SECTION 1
INTRODUCTION

The MC68328 DragonBall microprocessor is designed to save you time, power, cost, board space, pin count, and programming steps when designing your product. This functionality on a different microprocessor could require 20 separate components, each with 16-64 separate pins. Most of these components require interconnects, which may be duplicates. In fact, each of these connections could not only have a bad solder joint or misrouted trace, but may require another part to qualify, purchase, inventory, and maintain. These components take up valuable space on your PCB and they also consume more power. In addition, the signals between the CPU and a peripheral could be incompatible and may not run from the same clock, which could require time delays or other special design constraints.

All this combined makes the DragonBall the microprocessor of choice among many system designers. Its functionality and glue logic are all optimally connected, timed with the same clock, fully tested, and uniformly documented. Also, only the essential signals are brought out to the pins. The DragonBall's primary package consists of a surface-mount plastic TQFP designed to leave the smallest possible footprint on your board.

This manual will discuss the details of how to initialize, configure, and operate the DragonBall microprocessor. However, it assumes you have a basic knowledge of 68K architecture. If you are not familiar with 68K, you should get copies of the *M68000 User's Manual*, *M68000 Programmer's Reference Manual*, and *A Discussion of Interrupts for the MC68000* to use in conjunction with this manual. You can go to the Motorola website at http://www.motorola.com/pso to download these documents or you can contact your local sales office for printed versions.

1.1 FEATURES

The following list contains the main features of the DragonBall microprocessor:

- MC68EC000 Static Core Processor
  - ❏ 100% Compatibility with MC68000 and MC68EC000 Processors
  - ❏ 32-Bit Internal Address Bus
  - ❏ 24-Bit External Address Bus with Optional 32-Bit Address Bus for a 4G Address Space
  - ❏ 16-Bit On-Chip Data Bus for MC68EC000 Bus Operations
  - ❏ Static Design Allows Processor Clock to Be Stopped to Save Power
  - ❏ 2.7MIPS Performance Using a 16.67MHz Processor Clock
- Selectable Bus Sizing Support for Connecting to 8- and 16-Bit Devices

- System Integration Module Supports Glueless System Design
  - ❏ System Configuration and Programmable Address Mapping
  - ❏ Memory Interface for SRAM, EPROM, and Flash Memory
- Sixteen Programmable Peripheral Chip-Selects with Wait-State Generation Logic
  - ❏ PCMCIA 1.0 Support
- Interrupt Controller with 13 Flexible Inputs
  - ❏ Programmable Interrupt Vector Generator
- Maximum of 77 Individually Programmable Parallel Port Signals
- Dual-Channel 16-Bit General-Purpose Counter/Timer
  - ❏ Multimode Operation, Independent Capture and Compare Registers
  - ❏ Automatic Interrupt Generation
  - ❏ 60ns Resolution for a 16.67MHz System Clock
  - ❏ Hardware and Software Watchdog Timers
  - ❏ Separate Input and Output Pins for Capture and Compare
- Phase-Locked Loop and Power Management
  - ❏ 3.3V Operation
  - ❏ Fully Static HCMOS Technology
  - ❏ Programmable Clock Synthesizer for Full Frequency Control
  - ❏ Doze Mode Capability
  - ❏ Low-Power Mode Control
  - ❏ CPU Can Be Shut Down in Doze Mode
  - ❏ Sleep Mode Can Be Entered By Shutting Down the Phase-Locked Loop (PLL)
- LCD Controller
  - ❏ Software Programmable Screen Size to Drive Single Monochrome/STN Panels
  - ❏ Directly Drives Common LCD Drivers and Modules
  - ❏ Maximum of Four Grayscale Levels
  - ❏ System Memory Can Be Used as Display Memory
- UART
  - ❏ IrDA-Compliant Physical Layer Protocol Support
  - ❏ 8-Byte Transmit and Receive FIFOs
- Real-Time Clock
  - ❏ 24-Hour Time
  - ❏ Programmable Alarm
- Pulse-Width Modulation Output for Sound Generation
  - ❏ Programmable Frame Rate
  - ❏ 16-Bit Programmable
  - ❏ Motor Control Support
- Two Serial Peripheral Interface Ports

  - ❏ External POCSAG Decoder (Slave) Support
  - ❏ Digitizer For A/D Input or FLEX Pager (Master) Support
- IEEE 1149.1 Boundary Scan Test Access Port (JTAG)
- Operation From DC To 16.67MHz (Processor Clock)
- Operating Voltage of 3.3V ± 0.3V
- Compact 144-Lead Thin-Quad-Flat-Pack (TQFP) Packaging

SECTION 2
SIGNALS

This section contains a description of the MC68328 DragonBall microprocessor signals, as well as the pin assignment of the 144-lead plastic thin-quad flat package (TQFP).

The DragonBall uses a standard M68000 bus for communication between both on-chip and external peripherals with optional address extension to A31. This single continuous bus exists both on the and off the chip. Read accesses made by the core to internal memory-mapped registers of the device are invisible on the external bus. Write accesses made by the core to internal or external memory mapped locations are visible on the external bus.

2.1 SIGNAL DESCRIPTIONS

The DragonBall signals are grouped as shown in Table 2-1.

Table 2-1. Signal Groups

FUNCTIONAL GROUP	SIGNALS	NUMBER OF PINS
Clocks/PCIO	XTAL, EXTAL, CLKO, PC0/MOCLK	4
System Control/PCIO	$\overline{\text{RESET}}$, BBUSW	2
Address Bus, PAIO	PA7-PA0/A23-A16, A15–A0	24
Extended Address Bus/PFIO	PF7-PF0/A31-A24	8
Lower Data Bus/PBIO	PB7-PB0/D7–D0	8
Upper Data Bus	D15-D8	8
Bus Control/PCIO	PC1/$\overline{\text{UDS}}$, PC2/$\overline{\text{LDS}}$, PC5/$\overline{\text{DTACK}}$, $\overline{\text{AS}}$, R/$\overline{\text{W}}$, $\overline{\text{OE}}$, $\overline{\text{UWE}}$, $\overline{\text{LWE}}$	8
Interrupt Control/PDIO/Keyboard Interrupt	PD7-PD0/$\overline{\text{KB7}}$-$\overline{\text{KB0}}$/$\overline{\text{INT7}}$-$\overline{\text{INT0}}$	8
Interrupt Control/PMIO	$\overline{\text{PM5/IRQ1}}$, $\overline{\text{PM4/IRQ2}}$, $\overline{\text{PM3/IRQ3}}$, $\overline{\text{PM2/IRQ6}}$, $\overline{\text{PM6/PENIRQ}}$, $\overline{\text{PC4/IRQ7}}$	6
LCD Control	LACD, LCLK, LLP, LFRM, LD0-LD3	8
UART/PMIO/PGIO	PM7/GPIO, PG1/RXD, PG0/TXD, PM0/$\overline{\text{CTS}}$, PM1/$\overline{\text{RTS}}$	5
Timer/PGIO	PG4/TIN2, PG6/TIN1, PG3/TOUT2, PG5/TOUT1	4
RTC/PGIO	PG7/RTCO	1
PWM/PGIO	PG2/PWMO	1
Chip Select	$\overline{\text{CSA3}}$–$\overline{\text{CSA0}}$, $\overline{\text{CSB0}}$-$\overline{\text{CSB3}}$, $\overline{\text{CSC0}}$-$\overline{\text{CSC3}}$, $\overline{\text{CSD0}}$-$\overline{\text{CSD3}}$ PE7-PE0, PJ7-PJ0	16
SPI Master	PK0/SPMTXD0, PK1/SPMRXDI, PK2/ SPMCLKO	3
SPI Slave	PK3/SPSEN, PK4/SPSRXDI, PK5/SPSCLKI	3
PCMCIA 1.0	PC6/$\overline{\text{WE}}$, PK6/$\overline{\text{CE2}}$, PK7/$\overline{\text{CE1}}$	3
In-circuit Testing	ICHIZ	1
IEEE Testing	TMS, $\overline{\text{JTAGRST}}$, TDI, TDO, TCK	5
Power	V_{DD}, $PLLV_{DD}$	8
Ground	V_{SS}, $PLLV_{SS}$	10

Note: All pins except EXTAL support TTL levels. EXTAL, when used as an input clock, needs a CMOS level. To ensure proper low-power operation, all inputs should be driven CMOS level. Using a TTL level to drive those inputs could result in higher power consumption.

2.1.1 Power Pins

The MC68328 processor has 20 power supply pins. Users should be careful to reduce noise, potential crosstalk, and RF radiation from the output drivers. Inputs may be +5 V or +3.3V when VDD = +3.3V or +5V respectively without damaging the device.

- VDD (7)—7 power pins
- VSS (9)—9 ground pins
- PLLVDD (1)—1 power pin for the PLL
- PLLVSS (1)—1 ground pin for the PLL

2.1.2 4. Clock Pins

EXTAL—EXTERNAL CLOCK/CRYSTAL INPUT

This input provides 3 clock generation options: (1) low frequency crystal, (2) low frequency external clock, and (3) high frequency external clock. While PC0/MOCLK is low, the on-chip phase-locked loop is enabled, creating the high-speed system clock from a low frequency reference. EXTAL may be used (with XTAL) to connect an external crystal to the on-chip oscillator and clock generator. If an external clock instead of a crystal is used, the clock source should be connected to EXTAL, and XTAL left unconnected. The internal PLL generates the system clock at 16.58 MHz from a 32.768 kHz or 38.4 kHz source. When an external clock is used, it must provide a CMOS level at this input frequency.

While PC0/MOCLK is high, the PLL is disabled and the system clock must be connected to the EXTAL pin. If the real-time clock is used, 32.768 kHz or 38.4 kHz must be driven into PG7/RTCO.

XTAL—CRYSTAL OUTPUT

This output connects the on-chip oscillator output to an external crystal. If an external clock is used, XTAL should remain unconnected.

CLKO—CLOCK OUT

This output clock signal is derived from the on-chip clock oscillator and is internally connected to the clock output of the internal PLL. This signal is provided for external reference. The output can be disabled to reduce power consumption.

PC0/MOCLK—Clock Mode Select, Port C I/O

While this pin is high, the MC68328 processor is in the external clock mode and the on-chip PLL is disabled. The system clock must be driven into the EXTAL pin. While this pin is low, it enables the PLL. Either a 32.768 kHz or 38.4 kHz clock can be driven in to the EXTAL pin, or a crystal can be connected between EXTAL and XTAL to create an oscillator. PC0/MOCLK can be programmed as a general-purpose I/O while the internal PLL is enabled.

2.1.3 System Control Pins

$\overline{\text{RESET}}$

This active-low input signal causes the entire MC68328 processor (CPU and peripherals) to enter the reset state (cold reset). Users should drive this signal low for at least 100 msec at initial power-up to ensure that the crystal oscillator starts and stabilizes.

BBUSW—Boot Bus Width-Select

This input defines the data bus width for the boot chip-select, $\overline{\text{CSA0}}$. BBUSW = 0 means the boot chip-select addresses an 8-bit memory space. BBUSW=1 means the boot memory space is 16-bits wide. Users can create a mixed 8/16 bit memory system by programming the memory space widths in the various chip-select control registers.

2.1.4 Address Bus Pins

These are the address lines driven by the 68EC000 core or by the LCD controller for panel refresh DMA. The chip-select module can decode the entire 4 Gbyte address map. In many applications, only the lower portion of the address lines will be used, reserving any unused address pins for parallel I/O functions.

A15—A0

These address output lines are not multiplexed with any other I/O signals.

PA7-PA0/A16-A23

These address lines are multiplexed with I/O port A. When programmed as I/O ports, they serve as general-purpose I/O ports; otherwise, they are output-only address signals. These signals default to address lines at reset where the address lines are all zeroes. Users should note that there may be contention if any logic “1” levels are driving these pins during or after reset.

PF7-PF0/A31-A24

These bus pins are the extended address for 68EC000 core and are multiplexed with port F. In most systems, these lines are not used as addresses because most memory chips can be mapped into blocks of less than 16 Mbytes. These pins default to the port F I/O function after reset.

2.1.5 Data Bus Pins (D15–D0)

The flexible data bus interface design of the MC68328 processor allows users to program the lower byte of the data bus in an 8-bit-only system as general-purpose I/O signals.

D15–D8

The upper byte of the data bus is not multiplexed with any other signals. In pure 8-bit systems, this is the data bus. In mixed 8-/16-bit systems, 8-bit memory blocks or peripherals should be connected to this bus.

PB7–PB0/D7–D0

This bus is the lower data byte or general-purpose I/O. In pure 8-bit systems, this bus can serve as a general-purpose I/O. The WDTH8 bit in the system control register ($FFF000) should be set to one (1) by software before the port can be used. In 16-bit or mixed 8-/16-bit systems, these pins must function as the lower data byte.

2.1.6 Bus Control Pins

$\overline{AS}$—ADDRESS STROBE

This active-low output signal indicates that a valid address is present on the address bus. It is not asserted during LCD DMA accesses.

R/$\overline{W}$—READ/WRITE

This output signal defines the data bus transfer as a read or write cycle; read = 1; write = 0.

$\overline{PC1}$/$\overline{UDS}$—UPPER DATA STROBE/PORT C I/O

This pin can be programmed as $\overline{UDS}$ or as a general-purpose I/O. When used as upper data strobe ($\overline{UDS}$) output, this active-low signal is asserted when the internal EC000 core does a 16-bit word access or an even byte access. It is not asserted during LCD DMA accesses.

$\overline{PC2}$/$\overline{LDS}$—LOWER DATA STROBE/PORT C I/O

This pin can be programmed as $\overline{LDS}$or as a general-purpose I/O. When used as lower data strobe ($\overline{LDS}$) output, this active-low signal is asserted when the internal EC000 core does a 16-bit word access or an odd byte access. It is not asserted during LCD DMA accesses.

$\overline{LWE}$, $\overline{UWE}$ — LOWER BYTE WRITE-ENABLE AND UPPER BYTE WRITE-ENABLE

On a write cycle to a 16-bit port, these active-low output signals indicate when the upper or lower 8 bits of the data bus contain valid data. In 8-bit mode or when the BSW bit in the chip-select register is 0, use only the upper write-enable ($\overline{UWE}$) for write-enable control.

PC4/$\overline{IRQ7}$—LEVEL 7 INTERRUPT/PORT C I/O

When programmed as peripherals, this signal is an active-low input which, when asserted, will generate a level 7 interrupt to the CPU. When programmed as I/O, it becomes the PC 4 parallel I/O port.

PC5/$\overline{DTACK}$—DATA TRANSFER ACKNOWLEDGE, PC5

This pin can be programmed as parallel I/O PC5 or $\overline{DTACK}$. While programmed as $\overline{DTACK}$, this input signal indicates that the data transfer has been completed. $\overline{DTACK}$ is normally generated internally for all chip-selects. For systems that address spaces outside of the chip-select ranges, $\overline{DTACK}$ must be generated externally. PC5/$\overline{DTACK}$ must have an external pull-up resistor if programmed for the $\overline{DTACK}$ function.

$\overline{OE}$—OUTPUT-ENABLE

This active-low signal is asserted during a read cycle of the MC68328 processor, which enables the output of either ROM or SRAM. This signal also serves the PCMCIA 1.0 interface to indicate a read cycle.

2.1.7 Interrupt Control Pins

PD0-PD7/$\overline{KB0}$-$\overline{KB7}$/$\overline{INT0}$-$\overline{INT7}$—KEYBOARD AND GENERAL-PURPOSE INTERRUPT LINES

Users can program these signals as interrupt inputs or parallel I/O ports. For an interrupt port application, $\overline{INT0}$-$\overline{INT7}$ can be configured to perform keyboard interrupt functions. Keyboard interrupt pins $\overline{KB0}$-$\overline{KB7}$ are pulled high internally and connected to the rows of the keyboard matrix with the column driven low. When any one key of the row lines is pressed, an interrupt is generated to signal to the CPU to scan the keys. This feature, together with the pen interrupt, contributes a significant portion of the power management activities.

PM6/$\overline{PENIRQ}$—PEN INTERRUPT INPUT AND GENERAL-PURPOSE I/O

Users can program this pin as a general-purpose I/O PM6 or pen-interrupt input. When programmed as a pen-interrupt signal, this pin accepts an active low, level-triggered interrupt from the pen input device for a “pen-down” action.

PM5-PM2/$\overline{IRQ1}$, $\overline{IRQ2}$, $\overline{IRQ3}$, $\overline{IRQ6}$

These pins can be programmed to either parallel I/O PM2-PM5 or interrupt input. When they function as interrupt inputs, they can be programmed to be edge or level triggered with either high or low polarity. $\overline{IRQ6}$ generates a level 6 interrupt. $\overline{IRQ3}$, $\overline{IRQ2}$, and $\overline{IRQ1}$ generate level 3, 2 and 1 interrupts respectively.

2.1.8 Chip Select Pins

$\overline{CSA0}$—BOOT CHIP-SELECT

$\overline{CSA0}$ is the default chip-select after reset. It is set to 6 wait states and decodes all address ranges except internal register address space. It can be reprogrammed during the boot sequence to another address range and a different number of wait states.

$\overline{PE7}$-$\overline{PE1}$/$\overline{CSB3}$-$\overline{CSB0}$, $\overline{CSA3}$-$\overline{CSA1}$—CHIP-SELECT GROUP A AND B

These pins comprise the remainder of the Group A and Group B chip-selects and are individually programmable. Pins that are not needed as chip-selects can be programmed as general-purpose I/Os. By default after reset, $\overline{CSB3}$ is disabled and functions as a general-purpose input.

$\overline{PJ7}$-$\overline{PJ0}$/$\overline{CSD3}$-$\overline{CSD0}$/$\overline{CSC3}$-$\overline{CSC0}$—CHIP-SELECT GROUP C AND D

These pins comprise the Group C and Group D chip-selects and are individually programmable. Pins that are not needed as chip-selects can be programmed as general-purpose I/Os.

2.1.9 PCMCIA 1.0 PINS

PC6/$\overline{WE}$—WRITE ENABLE, PC6

This pin can be programmed as either PC6 parallel I/O or a write-enable signal for the PCMCIA 1.0 card interface. The MC68328 processor drives the active-low $\overline{WE}$ signal to indicate a memory-write transfer to the PCMCIA 1.0 card**.** When programmed as I/O, it serves as PC6.

PK7-PK6/$\overline{CE1}$-$\overline{CE2}$

These pins can be programmed as either parallel I/O port K7-6 or the PCMCIA 1.0 chip-enable signals. When programmed as the PCMCIA chip-enables, $\overline{CE1}$ and $\overline{CE2}$ are active-low, card-enable signals driven by the MC68328 processor; $\overline{CE1}$ enables even bytes; $\overline{CE2}$ enables odd bytes. $\overline{CE1}$ and $\overline{CE2}$ are decoded for assertion by $\overline{CSD3}$.

2.1.10 Master SPI Pins

PK0/SPMTXD—MASTER SPI TRANSMIT DATA, PORT K 0

This pin is the master SPI shift register output. By default after reset, this pin becomes general-purpose input, PK0.

PK1/SPMRXD—MASTER SPI RECEIVE DATA, PORT K 1

This pin is the input to the master SPI shift register. By default after reset, this pin becomes general-purpose input, PK1.

PK2/SPMCLK—MASTER SPI CLOCK, PORT K 2

This pin is the clock output when the SPIM is enabled. In polarity = 0 mode, this signal is low while the SPIM is idle. In polarity = 1 mode, this signal is high during idle. By default after reset, this pin becomes general-purpose input, PK2.

2.1.11 Slave SPI Pins

PK5/SPSCLK—SLAVE SPI CLOCK, PORT K 5

This pin is the slave SPI clock output. By default after reset, this pin becomes general-purpose input, PK5.

PK4/SPSRXD-SLAVE SPI RECEIVE DATA, PORT K 4

This pin is the slave SPI shift register input. By default after reset, this pin becomes general-purpose input, PK4.

PK3/SPSSEN-SLAVE SPI ENABLE, PORT K 3

This pin is the slave SPI enable. While this signal is active, 8 clocks shift data into the slave SPI. This bit is programmable to be active high or low. By default after reset, it becomes general-purpose input, PK3.

2.1.12 UART Pins

PG0/TXD—UART TRANSMIT DATA, PORT G 0

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 “zero” 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 IR LED or IR transceiver TXD signal. By default after reset, this pin becomes general-purpose input, PG0.

PG1/RXD—UART RECEIVE DATA, PORT G 1

This pin is the receiver serial input. While in normal operation, NRZ data is expected. While in infrared mode, a narrow pulse is expected for each “zero” bit received. An external IR transceiver RXD signal may be connected directly to this pin. RS-232 applications need an external RS-232 line-receiver to convert voltage levels. By default after reset, this pin becomes general-purpose input, PG1.

PM0/$\overline{\text{CTS}}$—CLEAR TO SEND, PORT M 0

This input controls the transmitter. Normally, the transmitter waits until this signal is active (low) before a character is transmitted. If the IGNORE CTS bit is set, the transmitter sends a character whenever a character is ready to transmit. This pin can then be used as a general-purpose input whose status is read in the CTS STATUS bit. This pin can post an interrupt on any transition of $\overline{\text{CTS}}$, if enabled. By default after reset, this pin becomes $\overline{\text{CTS}}$.

PM1/$\overline{\text{RTS}}$—REQUEST TO SEND, PORT M 1

This 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 be a general-purpose output controlled by the bit in the receiver register. When it is programmed as parallel I/O, it becomes PM1. By default after reset, this pin becomes general-purpose input, PM1.

PM7/GPIO—UART GENERAL PURPOSE I/O, PORT M 7

This pin provides several functions for the UART. It can provide a bit clock (input or output) and a master clock for the baud generator (input). By default after reset, this pin becomes UART GPIO.

2.1.13 Timer Pins

PG6/TIN1—TIMER 1 INPUT, PORT G 6

This bidirectional pin can be programmed as a clock input that causes events to occur in timer/counter channel 1; either causing a clock to the event counter or providing a trigger to the timer value capture logic. By default after reset, this pin becomes general-purpose input, PG6.

PG4/TIN2—TIMER 2 INPUT, PORT G 4

This bidirectional signal can be programmed as a clock input that causes events to occur in timer/counter channel 2; either causing a clock to the event counter or providing a trigger to the timer value capture logic. By default after reset, this pin becomes general-purpose input, PG4.

PG5/TOUT1 —TIMER 1 OUTPUT, PORT G 5

This bidirectional signal can be programmed to toggle or generate a pulse of one system clock duration when timer/counter channel 1 reaches a reference value. By default after reset, this pin becomes general-purpose input, PG5.

PG3/TOUT2 —TIMER 2 OUTPUT, PORT G 3

This bidirectional signal can be programmed to toggle or generate a pulse of one system clock duration when timer/counter channel 2 reaches a reference value. By default after reset, this pin becomes general-purpose input, PG3.

2.1.14 PWM Pin

PG2/PWMO—PULSE WIDTH MODULATOR OUTPUT, PORT G 2

This pin can serve as the PWM output signal. When it is PWMOUT, it produces synthesized sound, which can be connected to a filter and audio amplifier to generate melody and tone. By default after reset, this pin becomes general-purpose input, PG2.

2.1.15 Real-Time Clock Pins

PG7/RTCO—REAL-TIME CLOCK OUTPUT/INPUT, PORT G 7

While PC0/MOCLK is high, this pin is a dedicated input that provides the 32.768 kHz or 38.4 kHz clock to the real-time clock. While PC0/MOCLK is low, this pin can be programmed to output constant time tick pulses at the crystal frequency. By default after reset while PC0/MOCLK is low, this pin becomes general-purpose input, PG7.

2.1.16 LCD Controller Pins

LD3-LD0—LCD DATA BUS

This output bus transfers pixel data to the LCD panel for display. The pixel data is arranged to accommodate the programmable panel mode data width selection. Panel interfaces of one, two, or four bits are supported. Users can also program the output pixel data to be inverted for those LCD panels that require it.

The MC68328 LCD interface data bus uses LD0 to display pixel 0, 0. Some LCD panel manufacturers specify their LCD panel data bus where data bit 3 of the panel displays pixel 0,0. For these panels, the connections from the MC68328 LD bus to the LCD panel data bus are reversed in bit significance. Therefore, for these panels, connect LD0 of the MC68328 to LCD panel data bit 3, LD1 to LCD data 2, LD2 to LCD data 1, and LD3 to LCD data 0.

LFLM—FIRST LINE MARKER

This signal indicates the start of a new display frame. LFLM becomes active after the first line pulse of the frame and remains active until the next line pulse, at which point it de-asserts and remains inactive until the next frame. LFLM can be programmed to be an active-high or an active-low signal.

LP—LINE PULSE

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 remains asserted for 8 pixel clock periods. LP can be programmed to be either an active-high or an active-low signal.

LCLK—SHIFT CLOCK

This is the clock output to which the output data to the LCD panel is synchronized. LCLK can be programmed to be inverted.

LACD—ALTERNATE CRYSTAL DIRECTION

This output is toggled to alternate the crystal polarization on the panel and is used to protect the crystal from DC voltages. This signal can be programmed to toggle at a period from 1 to 16 frames.

2.1.17 JTAG Testing Pins

TCK—TEST CLOCK

This pin provides a test clock input for boundary scan test logic defined by the IEEE1149.1 standard. If JTAG is not used or during normal operation, TCK should be connected to an extrnal pullup resistor.

TMS—TEST MODE SELECT

This input controls test-mode operations for onboard test logic defined by the IEEE 1149.1 standard. If JTAG is not used, this pin should be connected to VDD or pulled up through an external pullup resistor.

TDI—TEST DATA IN

This input is used for serial test instructions and test data for internal test logic defined by the IEEE 1149.1 standard. If JTAG is not used, this pin should be connected to VDD or pulled up through an external pullup resistor.

TDO—TEST DATA OUT

This output is used for serial test instructions and test data for on-chip test logic defined by the IEEE 1149.1 standard. TDO may be left not connected or may drive the TDI pin of another device in a JTAG boundary scan chain.

$\overline{\text{JTAGRST}}$—JTAG RESET INPUT

This input is used for resetting the JTAG module for on-chip test logic defined by the IEEE 1149.1 standard for boundary scan. In normal operation, this pin should be connected to $\overline{\text{RESET}}$.

2.1.18 In-Circuit Test Pin

ICHIZ—IN-CIRCUIT HIGH IMPEDANCE

This input may be used as a means of isolating MC68328 signals during in-circuit testing. When $\overline{\text{ICHIZ}}$ is asserted, all of the MC68328 signal pins are high impedance. When $\overline{\text{HIZ}}$ is high, the MC68328 operates normally. $\overline{\text{HIZ}}$ may also be asserted to accomodate in-circuit-test programming of external memory components such as FLASH memories.

2.2 PIN ASSIGNMENT

The MC68328 pin assignment is shown in Figure 2-1. Mechanical specifications for the 144-pin Thin Quad Flat Pack (TQFP) are shown in Figure 2-2.

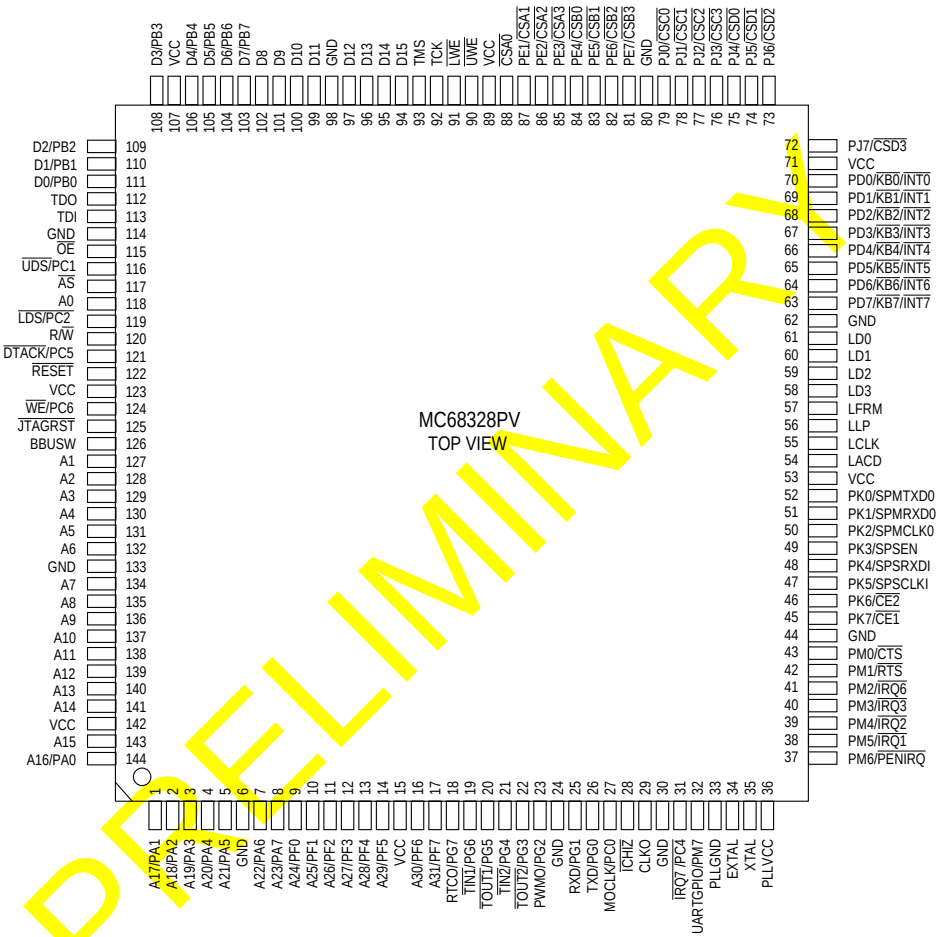
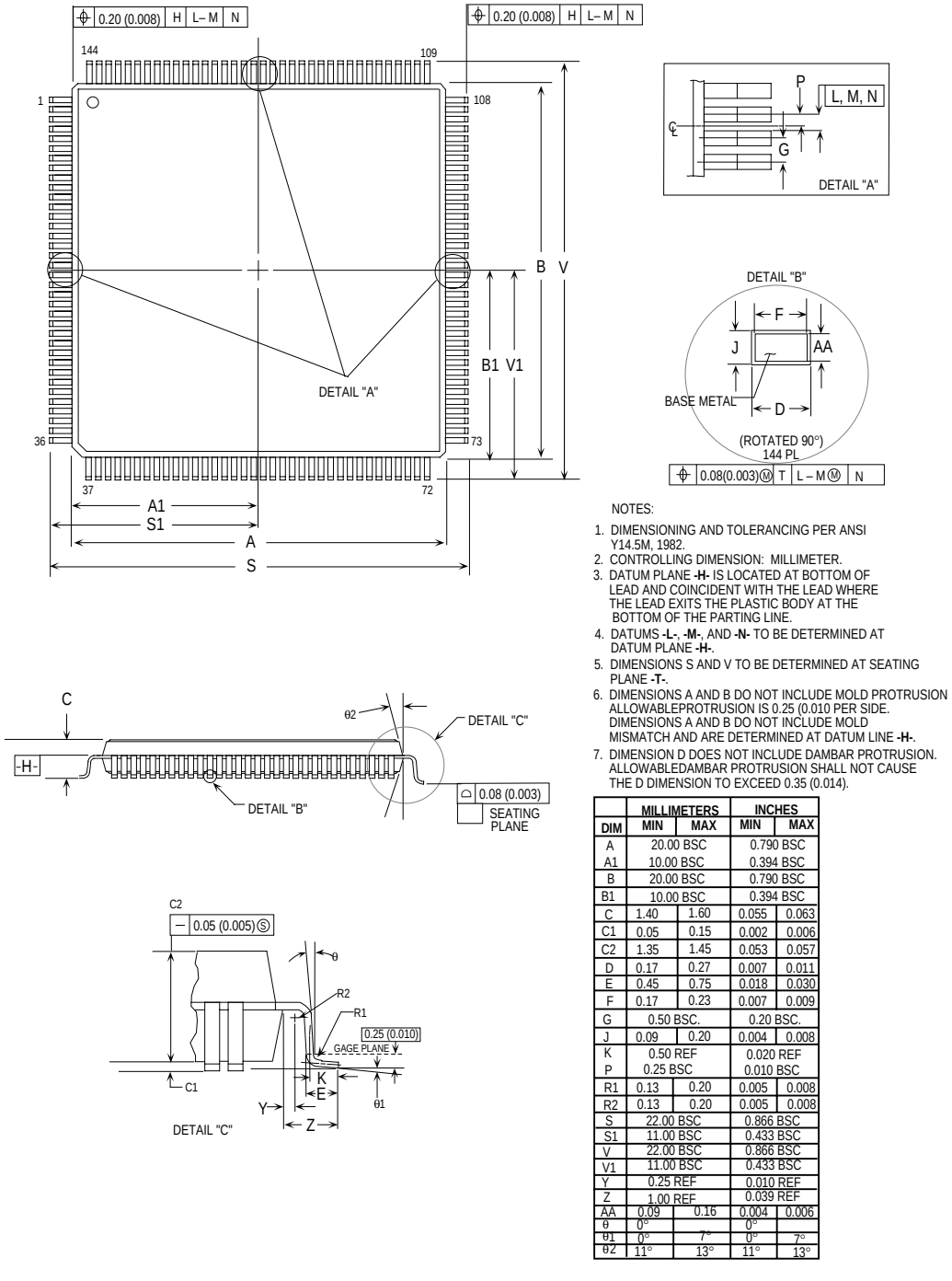


Figure 2-1. Pin Assignment



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 2-2. 144-Lead Plastic Thin-Quad Flat Package

SECTION 3
ARCHITECTURE

To improve total system throughput and reduce component count, board size, and cost of system implementation, the DragonBall combines a powerful MC68EC000 processor with intelligent peripheral modules and a typical system interface logic. The architecture of the DragonBall consists of the following blocks:

- EC000 core
- Chip-select logic and bus interface
- Clock synthesizer and power management
- Interrupt controller
- Parallel general-purpose I/O ports
- Timers
- Low-power stop logic
- LCD controller
- UART
- Real-time clock
- Pulse-width modulator
- Serial peripheral interface

This manual assumes you are familiar with 68K architecture. If you are not, get a copy of the *M68000 User's Manual* (part number M68000UM/AD) and *M68000 Programmer's Reference Manual* (part number M68000PM/AD) from your local Motorola sales office.

