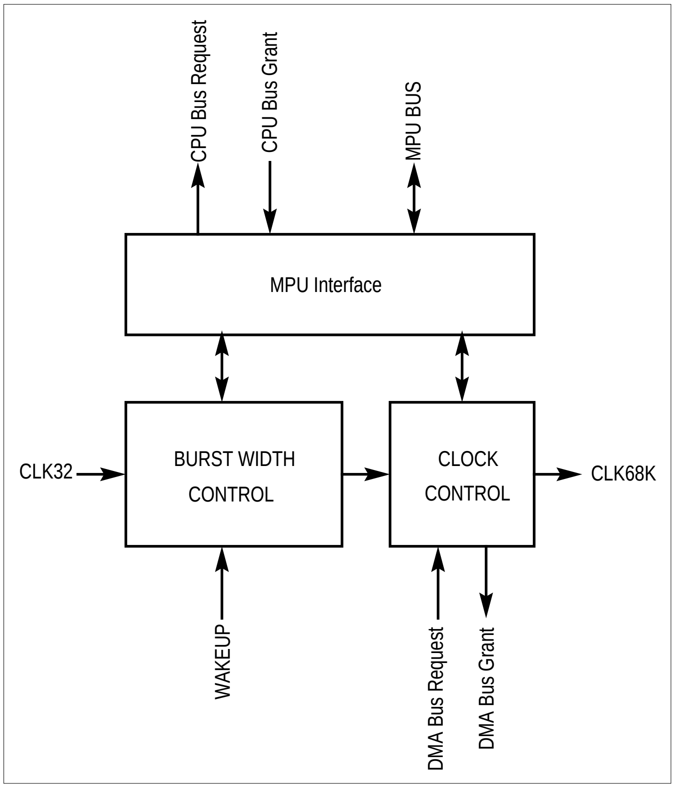


Figure 3-4. Power Control Module

a DMA access is in progress, the CPU will wait until the DMA controller has completed its access before interrupt processing begins.

Figure 3-5 describes the power controller operation. In this example, the clock bursts at about 15% duty cycle, so the MC68EC000 is active about 15% of the time. The remainder of the time, the MC68EC000 is in sleep mode. When a wakeup event occurs, the clock immediately restarts so the processor can service the wakeup event interrupt. The power-controller burst period is 31 CLK32 periods, or approximately 1 msec. 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.

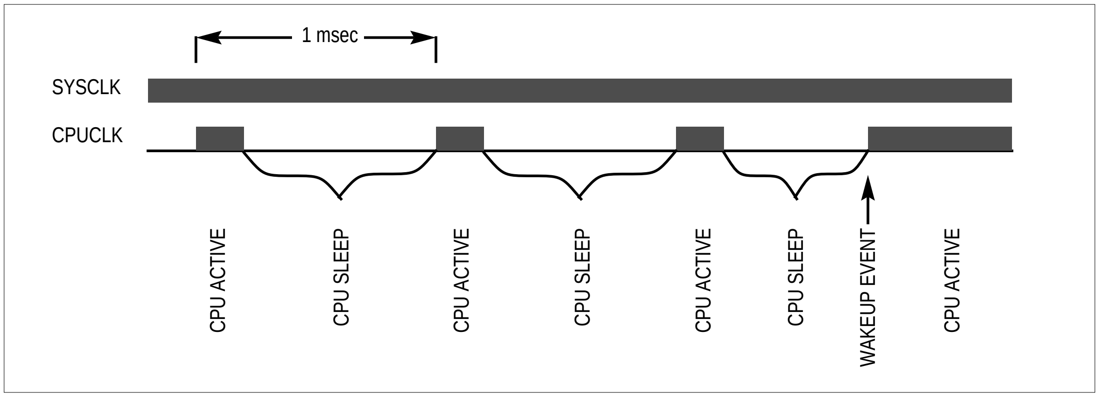


Figure 3-5. Power Control Operation

3.4.2 MPU Interface

One register is associated with the power control block. Figure 3-6 illustrates the bits in the register.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								PC EN	STOP	0	WIDTH				

Address: $(FF)FFF206 Reset Value: $001F

Figure 3-6. Power Control Register

PC EN

1 = Power-control enabled
0 = Power-control disabled

This bit controls the operation of the power controller. While this bit is low, the clock to the MC68EC000 is continuously on. While this bit is high, the clock is bursted to the MC68EC000 under control of the width comparator. An interrupt that can wake up the MC68EC000 disables the power controller by a negation of this bit. The user’s interrupt-service routine must re-enable this bit to re-enter power-save operation. This bit resets to zero.

STOP

1 = Stop CPU clock
0 = Normal CPU clock bursts

This bit immediately enters the power-save mode without waiting for the power controller to cycle through a complete burst period. This bit disables the CPU clock after the bus cycle that follows the next CLK32 rising edge. When the system is to enter the doze mode, this bit is set. On the next burst period, or interrupt, the clock will restart for its allotted period. This bit is reset to zero.

WIDTH

Width of CPU clock bursts. These bits reset to 11111 ($1F).

00000 = 0/31 duty cycle
00001 = 1/31 duty cycle
00010 = 2/31 duty cycle
...
11111 = 31/31 duty cycle

These bits control the width of the CPU clock bursts in 1/31 increments. While the WIDTH is 1 and the power controller is enabled, the clock is bursted to the CPU at a duty cycle of 1/31. While the WIDTH bits are 1F(hex), the clock is always on. While the WIDTH is zero, the clock is always off. Set the WIDTH to 0 when the CPU should be disabled for extended time periods, but it can be awakened without waiting for the PLL to re-acquire lock.

These bits are not affected by the PC EN bit. When an interrupt disables the power controller, these bits are not changed. Users should re-enable the power controller that services the interrupt.

3.4.3 Operation

This section describes how to use the power controller.

3.4.3.1 NORMAL OPERATION. When the MC68328 processor begins operation after reset, the power controller is disabled and the MC68EC000 clock runs continuously. To reduce the power consumed by the MC68EC000, the power controller is enabled when the PC EN bit is set. The value in the WIDTH register determines the duty cycle of the clock that is applied to the MC68EC000. If an interrupt is received, the power controller is automatically disabled. It is up to the interrupt-service routine to re-enable the power controller.

3.4.3.2 DOZE OPERATION. The MC68EC000 clock can be disabled for extended periods by setting the WIDTH register to 00000. The MC68EC000 clock is enabled when it receives an interrupt. At the end of the service routine, the power controller can be re-enabled, putting the MC68EC000 back into DOZE mode. Once the MC68EC000 clock is disabled, only an interrupt or hardware reset can re-enable it. For various MC68EC000 resource requirements, users can program the duty-cycle register for burst-duty cycles of any value between 0/31 and 31/31. This effectively provides a variable clock frequency (and power dissipation) of between 0% and 100% of the system clock frequency in 3% incremental steps.

3.4.3.3 SLEEP OPERATION. The PLL is disabled in the SLEEP mode. Only the 32 kHz clock continuously operates to keep the real-time clock operational. Wakeup events can activate the PLL and the system clock will begin to operate within 2 msec.

SECTION 4
LCD CONTROLLER MODULE

The liquid crystal display controller (LCDC) provides display data for an external LCD driver or LCD panel module. The key features include the following:

- Share system and display memory, no dedicated video memory required
- Standard panel interface for common LCD drivers
- Supports single (non-split) screen monochrome LCD panels
- Fast flyby type, 16-bit wide, burst-DMA screen refresh transfers from system memory
- Maximum display size is 1024x512; however, the typical non-split panel sizes are 320x240 and 640x200
- Panel interface: 1-, 2-, or 4-bit wide LCD data bus
- Black and white, or 4 simultaneous gray levels out of a palette of 7
- Hardware blinking cursor; programmable up to 32 x 32 pixels in size
- Hardware panning (soft horizontal scrolling)

The LCDC fetches display data directly from system memory through periodic DMA transfer cycles. The bus bandwidth used by the LCDC is low, thereby enabling the MC68EC000 core to have sufficient computing bandwidth for other tasks.

4.1 LCDC SYSTEM OVERVIEW

The LCDC is built of six basic blocks, namely MPU interface registers, screen DMA controller, line buffer, cursor logic, frame-rate control and LCD panel interface as shown in Figure 4-1.

4.1.1 MPU Interface

The MPU interface consists of all control registers that enable all different features of the LCDC. This block is connected directly to the 68K bus.

4.1.2 Direct Memory Access (DMA)

The DMA generates a bus-request signal to the MC68EC000 periodically and, upon receiving a bus grant, performs a 16- or 8-word memory burst to fill the line buffer. The number of DMA clock cycles per transfer is programmable (1, 2, 3, or 4 clocks/transfer), which makes it more versatile to support systems with memory of different speeds.

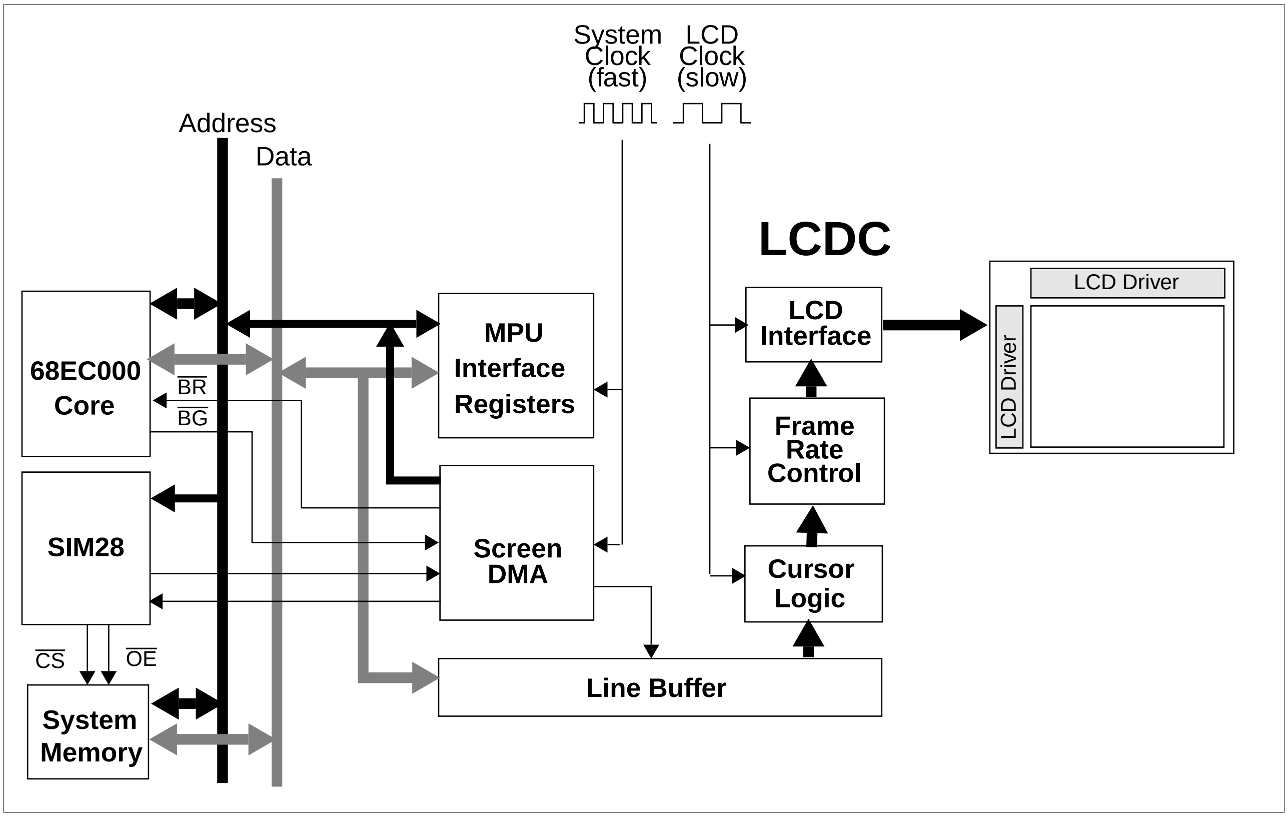


Figure 4-1. System Block Diagram of LCDC

4.1.3 Line Buffer

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.

4.1.4 Cursor Control Logic

The cursor control logic (when enabled) generates a block-shaped cursor on the display screen. Users can adjust the cursor height and width to any number between 1 to 31. The cursor can be full black or reversed video, and the blinking rate is adjustable when the blink-enable bit is on.

4.1.5 Frame Rate Control (FRC)

The frame rate control (FRC) is used primarily for gray-scale display and can generate up to 4 gray levels from the choice of 7 density levels (0, 1/4, 5/16, 1/2, 11/16, 3/4, 1 as in Table 4-3). The density level corresponds to the number of times the pixel is being turned on when the display is refreshed frame by frame. Because the crystal formulations and driving voltage may vary, the visual gray quality can be tuned by programming the gray palette-mapping register (GPMR) to obtain the best effect.

Because blinking or flickering will occur if all LCD pixel cells are synchronized, it is essential to program two 4-bit numbers, namely Xoff and Yoff in the FRC offset register (FOSR), to minimize flickering. As a general rule, select odd numbers that differ by 2. The optimal offset values could vary among different models of LCD panels—even from the same manufacturer—because of different inter-pixel crosstalk characteristics.

4.1.6 LCD Interface

The LCD interface logic packs the display data in the correct size and outputs it to the LCD panel data bus. The polarity of FRM, LP, and SCLK signals as well as pixel data can all be programmable to suit different types of LCD panel requirements.

4.2 INTERFACING LCDC WITH LCD PANEL

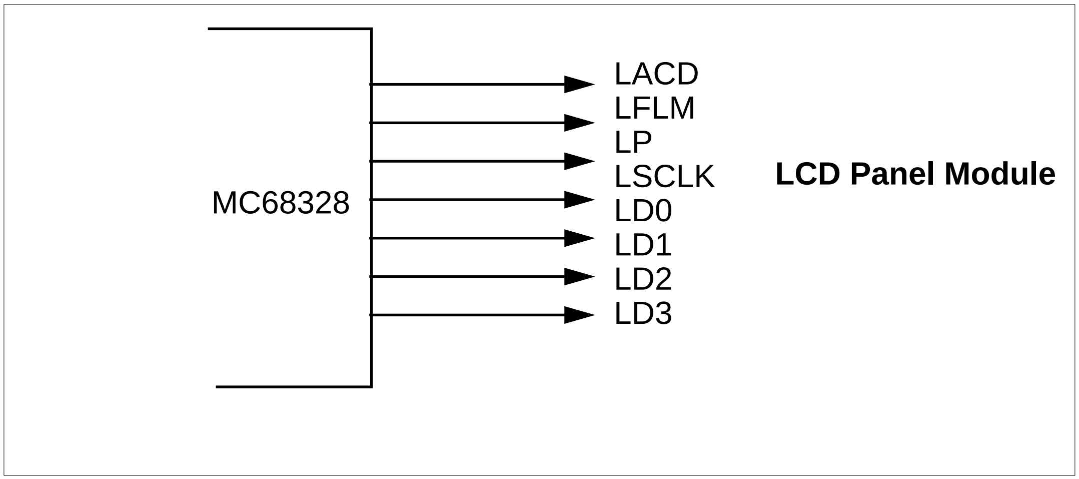


Figure 4-2. LCD Module Interface Signals

LCD Data Bus (LD3-LD0)

This output bus transfers pixel data to the LCD panel for display. Depending on which LCD panel mode was selected, data is arranged differently on the bus for each mode. Users can program the output pixel data to be negated. See the POLCF register description for details.

First Line Marker (LFLM)

This signal indicates the start of a new display frame. The LFLM signal becomes active after the first line pulse of the frame and remains active until the next line pulse, at which point it de-asserts and remains inactive until the next frame. Users can program the LFLM signal using software to be active-high or active-low. See the POLCF register description for details.

Line Pulse (LP)

This signal latches a line of shifted data onto the LCD panel. It becomes active when a line of pixel data is clocked into LCD panels and stays asserted for a duration of 8 pixel clock periods. Users can program the LP signal using software to be either active-high or active-low. See the POLCF register description for details.

Shift Clock (LSCLK)

This is the clock output that is synchronized to the LCD panel output data. Users can program the LSCLK signal using software to be either active-high or active-low. See the POLCF register description for details.

Alternate Crystal Direction (LACD)

This output is toggled to alternate the crystal polarization on the panel. Users can program this signal to toggle at a period of 1 to 16 frames. The alternate crystal direction (LACD, also called M) pin will toggle after a pre-programmed number of FLM pulses. Users can program the ACD rate-control register (ACDRC) so that LACD will toggle once every 1 to 16 frames. The targeted number of frames is equal to the alternation code's 4-bit value plus one. The default value for ACDRC is zero; that is, LACD will toggle on every frame. The LACD output signal is synchronized with the trailing (falling) edge of the line pulse (LP) enclosed by FLM.

Table 4-1. ACDRC Value and Number of Cycles

ACDRC	No of Cycles
0000	1
0001	2
0010	3
0100	5
1000	9
1111	16

4.3 PANEL I/F TIMING

The LCDC signal continuously pumps the pixel data into the LCD panel via the LCD data bus. The bus is timed by shift clock (LSCLK), line pulse (LLP) and first line marker (LFLM). The LSCLK clocks the pixel data into the display drivers' internal-shift register. The LP latches the shifted pixel data into a wide latch at the end of a line while the LFLM marks the first line of the displayed page.

The LCDC signal is designed for great flexibility to support most of the monochrome LCD panels available in the marketplace. Figure 4-3 shows the LCD interface timing for 8-bit LCD data-bus operations.

Figure 4-3 shows the LCD interface timing for 4-, 2-, and 1-bit LCD data-bus operations.

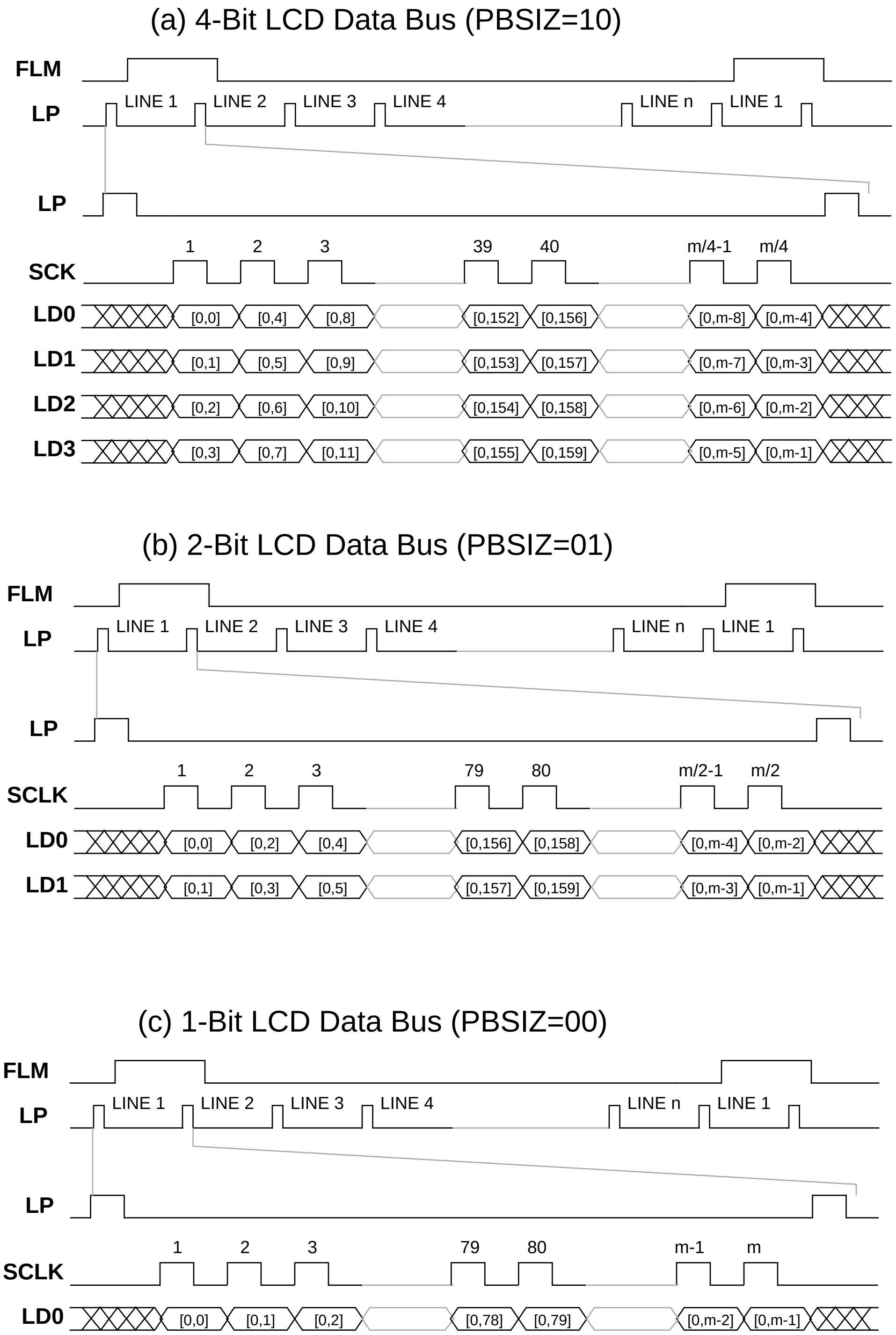
The line pulse signifies the end of the current line of serial data. The LLP enclosed by LFLM signal marks the end of the first line of the current frame.