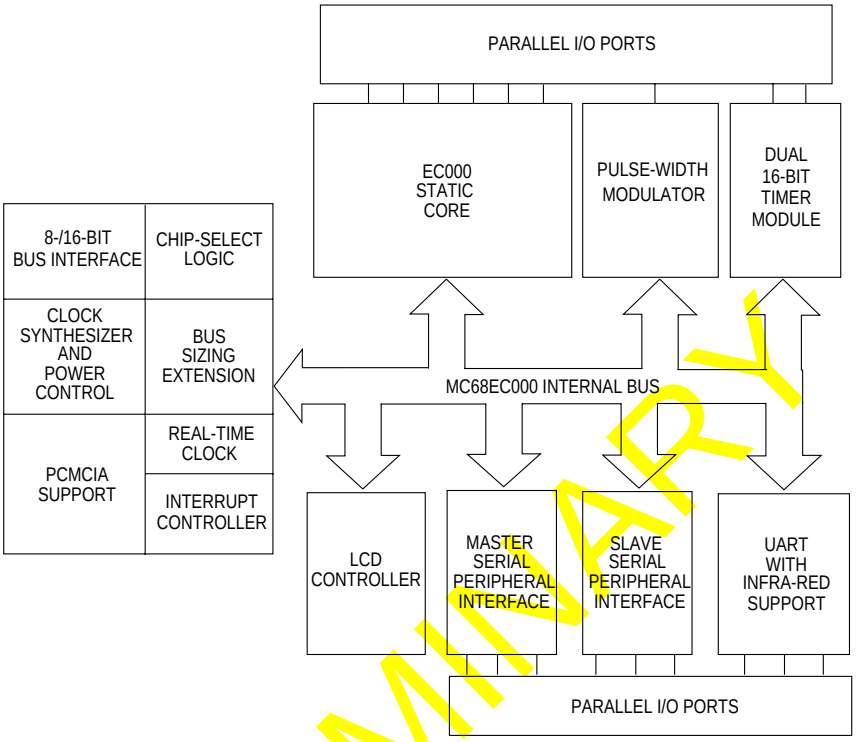


Figure 3-1. DragonBall Block Diagram

3.1 CORE

The MC68EC000 core in the DragonBall is an updated implementation of the M68000 32-bit microprocessor architecture. The main features of the core are:

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

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

3.1.1 Core Programming Model

The core has 32-bit registers and a 32-bit program counter (see Figure 3-2). The first eight registers (D7–D0) are data registers that are used for byte (8-bit), word (16-bit), and long-word (32-bit) operations. When using the data registers to manipulate data, they affect the status register (SR). The next seven registers (A6–A0) and the user stack pointer (USP) can function as software stack pointers and base address registers. These registers can be used for word and long-word operations, but they do not affect the status register. The D7-D0 and A6-A0 registers can be used as index registers.

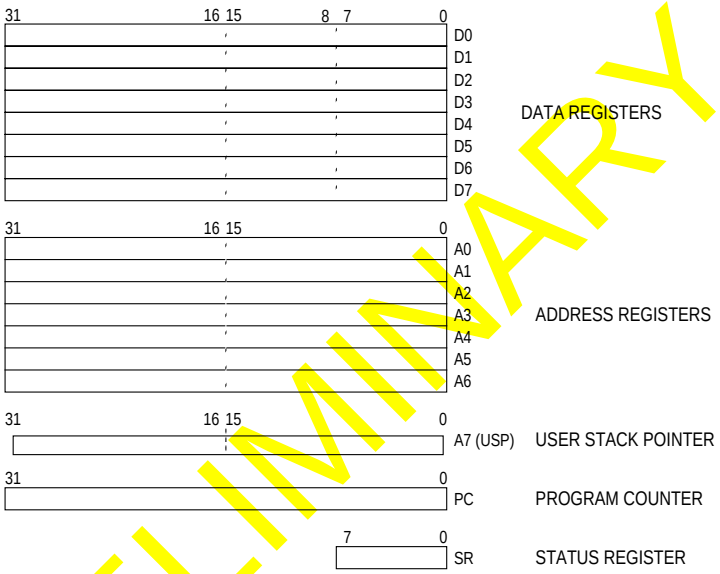


Figure 3-2. User Programming Model

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

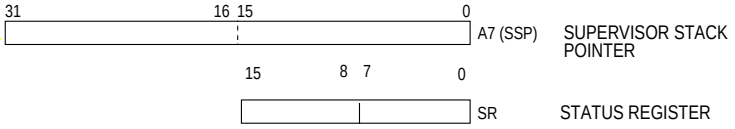


Figure 3-3. Supervisor Programming Model Supplement

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

3.1.2 Data and Address Mode Types

The core supports five types of data and six main types of address modes, as described in the following tables.

DATA TYPES	ADDRESS MODE TYPES
Bits	Register direct
Binary-coded decimal digits	Register indirect
Bytes	Absolute
Words	Program counter relative
Long words	Immediate
	Implied

Table 3-1. Address Modes

ADDRESS MODE	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

NOTE:
Dn = Data Register
An = Address Register
Xn = Address or Data Register Used as Index Register
SR = Status Register
PC = Program Counter
SP = Stack Pointer
USP = User Stack Pointer
<> = Effective Address
d_8 = 8-Bit Offset (Displacement)
d_{16} = 16-Bit Offset (Displacement)
#xxx = Immediate Data

3.1.3 Instruction Set

The EC000 core instruction set supports high-level languages that facilitate programming. Almost every instruction operates on bytes, words, and long-words, and most of them can use any of the 14 address modes. By combining instruction types, data types, and address modes, you can have access to over 1,000 instructions. These instructions include signed and unsigned, multiply and divide, quick arithmetic operations, binary-coded decimal (BCD) arithmetic, and expanded operations (through traps).

Table 3-2. Instruction Set

MNEMONIC	DESCRIPTION	MNEMONIC	DESCRIPTION
ABCD	Add decimal with extend	MOVEM	Move multiple registers
ADD	Add	MOVEP	Move peripheral data
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

Table 3-2. Instruction Set

MNEMONIC	DESCRIPTION	MNEMONIC	DESCRIPTION
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		

3.2 CHIP-SELECT LOGIC AND BUS INTERFACE

The system control register (SCR) allows you to configure the system status and control logic, register double-mapping, bus error generation, and module control register protection on the DragonBall.

The DragonBall contains 16 programmable general-purpose chip-select signals. Each chip-select block allows you to choose whether the chip-select allows read-only or both read and write accesses, whether a $\overline{\text{DTACK}}$ is automatically generated for the chip-select, the number of wait states (from zero to six) until the $\overline{\text{DTACK}}$ will be generated, and an 8- or 16-bit data bus.

The external bus interface handles the transfer of information between the internal core and the memory, peripherals, or other processing elements in the external address space. It consists of a 16-bit M68000 data bus interface for internal-only devices and an 8- or 16-bit (or mixed) data bus interface to external devices.

3.3 PLL CLOCK SYNTHESIZER AND POWER CONTROL

The clock synthesizer can operate with either an external crystal or an external oscillator using an internal phase-locked loop (PLL). An external clock can also be used to directly drive the clock signal at the operational frequency.

You can save power on the DragonBall by turning off peripherals that are not being used, reducing processor clock speed, or disabling the processor altogether. An interrupt at the interrupt controller logic that runs during low-power mode allows you to wake up from this mode. Programmable interrupt sources cause the system to wake up. On-chip peripherals can initiate a wake-up from doze mode and the external interrupts and real-time clock can wake up the core from sleep mode.

3.4 INTERRUPT CONTROLLER

The interrupt controller prioritizes internal and external interrupt requests and generates a vector number during the CPU interrupt-acknowledge cycle. Interrupt nesting is also provided so that an interrupt service routine of a lower priority interrupt may be suspended by a higher priority interrupt request. The on-chip interrupt controller has the following features:

- Prioritized interrupts
- Fully nested interrupt environment
- Programmable vector generation
- Unique vector number generated for each interrupt level
- Interrupt masking
- Wake-up interrupt masking

3.5 PARALLEL GENERAL-PURPOSE I/O PORTS

The DragonBall supports up to 77 general-purpose I/O ports that you can configure as general-purpose I/O pins or dedicated peripheral interface pins. Each 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.

3.6 TIMERS

The software watchdog timer protects against system failures by providing a way for you to 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 does reach its time-out value, the watchdog timer assumes that a system failure has occurred and the software watchdog logic resets or interrupts the core.

3.7 LCD CONTROLLER

The LCD controller is used to display data on an LCD module. It fetches display data from memory and provides control signals, frame line pulse, clocks, and data to the LCD module. It supports monochrome STN LCD modules with a maximum of four grayscale levels with frame rate control. System RAM can be used as display memory and DMA frees the CPU from panel refresh responsibilities.

3.8 UART

The UART communicates with external devices with a standard asynchronous protocol at baud rates from 300bps to 115.2kpbs. The UART provides the pulses to directly drive standard IrDA transceivers.

3.9 REAL-TIME CLOCK

A The real-time clock provides time-of-day with one-second resolution. It uses the crystal (either 32.76 or 38.4kHz) as a clock source to keep proper time. It keeps time as long as power is applied to the chip, which can be in sleep or doze mode.

3.10 PROGRAMMER'S MEMORY MAP

The memory map is a guide to all on-chip resources. Use the following table as a guide when configuring your chip. The base address used in the table is 0xFFFFF000 and 0xFFF000 from reset. If a double-mapped bit is cleared in the system control register, then the base address is 0xFFFFF000. Unpredictable results occur if you write to any 4K register space not documented in the table.

Table 3-3. Programmer's Memory Map

ADDRESS	NAME	WIDTH	DESCRIPTION	RESET VALUE(HEX)	PAGE #
Base+0x000	SCR	8	System Control Register	0x0C	System
Base+0x004	MRR	32	Mask Revision Register	(See note)	System
Base+0x100	GRPBASEA	16	Chip Select Group A Base Register	0x0000	CS
Base+0x102	GRPBASEB	16	Chip Select Group B Base Register	0x0000	CS
Base+0x104	GRPBASEC	16	Chip Select Group C Base Register	0x0000	CS
Base+0x106	GRPBASED	16	Chip Select Group D Base Register	0x0000	CS
Base+0x108	GRPMASKA	16	Chip Select Group A Mask Register	0x0000	CS
Base+0x10A	GRPMASKB	16	Chip Select Group B Mask Register	0x0000	CS
Base+0x10C	GRPMASKC	16	Chip Select Group C Mask Register	0x0000	CS
Base+0x10E	GRPMASKD	16	Chip Select Group D Mask Register	0x0000	CS
Base+0x110	CSA0	32	Group A Chip Select 0 Register	0x00010006	CS
Base+0x114	CSA1	32	Group A Chip Select 1 Register	0x00010006	CS
Base+0x118	CSA2	32	Group A Chip Select 2 Register	0x00010006	CS
Base+0x11C	CSA3	32	Group A Chip Select 3 Register	0x00010006	CS
Base+0x120	CSB0	32	Group B Chip Select 0 Register	0x00010006	CS
Base+0x124	CSB1	32	Group B Chip Select 1 Register	0x00010006	CS
Base+0x128	CSB2	32	Group B Chip Select 2 Register	0x00010006	CS
Base+0x12C	CSB3	32	Group B Chip Select 3 Register	0x00010006	CS
Base+0x130	CSC0	32	Group C Chip Select 0 Register	0x00010006	CS
Base+0x134	CSC1	32	Group C Chip Select 1 Register	0x00010006	CS
Base+0x138	CSC2	32	Group C Chip Select 2 Register	0x00010006	CS
Base+0x13C	CSC3	32	Group C Chip Select 3 Register	0x00010006	CS
Base+0x140	CSD0	32	Group D Chip Select 0 Register	0x00010006	CS
Base+0x144	CSD1	32	Group D Chip Select 1 Register	0x00010006	CS
Base+0x148	CSD2	32	Group D Chip Select 2 Register	0x00010006	CS
Base+0x14C	CSD3	32	Group D Chip Select 3 Register	0x00010006	CS
Base+0x200	PLLCR	16	PLL Control Register	0x2400	PLL
Base+0x202	PLLFSR	16	PLL Frequency Select Register	0x0123	PLL

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

ADDRESS	NAME	WIDTH	DESCRIPTION	RESET VALUE(HEX)	PAGE #
Base+0x207	PCTLR	8	Power Control Register	0x1F	PCTL
Base+0x300	IVR	8	Interrupt Vector Register	0x00	INTR
Base+0x302	ICR	16	Interrupt Control Register	0x0000	INTR
Base+0x304	IMR	32	Interrupt Mask Register	0x00FFFFFF	INTR
Base+0x308	IWR	32	Interrupt Wakeup Enable Register	0x00FFFFFF	INTR
Base+0x30C	ISR	32	Interrupt Status Register	0x00000000	INTR
Base+0x310	IPR	32	Interrupt Pending Register	—	INTR
Base+0x400	PADIR	8	Port A Direction Register	0x00	PIO
Base+0x401	PADATA	8	Port A Data Register	0x00	PIO
Base+0x403	PASEL	8	Port A Select Register	0x00	PIO
Base+0x408	PBDIR	8	Port B Direction Register	0x00	PIO
Base+0x409	PBDATA	8	Port B Data Register	0x00	PIO
Base+0x40B	PBSEL	8	Port B Select Register	0x00	PIO
Base+0x410	PCDIR	8	Port C Direction Register	0x00	PIO
Base+0x411	PCDATA	8	Port C Data Register	0x00	PIO
Base+0x413	PCSEL	8	Port C Select Register	0x00	PIO
Base+0x418	PDDIR	8	Port D Direction Register	0x00	PIO
Base+0x419	PDDATA	8	Port D Data Register	0x00	PIO
Base+0x41A	PDPUEN	8	Port D Pullup Enable Register	0xFF	PIO
Base+0x41C	PDPOL	8	Port D Polarity Register	0x00	PIO
Base+0x41D	PDIRQEN	8	Port D IRQ Enable Register	0x00	PIO
Base+0x41F	PDIRQEDGE	8	Port D IRQ Edge Register	0x00	PIO
Base+0x420	PEDIR	8	Port E Direction Register	0x00	PIO
Base+0x421	PEDATA	8	Port E Data Register	0x00	PIO
Base+0x422	PEPUEN	8	Port E Pullup Enable Register	0x80	PIO
Base+0x423	PESEL	8	Port E Select Register	0x80	PIO
Base+0x428	PFDIR	8	Port F Direction Register	0x00	PIO
Base+0x429	PFDATA	8	Port F Data Register	0x00	PIO
Base+0x42A	PFPUEN	8	Port F Pullup Enable Register	0xFF	PIO
Base+0x42B	PFSEL	8	Port F Select Register	0xFF	PIO
Base+0x430	PGDIR	8	Port G Direction Register	0x00	PIO
Base+0x431	PGDATA	8	Port G Data Register	0x00	PIO
Base+0x432	PGPUEN	8	Port G Pullup Enable Register	0xFF	PIO
Base+0x433	PGSEL	8	Port G Select Register	0xFF	PIO
Base+0x438	PJDIR	8	Port J Direction Register	0x00	PIO

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

ADDRESS	NAME	WIDTH	DESCRIPTION	RESET VALUE(HEX)	PAGE #
Base+0x439	PJDATA	8	Port J Data Register	0x00	PIO
Base+0x43B	PJSEL	8	Port J Select Register	0x00	PIO
Base+0x440	PKDIR	8	Port K Direction Register	0x00	PIO
Base+0x441	PKDATA	8	Port K Data Register	0x00	PIO
Base+0x442	PKPUEN	8	Port K Pullup Enable Register	0x3F	PIO
Base+0x443	PKSEL	8	Port K Select Register	0x3F	PIO
Base+0x448	PMDIR	8	Port M Direction Register	0x00	PIO
Base+0x449	PMDATA	8	Port M Data Register	0x00	PIO
Base+0x44A	PMPUEN	8	Port M Pullup Enable Register	0xFF	PIO
Base+0x44B	PMSEL	8	Port M Select Register	0x02	PIO
Base+0x500	PWMC	16	PWM Control Register	0x0000	PWM
Base+0x502	PWMP	16	PWM Period Register	0x0000	PWM
Base+0x504	PWMW	16	PWM Width Register	0x0000	PWM
Base+0x506	PWMCNT	16	PWM Counter	0x0000	PWM
Base+0x600	TCTL1	16	Timer Unit 1 Control Register	0x0000	Timer
Base+0x602	TPRER1	16	Timer Unit 1 Prescalar Register	0x0000	Timer
Base+0x604	TCMP1	16	Timer Unit 1 Compare Register	0xFFFF	Timer
Base+0x606	TCR1	16	Timer Unit 1 Capture Register	0x0000	Timer
Base+0x608	TCN1	16	Timer Unit 1 Counter	0x0000	Timer
Base+0x60A	TSTAT1	16	Timer Unit 1 Status Register	0x0000	Timer
Base+0x60C	TCTL2	16	Timer Unit 2 Control Register	0x0000	Timer
Base+0x60E	TPREP2	16	Timer Unit 2 Prescaler Register	0x0000	Timer
Base+0x610	TCMP2	16	Timer Unit 2 Compare Register	0xFFFF	Timer
Base+0x612	TCR2	16	Timer Unit 2 Capture Register	0x0000	Timer
Base+0x614	TCN2	16	Timer Unit 2 Counter	0x0000	Timer
Base+0x616	TSTAT2	16	Timer Unit Status Register	0x0000	Timer
Base+0x618	WCR	16	Watchdog Control Register	0x0001	WD
Base+0x61A	WRR	16	Watchdog Compare Register	0xFFFF	WD
Base+0x61C	WCN	16	Watchdog Counter	0x0000	WD
Base+0x700	SPISR	16	SPIS Register	0x0000	SPIS
Base+0x800	SPIMDATA	16	SPIM Data Register	0x0000	SPIM
Base+0x802	SPIMCONT	16	SPIM Control/Status Register	0x0000	SPIM
Base+0x900	USTCNT	16	UART Status/Control Register	0x0000	UART
Base+0x902	UBAUD	16	UART Baud Control Register	0x003F	UART
Base+0x904	URX	16	UART RX Register	0x0000	UART

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

ADDRESS	NAME	WIDTH	DESCRIPTION	RESET VALUE(HEX)	PAGE #
Base+0x906	UTX	16	UART TX Register	0x0000	UART
Base+0x908	UMISC	16	UART Misc Register	0x0000	UART
Base+0xA00	LSSA	32	Screen Starting Address Register	0x00000000	LCDC
Base+0xA05	LVPW	8	Virtual Page Width Register	0xFF	LCDC
Base+0xA08	LXMAX	16	Screen Width Register	0x03FF	LCDC
Base+0xA0A	LYMAX	16	Screen Height Register	0x01FF	LCDC
Base+0xA18	LCXP	16	Cursor X Position	0x0000	LCDC
Base+0xA1A	LCYP	16	Cursor Y Position	0x0000	LCDC
Base+0xA1C	LCWCH	16	Cursor Width & Height Register	0x0101	LCDC
Base+0xA1F	LBLKC	8	Blink Control Register	0x7F	LCDC
Base+0xA20	LPICF	8	Panel Interface Config Register	0x00	LCDC
Base+0xA21	LPOLCF	8	Polarity Config Register	0x00	LCDC
Base+0xA23	LACDRC	8	ACD (M) Rate Control Register	0x00	LCDC
Base+0xA25	LPXCD	8	Pixel Clock Divider Register	0x00	LCDC
Base+0xA27	LCKCON	8	Clocking Control Register	0x40	LCDC
Base+0xA29	LLBAR	8	Last Buffer Address Register	0x3E	LCDC
Base+0xA2B	LOTCR	8	Octet Terminal Count Register	0x3F	LCDC
Base+0xA2D	LPOSR	8	Panning Offset Register	0x00	LCDC
Base+0xA31	LFRCM	8	Frame Rate Control Modulation Register	0xB9	LCDC
Base+0xA32	LGPMR	16	Gray Palette Mapping Register	0x1073	LCDC
Base+0xB00	HMSR	32	RTC Hours Minutes Seconds Register	0xXXXXXXXX	RTC
Base+0xB04	ALARM	32	RTC Alarm Register	0x00000000	RTC
Base+0xB0C	CTL	8	RTC Control Register	0xXX	RTC
Base+0xB0E	ISR	8	RTC Interrupt Status Register	0x00	RTC
Base+0xB10	IENR	8	RTC Interrupt Enable Register	0x00	RTC
Base+0xB12	STPWCH	8	Stopwatch Minutes	0x00	RTC

NOTE: The MRR

SECTION 4
SYSTEM CONTROL

The DragonBall microprocessor contains a system control register that enables the system software to customize the following functions:

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

4.1 OPERATION

The on-chip resources use a reserved 4,096-byte block of address space for their registers. This block is double-mapped to two locations—0xFFFFF000 (24-bit) and 0xFFF000 (32-bit)—at reset. The DMAP bit in the system control register disables double-mapping in a 32-bit system. If you clear this bit, the on-chip peripheral registers appear only at the top of the 4G address range starting at 0xFFFFF000.

The system control register allows you to control system operation functions like bus interface and hardware watchdog protection. It contains status bits that allow exception handler code to investigate the cause of exceptions and resets. The hardware watchdog (bus timeout monitor) and the software watchdog timer provide system protection. The hardware watchdog provides a bus monitor that causes a bus error when a bus cycle is not terminated by the $\overline{\text{DTACK}}$ signal before 128 clock cycles have elapsed.

The bus error timeout logic consists of a watchdog counter that, when enabled, begins to count clock cycles as the $\overline{\text{AS}}$ pin is asserted for internal or external bus accesses. The negation of $\overline{\text{AS}}$ normally terminates the count, but 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 system control register. The BETO bit in the system control register is set after a bus timeout, which could be caused by a write-protect violation.

The software watchdog timer resets the DragonBall if enabled and not cleared or disabled before reaching terminal count. The software watchdog timer is enabled at reset. For information about timer operation, see **Section 8.3 Software Watchdog Timer**.

4.1.1 System Control Register

The 8-bit read/write system control register (SCR) resides at 0xFFF000 or 0xFFFFF000 after reset. The SCR cannot be accessed in user data space if the SO bit is set to 1. Writing a 1 to the status bits in this register clears them, but writing a 0 has no effect.

SCR

BIT	7	6	5	4	3	2	1	0
FIELD	BETO	WPV	PRV	BETEN	SO	DMAP	RESERVED	WDTHB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	$0C							
ADDRESS	0x(FF)FFF000							

BETO—Bus Error Time-Out

This status bit indicates whether or not a bus error timer time-out has occurred. When a bus cycle is not terminated by the $\overline{\text{DTACK}}$ signal after 128 clock cycles have elapsed, the BETO bit is set. However, the BETEN bit must be set for a bus error timeout to occur. This bit is cleared by writing a 1 (writing a 0 has no effect).

0 = A bus error timer time-out did not occur.
1 = A bus error timer time-out occurs because an undecoded address space has been accessed or because a write-protect or privilege violation has occurred.

WPV—Write-Protect Violation

This status bit indicates that a write-protect violation has occurred. If a write-protect violation occurs and the BETEN bit is not set, the cycle will not terminate. The BETEN bit must be set for a bus error exception to occur during a write-protect violation. This bit is cleared by writing a 1 (writing a 0 has no effect).

0 = A write-protect violation did not occur.
1 = A write-protect violation has occurred.

PRV—Privilege Violation

This status bit indicates that If a privilege violation occurs and the BETEN bit is not set, the cycle will not terminate. The BETEN bit must be set for a bus error exception to occur during a privilege violation. This bit is cleared by writing a 1 (writing a 0 has no effect).

0 = A privilege violation did not occur.
1 = A privilege violation has occurred.

BETEN—Bus-Error Timeout Enable

This control bit enables the bus error timer.

0 = Disable the bus error timer.
1 = Enable the bus error timer.

SO—Supervisor Only

This control bit limits on-chip registers to supervisor accesses only.

0 = User and supervisor mode.
1 = Supervisor-only mode.

DMAP—Double Map

This control bit controls the double-mapping function.

0 = The on-chip registers are mapped at 0xFFFFF000–0xFFFFFFFF.
1 = The on-chip registers are mapped at 0xXXFFF000–0xXXFFFFFF.

Bit 1—Reserved

This bit is reserved and reads 0.

WDTH8—8-Bit Width Select

This control bit allows the D[7:0] pins to be used for port B input/output.

0 = Not an 8-bit system.
1 = 8-bit system.

SECTION 5
CHIP-SELECT LOGIC

The DragonBall microprocessor contains 16 general-purpose, programmable, chip-select signals, which are arranged in four groups of four. Among them, there are two special-purpose chip-select signals—CSA0 and CSD3. The $\overline{\text{CSA0}}$ signal is special in that it is also a boot device chip-select. From reset, all the addresses are mapped to $\overline{\text{CSA0}}$ until group-base address A is programmed and the V 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 $\overline{\text{CSD3}}$ signal is also special because it is associated with the PCMCIA support logic. When $\overline{\text{CSD3}}$ is asserted, the PCMCIA control signals are asserted. For each memory area, you can define an internally generated cycle-termination signal, called $\overline{\text{DTACK}}$, with a programmable number of wait states. This feature saves board space that would otherwise be used for cycle-termination logic.

The 16 general-purpose chip-selects allow different classes of devices and memory to be used in a system without external decode or wait-state generation logic. A typical configuration would be an 8-bit EPROM, a fast 16-bit SRAM, four simple I/O peripherals, and a nonvolatile flash memory. See **Section 16 Applications and Design Examples** for more information.

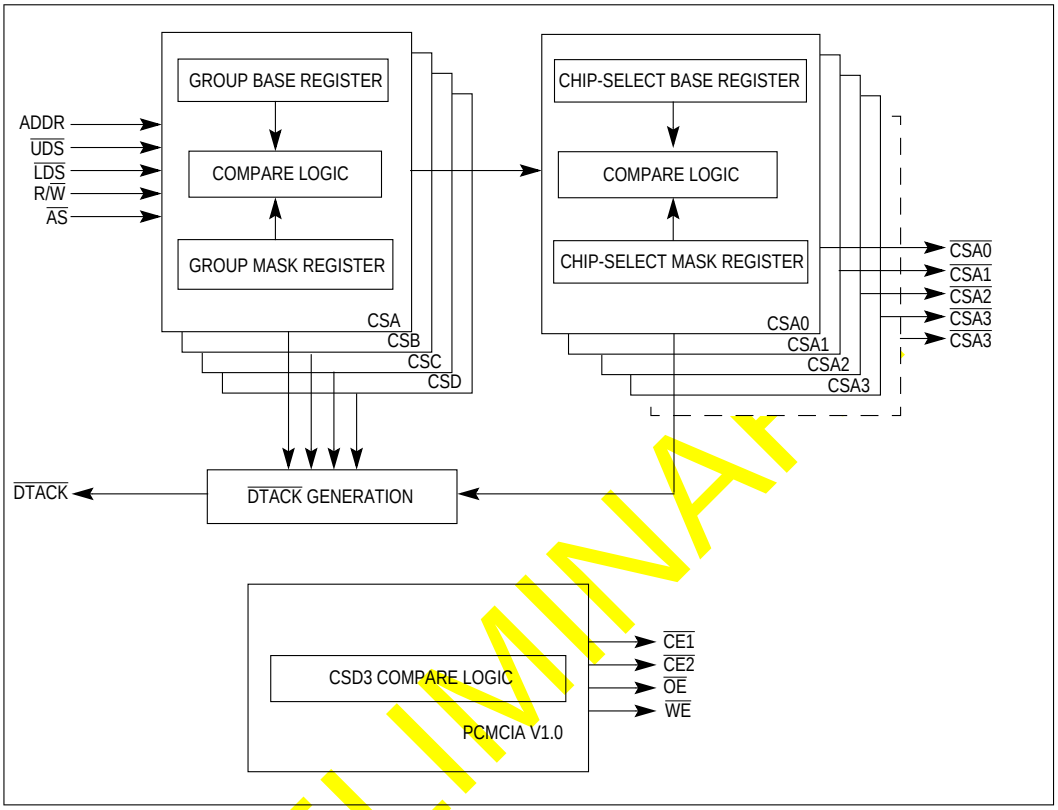


Figure 5-1. Chip-Select Operation

5.1 CHIP-SELECT OPERATION

A chip-select output signal is asserted when an address is matched and after the AS signal goes low. The base address and address mask registers are used in the compare logic to generate an address match. The byte size of the matching block must be a power of 2 and the base address must be an integer multiple of this size. Therefore, an 8K block size must begin on an 8K boundary and a 64K block size can only begin on a 64K boundary. Each chip-select is programmable and the registers have read/write capability so that the programmed values can be read back.

You can choose whether the chip-select allows read-only or read/write accesses, whether a DTACK signal is automatically generated for the chip-select, the number of wait states (from zero to six), and data bus size selection.

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

5.1.1 Programmable Data Bus Size

Each chip-select can be configured to address an 8- or 16-bit space. You can mix 16- and 8-bit contiguous address memory devices on a 16-bit data bus system. If the core performs a 16-bit data transfer in an 8-bit memory space, then two 8-bit cycles will occur. However, the address and data strobes remain asserted until the end of the second 8-bit cycle. In this case, only the external core data bus upper byte (D[15:8]) is used and 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 or written. For an external peripheral that only needs an 8-bit data bus interface and does not require contiguous address locations (unused bytes on empty addresses), use a chip-select configured to a 16-bit data bus width and connect to the D[7:0] pins. This balances the load of the two data bus halves in an 8-bit system. The internal data bus is16 bits wide. All internal registers can be read or written in one zero wait-state cycle.

Each chip-select defaults to a 16-bit data bus width. The BUSW field in the chip-select option registers enable 16- and 8-bit data bus widths for each of the 16 chip-select ranges. You can select the initial bus width for the boot chip-select by placing a logic 0 or 1 on the BBUSW pin at reset to specify the width of the data bus. This allows a boot EPROM of the data bus width to be used in any given system. All external accesses that do not match one of the chip-select address ranges are assumed to be a 16-bit device. That is just one access performed for a 16-bit transfer. It can also be a 8-bit port accessed every other byte.

The boot chip-select is initialized from reset to assert in response to any address except the on-chip register space (0xXXFFF000 to 0xXXFFFFFF). This ensures that a chip-select to the boot ROM or EPROM will fetch the reset vector and execute the initialization code, which should set up the chip-select ranges.

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

Note: If the group address and chip-select registers are programmed to overlap, the $\overline{\text{CSx}}$ signals will overlap too. Unused chip-selects must be programmed to 0 wait states and 16 bits wide. Map them to dummy space if necessary. When you are configuring the chip-select signals, the core can be set to write to a read-only location. This causes the $\overline{\text{CS}}$ and $\overline{\text{DTACK}}$ signals to not be asserted and the $\overline{\text{BERR}}$ signal to be asserted if a bus error timer is enabled.