Some LCD panels may use an active-low LFLM signal, LLP signal, LSCLK signal, and reversed pixel data. To change the polarities of these signals, set the first-line marker polarity (FLMPOL), line-pulse polarity (LPPOL), shift-clock polarity (SCLKPOL), and pixel polarity

(PIXPOL) bits to 1, respectively. The LLP and LFLM timing are similar for all panel modes supported by LCDC.

In additional to the interface timing pins discussed above, an alternate crystal direction (LACD) pin in LCDC will toggle after a pre-programmed number of LFLM pulses. This pin prevents crystal degradation in the LCD panel.

.


(a) 4-Bit LCD Data Bus (PBSIZ=10)

(b) 2-Bit LCD Data Bus (PBSIZ=01)

(c) 1-Bit LCD Data Bus (PBSIZ=00)

Figure 4-3. LCD Interface Timing for 4-, 2-, and 1-Bit Data Widths

4.4 OPERATION OVERVIEW

4.5 DISPLAY CONTROL

The LCDC signal drives single-screen monochrome STN LCD panels with up to 1024x512 pixels in the gray-scale mode at a refresh rate of 60-70 Hz. In any case, the best efficiency is achieved when the screen width is a multiple of the DMA controller’s 16-bit bus width. Because of LCD driver-technology limitations, large screens, such as 640x480, are usually organized in spilt-screen format, which the MC68328 processor does not support. The actual limit is the number of rows that require high driving voltage. The MC68328 processor 4-bit LCD interface will drive up to 240 rows with a maximum of 1024 columns.

4.5.1 LCD Screen Format

The screen width and height of the LCD panel are software-programmable. Figure 4-4 shows the relationship between the portion of a large graphics file displayed on screen vs. the actual page. All units are measured in pixel counts in this figure.

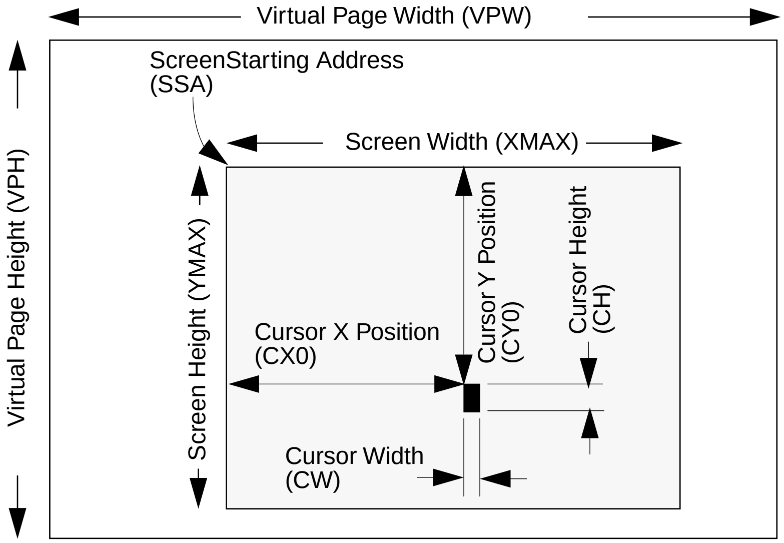


Figure 4-4. LCD Screen Format

The screen width (XMAX) and screen height (YMAX) registers specify the LCD panel size. The LCD will start scanning the display memory at the location pointed to by the screen starting address (SSA) register. Therefore, the LCD panel will display the shaded area in Figure 4-4.

The virtual page width (VPW) and virtual page height (VPH) parameters specify the maximum page width and height, respectively. By changing the screen-starting address (SSA) register, a screen-sized window can be vertically or horizontally scrolled (panned) anywhere inside the virtual-page boundaries. The software must position the starting address (SSA) properly so that the scanning logic’s system memory pointer (SMP) does not stretch beyond VPW nor VPH. Otherwise, strange artifacts will display on the screen. The programmer uses the VPH only for boundary checks. There is no VPH register internal to the LCDC.

4.5.2 Cursor Control Logic

To define the position of the hardware cursor, the LCDC maintains a vertical line counter (YCNT) to keep track of the pixel's current vertical position. YCNT in conjunction with the horizontal pixel counter (XCNT) specify the screen position of the current pixel data being processed. When the pixel falls within a window specified by the cursor reference position (CXP:10-bit register, CYP: 9-bit register), cursor width (CW:5-bit counter), and cursor height (CH:5-bit counter), the original pixel bits (outputs of HSRA and B are affected but the latter's output is ignored, if not applicable) wil be passed transparently (cursor control bits=00), replaced with full black, or a complement for reversed video (CC bits=01,10 respectively; 11 not allowed) if a static cursor is chosen (BK_EN=0). Reversed video is preferable for a static cursor as it will block the original pixels if CC=01. A blinking cursor will display if BK_EN=1, in which case the original signal and the cursor will alternate periodically.

4.5.3 Display Data Mapping

The LCDC supports 1-bit-per-pixel or 2-bits-per-pixel graphics mode. In the binary mode (GS=0), 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-per-pixel operations (GS=1), the frame-rate control circuitry inside the LCDC will generate intermediate gray tones on the LCD panel by adjusting the densities of 1's and 0's over many frames. A maximum of 4 gray levels can be simultaneously displayed on the LCD screen.

The system memory data in both 1- and 2- bit-per-pixel modes are mapped as shown in Figure 4-5.

4.5.4 Gray Scale Generation

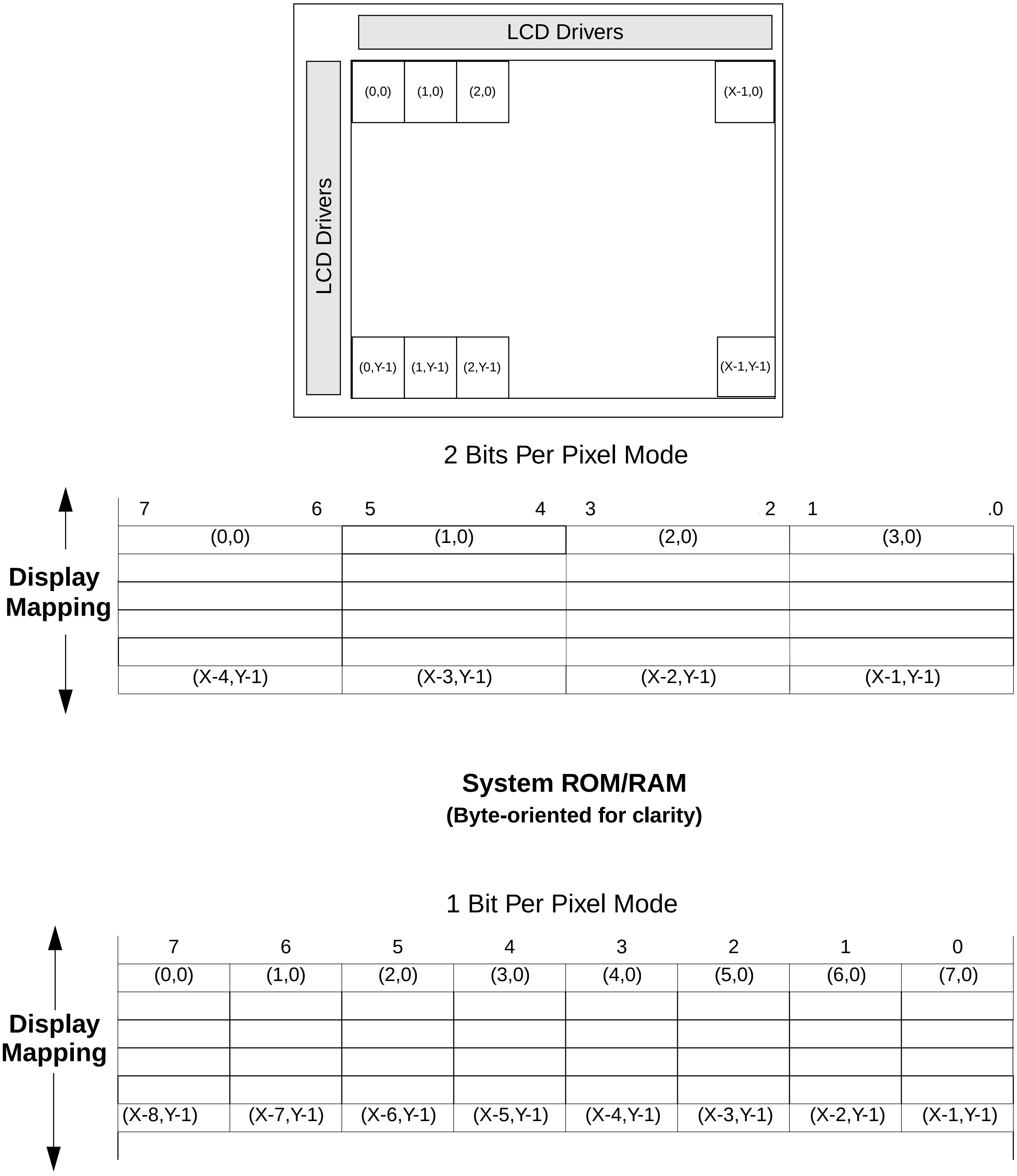
The LCDC is configured to drive only a single-screen monochrome LCD panel. It cannot handle color STN or TFT panels. Users can configure the data bus size for the LCD panel to 1-bit, 2-bit, or 4-bit by programming the LCD panel bus size (PBSIZ) register.

4.5.5 Gray Palette Mapping

Through a proprietary frame-rate control (FRC) algorithm, the LCDC can generate up to 4 simultaneous gray levels out of 7 available by first mapping the 2-bit data into four 3-bit gray codes which then select 4 out of 7 bit densities from the gray palette table.

Figure 4-5 shows the mapping of the 2-bit pixel data into 3-bit gray codes. Bits GMN are defined in the software-programmable gray palette mapping registers (GPMR). Each of the four 3-bit codes obtained from the first table then selects a density level (0, 1/4, 5/16, 1/2, 11/16, 3/4 and 1) from the gray palette table as shown in Table 4-3.

Because crystal formulations and driving voltages vary, the visual gray effect may or may not have a linear relationship to the frame rate. A logarithmic scale such as 0, 1/4, 1/2, and 1 might be more pleasing than a linear-spaced scale such as 0, 5/16, 11/16, and 1 for certain graphics. A flexible mapping scheme lets users optimize the visual effect for the specific panel or application.



2 Bits Per Pixel Mode

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

System ROM/RAM
(Byte-oriented for clarity)

1 Bit Per Pixel Mode

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

System ROM/RAM
(Byte-oriented for clarity)

Figure 4-5. Mapping of Memory Data on the Screen

Table 4-2. Gray Scale Code Mapping

Code Mapping	
Data	Gray code
00	G02G01G00
01	G12G11G10
10	G22G21G20
11	G32G31G30

Table 4-3. Gray Palette Selection

Gray Palette	
Gray Code	Density
000	0
001	1/4
010	5/16
011	1/2
100	11/16
101	3/4
110	1
111	1

4.5.6 FRC Offset Control

4.5.7 Cursor and Blinking Rate Control

4.5.8 Low-Power Mode

Some panels may have a signal called PANEL_OFF that turns off the panel for low-power mode. In the MC68328 processor system, this signal is not supported. Instead, use a parallel I/O pin to perform this function.

The software sequence to achieve PANEL_OFF using parallel I/O consists of 2 steps:

1. Turn off the VLCD (+15V or -15V) by I/O driving a transistor
2. Turn off the LCDON bit

To exit from LCDC-off mode:

1. Turn on the LCDON bit
2. Delay for 1-2ms
3. Turn on the VLCD by I/O driving a transistor

When setting the LCDON bit (register CKCON bit 7) to 1, LCDC itself will enter a low-power mode by stopping its own pixel clock prior to the next line-buffer-fill DMA. Additional screen DMA and display-refresh operations will then be stopped in this mode. When the LCDC is switched back on, DMA and screen-refresh activities will resume in a synchronous fashion. Software should check that the actual PANEL_OFF signal is de-asserted before setting LCDON to a 1.

4.6 DMA CONTROLLER OVERVIEW

The LCD DMA controller is a flyby type, 16-bit wide, fast-data transfer machine. Because the LCD screen has to be refreshed continuously at a rate of about 50-70 Hz, in this case, the pixel bits in the memory will be read and transferred to corresponding pixels on the screen. To minimize the bus obstruction because of bus-sharing with the system, a burst type and flyby transfer is therefore required. The refresh is divided into small packs of 8- or 16-word reads. Every time the internal line buffer needs data, it will assert the $\overline{\text{BR}}$ signal to request the bus from the MC68EC000. Once the MC68EC000 core grants the bus (i.e. $\overline{\text{BG}}$ is asserted), the DMA controller will get control of the bus signal and issue 8- or 16-word reads (see setting of CKCON register) from memory. The read data is then passed to the next stage internally to generate the LCD timing (flyby). During the LCD access cycles, output- enable and chip-select signals for the corresponding system SRAM chip will be asserted by the chip-select logic inside the SIM. The minimum bus bandwidth obstruct can be achieved by using zero LCD-access wait states (1 clock per access). See Section 4-8 Bandwidth Calculation and Saving for more details.

4.6.1 Basic Operation

As shown in Figure 4-6 and Figure 4-7, data is fetched from memory in a very efficient manner. Each burst is limited to 8/16 words, which reduces possible latency for other peripherals such as the interrupt controller. For example, the average time latency for LCDCLK = 5MHz with 16-word burst is approximately 2.4µs.

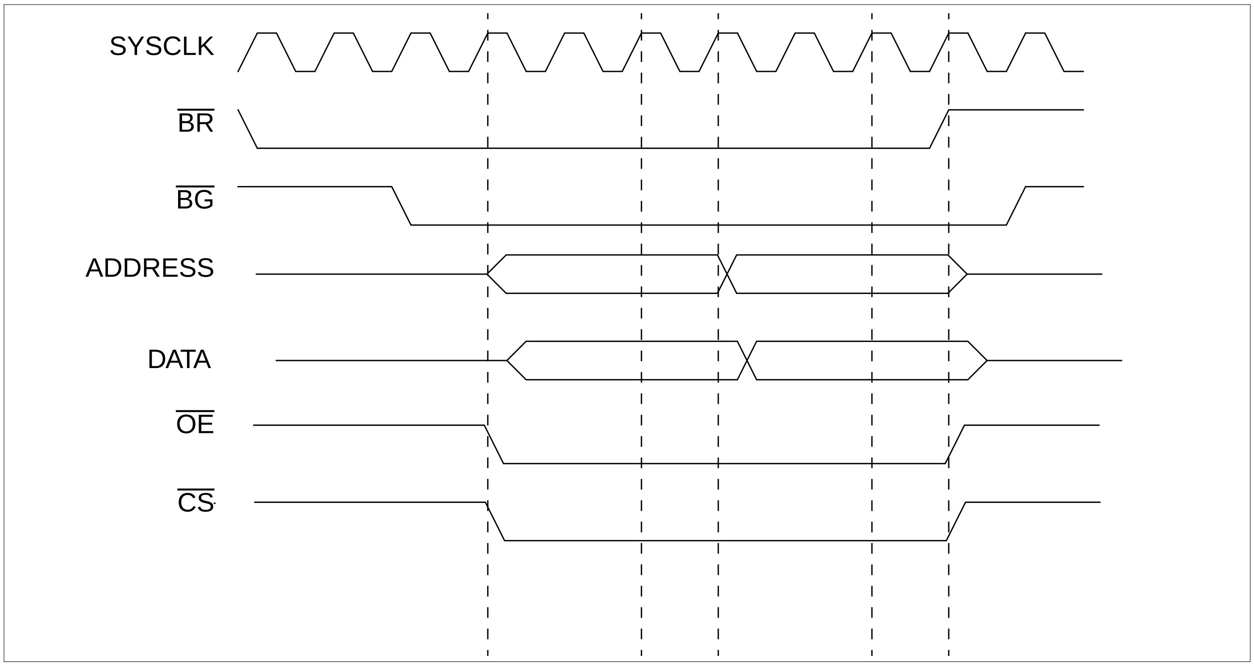


Figure 4-6. Three Clock per LCD DMA Transfer (2 Wait States)

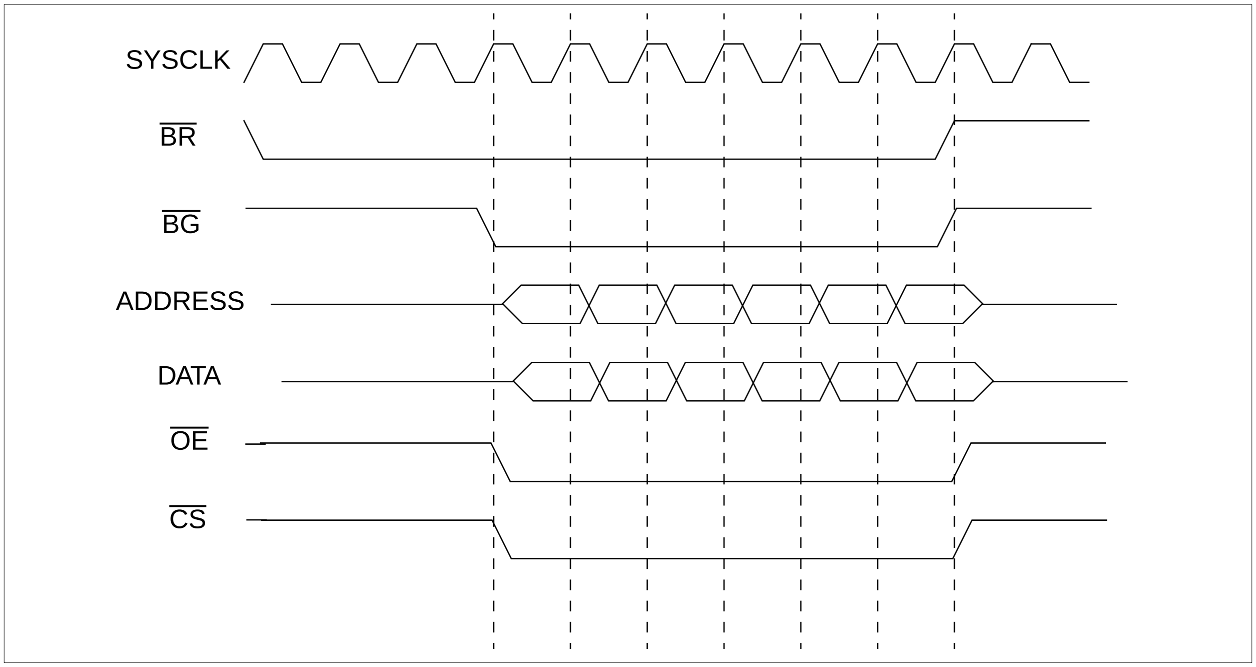


Figure 4-7. One Clock per DMA Transfer (0 Wait State)

4.7 REGISTER DESCRIPTIONS

4.7.1 System Memory Control Registers

4.7.1.1 SCREEN STARTING ADDRESS REGISTER (SSA).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SSA31	SSA30	SSA29	SSA28	SSA27	SSA26	SSA25	SSA24	SSA23	SSA22	SSA21	SSA20	SSA19	SSA18	SSA17	SSA16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSA15	SSA14	SSA13	SSA12	SSA11	SSA10	SSA9	SSA8	SSA7	SSA6	SSA5	SSA4	SSA3	SSA2	SSA1	0

Address: $(FF)FFFA00 Reset Value: $00000000

Figure 4-8. Screen Starting Address Register

SSA31-SSA1 Screen-Starting Address Register

32-bit screen-starting address of the LCD panel (see Figure 4-8). The LCDC fetches pixel data from system memory at this address.

4.7.1.2 VIRTUAL PAGE WIDTH REGISTER (VPW).

7	6	5	4	3	2	1	0
VP8	VP7	VP6	VP5	VP4	VP3	VP2	VP1

Address: $(FF)FFFA05 Reset Value: $FF

Figure 4-9. Virtual Page Width Register

VP8-VP0M Virtual Page Width Register

This register (see Figure 4-9) specifies the virtual page width of the LCD panel in terms of byte count. VP0 defaults to zero because of the 16-bit transfers.

VPW = virtual page width in pixels divided by c where c is 16 for black-and-white display and 8 for gray level.