5.2 PROGRAMMING MODEL

The chip-select module of the DragonBall microprocessor contains registers that you can use to control external devices, such as memory. Chip-selects do not operate until the register in a particular group of devices is initialized and the V bit is set in the corresponding

group-base address register. The only exception is the $\overline{CSA0}$ signal, which is the boot device chip-select.

5.2.1 Group Base Address Registers

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

GRPBASEA–GRPBASED

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	GBA31	GBA30	GBA29	GBA28	GBA27	GBA26	GBA25	GBA24	GBA23	GBA22	GBA21	GBA20	RESERVED			V
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—			R/W
RESET	0x0000															
ADDRESS	0x(FF)FFF100, 102, 104, 106															

GBA—Group Base Address 31–20

This bit selects the starting address for the group address range. The corresponding bits in the group base mask register define the block size for the group. This field is compared to the address on the address bus to determine if the group is decoded.

Bits 1–3—Reserved

These bits are reserved.

V—Valid

This bit controls whether or not the contents of its base address register and address mask register pair are valid. The programmed chip-selects do not assert until this bit is set. A reset clears this bit in each base address register.

0 = Content is not valid.
1 = Content is valid.

5.2.2 Group Base Address Mask Registers

The group base address mask registers (GRPMASKA–GRPMASKD) define the address comparison range for a group of devices. When the bits in this register are set to 1 the bits in the corresponding address lines (A[31:20]) compare true (“don’t care”).

GRPMASKA–GRPMASKD

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	GMA31	GMA30	GMA29	GMA28	GMA27	GMA26	GMA25	GMA24	GMA23	GMA22	GMA21	GMA20	RESERVED			

GRPMASKA–GRPMASKD

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—			
RESET	0x0000															
ADDRESS	0x(FF)FFF108, 10A, 10C, 10E															

GMA—Group Mask

These bits mask address A[31:20]. They are used to select the group size. For example, if all the mask bits are cleared, A[31:20] are compared against the value programmed in the group base address register. In this case, the group has a 1M space. If the GMA20 bit is set and the rest of the bits are clear, the group is selected if A[31:21] are the same as the value programmed in the group-base address register. This provides 2M of space for the group. You can decode each chip-select by comparing the lower address lines with the contents of the chip-select option registers. If the memory space for a group of devices is small, you can program the groups to a common space and use the chip-select option registers to decode the areas for each chip-select.

0 = For a match to occur, the address line must match the corresponding bit in the GRPBASEx register.
1 = The corresponding address line compares true (“don’t care”).

Bits 0–3—Reserved

These bits are reserved.

5.2.3 Chip-Select Option Registers

There are four 32-bit chip-select option registers (CSA0–3 and CSB0–3) in each chip-select group, one for each chip-select signal. Chip-selects in group A and B decode address lines A[23:16] for a minimum 64K space. Chip-selects in group C and D decode address lines A[23:12] for a minimum 4K space. When a group address match and a chip-select option address match occurs, a chip-select is generated.

CSC AND CSD

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIELD	AC23	AC22	AC21	AC20	AC19	AC18	AC17	AC16	AC15	AC14	AC13	AC12	RESERVED			BUSW
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RW	R/W	R/W	—			R/W
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	AM23	AM22	AM21	AM20	AM19	AM18	AM17	AM16	AM15	AM14	AM13	AM12	RO	WAIT		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
RESET	0x00010006															
ADDRESS	0x(FF)FFF130, 134, 138, 13C, 140, 144, 148, 14C															

AC23–AC16—Address Compare 23-16 for Groups A and B

This field, together with the AM field, defines the comparison range for this chip-select. Some of the address bits overlap in the group base address and mask registers and chip-select register. The overlapping gives you the flexibility to select a large group and finally decode a chip-select.

AM23–AM16—Address Mask 23-16

This field masks the corresponding bits in the AC field.

0 = For a match to occur, the address line must match the corresponding bit in the CSA or CSB register.
1 = The corresponding address line compares true (“don’t care”).

BUSW—Bus Width

This bit sets the bus width of the memory space selected by this chip-select.

0 = 8-bit.
1 = 16-bit.

Note: This bit is read-only for CSA0 and reflects the logic level on the BBUSW pin.

RO—Read Only

This bit configures the memory space selected by this chip-select as read-only. Otherwise, read or write accesses are allowed. However, writes to read-only space cause a write-protection violation to occur, as described in **Section 4.1.1 System Control Register**.

0 = Read/write.
1 = Read-only.

WAIT—Wait-State Selection

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

AC23–AC12—Address Compare 23-12 for Groups C and D

This field, together with the AM field, defines the comparison range for this chip-select. Some of the address bits overlap in the group base address and mask registers and

chip-select register. The overlapping gives you the flexibility to select a large group and finally decode a chip-select.

AM23–AM12—Address Mask 23-12

This field masks the corresponding bits in the AC field.

0 = For a match to occur, the address line must match the corresponding bit in the CSC or CSD register.
1 = The corresponding address line compares true (“don’t care”).

BUSW—Bus Width

This bit sets the bus width of the memory space selected by this chip-select.

0 = 8-bit.
1 = 16-bit.

RO—Read Only

This bit configures the memory space selected by this chip-select as read-only. Otherwise, read or write accesses are allowed. However, writes to read-only space cause a write-protection violation to occur, as described in **Section 4.1.1 System Control Register**.

0 = Read/write.
1 = Read-only.

WAIT—Wait-State Selection

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

5.3 PCMCIA 1.0 SUPPORT

The Dragonball 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. The PCMCIA address range is decoded with CSD3.

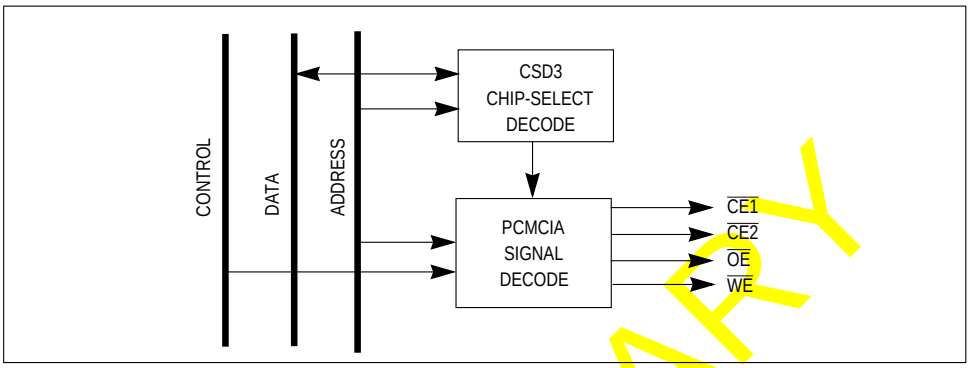


Figure 5-2. PCMCIA Block Diagram

5.3.1 PCMCIA Configuration

The DragonBall can access PCMCIA 1.0 memory cards by setting the decode range for CSD3 to the area of memory where the memory card is to be accessed. You must configure Bit 6 in Port C for its PCMCIA function ($\overline{WE}$) and Bits 6 and 7 of Port K bits must also be configured for their PCMCIA function ($\overline{CE2}$ and $\overline{CE1}$). Read or write accesses to the area decoded by CSD3 asserts the corresponding PCMCIA control signals.

SECTION 6
INTERRUPT CONTROLLER

The interrupt controller supports a variety of interrupts from internal and external sources. This block prioritizes and encodes pending interrupts and generates interrupt vector numbers during the interrupt acknowledge cycle.

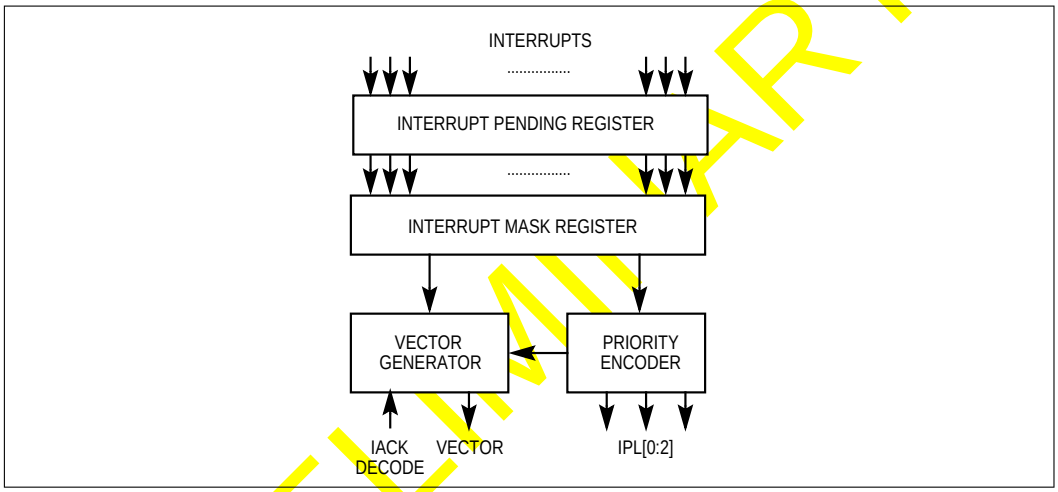


Figure 6-1. Interrupt Controller Block Diagram

The interrupt controller supports 23 edge- and level-sensitive interrupts. There are seven interrupt levels. Level 7 has the highest priority and level 1 has the lowest. Interrupts can originate from the following sources:

- $\overline{\text{IRQ7}}$ (level 7)
- Serial peripheral interface (SPI) slave that needs service (level 6)
- Timer 1 event (level 6)
- $\overline{\text{IRQ6}}$ external interrupt (level 6)
- $\overline{\text{PENIRQ}}$ (level 5)
- Serial peripheral interface (SPI) master that needs service (level 4)
- Timer 2 event (level 4)
- UART event (level 4)

- Watchdog timer interrupt (level 4)
- Real-time clock interrupt (level 4)
- Keyboard interrupt (level 4)
- PWM interrupt (level 4)
- General-purpose interrupt $\overline{INT}$[0:7] (level 4)
- $\overline{IRQ3}$ external interrupt (level 3)
- $\overline{IRQ2}$ external interrupt (level 2)
- $\overline{IRQ1}$ external interrupt (level1)

The interrupt controller generates a programmable vector for each interrupt level listed above. These interrupt vectors are referred to as interrupt autovectors or user interrupt vectors in the 68EC000-compatible exception vector table in Table 6-1.

6.1 EXCEPTION VECTORS

A vector number is an 8-bit number that can be multiplied by 4 to obtain the address of an exception vector. An exception vector is the memory location from which the processor fetches the address of a software routine that is used to handle an exception. Each exception has a vector number and a exception vector, as described in the Table 6-1. User interrupts are part of the exception processing on the DragonBall and the vector numbers for user interrupts are configurable. For additional information regarding exception processing, see the *M68000 Programmer's Reference Manual*.

Table 6-1. Exception Vector Assignment

VECTORS NUMBERS		ADDRESS		SPACE	ASSIGNMENT
HEX	DECIMAL	DECIMAL	HEX		
0	0	0	000	SP	Reset: Initial SSP
1	1	4	004	SP	Reset: Initial PC
2	2	8	008	SD	Bus Error
3	3	12	00C	SD	Address Error
4	4	16	010	SD	Illegal Instruction
5	5	20	014	SD	Divide-by-Zero
6	6	24	018	SD	CHK Instruction
7	7	28	01C	SD	TRAPV Instruction
8	8	32	020	SD	Privilege Violation
9	9	36	024	SD	Trace
A	10	40	028	SD	Line 1010 Emulator
B	11	44	02C	SD	Line 1111 Emulator
C	12	48	030	SD	(Unassigned, Reserved)

Table 6-1. Exception Vector Assignment (Continued)

D	13	52	034	SD	(Unassigned, Reserved)
E	14	56	038	SD	(Unassigned, Reserved)
F	15	60	03C	SD	Uninitialized Interrupt Vector
10–17	16–23	64	040	SD	(Unassigned, Reserved)
		92	05C		
18	24	96	060	SD	Spurious Interrupt
19	25	100	064	SD	Level 1 Interrupt Autovector
1A	26	104	068	SD	Level 2 Interrupt Autovector
1B	27	108	06C	SD	Level 3 Interrupt Autovector
1C	28	112	070	SD	Level 4 Interrupt Autovector
1D	29	116	074	SD	Level 5 Interrupt Autovector
1E	30	120	078	SD	Level 6 Interrupt Autovector
1F	31	124	07C	SD	Level 7 Interrupt Autovector
20–2F	32–47	128	080	SD	TRAP Instruction Vectors
		188	0BC		
30–3F	48–63	192	0C0	SD	(Unassigned, Reserved)
		255	0FF		
40–FF	64–255	256	100	SD	User Interrupt Vectors
		1020	3FC		

NOTES:

1. Vector numbers 12–14, 16–23, and 48–63 are reserved for future enhancements by Motorola. None of your peripheral devices should be assigned to these numbers.
2. Reset vector 0 requires four words, unlike the other vectors which only require two words, and it is located in the supervisor program space.
3. The spurious interrupt vector is taken when there is a bus error indication during interrupt processing.
4. TRAP #n uses vector number 32+ n (decimal).
5. SP denotes supervisor program space and SD denotes supervisor data space.

Note: The DragonBall does not provide autovector interrupts. At system start-up, you need to program the user interrupt vector so that the processor can handle interrupts properly.

6.2 RESET

The reset exception corresponds to the highest exception level. A reset exception is processed for system initialization and to recover from a catastrophic failure. Any processing that is in progress at the time of the reset is aborted and cannot be recovered. Neither the program counter nor the status register is saved. The processor is forced into the supervisor state. The interrupt priority mask is set at level 7. The address in the first 2 words of the reset exception vector is fetched by the processor as the initial SSP (Supervisor Stack Pointer), and the address in the next two words of the reset exception vector is fetched as the initial program counter.

At start-up or reset, the default chip-select (CSA0) is asserted and all other chip-selects are negated. You should use CSA0 to decode an EPROM/ROM memory space. In this case, the first two long-words of the EPROM/ROM memory space should be programmed to contain the initial SSP and PC. The initial SSP should point to a RAM space and the initial PC should point to the start-up code within the EPROM/ROM space so that the processor can execute the start-up code to bring up the system.

Note: The DragonBall does not support the **reset** instruction. It will not cause a reset exception or an assertion of the $\overline{\text{RESET}}$ pin.

$\overline{\text{RESET}}$ is an input-only pin on the DragonBall. For more information about core interrupts, see the Motorola application note called *A Discussion of Interrupts for the MC68000* (part number AN1012).

6.3 INTERRUPT CONTROLLER

When interrupts are received by the controller, they are prioritized and the highest enabled pending interrupt is posted to the core. Before the CPU responds to this interrupt, the status register is copied internally. Then the S bit of the status register is set, which puts the processor into supervisor mode. The CPU then responds with an interrupt acknowledge cycle, in which the lower 3 bits of the address bus reflect the level of the current interrupt. The interrupt controller generates a vector number during the interrupt acknowledge cycle and the CPU uses this vector number to generate a vector address. Except for the reset exception, the CPU saves the current processor status, including the program counter value (which points to the next instruction to be executed after the interrupt), and the saved copy of the status register. The new program counter is updated to the content of the interrupt vector, which points to the interrupt service routine. The CPU then resumes instruction execution to execute the interrupt service routine.

Interrupt priority is based on the interrupt level. If the CPU is currently processing an interrupt service routine and a higher priority interrupt is posted, the process described above repeats, and the higher priority interrupt is serviced. If the priority of the newer interrupt is lower than or equal to the priority of the current interrupt, execution of the current interrupt handler continues. This newer interrupt is postponed until its priority becomes the

highest. Interrupts within a same level should be prioritized in software by the interrupt handler. The interrupt service routine should end with the **rte** instruction, which restores the processing state prior to the interrupt.

The DragonBall provides one interrupt vector for each of the seven user interrupt levels. These interrupt vectors form the user interrupt vector section of Table 6-1. The user interrupt vectors can be located anywhere within the address range 0x100 to 0x400. You can program the five most-significant bits of the interrupt vector number, but the lower three bits reflect the interrupt level that is being serviced. All interrupts are maskable by the interrupt controller. If an interrupt is masked, its status can still be accessed in the interrupt pending register (IPR).

6.4 VECTOR GENERATOR

The interrupt controller provides a vector number to the core. You can program the upper five bits of the interrupt vector register (IVR) to allow the interrupt vector number to point to any address in the exception vector table. However, many of the vector addresses are assigned to the core's internal exceptions and cannot be reused. This leaves only a small range of address space (from 0x100 through 0x400) where you can configure the IVR to locate user interrupt vectors. For example, if you write a value of 0x40 to the IVR, the interrupt vector base is set to point to 0x100 (0x40<<2), which is the beginning of the user interrupt vectors shown in Table 6-1. The coding for the vector numbers is provided in Table 6-2.

Table 6-2. Interrupt Vector Numbers

INTERRUPT	VECTOR NUMBER
Level 7	xxxxx111
Level 6	xxxxx110
Level 5	xxxxx101
Level 4	xxxxx100
Level 3	xxxxx011
Level 2	xxxxx010
Level 1	xxxxx001

NOTE: xxxxx is replaced by the upper five bits of the interrupt vector register.

6.5 PROGRAMMING MODEL

This section describes registers that you may need to configure so that the interrupt controller can properly process interrupts, generate vector numbers, and post interrupts to the core.

6.5.1 Interrupt Vector Register

The interrupt vector register (IVR) programs the upper five bits of the interrupt vector number. During the interrupt-acknowledge cycle, the lower three bits, encoded from the interrupt level, are combined with the upper five bits to form an 8-bit vector number. The CPU uses the vector number to generate a vector address. During system start-up, the IVR register should be configured so that Dragonball's external and internal interrupts can be handled properly by their software handlers. If an interrupt occurs before the IVR has been programmed, the interrupt vector number 0x0F is returned to the CPU as an uninitialized interrupt, which has the interrupt vector 0x3C.

IVR

BITS	7	6	5	4	3	2	1	0
FIELD	VECTOR					0	0	0
R/W	R/W					R	R	R
RESET	0x00							
ADDR	0x(FF)FFF300							

VECTOR—
This field is the upper five bits of the interrupt vector number.

0
These lower three bits are always set to 0.

6.5.2 Interrupt Control Register

The interrupt control register (ICR) controls the behavior of the external interrupt inputs. It infroms the interrupt controller whether the interrupt signal is an edge-triggered or level-sensitive interrupt, as well as whether it has positive or negative polarity.

ICR

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	POL1	POL2	POL3	POL6	ET1	ET2	ET3	ET6	RESERVED							
RESET	0x0000															
R/W	R/W								—							
ADDR	0x(FF)FFF302															

POL1—Polarity 1
This bit controls interrupt polarity on $\overline{\text{IRQ1}}$. In level-sensitive mode, negative polarity means that an interrupt occurs when the signal is at logic level low. Positive polarity means that an interrupt occurs when the signal is at logic level high. In edge-triggered mode, negative polarity means that an interrupt occurs when the signal goes from logic level high to logic

level low. Positive polarity means that an interrupt occurs when the signal goes from logic level low to logic level high.

0 = Negative polarity (default at reset).
1 = Positive polarity.

POL2—Polarity 2

This bit controls interrupt polarity for $\overline{IRQ2}$. In level-sensitive mode, negative polarity means that an interrupt occurs when the signal is at logic level low. Positive polarity means that an interrupt occurs when the signal is at logic level high. In edge-triggered mode, negative polarity means that an interrupt occurs when the signal goes from logic level high to logic level low. Positive polarity means that an interrupt occurs when the signal goes from logic level low to logic level high.

0 = Negative polarity (default at reset).
1 = Positive polarity.

POL3—Polarity 3

This bit controls interrupt polarity for $\overline{IRQ3}$. In level-sensitive mode, negative polarity means that an interrupt occurs when the signal is at logic level low. Positive polarity means that an interrupt occurs when the signal is at logic level high. In edge-triggered mode, negative polarity means that an interrupt occurs when the signal goes from logic level high to logic level low. Positive polarity means that an interrupt occurs when the signal goes from logic level low to logic level high.

0 = Negative polarity (default at reset).
1 = Positive polarity.

POL6—Polarity 6

This bit controls interrupt polarity for $\overline{IRQ6}$. In level-sensitive mode, negative polarity means that an interrupt occurs when the signal is at logic level low. Positive polarity means that an interrupt occurs when the signal is at logic level high. In edge-triggered mode, negative polarity means that an interrupt occurs when the signal goes from logic level high to logic level low. Positive polarity means that an interrupt occurs when the signal goes from logic level low to logic level high.

0 = Negative polarity (default at reset).
1 = Positive polarity.

Note: Clear your interrupts after you change modes. When you change modes from level to edge interrupts, an edge can be created, which causes an interrupt to be posted.

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

When this bit is set, the $\overline{IRQ1}$ signal is an edge-triggered interrupt. In edge-triggered mode, you must write a 1 to the IRQ1 bit in the interrupt status register to clear this interrupt. When

this bit is low, $\overline{IRQ1}$ is a level-sensitive interrupt. In this case, you must clear the external source of the interrupt.

0 = Level-sensitive interrupt (default at reset).
1 = Edge-sensitive interrupt.

ET2—$\overline{RQ2}$ Edge Trigger Select

When this bit is set, the $\overline{IRQ2}$ signal is an edge-triggered interrupt. In edge-triggered mode, you must write a 1 to the IRQ2 bit in the interrupt status register to clear this interrupt. When this bit is low, $\overline{IRQ2}$ is a level-sensitive interrupt. In this case, you must clear the external source of the interrupt.

0 = Level-sensitive interrupt (default at reset).
1 = Edge-sensitive interrupt.

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

When this bit is set, the $\overline{IRQ3}$ signal is an edge-triggered interrupt. In edge-triggered mode, you must write a 1 to the IRQ3 bit in the interrupt status register to clear this interrupt. When this bit is low, $\overline{IRQ3}$ is a level-sensitive interrupt. In this case, you must clear the external source of the interrupt.

0 = Level-sensitive interrupt (default at reset).
1 = Edge-sensitive interrupt.

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

When this bit is set, the $\overline{IRQ6}$ signal is an edge-triggered interrupt. In edge-triggered mode, you must write a 1 to the IRQ6 bit in the interrupt status register to clear this interrupt. When this bit is low, $\overline{IRQ6}$ is a level-sensitive interrupt. In this case, you must clear the external source of the interrupt.

0 = Level-sensitive interrupt (default at reset).
1 = Edge-sensitive interrupt.

Bits 0–7—Reserved

These bits are reserved and should remain at their default value.

6.5.2.1 INTERRUPT MASK REGISTER. The interrupt mask (IMR) is a control register that can mask out a particular interrupt if the corresponding bit for the interrupt is set. There is one control bit for each interrupt source. When an interrupt is masked, the interrupt controller will not generate an interrupt request to the CPU, but its status can still be observed in the

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

IMR

BITS	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIELD	RESERVED								MIRQ7	MTMR1	MSPIS	MPEN	MIRQ6	MIRQ3	MIRQ2	MIRQ1
R/W	—								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0x00FF															
BITS	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	MINT7	MINT6	MINT5	MINT4	MINT3	MINT2	MINT1	MINT0	MPWM	MKB	RSVD	MRTC	MWDT	MUART	MTMR2	MSPIM
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0xFFFF															
ADDR	0x(FF)FFF304															

MSPIM—Mask SPI Master Interrupt.

0 = Enable serial peripheral interface master interrupt.
1 = Mask serial peripheral interface master interrupt (default at reset).

MTMR2—Mask Timer 2 Interrupt

0 = Enable timer 2 interrupt.
1 = Mask timer 2 interrupt (default at reset).

MUART—Mask UART Interrupt

0 = Enable UART interrupt.
1 = Mask UART interrupt (default at reset).

MWDT—Mask Watchdog Timer Interrupt

0 = Enable watchdog timer interrupt.
1 = Mask watchdog timer interrupt (default at reset).

MRTC—Mask RTC Interrupt

0 = Enable real-time clock interrupt.
1 = Mask real-time clock interrupt (default at reset).

Bit 5—Reserved

MKB—Mask Keyboard Interrupt

0 = Enable keyboard interrupt.
1 = Mask keyboard interrupt (default at reset).

MPWM—Mask PWM Interrupt

0 = Enable pulse-width modulator interrupt.
1 = Mask pulse-width modulator interrupt (default at reset).

MINT0—Mask External INT0 interrupt
- 0 = Enable INT0 interrupt.
- 1 = Mask INT0 interrupt (default at reset).

MINT1—Mask External INT1 interrupt
- 0 = Enable INT1 interrupt.
- 1 = Mask INT1 interrupt (default at reset).

MINT2—Mask External INT2 interrupt
- 0 = Enable INT2 interrupt.
- 1 = Mask INT2 interrupt (default at reset).

MINT3—Mask External INT3 interrupt
- 0 = Enable INT3 interrupt.
- 1 = Mask INT3 interrupt (default at reset).

MINT4—Mask External INT4 interrupt
- 0 = Enable INT4 interrupt.
- 1 = Mask INT4 interrupt (default at reset).

MINT5—Mask External INT5 interrupt
- 0 = Enable INT5 interrupt.
- 1 = Mask INT5 interrupt (default at reset).

MINT6—Mask External INT6 interrupt
- 0 = Enable INT6 interrupt.
- 1 = Mask INT6 interrupt (default at reset).

MINT7—Mask External INT7 interrupt
- 0 = Enable INT7 interrupt.
- 1 = Mask INT7 interrupt (default at reset).

MIRQ1—Mask IRQ1Interrupt
- 0 = Enable IRQ1 interrupt.
- 1 = Mask IRQ1 interrupt (default at reset).

MIRQ2—Mask IRQ2 Interrupt
- 0 = Enable IRQ2 interrupt.
- 1 = MaskIRQ2 interrupt (default value at reset).

MIRQ3—Mask IRQ3 Interrupt
- 0 = Enable IRQ3 interrupt.
- 1 = Mask IRQ3 interrupt (default value at reset).

MIRQ6—Mask IRQ6 Interrupt

0 = Enable IRQ6 interrupt.
1 = Mask IRQ6 interrupt (default value at reset).

MPEN—Mask Pen Interrupt

0 = Enable pen-down interrupt.
1 = Mask pen-down interrupt (default value at reset).

MSPIS—Mask Serial Peripheral Interface (SPI) Slave Interrupt

0 = Enable SPI slave interrupt.
1 = Mask SPI slave interrupt (default value at reset).

MTMR1—Mask Timer 1 Interrupt

0 = Enable timer 1 interrupt.
1 = Mask timer 1 interrupt (default value at reset).

MIRQ7—Mask IRQ7 Interrupt

When set, this bit indicates that the IRQ7 interrupt is masked. Traditionally, the level 7 interrupt is known as nonmaskable interrupt in the M68000 architecture. This means that when all interrupts are masked in the CPU status register, the level 7 interrupt can still be recognized by the microprocessor. On the MC68328, however, you can set this bit to block IRQ7 interrupt from being sent to the processor.

0 = Enable IRQ7 interrupt.
1 = Mask IRQ7 interrupt.

6.5.2.2 INTERRUPT WAKE-UP ENABLE REGISTER. The interrupt wake-up enable register (IWR) enables the corresponding interrupt source to start the power-control wake-up sequence. When a bit in this register is set, the corresponding interrupt can cause the processor to wake up from sleep or doze mode. When a bit in this register is cleared, the corresponding interrupt will not be able to wake up the processor. At reset, all bits in this register are set to enable the interrupts to cause a wake-up event. The bit positions in this register correspond to the bits in the interrupt status register, interrupt pending register, and interrupt mask register.

IWR

BITS	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIELD	RESERVED								IRQ7	TMR1	SPIS	PEN	IRQ6	IRQ3	IRQ2	IRQ1
BITS	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	PWM	WKB	RSVD	RTC	WDT	UART	TMR2	SPIM
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0xFFFF															
ADDR	0x(FF)FFF308															

IWR

BITS	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	—								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0x00FF															
BITS	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	PWM	WKB	RSVD	RTC	WDT	UART	TMR2	SPIM
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0xFFFF															
ADDR	0x(FF)FFF308															

SPIM—Wake-Up SPI Master Interrupt

0 = Disallow SPI master interrupt from waking up the processor.
1 = Enable SPI master interrupt to wake up the processor. (default at reset)

TMR2—Wake-Up Timer 2 Interrupt

0 = Disallow timer 2 interrupt from waking up the processor.
1 = Enable timer 2 interrupt to wake up the processor (default at reset).

UART—Wake-Up UART Interrupt

0 = Disallow UART interrupt from waking up the processor.
1 = Enable UART interrupt to wake up the processor (default at reset).

WDT—Wake-up Watchdog Timer Interrupt

0 = Disallow watchdog timer interrupt from waking up the processor.
1 = Enable watchdog timer interrupt to wake up the processor (default at reset).

RTC—Wake-up RTC Interrupt

0 = Disallow real-time clock interrupt from waking up the processor.
1 = Enable real-time clock interrupt to wake up the processor (default at reset).

Bit 5—Reserved

These bits are reserved and should remain at their default value.

WKB—Wake-up Keyboard Interrupt

0 = Disallow keyboard interrupt from waking up the processor.
1 = Enable keyboard interrupt to wake up the processor (default at reset).

PWM—Wake-up PWM Interrupt

0 = Disallow pulse-width modulator interrupt from waking up the processor.
1 = Enable pulse-width modulator interrupt to wake up the processor (default at reset).

INT0—Wake-up External INT0

0 = Disallow INT0 interrupt from waking up the processor.
1 = Enable INT0 interrupt to wake up the processor (default at reset).

INT1—Wake-up External INT1

0 = Disallow INT1 interrupt from waking up the processor.
1 = Enable INT1 interrupt to wake up the processor (default at reset).

INT2—Wake-up External INT2

0 = Disallow INT2 interrupt from waking up the processor.
1 = Enable INT2 interrupt to wake up the processor (default at reset).

INT3—Wake-up External INT3

0 = Disallow INT3 interrupt from waking up the processor.
1 = Enable INT3 interrupt to wake up the processor (default at reset).

INT4—Wake-up External INT4

0 = Disallow INT4 interrupt from waking up the processor.
1 = Enable INT4 interrupt to wake up the processor (default at reset).

INT5—Wake-up External INT5

0 = Disallow INT5 interrupt to wake up the processor.
1 = Enable INT5 interrupt to wake up the processor (default at reset).

INT6—Wake-up External INT6

0 = Disallow INT6 interrupt from waking up the processor.
1 = Enable INT6 interrupt to wake up the processor (default at reset).

INT7—Wake-up External INT7

0 = Disallow INT7 interrupt from waking up the processor.
1 = Enable INT7 interrupt to wake up the processor (default at reset).

IRQ1—Wake-up IRQ1Interrupt

0 = Disallow IRQ1 interrupt from waking up the processor.
1 = Enable IRQ1interrupt to wake up the processor (default at reset).

IRQ2—Wake-up IRQ2 Interrupt

0 = Disallow IRQ2 interrupt from waking up the processor.
1 = Enable IRQ2 interrupt to wake up the processor (default at reset).

IRQ3—Wake-up IRQ3 Interrupt

0 = Disallow IRQ3 interrupt from waking up the processor.
1 = Enable IRQ3 interrupt to wake up the processor (default at reset).

IRQ6—Wake-up IRQ6 Interrupt

0 = Disallow IRQ6 interrupt from waking up the processor.
1 = Enable IRQ6 interrupt to wake up the processor (default at reset).

PEN—Wake-up Pen Interrupt

0 = Disallow pen-down interrupt from waking up the processor.
1 = Enable pen-down interrupt to wake up the processor (default at reset).

SPIS—Wake-up Serial Peripheral Interface (SPI) Slave Interrupt

0 = Disallow SPI slave interrupt from waking up the processor.
1 = Enable SPI slave interrupt to wake up the processor (default at reset).

TMR1—Wake-up Timer 1 Interrupt

0 = Disallow timer 1 interrupt from waking up the processor.
1 = Enable timer 1 interrupt to wake up the processor (default at reset).

IRQ7—Wake-up IRQ7 Interrupt

0 = Disallow IRQ7 interrupt from waking up the processor.
1 = Enable IRQ7 interrupt to wake up the processor (default at reset).

6.5.2.3 INTERRUPT STATUS REGISTER. During the interrupt service, the interrupt handler can determine the source of the interrupt by examining the interrupt status register (ISR). Each bit in this register, when set, indicates the corresponding interrupt is posted to the core. If there are multiple interrupt sources at the same level, the software handler may need to prioritize them, depending on the application.

Each interrupt status bit in the interrupt status register reflects the interrupt request from their respective interrupt sources. Refer to the specific sections about the timer, SPI master, SPI slave, real-time clock, and pulse-width modulation for details about how to clear the interrupt. The $\overline{\text{IRQ7}}$ signal is an active-low edge-triggered interrupt request and an IRQ7 interrupt is clear by writing a 1 to the IRQ7 status bit. When programmed as edge-triggered interrupts, IRQ1, IRQ2, IRQ3, and IRQ6 interrupts can be cleared by writing a 1 to the corresponding status bit in the register. When programmed as level-triggered interrupts, these interrupts are cleared at the requesting sources.

SPIM—SPI Master Interrupt Request

When this bit is set, it indicates that a data transfer is complete. You must clear this interrupt in the serial peripheral interface control register. This interrupt is a level 4 interrupt.

0 = No SPI Master interrupt pending.
1 = A SPI Master interrupt is posted.

TIMER2—Timer 2 Interrupt Request

This bit indicates that a timer 2 event has occurred. This is a level 4 interrupt. See **Section 8 Timers** for more information about timer operation.

0 = No timer 2 event occurred.
1 = Timer 2 event has occurred.

UART—UART Interrupt Request

When this bit is set, it indicates that the UART module needs service. This is a level 4 interrupt.

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

WDT—Watchdog Timer Interrupt Request

This bit indicates that a watchdog timer interrupt is pending.

0 = No watchdog timer interrupt
1 = A watchdog timer interrupt is pending

RTC—Real-Time Clock Interrupt Request

This bit indicates that the real-time clock is requesting an interrupt.

0 = No real-time clock interrupt
1 = A real-time clock interrupt is pending

Bit 5—Reserved

This bit is reserved and should remain at its default value.

KB—Keyboard Interrupt Request

This bit indicates whether there is a keyboard interrupt.

0 = No keyboard interrupt.
1 = Keyboard interrupt is pending.

PWM—Pulse-Width Modulator Interrupt

This bit indicates that a PWM period rollover event occurs. This is a level 4 interrupt.

0 = No pulse-width modulator period rollover event occurred.
1 = Pulse-width modulator period rolled over.

INT0—External INT0 Interrupt

0 = No INT0 interrupt.
1 = INT0 interrupt is pending.

INT1—External INT1 Interrupt

0 = No INT1 interrupt.
1 = INT1 interrrupt is pending.

INT2—External INT2 Interrupt

0 = No INT2 interrupt.
1 = INT2 interrupt is pending.

INT3—External INT3 Interrupt

0 = No INT3 interrupt.
1 = INT3 interrupt is pending.

INT4—External INT4 Interrupt

0 = No INT4 interrupt.
1 = INT4 interrupt is pending.

INT5—External INT5 Interrupt

0 = No INT5 interrupt.
1 = INT5 interrupt is pending.

INT6— External INT6 Interrupt

0 = No INT6 interrupt.
1 = INT6 interrupt is pending.

INT7— External INT7 Interrupt

0 = No INT7 interrupt.
1 = INT7 interrupt is pending.

IRQ1—Interrupt Request Level 1

When set, this bit indicates that an external device has requested an interrupt on level 1. If IRQ1 signal is set to be a level-sensitive interrupt, you must clear the source of the interrupt. If IRQ1 signal is set to be an edge-triggered interrupt, you must clear the interrupt by writing a 1 to this bit (writing a 0 has no effect).

0 = No level 1 interrupt is pending.
1 = Level 1 interrupt is pending.

IRQ2—Interrupt Request Level 2

When set, this bit indicates that an external device has requested an interrupt on level 2. If IRQ2 signal is set to be a level-sensitive interrupt, you must clear the source of the interrupt. If IRQ2 signal is set to be an edge-triggered interrupt, you must clear the interrupt by writing a 1 to this bit (writing a 0 has no effect).

0 = No level 2 interrupt is pending.
1 = Level 2 interrupt is pending.

IRQ3—Interrupt Request Level 3

When set, this bit indicates that an external device has requested an interrupt on level 3. If IRQ3 signal is set to be a level-sensitive interrupt, you must clear the source of the interrupt.

If IRQ3 signal is set to be an edge-triggered interrupt, you must clear the interrupt by writing a 1 to this bit (writing a 0 has no effect).

0 = No level 3 interrupt is pending.
1 = Level 3 interrupt is pending.

IRQ6—Interrupt Request Level 6

When this bit is set, it indicates that an external device is requesting an interrupt on level 6. If IRQ6 is set to be a level-sensitive interrupt, you must clear the source of the interrupt. If IRQ6 is set to be an edge-triggered interrupt, you must clear the interrupt by writing a 1 to this bit. Writing a 0 to this bit has no effect.

0 = No level 6 interrupt pending.
1 = Level 6 interrupt is posted.

6.5.2.4 INTERRUPT PENDING REGISTER. The read-only interrupt pending register (IPR) indicates which interrupts are pending. If an interrupt source requests an interrupt, but that interrupt is masked by the interrupt maskregister, then that interrupt bit will be set in this register, but not in the interrupt status register. If the pending interrupt is not masked, the interrupt bit will be set in both registers.

IPR

BITS	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIELD	RESERVED								IRQ7	TMR1	SPIS	PEN	IRQ6	IRQ3	IRQ2	IRQ1
R/W	—								R	R	R	R	R	R	R	R
RESET	—															
BITS	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	PWM	KB	RSVD	RTC	WDT	UART	TMR2	SPIM
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
RESET	—															
ADDR	0x(FF)FFF310															

SPIM—SPI Master Interrupt Request

When this bit is set, it indicates that a data transfer is complete. You must clear this interrupt in the serial peripheral interface control register. This interrupt is a level 4 interrupt.

0 = No SPI Master interrupt pending.
1 = A SPI Master interrupt is posted.

TIMER2—Timer 2 Interrupt Request

This bit indicates that a timer 2 event has occurred. This is a level 4 interrupt. See **Section 8 Timers** for more information about timer operation.

0 = No timer 2 event occurred.
1 = Timer 2 event has occurred.

UART—UART Interrupt Request

When this bit is set, it indicates that the UART module needs service. This is a level 4 interrupt.

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

WDT—Watchdog Timer Interrupt Request

This bit indicates that a watchdog timer interrupt is pending.

0 = No watchdog timer interrupt
1 = A watchdog timer interrupt is pending

RTC—Real-Time Clock Interrupt Request

This bit indicates that the real-time clock is requesting an interrupt.

0 = No real-time clock interrupt
1 = A real-time clock interrupt is pending

Bit 5—Reserved

This bit is reserved and should remain at its default value.

KB—Keyboard Interrupt Request

This bit indicates whether there is a keyboard interrupt.

0 = No keyboard interrupt.
1 = Keyboard interrupt is pending.

PWM—Pulse-Width Modulator Interrupt

This bit indicates that a PWM period rollover event occurs. This is a level 4 interrupt.

0 = No pulse-width modulator period rollover event occurred.
1 = Pulse-width modulator period rolled over.

INT0—External INT0 Interrupt

0 = No INT0 interrupt.
1 = INT0 interrupt is pending.

INT1—External INT1 Interrupt

0 = No INT1 interrupt.
1 = INT1 interrrupt is pending.

INT2—External INT2 Interrupt

0 = No INT2 interrupt.
1 = INT2 interrupt is pending.

INT3—External INT3 Interrupt

0 = No INT3 interrupt.
1 = INT3 interrupt is pending.

INT4—External INT4 Interrupt

0 = No INT4 interrupt.
1 = INT4 interrupt is pending.

INT5—External INT5 Interrupt

0 = No INT5 interrupt.
1 = INT5 interrupt is pending.

INT6— External INT6 Interrupt

0 = No INT6 interrupt.
1 = INT6 interrupt is pending.

INT7— External INT7 Interrupt

0 = No INT7 interrupt.
1 = INT7 interrupt is pending.

IRQ1—Interrupt Request Level 1

When set, this bit indicates that an external device has requested an interrupt on level 1. If IRQ1 signal is set to be a level-sensitive interrupt, you must clear the source of the interrupt. If IRQ1 signal is set to be an edge-triggered interrupt, you must clear the interrupt by writing a 1 to this bit (writing a 0 has no effect).

0 = No level 1 interrupt is pending.
1 = Level 1 interrupt is pending.

IRQ2—Interrupt Request Level 2

When set, this bit indicates that an external device has requested an interrupt on level 2. If IRQ2 signal is set to be a level-sensitive interrupt, you must clear the source of the interrupt. If IRQ2 signal is set to be an edge-triggered interrupt, you must clear the interrupt by writing a 1 to this bit (writing a 0 has no effect).

0 = No level 2 interrupt is pending.
1 = Level 2 interrupt is pending.

IRQ3—Interrupt Request Level 3

When set, this bit indicates that an external device has requested an interrupt on level 3. If IRQ3 signal is set to be a level-sensitive interrupt, you must clear the source of the interrupt.

If IRQ3 signal is set to be an edge-triggered interrupt, you must clear the interrupt by writing a 1 to this bit (writing a 0 has no effect).

0 = No level 3 interrupt is pending.
1 = Level 3 interrupt is pending.

IRQ6—Interrupt Request Level 6

When this bit is set, it indicates that an external device is requesting an interrupt on level 6. If IRQ6 is set to be a level-sensitive interrupt, you must clear the source of the interrupt. If IRQ6 is set to be an edge-triggered interrupt, you must clear the interrupt by writing a 1 to this bit. Writing a 0 to this bit has no effect.

0 = No level 6 interrupt pending.
1 = Level 6 interrupt is posted.

SECTION 7
PARALLEL PORTS

The DragonBall microprocessor supports up to 10 parallel ports, that can be configured either as general-purpose input/output ports or as a dedicated peripheral interface. There are three types of ports—basic, pull-up, and interrupt. This section describes these ports and how to configure their functions.

7.1 BASIC PORTS

The basic ports (A, B, C, E, F, G, J, and K) multiplex two functions on each port pin:

- A general I/O function
- A special-purpose function to support on-chip modules

Each port has three configurable 8-bit registers—a data register, direction register, and select register. Each of the 8 bits in the registers affect the function, direction, and output data of each of the corresponding eight pins of a given port. For example, you can configure each bit in the select register of a particular port to choose the associated pin's function.

The direction register determines whether a port pin is an output or input pin. If you configure the SEL bit for a particular pin to be used with a special-purpose function, then the direction of the pin will depend on that function. In this case, configuring the corresponding bit in the direction register does not have any effect on the pin. The corresponding bit in the data register reflects the current logic level on the pin. If you configure the SEL bit for a particular pin to be used with a general I/O function, then you may have to choose whether this pin will be used as an input or output. If the port pin is used as an input pin, the corresponding bit in the data register reflects the current logic level on the pin. If the port pin is used as an output pin, setting the corresponding bit in the data register will output a "1", and clearing the bit will produce a "0" output.

If the SEL bit for a particular pin is cleared, you configure the port pin to have a special-purpose function. For example, while the port K Bit 0 of the select register is cleared, this pin is the output master serial peripheral interface TXD signal. In this case, *Data to Module* signal in Figure 7-1 is connected to the master serial peripheral interface TXD signal. Since this bit is output-only, the *Output Enable from module* signal is always asserted and the *Data to module* signal is not used. Another example is that the port K Bit 1 can be used as the master serial peripheral interface RXD input-only signal. In this case, the *Output Enable from module* input is negated and the *Data from module* signal is not used. The *Data to module* signal is connected to the master serial peripheral interface RXD input.

Figure 7-1 illustrates the internal logic of each basic port.

Note: To prevent a glitch, write the intended data to the data register before you change the mode in the select register from unselected (0) to selected (1).

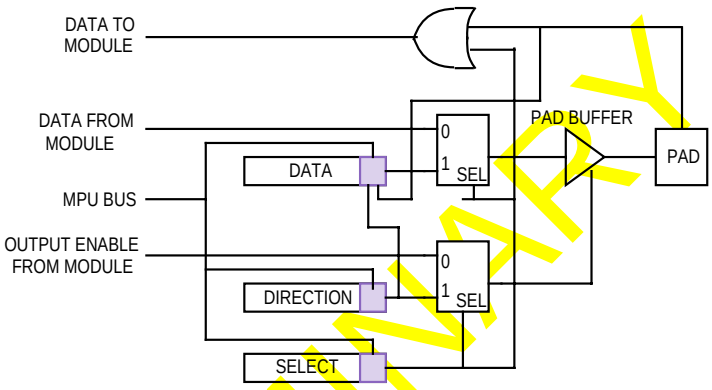


Figure 7-1. Basic Port Operation

7.1.1 Port K Programming Example

Assume the slave SPI is to be enabled, the master SPI and the PCMCIA are not used, we would like to configure port K pins 0, 1, and 2 as general I/O input ports, pins 6 and 7 as output ports, and the remaining pins 3, 4 and 5 are used for Slave SPI module.

Value in port K select register:
0xC7 bits 7-6 are set for I/O mode
bits 5-3 are clear to be used as slave SPI (special-purpose mode)
bits 2-0 are set for I/O mode
Value in port K direction register:
0xC0 bits 7-6 are set so that port pins are outputs
bits 5-3 are clear (have no effect on direction of the pins)
bits 2-0 are clear so that port pins are inputs
Value in port K data register:
bits 7-6 are written with the value to be output on port K pins 6 and 7
bits 5-3, when read contain the current value on the slave SPI pins
bits 2-0 contain current value on port K pins 0, 1 and 2

Special-purpose pins (SEL = 0) can be either read from or written to. If the pin is output-only, data can be written and read back via the data register. If the pin is input-only, the value in

the corresponding bit in the data register can be read. This value reflects the current logic level on the pin. Writing to this bit has no effect.

7.2 PULL-UP PORTS

The pull-up ports (E, F, G, K, and M) operate just like the basic ports, except they have some additional pull-up resistors to be configured. Each port has four configurable registers—a select register, direction register, pull-up register, and data register. You can enable the pull-up resistor of a particular pin by setting the corresponding bit in the pull-up register. Pull-up resistor can be individually enabled for each port pin, whether it is configured as a general I/O port or a special-purpose port. Figure 7.2 illustrates the internal logic of a pull-up port.

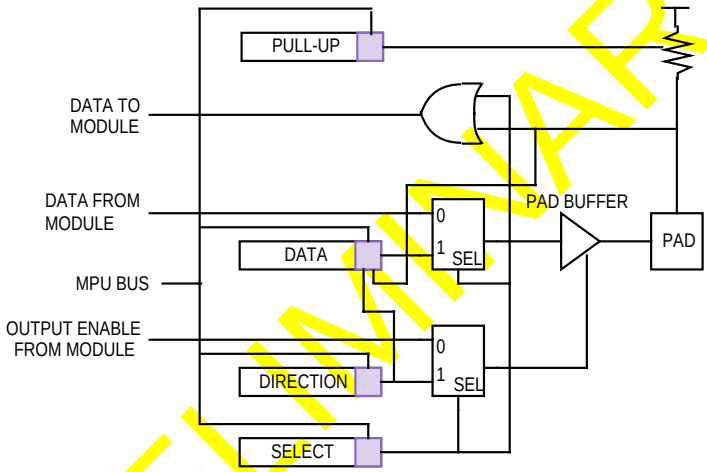


Figure 7-2. Pull-Up Port Operation

7.3 INTERRUPT PORT

The interrupt port (port D) is a basic port and a pull-up port, except it has additional interrupt capabilities. Figure 7-3 illustrates the internal logic of the interrupt port.

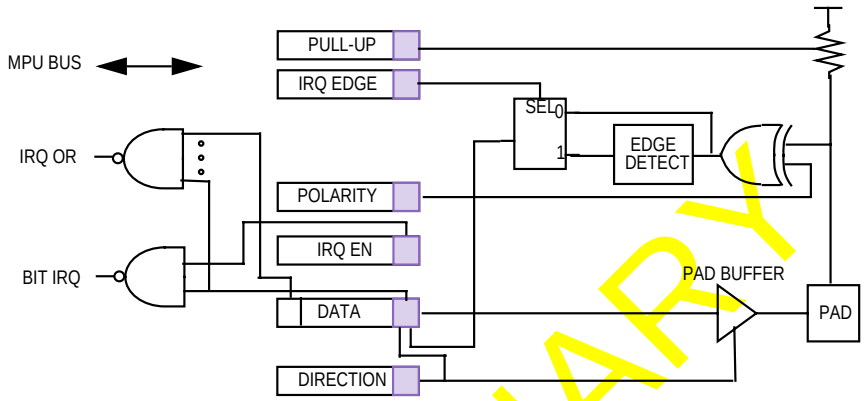


Figure 7-3. Interrupt Port Operation

Port D does not share its port pins with other peripheral modules. It is intended to be used as a general-purpose I/O port, interrupt-generating port, or keyboard-input port. Port D has six configurable registers—a direction register, data register, pull-up register, polarity register, INT enable register, and edge enable register.

The interrupt port generates nine interrupt signals. The individual port pins generate eight of the interrupts. These interrupt signals are known as INT[0:7]. Each interrupt signal either can be masked in the interrupt mask register or disabled in the port D INT enable register. The remaining interrupt is the logical-OR of these eight interrupt signals and the result is the keyboard (KB) interrupt input to the interrupt controller. The KB interrupt can also be enable or masked in the interrupt control module.

Each of the eight interrupt signals can be configured as an edge-triggered or level-sensitive signal in the edge enable register. Its polarity can be selected in the polarity register.

Note: Edge-protect circuitry on port D uses the system clock. Therefore, edge interrupts cannot generate wake-up interrupts when the chip is in sleep or doze mode.

7.4 DEFAULT PORT CONFIGURATIONS

At reset, or when the $\overline{\text{RESET}}$ signal is asserted, the DragonBall ports default to their reset configurations. You should examine the default value for each port carefully and, if necessary, reconfigure the port according to the requirements of your application. Table 7-1 contains the default port configurations.

Table 7-1. Reset Default Port Assignments

PORT NAME	DEFAULT RESET FUNCTION	PORT I/O TYPE	DEFAULT PORT RESET STATE	DEFAULT RESET PULL-UP STATUS	PIN NUMBER
PA7/A23	A23	Basic I/O	Output, Low	N/A	8
PA6/A22	A22	Basic I/O	Output, Low	N/A	7
PA5/A21	A21	Basic I/O	Output, Low	N/A	5
PA4/A20	A20	Basic I/O	Output, Low	N/A	4
PA3/A19	A19	Basic I/O	Output, Low	N/A	3
PA2/A18	A18	Basic I/O	Output, Low	N/A	2
PA1/A17	A17	Basic I/O	Output, Low	N/A	1
PA0/A16	A16	Basic I/O	Output, Low	N/A	144
PB7/D7	D7	Basic I/O	Input		103
PB6/D6	D6	Basic I/O	Input		104
PB5/D5	D5	Basic I/O	Input		105
PB4/D4	D4	Basic I/O	Input		106
PB3/D3	D3	Basic I/O	Input		108
PB2/D2	D2	Basic I/O	Input		109
PB1/D1	D1	Basic I/O	Input		110
PB0/D0	D0	Basic I/O	Input		111
PC7	N/A	N/A	N/A	N/A	N/A
PC6/$\overline{\text{WE}}$ (PCMCIA)	WE	Basic I/O	Output, High	N/A	124
PC5/$\overline{\text{DTACK}}$	DTACK	Basic I/O	Input	External Pull-up Required	121
PC4/$\overline{\text{IRQ7}}$	IRQ7	Basic I/O	Input	External Pull-up Required	31
PC3	N/A	N/A	N/A	N/A	N/A
PC2/$\overline{\text{LDS}}$	LDS	Basic I/O	Output, High	N/A	119
PC1/$\overline{\text{UDS}}$	UDS	Basic I/O	Output, High	N/A	116
PC0/MOCLK	MOCLK	Basic I/O	Input	N/A	27
PD7 (KB7)	PD7	Interrupt I/O	Input, Interrupt Disabled	Internal Pull-up Enabled	63
PD6 (KB6)	PD6	Interrupt I/O	Input, Interrupt Disabled	Internal Pull-up Enabled	64

Table 7-1. Reset Default Port Assignments (Continued)

PORT NAME	DEFAULT RESET FUNCTION	PORT I/O TYPE	DEFAULT PORT RESET STATE	DEFAULT RESET PULL-UP STATUS	PIN NUMBER
PD5 (KB5)	PD5	Interrupt I/O	Input, Interrupt Disabled	Internal Pull-up Enabled	65
PD4 (KB4)	PD4	Interrupt I/O	Input, Interrupt Disabled	Internal Pull-up Enabled	66
PD3 (KB3)	PD3	Interrupt I/O	Input, Interrupt Disabled	Internal Pull-up Enabled	67
PD2 (KB2)	PD2	Interrupt I/O	Input, Interrupt Disabled	Internal Pull-up Enabled	68
PD1 (KB1)	PD1	Interrupt I/O	Input, Interrupt Disabled	Internal Pull-up Enabled	69
PD0 (KB0)	PD0	Interrupt I/O	Input, Interrupt Disabled	Internal Pull-up Enabled	70
PE7/$\overline{\text{CSB3}}$	PE7	Pull-up I/O	Input	Internal Pull-up Enabled	81
PE6/$\overline{\text{CSB2}}$	$\overline{\text{CSB2}}$	Pull-up I/O	Output, High	Internal Pull-up Disabled	82
PE5/$\overline{\text{CSB1}}$	$\overline{\text{CSB1}}$	Pull-up I/O	Output, High	Internal Pull-up Disabled	83
PE4/$\overline{\text{CSB0}}$	$\overline{\text{CSB0}}$	Pull-up I/O	Output, High	Internal Pull-up Disabled	84
PE3/$\overline{\text{CSA3}}$	$\overline{\text{CSA3}}$	Pull-up I/O	Output, High	Internal Pull-up Disabled	85
PE2/$\overline{\text{CSA2}}$	$\overline{\text{CSA2}}$	Pull-up I/O	Output, High	Internal Pull-up Disabled	86
PE1/$\overline{\text{CSA1}}$	$\overline{\text{CSA1}}$	Pull-up I/O	Output, High	Internal Pull-up Disabled	87
PE0	N/A	N/A	N/A	N/A	N/A
PF7/A31	PF7	Pull-up I/O	Input	Internal Pull-up Enabled	17
PF6/A30	PF6	Pull-up I/O	Input	Internal Pull-up Enabled	16
PF5/A29	PF5	Pull-up I/O	Input	Internal Pull-up Enabled	14
PF4/A28	PF4	Pull-up I/O	Input	Internal Pull-up Enabled	13
PF3/A27	PF3	Pull-up I/O	Input	Internal Pull-up Enabled	12
PF2/A26	PF2	Pull-up I/O	Input	Internal Pull-up Enabled	11
PF1/A25	PF1	Pull-up I/O	Input	Internal Pull-up Enabled	10
PF0/A24	PF0	Pull-up I/O	Input	Internal Pull-up Enabled	9
PG7/RTCOUT	PG7	Pull-up I/O	Input	Internal Pull-up Enabled	18

Table 7-1. Reset Default Port Assignments (Continued)

PORT NAME	DEFAULT RESET FUNCTION	PORT I/O TYPE	DEFAULT PORT RESET STATE	DEFAULT RESET PULL-UP STATUS	PIN NUMBER
PG6/TIN1	PG6	Pull-up I/O	Input	Internal Pull-up Enabled	19
PG5/TOUT1	PG5	Pull-up I/O	Input	Internal Pull-up Enabled	20
PG4/TIN2	PG4	Pull-up I/O	Input	Internal Pull-up Enabled	21
PG3/TOUT2	PG3	Pull-up I/O	Input	Internal Pull-up Enabled	22
PG2/PWMOUT	PG2	Pull-up I/O	Input	Internal Pull-up Enabled	23
PG1/UART RXD	PG1	Pull-up I/O	Input	Internal Pull-up Enabled	25
PG0/UART TXD	PG0	Pull-up I/O	Input	Internal Pull-up Enabled	26
PJ7/$\overline{\text{CSD3}}$	$\overline{\text{CSD3}}$	Basic I/O	Output, High	N/A	72

7.5 PROGRAMMING MODEL

7.5.1 Port A

Port A is multiplexed with address lines A[16:23]. Depending on the system's specifications, you can decide which of these address lines to use. Unused address pins can serve as parallel I/O pins. After reset, port A defaults to the address function. There are three configurable registers that affect the behavior of each pin in this port—the select register, direction register, and data register. The corresponding bits in these registers correspond to each port pin.

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0x0000															
ADDR	0xFFFFF400															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	0	0	0	0	0	0	0	0	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
RESET	0x0000															
ADDR	0xFFFFF402															

DIR—Direction 0–7

These bits control the direction of the corresponding port pins. When a bit is high, the corresponding port pin is an output pin and while it is low, the corresponding port pin is an input pin. These bits are reset to 0 and have no effect on the pins while the SEL bits are low.

D—Data 0–7

These bits control or report the data on the pins when the corresponding SEL bits are high. If the DIR bits are high (pins configured as output), D[7:0] control the data to the pins. When the DIR bits are low (pins configured as input), D[7:0] report the signal level on the pins. The D bits may be read or written at any time. If a pin is configured as input-only, a write to the corresponding bit in this register does not affect the pin. These bits reset to 0.

SEL—Select 0–7

The select register allows you to individually select the function for each port pin. When you set a bit in this register, the corresponding port pin is configured as a general-purpose I/O. When a bit is clear, the corresponding port pin is configured as an address line A[16:23]. At reset, all bits in the select register are cleared.

7.5.2 Port B

Port B is multiplexed with data lines D7-D0. On reset, the data lines are connected to the pins. In an 8-bit only system, these pins can be used as general I/O. In this case, the boot-up sequence must configure port B as general I/O. This port is not affected by the BUSW pin.

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0x0000															
ADDR	0xFFFFF408															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	0	0	0	0	0	0	0	0	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
RESET	0x0000															
ADDR	0xFFFFF40A															

DIR—Direction 0–7

These bits control the direction of the corresponding port pins. When a bit is high, the corresponding port pin is an output pin and when it is low the corresponding port pin is an input pin. These bits reset to 0 and have no effect on the pins while the SEL bits are low.

D—Data 0–7

These bits control or report the data on the pins when the corresponding SEL bits are high. If the DIR bits are high (pins configured as output), D[7:0] control the data to the pins. While the DIR bits are low (pins configured as input), D[7:0] report the signal level on the pins. The

data bits may be read or written at any time. If a pin is configured as input-only, a write to the corresponding bit in this register does not affect the pin. These bits reset to 0.

SEL—Select 0–7

These bits select whether the data bus D0-D7 or general-purpose I/O signals are connected to the pins. When a bit is high, the corresponding port pin is configured as a general-purpose I/O and when they are low port B is a lower-byte data bus.

7.5.3 Port C

Port C is multiplexed with the M68000 bus control signals described in Table 7-2.

Table 7-2. 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	NMI
5	Bit 5	DTACK
6	Bit 6	$\overline{WE}$ (PCMCIA)
7	None	None

Port C has three configurable registers—the select register, direction register, and data register. Although these are 8-bit registers, only six bits are used to individually configure the six port pins. Bit 0 can be used only when the on-chip PLL is selected because the MOCLK function is needed to disable the PLL. When the on-chip PLL is enabled, Bit 0 can be used as a general I/O.

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	0	DIR6	DIR5	DIR4	0	DIR2	DIR1	DIR0	0	D6	D5	D4	0	D2	D1	D0
RESET	0x0000															
ADDR	0xFFFFF410															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	0	0	0	0	0	0	0	0	0	SEL6	SEL5	SEL4	0	SEL2	SEL1	SEL0
RESET	0x0000															
ADDR	0xFFFFF412															

DIR—Direction 0–7

These bits control the direction of the corresponding port pin. When a bit is high, the corresponding port pin is an output pin and when it is low the corresponding port pin is an input pin. These bits reset to 0 and have no effect on the pins while the SEL bits are low.

D—Data 0–7

These bits control or report the data on the pins while the associated SEL bits are high. If the DIR bits are high (pins configured as output), D[7:0] control the data to the pins. When the DIR bits are low (pins configured as input), D[7:0] report the actual signal level on the pins. The data bits may be read or written at any time. If a pin is configured as input-only, a write to the corresponding bit in this register does not effect the pin. These bits reset to 0.

SEL—Select 0–7

The select register allows you to individually select the function for each port pin. When you set a bit in this register, the corresponding port pin is configured as a general-purpose I/O. When a bit is clear, the corresponding port pin is configured as a bus control signal. At reset, all bits in the select register are cleared.

7.5.4 Port D

Port D has special features that allow it to be used as a keyboard input port. You can also use port D as a general-purpose I/O port or a general-purpose interrupt port. As with the other ports, each pin can be configured as an input or output on a bit-by-bit basis. When configured as an input, each pin can generate a CPU interrupt. These individual interrupt signals are presented to the interrupt controller module as INT[7:0]. In addition, a group interrupt can be generated. This interrupt is the OR (negative logic) of all pins on the port and it is presented to the interrupt controller as a keyboard (KB) interrupt.

Each interrupt can be configured either as a level-sensitive interrupt or an edge-triggered interrupt. The polarity of each interrupt can also be selected. Each pin is equipped with a switchable pull-up resistor.

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0X0000															
ADDR	0XFFFFF418															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	0	0	0	0	0	0	0	0
RESET	0XFF00															
ADDR	0XFFFFF41A															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0	IQEN7	IQEN6	IQEN5	IQEN4	IQEN3	IQEN2	IQEN1	IQEN0
RESET	0x0000															
ADDR	0xFFFFF41C															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	0	0	0	0	0	0	0	0	IQEG7	IQEG6	IQEG5	IQEG4	IQEG3	IQEG2	IQEG1	IQEG0
RESET	0x0000															
ADDR	0xFFFFF41E															

DIR—Direction 0–7

These bits control the direction of the corresponding port pin. When a bit is high, the corresponding port pin is an output pin and when it is low, the corresponding port pin is an input pin. At reset, these bits default to 0.

D—Data 0–7

These bits control or report the data on the pins. When the DIR bits are high, D[7:0] control the data to the pins. Data can be read or written to any bit. While the DIR bits are low, D[7:0] report the signal level on the pins. In this case, writing to a read-only bit does not affect the pin. Notice that the actual value on the pin is reported when a pin is read. Bits that are configured as edge-sensitive interrupts will read 1 when an edge is detected. The interrupt is cleared by writing a 1 to the set bit.

PU—Pull-Up 0–7

These bits enable the pull-up resistors on the port. When a bit is set in this register, the pull-up is enabled for the corresponding port pin. When a bit is cleared, the pull-up is disabled. At reset, all pull-ups are enabled for this port.

Note: Port D edge interrupts cannot generate wake-up events when the chip is in sleep mode.

POL—Polarity 0–7

These bits select the input signal polarity. When the bits are high, the input data is inverted before it is presented to the holding register and when the bits are low, the data is presented as is. Interrupts are active-high (or rising edge) while these bits are low. Interrupts are active-low (or falling edge) while these bits are high.

IQEN—Inerrupt Enable 0–7

These bits allow the individual interrupts to be presented to the interrupt controller block. These bits, when high, enable INT[7:0] interrupt generation. When the bits are low, INT[7:0] interrupt generation is disabled.

IQEG—Edge Enable 0–7

These bits, when high, enable edge-triggered interrupts. When the bits are low, level-sensitive interrupts are selected. The polarity of the edge (rising or falling) is selected by the POL bits.

7.5.5 Port E

Port E is multiplexed with seven chip-select signals that are described in the table below.

Table 7-3. Port E Bit Functions

BIT	PORT FUNCTION	OTHER FUNCTION
0	none	none
1	Bit 1	CSA1
2	Bit 2	CSA2
3	Bit 3	CSA3
4	Bit 4	CSB0
5	Bit 5	CSB1
6	Bit 6	CSB2
7	Bit 7	CSB3

There are only 6 pins available in this port. As with other ports, each pin can be individually configured for use as chip-selects or general-purpose I/O pin, as needed.

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0x0000															
ADDR	0xFFFFF420															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
RESET	0x8080															
ADDR	0xFFFFF422															

DIR—Direction 0–7

These bits control the directions for the associated port pin. When a bit is high, the corresponding port pin is an output pin and when it is low the corresponding port pin is an input pin. These bits reset to 0 and do not affect the pins while the SEL bits are low.