4.7.2 Screen Format Registers

4.7.2.1 SCREEN WIDTH REGISTER (XMAX).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED						XM9	XM8	XM7	XM6	XM5	XM4	XM3	XM2	XM1	XM0

Address: $(FF)FFFA08 Reset Value: $03FF

Figure 4-10. Screen Width Register XMAX

XM9-XM0

Pixels on a line are numbered 0 to XMAX for a screen width of XMAX +1 pixels. XMAX+1 must be a multiple of 16 (see Figure 4-10).

4.7.2.2 SCREEN HEIGHT REGISTER (YMAX).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED						YM9	YM8	YM7	YM6	YM5	YM4	YM3	YM2	YM1	YM0

Address: $(FF)FFFA0A Reset Value: $01FF

Figure 4-11. Screen Height Register YMAX

YM8-YM0

This register (Figure 4-11) specifies the LCD panel height in term of pixels or lines. The lines are numbered from 0 to YMAX for a total of YMAX + 1 lines, which is equal to the screen height in pixel count.

4.7.3 Cursor Control Registers

4.7.3.1 CURSOR X POSITION REGISTER (CXP).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC1	CC0	UNUSED				CXP9	CXP8	CXP7	CXP6	CXP5	CXP4	CXP3	CXP2	CXP1	CXP0

Address: $(FF)FFFA18 Reset Value: $0000

Figure 4-12. Cursor X Position Register

CC1-CC0

Cursor control bits

00= Transparent, cursor is disabled
01= Full density (black) cursor
10= Reversed video
11= Do not use

CXP9-CXP0

Cursor's horizontal starting position X in pixel count (from 0 to XMAX).

4.7.3.2 CURSOR Y POSITION REGISTER (CYP).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED							CYP8	CYP7	CYP6	CYP5	CYP4	CYP3	CYP2	CYP1	CYP0

Address: $(FF)FFFA1A Reset Value: $0000

Figure 4-13. Cursor Y Position Register

CYP8-CYP0

Cursor's vertical starting position Y in pixel count (from 0 to YMAX).

4.7.3.3 CURSOR WIDTH & HEIGHT REGISTER (CWCH).

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

Address: $(FF)FFFA1C Reset Value: $0101

Figure 4-14. Cursor Width & Height Register

CW4-CW0

Cursor width. This 5-bit group specifies the width of the hardware cursor in pixel count (from 1 to 31).

CH4-CH0

Cursor height. This 5-bit group specifies the height of the hardware cursor in pixel count (from 1 to 31).

NOTE

The cursor is disabled if either CW or CH are set to zero.

4.7.3.4 BLINK CONTROL REGISTER (BLKC).

7	6	5	4	3	2	1	0
BKEN	BD6	BD5	BD4	BD3	BD2	BD1	BD0

Address: $(FF)FFFA1F Reset Value: $7F

Figure 4-15. Blink Control Register

BKEN

Blink-enable cursor will remain on instead of blinking if this bit is cleared. Defaults to zero.

1 = Blink enable
0 = Blink disable

BD6-BD0

Blink divisor. The cursor will toggle once per specified number of internal frame pulses plus one. The half-period may be as long as 2 seconds.

4.7.4 LCD Panel Interface Registers

4.7.4.1 PANEL INTERFACE CONFIGURATION REGISTER (PICF).

7	6	5	4	3	2	1	0
UNUSED					PBSIZ1	PBSIZ0	GS

Address: $(FF)FFFA20 Reset Value: $00

Figure 4-16. Panel Interface Configuration Register

PBSIZ1-PBSIZ0 Panel Bus Width

LCD panel bus size.

00 = 1-bit
01 = 2-bit
10 = 4-bit
11 = unused

GS Gray Scale

Gray scale mode bit. This bit, if set, enables 4-gray level (2 bits per pixel) mode. Its default value is 0, which selects binary pixel (no gray scale) operation.

1 = Gray scale enable
2 = No gray scale

4.7.4.2 POLARITY CONFIGURATION REGISTER (POLCF).

7	6	5	4	3	2	1	0
UNUSED				LCKPOL	FLMPOL	LPPOL	PIXPOL

Address: $(FF)FFFA21 Reset Value: $00

Figure 4-17. Polarity Configuration Register

LCKPOL LCD Shift Clock Polarity

This bit controls the polarity of the LCD shift-clock active edge.

0 = Active negative edge of LCLK
1 = Active positive edge of LCLK

FLMPOL

First-line marker polarity

0 = Active High
1 = Active Low

LPPOL

Line-pulse polarity

0 = Active-high
1 = Active-low

PIXPOL

Pixel polarity

0 = Active-high
1 = Active-low

4.7.4.3 LACD (M) RATE CONTROL REGISTER (ACDRC).

7	6	5	4	3	2	1	0
UNUSED				ACD3	ACD2	ACD1	ACD0

Address: $(FF)FFFA23 Reset Value: $00

Figure 4-18. LACD Rate Control Register

ACD3-ACD0 Alternate Crystal Direction Control

ACD toggle-rate control code. The ACD signal will toggle once every 1 to 16 FLM cycles based on the value specified in ACDRC register. The actual number of FLM cycles is the value programmed plus one. Shorter cycles tend to give better results.

4.7.5 Line Buffer Control Registers

4.7.5.1 PIXEL CLOCK DIVIDER REGISTER (PXCD).

7	6	5	4	3	2	1	0
UNUSED		PCD5	PCD4	PCD3	PCD2	PCD1	PCD0

Address: $(FF)FFFA25 Reset Value: $00

Figure 4-19. Pixel Clock Divider Register

PCD5-PCD0 Pixel Clock Divider

The PIX clock from the PLL is divided by N (PCD5-0 plus one) to yield the actual pixel clock. Values of 1-63 will yield N=2 to 64. If set to 0 (N=1), the PIX clock will be used directly, bypassing the divider circuit. Input source is selected by PCDS in CKCON register.

4.7.5.2 CLOCKING CONTROL REGISTER (CKCON).

7	6	5	4	3	2	1	0
LCDON	DMA16	WS1	WS0	UNUSED		DWIDTH	PCDS

Address: $(FF)FFFA27 Reset Value: $00

Figure 4-20. Clocking Control Register

LCDCON

This bit controls the LCDC block.

0 = Disable LCDC
1 = Enable LCDC

NOTE

The internal LCDC logic will be switched off in step with the FLM pulse.

DMA16

This bit controls the length of the DMA burst.

0 = 8 words burst length
1 = 16 words burst length

WS1-WS0 DMA Bursting Clock Control

Number of clock cycles per DMA word access

00 = Single clock-cycle transfer
01 = Two clock-cycle transfer
10 = Three clock-cycle transfer
11 = Four clock-cycle transfer

DWIDTH

Displays memory-data width indicating the size of the external bus interface.

0 = 16-bits memory
1 = 8-bits memory

PCDS Pixel Clock Divider Source Select

0 = The SYS CLK output of PLL is selected
1 = The PIX CLK output of PLL is selected

4.7.5.3 LAST BUFFER ADDRESS REGISTER (LBAR).

7	6	5	4	3	2	1	0
UNUSED	LBAR7	LBAR6	LBAR5	LBAR4	LBAR3	LBAR2	LBAR1

Address: $(FF)FFFA29 Reset Value: $3E

Figure 4-21. Last Buffer Address Register

LBA7-LBA1

The number of memory words required to fill one line on the display panel. The count is typically equal to the screen width in pixels divided by 16 for black-and-white display, or by 8 if in gray scale. For panning, add one more count for black-and-white and two for gray display.

4.7.5.4 OCTET TERMINAL COUNT REGISTER(OTCR).

7	6	5	4	3	2	1	0
OTC8	OTC7	OCT6	OCT5	OTC4	OTC3	OTC2	OTC1

Address: $(FF)FFFA2B Reset Value: $3F

Figure 4-22. Octet Terminal Count Register

OTC8-OTC1

Controls the time interval between two lines; therefore, the frame refresh rate can also be finely adjusted. The register value must be greater than LBAR by 4 for black-and-white display and 8 for gray display.

4.7.5.5 PANNING OFFSET REGISTER (POSR).

7	6	5	4	3	2	1	0
UNUSED				BOS	POS2	POS1	POS0

Address: $(FF)FFFA2D Reset Value: $00

Figure 4-23. Panning Offset Register

BOS Byte Offset

BOS is used primarily in the non-gray scale mode and in conjunction with POS0-2. (BOS must be set to zero for gray-scale data).

0 = Start from the first byte when retrieving binary pixel data for display
1 = Active display will start from the second byte instead

NOTE

The cursor reference position must be adjusted separately with software when this register is changed.

POS2-POS0 Pixel Offset Code

POS specifies which of the 8 pixels in the first or second (GS=0, BOS=1 only) octet retrieved from the line buffer is the first to be displayed on the screen. (e.g. 000 implies that pixel 7, the first shifted out, will be the first to be displayed on every horizontal line in the current frame).

4.7.6 Gray-Scale Control Registers

4.7.6.1 FRAME-RATE MODULATION CONTROL REGISTER (FRCM) .

7	6	5	4	3	2	1	0
XMOD3-XMOD0				YMOD3-YMOD0			

Address: $(FF)FFFA31 Reset Value: $B9

Figure 4-24. Frame-Rate Modulation Control Register

XMOD, YMOD Frame-Rate Modulation Control

These numbers modulate adjacent pixels at different time periods to avoid spatial flicker or jitter when using FRC. These values must be optimized by manually fine tuning the target LCD panel. See **Section 4.5.5 Gray Palette Mapping** for details.

4.7.6.2 GRAY PALETTE MAPPING REGISTER (GPMR).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	G12	G11	G10	0	G02	G01	G00	0	G32	G31	G30	0	G22	G21	G20

Address: $(FF)FFFA32 Reset Value: $0173

Figure 4-25. Gray Palette Mapping Register

GMN

Gray palette code (bit position n=0, 1, 2) output for pixel-input data m (0 for pixel data 00, 1=01, 2=10, 3=11). This 3-bit code will then select one of 7 bitstreams of different densities. See **Section 4.5.5 Gray Palette Mapping** for details.

4.8 BANDWIDTH CALCULATION AND SAVING

Because LCD screen refresh is a periodic task, the load LCDC puts on the host data bus becomes an important consideration to the high-performance handheld system designer.

4.8.1 Bus Overhead Considerations

The following example illustrates the issues involved in the estimation of bandwidth overhead to the data bus.

Consider a typical case scenario:

Screen size: 320 x 240 pixels

Bits per pixel: 2 bits / pixel

Screen refresh rate: 60 Hz

System clock = 16.67 MHz

Host bus size: 16 bit

DMA access cycle: 2 cycles per 16-bit word

The period, T_l,, that LCDC must update one line of the screen is,

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

At the same period, the line buffer must be filled. The duration, T_{DMA}, which the DMA cycle will take up the bus is,

$$\begin{aligned} T_{DMA} &= \frac{320\mathrm{pixels} \times 2 bitperpixel \times 2 clock}{16.67 MHz \times 16 bitbus} \\ &= 4.8\mu s \end{aligned} \tag{EQ 2}$$

Thus, the percentage of host bus time taken up by the LCDC DMA is P_{DMA},

$$\begin{aligned} P_{DMA} &= \frac{4.8\ \mu s}{69.4\ \mu s} \\ &= 6.92\% \end{aligned} \tag{EQ 3}$$

SECTION 5
REAL-TIME CLOCK MODULE

The real-time clock (RTC) module provides a current time stamp of seconds, minutes, and hours. The RTC operates on the low-frequency, 32 kHz (or 38.4 kHz) reference clock crystal.

Timer features include:

- Full clock features - seconds, minutes, hours
- Minute countdown timer with interrupt
- Programmable alarm with interrupt
- Once-per-second, once-per-minute, and once-per-day interrupts
- 32.768 kHz or 38.4 kHz operation

5.1 OPERATING CHARACTERISTICS

Figure 5-1 is a block diagram of the RTC. This section describes the RTC operation.

5.1.1 Prescaler and Counter

The prescaler divides the 32.768 kHz reference clock down to 1 pulse per second. An alternate reference frequency of 38.4 kHz is also supported. The counter portion of this device consists of 3 groups that are toggled by a 1 Hz clock. The seconds and minutes counters are 6 bits long while the hours counter is 5 bits long.

The time counters offer seconds, minutes, and hours data in 24-hour format. The prescaler stages are tapped to support several features. Periodic interrupts at 1 Hz and 1 minute are available as well as an interrupt at the midnight rollover of the hours counter.

5.1.2 Alarm

Users can set an alarm by accessing the 3 alarm fields and loading the hours, minutes and seconds for the time that the alarm is to generate an interrupt. The interrupt is enabled when the alarm bit in the interrupt-enable register is set. The alarm is actually posted when the current time matches the time in the alarm register.

5.1.3 Minute Stopwatch

The minute stopwatch performs a countdown with a resolution of one minute. It can generate an interrupt after some length of time. Example: The LCD is to turn off after 5 minutes of inactivity. The minute stopwatch is programmed for 5 minutes and enabled. At consecutive

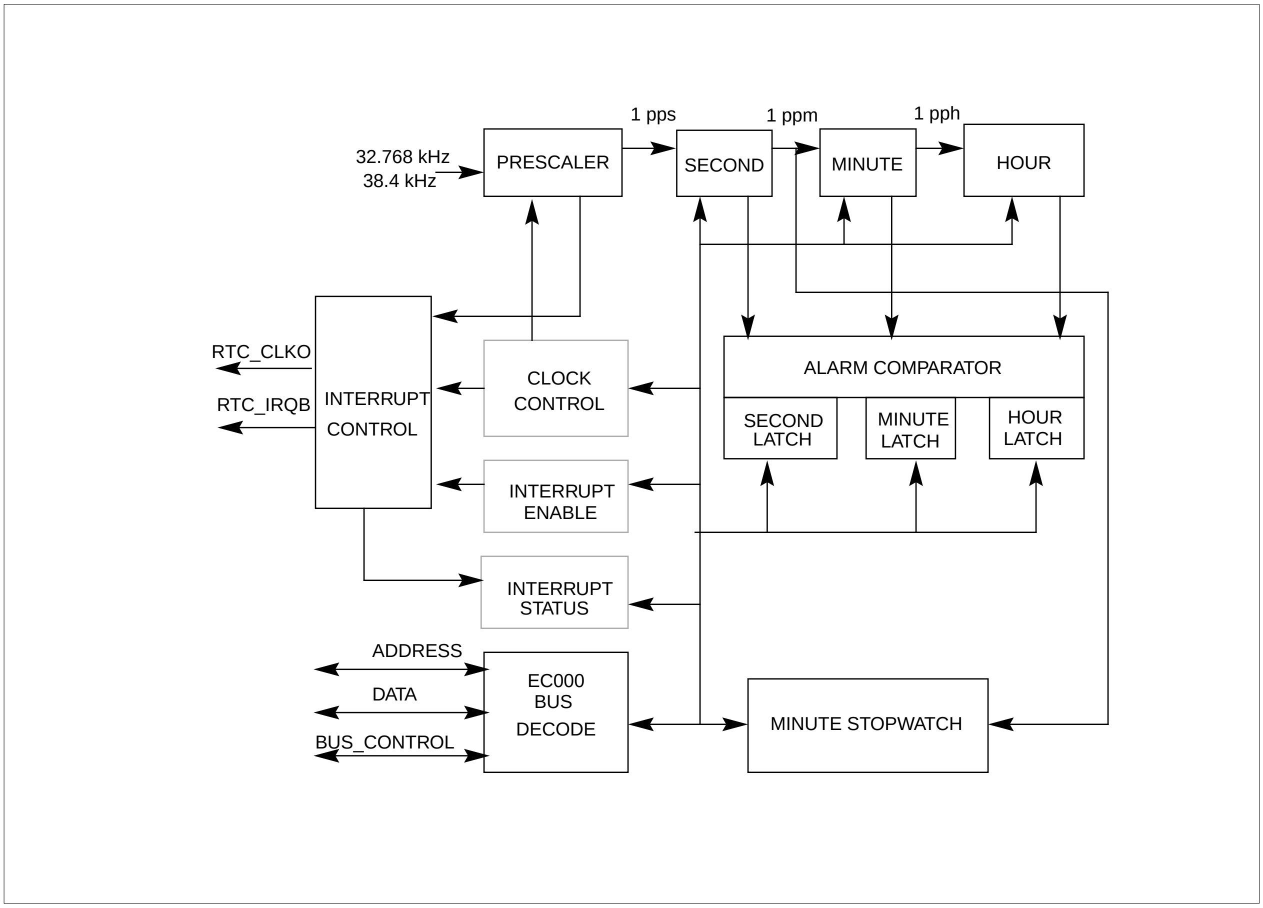


Figure 5-1. Real-Time Clock

minute increments, the minute stopwatch value is decremented. The interrupt is generated when the counter counts to -1.

5.1.4 Registers

There are several registers (described below) associated with the RTC.

5.1.4.1 RTC HOURS-MINUTES-SECONDS REGISTER (RTCHMS).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Unused			HOURS					Unused		MINUTES					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Unused										SECONDS					

Address: $(FF)FFFB00 Reset Value: $00000000

Figure 5-2. Hours-Minutes-Seconds Register

The hours, minutes and seconds can be read or written at any time. After a write, the current time assumes the new values. Unused bits read 0.

HOURS

These 5 bits, when read, indicate the current hour and can be set to any value between 0 and 23.

MINUTES

These 6 bits, when read, indicate the current minute and can be set to any value between 0 and 59.

SECONDS

These 6 bits, when read, indicate the current second and can be set to any value between 0 and 59.

5.1.4.2 ALARM REGISTER (RTCALRM).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Unused			HOURS					Unused		MINUTES					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Unused										SECONDS					

Address: $(FF)FFFB04 Reset Value: $00000000

Figure 5-3. Alarm Register

The hours, minutes, and seconds can be read or written at any time. After a write, the current time assumes the new values. Unused bits read 0.

HOURS

These 5 bits, when read, indicate the current setting of the alarm's hour and can be set to any value between 0 and 23.

MINUTES

These 6 bits, when read, indicate the current setting of the alarm's minute and can be set to any value between 0 and 59.

SECONDS

These 6 bits, when read, indicate the current setting of the alarm's second and can be set to any value between 0 and 59.

5.1.4.3 RTC CONTROL REGISTER (RTCCTL).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED								ENABLE	UNUSED	38.4	RESERVED				

ADDRESS: $(FF)FFFB0C Reset Value: $0000

Figure 5-4. Control Register

ENABLE RTC Enable

This bit enables the RTC.

1 = Enable RTC
0 = Disable RTC

38.4 38.4 kHz Reference Select

1 = Reference frequency is 38.4 kHz
0 = Reference frequency is 32.768 kHz

UNUSED

These bits are not used and read 0.

5.1.4.4 INTERRUPT STATUS REGISTER (RTCISR).

This register indicates the status of the various real-time clock interrupts. Each bit is set when its corresponding event occurs. Users must clear these bits by writing 1’s to clear the interrupt. The interrupt registers can post interrupts while the system clock is idle (sleep mode).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED											1 Hz FLAG	DAY FLAG	MIN FLAG	RVSD	SW FLAG

ADDRESS: $(FF)FFFB0E Reset Value: $0000

Figure 5-5. Interrupt Status Register

UNUSED

These unused bits read 0.

1 Hz FLAG

If enabled, this bit is set every second and an interrupt posted.

1 = 1-Hz interrupt occurred
0 = No 1-Hz interrupt occurred

DAY FLAG

If enabled, this bit is set for every 24-hour clock increment (at midnight) and an interrupt posted.

1 = 24-hour rollover interrupt occurred
0 = No 24-hour rollover interrupt

ALARM FLAG

If enabled, an alarm flag is set on a “compare” match between the RTC and the alarm register value. (Note: the alarm will recur every 24 hours. If a single alarm is required, clear the interrupt-enable in the interrupt-service routine.)

1 = Alarm interrupt occurred
0 = No alarm interrupt occurred

MIN FLAG

If enabled, a minute flag is set on every minute tick.

1 = Minute tick occurred
0 = No minute tick occurred

SW FLAG

If enabled, the stopwatch flag is set when the stopwatch minute countdown experiences a time out.

1 = Stopwatch timed out
0 = No stopwatch time out

5.1.4.5 INTERRUPT-ENABLE REGISTER (RTCIENR).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED											1HZ Enable	24 Hr Enable	Alarm Enable	MIN INT Enable	SW Enable

Address: $(FF)FFFB10 Reset Value: $0000

Figure 5-6. Interrupt-Enable Register

UNUSED

These unused bits read 0.

1-Hz INTERRUPT-ENABLE

This bit enables an interrupt at a 1 Hz rate.

1 = 1-Hz interrupt enabled
0 = 1-Hz interrupt disabled

24-HOUR INTERRUPT-ENABLE

This bit enables an interrupt at midnight rollover.

1 = 24-hour interrupt enabled
0 = 24-hour interrupt disabled

ALARM INTERRUPT ENABLE

This bit enables the alarm interrupt.