D—Data 0–7

These bits control or report the data on the pins. When the DIR bits are high, D[7:0] control the data to the pins. Data can be read from or written to any bit. When the DIR bits are low, D[7:0] report the signal level on the pins. In this case, writing to a read-only bit does not affect the pin. Notice that the actual value on the pin is reported when a pin is read. At reset, all data bits default to 0.

PU—Pull-Up 0–7

These bits enable the pull-up resistors on the port. When high, the pull-up resistors are enabled and when they are low they are disabled. The port E Bit 7 pull-up resistor is enabled after reset.

SEL—Select 0–7

The select register allows you to individually select the function for each port pin. When you set a bit in this register, the corresponding port pin is configured as a general-purpose I/O. When a bit is cleared, the corresponding port pin is configured as a chip-select signal.

7.5.6 Port F

Port F is multiplexed with address lines A[31:24]. Depending on the system's specification, you can decide which of these address lines to use outside the chip. Unused address pins can serve as parallel I/O pins.

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0x0000															
ADDR	0xFFFFF428															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
RESET	0xFFFF															
ADDR	0xFFFFF42A															

DIR—Direction 0–7

These bits control the direction of the corresponding port pin. When a bit is high, the corresponding port pin is an output pin and when it is low the corresponding port pin is an input pin. These bits reset to 0 and do not affect the behavior of the pins while the SEL bits are low.

D—Data 0–7

These bits control or report the data on the pins. When the DIR bits are high, D[7:0] control the data to the pins. Data can be read from or written to any bit. When the DIR bits are low, D[7:0] report the signal level on the pins. In this case, writing to a read-only bit does not affect the pin. Notice that the actual value on the pin is reported when a pin is read. At reset, all D bits default to 0.

PU—Pull-Up 0–7

These bits enable the pull-up resistors on the port. When high, the pull-up resistors are enabled and when they are low they are disabled. The pull-up resistors are enabled at reset.

SEL—Select 0–7

The select register allows you to individually select the function for each port pin. When you set a bit in this register, the corresponding port pin is configured as a general-purpose I/O. When a bit is cleared, the corresponding port pin is configured as an address line.

7.5.7 Port G

Port G is multiplexed with timer and serial communication signals. Refer to the timer, pulse modulator, real-time clock, and UART sections for a description of the signal functions. All 8 bits are implemented in the registers and each is connected to an I/O pin. As with other ports, each bit can be individually configured. However, PG7_RTCO is a dedicated input for a 32.768kHz real-time clock reference signal when the MOCLK bit (port C pin 0) is high. The signals are described in Table 7-4.

Table 7-4. 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

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0x00xx															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR	0xFFFFF430															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
RESET	0xFFFF															
ADDR	0xFFFFF432															

DIR—Direction 0–7

These bits control the direction of the corresponding port pin. When a bit is high, the corresponding port pin is an output pin and when it is low the corresponding port pin is an input pin. These bits reset to 0 and do not affect the behavior of the pins while the SEL bits are low.

D—Data 0–7

These bits control or report the data on the pins. When the DIR bits are high, D[7:0] control the data to the pins. Data can be read from or written to any bit. While the DIR bits are low, D[7:0] report the signal level on the pins. In this case, writing to a read-only bit does not affect the pin. Notice that the actual value on the pin is reported when a pin is read. At reset, all data bits default to 0.

Note: When PC0/MOCLK is high, the RTCOUT bit is disabled and PG7 becomes an input for the 32.768kHz real-time clock reference.

PU—Pull-Up 0–7

These bits enable the pull-up resistors on the port. When these bits are high, the pull-up resistors are enabled and when they are low they are disabled.

SEL—Select 0–7

The select register allows you to individually select the function for each port pin. When you set a bit in this register, the corresponding port pin is configured as a general-purpose I/O. When a bit is cleared, the corresponding port pin is configured as a peripheral interface. At reset, all bits in the select register are cleared.

7.5.8 Port J

Port J is multiplexed with eight chip-select signals that are described in the table below.

Table 7-5. Port J Bit Functions

BIT	PORT FUNCTION	OTHER FUNCTION
0	Bit 0	CSC0
1	Bit 1	CSC1
2	Bit 2	CSC2
3	Bit 3	CSC3
4	Bit 4	CSD0
5	Bit 5	CSD1
6	Bit 6	CSD2
7	Bit 7	CSD3

As with other ports, each bit can be individually configured for use as general-purpose IO or chip-selects.

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0X0000															
ADDR	0XFFFFF438															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	0	0	0	0	0	0	0	0	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
RESET	0X0000															
ADDR	0XFFFFF43A															

DIR—Direction 0–7

These bits control the direction of the corresponding port pin. When a bit is high, the corresponding port pin is an output pin and when it is low the corresponding port pin is an input pin. These bits reset to 0 and do not affect the behavior of the pins while the SEL bits are low.

D—Data 0–7

These bits control or report the data on the pins. When the DIR bits are high, D[7:0] control the data to the pins. Data can be read from or written to any bit. While the DIR bits are low, D[7:0] report the signal level on the pins. In this case, writing to a read-only bit does not affect

the pin. Notice that the actual value on the pin is reported when a pin is read. At reset, all data bits default to 0.

SEL—Select 0–7

The select register allows you to individually select the function for each port pin. When you set a bit in this register, the corresponding port pin is configured as a general-purpose I/O. When a bit is cleared, the corresponding port pin is configured as a chip-select.

7.5.9 Port K

Port K is multiplexed with signals that are related to the serial peripheral interfaces and PCMCIA. Table 7-6 describes these signals..

Table 7-6. 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

As with other ports, each bit can be individually configured as a general-purpose I/O or a peripheral interface.

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0X0000															
ADDR	0XFFFFF440															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
RESET	0X3F3F															
ADDR	0XFFFFF442															

DIR—Direction 0–7

These bits control the direction of the corresponding port pin. When a bit is high, the corresponding port pin is an output pin and when it is low the corresponding port pin is an input pin. These bits reset to 0 and do not affect the behavior of the pins while the SEL bits are low.

D—Data 0–7

These bits control or report the data on the pins. When the DIR bits are high, D[7:0] control the data to the pins. Data can be read from or written to any bit. While the DIR bits are low, D[7:0] report the signal level on the pins. In this case, writing to a read-only bit does not affect the pin. Notice that the actual value on the pin is reported when a pin is read. At reset, all data bits default to 0.

PU—Pull-Up 0–7

These bits enable the pull-up resistors on the port. When high, the pull-up resistors are enabled and when they are low they are disabled.

SEL—Select 0–7

The select register allows you to individually select the function for each port pin. When you set a bit in this register, the corresponding port pin is configured as a general-purpose I/O. When a bit is clear, the corresponding port pin is configured as a peripheral interface.

7.5.10 Port M

Port M is multiplexed with signals that are related to the interrupts and UART. Each bit has a selectable pull-up resistor that corresponds to it. Table 7-7 describes these signals.

Table 7-7. Port M Bit Functions

BIT	PORT FUNCTION	OTHER FUNCTION
0	Bit 0	CTS
1	Bit 1	RTS
2	Bit 2	IRQ 6
3	Bit 3	IRQ 3
4	Bit 4	IRQ 2
5	Bit 5	IRQ 1
6	Bit 6	PEN IRQ
7	Bit 7	UART GPIO

As with other ports, each pin can be individually configured, as needed. Notice that Pin 7 is a dedicated pin for use as a UART GPIO and cannot be used as a general-purpose IO pin.

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0	D7	D6	D5	D4	D3	D2	D1	D0
RESET	0X0000															
ADDR	0XFFFFF448															

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0	SEL7	SEL6	SEL5	SEL4	SEL3	SEL2	SEL1	SEL0
RESET	0XFF02															
ADDR	0XFFFFF44A															

DIR—Direction 0–7

These bits control the direction of the corresponding port pin. When a bit is high, the associated port pin is an output pin and when it is low the corresponding port pin is an input pin. These bits reset to 0 and do not affect the behavior of the pins while the SEL bits are low.

D—Data 0–7

These bits control or report the data on the pins. When the DIR bits are high, D[7:0] control the data to the pins. Data can be read from or written to any bit. When the DIR bits are low, D[7:0] report the signal level on the pins. In this case, writing to a read-only bit does not affect on the pin. Notice that the actual value on the pin is reported when a pin is read. At reset, all data bits default to 0.

PU—Pull-Up 0–7

These bits enable the pull-up resistors on the port. When high, the pull-up resistors are enabled and when they are low the pull-up resistors are disabled. The pull-ups are enabled on reset.

SEL—Select 0–7

The select register allows you to individually select the function for each port pin. When you set a bit in this register, the corresponding port pin is configured as a general-purpose I/O. When a bit is cleared, the corresponding port pin is configured as a peripheral interface.

SECTION 8
LCD CONTROLLER

The DragonBall liquid crystal display controller provides display data for an external LCD driver or LCD panel module. The following list summarizes the main features of the LCD controller:

- System Memory is Used as Display Memory, Which Eliminates the Need for Dedicated Video Memory
- Provides a Standard Panel Interface for Common LCD Drivers
- Supports Single Screen (Non-Split) Monochrome LCD Panels
- Fast Flyby, 16 Bits Wide, Burst-DMA Screen Refresh Transfers from System Memory
- Maximum Display Size is 1,024 x 512 and the Typical Non-Split Panel Sizes Are 320 x 240 and 640 x 200
- Supports 1-, 2-, or 4-Bit Wide LCD Data Bus Panel Interfaces
- Black and White or Four Out of Seven Simultaneous Gray Levels
- Hardware Blinking Cursor that is Programmable to a Maximum of 32 x 32 Pixels
- Supports Hardware Panning or Soft Horizontal Scrolling

The LCD controller fetches display data directly from system memory using periodic DMA transfer cycles. The bus bandwidth used by the LCD controller is low, which enables the MC68EC000 core to have sufficient computing bandwidth for other tasks.

8.1 ARCHITECTURE

As illustrated in Figure 8-1, the LCD controller consists of six main blocks:

- MPU interface registers
- Screen DMA controller
- Line buffer
- Cursor control logic
- Frame rate control
- LCD panel interface

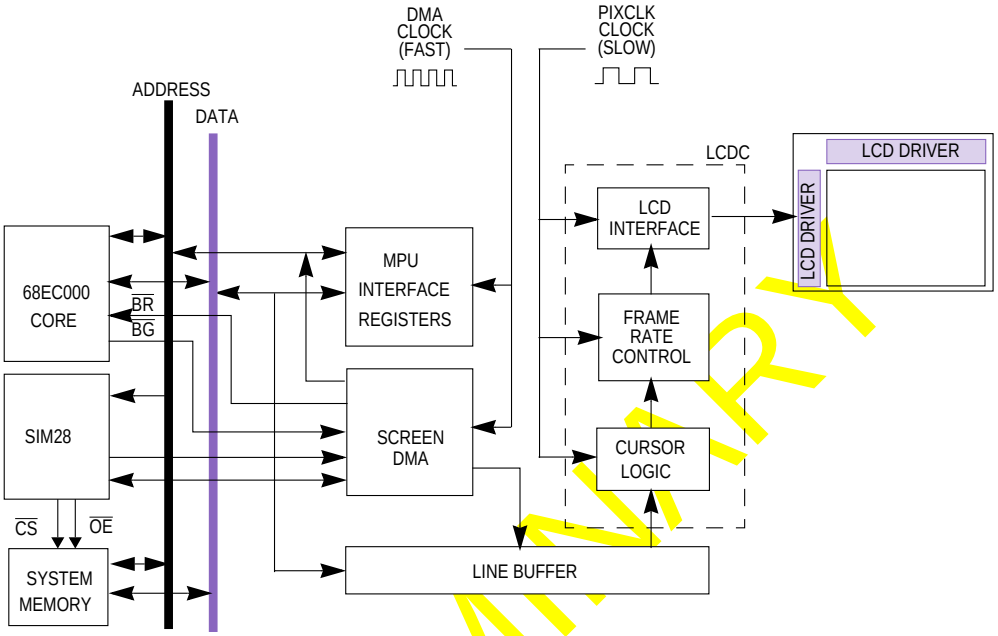


Figure 8-1. LCD Controller Block Diagram

8.1.1 MPU Interface

The MPU interface consists of control registers that enable you to operate different features of the LCD controller. This block is connected directly to the MC68EC000 core internal bus.

8.1.2 Screen DMA Controller

The screen DMA controller periodically generates a bus request ($\overline{BR}$) signal to the MC68EC000 core and when it receives a bus grant (BG) signal, it performs a 16- or 8-word memory burst to fill the line buffer. You can program the number of DMA clock cycles (1 to 16) per transfer to support systems with varieable memory speed requirements. When the DMA transfer begins, one clock cycle is added to the first data word has one clock cycle added to the transfer and the number of clock cycles selected. As shown in Figure 8-2 and Figure 8-3, this gives the initial chip-select memory more time without generating a clock cycle penalty on subsequent data transfers of the line fill.

8.1.3 Line Buffer

The line buffer collects display data from system memory during DMA cycles and outputs it to the cursor control logic. The input is synchronized with the fast DMA clock and the output is synchronized to the relatively slow LCD pixel clock (PIXCLK).

8.1.4 Cursor Logic

When enabled, the cursor control logic generates a block-shaped cursor on the display screen. You can adjust the cursor height and width anywhere between 1 and 31 pixel counts. The cursor can be black or it can be in reversed video. The blinking rate is adjustable when the blink enable bit is set.

8.1.5 Frame Rate Control

The frame rate control (FRC) is primarily used for a grayscale display and can generate a maximum of four gray levels out of seven density levels (0, 1/4, 5/16, 1/2, 11/16, 3/4, 1 as shown in Table 8-2). The density level corresponds to the number of times the pixel is being turned on when the display is refreshed frame by frame. Since the crystal formulations and driving voltage may vary, you can tune the visual gray quality by programming the gray palette-mapping register (GPMR) to obtain the best effect.

Blinking or flickering will occur if all LCD pixel cells are synchronized, so you must program two 4-bit numbers (XOFF and YOFF in the FRC offset register). As a general rule, select odd numbers that differ by two. Different inter-pixel crosstalk characteristics cause the optimal offset values to vary among different types of LCD panels (even from the same manufacturer).

8.1.6 LCD Panel Interface

The LCD panel interface logic packs the display data in the correct size and outputs it to the LCD panel data bus. You can program the polarity of the FRM, LP, and SCLK signals, as well as pixel data, for different types of LCD panels.

8.2 OPERATION

The LCD's DMA controller is a fly-by type, 16-bit wide, fast-data transfer machine. The LCD screen has to be refreshed continuously at a rate of approximately 50-70Hz, which means the image data in memory is periodically read and transferred to the corresponding pixels on the screen. You need a burst type and fly-by transfer to minimize bus obstruction caused by sharing the bus with the core. The refresh is divided into small packs of 8- or 16-word reads. Every time the internal line buffer needs data, it asserts the $\overline{BR}$ signal to request the bus from the core. Once the core grants the bus ($\overline{BG}$ is asserted), the DMA controller gets control of the bus signal and issues 8- or 16-word reads from memory. The read data is then internally passed to the next stage to generate the LCD timing (flyby). During the LCD access cycles, the chip-select logic asserts the output enable and chip-select signals for the corresponding system memory chip inside the system integration module. You can achieve the minimum bus bandwidth obstruction by using zero LCD access wait states (1 clock per access). See **Section 8.9 Calculating and Saving Bandwidth** for more information.

8.2.1 DMA Bus

As shown in Figure 8-2 and Figure 8-3, data is efficiently fetched from memory. Each burst is limited to 8/16 words, which reduces the possible latency for other peripherals, such as the interrupt controller. For example, the average latency for LCDCLK = 5MHz with a 16-word burst is approximately 2.4µs.

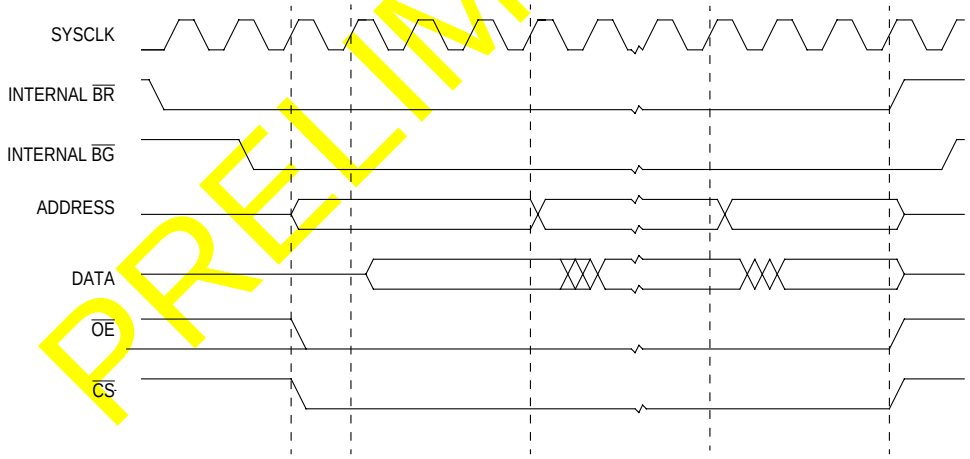


Figure 8-2. Three Clocks per LCD DMA Transfer (Two Wait States)

To operate, the DMA begins with an internal bus request from the DMA to the core and a bus grant to the DMA controller when the processor has given up the bus. The DMA cycle on the bus begins with chip-select assertion and addressing for the first data word to be transferred. A DMA cycle is indicated on the bus when a chip select is asserted with $\overline{AS}$ negated.

One clock cycle is added to the first data transfer of the DMA to allow for chip-select access time. For example, if the number of clock cycles selected for each DMA data word transfer is two, the first data word would transfer in three clocks. Subsequent data word transfers occur in two clock cycles to the end of the DMA burst. If the number of wait-states selected is zero, the DMA controller will complete each data transaction of the DMA burst in one clock with the exception of the first data word transfer, which occurs in two clocks as shown in Figure 8-7.

When the DMA controller has filled the line buffer, it negates the internal $\overline{BR}$ signal and relinquishes the bus to the core. If the core is in sleep mode, the internal bus is always granted to the DMA controller, which allows LCD images to be displayed while saving system power.

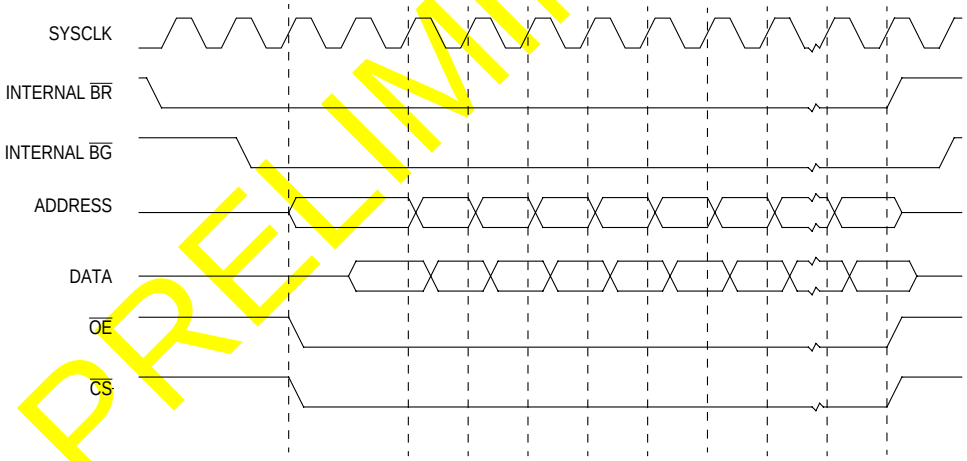


Figure 8-3. One Clock per DMA Transfer (0 Wait States)

8.2.2 Interfacing the LCD Controller with an LCD Panel

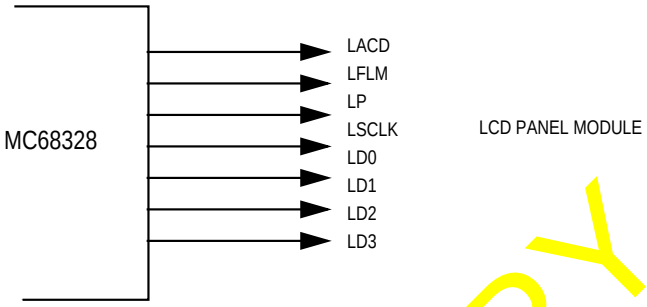


Figure 8-4. LCD Module Interface Signals

8.2.2.1 LCD DATA BUS SIGNALS. The LD3–LD0 signals output bus transfers pixel data to the LCD panel for display. Data is arranged differently on the bus depending on the LCD data-width mode selected, as shown in Figure 8-3. System software can also program the output pixel data to be inverted. See **Section 8.6.2 Polarity Configuration Register** for more information.

The LCD data bus uses LD0 to display pixel 0, 0. Some LCD panel manufacturers specify their LCD panel data bus in which data bit 3 of the panel displays pixel 0,0. For these panels, connections from the DragonBall's LD bus to the LCD panel data bus are reversed in bit significance (LD3 connects to panel data 0, LD2 to panel data 1, LD1 to panel data 2, and LD0 to panel data 3).

8.2.2.2 FIRST LINE MARKER SIGNAL. The LFLM signal indicates the start of a new display frame. It becomes active after the first line pulse of the frame and remains active until the next line pulse, at which point it deasserts and remains inactive until the next frame. You can program the LFLM signal to be active-high or active-low. See **Section 8.6.2 Polarity Configuration Register** for details.

8.2.2.3 LINE PULSE SIGNAL. The LP signal latches a line of shifted data into the LCD panel. It becomes active when a line of pixel data is clocked into the LCD panel and stays asserted for a duration of eight pixel clock periods. You can program the LP signal to be either active-high or active-low. See **Section 8.6.2 Polarity Configuration Register** for details.

8.2.2.4 SHIFT CLOCK SIGNAL. The LSCLK signal is the clock output that is synchronized to the LCD panel output data. Your system software can program the LSCLK signal to be either active-high or active-low. See **Section 8.6.2 Polarity Configuration Register** for details.

8.2.2.5 ALTERNATE CRYSTAL DIRECTION SIGNAL. The LACD output signal is toggled to alternate the crystal polarization of the panel. This provides an AC polarity change that prevents crystal degradation of the LCD panel caused by DC voltage. Your system software can program this signal to toggle at a 1 to 16 frame period. The alternate crystal direction (LACD or M) pin will toggle after the preprogrammed number of FLM pulses. Your system software can also program the ACD rate-control register (ACDRC) so that LACD will toggle once every 1–16 frames. The targeted number of frames is equal to the alternate code's 4-bit value plus one. The default value for ACDRC is zero, which means LACD will toggle on every frame. The LACD output signal is synchronized with the trailing or falling edge of the line pulse (LP) enclosed by FLM.

Table 8-1. ACDRC Value and Number of Cycles

ACDRC	NO OF CYCLES
0000	1
0001	2
0010	3
0100	5
1000	9
1111	16

8.2.3 LCD Panel Interface Timing

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

The LCD controller is designed to support most monochrome LCD panels. Figure 8-5 illustrates 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 signal enclosed by the LFLM signal marks the end of the first line of the current frame. Some LCD panels might use an active-low LFLM, LLP, or LSCLK signal,and reversed pixel data. To change the polarity of these signals, set the FLMPOL, LPPOL, SCLKPOL, and PIXPOL bits to 1. The LLP and LFLM signals' timing is similar for all panel modes that the LCD controller supports.

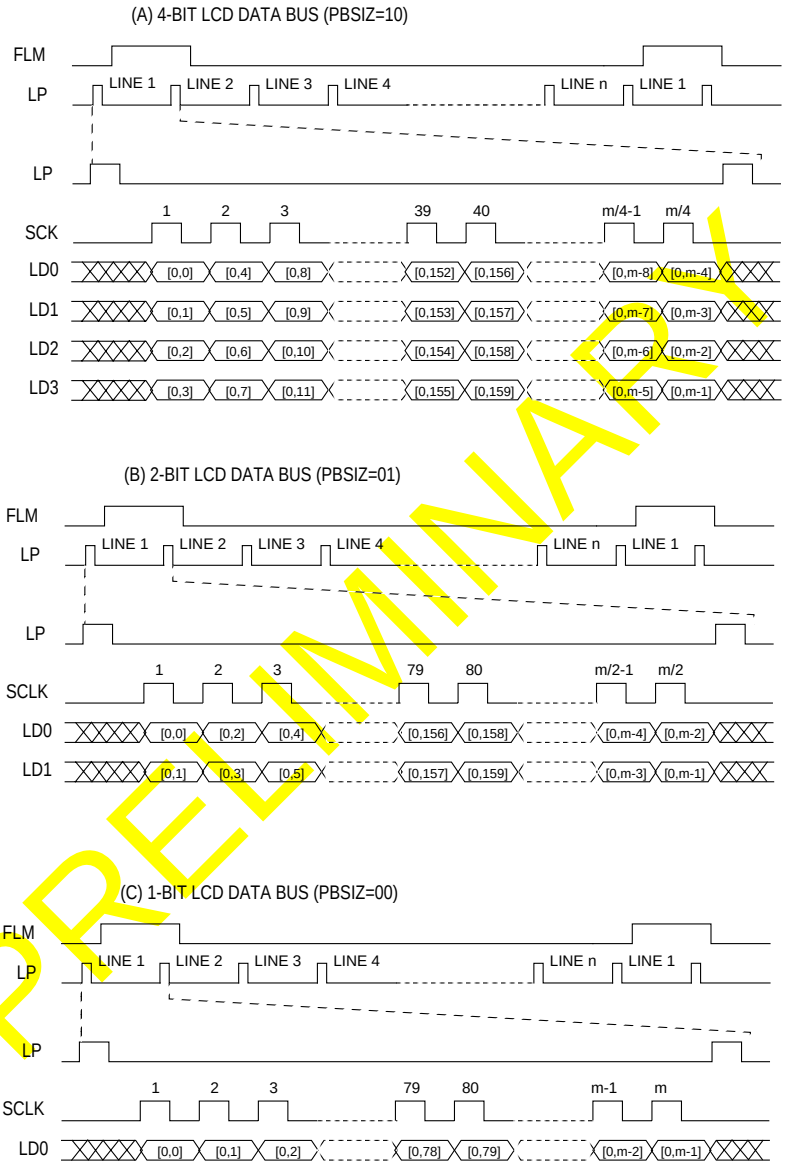


Figure 8-5. LCD Interface Timing for 4-, 2-, and 1-Bit Data Widths

8.2.4 DISPLAY CONTROL

The LCD controller signal drives single-scan monochrome STN LCD panels with a maximum of 1,024 x 512 pixels in the grayscale mode at a refresh rate of 60-70Hz. It is most efficient when the screen width is a multiple of the DMA controller's 16-bit bus width. Due to the limitations of LCD driver technology, large screens like 640 x 480 are usually organized in dual-scan format, which is not supported by DragonBall. The actual limit is the number of rows that require a high-driving voltage. The DragonBall's 4-bit LCD interface will drive up to 1,024 rows and a maximum of 512 columns.

8.2.4.1 LCD SCREEN FORMAT. The LCD panel's screen width and height can be programmed in the software. Figure 8-6 illustrates the relationship between part of a large graphics file displayed on screen versus the actual page in pixel counts.

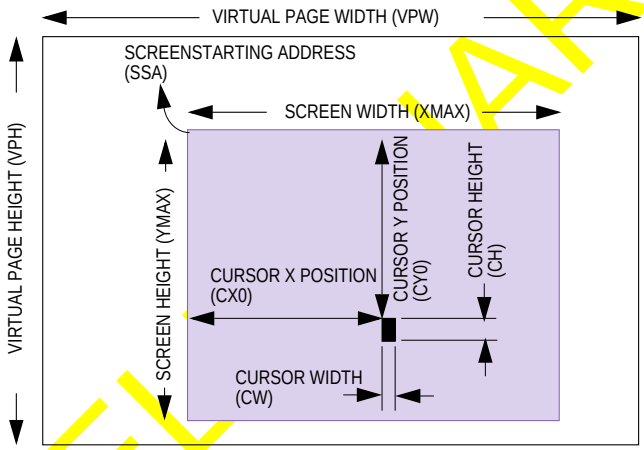


Figure 8-6. LCD Screen Format

The screen width (XMAX) and screen height (YMAX) registers specify the size of the LCD panel. The LCD controller starts scanning the display memory at the location pointed to by the screen starting address (SSA) register, which means the LCD panel will display the shaded area in Figure 8-6.

The virtual page width (VPW) and virtual page height (VPH) parameters specify the maximum page width and height, respectively. By changing the SSA register, a screen-sized window can be vertically or horizontally scrolled or panned anywhere inside the virtual page boundaries. The software must position the SSA properly so that the scanning logic's system memory pointer (SMP) does not stretch beyond VPW nor VPH. You should only use the VPH parameter for boundary checks. There is no VPH register that is internal to the LCD controller.

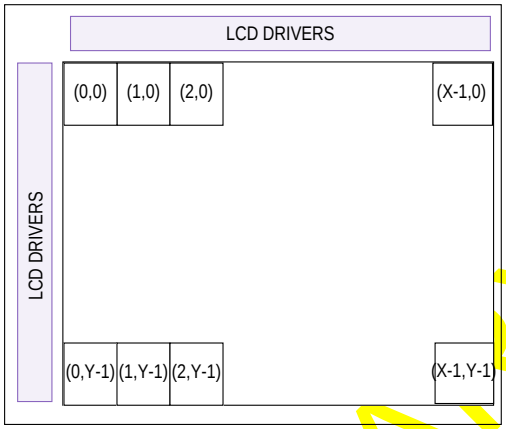
8.2.5 Cursor Control Logic

To define the position of the hardware cursor, the LCD controller 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) specifies the screen position of the current pixel data that is 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 are passed transparently, replaced with a full black, or complemented for reversed video. The cursor can be static or blinking, which is determined in the blink control register (BLKC). If you choose a static cursor, reversed video is optimal because it can be seen by inverting the original pixels. If you choose a blinking cursor, the original pixels and cursor display will periodically alternate.

8.2.6 Display Data Mapping

The LCD controller supports 1- or 2-bit per pixel graphics mode. In non-grayscale mode (1-bit per pixel), each bit in the display memory corresponds to a pixel in the LCD panel. The corresponding pixel on the screen is either completely on or off. In grayscale mode, (2-bit per pixel) the frame rate control circuitry in the LCD controller 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 four gray levels can be simultaneously displayed on the LCD screen.

The system memory data in 1- and 2- bit per pixel modes are mapped, as illustrated in Figure 8-7.



2-BIT PER PIXEL MODE

DISPLAY MAPPING

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

DISPLAY MAPPING

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 8-7. Mapping Memory Data on the Screen

8.2.7 Grayscale Generation

The LCD controller is configured to only drive single-scan monochrome LCD panels. It cannot handle color STN or TFT panels. However, grayscale generation can be selected by setting the GS bit in the panel interface configuration register (PICF). In grayscale mode, the number of data words for displaying the gray levels increases as opposed to black and white mode. Also, the line buffer must be filled before the next line is displayed. Therefore, the LCD pixel clocking, LCD frame refresh rate, line buffer fill, and line to line interval are affected by the selection of grayscale mode. The frame refresh rate and pixel clocking are determined by the pixel clock divider register (PXCD). Frame refresh is also a function of the LCD screen size in pixels. Therefore, the XMAX and YMAX registers are factors for calculating frame refresh. Figure 8-6 illustrates a formula and examples for.

8.2.8 Gray Palette Mapping

Using a proprietary frame-rate control (FRC) algorithm, the LCD controller can generate a maximum of four out of seven simultaneous gray levels by mapping the 2-bit data into four 3-bit gray codes. Then four out of seven bit densities are selected from the gray palette table.

Figure 8-7 illustrates how the 2-bit pixel data is mapped into 3-bit gray codes. The GMN bits are defined in the software-programmable gray palette mapping registers (GPMR). Each of the four 3-bit codes obtained from the first table select 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 8-2. Crystal formulations and driving voltages vary,which means the visual gray effect may or may not be linearly related to the frame rate. For certain graphics, a logarithmic scale, such as 0, 1/4, 1/2, and 1, might be more effective than a linearly spaced scale like 0, 5/16, 11/16, and 1. A flexible mapping scheme lets you optimize the visual effect for your particular panel or application.

Table 8-2. Gray Scale Code Mapping

CODE MAPPING	
Data	Gray code
00	G02G01G00
01	G12G11G10
10	G22G21G20
11	G32G31G30

Table 8-3. 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

8.2.9 Low-Power Mode

When the LCDON bit (register CKCON bit 7) is set to 0, the LCD controller enters a low-power mode by stopping the pixel clock before the next line-buffer-fill DMA. In this mode, there can be no additional screen DMA and display refresh operations. Before entering low-power mode, the DMA controller should be pointed to a blank screen image to prevent high logic levels on the LCD data bus from being driven to the LCD panel. The high logic levels could cause increased power consumption.