1 = Alarm interrupt enabled
0 = Alarm interrupt disabled

MIN INT ENABLE

This bit enables the minute-tick interrupt.

1 = Minute tick interrupt enables
0 = Minute tick interrupt disabled

SW INTERRUPT ENABLE

This bit enables the stopwatch interrupt. The stopwatch counts down and remains at decimal -1 until it is reprogrammed. Note: If this bit is enabled with -1 (decimal) in the stopwatch register, an interrupt will be posted on the next minute tick.

5.1.4.6 STOPWATCH REGISTER (STPWTCH).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED										STOPWATCH COUNT					

ADDRESS: $(FF)FFFB12 Reset Value: $0000

Figure 5-7. Stopwatch Register

UNUSED

These unused bits read 0.

STOPWATCH COUNT

This field contains the stopwatch countdown value. The highest allowable value is 62 minutes. The countdown will not be activated again until a nonzero value (less than 63 minutes) is written to the stopwatch-count register.

SECTION 6
TIMER

The MC68328 processor contains two identical general-purpose 16-bit timers with a programmable prescaler and a software watchdog timer. Figure 6-1 shows the block diagram of the time module.

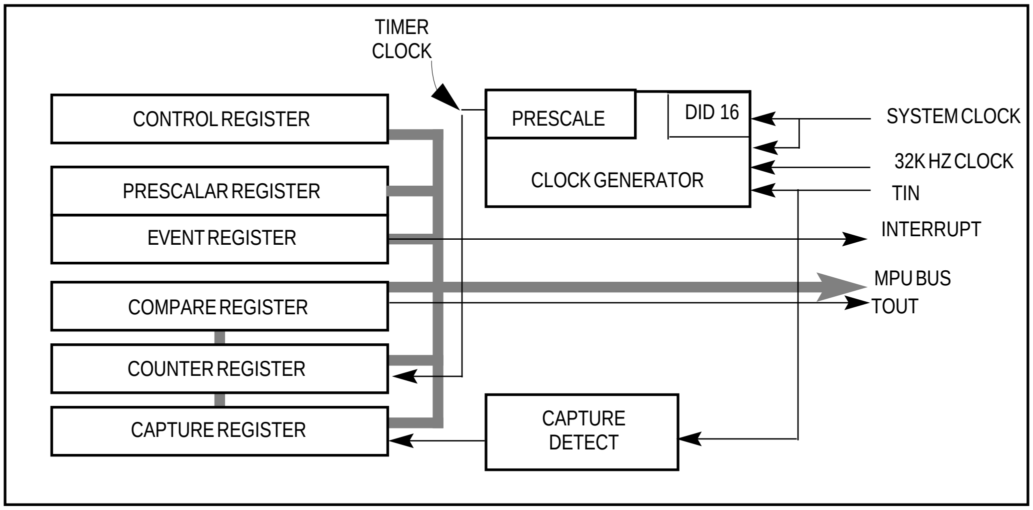


Figure 6-1. Timer Block Diagram

The key features of the timers include:

- Maximum period of 524 seconds (at 32kHz)
- 240-ns resolution (at 16.67 MHz)
- Programmable sources for the clock input, including external clock
- Input capture capability with programmable trigger edge on input pins
- Output compare with programmable mode for the output pins
- Cascading timers for constructing one 32-bit timer
- Free run and restart modes

The software watchdog timer has the following features:

- 16-bit counter and reference register
- Maximum period of 16.38 seconds

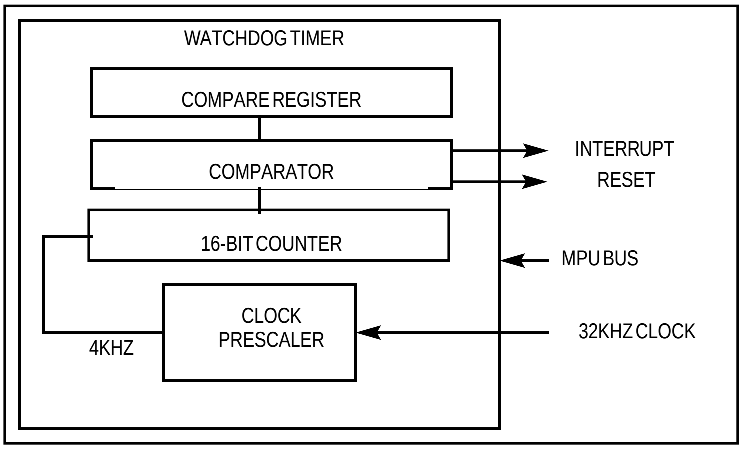


Figure 6-2. Watchdog Timer Block Diagram

- 0.25-ms resolution
- Time-out causes system RESET or issues a maskable interrupt

6.1 GENERAL PURPOSE TIMERS

The clock input to the prescaler may be selected from the main clock (divided by 1 or by 16), from the corresponding timer input (TIN1 or TIN2) pin, or from the 32-kHZ clock. TIN is synchronized to the internal clock. The clock input source is determined by the CLK SOURCE bits of the corresponding timer control register (TCR). The timer prescaler register is an 8-bit wide read/write register.The prescaler is programmed to divide the clock input by values from 1 to 256 (i.e. 0 to 255 in the register). The prescaler output serves as an input to the 16-bit counter.

Each timer may be configured to count until it reaches a reference. Then, it either starts a new time count immediately or continues to run. The free run/restart (FRR) bit of the corresponding TCR selects each mode. Upon reaching the reference value, the corresponding timer-status register (TSR) bit is set and an interrupt is issued if the interrupt-enable bit in TCR is set.

Each timer may output a signal on the timer-output (TOUT1 or TOUT2) pin when it reaches the reference value, as selected by the output mode (OM) bit of the corresponding control register, TCR. This signal can be an active-low pulse for a system clock-wide, or a toggle of the current output under program control. The output can also serve as an input to the other timer, resulting in a 32-bit timer.

Each timer has a 16-bit timer-capture register that latches the value of the counter when a defined transition (of TIN1 or TIN2) is sensed by the corresponding input-capture edge detector. The type of transition triggering the capture is selected by the capture edge (CE) bits in the corresponding TCR.

When a capture or reference event occurs, the corresponding TSR bit is set and a maskable interrupt is issued. The timer is not activated after reset and must be programmed as users require.

6.2 SOFTWARE WATCHDOG TIMER

The software watchdog timer protects against system failures by providing a means to escape from unexpected input conditions, external events, or programming errors. The third 16-bit timer block serves as a software watchdog timer for providing protection.

Once started, software must clear the software watchdog timer on a regular basis so that it never reaches its time-out value. Upon reaching the time-out value, it is assumed that a system failure has occurred, and the software-watchdog logic initiates a hardware reset of the chip or a maskable interrupt to the CPU, depending on the force-interrupt (FI) control bit in the watchdog control register (WCR).

The software watchdog timer uses the 32 kHz clock as the input to the prescaler. The prescaler circuitry divides the clock input by a fixed value of 8. The output of this prescaler circuitry is connected to the input of the 16-bit counter. The reference/compare register is a 16-bit programmable register. The maximum value that can be programmed is 65535, i.e. FFFF in hex.

The watchdog timer starts counting once it is activated by setting the enable bit in the control /status register. The counter is locked after it starts running; it will be disabled and cleared if and only if a software reset or external reset is asserted. Once the count reaches the reference value programmed in the reference register, either a maskable interrupt or a software reset will be issued to the system, depending on the FI bit in the control/status register. The counter asserts an internal output to the system-reset logic for an input clock cycle, i.e. the 32 kHZ clock if the FI bit is clear. Otherwise, the maskable interrupt is asserted to the CPU. In the case of an interrupt, the counter will continue to count. Both the interrupt and counter will be cleared by writing into the counter.

The reset source bits (RS1-0) in the SCR are updated with the cause of the reset identified as the software watchdog. Users can also check the reset-status bit in the watchdog timer control/status register to identify the reset source.

The value of the software watchdog timer can be read at any time.

6.3 SIGNAL DESCRIPTIONS

TIN

This pin is the input to the timer and can be used in capture mode to latch the contents of the free-running counter. It can also serve as the source of the clock to the prescaler.

TOUT

This pin is the output of the timer and can be programmed to toggle or pulse whenever a “compare” event occurs.

6.4 PROGRAMMER’S MODEL

Users may modify the general-purpose timer registers at any time.

6.4.1 General Purpose Timer

This section describes the timer registers.

6.4.1.1 COUNTER REGISTER. The counter register is a 16-bit read-only register that can be read at anytime without disturbing the current count.

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

Timer 1 Address: $(FF)FFF608 or

Timer 2 Address: (FF)FFF614

Reset Value: $0000

Figure 6-3. Timer Counter Register

COUNT

This is the current count value.

6.4.1.2 TIMER CONTROL REGISTERS. These identical registers control the overall individual timer operation.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED							FRR	CAPTURE EDGE		OM	IRQEN	CLKSOURCE			TEN

Timer 1 Address: (FF)FFF600

Timer 2 Address: (FF)FFF60C

Reset Value: $0000

Figure 6-4. Timer Control Registers

CAPTURE EDGE

These bits control the operation of the capture function. The bits are encoded as:

00 = Disable interrupt on capture event
01 = Capture on rising edge and generate interrupt on capture
10 = Capture on falling edge and generate interrupt on capture
11 = Capture on rising or falling edges and generate interrupt on capture

OM   Output Mode

This bit controls the output mode of the timer after a reference-compare event.

0 = Active-low pulse for one SYSCLK period
1 = Toggle output

IRQEN   Reference Event Interrupt-Enable

This bit controls the generation of an interrupt on a reference-compare event.

0 = Disable interrupt on reference event
1 = Enable interrupt on reference event

FRR Free Run/Restart

This bit controls the timer operation after a “reference” event occurs. In the free-run mode, the timer continues running. In the restart mode, the counter is reset to $0000, then resumes counting.

0 = Restart mode
1 = Free-run mode

CLKSOURCE

These bits control the clock source to the timer. Stop-count freezes the timer without causing the value in the counter to be reset to $0000.

000= Stop count (clock disabled)
001= System clock to timer
010= System clock divided by 16
011= TIN pin is the clock source
1xx = 32kHz clock

TEN Timer Enable

This bit enables the timer module.

0 = Timer disabled
1 = Timer enabled

NOTE

When this bit transitions from 0 to 1, the counter is reset to $0000. The other registers are not disturbed.

6.4.1.3 TIMER PRESCALER REGISTER. These identical registers control the overall individual timer operation.

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

Timer 1 Address: (FF)FFF602
Timer 2 Address: (FF)FFF60E

Reset Value: $0000

Figure 6-5. Timer Prescaler Registers

PRESCALER

These bits determine the divide value of the prescaler between 1 and 256. $00 divides by 1 and $FF divides by 256.

6.4.1.4 TIMER-COMPARE REGISTER. Each “compare” register is a 16-bit register that contains the value that is compared with the free-running counter as part of the output-compare function. This is a memory-mapped read-write register.

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

Timer 1 Address: (FF)FFF604
Timer 2 Address: (FF)FFF610

Reset Value: $FFFF

Figure 6-6. Timer-Compare Register

This register is set to all 1's at system reset. The "compare" value is not matched until the counter increments to equal this value.

6.4.1.5 TIMER-CAPTURE REGISTER. Each capture register is a 16-bit register that latches the counter value during a capture operation when an edge occurs on the TIN pin, as programmed in the timer-control register. This register appears as a memory-mapped read-only register to users and is cleared at system reset.

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

Timer 1 Address: $(FF)FFF606
Timer 2 Address: $(FF)FFF612

Reset Value: $0000

Figure 6-7. Timer-Capture Register

6.4.1.6 TIMER-STATUS REGISTER.

The status register indicates the timer status. When a "capture" event occurs, it is posted by setting the CAPT bit. When a "compare" event occurs, the COMP bit is set. Users must clear these bits to clear the interrupt (if enabled). These bits are cleared by writing $00 and will clear only if they have been read while set, which ensures that an interrupt will not be missed if it occurs between the status-read and the interrupt-clear.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED														CAPT	COMP

Timer 1 Address: $(FF)FFF60A
Timer 2 Address: $(FF)FFF616

Reset Value: $0000

Figure 6-8. Timer Status Register

CAPT — Capture Event

While high, this bit indicates that a "capture" event occurred.

0 = No capture event occurred
1 = Capture event occurred

COMP — Compare Event

While high, this bit indicates that a "compare" event occurred.

0 = No compare event occurred
1 = Compare event occurred

6.4.2 Software Watchdog Timer

The software watchdog timer module has a 3-bit prescaler that is not accessible to users: a watchdog-compare register (WRR), a read-only 16-bit watchdog counter register (WCN), and a 4-bit watchdog-control/status register (WCR).

6.4.2.1 WATCHDOG-COMPARE REGISTER. The 16-bit compare register contains the “compare” value for the watchdog time-out. It appears as a memory-mapped read-write register to users.The reset value of the register is $FFFF.

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

Address: $(FF)FFF61A Reset Value: $FFFF

Figure 6-9. Watchdog-Compare Register

COMPARE VALUE

When the counter counts up to the value in this register, it generates a system reset . This register resets to $FFFF. The programmed value in the register will not be affected if the force-interrupt mode is set in the control register.

6.4.2.2 WATCHDOG COUNTER REGISTER. The watchdog counter register is a 16-bit up-counter and appears as a memory-mapped register that may be read at any time.

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

Address: $(FF)FFF61C Reset Value: $0000

Figure 6-10. Watchdog Counter Register

COUNT

This is the current count value.

A read cycle to the counter register causes the current value of the watchdog timer to be read. Reading the watchdog timer does not affect the counting operation.

A write cycle to the counter register causes the counter and prescaler to be reset. A write cycle should be executed on a regular basis so that the watchdog timer is never allowed to reach the reference value during normal program operation.

6.4.2.3 WATCHDOG-CONTROL/STATUS REGISTER (WCR).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED												W/DRST	LOCK	FI	WDEN

Address: $(FF)FFF618 Reset Value: $0000

Figure 6-11. Watchdog-Control/Status Register

This register consists of 4 control or status bits. Upon reset, the watchdog timer is disabled. All bits are cleared. The LOCK bit will set if and only if the watchdog timer is activated. Once the bit is set, it will be cleared only by a software reset or external reset.

WDEN

This bit enables the watchdog timer. While this bit is low, the watchdog is disabled.

0 = Watchdog disabled
1 = Watchdog enabled

FI

This bit indicates that the interrupt should be generated instead of a software reset.

0 = Software reset mode, the watchdog interrupt is disabled
1 = Forced watchdog interrupt instead of software reset

LOCK

This bit is not user programmable. It is set when the watchdog timer is activated.

0 = Watchdog timer is not locked
1 = Watchdog timer is locked; disable writing to WDEN bit

W/DRST

This bit indicates software reset status.

0 = Not reset
1 = Set when software reset is activated.

This bit can be cleared only by writing a 0 to the bit in the control register.

SECTION 7
PARALLEL PORTS

7.1 I/O PORTS MODULE

The MC68328 processor provides 10 multipurpose, configurable parallel ports. This section describes the operation of the ports and offers suggestions on how to best use them. Each port is described and, where needed, individual pins are detailed.

7.1.1 Port Operation

There are three types of ports on the MC68328 processor. This section describes the functionality of each port.

7.1.1.1 BASIC PORT. Ports A, B, C, E, F, G, J and K are basic ports. Figure 7-1 illustrates their operation.

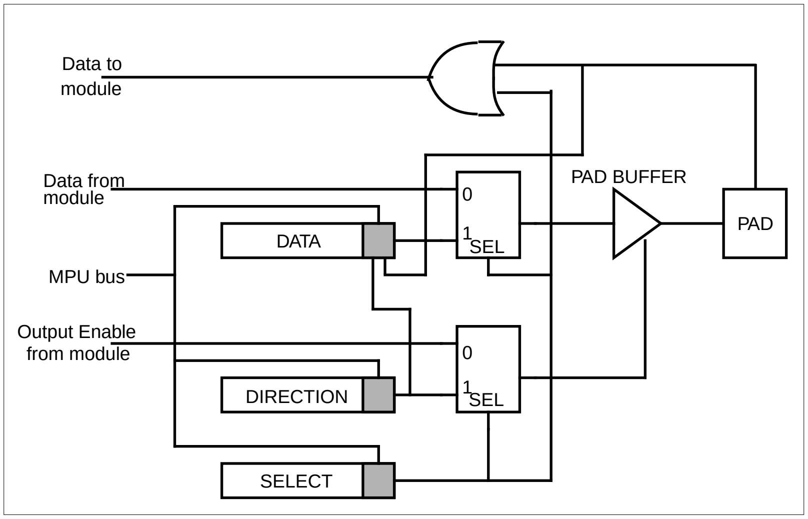


Figure 7-1. Basic Port

Basic ports multiplex two functions onto one pin. One function is the I/O and the other is the internal module connected to this pin. Figure 7-1 refers to signals to and from a module. For example, for port K, bit 0, the “Data from module” signal is connected to the master SPI module TXD signal. Because this bit is output-only, the “Output Enable from module” signal is always asserted (enabling the output) and the “Data to module” signal in Figure 7-1 is not used. Another example is port K, bit 1, where this signal’s module function is the master SPI RXD signal, which is input-only. In this case, the “Output Enable from module” input is negated and the “Data from module” signal is not used (0). The “Data to module” signal is connected to the master SPI RXD input.

While the SELECT bit is clear (default), the module pin function is enabled. Port K, bit 0 is the master SPI TXD signal. The master SPI module controls the direction of data flow (always output). While the SELECT bit is set (if the DIRECTION bit is 1 [direction = output]), data written to the DATA register is presented to the pin. If DIRECTION is 0 (input), data present on the pin is sampled and presented to the CPU when a read cycle is executed. While the DIRECTION is output, the actual pin level is presented during read accesses. This may not be the same as the data that was written if the pin is overdriven. To prevent glitches during a mode change from unselected to selected, the intended data should be written to the DATA register before going to the selected mode.

A Programming Example for Port K:

> Assume the slave SPI is to be enabled, the master SPI and the PCMCIA are not used. The pins associated with the unused modules will serve as I/Os. Port K bits 2-0 will be inputs and Port K bits 7-6 will be outputs.
>
> Value in port K SELECT register:
> $C7 bits 7-6 in I/O mode (1’s)
> bits 5-3 in slave SPI mode (0’s)
> bits 2-0 in I/O mode (1’s)
> Value in port K DIRECTION register:
> $C0 bits 7-6 outputs (1’s)
> bits 5-3 are inputs (0’s) *see Note below
> bits 2-0 inputs (0’s)
> Value in port K DATA register:
> bits 7-6 are written with the value to be output on port K, bits 7-6
> bits 5-3, when read contain the current value on the slave SPI pins
> bits 2-0 contain current value on port K, bits 2-0
>
> Bits that are in the module mode (SELECT = 0) can be read and written. While bits are configured as outputs, data that is written can be read back. Bits configured as inputs can be written but the current pin value is read back. In either case, writing to bits has no effect on the pins.

7.1.1.2 PULLUP PORT. Port M is a pullup port that operates like a basic port, but adds a switchable pullup resistor to the pin. Figure 7.2 illustrates its operation. Users enable the pullup resistor by writing bits in the PULLUP register to 1. The pullup resistors can be individually selected and operate whether the I/O port is selected or deselected. Refer to the section describing port M for more details.

7.1.1.3 INTERRUPT PORT. Port D is an interrupt port that has all of the basic and pullup-port capabilities with the addition of interrupt capabilities. Figure 7-3 illustrates the operation of the interrupt port. Port D does not have module signals associated with its signals; it is intended as a general-purpose, interrupt-generating port or a keyboard-input port.

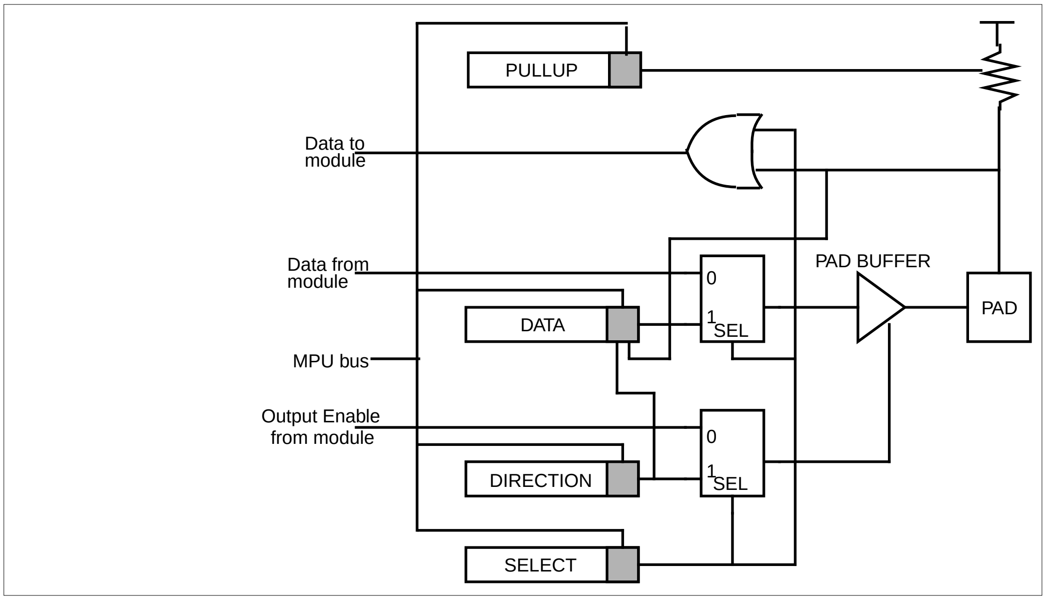


Figure 7-2. Pullup Port

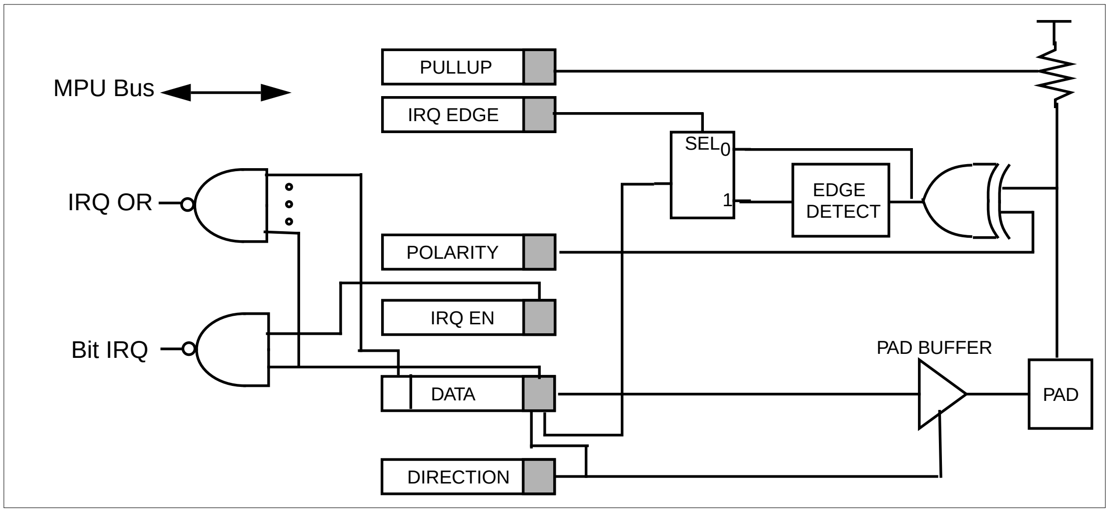


Figure 7-3. Interrupt Port

The interrupt port generates 9 interrupt signals. The individual port bits generate 8 of the interrupts. One interrupt is the logical-OR of all 8 bits. The individual interrupt bits can be masked on a bit-by-bit basis. The OR interrupt must be enabled or disabled in the interrupt module. Refer to the interrupt controller block (Section 2.3) for more information. Individual interrupts can be configured as edge- or level-sensitive, and the polarity can be selected.

7.1.2 Port A

Port A is multiplexed with address lines A16-A23. Unused address pins can serve as parallel I/Os on a bit-by-bit basis. After reset, these signals default to their address function. Three bits are associated with each port pin: data, direction, and select. Bit “n” of each register byte controls its associated pin function. The programmer’s model for port A is:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRECTION								DATA							
DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0

Address: $FFFFF400 Reset Value: $0000

Figure 7-4. Port A Data/Direction Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED								SELECT							
0	0	0	0	0	0	0	0	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0

Address: $FFFFF402 Reset Value: $0000

Figure 7-5. Port A Select Register

DIRECTION- DIR[7:0]

These bits control the pin directions. While high, the pins are outputs; while low, the pins are inputs. These bits reset to 0 and have no function while the SELECT bits are low.

DATA- D[7:0]

These bits control or report the data on the pins while the associated SELECT bits are high. While the DIRECTION bits are high (output), D[7:0] controls the data to the pins. While the DIRECTION bits are low (input), D[7:0] reports the signal level on the pins. The data bits may be written at any time. Bits that are configured as inputs will accept the data, but the written data will not be accessible until their respective pins are configured as outputs. The actual value on the pin is reported when these bits are read regardless of whether they are configured as input or output. These bits reset to 0 while the SELECT bits are low.

SELECT- SEL[7:0]

These bits select whether address [23:16] or I/O port signals are connected to the pins. While high, the port I/O functions are connected to the pin. While low, the address function is connected.

7.1.3 Port B

Port B is multiplexed with data lines D7-D0. In an 8-bit-only system, these pins can be configured as a parallel port. However, the boot-up sequence must configure port B as I/O. This port is not affected by the BUSW pin. On reset, the data lines are connected to the pins. The programmer’s model for Port B is:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRECTION								DATA							
DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0

Address: $FFFFF408 Reset Value: $0000

Figure 7-6. Port B Data/Direction Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED								SELECT							
0	0	0	0	0	0	0	0	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0

Address: $FFFFF40A Reset Value: $0000

Figure 7-7. Port B Select Register

DIRECTION- DIR[7:0]

These bits control the pin directions. While high, the pins are outputs; while low, the pins are inputs. These bits reset to 0 and have no function while the SELECT bits are low.

DATA- D[7:0]

These bits control or report the data on the pins while the associated SELECT bits are high. While the DIRECTION bits are high (output), D[7:0] controls the data to the pins. While the DIRECTION bits are low (input), D[7:0] reports the signal level on the pins. The data bits may be written at any time. Bits that are configured as inputs will accept the data, but the written data will not be accessible until their respective pins are configured as outputs. The actual value on the pin is reported when these bits are read regardless of whether they are configured as input or output. These bits reset to 0 while the SELECT bits are low.

SELECT- SEL[7:0]

These bits select whether CPU data-bus low byte or I/O port signals are connected to the pins. While high, the port I/O function is connected to the pin. While low, the D7-D0 functions are connected.

7.1.4 Port C

Port C is multiplexed with various 68000 bus-control signals that are identified below.

Table 7-1. Port C Bit Functions

Bit	Port Function	Other Function
0	Bit 0	MOCLK
1	Bit 1	UDS
2	Bit 2	LDS
3	none	none
4	Bit 4	$\overline{\text{NMI}}$
5	Bit 5	$\overline{\text{DTACK}}$
6	Bit 6	$\overline{\text{WE}}$ (PCMCIA)
7	none	none

All 8 bits are implemented in the registers, but only 6 bits connect to the outside. As with other ports, each bit can be individually configured. Bit 0 can be used only when the on-chip PLL is selected because the MOCLK function is needed to disable the PLL. The on-chip PLL will be selected when this bit serves as a port I/O. The programmer's model for port C follows.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRECTION								DATA							
DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0

Address: $FFFFF410 Reset Value: $0000

Figure 7-8. Port C Data/Direction Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED								SELECT							
0	0	0	0	0	0	0	0	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0

Address: $FFFFF412 Reset Value: $0000

Figure 7-9. Port C Select Register

DIRECTION- DIR[7:0]

These bits control the pin directions. While high, the pins are outputs; while low, the pins are inputs. These bits reset to 0 and have no function while the SELECT bits are low. Note that bits 2,3, and 7, while implemented, do not control any I/O port pins.

DATA- D[7:0]

These bits control or report the data on the pins while the associated SELECT bits are high. While the DIRECTION bits are high, the DATA register bits control the pins. While the DIRECTION bits are low, “other functions” report the signal driving the pins. These bits reset to 0. The data bits may be written at any time. Bits that are configured as inputs will accept the written data, but it will not be accessible until the respective pins are configured as outputs. The actual value on the pin is reported when these bits are read.

SELECT- SEL[7:0]

These bits select whether address or port signals are connected to the pins. While high, the port I/O function is connected to the pin; while low, the address pins are connected.

7.1.5 Port D

Port D has features intended for use as a keyboard input port; however, it can be used as a general-purpose port. Multiple keyboard support functions are provided. As with the other ports, each pin can be configured as an input or output on a bit-by-bit basis. While configured as inputs, each pin can generate a CPU interrupt. Additionally, a group interrupt can be generated. This interrupt is the OR (negative logic) of all pins on the port. Generated interrupts can be sensitive to either level or edges as selected by users. Additionally, the polarity can be selected. Each pin is also equipped with a switchable pullup resistor. The programmer’s model for port D is shown below.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRECTION								DATA							
DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0

Address: $FFFFF418 Reset Value: $0000

Figure 7-10. Port D Data/Direction Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULLUP								UNUSED							
PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	0	0	0	0	0	0	0	0

Address: $FFFFF41A Reset Value: $0000

Figure 7-11. Port D Pullup Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POLARITY								INT ENABLE							
POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0	IQEN7	IQEN6	IQEN5	IQEN4	IQEN3	IQEN2	IQEN1	IQEN0

Address: $FFFFF41C Reset Value: $0000

Figure 7-12. Port D INT Enable/Polarity Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED								INT EDGE							
0	0	0	0	0	0	0	0	IQEG7	IQEG6	IQEG5	IQEG4	IQEG3	IQEG2	IQEG1	IQEG0

Address: $FFFFF41E Reset Value: $0000

Figure 7-13. Port D INT Edge Register

DIRECTION- DIR[7:0]

These bits control the pin directions. While high, the pins are outputs; while low, the pins are inputs. Because there are no SELECT bits associated with this port, the I/O function is always enabled.

DATA- D[7:0]

These bits control or report the data on the pins. While the DIRECTION bits are high, DATA[7:0] controls the data to the pins. While the DIRECTION bits are low, DATA[7:0] reports the signal driving the pins. These bits reset to 0 while the DIRECTION bits are low. The data bits may be written at any time. Bits that are configured as inputs will accept the written data, but the data will not be accessible until the respective pins are configured as outputs. Note that the actual value on the pin is reported when these bits are read. Bits that are configured as edge-sensitive interrupts will read 1 when an edge is detected. The interrupt is cleared by writing 1to the set bits.

PULLUP- PU[7:0]

These bits enable the pullup resistors on the port. While high, the pullup resistors are enabled. While low, the pullup resistors are disabled. The pullups are enabled on reset.

POLARITY- POL[7:0]

These bits select the input signal polarity. While high, the input data is inverted before being presented to the holding register; while low, the data is unchanged. Interrupts are active-high (or rising edge) while these bits are low. Interrupts are active-low (or falling edge) while these bits are high.

INT ENABLE- IQEN[7:0]

These bits allow the interrupts to be presented to the interrupt controller block.

EDGE ENABLE- IQEG[7:0]

These bits, while high, enable edge interrupts. While low, level-sensitive interrupts are selected. The polarity of the edge (rising or falling) is selected by the POLARITY bits.

7.1.6 Port E

Port E is multiplexed with 7 chip-select signals that are identified below.

Table 7-2. Port E Bit Functions

Bit	Port Function	Other Function
0	none	none
1	Bit 1	$\overline{\text{CSA1}}$
2	Bit 2	$\overline{\text{CSA2}}$
3	Bit 3	$\overline{\text{CSA3}}$
4	Bit 4	$\overline{\text{CSB0}}$
5	Bit 5	$\overline{\text{CSB1}}$
6	Bit 6	$\overline{\text{CSB2}}$
7	Bit 7	$\overline{\text{CSB3}}$

All 8 bits are implemented in the registers, but only bits 7-1 connect to the outside. As with other ports, each bit can be individually configured, as needed. The programmer's model for port E is shown below.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRECTION								DATA							
DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0

Address: $FFFFF420 Reset Value: $0000

Figure 7-14. Port E Data/Direction Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULLUP								SELECT							
PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0

Address: $FFFFF422 Reset Value: $8080

Figure 7-15. Port E Select Register

DIRECTION- DIR[7:0]

These bits control the pin directions. While high, the pins are outputs; while low, the pins are inputs. These bits reset to 0 and have no function while the SELECT bits are low.

DATA- D[7:0]

These bits control or report the data on the pins while the associated SELECT bits are high. While the DIRECTION bits are high (output), D[7:0] controls the data to the pins. While the DIRECTION bits are low (input), D[7:0] reports the signal level on the pins. The data bits may be written at any time. Bits that are configured as inputs will accept the data but the written data will not be accessible until their respective pins are configured as outputs. Note that the actual value on the pin is reported when these bits are read regardless of whether they are configured as input or output. These bits reset to 0.

PULLUP - PU[7:0]

These bits enable the pullup resistors on the port. While high, the pullup resistors are enabled; while low, they are disabled. Port E, bit 7 pullup resistor is enabled after reset.

SELECT- SEL[7:0]

These bits select whether CSA1B-CSB3B or I/O port signals are connected to the pins. While high, the port I/O functions are connected to the pin; while low, the chip-select function is connected.

7.1.7 Port F

Port F is multiplexed with address lines A23-A31. Unused address pins can serve as parallel I/Os on a bit-by-bit basis. The programmer's model for port F is shown below.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRECTION								DATA							
DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0

Address: $FFFFF428 Reset Value: $0000

Figure 7-16. Port F Data/Direction Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULLUP								SELECT							
PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0

Address: $FFFFF42A Reset Value: $FFFF

Figure 7-17. Port F Select Register

DIRECTION- DIR[7:0]

These bits control the pin directions. While high, the pins are outputs; while low, the pins are inputs. These bits reset to 0 and have no function while the select bits are low.

DATA- D[7:0]

These bits control or report the data on the pins while the associated SELECT bits are high. While the DIRECTION bits are high, D[7:0] controls the data to the pins. While the DIRECTION bits are low, D[7:0] reports the signal driving the pins. These bits reset to 0 while the SELECT bits are low. The data bits may be written at any time. Bits that are configured as inputs will accept the written data but it will not be accessible until the respective pins are configured as outputs. The actual value on the pin is reported when these bits are read.

PULLUP - PU[7:0]

These bits enable the pullup resistors on the port. While high, the pullup resistors are enabled; while low, they are disabled. The pullup resistors are enabled after reset.

SELECT- SEL[7:0]

These bits select whether address or port signals are connected to the pins. While high, the port I/O function is connected to the pin. While low, the address pins are connected.The I/O function is selected following reset.

7.1.8 Port G

Port G is multiplexed with timer and serial communication signals. The signals are identified below. Refer to the timer, PWM, RTC, and UART sections for descriptions of signal functions. All 8 bits are implemented in the registers and each is connected to the outside. As with other ports, each bit can be individually configured, as needed. The programmer's model for port G is shown below.

Table 7-3. Port G Bit Functions

Bit	Port Function	Other Function
0	Bit 0	UART TXD
1	Bit 1	UART RXD
2	Bit 2	PWMOUT
3	Bit 3	TOUT2
4	Bit 4	TIN2
5	Bit 5	TOUT1
6	Bit 6	TIN1
7	Bit 7	RTCOUT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRECTION								DATA							
DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0

Address: $FFFFF430 Reset Value: $0000

Figure 7-18. Port G Data/Direction Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULLUP								SELECT							
PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0

Address: $FFFFF432 Reset Value: $0000

Figure 7-19. Port G Select Register

DIRECTION- DIR[7:0]

These bits control the direction of the pins. While high, the pins are outputs. While low, the pins are inputs. These bits reset to 0 and have no function while the SELECT bits are low.

DATA- D[7:0]

These bits control or report the data on the pins while the associated SELECT bits are high. While the DIRECTION bits are high, the DATA bits control the pins. While the DIRECTION bits are low, the actual levels on the pins are reported. These bits reset to 0 while the SELECT bits are low. The data bits may be written at any time. Bits that are configured as inputs will accept the written data but it will not be accessible until the respective pins are configured as outputs. Note that the actual value on the pin is reported when these bits are read.

Note

While PC0MOCLK is high, the RTCOUT bit is disabled and becomes an input for the 32 KHz real-time clock reference.

PULLUP - PU[7:0]

These bits enable the pullup resistors on the port. While high, the pullup resistors are enabled; while low, they are disabled. Port E, bit 7 pullup resistor is enabled after reset.

SELECT- SEL[7:0]

These bits select whether the various functions or port I/O signals are connected to the pins. While high, the port I/O function is connected to the pin. While low, the various functions are connected.

7.1.9 Port J

Port J is multiplexed with 8 chip-select signals identified below.

Table 7-4. Port J Bit Functions

Bit	Port Function	Other Function
0	Bit 0	$\overline{\text{CSC0}}$
1	Bit 1	$\overline{\text{CSC1}}$
2	Bit 2	$\overline{\text{CSC2}}$
3	Bit 3	$\overline{\text{CSC3}}$
4	Bit 4	$\overline{\text{CSD0}}$
5	Bit 5	$\overline{\text{CSD1}}$
6	Bit 6	$\overline{\text{CSD2}}$
7	Bit 7	$\overline{\text{CSD3}}$

All 8 bits are implemented in the registers and all 8 are connected to the outside. As with other ports, each bit can be individually configured, as needed. The programmer’s model for port J is shown below.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRECTION								DATA							
DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0

Address: $FFFFF438 Reset Value: $0000

Figure 7-20. Port J Data/Direction Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED								SELECT							
0	0	0	0	0	0	0	0	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0

Address: $FFFFF43A Reset Value: $0000

Figure 7-21. Port J Select Register

DIRECTION- DIR[7:0]

These bits control the pin directions. While high, the pins are outputs; while low, the pins are inputs. These bits reset to 0 and have no function while the SELECT bits are low.

DATA- D[7:0]

These bits control or report the data on the pins while the associated SELECT bits are high. While the DIRECTION bits are high, the DATA bits control the pins. While the DIRECTION bits are low, the value on the pins is reported. These bits reset to 0 while the SELECT bits are low. The data bits may be written at any time. Bits that are configured as inputs will accept the written data but it will not be accessible until the respective pins are configured as outputs. The actual value on the pin is reported when these bits are read.

SELECT- SEL[7:0]

These bits select whether chip-select or port I/O signals are connected to the pins. While high, the port I/O function is connected to the pin; while low, the chip-select functions are connected.

7.1.10 Port K

Port K is multiplexed with signals related to the serial peripheral interfaces and PCMCIA. The signals are identified below.

Table 7-5. Port K Bit Functions

Bit	Port Function	Other Function
0	Bit 0	SPIM TXD
1	Bit 1	SPIM RXD
2	Bit 2	SPIM CLKO
3	Bit 3	SPIS EN
4	Bit 4	SPIS RXD
5	Bit 5	SPIS CLKI
6	Bit 6	PCMCIA CE2
7	Bit 7	PCMCIA CE1

All 8 bits are implemented in the registers, and each is connected to the outside. As with other ports, each bit can be individually configured, as needed. The programmer's model for port G is:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRECTION								DATA							
DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0

Address: $FFFFF440 Reset Value: $0000

Figure 7-22. Port K Data/Direction Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULLUP								SELECT							
PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0

Address: $FFFFF442 Reset Value: $FFFF

Figure 7-23. Port K Select Register

DIRECTION- DIR[7:0]

These bits control the pin directions. While high, the pins are outputs; while low, the pins are inputs. These bits reset to 0 and have no function while the SELECT bits are low.

DATA- D[7:0]

These bits control or report the data on the pins while the associated SELECT bits are high. While the DIRECTION bits are high, the DATA bits control the pins. While the DIRECTION bits are low, the levels on the pins are reported. These bits reset to 0 while the SELECT bits are low. The data bits may be written at any time. Bits that are configured as inputs will accept the written data but it will not be accessible until the respective pins are configured as outputs. The actual value on the pin is reported when these bits are read.

PULLUP - PU[7:0]

These bits enable the pullup resistors on the port. While high, the pullup resistors are enabled; while low, they are disabled. Port E, bit 7 pullup resistor is enabled after reset.

SELECT- SEL[7:0]

These bits select whether serial-module or port I/O signals are connected to the pins. While high, the port I/O function is connected to the pin. While low, the serial module functions are connected.

7.1.11 Port M

Port M is multiplexed with signals related to the interrupts and UART. The signals are identified below. Each bit has a selectable pullup resistor associated with it.

Table 7-6. Port M Bit Functions

Bit	Port Function	Other Function
0	**Bit 0**	**$\overline{\text{CTS}}$**
1	**Bit 1**	**$\overline{\text{RTS}}$**
2	**Bit 2**	**$\overline{\text{IRQ 6}}$**
3	**Bit 3**	**$\overline{\text{IRQ 3}}$**
4	**Bit 4**	**$\overline{\text{IRQ 2}}$**
5	**Bit 5**	**$\overline{\text{IRQ 1}}$**
6	**Bit 6**	**$\overline{\text{PEN IRQ}}$**
7	**Bit 7**	**UART GPIO**

All 8 bits are implemented in the registers and each is connected to the outside. As with other ports, each bit can be individually configured, as needed. Note that there is no alternate signal associated with bit 7. The programmer’s model for port M is shown below.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIRECTION								DATA							
DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0

Address: $FFFFF448 Reset Value: $0000

Figure 7-24. Port M Data/Direction Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PULLUP								SELECT							
PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0

Address: $FFFFF44A Reset Value: $0000

Figure 7-25. Port M Select Register

DIRECTION- DIR[7:0]

These bits control the pin directions. While high, the pins are outputs; while low, the pins are inputs. These bits reset to 0 and have no function while the SELECT bits are low.