Some LCD panels may have a signal called PANEL_OFF that turns off the panel for low-power mode. The DragonBall does not support this signal. However, a parallel I/O pin used in conjunction with an external switching device can be used to perform this function. To turn off the panel using a parallel I/O pin, follow these steps:

1. Turn off the VLCD (+15V or -15V) by programming the I/O pin to turn off an external transistor or MOSFET. The transistor should be selected for a low conducting impedance to minimize voltage drop and power consumption.
2. Turn off the LCD controller by clearing the LCDON bit.
3. Your system software should ensure that the PANEL_OFF signal is deasserted before writing LCDON to 0.

When the LCD controller is switched back on (LCDON set to 1), DMA and screen-refresh activities will resume synchronously. To continue an LCD display from PANEL-OFF mode, follow these steps:

1. Turn on the LCD controller by setting the LCDON bit.
2. Delay 1-2ms.
3. Turn on the VLCD by programming the I/O pin to turn on the external transistor.

8.3 SYSTEM MEMORY CONTROL REGISTERS

8.3.1 Screen Starting Address Register

You should program the screen starting register (SSA) with the 32-bit screen starting address of the LCD panel. Notice that only the upper 31 bits of this register are programmable. Bit 0 is always 0, so that starting address is always at an even address. The LCD controller DMA's pixel data from system memory beginning at this address pointer stored in this register.

SSA

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIELD	SSA31	SSA30	SSA29	SSA28	SSA27	SSA26	SSA25	SSA24	SSA23	SSA22	SSA21	SSA20	SSA19	SSA18	SSA17	SSA16
R/W																
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	SSA15	SSA14	SSA13	SSA12	SSA11	SSA10	SSA9	SSA8	SSA7	SSA6	SSA5	SSA4	SSA3	SSA2	SSA1	SSA0
R/W																
RESET	0x00000000															
ADDR	0x(FF)FFFA00															

8.3.2 Virtual Page Width Register

The virtual page width (VPW) register specifies the virtual page width of the LCD panel in terms of byte count. VPW is a 9-bit value. However, VP0 is a fixed value of zero that aligns this register on 16-bit boundaries. The value of this register is the actual virtual page width in pixels divided by c (c is 16 for black and white displays and 8 for gray level displays).

VPW

BIT	7	6	5	4	3	2	1	0
FIELD	VP8	VP7	VP6	VP5	VP4	VP3	VP2	VP1
R/W								
RESET	0xFF							
ADDR	0x(FF)FFFA05							

8.4 SCREEN FORMAT REGISTERS

8.4.1 Screen Width Register

Th screen width register (XMAX) specifies the width of the LCD panel in pixels. On a line, pixels are numbered 0 to XMAX for a screen width of XMAX +1 pixels. XMAX+1 must be a multiple of 16.

XMAX

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	—						XM9	XM8	XM7	XM6	XM5	XM4	XM3	XM2	XM1	XM0
R/W																
RESET	0x03FF															
ADDR	0x(FF)FFFA08															

8.4.2 Screen Height Register (YMAX)

The screen height register (YMAX) specifies the height of the LCD panel in 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.

YMAX

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	—						YM9	YM8	YM7	YM6	YM5	YM4	YM3	YM2	YM1	YM0
R/W																
RESET	0x01FF															
ADDR	0x(FF)FFFA0A															

8.5 CURSOR CONTROL REGISTERS

8.5.1 Cursor X Position Register

CXP

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	CC1	CC0	—				CXP9	CXP8	CXP7	CXP6	CXP5	CXP4	CXP3	CXP2	CXP1	CXP0
R/W																
RESET	0x0000															
ADDR	0x(FF)FFFA18															

CC—Cursor Control

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

CXP—Cursor X Position 8–0

These bits represent the cursor's horizontal starting position X in pixel count (from 0 to XMAX).

8.5.2 Cursor Y Position Register

CYP

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	—							CYP8	CYP7	CYP6	CYP5	CYP4	CYP3	CYP2	CYP1	CYP0
R/W																
RESET	0x0000															
ADDR	0x(FF)FFFA1A															

CYP—Cursor Y Position 8–0

These bits represent the cursor's vertical starting position Y in pixel count (from 0 to YMAX).

8.5.3 Cursor Width & Height Register

CWCH

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	—			CW4	CW3	CW2	CW1	CW0	—			CH4	CH3	CH2	CH1	CH0
R/W																
RESET	0x0101															
ADDR	0x(FF)FFFA1C															

CW—Cursor Width 0–4

These bits specify the width of the hardware cursor in pixel count (from 1 to 31).

CH—Cursor Height 0–4

These bits specify the height of the hardware cursor in pixel count (from 1 to 31).

Note: The cursor is disabled if the CW or CH bits are set to zero.

8.5.4 Blink Control Register

BLKC

BIT	7	6	5	4	3	2	1	0
FIELD	BKEN	BD6	BD5	BD4	BD3	BD2	BD1	BD0
R/W								
RESET	0x7F							
ADDR	0x(FF)FFFA1F							

BKEN—Blink Enable

This bit indicates when the link-enable cursor stays on, instead of blinking. This bit defaults to zero.

1 = Blink enable.
0 = Blink disable.

BD—Blink Divisor 0–6

These bits indicate when the cursor will toggle once per specified number of internal frame pulses plus one. The half-period can be as long as two seconds.

8.6 LCD PANEL INTERFACE REGISTERS

8.6.1 Panel Interface Configuration Register

PICF

BIT	7	6	5	4	3	2	1	0
FIELD	—					PBSIZ1	PBSIZ0	GS
R/W								
RESET	0x00							
ADDR	0x(FF)FFFA20							

PBSIZ—Panel Bus Width 0–1

This bit indicates the size of the LCD panel bus.

00 = 1-bit.
01 = 2-bit.
10 = 4-bit.
11 = Unused.

GS—Grayscale

This bit is the grayscale mode bit. If this bit is set, it enables 4-grayscale level (2 bits per pixel) mode. Its default value is 0, which selects binary pixel (non grayscale) operation.

1 = Grayscale enable.
0 = No grayscale.

Note: The GS bit requires an LCD controller off/on sequence before any changes will occur.

8.6.2 Polarity Configuration Register

POLCF

BIT	7	6	5	4	3	2	1	0
FIELD	—				LCKPOL	FLMPOL	LPPOL	PIXPOL
R/W								
RESET	0x00							
ADDR	0x(FF)FFFA21							

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

This bit indicates the first-line marker polarity.

0 = Active high.
1 = Active low.

LPPOL—Line Pulse Polarity

This bit indicates the line-pulse polarity.

0 = Active high.
1 = Active low.

Note: The LCKPOL, FLMPOL, and LPPOL bits require an LCD controller off/on sequence before any changes will occur.

PIXPOL—Pixel Polarity

This bit indicates pixel polarity.

0 = Active high.
1 = Active low.

8.6.3 LACD (M) Rate Control Register

The LACD signal will toggle once every 1 to 16 FLM cycles based on the value specified in the LACD (M) rate control register (ACDRC). The actual number of FLM cycles is the value programmed plus one. The default value will toggle the LACD signal on each new frame, which is required by many LCD panel manufacturers. LACD is also referred to as the “M” signal by some LCD panel manufacturers.

ACDRC

BIT	7	6	5	4	3	2	1	0
FIELD	—				ACD3	ACD2	ACD1	ACD0
R/W								
RESET	0x00							
ADDR	0x(FF)FFFA23							

ACD—Alternate Crystal Direction Control 0–3

This bit represents the LACD toggle rate control code.

8.7 LINE BUFFER CONTROL REGISTERS

8.7.1 Pixel Clock Divider Register

PXCD

BIT	7	6	5	4	3	2	1	0
FIELD	—		PCD5	PCD4	PCD3	PCD2	PCD1	PCD0
R/W								
RESET	0x00							
ADDR	0x(FF)FFFA25							

PCD—Pixel Clock Divider 0–5

These bits indicate that the PIX clock from the PLL is divided by N (PCD5-0 plus one) to yield the actual pixel clock. Values 1-63 yield N = 2 to 64. If set to 0 (N = 1), the PIX clock will be used directly, which bypasses the divider circuit. The PIXCLK source is selected by the PCDS bit in the CKCON register.

8.7.2 Clocking Control Register

CKCON

BIT	7	6	5	4	3	2	1	0
FIELD	LCDON	DMA16	WS3	WS2	WS1	WS0	DWIDTH	PCDS
R/W								
RESET	0x40							
ADDR	0x(FF)FFFA27							

LCDCON

This bit controls the LCD controller.

0 = Disable the LCD controller.
1 = Enable the LCD controller.

Note: The internal LCD controller logic will be simultaneously switched off with the FLM pulse.

DMA16

This bit controls the length of the DMA burst.

0 = 8-word burst length.
1 = 16-word burst length.

WS—DMA Bursting Clock Control 0–3

This bit represents the number of clock cycles per DMA word access.

0000 = 1 clock cycle transfer	1000 = 9 clock cycle transfer
0001 = 2 clock cycle transfer	1001 = 10 clock cycle transfer
0010 = 3 clock cycle transfer	1010 = 11 clock cycle transfer
0011 = 4 clock cycle transfer	1011 = 12 clock cycle transfer
0100 = 5 clock cycle transfer	1100 = 13 clock cycle transfer
0101 = 6 clock cycle transfer	1101 = 14 clock cycle transfer
0110 = 7 clock cycle transfer	1110 = 15 clock cycle transfer
0111 = 8 clock cycle transfer	1111 = 16 clock cycle transfer

DWIDTH

This bit displays memory data width that indicates the size of the external bus interface.

0 = 16-bit memory.
1 = 8-bit memory.

PCDS—Pixel Clock Divider Source Select

0 = Selects the SYSCLK output of the PLL.
1 = Selects the PIXCLK output of the PLL.

Note: The PCDS bit requires an LCD controller off/on sequence before any changes will occur.

8.7.3 Last Buffer Address Register

LBAR

BIT	7	6	5	4	3	2	1	0
FIELD	—	LBAR7	LBAR6	LBAR5	LBAR4	LBAR3	LBAR2	LBAR1
R/W								
RESET	0X3E							
ADDR	0X(FF)FFFA29							

LBAR—Last Buffer Address 7–0

These bits represent the number of memory words required to fill one line on the display panel. The value is typically equal to the screen width in pixels divided by 16 for a black and white display or by eight for a grayscale display. For panning, add one more count in black and white mode and two for gray display.

8.7.4 Octet Terminal Count Register

OTCR

BIT	7	6	5	4	3	2	1	0
FIELD	OTC8	OTC7	OTC6	OTC5	OTC4	OTC3	OTC2	OTC1
R/W								
RESET	0X3F							
ADDR	0X(FF)FFFA2B							

OTC—Octet Terminal Count 8–1

These bits control the time interval from the end of the current line to the beginning of the next line. They allow the frame refresh rate to be finely adjusted. The register value must be greater than LBAR by four for a black and white display and eight for a grayscale display.

8.7.5 Panning Offset Register

POSR

BIT	7	6	5	4	3	2	1	0
FIELD	—				BOS	POS2	POS1	POS0
R/W								
RESET	0x00							
ADDR	0x(FF)FFFA2D							

BOS—Byte Offset

This bit is primarily used in the non-grayscale mode in conjunction with the POS bits. BOS must be set to zero for grayscale data.

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

Note: The cursor reference position must be adjusted separately from software when this register has been modified.

POS—Pixel Offset Code 0–2

These bits specify which of the eight 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. 000 implies that pixel 7, the first shifted out, will be the first to be displayed on every horizontal line in the current frame.

8.8 GRAYSCALE CONTROL REGISTERS

8.8.1 Frame-Rate Modulation Control Register

FRCM

BIT	7	6	5	4	3	2	1	0
FIELD	XMOD3-XMOD0				YMOD3-YMOD0			
R/W								
RESET	0XB9							
ADDR	0X(FF)FFFA31							

XMOD and YMOD—Frame-Rate Modulation Control

These bits modulate adjacent pixels at different time periods to avoid spatial flicker or jitter when using frame-rate control. These values must be optimized by manually fine-tuning the targeted LCD panel. See **Section 8.2.8 Gray Palette Mapping** for details.

8.8.2 Gray Palette Mapping Register

GPMR

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	0	G12	G11	G10	0	G02	G01	G00	0	G32	G31	G30	0	G22	G21	G20
R/W																
RESET	0X0173															
ADDR	0X(FF)FFFA32															

Gmn—

These bits represent the 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 8.2.8 Gray Palette Mapping** for details.

8.9 CALCULATING AND SAVING BANDWIDTH

LCD screen refresh is a periodic task. Therefore, the load that the LCD controller DMA burst cycles puts on the host data bus is an important consideration for the high-performance handheld system designer.

8.9.1 Bus Overhead Considerations

The following example describes the issues for estimating bandwidth overhead to the data bus. Consider a typical 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 the LCD controller must update one line of the screen is,

$$T_l = \frac{1}{60\text{Hz}} \times \frac{1}{240\text{lines}} = 69.4\mu\text{s} \quad \text{(EQ 1)}$$

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

$$T_{DMA} = \frac{320\text{pixels} \times 2\text{bitperpixel} \times 2\text{clock}}{16.67\text{MHz} \times 16\text{bitbus}} = 4.8\mu\text{s} \quad \text{(EQ 2)}$$

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

$$P_{DMA} = \frac{4.8\ \mu\text{s}}{69.4\ \mu\text{s}} = 6.92\% \quad \text{(EQ 3)}$$

SECTION 9
REAL-TIME CLOCK

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

9.1 FEATURES

The following list summarizes the features of the real-time clock:

- 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
- 32.768kHz or 38.4kHz operation

9.2 OPERATION

The real-time clock block diagram is illustrated in Figure 9-1.

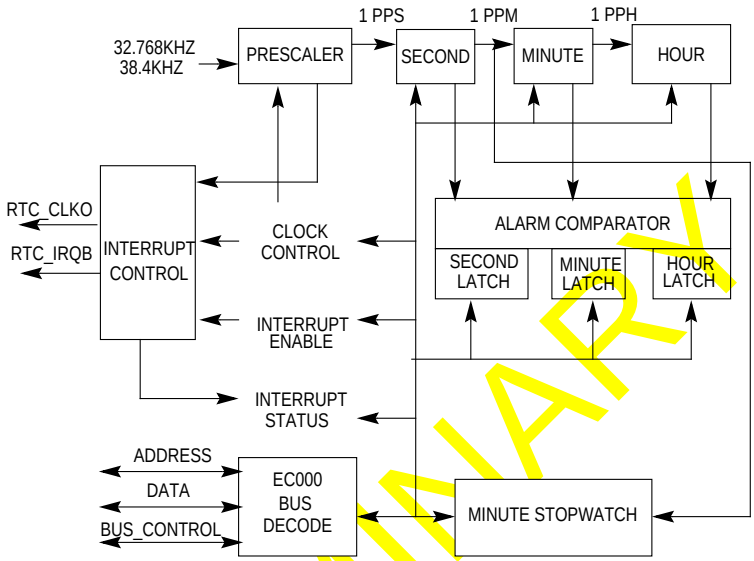


Figure 9-1. Real-Time Clock

9.2.1 Prescaler and Counter

The prescaler divides the 32.768kHz reference clock down to 1 pulse per second. An alternate reference frequency of 38.4kHz is also supported. The real-time clock hours/minutes/seconds (RTCHMS) register consists of three groups of bits that track the current time in hours, minutes, and seconds in a 24-hour format. The seconds and minutes counters are 6 bits long and the hours counter is 5 bits long. The prescaler stages are tapped to support several features. Periodic interrupts at 1Hz (or 1 second) and 1 minute, as well as the midnight-rollover interrupt are supported.

9.2.2 Alarm

You can set an alarm interrupt by setting the HOURS, MINUTES, and SECONDS fields in the RTC alarm (RTCALRM) register. An interrupt is enabled when the ALMEN bit in the RTC interrupt enable register (RTCIENR) is set. An interrupt is posted when the current time matches the time in the RTCALRM register.

9.2.3 Minute Stopwatch

The minute stopwatch performs a countdown with a one minute resolution. It generates an interrupt when the programmable length of time (in minutes) expires. For example, to save power, you can use the minute stopwatch to turn off the LCD display after a minute of idle.

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

9.3 PROGRAMMING MODEL

9.3.1 RTC Hours/Minutes/Seconds Register

The real-time clock hours/minutes/seconds (RTCHMS) register is not affected by the $\overline{\text{RESET}}$ signal. This allows the real-time clock to continue keeping time when the system is reset. At power-up, this register may contain a random value. Therefore, you may want to program it with the current time. The HOURS, MINUTES, and SECONDS fields can be read or written at any time. After a write, the current time assumes the newly written values. Unused bits read 0.

The real-time clock may be in the process of updating the hours, minutes, or seconds data, so the data value may be incorrect if a read and an update occur at the same time. When reading this register, you should make two reads and compare the results. If the reads do not compare, you make another read and use the newly-matched value. The following code fragment illustrates the preferred method. Hours, minutes, and seconds values are returned in the D0 register.

```
        move.l HMS,D0          make the first read
        cmp.l HMS,D0           make the second read and compare
        beq.s GOOD             if the reads compare, done
        move.l HMS,D0          bad compare, make another read
GOOD    rts                    return
```

RTCHMS

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIELD	—			HOURS					—		MINUTES					
R/W																
RESET	0x00000000															
ADDR	0x(FF)FFFB00															
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	—										SECONDS					
R/W																
RESET	0x00000000															
ADDR	0x(FF)FFFB00															

HOURS

When this field is read, the current hour can be set to any value between 0 and 23 (decimal). Writing a new value to this field will update the hour of the current time.

MINUTES

When this field is read, the current minute can be set to any value between 0 and 59 (decimal). Writing a new value to these bits will update the minute of the current time.

SECONDS

When this field is read, the current second and can be set to any value between 0 and 59 (decimal). Writing a new value to these bits will update the new second of the current time.

9.3.2 RTC Alarm Register

The real-time clock alarm (RTCALRM) register specifies the exact time that the real-time clock will generate an alarm interrupt to the processor. The HOURS, MINUTES, and SECONDS fields in this register can be read or written at any time. After a write, the alarm timer assumes the new values. Unused bits read 0.

RTCALRM

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIELD	—			HOURS					—		MINUTES					
R/W																
RESET	0x00000000															
ADDR	0x(FF)FFFB04															
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	—										SECONDS					
R/W																
RESET	0x00000000															
ADDR	0x(FF)FFFB04															

HOURS

When this field is read, the current setting of the alarm's hour can be set to any value between 0 and 23 (decimal).

MINUTES

When this field is read, the current setting of the alarm's minute can be set to any value between 0 and 59 (decimal).

SECONDS

When this field is read, the current setting of the alarm's second can be set to any value between 0 and 59 (decimal).

9.3.3 RTC Control Register

The real-time clock control (RTCCTL) register has two configurable bits that enable or disable real-time clock operation and select the frequency for the reference crystal clock.

RTCCTL

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	—							ENABLE		—	38.4	RESERVED				
R/W																
RESET	0x0000															
ADDR	0x(FF)FFFB0C															

ENABLE—RTC Enable

1 = Enable real-time clock.
0 = Disable real-time clock.

38.4—38.4kHz Reference Select

1 = Reference frequency is 38.4kHz.
0 = Reference frequency is 32.768kHz (default value).

Bits 4–0—Reserved

These bits are reserved and should be set to 0.

9.3.4 RTC Interrupt Status Register

The real-time clock interrupt status register (RTCISR) indicates the status of the various real-time clock interrupts. Each bit is set when its corresponding event occurs. You can clear the associated RTC interrupt by writing a 1 to the interrupt bit in this register. Since the real-time clock uses an external clock, it is not affected when the processor enters doze or sleep mode. This allows real-time clock interrupts to be used to wake up the processor.

RTCISR

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	—											1HZ	DAY FLAG	ALARM FLAG	MIN FLAG	SW FLAG
R/W																
RESET	0x0000															
ADDR	0x(FF)FFFB0E															

1Hz FLAG—1Hz Flag

When the 1Hz interrupt is enabled, this bit is set every second and an interrupt is generated to the processor.

1 = A 1Hz interrupt occurs.
0 = No 1Hz interrupt occurs.

DAY FLAG—Day Flag

When the 24-hour interrupt is enabled, this bit is set for every 24-hour clock increment (at midnight) and an interrupt is generated to the processor.

1 = A 24-hour rollover interrupt occurs.
0 = No 24-hour rollover interrupt occurs.

ALARM FLAG—Alarm Flag

When an alarm interrupt is enabled, this bit is set on a “compare” match between the real-time clock (current time) and the alarm register’s value. The alarm will reoccur every 24 hours. If a single alarm is required, you must clear the interrupt and disable the alarm interrupt in the RTC interrupt enable register.

1 = An alarm interrupt occurs.
0 = No alarm interrupt occurs.

MIN FLAG—Minute Flag

When a minute interrupt is enabled, this bit is set on every minute tick and an interrupt is generated to the processor.

1 = A minute tick occurs.
0 = No minute tick occurs.

SW FLAG—Stopwatch Flag

When a stopwatch interrupt is enabled, this bit is set when the stopwatch minute countdown experiences and an interrupt is posted to the processor.

1 = The stopwatch timed out.
0 = No stopwatch timed out.

9.3.5 RTC INTERRUPT ENABLE REGISTER

The real-time clock interrupt enable register (RTCENR) allows you to individually enable different real-time clock interrupts. The real-time clock provides five interrupts—a 1 second interrupt, 1 minute interrupt, 1 day (at midnight) interrupt, alarm interrupt, and a stopwatch interrupt. At reset, all RTC interrupts are disabled.

RTCIENR

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	—											1HZEN	24HR EN	ALM EN	MINEN	SWEN
R/W																
RESET	0x0000															
ADDR	0x(FF)FFFB10															

1HZEN—1Hz Interrupt Enable

This bit enables an interrupt at a 1Hz rate.

1 = A 1Hz interrupt is enabled.
0 = A 1Hz interrupt is disabled.

24HREN—24-Hour Interrupt Enable

This bit enables an interrupt at midnight rollover.

1 = A 24-hour interrupt is enabled.
0 = A 24-hour interrupt is disabled.

ALMEN—Alarm Interrupt Enable

This bit enables the alarm interrupt.

1 = An alarm interrupt is enabled.
0 = An alarm interrupt is disabled.

MINEN—Minute Interrupt Enable

This bit enables the minute tick interrupt.

1 = A minute tick interrupt is enabled.
0 = A minute tick interrupt is disabled.

SWEN—Stopwatch Interrupt Enable

This bit enables the stopwatch interrupt.

1 = A stopwatch interrupt is enabled.
0 = A stopwatch interrupt is disabled.

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

9.3.6 RTC STOPWATCH REGISTER

The real-time clock stopwatch (STPWTCH) programmable register contains the countdown value. When the stopwatch interrupt is enabled, this countdown value is decremented every minute and an interrupt is posted when the countdown value reaches -1.

STPWTCH

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	—										STOPWATCH COUNT					
R/W																
RESET	0x0000															
ADDR	0x(FF)FFFB12															

STOPWATCH COUNT

This field contains the stopwatch countdown value, which can be a maximum of 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 10
TIMERS

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.

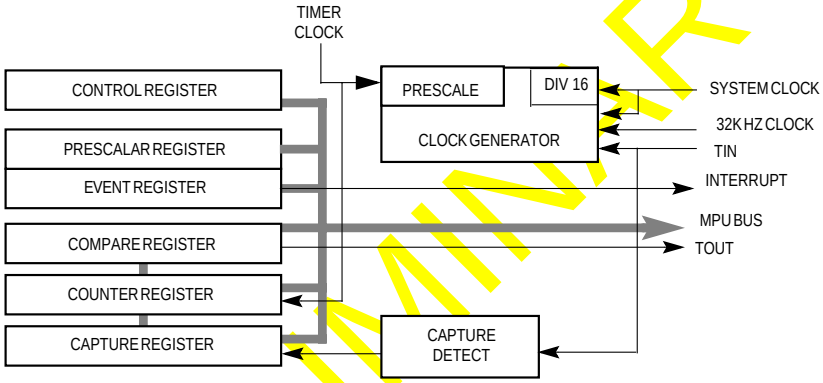


Figure 10-1. Timer Block Diagram

10.1 FEATURES

The following list contains the main features of the timers:

- 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

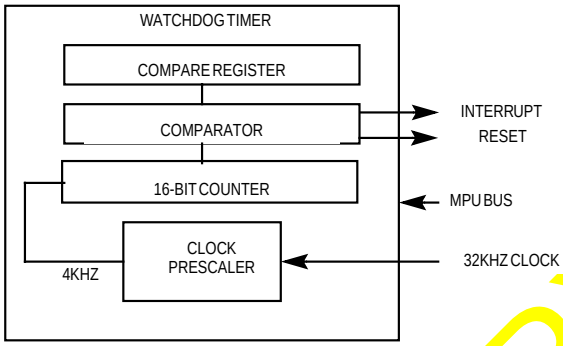


Figure 10-2. Watchdog Timer Block Diagram

The software watchdog timer has the following features:

- 16-bit counter and reference register
- Maximum period of 16.38 seconds
- 0.25-ms resolution
- Time-out causes system RESET or issues a maskable interrupt

10.2 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.

10.3 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.

When enabled, 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 enable bit is set (register WCR bit WDEN=1)by default at reset activating the watchdog timer. To prevent a reset from occurring, the WDEN bit must be cleared or the counter must be periodically written to prevent the counter from reaching the value of the reference register. 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 are cleared by writing to the counter.

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

10.4 SIGNAL DESCRIPTIONS

These signals are multiplexed with Port G I/O signals. Refer to **Section 7.1.9** for more information about enabling the timer functionality.

TIN1/PG6 AND TIN2/PG4

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.

TOUT1/PG5 AND TOUT2/PG3

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

10.5 PROGRAMMING MODEL

You can modify the general-purpose timer registers at any time.

10.5.1 General-Purpose Timer

This section describes the timer registers.

10.5.1.1 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	IRQ EN	CLKSOURCE			TEN

TIMER 1 ADDRESS: (FF)FFF600
TIMER 2 ADDRESS: (FF)FFF60C

RESET VALUE: $0000

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 of TINx and generate interrupt on capture
10 = Capture on falling edge of TINx and generate interrupt on capture
11 = Capture on rising or falling edges of TINx 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 or 38kHz 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.

10.5.1.2 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

PRESCALER

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

10.5.1.3 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

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.

10.5.1.4 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

10.5.1.5 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
COUNT															

TIMER 1 ADDRESS:
$(FF)FFF608 OR
TIMER 2 ADDRESS:
(FF)FFF614

RESET VALUE: $0000

COUNT
This is the current count value.

10.5.1.6 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

CAPT—Capture Event

While high, this bit indicates that a “capture” event occurred. Resetting this bit will clear the timer-capture register.

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

10.5.2 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).

10.5.2.1 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

COMPARE VALUE

When the counter counts up to the value in this register, an internal MC68328 reset is generated. 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.

10.5.2.2 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

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.

10.5.2.3 WATCHDOG-CONTROL/STATUS REGISTER (WCR).

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

ADDRESS: $(FF)FFF618 RESET VALUE: $0001

This register consists of three 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

W/DRST

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

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

SECTION 11
UNIVERSAL ASYNCHRONOUS RECEIVER/ TRANSMITTER

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.

11.1 FEATURES

The following list contains the main features of the UART:

- 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- and 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
- Eight 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 11-1 shows a high-level block diagram of the module.

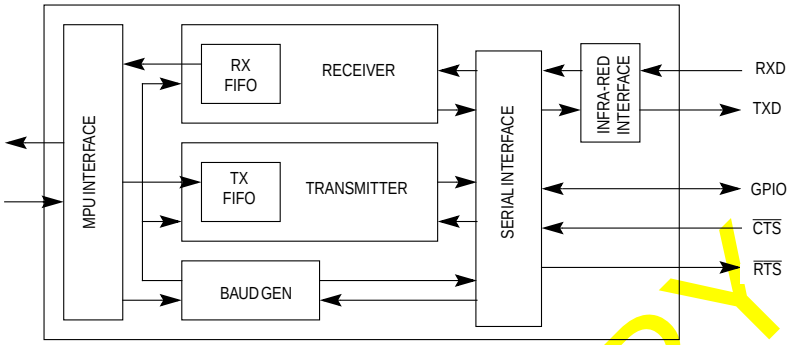


Figure 11-1. UART Block Diagram

11.2 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.

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.

$\overline{\text{CTS}}$—$\overline{\text{CLEAR TO SEND}}$

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.

11.3 SERIAL OPERATION

The UART module provides two operating modes: NRZ and IrDA. This section describes each operating mode.

11.3.1 NRZ Mode

NRZ (Non Return to Zero) mode is usually associated with RS-232. Each character is transmitted as a frame delimited by a Start Bit (0) at the beginning and a Stop Bit (1) at the end. Data bits are transmitted Least Significant Bit first and each bit is presented for a full bit time. If parity is used the parity bit is transmitted after the Most Significant Bit. Figure 8-2 shows an 8 bit ASCII "A" character (41 hex) with odd parity.

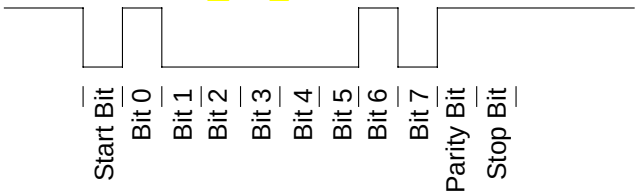


Figure 11-2. NRZ ASCII "A" Character with Odd Parity

11.3.2 IrDA Mode

IrDA operation uses character frames like NRZ mode but instead of driving ones and zeros for a full bit time zeros are transmitted as 3/16 bit-time pulses and ones remain low. The polarity of transmitted pulses and expected receive pulses can be inverted so that a direct connection can be made to external IrDA transceiver modules that use active low pulses. Figure 8-3 shows a character in IrDA mode.

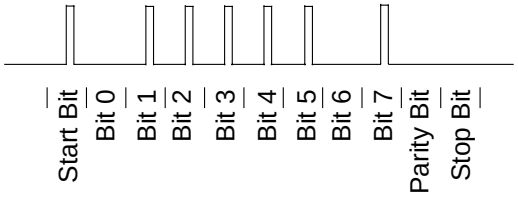


Figure 11-3. IrDA ASCII “A” Character with Odd Parity

11.4 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. $\overline{CTS}$ 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. $\overline{RTS}$ 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.

11.4.1 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.

$\overline{CTS}$ can control the serial data flow. If $\overline{CTS}$ is negated (high), the transmitter finishes sending the character in progress (if any), then waits for $\overline{CTS}$ 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 directly interfaces with popular IrDA transceivers.

11.4.2 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.

11.4.3 Baud Rate Generator

The baud generator provides the bit clocks to the transmitter and receiver blocks. It consists of a prescaler and a 2^n divider. Figure 8-4 is a block diagram of the Baud Rate Generator. The Baud Rate Generator Master Clock source can either be the system clock (SYSCLK) or it can be provided by the GPIO pin (input mode). By setting the Baud Source bit (bit 11, Baud Control Register) to 1, an external clock can directly drive the Baud Rate Generator. For synchronous applications, the UCLK pin can be configured to serve as an input or output for the 1x bit-clock.

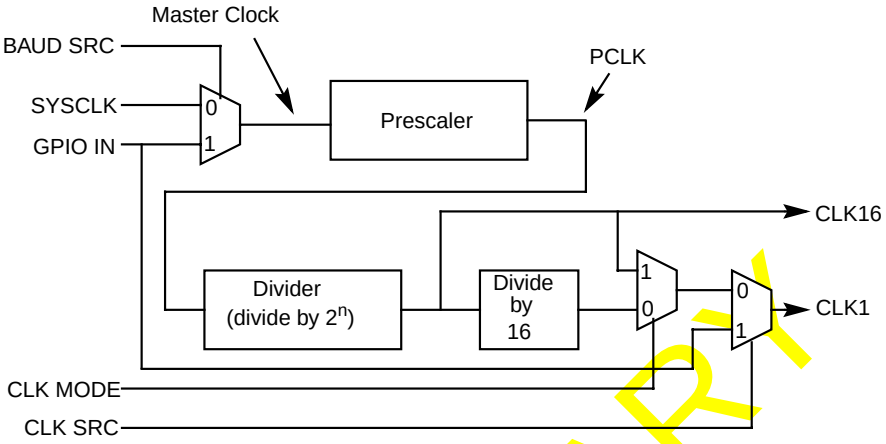


Figure 11-4. Baud Rate Generator

11.4.3.1 DIVIDER. The Divider is a 2^n binary divider with eight taps. Available taps are: 1, 2, 4, 8, 16, 32, 64, and 128. The selected tap is the 16x clock (CLK16) for the receiver. This clock is further divided by 16 to provide the 50% duty cycle 1x clock (CLK1) to the transmitter. While the CLK MODE bit (bit 12, UART Control Register) is high, CLK1 is directly sourced by the CLK16 signal.