DATA- D[7:0]

These bits control or report the data on the pins while the associated SELECT bits are high. While the DIRECTION bits are high, the DATA bits control the pins. While the DIRECTION bits are low, the pin levels are reported. These bits reset to 0 while the SELECT bits are low. The data bits may be written at any time. Bits that are configured as inputs will accept the written data but it will not be accessible until the respective pins are configured as outputs. The actual value on the pin is reported when these bits are read.

SELECT[7:0]

These bits select whether interrupt or port I/O signals are connected to the pins. While high, the port I/O function is connected to the pin; while low, the interrupt functions are connected. The I/O functions are enabled after reset.

PULLUP- PU[7:0]

These bits enable the pullup resistors on the port. While high, the pullup resistors are enabled; while low, the pullup resistors are disabled. The pullups are enabled on reset.

SECTION 8
UNIVERSAL ASYNCHRONOUS RECEIVER/ TRANSMITTER (UART)

The universal asynchronous receiver/transmitter (UART) provides serial communication with external devices such as modems and other computers. Data is transported in character blocks at data rates ranging from 300 bps to over 1 Mbps using a standard “start-stop” format. Some of the key features of the UART include:

- Full duplex operation
- Flexible 5-wire serial interface
- Direct support of IrDA physical layer protocol
- Robust receiver data sampling with noise filtering
- 8-byte FIFOs for transmit and receive
- 7-, 8-bit operation with optional parity
- Generation and break detection
- Baud-rate generator
- Flexible clocking options
- Standard baud rates 300bps to 115.2kbps with 16x sample clock
- External 1x clock for high-speed synchronous communication
- Programmer’s model optimized for 16-bit bus
- 8 maskable interrupts
- Low-power idle mode

The UART performs all 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 where users provide the data-bit clock. Figure 8-1 shows a high-level block diagram of the module.

8.1 SERIAL INTERFACE SIGNALS

There are 5 signals accessible to users and are described below. If users need any or all UART signals, the appropriate port bits can be programmed to assume their UART function. Refer to **Section 7 Parallel Ports** for information about programming the ports.

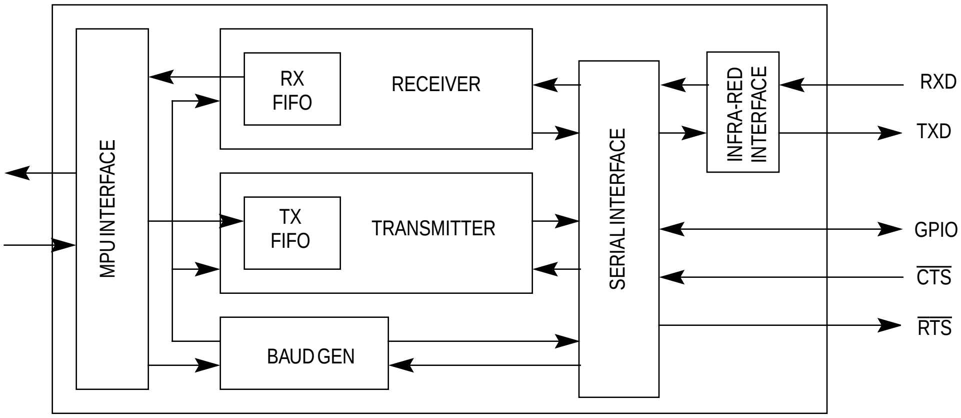


Figure 8-1. UART Block Diagram

TXD Transmit Data

This pin is the transmitter serial output. While in normal mode, NRZ data is output. While in IrDA mode, a 3/16 bit-period pulse is output for each 0 bit transmitted. For RS-232 applications, this pin must be connected to an RS-232 transmitter. For infrared applications, this pin can directly drive an infrared tranceiver module.

$\overline{\text{CTS}}$ Clear To Send

This active-low input controls the transmitter. Normally, the transmitter waits until this signal is active (low) before transmitting a character. If the IGNORE CTS bit is set, the transmitter sends a character whenever a character is ready to transmit. This signal can then serve as a general-purpose input with a read status in the CTS STATUS bit. This pin can post an interrupt on any transition of itself if the interrupt is enabled.

RXD Receive Data

This pin is the receiver serial input. While in normal operation, NRZ data is expected. While in IrDA mode, a narrow pulse is expected for each 0 bit received. Use external circuitry to convert the infrared signal to an electrical signal. RS-232 applications need an external RS-232 receiver to convert from RS-232 voltage levels.

$\overline{\text{RTS}}$ Ready To Send

This output pin serves two purposes. Normally, the receiver 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. For other applications, this pin can serve as a general-purpose output controlled by the $\overline{\text{RTS}}$ bit in the receiver register.

GPIO General-Purpose Input/Output

This bidirectional pin serves several functions. It can be a general-purpose input that: (1) can post interrupts on any transition, (2) is controlled by the GPIO bit in the baud register,

(3) can serve as the source of the clock to the baud-rate generator, and (4) can output the bit clock at the selected baud rate.

8.2 SUB-BLOCK DESCRIPTION

The UART module is easy to use from both a hardware and software perspective. Five working registers provide all status and control functions. The registers are optimized for a 16-bit bus. For example, all status bits associated with the received data are available along with the data byte in a single 16-bit read. All register bits are readable and most are read/write.

The modem-control signals are flexible. $\overline{\text{CTS}}$ is an input that can provide hardware flow-control to the transmitter, or it can serve as a general-purpose input. A maskable interrupt is posted on each transition of this signal. $\overline{\text{RTS}}$ is an output from the receiver that indicates that the receiver has room in the FIFO for data. This bit can be configured as a general-purpose output. A GPIO pin is provided that can bring an external bit-clock into the module. It can also serve as a general-purpose input with a maskable interrupt posted on each transition. It can be configured as an output that provides a bit-clock or a signal under software control.

The UART consists of four submodules. This section briefly describes the basic functionality of the 4 blocks.

8.2.1 Transmitter

The transmitter accepts a character (byte) from the MPU bus and transmits it serially. While the FIFO is empty, the transmitter outputs continuous IDLE (1 while in NRZ, 0 while 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 at the selected bit rate. The transmitter posts a maskable interrupt when it needs parallel data. Three interrupts are available. If users want to take full advantage of the 8-byte FIFO, the FIFO EMPTY interrupt should be enabled. In the interrupt-service routine, the FIFO should be interrogated after each byte is loaded. If space is available (the TX AVAIL bit is set), more data will be loaded into the FIFO. The transmitter will not generate another interrupt until the FIFO has completely emptied. If working with software that has a large interrupt-service latency, use the FIFO HALF interrupt. In this case, the transmitter generates an interrupt when the FIFO occupancy is less than 4 bytes. If the FIFO is not needed, use the TX AVAIL interrupt. An interrupt will be generated whenever at least one space is available in the FIFO.

$\overline{\text{CTS}}$ can control the serial data flow. If $\overline{\text{CTS}}$ is negated (high), the transmitter finishes sending the character in progress (if any), then waits for $\overline{\text{CTS}}$ to again become asserted (low). Set the SEND BREAK bit in the transmitter register to generate a BREAK character (continuous 0's). Users' software must know the baud rate. The SEND BREAK bit must be asserted for a sufficient time to generate a valid BREAK character. Users can generate parity errors for debugging purposes. The transmitter operates from the 1x clock provided by the baud-rate generator.

While the infrared interface is enabled, the transmitter produces a pulse that is 3/16 of a bit time for each 0 bit sent. The TXD port can directly drive an infrared LED or directly interface with popular IrDA transceivers.

8.2.2 Receiver

The receiver accepts a serial data stream and converts it into a parallel character. It operates in two modes, 16x and 1x. In 16x mode, it searches for a start bit, qualifies it, then samples the succeeding data bits at the bit center. Jitter tolerance and noise immunity are provided by sampling at a 16x rate and using a voting technique to clean up the samples. In 1x mode, RXD is sampled on each rising edge of the bit clock.

After locating the start bit, the data bits, parity bit (if enabled), and stop bits are shifted in. If parity is enabled, it is checked and its status is reported in the receiver register. Similarly, frame errors and breaks are checked and reported. When the host is ready to read a new character, $\overline{\text{RTS}}$ is asserted and an interrupt is posted (if enabled). When the receiver register is read as a 16-bit word, the 68000 core reads the complete FIFO status, the four status bits, and the received character byte. The $\overline{\text{RTS}}$ pin can be configured as an output, which indicates the receiver is ready for data or software can directly control the pin.

As with the transmitter, the receiver FIFO is flexible. If a user's software has a short interrupt latency, the FIFO FULL interrupt can be enabled. One space is available in the FIFO when this interrupt is generated. By reading the receive register as a word, the FIFO status is presented to the M68EC000 along with the data. If the FIFO status indicates that data remains in the FIFO, the FIFO can then be emptied byte-by-byte. If the software has a longer latency, the FIFO HALF interrupt is used. This interrupt is generated when 4 bytes have been entered into the FIFO. If the FIFO is not needed, the DATA READY interrupt is used. This interrupt is generated whenever one or more characters are present in the FIFO.

While the IrDA interface is enabled, the receiver expects narrow pulses for each 0 bit received; otherwise, normal NRZ is expected. An IrDA transceiver, external to the MC68328 processor, transforms the infrared signal to an electrical signal.

8.2.3 Baud Rate Generator

The baud generator provides the bit clocks to the transmitter and receiver blocks. It consists of a prescaler that divides the clock source by any integer between 2 and 64. The output of the prescaler is then further divided by a 2^n divider. Eight taps are available at 1, 2, 4, 8, 16, 32, 64, and 128. The selected tap is the 16x clock for the receiver. This clock is further divided by 16 to provide the 50% duty cycle 1x clock to the transmitter. The baud generator is flexible enough to provide almost any "standard" baud rate from a variety of clock frequencies.The system default frequency of 16.58 MHz provides a good base for generating standard baud rates within 0.01%.

The baud generator master clock source can either be the system clock or it can be provided through the GPIO pin. By configuring port M bit 7 as an input and setting the baud source bit to 1, an external signal can directly drive the baud generator. For synchronous applications, the GPIO pin can be configured to serve as an input or output for the 1x bit-clock.

8.2.4 MPU Interface

The MPU interface contains all status/control registers and all miscellaneous logic. This block directly connects to the internal 68000 bus and provides address decode for three address lines and a full 16-bit read/write port. The interrupt line is the logical-OR of the 8 interrupt sources. While the UART EN bit is low, the master clock to all UART blocks is disabled, reducing power consumption to a minimum.

8.2.5 UART Control Register

This register controls the overall UART operation. This register resets to $0000.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UART ENABLE	RX ENABLE	TX ENABLE	RX CLK CONT	PARITY EN	ODD EVEN	STOP BITS	8/7	GPIO DELTA ENABLE	CTS DELTA ENABLE	RX FULL ENABLE	RX HALF ENABLE	RX RDY ENABLE	TX EMPTY ENABLE	TX HALF ENABLE	TX AVAIL ENABLE

ADDRESS: $(FF)FFF900 Reset Value: $0000

Figure 8-2. UART Control Register

UART ENABLE

This bit enables the UART. While this bit is low, the UART is disabled and in low-power mode. While this bit is high, the UART is active. This bit resets to 0.

0 = UART disabled
1 = UART enabled

NOTE

When the UART is first enabled after cold reset, before enabling interrupts, set the UART ENABLE and RX ENABLE bits and perform a word-read operation on the receiver register to initialize the FIFO and character-status bits.

RX ENABLE

This bit enables the receiver block. While this bit is low, the receiver is disabled and the receive FIFO is flushed. This bit resets to 0.

0 = Receiver disabled and receive FIFO flushed
1 = Receiver enabled

TX ENABLE

This bit enables the transmitter block. While this bit is low, the transmitter is disabled and the transmit FIFO is flushed. This bit resets to 0.

0 = Transmitter disabled and transmit FIFO flushed
1 = Transmitter enabled

RX CLK CONT Receiver Clock Control

This bit controls the receiver operating mode of the receiver. While low, the receiver is in 16x mode where it synchronizes to the incoming data stream and samples at the perceived center of each bit period. While high, the receiver is in 1x mode where it samples the data stream on each rising edge of the bit clock. This bit resets to 0.

0 = 16x clock mode
1 = 1x clock mode

PARITY ENABLE

This bit controls the parity generator in the transmitter and parity checker in the receiver. While high, they are enabled. While low, they are disabled.

0 = Parity disabled
1 = Parity enabled

ODD EVEN

This bit controls the sense of the parity generator and checker. While high, odd parity is generated and expected. While low, even parity is generated and expected. This bit has no function if PARITY EN is low.

0 = Even parity
1 = Odd parity

STOP BITS

This bit controls the number of stop bits transmitted after a character. While high, two stop bits are sent. While low, one stop bit is sent. This bit has no effect on the receiver, which expects one or more stop bits.

0 = 1 stop-bit transmitted
1 = 2 stop-bits transmitted

8/7

This bit controls the character length. While high, the transmitter and receiver are in 8-bit mode. While low, they are in 7-bit mode. The transmitter then ignores B7 and the receiver sets B7 to 0.

0 = 7-bit transmit-and-receive character length
1 = 8-bit transmit-and-receive character length

GPIO DELTA ENABLE

This bit enables an interrupt when the GPIO pin (while configured as an input) changes state. The current state of the GPIO pin is read in the baud-control register.

0 = GPIO interrupt disabled
1 = GPIO interrupt enabled

CTS DELTA ENABLE

While high, this bit enables an interrupt when the CTS pin changes state. While low, this interrupt is disabled. The current status of the CTS pin is read in the transmit status/control register.

0 = CTS interrupt disabled
1 = CTS interrupt enabled

RX FULL ENABLE

While high, this bit enables an interrupt when the receiver FIFO is full. This bit resets to 0.

0 = RX FULL interrupt disabled
1 = RX FULL interrupt enabled

RX HALF ENABLE

While high, this bit enables an interrupt when the receiver FIFO is more than half full. This bit resets to 0.

0 = RX HALF interrupt disabled
1 = RX HALF interrupt enabled

RX READY ENABLE

While high, this bit enables an interrupt when the receiver has at least one data byte in the FIFO. While low, this interrupt is disabled.

0 = RX interrupt disabled
1 = RX interrupt enabled

TX EMPTY ENABLE

While high, this bit enables an interrupt when the transmitter FIFO is empty and needs data. While low, this interrupt is disabled.

0 = TX EMPTY interrupt disabled
1 = TX EMPTY interrupt enabled

TX HALF ENABLE

While high, this bit enables an interrupt when the transmit FIFO is less than half full. While this bit is low, the TX HALF interrupt is disabled. This bit resets to 0.

0 = TX HALF interrupt disabled
1 = TX HALF interrupt enabled

TX AVAIL ENABLE Transmitter Available For New Data

While high, this bit enables an interrupt when the transmitter has a slot available in the FIFO. While low, this interrupt is disabled. This bit resets to 0.

0 = TX AVAIL interrupt disabled
1 = TX AVAIL interrupt enabled

8.2.6 Baud Control Register

This register controls the operation of the baud-rate generator and the GPIO pin and resets to $003F.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO DELTA	GPIO	GPIO DIR	GPIO SRC	BAUD SRC	DIVIDE			UNUSED		PRESCALER					

ADDRESS: $(FF)FFF902 Reset Value: $003F

Figure 8-3. Baud Control Register

GPIO DELTA

This bit indicates that a change occurred on the GPIO pin. If the GPIO interrupt is enabled, this bit posts an interrupt. Users can write this bit to a 1, posting an immediate interrupt for debugging purposes. This bit must be cleared by writing 0 to clear the GPIO interrupt.

0 = no GPIO interrupt posted
1 = GPIO interrupt posted

GPIO

This bit contains the current status of the GPIO pin. If GPIO is configured as an input, users can read this bit. If GPIO is configured as an output, this bit controls the pin.

0 = GPIO pin is low
1 = GPIO pin is high

GPIO DIR

This bit controls the direction of the GPIO pin. While this bit is high, the pin is an input. While this bit is low, GPIO is an output.

0 = GPIO is input
1 = GPIO is output

GPIO SRC

This bit controls the source of the GPIO pin. While high, the source is the 1x clock from the baud-rate generator. While low, the source is the GPIO bit. While the GPIO DIR bit is 0, this bit has no function.

0 = GPIO driven by GPIO bit
1 = GPIO driven by bit clock from baud generator

BAUD SRC

This bit controls the clock source to the baud-rate generator.

0 = Baud generator source is system clock
1 = Baud generator source is GPIO pin (GPIO Dir must be 0)

DIVIDER

These bits control the clock frequency produced by the baud generator. The bits are encoded as follows:

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

UNUSED

These bits are unused and read 0.

PRESCALER

These bits control the divide value of the baud generator prescaler. The divide value is determined by the following formula:

prescaler divide value = 65 (decimal) - PRESCALER

8.2.7 Receiver Register

This register controls the receiver. Status of each received character is also read from this register. The high byte of this register resets to $00. The low byte contains random data until the first character is received.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFO FULL	FIFO HALF	DATA READY	0	OVRUN	FRAME ERROR	BREAK	PARITY ERROR	RX DATA							

ADDRESS: $(FF)FFF904 Reset Value: $0000

Figure 8-4. Receiver Register

FIFO FULL

This read-only bit indicates that the receiver FIFO is full and may generate an overrun.

0 = Receive FIFO not full (no interrupt posted)
1 = Receive FIFO full (interrupt posted)

FIFO HALF

This read-only bit indicates that the receiver FIFO is more than half full.

0 = Receive FIFO less than half full (no interrupt posted)
1= Receive FIFO more than half full (interrupt posted)

DATA READY

This read-only bit indicates that at least one byte is present in the receive FIFO.

0 = No data in receive FIFO
1 = Data in receive FIFO

OVRUN FIFO Overrun

While high, 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. Under normal circumstances, this bit should never be set. It indicates that the user’s software is not keeping up with the incoming data rate. This bit is updated and valid for each received character.

0 = No FIFO overrun
1 = FIFO overrun detected

FRAME ERROR

While high, this read-only bit indicates that the current character had a framing error (missing stop bit), indicating the possibility of 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

While high, this read-only bit indicates that the current character was detected as a BREAK. The data bits are all 0 and the stop bit was also 0. The FRAME ERROR bit will always be set when this bit is set. If odd parity is selected, PARITY ERROR will also be set along with this bit. This bit is updated for each character read from the FIFO.

0 = Character is not a break character
1 = Character is a break character

PARITY ERROR

While high, this read-only bit indicates that the current character was detected with a parity error, indicating the possibility of corrupted data. This bit is updated for each character read from the FIFO. While parity is disabled, this bit always reads zero.

DATA

These read-only bits are the next receive character in the FIFO. These bits have no meaning if the DATA READY bit is 0. While in 7-bit mode, the MSB is forced to 0. While in 8-bit mode, all bits are active.

8.2.8 Transmitter Register

This register controls the transmitter operation.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFO EMPTY	FIFO HALF	TX AVAIL	SEND BREAK	IGNORE CTS	0	CTS STATUS	CTS DELTA	TX DATA							

ADDRESS: $(FF)FFF906 Reset Value: $0000

Figure 8-5. Transmitter Register

FIFO EMPTY

This read-only bit indicates that the transmit FIFO is empty.

0 = Transmitter FIFO is not empty
1 = Transmitter FIFO empty

FIFO HALF

This read-only bit indicates that the transmit FIFO is less than half full.

0 = Transmitter FIFO more than half full
1 = Transmitter FIFO less than half full

TX AVAIL Transmit FIFO Has A Slot Available

This bit indicates that the transmit FIFO has at least one slot available for data.

0 = Transmitter does not need data
1 = Transmitter needs data

SEND BREAK

This bit forces the transmitter to send a BREAK character. The transmitter will finish sending the character in progress (if any), then send BREAK until this bit is reset. Users are responsible to ensure that this bit is high for a sufficient period of time to generate a valid BREAK. Users can continue to fill the FIFO and any characters remaining will be transmitted when the BREAK is terminated.

0 = Do not send break
1 = Send break (continuous 0's)

IGNORE CTS

This bit, while high, forces the CTS signal presented to the transmitter to always be asserted, effectively ignoring the external pin. While in this mode, the CTS pin can serve as a general-purpose input.

0 = Transmit only while CTS pin is asserted
1 = Ignore CTS pin

CTS STATUS

This bit indicates the current status of the CTS pin. A "snapshot" of the pin is taken immediately before this bit is presented to the data bus. While IGNORE CTS is high, this bit can serve as a general-purpose input.

0 = CTS pin is low
1 = CTS pin is high

CTS DELTA

While high, this bit indicates that the CTS pin changed state and generates a maskable interrupt. The current state of the CTS pin is available on the CTS STATUS bit. Users can generate an immediate interrupt by setting this bit high. This feature is useful for software debugging. The CTS interrupt is cleared by writing 0 to this bit.

0 = CTS pin did not change state since last cleared
1 = CTS pin changed state

DATA

These bits are the parallel transmit-data inputs. While in 7-bit mode, D7 is ignored. While in 8-bit mode, all bits are used. Data is transmitted LSB first. A new character is transmitted when these bits are written. These bits read as 0.

8.2.9 Miscellaneous Register

This register contains miscellaneous bits to control test features of the UART block. Some bits are intended for factory use only and should not be disturbed by users.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	CLK SRC	FORCE PERR	LOOP	RSVD				RTS CONT	RTS	IRDA ENABLE	IRDA LOOP	UNUSED			

ADDRESS: $(FF)FFF908 Reset Value: $0000

Figure 8-6. Miscellaneous Register

RSVD

These bits are reserved and should remain 0.

CLK SRC

This bit selects the source of the 1x bit clock for transmit and receive. While high, the bit clock is derived directly from the GPIO pin (it must be configured as an input.) While low, (normal) the bit clock is supplied by the baud generator. This bit allows high-speed synchronous applications where a clock is provided by the external system.

0 = Bit clock generated by baud generator
1 = Bit clock supplied from GPIO (input)

FORCE PERR

While high, this bit forces the transmitter to generate parity errors if parity is enabled. This bit is provided for system debugging.

0 = Generate normal parity
1 = Generate inverted parity (error)

LOOP

This bit controls loopback for system-test purposes. While this bit is high, the receiver input is internally connected to the transmitter and ignores the RXD pin. The transmitter is unaffected by this bit. The receiver can be in either clock mode (16x or 1x) for proper operation.

0 = Normal receiver operation
1 = Internally connect transmitter output to receiver input

RTS CONT RTS Control

This bit selects the function of the $\overline{\text{RTS}}$ pin.

0 = RTS pin is controlled by the RTS bit
1 = RTS pin is controlled by the receiver FIFO. When the FIFO is full (one slot remaining) RTS is negated.

RTS

This bit controls the RTS pin while the RTS CONT bit is 0.

0 = RTS pin is 1
1 = RTS pin is 0

IRDA ENABLE

This bit enables the IrDA interface.

0 = Normal NRZ operation
1 = IRDA operation

LOOP IR

This bit controls a loopback from transmitter to receiver in the IrDA interface. This bit is provided for system testing.

0 = No IR loop
1 = Connect IR transmit to IR receiver

UNUSED

These bits are unused and read 0.

SECTION 9
SERIAL PERIPHERAL INTERFACE— SLAVE (SPIS)

The slave serial peripheral interface (SPI) operates as an externally clocked slave, allowing the MC68328 processor to interface with external master devices (for example, POCSAG paging decoder). The interface is a 3-wire system consisting of the clock, enable, and data-input pins. It is compatible with SPIs that are popular on Motorola's 68HC05 microcomputer chips.

9.1 OVERVIEW

The SPI transfers data to the MC68328 processor from a peripheral device over a serial link. A clock, controlled by the external device, controls transfer. After counting 8 clock cycles, the shift register data moves to a read buffer, generating an interrupt in the process. Figure 9-1 is a block diagram of the slave SPI.

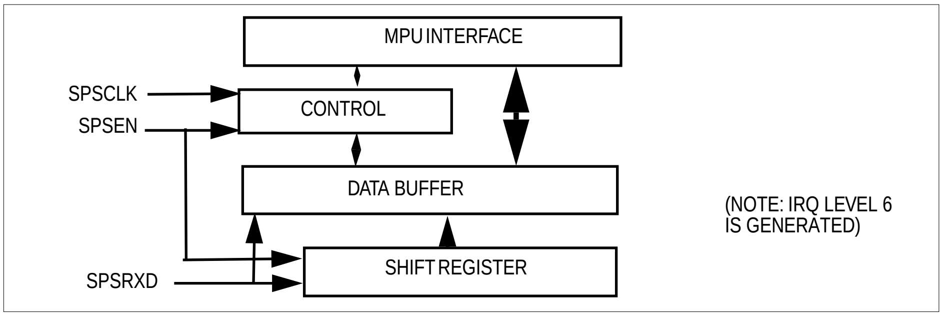


Figure 9-1. SPI Block Diagram

9.2 OPERATION

Users first initialize the SPIS program register. The SPIS then waits for the input-enable (SPSEN) and clock (SPSCLK) to control the data transfer. The shift register fills with data over the next 8 clock cycles. On the eighth clock, the shift register contents loads into the data buffer. The SPISIRQ bit is set, posting an interrupt. The valid data in the buffer awaits the service routine access.

The clock input performs shifts depending on phase and polarity. In **phase 0** mode(PHA=0), serial data are strobed on the leading edges of SPSCLK. In **phase 1** mode(PHA=1), data are strobed in on trailing edges. The **polarity (POL)** specifies the inactive state value of SPSCLK. While POL=1, the idle state of the SPSCLK is high. While POL = 0, the idle state of the SPSCLK is low. This flexibility allows operation with most serial peripheral devices on the market.

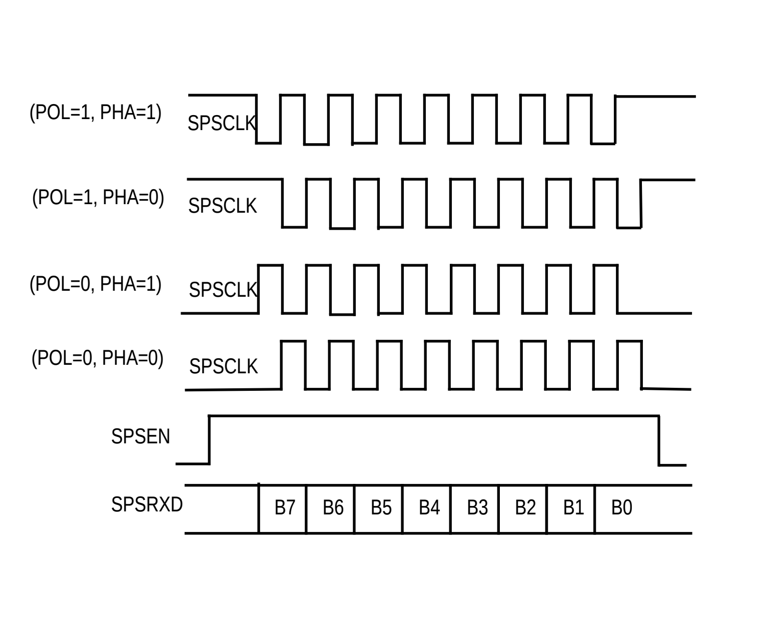


Figure 9-2. SPIS Operation

If enabled, the SPIS operates even if the system clock is inactive. After the SPIS receives a data byte from an external master, the SPIS interrupt is posted. If the system is in sleep mode, this interrupt can initiate the wakeup sequence for restoring the system clock. The SPIS bit (bit 21) in the wakeup control register IWR (at location 0x(FF)FFF308) must be set for the wakeup sequence to occur.

9.3 SIGNAL DESCRIPTIONS

SPSRXD

This pin is the serial data input to the shift register. A new bit is shifted in on each leading edge of SPSCLK while in normal mode(POL=0) or on each trailing edge of SPSCLK in polarity-inverted mode(POL=1). This signal pin is multiplexed with other signals to port K, bit 4. Refer to Section 7.1.10 for more details.

SPSCLK

This pin is the shift clock input.

SPSEN

This pin indicates that an SPI transfer is in progress. After the enable becomes active, the SPIS state machine responds to clock edges for data transfer.

9.4 SPIS REGISTER

This register controls the SPIS operation and reports its status. The data register contains the data transmitted by the external master. After reset, all bits are set to $0000.

9.4.1 SPI Slave Register

This register controls the SPI operation and reports its status. The lower byte is the input data received from an external source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPIS IRQ	IRQEN	ENPOL	DATA RDY	OVRWR	PHA	POL	SPISEN	DATA							

Address: $(FF)FFF700: Reset Value: $0000

Figure 9-3. SPI Slave Register

SPISIRQ- SPIS IRQ

This interrupt-flag bit is asserted at the end of an 8-bit transfer. The data buffer should be read before the completion of another 8-bit transfer. This flag is automatically cleared after reading.

0 = No interrupt posted
1 = Interrupt posted

IRQEN- SPIS_IRQ Enable

This bit enables the SPIS_IRQ interrupt and is cleared on reset.

0 = Interrupt disabled
1 = Interrupt enabled

ENPOL- SPSEN Polarity Control

This bit controls the polarity of the SPSEN signal and is initially set to 0.

0= SPSEN is active-low
1= SPSEN is active-high

DATARDY- Data Ready

This flag indicates that the data buffer contains updated data. The DATARDY flag is automatically cleared after the data is read.

0=Buffer is empty
1=Buffer has data

OVRWR- Overwrite

This bit indicates that the data buffer was overwritten. An interrupt is posted when an overwrite occurs. The OVRWR flag is automatically cleared after the data is read.

0=Data buffer is intact
1=Data buffer has been overwritten, data stream is corrupted

PHA

This bit sets the phase relationship between SPSCLK and SPSRxD. Refer to Figure 9-2.

0 = Phase 0 (normal); data is captured on the leading edge of SPSCLK
1 = Phase 1; data is captured on the trailing edge of SPSCLK

POL

This bit controls the polarity of the SPSCLK.

0 = The inactive state value of the clock is low (idle = 0)
1 = The inactive state value of the clock is high (idle = 1)

SPISEN

This status bit enables the slave SPI module.

1 = SPIS module enabled
0 = SPIS module disabled (default)

DATA

These are the data bits shifted from the external device. At every 8th SPSCLK edge, data from the peripheral is loaded into this buffered register. If the data buffer is not accessed before the next byte is received, it will be overwritten and the OVRWR bit will be set, posting an interrupt.

SECTION 10
SERIAL PERIPHERAL INTERFACE— MASTER (SPIM)

The serial peripheral interface (SPI) is a high-speed synchronous serial port for communicating to external devices such as A/D converters and nonvolatile RAMs. The interface is a 3- or 4-wire system, depending on unidirectional or bidirectional communication mode. The SPIM provides the clock for data transfer and can only function as a master device. It is upward-compatible with SPIs that are popular on Motorola's 6805 microcomputer chips.

10.1 OVERVIEW

The SPIM transfers data between the MC68328 processor and peripheral devices in bursts over a serial link. Enable and clock signals control the exchange data between the two devices. If the external device is a talk-only device, the SPIM output port can be ignored and used for other purposes. Figure 10-1 is a block diagram of the SPIM.

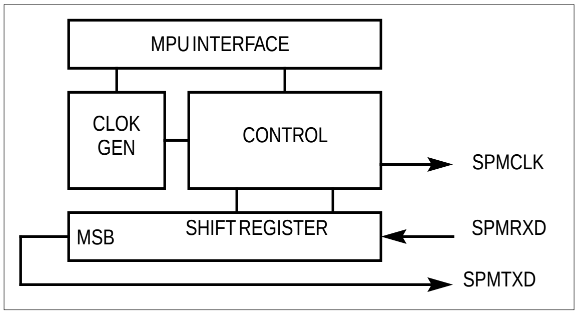


Figure 10-1. SPIM Block Diagram

10.2 OPERATION

10.2.1 Operation within SPIM module

To perform a serial data transfer, follow the sequence below:

- Set

  —Data rate bits 15—13
  —SPMEN (enable) bit 9
  —IRQEN (interrupt enable) bit 6
  —PHA (phase control) bit 5
  —POL (polarity control) bit 4
  —BIT count (data burst length) bits 3—0

- Load data
- Set XCH bit 8

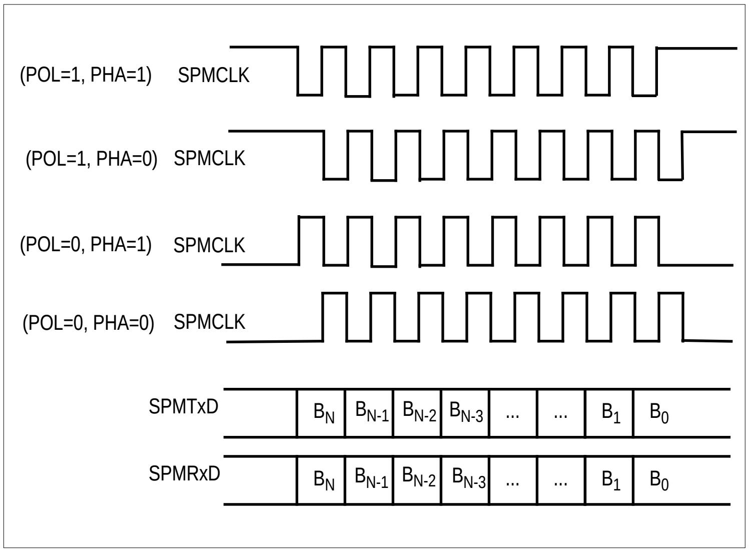

(POL=1, PHA=1) SPMCLK
(POL=1, PHA=0) SPMCLK
(POL=0, PHA=1) SPMCLK
(POL=0, PHA=0) SPMCLK
SPMTxD
B_N B_{N-1} B_{N-2} B_{N-3} ... ... B_1 B_0
SPMRxD
B_N B_{N-1} B_{N-2} B_{N-3} ... ... B_1 B_0

Figure 10-2. Master SPI Operation

- Wait for interrupt or poll SPMIRQ bit

For systems that need more than 16 clocks to transfer data, the SPMEN bit can remain asserted between exchanges. The enable signal needed in some SPI slave devices should be provided by an I/O port bit.

10.2.2 Phase/Polarity Configurations

The SPIM transfers data in and out of the shift register with the SPICLK. Data is clocked using any one of the variations of clock phase and clock polarity. The clocked transfer may be programmed in phase and in polarity (Figure 10-2). In **phase 0** operation, output data changes on falling clock edges, while input data is shifted on rising edges. In a **phase 1** operation, output data changes on rising edges of the clock and is shifted on falling edges. **Polarity = 1** inverts the data-clock relationships. This flexibility allows operation with most serial peripheral devices on the market.

10.3 SIGNAL DESCRIPTIONS

The following signals are multiplexed with other signals in port K. Refer to **Section 7.1.10** for more information.

SPMTxD Transmit Data

This pin is the shift-register output. A new data bit is presented on each rising edge of the SPMCLK in normal mode or on each falling edge of SPMCLK in inverted mode.

SPMRxD Receive Data

This pin is the shift-register input. A new bit is shifted in on each falling edge of SPMCLK while in normal mode or on each rising edge of SPMCLK in inverted mode.

SPMCLK Shift Clock

This pin is the clock output. When the SPIM is enabled, a selectable number of clock pulses is issued. While POL = 0, this signal is low while the SPIM is idle. When POL = 1, this signal is high during idle.

10.4 SPIM REGISTERS

These registers control the SPIM operation and report its status. The data register exchanges data with external slave devices. After reset, all bits are set to $0000.

10.4.1 SPIM Control/Status Register

This register controls the SPIM operation and reports its status.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA RATE			RESVD	RESVD	RESVD	SPMEN	XCH	SPMIRQ	IRQEN	PHA	POL	BIT COUNT			

Address: (FF)FFF802 Reset Value: $0000

Figure 10-3. SPIM Control/Status Register

DATA RATE

These bits select the baud rate of the SPMCLK based of divisions of the system clock. The master clock for the SPIM is SYSCLK. The bits are encoded as:

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

SPMEN

This bit enables the SPIM. The enable should be asserted before initiating an exchange and should be negated after the exchange is complete.

0 = SPI master disable
1 = SPI master enable

XCH

This bit triggers the state machine to generate (n= clock count) clocks at the selected bit rate. After the n-bit transfer, new data may be loaded and another exchange initiated. At least 2 SPI clocks should elapse before re-enabling this bit. This bit clears automatically.

1 = Initiate exchange
0 = SPI is idle or exchange in progress

NOTE

To ensure a complete exchange, users should check the SPIMIRQ bit rather than XCH bit. The IRQEN should be on. Users not wanting to receive interrupt upon completion of exchange can disable the incoming SPIM interrupt by masking it in the IMR in the interrupt controller.

SPIMIRQ

An interrupt is asserted at the end of an exchange (assuming IRQEN is enabled). This bit is asserted until users clear it by writing a 0. Users can write these bits to generate an IRQ on demand.

0 = No interrupt posted
1 = Interrupt posted

IRQEN

This bit will enable the SPIM interrupt. This bit is cleared to 0 on reset and must be enabled by software.

0 = Interrupts disabled
1 = Interrupts enabled

PHA

This bit controls the SPMCLK phase shift.

0 = Normal phase
1 = Shift advance to opposite phase

POL

This bit controls the SPMCLK polarity.

0 = Active-high polarity (0=idle)
1 = Inverted polarity (1= idle)

CLOCK COUNT

These bits select the transfer length (up to 16 bits can be transferred).

0000 = 1 bit transfer
1111 = 16 bit transfer

10.4.2 SPIM Data Register

This register exchanges data with external slave devices.

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

Address: (FF)FFF800 Reset Value: $0000

Figure 10-4. SPIM Data Register

DATA

These are the data bits to be exchanged with the external device. The data must be loaded before the XCH bit is set. At the end of the exchange, data from the peripheral is present in this register. These bits contain unknown data if they are read while the XCH bit is set. A write to these bits will be ignored while the XCH bit is set. As data is shifted MSB first, outgoing data is automatically MSB-justified. For example, if the exchange length is 10 bits, the MSB of the outgoing data is bit 9. The first bit presented to the external slave device will be bit 9, followed by the remaining 9 bits.

NOTE

Users should reload the data every time for each transfer before setting the XCH bit or the enable bit.

SECTION 11
PULSE-WIDTH MODULATOR

The pulse-width modulator (PWM) provides high-quality sound generation and accurate motor control. This section describes the PWM block.

11.1 OVERVIEW

The PWM is a simple free-running counter with two “compare” registers that each perform a particular task when they match the count value. The period comparator sets the output pin and the free-running counter resets when its value matches the period value. The width comparator resets the output pin when the counter value matches. With a suitable low-pass filter, the PWM can be used as a digital-to-analog converter.

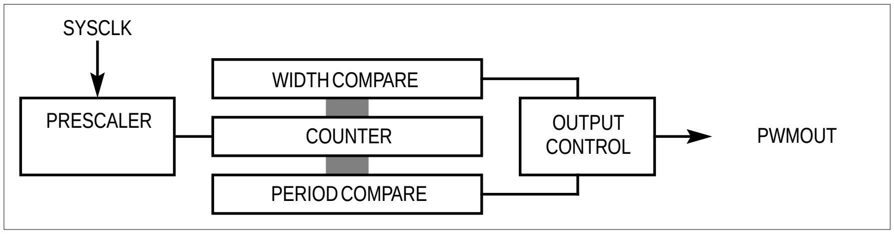


Figure 11-1. PWM Block Diagram

The output pin feeds a stream of sample values to the PWM into the width register and provides a low-pass filter on the output. It also provides a high-quality, digitally generated sound source. The selected period determines the reconstruction rate. Typically for voice quality, the rate will be between 6 kHz and 8 kHz. The following figure relates the pulse stream to the filtered audio output.

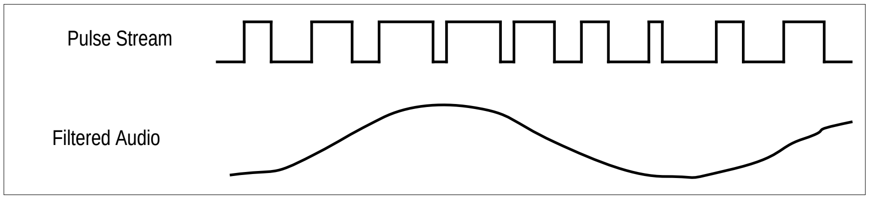


Figure 11-2. PWM Generating Audio

The width and period registers are double-buffered so that a new value can be loaded for the next cycle without disturbing the current cycle. At the beginning of each period, the contents of the buffer registers are loaded into the comparator for the next cycle. Sampled audio can be recreated by feeding a new sample value into the width register on each interrupt.

The prescaler provides operating flexibility. Figure 11-3 illustrates its functionality. The prescaler contains a variable divider that can divide the incoming clock by certain values between 2 and 256.