11.4.3.2 PRESCALER. The baud generator can provide standard baud rates from many system clock frequencies. However it is best if the PLL is operated at the default multiplier (506, [P = $23, Q = $1]) with a 32.768 kHz crystal or multiplier of 432, (P = $1D, Q = $B) with a 38.400 kHz crystal. With a 32.768 kHz crystal standard baud clocks can be generated to within 0.05% accuracy. With a 38.400 kHz crystal the baud clocks are generated with 0% error. Table 8-1indicates the values to use in the Baud Register for these system frequencies.

Note: While CLK MODE is 1 (1X mode), the divide by 16 block is bypassed so the generated clock will be 16 times faster.

11.4.4 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.

11.4.5 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 EN	RX EN	TX EN	CLK MODE	PARITY EN	ODD EVEN	STOP	8/7	GPIO DELTA EN	CTS DELTA EN	RX FULL EN	RX HALF EN	RX RDY EN	TX EMPTY EN	TX HALF EN	TX AVAIL EN

ADDRESS: $(FF)FFF900

RESET VALUE: $0000

UARTEN—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 UARTEN and RXEN bits and perform a word-read operation on the receiver register to initialize the FIFO and character-status bits.

RXEN—Receiver 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

TXEN—Transmitter 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

CLKMODE—Receiver Clock Mode

This bit controls the 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

PAREN—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

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

GPIODELTAEN—General-Purpose I/O 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

CTSDELTAEN—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

RXFULLEN—Receiver 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

RXHALFEN—Receiver 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

RXRDYEN—Receiver 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

TXEMPTYEN—Transmitter 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

TXHALFEN—Transmitter 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

TXAVAILEN—Transmitter Available Enable

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

11.4.6 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

Table 11-1. Baud Rate Divider/Prescaler Values

BAUD RATE	DIVIDER	PRESCALER (HEX)
115200	0	$38
57600	1	$38
28800	2	$38
14400	3	$38
38400	0	$26
19200	1	$26
9600	2	$26
4800	3	$26
2400	4	$26
1200	5	$26
600	6	$26
300	7	$26

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

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

11.4.7 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. The character status bits are upgraded and valid with each data character. Status and data must be read as a 16-bit word.

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

ADDRESS: $(FF)FFF904

RESET VALUE: $0000

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.

11.4.8 Transmitter Register

This register controls the transmitter operation and reports the status.

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

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.

11.4.9 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	RESERVED	RTS CONT	RTS	IRDA EN	IRDA LOOP	RXPOL	TXPOL	UNUSED

ADDRESS:
$(FF)FFF908

RESET VALUE: $0000

Bits 8–11—Reserved

These bits are reserved for factory testing and should be set to 0.

TXPOL

This bit inverts the polarity of the UART transmit data. In IrDA mode, some infra-red transceivers require that the data be inverted before being presented to the input of optical transmit device.

0=Normal
1=Inverted

RXPOL

This bit inverts the polarity of the UART receive data. In IrDA mode, some infra-red transceivers invert the data being presented to the UART receive input.

0=Normal
1=Inverted

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

IRDAEN—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

RXPOL—Receiver Polarity

This bit controls the expected polarity of the received data.

0 = Normal polarity 1 = Idle (0 = IrDA idle)
1 = Inverted polarity 0 = idle (1 = IrDA idle)

TXPOL—Transmitter Polarity

This bit controls the expected polarity of the transmitted data.

0 = Normal polarity 1 = Idle (0 = IrDA idle)
1 = Inverted polarity 0 = Idle (1 = IrDA idle)

UNUSED

These bits are unused and read 0.

SECTION 12
SERIAL PERIPHERAL INTERFACE—MASTER

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.

The SPI master (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.

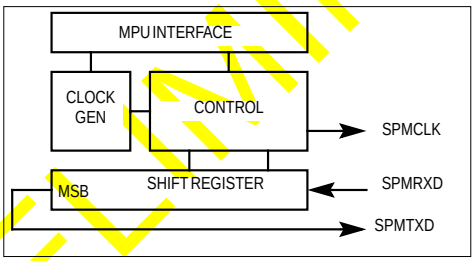


Figure 12-1. SPI Master Block Diagram

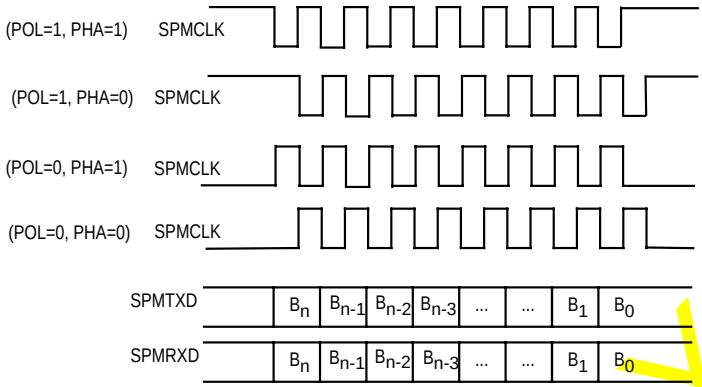


Figure 12-2. Master SPI Operation

12.1 OPERATION

12.1.1 Operation Within the 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
- 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.

12.1.2 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 12-2). 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.

12.2 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.

12.3 SPI MASTER 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.

12.3.1 SPI Master 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			RES	RES	RES	SPIM EN	XCH	SPIM IRQ	IRQ EN	PHA	POL	BIT COUNT			

ADDRESS: (FF)FFF802 RESET VALUE: $0000

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

SPIMEN—SPI Master Enable

This bit enables the SPIM. The enable should be asserted before initiating an exchange and should be negated after the exchange is complete. This bit must be set before data can be written into the SPIM data register.

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 remains set until the transfer is completed.

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—SPI Master Interrupt Request

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. This bit can also be polled with IRQEN bit clear.

0 = No interrupt posted
1 = Interrupt posted

IRQEN—Interrupt Request Enable

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—Phase

This bit controls the SPMCLK phase shift.

0 = Normal phase
1 = Shift advance to opposite phase

POL—Polarity

This bit controls the SPMCLK polarity.

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

BIT COUNT

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

0000 = 1 bit transfer
1111 = 16 bit transfer

12.3.2 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

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.

SECTION 13
SERIAL PERIPHERAL INTERFACE—SLAVE

The slave serial peripheral interface (SPI) operates as an externally clocked slave, allowing the MC68328 processor to interface with external master devices (for example, FLEX™ 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.

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 13-1 is a block diagram of the slave SPI.

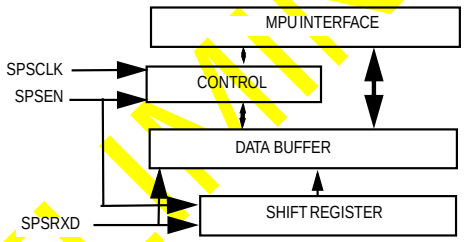


Figure 13-1. SPI Block Diagram

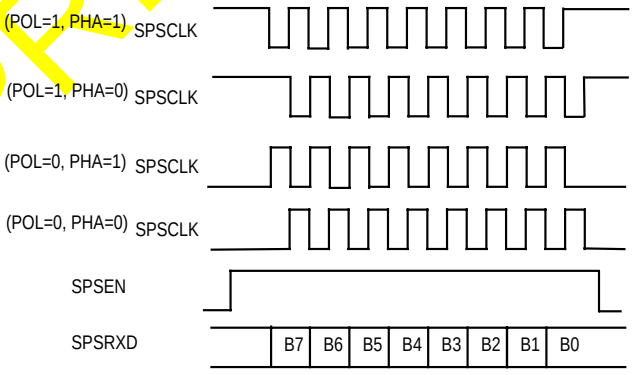


Figure 13-2. SPI Slave Operation

13.1 OPERATION

Users first initialize the SPI slave (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.

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.

13.2 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.

13.3 SPI SLAVE 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.

13.3.1 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	IRQ EN	EN POL	DATA RDY	OV WR	PHA	POL	SPIS EN	DATA							

ADDRESS:
$(FF)FFF700:

RESET VALUE: $0000

SPISIRQ—Slave SPI Interrupt Request

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 bit is cleared by writing a one to it. It is a leve 6 IRQ.

0 = No interrupt posted
1 = Interrupt posted

IRQEN—Slave SPI Interrupt Request Enable

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

0 = Interrupt disabled
1 = Interrupt enabled

ENPOL—Enable Polarity

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

OVWR—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—Phase

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—Polarity

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—Slave SPI Enable

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 14
PULSE-WIDTH MODULATOR

The pulse-width modulator (PWM) provides high-quality sound generation and accurate motor control. This section describes the PWM block. 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 signal and the free-running counter resets when its value matches the period value. The width comparator resets the output signal when the counter value matches. With a suitable external low-pass filter, the PWM can be used as a digital-to-analog converter.

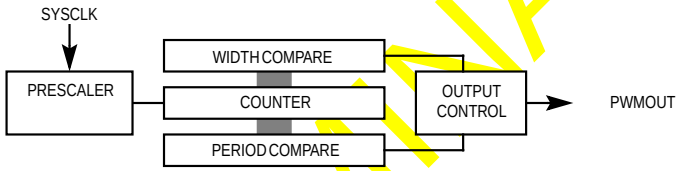


Figure 14-1. PWM Block Diagram

The PWM outputs a pulse stream of varying frequency and duty-cycle as determined by the perid and width registers. This makes the PWM ideal for motor control. The PWM output may alternately be used to generate tones with a simple external low-pass filter connected to the output pin. It is also possible to produce high-quality, digitally generated voice. 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.

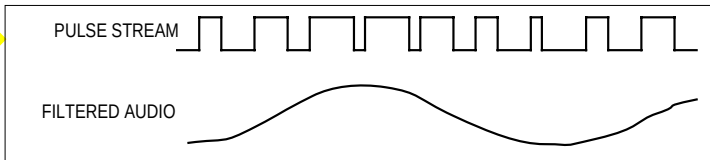


Figure 14-2. PWM Generating Audio

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.

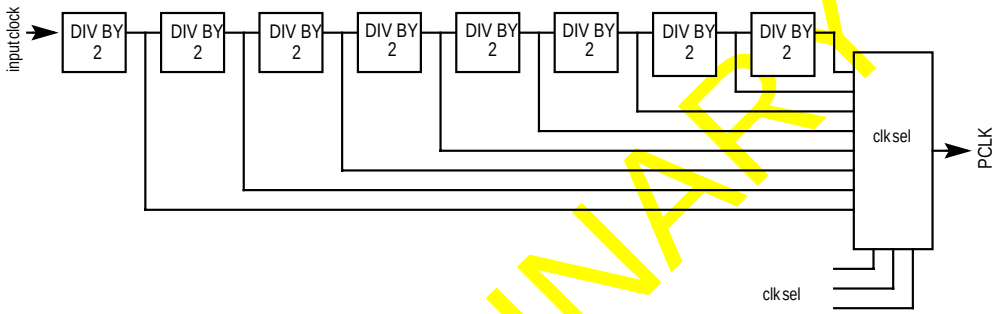


Figure 14-3. PWM Prescaler

14.1 PROGRAMMING MODEL

This section describes the PWM module registers and control bits.

14.1.1 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	IRQ EN	UNUSED					LOAD	PIN	0	POL	PWM EN	0	CLKSEL		

ADDRESS: $(FF)FFF500

RESET VALUE: $0000

PWMIRQ—Pulse-Width Module Interrupt Request

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—Interrupt Request Enable

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

0 = PWM interrupt disabled
1 = PWM interrupt enabled

LOAD—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—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—Polarity

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—Pulse-Width Module Enable

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—Clock Select

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

000 = Divide by 4	001 = Divide by 8	010 = Divide by 16
011 = Divide by 32	100 = Divide by 64	101 = Divide by 128
110 = Divide by 256	111 = Divide by 512	

14.1.2 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

PERIOD—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).

14.1.3 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

WIDTH—Width

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

14.1.4 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

COUNT—Count

This is the current count value.

SECTION 15
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.

The PLL is a flexible clock source for the MC68328. It provides a crystal-controlled master clock at frequencies from 13 MHz 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 master clock frequency. For the default master clock frequency of 16.58 MHz, the slowest system clock frequency is 1.0363 MHz. This block, in conjunction with the power-control block, provides an efficient power-control mechanism for the MC68328 processor (see Figure 3-1 below).

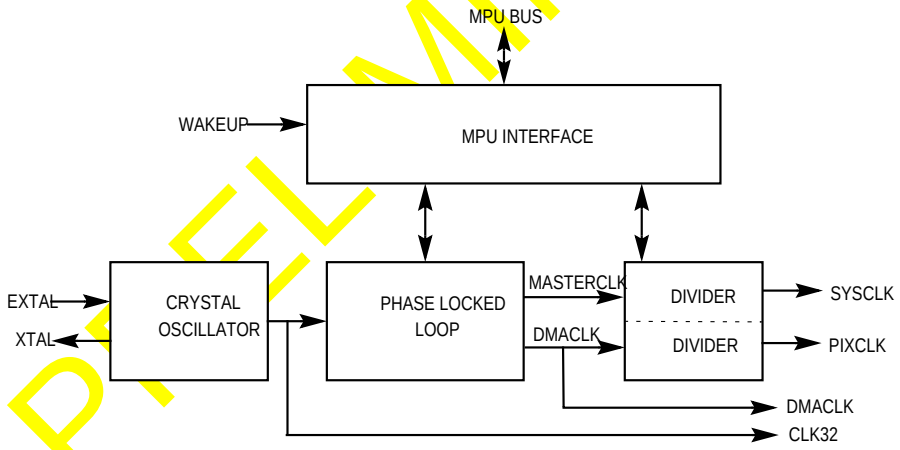


Figure 15-1. PLL Block Diagram

15.1 PROGRAMMING MODEL

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

15.1.1 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)FFF200 Reset Value: $2410

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 3.2.5** for a description of the preferred method for PLL shutdown. Once the PLL is disabled, only a wake-up interrupt or reset can re-enable it.

1 = PLL disabled
0 = PLL enabled

15.1.2 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

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 **Section 15.2 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.

15.2 PLL OPERATION

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

15.2.1 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 0.05%.

15.2.2 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.

15.2.3 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.

15.2.4 Change of Frequency

To change the VCO frequency, use the sequence below. This fragment assumes all peripherals have been disabled and the CPU is operating at the highest possible frequency (SYSCLK SEL = 100). NEWFREQ is the new frequency value (P and Q values) to be programmed. This routine enables Timer 2 to wake up the PLL after two CLK32 “ticks”. When the PLL wakes up, it will be at the new frequency. The interrupt service routine for the temporary Timer 2 interrupt should just clear the Timer 2 interrupt and return. This code was written for clarity, not efficiency.

```
        NEWFREQ equ somevalue          ;P and Q value of new frequency
        PLLCONTROL equ $FFF200         ‘PLL Control Register
        PLLFREQ equ $FFFF202           ;PLL Frequency Control Register
        T2COMPARE equ $FFF610          ;Timer 2 Compare Value Register
        T2CONTROL equ $FFF60C          ;Timer 2 Control Register
        IMR equ $FFF304                ;Interrupt Mask Register

        move.1 IMR,-(SP)               ;save the Interrupt Mask register
        move.1 #$fffffffd,IMR          ;enable ONLY Timer2 interrupt
        move.w #$0001,T2COMPARE        ;set compare value to 2
        move.w #$0119,T2CONTROL        ;enable Timer 2 with CLK32 source
SYNC    btst.b #47,PLLFREQ             ;synchronize to CLK32 high level
        beq.s SYNC                     ;CLK32 is still not high, go back
        move.w #NEWFREQ,PLLFREQ        ‘load the new frequency
        ori.b #$8, PLLCONTROL+1        ;disable the PLL (in 30 clocks)
        stop #$2000                    ;stop, enable all interrupts
;the PLL shuts down here and waits for the Timer 2 interrupt
```

```
;interrupt service for Timer 2 occurs here
       move.w (SP)+,IMR              ;restore the Interrupt Mask Register
       rts                           ;PLL is now at the new frequency
;The PLL has reacquired lock and SYSCLK is stable
```

The master frequency should only be changed during an early phase of the boot-up sequence.

15.2.5 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, A1      point to the Freq Sel Register
               lea #FFF200, A0       also point to PLL Control Register
WAIT           move.w (A1),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
```

Note: If setting the DISPLL bit does not put the MC68328 into sleep mode, it is probably because an interrupt was asserted and was not disabled as a wake-up event in the Interrupt Wake-Up Enable Register. An interrupt can be a wake-up event and still be masked from causing a CPU interrupt. See **Section 2.3.3.5.**

15.3 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.

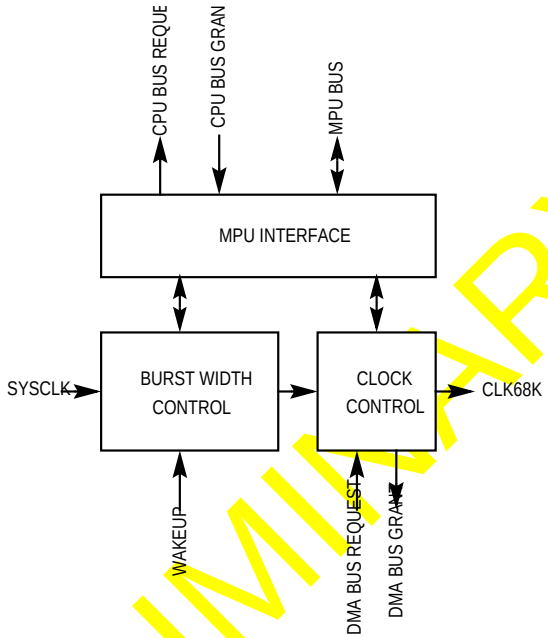


Figure 15-2. Power Control Module

15.3.1 Description

Figure 15-2 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 a DMA access is in progress, the CPU will wait until the DMA controller has completed its access before interrupt processing begins.

Figure 15-3 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.

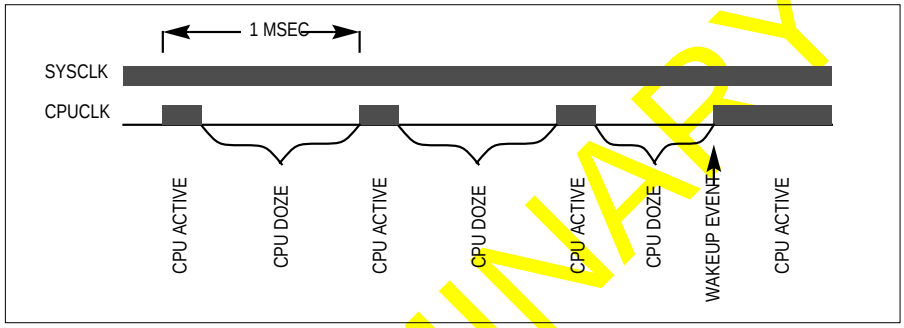


Figure 15-3. Power Control Operation

15.3.2 MPU Interface

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

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

PC EN

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 reenable this bit to reenter power-save operation. This bit resets to zero. In association with the width bits, this bit acts as a throttle from doze to CPU-active.

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

STOP

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 and is cleared on wake-up events.

1 = Stop CPU clock (enter doze mode)
0 = Normal CPU clock bursts

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 clearing the PC EN bit. When an interrupt disables the power controller, these bits are not changed. Users should enter the doze mode by setting the PCEN or STOP bit after an interrupt has been serviced. Typically, it is the STOP bit that is set.

15.3.3 Operation

This section describes how to use the variable duty-cycle or STOP bit in the power controller to enter doze mode.

15.3.3.1 NORMAL OPERATION

When the MC68328 processor begins operation after reset, the power controller is disabled and the MC68EC000 internal clock runs continuously. To reduce the power consumed by the MC68EC000, the power controller is enabled when the STOP or PCEN bit is set. The value in the WIDTH register determines the duty cycle of the clock-bursts that are applied to the MC68EC000 when PCEN is set. 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.

The CLKO pin is an external reference of the internal MC68EC000 clock. If the external system does not require CLKO, it can be disabled by clearing the CLKEN bit in the PLL control register further reducing the normal operation power consumption.

15.3.3.2 DOZE OPERATION

The MC68EC000 clock can be disabled for extended periods by setting the STOP bit or WIDTH register bits to 00000 and setting the PC EN bit to 1. The MC68EC000 clock is enabled again 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.

Users can program the duty-cycle register (WIDTH bits) 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. However, by setting the STOP bit, most applications will consume the least amount of power by entering doze mode whenever possible.

While in DOZE mode, the port I/O pins remain in the state prior to entering DOZE mode. CLKO may still be active while the processor core is in DOZE mode. For lowest power operation, CLKO should be disabled. CLKO is controlled by the CLKEN bit of the PLL control register. The data bus is pulled up by internal pullup resistors while the MC68EC000 core is in DOZE mode and the LCDC is not using the data bus to load screen refresh data.

15.3.3.3 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. SLEEP mode is entered by setting the DISPLL bit in the PLL Control Register. See **Section 3.2.5** for more information.

While in SLEEP mode, the port I/O pins remain in the state prior to entering SLEEP mode. CLKO will not be active while the processor core is in SLEEP. The data bus is pulled up by internal pullup resistors while the PLL is disabled. To maintain reliable edges on CLKO, it is a good practice to disable CLKO before entering SLEEP mode and re-enable CLK0 after wake-up.

SECTION 16
APPLICATIONS AND DESIGN EXAMPLES

This section discusses the details of a simple M68328 processor base system, which will illustrate the simplicity of the system hardware design and glueless interface to various memory and peripheral devices when using the M68328 processor. The ability of a mixed 8-bit and 16-bit memory is also demonstrated. This system consists of:

- M68328 16.58MHz (using 32.768kHz crystal)
- 512K boot EPROM x 1 (8-bit interface)
- 2M general-purpose EPROM (16-bit interface)
- 512K SRAM (8-bit interface)
- 2M general-purpose SRAM (16-bit interface)

16.1 MEMORY MAP

The M68328 processor can support up to 4G address space. This example uses an address space under 16M for simplicity.

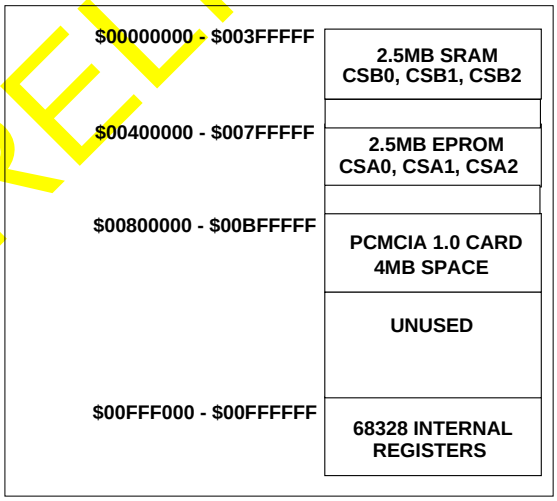


Figure 16-1. Memory Map

16.1.1 M68328 Processor Signal Configuration

A low boot bus width-select (BBUSW) at reset means that the system will boot up with an 8-bit ROM. A low mode-clock (MOCLK) signal at reset means that the M68328 processor uses the 32.768kHz crystal and enables its internal PLL to supply the clock for the entire chip. The suggested circuit for PLL is shown in Figure 14-2. The RESET signal should be connected to the JTAGRST signal to ensure that the entire chip (including the JTAG block) is reset properly at the same time. The RESET and JTAGRST should be held low for more than 100ms before they are released as required by the M68EC000 user's manual. There is no power-on-reset circuit implemented in the M68328 processor; therefore, use external circuits (either from a MC1455 mono-stable timer or power-on-reset signal from most of the power supply chips) to generate the RESET signal. The RESET signal in this example connects to a debouncing circuit to generate a reset for the M68328 processor.

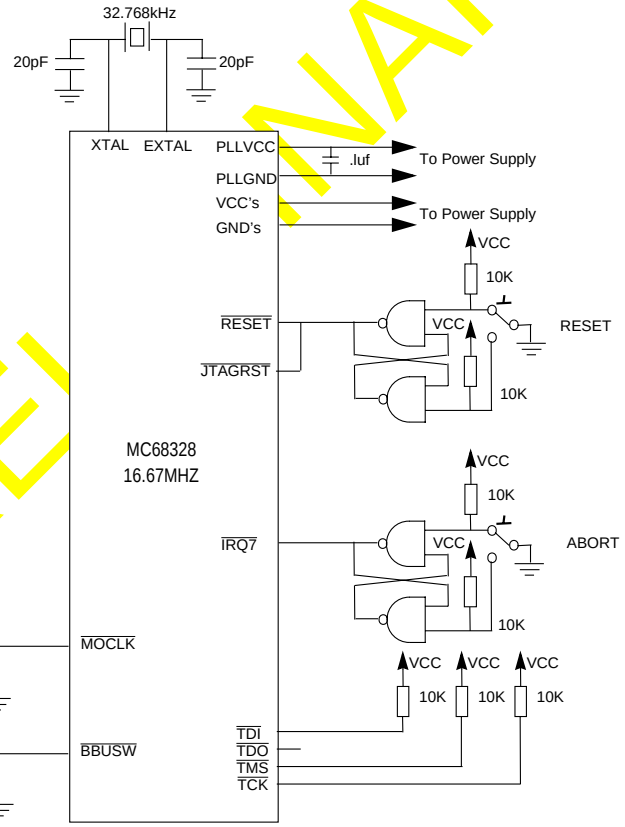


Figure 16-2. M68328 Processor Signal Configuration

The PLLVCC and PLLGND pins should have a .luf bypass capacitor as close to them as possible to reduce the noise induced into the PLL clock synthesis circuit. In general, PLLVCC and PLLGND can be connected to the main power supply with adequate filtering. Bypass capacitors for normal VCC and GND pins are also recommended. Because JTAG is not used in general operations, those related signals (TDI, TMS, and TCK) should be pulled high through resistors to avoid unknown states. The $\overline{IRQ7}$ connects to a debouncing circuit in this example to generate a Level 7 interrupt of the M68328 processor when the ABORT button is activated.

Note: PLLVCC must always operate at 3.3V, even when using the 5V MC68328 version.

16.1.2 Memory Interface

16.1.2.1 EPROM INTERFACE

The M68328 processor supports both 8-bit and 16-bit devices. Any chip-select can be individually programmed for 8 or 16 bits; however, CSA0 can be programmed only by the externally supplied BBUSW signal. Figure 14-3 shows the connection of M68328 processor to EPROM. The $\overline{OE}$ signal controls the EPROM output-enable. When the M68328 processor performs a read cycle (8-bit or 16-bit), the $\overline{OE}$ signal will always be asserted. $\overline{CSA0}$ is connected to the $\overline{CE}$ signal of the boot EPROM and the $\overline{CSA1}$ and $\overline{CSA2}$ are connected to the $\overline{CE}$ signals of the other EPROMs. To achieve better (faster) timing, users can tie the EPROM $\overline{CE}$ signal to GND and hook the $\overline{CSAx}$ signal to the EPROM $\overline{OE}$ signal. However, **this is not recommended in a power-conscious system because the EPROM will be on all the time**.

16.1.2.2 SRAM INTERFACE

The M68328 processor provides internally decoded write-enable signals, $\overline{LWE}$ and $\overline{UWE}$, for a glueless interface to SRAM. They are connected to the write-enable ($\overline{W}$) signals of the 16-bit (8-bit x 2) SRAM. The $\overline{OE}$ signal is also used to gate the output signal from the SRAM during a write cycle that is connected to the SRAM $\overline{G}$ signal. CSB0 is connected to the E signal. The 8-bit SRAM can also be mixed with 16-bit SRAM. For 8-bit-oriented SRAM, only the $\overline{UWE}$ is used for the $\overline{W}$ signal.

Note: The $\overline{WE}$ signal of the M68328 processor supports PCMCIA 1.0 (when CSD3 is in use). It cannot be used for general SRAM write-enable signals.

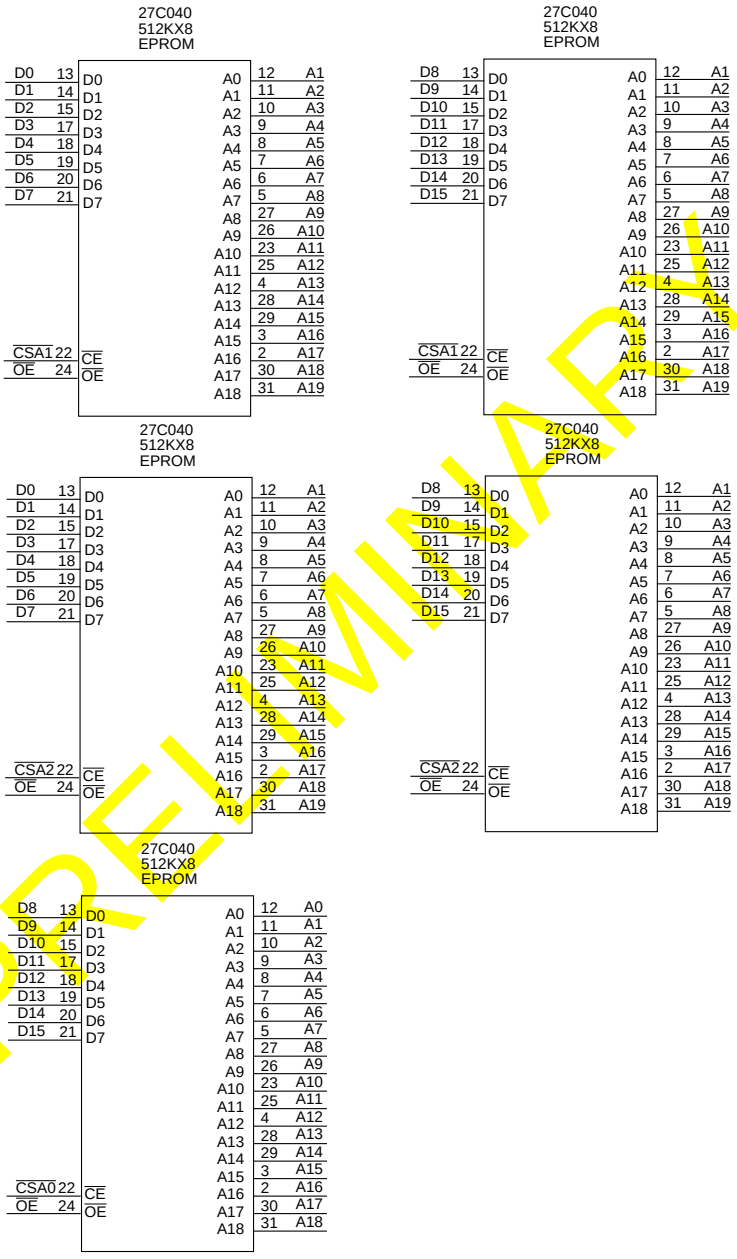


Figure 16-3. EPROM Interface

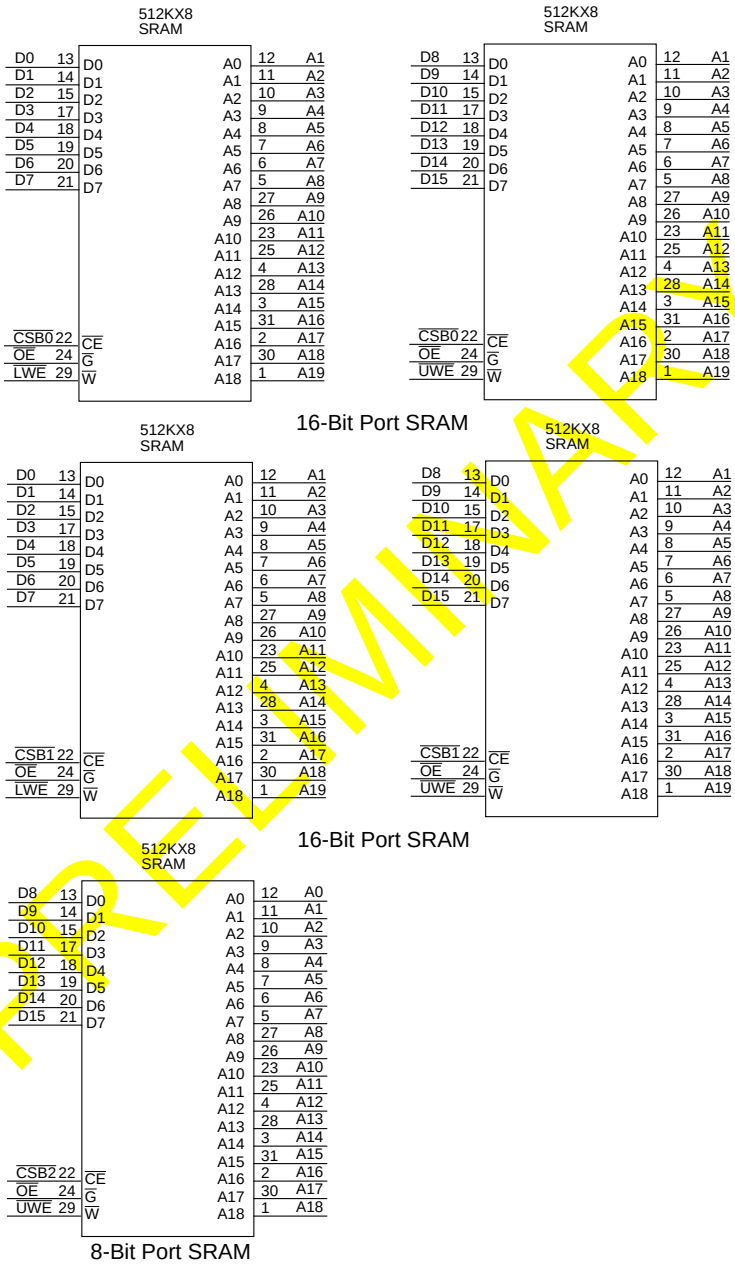


Figure 16-4. SRAM Interface

16.2 SYSTEM INITIALIZATION EXAMPLES

16.2.1 Software Listing 1 - Initialization Code