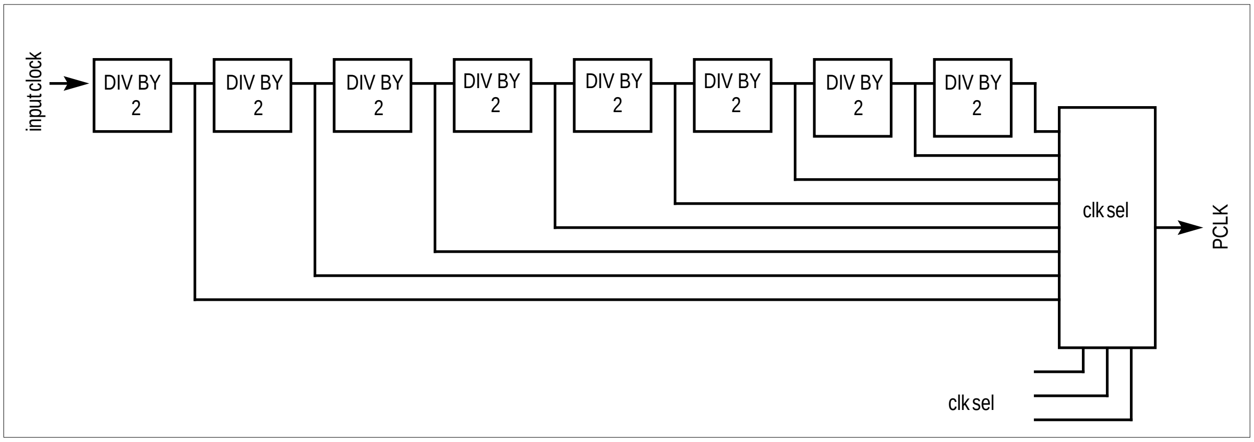


Figure 11-3. PWM Prescaler

11.2 PROGRAMMER'S MODEL

This section describes the PWM module registers and control bits.

11.2.1 PWM Control Register

This register controls the overall PWM operation. Output pin status is also accessible.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM IRQ	IRQEN	UNUSED					LOAD	PIN	0	POL	PWMEN	0	CLKSEL		

ADDRESS: $(FF)FFF500 Reset Value: $0000

Figure 11-4. PWM Control Register

PWMIRQ

This bit indicates that a period-compare posted an interrupt. Users can set this bit to immediately post a PWM interrupt for debugging purposes. This bit automatically clears itself after it is read while set, 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.

0 = No PWM period rollover
1 = PWM period rolled over

IRQEN

This bit controls the PWM interrupt. While this bit is low, the interrupt is disabled.

0 = PWM interrupt disabled
1 = PWM interrupt enabled

LOAD

This bit forces a new period. It loads the period and width registers and automatically clears itself after the load has been performed. For slow PCLK periods, the actual load may occur some time after the MPU writes this bit as the load occurs on the next rising PCLK edge.

PIN

This bit indicates the current status of the PWM output pin and can change immediately after it is read, depending on the current state of the pin.

0 = PWM output is low
1 = PWM output is high

POL

This bit controls the PWM output pin polarity. Normally, the output pin is set high at period boundaries and goes low when a width-compare event occurs.

0 = Normal polarity
1 = Inverted polarity

PWMEN

This bit enables the PWM. While disabled, the PWM is in low-power mode and the prescaler does not count. The output pin is forced to 1 or 0 depending on the setting of the POL bit.

0 = PWM disabled

- The clock prescaler is reset and frozen.
- The counter is reset to 0001 and frozen.
- The contents of the width and period registers are loaded into the comparators.
- The comparators are disabled.

Disabling the PWM may cause a ‘‘glitch’’ on the output, depending on the current state of the counter.

1 = PWM enabled

When this bit is set high, the PWM is enabled and begins a new period. The following actions occur:

- The output pin changes state to start a new period.
- The clock prescaler is released and begins counting.
- The counter begins counting.
- The comparators are enabled.
- The IRQ bit is set indicating the start of a new period if IRQEN is set.

CLKSEL

These bits select the output of the divider chain. The codings are

000 = Divide by 2   001 = Divide by 4   010 = Divide by 8
011 = Divide by 16   100 = Divide by 32   101 = Divide by 64
110 = Divide by 128   111 = Divide by 256

11.2.2 Period Register

This register controls the PWM period. When the counter value matches the value in this register, an interrupt is posted and the counter is reset to start another period.

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

ADDRESS: $(FF)FFF502   Reset Value: 0000

Figure 11-5. Period Register

PERIOD

This is the value that resets the counter. There is one special case: when this register is $00, the output is never set high (0% duty cycle).

11.2.3 Width Register

This register controls the pulse width. When the counter matches the value in this register, the output is reset for the duration of the period. Note that if the value in this register is higher than the period register, the output will never be reset, resulting in a 100% duty cycle.

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

ADDRESS: $(FF)FFF504   Reset Value: 0000

Figure 11-6. Width Register

WIDTH

When the counter reaches the value in this register, the output is reset.

11.2.4 Counter

This read-only register is the current count value and can be read at any time without disturbing the counter.

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

ADDRESS: $(FF)FFF506   Reset Value: 0000

Figure 11-7. Counter Register

COUNT

This is the current count value.

SECTION 12
PIN ASSIGNMENT

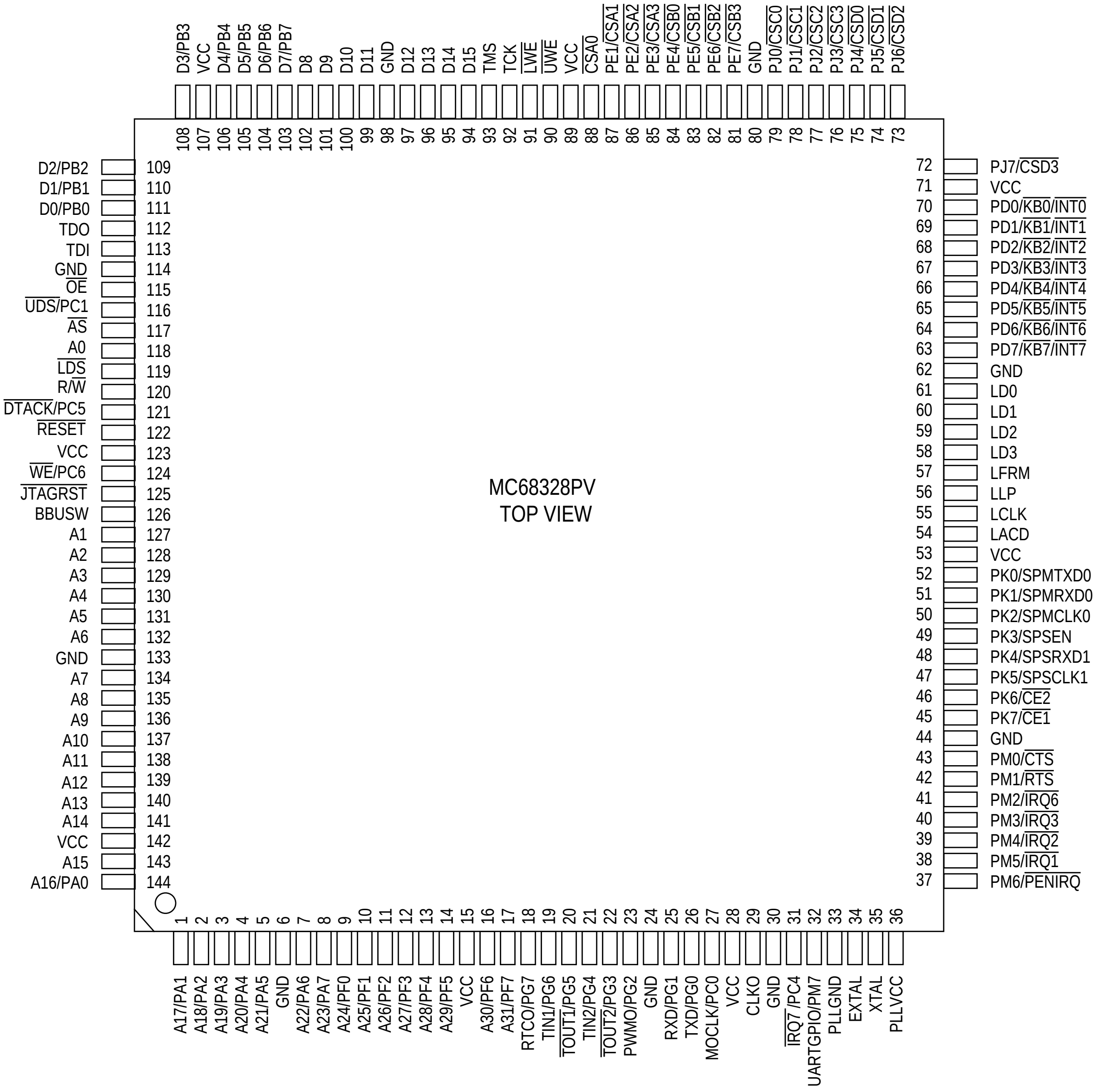
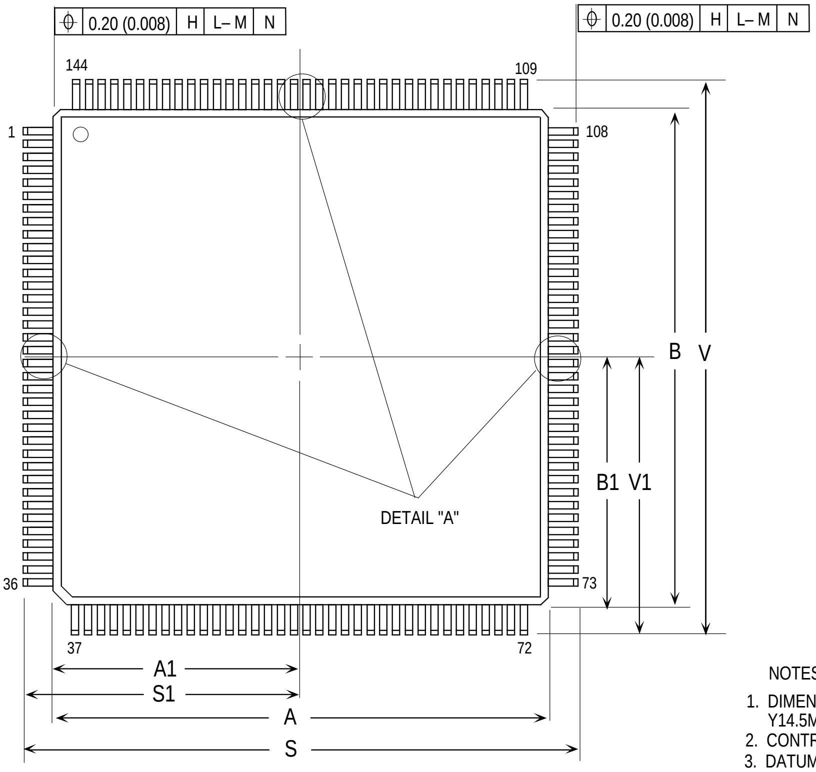
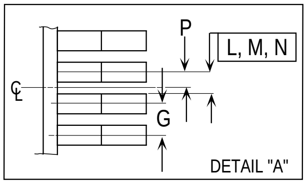
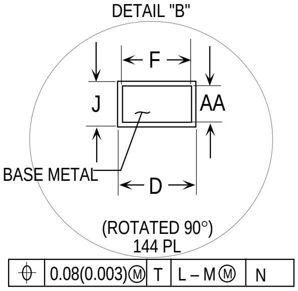


Figure 12-1. MC68328 144-Lead Plastic Thin-Quad Flat Pack Pin Assignment

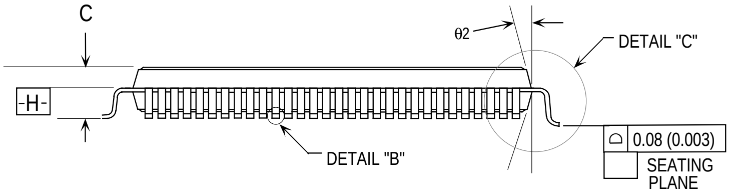


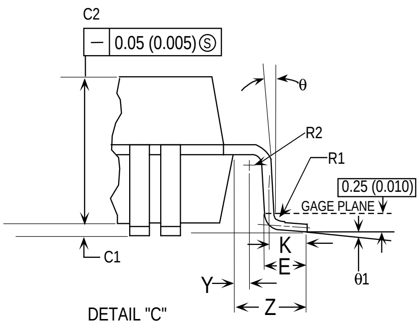




NOTES:

1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DATUM PLANE **-H-** IS LOCATED AT BOTTOM OF LEAD AND COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. DATUMS **-L-**, **-M-**, AND **-N-** TO BE DETERMINED AT DATUM PLANE **-H-**.
5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE **-T-**.
6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLEPROTRUSION IS 0.25 (0.010 PER SIDE. DIMENSIONS A AND B DO NOT INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM LINE **-H-**.
7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLEDAMBAR PROTRUSION SHALL NOT CAUSE THE D DIMENSION TO EXCEED 0.35 (0.014).





	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	20.00 BSC		0.790 BSC	
A1	10.00 BSC		0.394 BSC	
B	20.00 BSC		0.790 BSC	
B1	10.00 BSC		0.394 BSC	
C	1.40	1.60	0.055	0.063
C1	0.05	0.15	0.002	0.006
C2	1.35	1.45	0.053	0.057
D	0.17	0.27	0.007	0.011
E	0.45	0.75	0.018	0.030
F	0.17	0.23	0.007	0.009
G	0.50 BSC.		0.20 BSC.	
J	0.09	0.20	0.004	0.008
K	0.50 REF		0.020 REF	
P	0.25 BSC		0.010 BSC	
R1	0.13	0.20	0.005	0.008
R2	0.13	0.20	0.005	0.008
S	22.00 BSC		0.866 BSC	
S1	11.00 BSC		0.433 BSC	
V	22.00 BSC		0.866 BSC	
V1	11.00 BSC		0.433 BSC	
Y	0.25 REF		0.010 REF	
Z	1.00 REF		0.039 REF	
AA	0.09	0.16	0.004	0.006
θ	0°		0°	
θ1	0°	7°	0°	7°
θ2	11°	13°	11°	13°

Figure 13-2. MC68328 144-Lead Plastic Thin-Quad Flat Pack Specs

SECTION 13
ELECTRICAL CHARACTERISTICS—PRELIMINARY

This section provides **PRELIMINARY** information on the maximum ratings for the MC68328 processor.

13.1 MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	–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	T_{stg}	–55 to 150	°C

13.2 POWER CONSUMPTION

Characteristic	3.3 V		Unit
	Typical	Max	
Operating Current @16 MHz	15	30	mA
Standby Current	TBD	TBD	mA

13.3 AC ELECTRICAL SPECIFICATION DEFINITIONS

The AC specifications presented consist of output delays, input setup and hold times, and signal skew times. All signals are specified relative to an appropriate edge of the clock and possibly to one or more other signals.

Table 13-1. AC Electrical Specifications—Chip-Select Write Cycle Timing PRELIMINARY

Num	Characteristic	3.3 V		Unit
		Min	Max	
1	Addr Valid to CSx	40	—	ns
2	ASx to CSx	1	20	ns
3	CSx Width Asserted	140	—	ns
4	CLKOUT to ↑ Data	T/2 + 0	T/2 + 30	ns
5	UWE, LWE ↑ to Data ↑ Change	15	—	ns
6	UWE, LWE to CS	T/2- 10	T/2 + 5	ns

Note: T is system clock cycle time in nanoseconds.

13.4 AC ELECTRICAL SPECIFICATIONS—READ AND WRITE CYCLES

(Frequency = 0 to 16 MHz; GND = 0 V; $T_A = T_L$ to T_H; see Figure 13-1 and Figure 13-2)

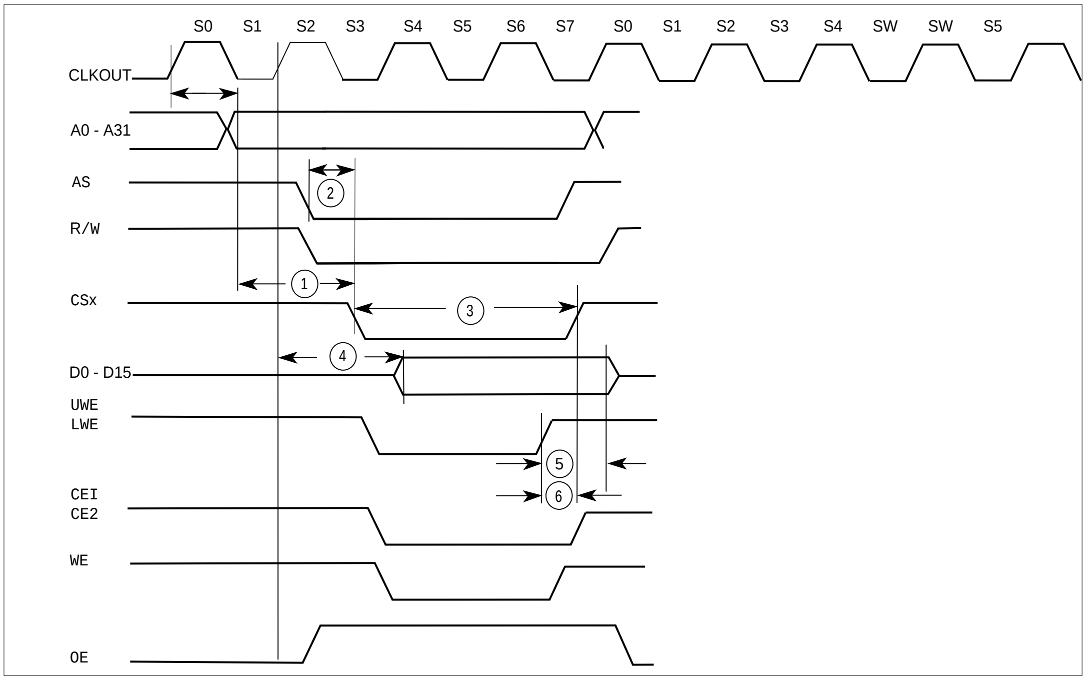


Figure 13-1. Chip-Select Write Cycle Timing (when CPU is Bus Master)—PRELIMINARY

Table 13-2. AC Electrical Specifications—Chip-Select Read Cycle Timing PRELIMINARY

Num	Characteristic	3.3 V		Unit
		Min	Max	
1	Addr Valid to CSx Asserted	40	—	ns
2	ASx to CSx	1	20	ns
3	CSx Width Asserted	140	—	ns
4	CLKOUT ↑ to Addr	0	30	ns
5	CLKOUT ↑ toR/W	0	30	ns
6	CLKOUT ↑ to OE	0	30	ns

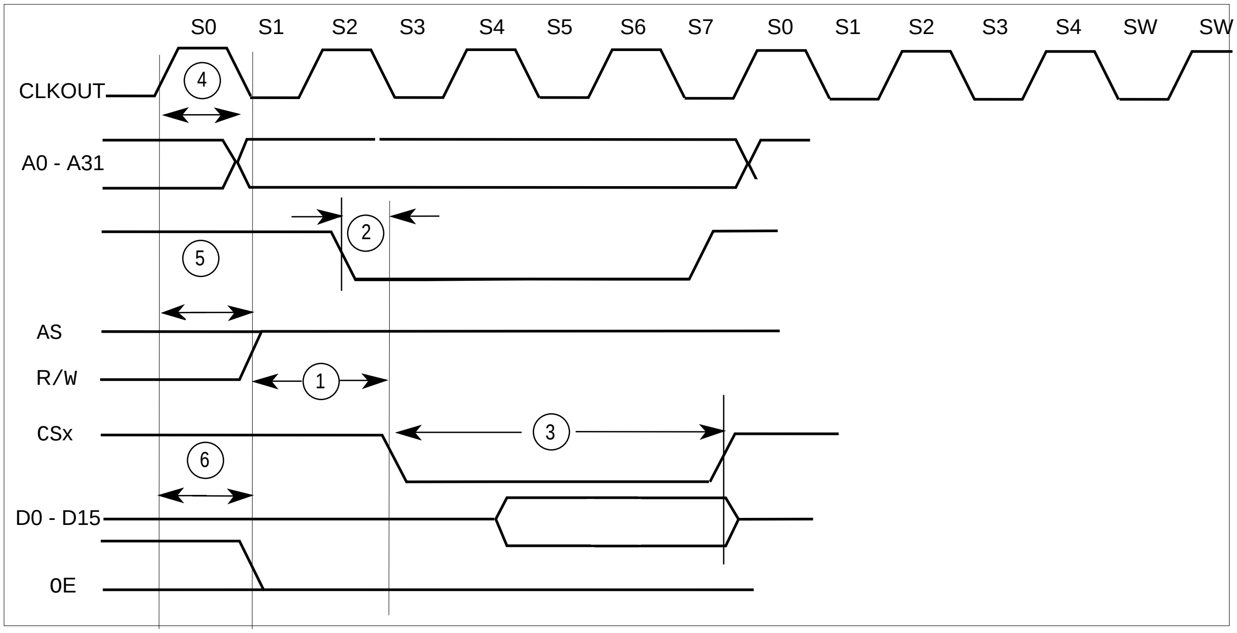


Figure 13-2. Chip-Select Read Cycle Timing (when the CPU is the Bus Master)—PRELIMINARY