```
M328BASE  equ $FFF000

; SIM28 System Configuration Registers
SCR       equ    (M328BASE+$000)

; Chip Select Registers
                                ; CS Group Base Registers
GRPBASEA  equ    (M328BASE+$100)
GRPBASEB  equ    (M328BASE+$102)
GRPBASEC  equ    (M328BASE+$104)
GRPBASED  equ    (M328BASE+$106)
                                ; CS Group Mask Registers
GRPMASKA  equ    (M328BASE+$108)
GRPMASKB  equ    (M328BASE+$10A)
GRPMASKC  equ    (M328BASE+$10C)
GRPMASKD  equ    (M328BASE+$10E)
                                ; Group A CS Registers
CSA0      equ    (M328BASE+$110)
CSA1      equ    (M328BASE+$114)
CSA2      equ    (M328BASE+$118)
CSA3      equ    (M328BASE+$11C)
                                ; Group B CS Registers
CSB0      equ    (M328BASE+$120)
CSB1      equ    (M328BASE+$124)
CSB2      equ    (M328BASE+$128)
CSB3      equ    (M328BASE+$12C)
                                ; Group C CS Registers
CSC0      equ    (M328BASE+$130)
CSC1      equ    (M328BASE+$134)
CSC2      equ    (M328BASE+$138)
CSC3      equ    (M328BASE+$13C)
                                ; Group D CS Registers
CSD0      equ    (M328BASE+$140)
CSD1      equ    (M328BASE+$144)
CSD2      equ    (M328BASE+$148)
CSD3      equ    (M328BASE+$14C)

; PLL Registers
PLLCR     equ    (M328BASE+$200); Control Reg
PLLFSR    equ    (M328BASE+$202); Freq Select Reg
PLLTSR    equ    (M328BASE+$204); Test Reg

; Power Control Registers
PCTLR     equ    (M328BASE+$206); Control Reg

; Interrupt Registers
IVR       equ    (M328BASE+$300); Interrupt Vector Reg
ICR       equ    (M328BASE+$302); Interrupt Control Reg
IMR       equ    (M328BASE+$304); Interrupt Mask Reg
IWR       equ    (M328BASE+$308); Wakeup Enable Reg
ISR       equ    (M328BASE+$30C); Interrupt Status Reg
IPR       equ    (M328BASE+$310); Interrupt Pending Reg

; PIO Registers
                                ; Port A Registers
```

```
PADIR     equ    (M328BASE+$400); Direction Reg
PADATA    equ    (M328BASE+$401); Data Reg
PASEL     equ    (M328BASE+$403); Select Reg
                                  ; Port B Registers
PBDIR     equ    (M328BASE+$408); Direction Reg
PBDATA    equ    (M328BASE+$409); Data Reg
PBSEL     equ    (M328BASE+$40B); Select Reg
                                  ; Port C Registers
PCDIR     equ    (M328BASE+$410); Direction Reg
PCDATA    equ    (M328BASE+$411); Data Reg
PCSEL     equ    (M328BASE+$413); Select Reg
                                  ; Port D Registers
PDDIR     equ    (M328BASE+$418); Direction Reg
PDDATA    equ    (M328BASE+$419); Data Reg
PDPUEN    equ    (M328BASE+$41A); Pullup Enable Reg
PDPOL     equ    (M328BASE+$41C); Polarity Reg
PDIRQEN   equ    (M328BASE+$41D); IRQ Enable Reg
PDIRQEDGE equ    (M328BASE+$41F); IRQ Edge Reg
                                  ; Port E Registers
PEDIR     equ    (M328BASE+$420); Direction Reg
PEDATA    equ    (M328BASE+$421); Data Reg
PESEL     equ    (M328BASE+$423); Select Reg
                                  ; Port F Registers
PFDIR     equ    (M328BASE+$428); Direction Reg
PFDATA    equ    (M328BASE+$429); Data Reg
PFSEL     equ    (M328BASE+$42B); Select Reg
                                  ; Port G Registers
PGDIR     equ    (M328BASE+$430); Direction Reg
PGDATA    equ    (M328BASE+$431); Data Reg
PGSEL     equ    (M328BASE+$433); Select Reg
                                  ; Port J Registers
PJDIR     equ    (M328BASE+$438); Direction Reg
PJDATA    equ    (M328BASE+$439); Data Reg
PJSEL     equ    (M328BASE+$43B); Select Reg
                                  ; Port K Registers
PKDIR     equ    (M328BASE+$440); Direction Reg
PKDATA    equ    (M328BASE+$441); Data Reg
PKSEL     equ    (M328BASE+$443); Select Reg
                                  ; Port M Registers
PMDIR     equ    (M328BASE+$448); Direction Reg
PMDATA    equ    (M328BASE+$449); Data Reg
PMPUEN    equ    (M328BASE+$44A); Pullup Enable Reg
PMSEL     equ    (M328BASE+$44B); Select Reg

; PWM Registers
PWMC      equ    (M328BASE+$500); Control Reg
PWMP      equ    (M328BASE+$502); Period Reg
PWMW      equ    (M328BASE+$504); Width Reg
PWMCNT    equ    (M328BASE+$506); Counter

; Timer Registers
                                  ; Timer 1 Registers
TCTL1     equ    (M328BASE+$600); Control Reg
TPRER1    equ    (M328BASE+$602); Prescalar Reg
TCMP1     equ    (M328BASE+$604); Compare Reg
TCR1      equ    (M328BASE+$606); Capture Reg
TCN1      equ    (M328BASE+$608); Counter
TSTAT1    equ    (M328BASE+$60A); Status Reg
                                  ; Timer 2 Registers
TCTL2     equ    (M328BASE+$60C); Control Reg
TPRER2    equ    (M328BASE+$60E); Prescalar Reg
```

```
TCMP2     equ    (M328BASE+$610); Compare Reg
TCR2      equ    (M328BASE+$612); Capture Reg
TCN2      equ    (M328BASE+$614); Counter
TSTAT2    equ    (M328BASE+$616); Status Reg

; Watchdog Registers
WCR       equ    (M328BASE+$618); Control Reg
WRR       equ    (M328BASE+$61A); Reference Reg
WCN       equ    (M328BASE+$61C); Counter

; SPI Registers
                                ; SPI Slave Registers
SPISR     equ    (M328BASE+$700); SPIS Reg
                                ;SPI Master Registers
SPIMDATA  equ    (M328BASE+$800); Control/Status Reg
SPIMCONT  equ    (M328BASE+$802); Data Reg

; UART Registers
USTCNT    equ    (M328BASE+$900); Status Control Reg
UBAUD     equ    (M328BASE+$902); Baud Control Reg
UARTRX    equ    (M328BASE+$904); Rx Reg
UARTTX    equ    (M328BASE+$906); Tx Reg
UARTMISC  equ    (M328BASE+$908); Misc Reg

; LCDC Registers
LSSA      equ    (M328BASE+$A00); Screen Start Addr Reg
LVPW      equ    (M328BASE+$A05); Virtual Page Width Reg
LXMAX     equ    (M328BASE+$A08); Screen Width Reg
LYMAX     equ    (M328BASE+$A0A); Screen Height Reg
LCXP      equ    (M328BASE+$A18); Cursor X Position
LCYP      equ    (M328BASE+$A1A); Cursor Y Position
LCWCH     equ    (M328BASE+$A1C); Cursor Width & Height Reg
LBLKC     equ    (M328BASE+$A1F); Blink Control Reg
LPICF     equ    (M328BASE+$A20); Panel Interface Config Reg
LPOLCF    equ    (M328BASE+$A21); Polarity Config Reg
LACDRC    equ    (M328BASE+$A23); ACD (M) Rate Control Reg
LPXCD     equ    (M328BASE+$A25); Pixel Clock Divider Reg
LCKCON    equ    (M328BASE+$A27); Clocking Control Reg
LLBAR     equ    (M328BASE+$A29); Last Buffer Addr Reg
LOTCR     equ    (M328BASE+$A2B); Octet Terminal Count Reg
LPOSR     equ    (M328BASE+$A2D); Panning Offset Reg
LFRCM     equ    (M328BASE+$A31); Frame Rate Control Mod Reg
LGPMR     equ    (M328BASE+$A32); Gray Palette Mapping Reg
LIRQR     equ    (M328BASE+$A34); Interrupt Control Reg

; RTC Registers
RTCHMSR   equ    (M328BASE+$B00); Hrs Mins Secs Reg
RTCALM0R  equ    (M328BASE+$B04); Alarm Register 0
RTCALM1R  equ    (M328BASE+$B08); Alarm Register 1
RTCCTL    equ    (M328BASE+$B0C); Control Reg
RTCISR    equ    (M328BASE+$B0E); Interrupt Status Reg
RTCIENR   equ    (M328BASE+$B10); Interrupt Enable Reg
RSTPWCH   equ    (M328BASE+$B12) ; Stopwatch Minutes

Reset_start:

    move.w  #$2700,sr       ; mask off all interrupts
;*****************************************************************
;
; Initialize I/O according to user's configuration
;
```

```
;*********************************************************************


;*********************************************************************
;
;   LCD Temp init for screen protection
;
;*********************************************************************


    move.l  #$00410000,LSSA      ;frame starting address
    move.b  #$14,LVPW            ;virtual width
    move.w  #319,LXMAX           ;xmax = 16
    move.w  #199,LYMAX           ;ymax = 16
    move.b  #$14,LLBAR           ;LBAR
    move.b  #$1b,LOTCR           ;RPTC
    move.b  #2,LPXCD             ;pixel clock = 1/3 sysclk
    move.b  #$04,LPICF           ;4 bit LCD data no grey
    move.b  #$e0,LCKCON          ;enable lcdc, 16 words per DMA


    move.b  #$1c,SCR       ; enable bus error bit
    clr.l       d0
    clr.l       d1
    clr.l       d2
    clr.l       d3
    clr.l       d4
    clr.l       d5
    clr.l       d6
    clr.l       d7
    move.l  #$40010F09,CSA0
    move.l  #$50010F09,CSA1
    move.l  #$60010F09,CSA2
    move.l  #$70010F09,CSA3
    move.w  #$000F,GRPMASKA      ; GROUPA MASK init
    move.w  #$0000,GRPBASEA      ; GROUPA BASE init
                                 ; Start add.=0x0, Size=8MB


testq:
    move.w  #$0000,GRPBASEB      ; SRAM0 = 0x00000000-0x003FFFFF
    move.l  #$00010F00,CSB0      ; CSB0 = 0x0-0x000FFFFF
    move.l  #$10010F00,CSB1      ; CSB1 = 0x00100000-0x001FFFFF
    move.l  #$20010F00,CSB2      ; CSB2 = 0x00200000-0x002FFFFF
    move.l  #$30010F00,CSB3      ; CSB3 = 0x00300000-0x003FFFFF
    move.w  #$0030,GRPMASKB      ; Group Base=0x00000000, Group Mask=4MB
    move.w  #$0001,GRPBASEB      ; SRAM0 = 0x00000000-0x003FFFFF

;    Enable Valid Bit after setting up registers

    move.l  #$f0010007,CSD3      ; CSD3 = PCMCIA 4MB 6 w/s
    move.w  #$0070,GRPMASKD      ; Group Base=0x00800000, Group Mask=8MB
    move.w  #$0081,GRPBASED      ; PERI = 0x00800000-0x00FFFFFF
    move.w    #$0001,GRPBASEA        ; Enable at the end
;*********************************************************************
;
;   LCD init
;
;*********************************************************************
```

```
        jmp _main                           ; jmp to main program
```

16.2.2 Chip-Select Access Time Calculation

The time path that is critical in the M68328 processor application using the CS signals is shown in Figure 14-5. The chip-select access time is from CS asserted to when data must be valid to the processor.

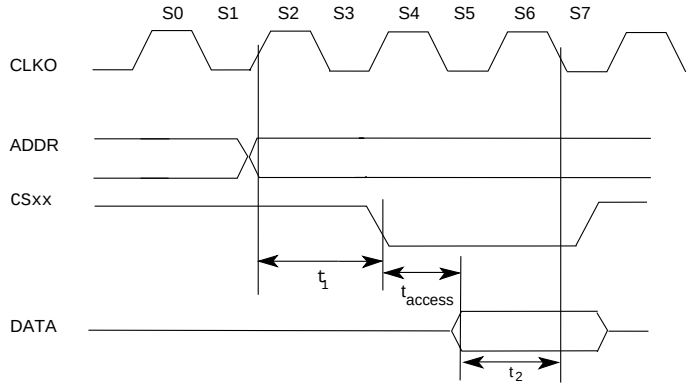


Figure 16-5. Chip-Select Access Timing

An equation for the access time, t_{access}, can be developed from Figure 14-5. This equation applies to EC000 core accesses.

$t_{access} = (2 + WS) / T - (t_1 + t_2)$

where t = system clock cycle time

where WS = number of wait states to be programmed in the chip-select register

where t_1 = CSxx delay = 30ns

where t_2 = DATA setup requirement = 20 ns

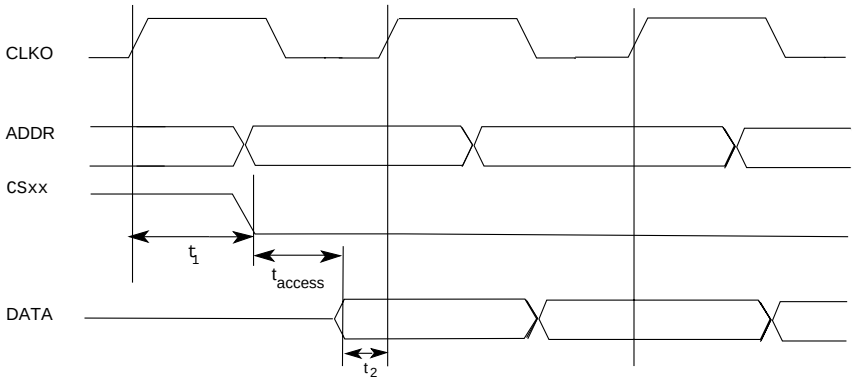


Figure 16-6. LCD Chip-Select Access Timing

An equation for the access time, t_{acess}, can be developed from Figure 14-6. This equation applies to EC000 core accesses.

$t_{acess} = (1 + WS)\ T - (t_1 + t_2)$

where WS = wait states

where T = system clock period

where t_1 = CSxx delay = 30ns

where t_2 = DATA setup requirement = 20ns

16.2.3 LCDC Temporary Initialization

At reset, the LCDC output is idle-low and an LCD pixel clock is not provided to the LCD panel. Because a DC current may be induced in the LCD panel glass due to unknown values in the driver shift registers, this DC current causes permanent demage to the LCD panel glass. To avoid this situation, users may either turn off the LCD VEE (high -ve or +ve voltage) as power-on default or initalize the LCD controller immediately after the processor has started. The temporary initialization routine protects the LCD panel screen.

16.2.4 Programming the Interrupt Controller

The interrupt controller level is summarized as shown in Figure 14-5. Basically, all 7 levels of the 68EC000 are used. Because some of the interrupts are lumped into one interrupt level such as Level 4, a user's interrupt service routine should resolve the priority by reading the

interrupt status register (ISR) and the interrupt pending register (IPR). Among interrupt levels, their priority is shown in Figure 14-5, with Level 7 being the highest.

Note: Keyboard down is the OR (negative logic) of all inputs on the keyboard port.

16.2.4.1 METHOD FOR CLEARING INTERRUPTS

Some of the interrupt sources have different clearing conventions. Figure 14-6 summarizes interrupt-status clearing.

16.2.4.2 INTERRUPT LEVEL SUMMARY

```
Level 7
        IRQ7
Level 6
        SPIS
        Timer 1
        IRQ6
Level 5
        PENIRQ
Level 4
        SPIM
        Timer 2
        UART
        Parallel I/O (8 interrupts)
        WatchDog (while not in "Force Reset" mode
        Real Time Clock
        Keyboard Down
        Pulse Width Modulator
Level 3
        IRQ3
Level 2
        IRQ2
Level 1
        IRQ1
```


16.2.4.3 INTERRUPT CLEARING SUMMARY

	ADDRESS	BIT POSITION	CLEAR BY:
UART block			
GPIO Delta	$(ff)fff902	15	clear by writing zero
CTS Delta	$(ff)fff906	8	clear by writing zero
RX FIFO Full	$(ff)fff904	15	auto clears when FIFO not full
RX FIFO Half	$(ff)fff904	14	auto clears when FIFO less than half full
RX Data Ready	$(ff)fff904	13	auto clears when FIFO is empty
TX FIFO Empty	$(ff)fff906	15	auto clears when FIFO not empty
TX FIFO Half	$(ff)fff906	14	auto clears when FIFO greater than half full
TX Available	$(ff)fff906	13	auto clears when FIFO is full

```
PWM block
        PWM IRQ                 $(ff)fff500             15          auto clear after reading

Timer block
        timer1 capture          $(ff)fff60a              0          clear by writing zero
        timer2 capture          $(ff)fff616              1          clear by writing zero
        timer2 compare          $(ff)fff616              0          clear by writing zero

SPI Master block
        SPIM IRQ                $(ff)fff802              7          clear by writing zero

SPI Slave block
        SPIS IRQ                $(ff)fff700             15          auto clear after reading

Real Time Clock block
        Period IRQ              $(ff)fffb0e              6          clear by writing zero    (eliminate)
        1 Hz Flag               $(ff)fffb0e              4          clear by writing zero
        Day Flag                $(ff)fffb0e              3          clear by writing zero
        Alarm Flag              $(ff)fffb0e              2          clear by writing zero    (eliminate)

Port D block
        (edge mode) bit IRQ     $(ff)fff418            7-0          clear external IRQ source

System Interrupts
        IRQ7                    $(ff)fff30c             23          clear by writing one
        (level mode) IRQ1       $(ff)fff30c             16          clear external IRQ source
        (edge mode) IRQ1        $(ff)fff30c             16          clear by writing one
        (level mode) IRQ2       $(ff)fff30c             17          clear external IRQ source
        (edge mode) IRQ2        $(ff)fff30c             17          clear by writing one
        (level mode) IRQ3       $(ff)fff30c             18          clear external IRQ source
        (edge mode) IRQ3        $(ff)fff30c             18          clear by writing one
        (level mode) IRQ6       $(ff)fff30c             19          clear external IRQ source
        (edge mode) IRQ6        $(ff)fff30c             19          clear by writing one
```


16.2.5 SPIM Initialization Examples

The following is an example system using the SPI master (SPIM) to capture data from an external device such as an A/D converter. The device can be directly connected to the M68328 processor. Port PK7 serves as the enable signal to the A/D converter. This example uses a polling mechanism to capture the data from the A/D converter and illustrates the SPIM operation. The hardware connection is shown in Figure 16-8.

The SPIM data should be reloaded each time before initiating the SPIM control register exchange bit. Although the example does not use interrupt service, the polling routine

should poll the IRQ bit for an indication of a completed transfer. Do not poll the exchange bit for exchange completion.

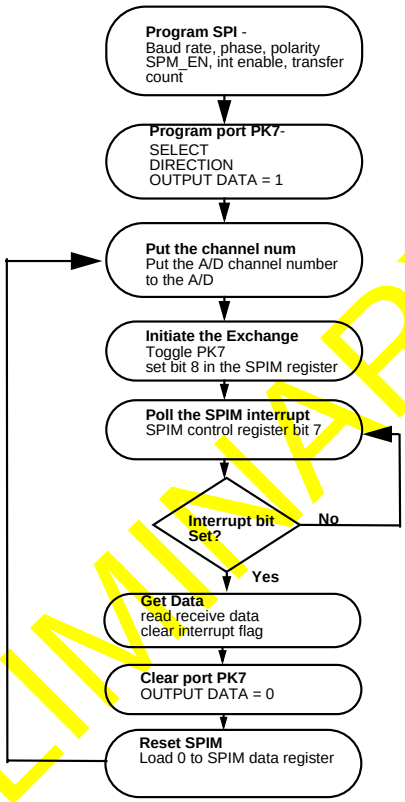


Figure 16-7. SPIM Data-Capture Flowchart

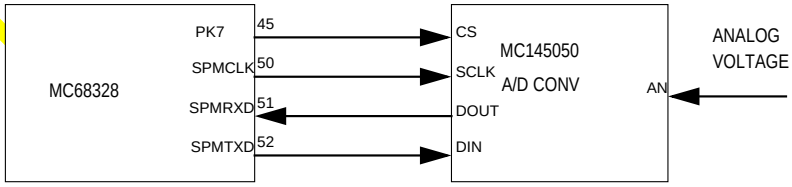


Figure 16-8. Hardware Configuration

16.3 POWER-SAVING TIPS

You can save power on the MC68328 by following these tips:

- Keep the core shut down as much as possible by putting it in doze mode. Set the STOP bit in the power control register during any peiod of inactivity.
- Shut down the PLL to enter sleep mode. Set the DISPLL bit in the PLL control register to conserve power when the processor is inactive for an extended period of time.
- Avoid the following pull-up resistor conditions to save the most power:
  - ❏ When an internal pull-up is enabled on an input with an external device driving it low.
  - ❏ When an internal pull-up is enabled on an I/O with the internal output device driving it low.
- Drive the outputs to their high state before entering sleep mode. The outputs have an internal pull-up that will not consume excess current when the pin is driven high. However, some external devices may require that the signal from the MC68328 port be low to save the most power.
- Disable the pull-ups on inputs that are always driven externally. Inputs that are continuously driven from an external device do not require an internal pull-up resistor. You can save power by disabling the internal pull-up resistor on the MC68328 I/O ports that are configured as inputs and always driven to a high or low state by the external device.
- Disable the I/O port internal pull-ups one by one to isolate excess current drain in sleep mode.
- Beware of outputs sourcing current for externally powered-down devices. When the MC68328 is powered down in sleep or doze mode, the output pins remain in the state they were in before going to sleep. This may cause an external device with its power turned off to try to power itself from the MC68328 output pins that are connected to it. In this case, it may be best to make sure the MC68328 output pin is in the low state and that internal pull-up is disabled.
- Open inputs will cause excess current drain. JTAG inputs are often forgotten.
- Inputs that are not being driven rail to rail consume excess current.

16.3.1 LCD Refresh Frequency

The following sequence of steps can be used to determine the values required for the PIXCLK divider (LCDC PXCD), last buffer address register (LBAR), and octet terminal count register (OTCR). These registers all affect the refresh rate of the LCD panel. The following example assumes a screes size of 240 × 160.

1. Calculate the pixel clock divider for grayscale.

   DIV = (PIX_CLK_SOURCE) / (XMAX × 2 × YMAX × REFRESH_RATE) (for grayscale)

   DIV = (PIX_CLK_SOURCE) / (XMAX × YMAX × REFRESH_RATE)

   DIV = (16.58MHz) / (240 × 2 × 160 × 70) = 3.08 ≥3

   Divider must be a whole number, so round to 3. Byte write $02 to PXCD register to set the divider equal to 3. Notice that PIX_CLK_SOURCE is derived from the PLL control register.

2. Last buffer address register (LBAR).

   LBAR = 240 / 16 = 15

   LBAR = 240 / 8 = 30 (for grayscale)

3. Calculate the value of the OTCR register. This value controls the time required between two line pulses. Select a value so that OTCR > LBAR, as required.

   OTCR = LBAR + 4 = 19

   OTCR = LBAR + 8 = 38 (for grayscale)

   Increasing the value in OTCR will increase the LP period.

4. Calculate the frequency.

   Frequency = 16.58MHz / (DIV × YMAX × OCTET × 16) = 16.58MHz / (3 × 160 × 38 × 16) (for grayscale) = 56MHz

   If OCTET = 31, then frequency = 69Hz. OCTET is the OTCR register value.

SECTION 17
ELECTRICAL CHARACTERISTICS

<add info about data strobes>This section provides information on the maximum ratings for the MC68328 processor.

17.1 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

17.2 POWER CONSUMPTION

CHARACTERISTIC	3.3 V		UNIT
	TYPICAL	MAX	
Operating Current @16 MHz	15	30	mA
Standby Current	10	20	uA

17.3 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. The MC68328 supports internal and external transfer acknowledge as well as 16-bit or 8-bit data transfer mode. The diagrams in following sections illustrate the bus-timing for these various modes of operation.

17.4 AC ELECTRICAL SPECIFICATIONS—READ AND WRITE CYCLES

(Frequency = 0 to 16 MHz; GND = 0 V; $T_A = T_L$ to T_H; see Figures 13-1 through 13-5)

Table 17-1. AC Electrical Specifications—Write Cycle Timing

NUM	CHARACTERISTIC	3.3 V		UNIT
		MIN	MAX	
1	CLKO High to Address Valid	0	30	ns
2	Addr Valid to $\overline{CS}$xx Asserted	40	—	ns
3	CLKO High to $\overline{AS}$ Asserted	0	30	ns
4	$\overline{AS}$ Asserted to $\overline{CS}$xx Asserted	0	30	ns
5	$\overline{AS}$ Width Asserted	110	—	ns
6	$\overline{CS}$xx Width Asserted	110	—	ns
7	CLKO High to $\overline{UDS}$, $\overline{LDS}$ Asserted	0	35	ns
8	$\overline{DTACK}$ Asynchronous Input Setup Time	25	—	ns
9	CLKO High to Address, R/$\overline{W}$ Invalid	0	30	ns
10	$\overline{AS}$ Width Negated	0	50	ns
11	CLKO High to R/$\overline{W}$ Valid	0	30	ns
12	$\overline{AS}$ High to R/$\overline{W}$ Invalid	0	—	ns
13	$\overline{AS}$ High to $\overline{CS}$xx Negated	0	20	ns
14	$\overline{CS}$xx Width Negated	0	40	ns
15	CLKO High to OE Invalid	0	30	ns
16	CLKO Low to $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$ Negated	0	30	ns
18	$\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$ Negated to $\overline{DTACK}$ Negated	0	65	ns
19	CLKO High to Data Valid	T/2+0	T/2+30	ns
20	$\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$ Negated to Data High Impedance	0	30	ns
21	$\overline{AS}$ Asserted to $\overline{UWE}$, $\overline{LWE}$ Asserted	0	30	ns
22	$\overline{UWE}$, $\overline{LWE}$ Width Asserted	50	—	ns
23	$\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$ Negated to Data Invalid	5	—	ns
24	$\overline{UWE}$, $\overline{LWE}$ Negated to Data Invalid	15	—	ns
25	$\overline{UWE}$, $\overline{LWE}$ Negated to $\overline{CS}$xx Negated	T/2-10	T/2+5	ns

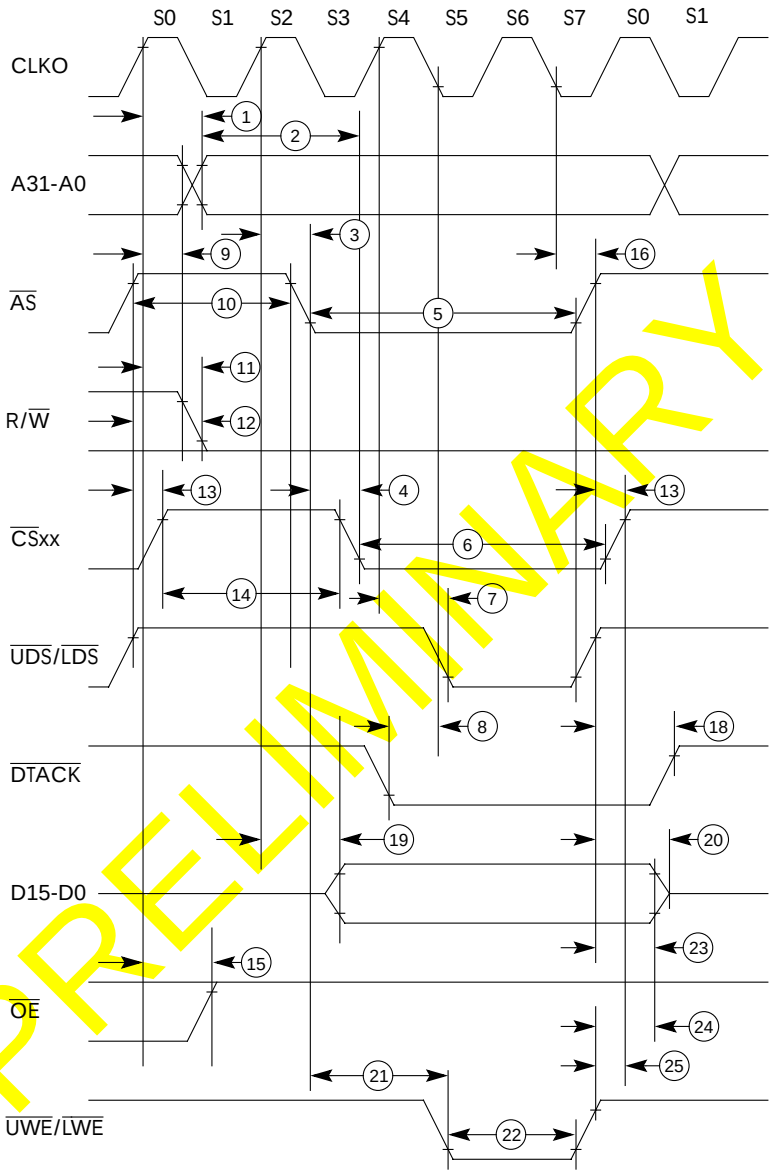


Figure 17-1. Chip-Select Write Cycle Timing (when CPU is Bus Master)

Table 17-2. AC Electrical Specifications—Chip-Select Read Cycle Timing

NUM	CHARACTERISTIC	3.3 V		UNIT
		MIN	MAX	
1	CLKO High to Address Valid	0	30	ns
2	Addr Valid to $\overline{CS}$xx Asserted	40	—	ns
3	CLKO High to $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$ Asserted	0	30	ns
4	$\overline{AS}$ Asserted to $\overline{CS}$xx Asserted	0	30	ns
5	$\overline{AS}$ Width Asserted	110	—	ns
6	$\overline{CS}$xx Width Asserted	110	—	ns
8	$\overline{DTACK}$ Asynchronous Input Setup Time	25	—	ns
9	CLKO High to Address, R/$\overline{W}$ Invalid	0	30	ns
10	$\overline{AS}$ Width Negated	0	50	ns
11	CLKO High to R/$\overline{W}$ Valid	0	30	ns
12	$\overline{AS}$ High to R/$\overline{W}$ Invalid	0	—	ns
13	$\overline{AS}$ High to $\overline{CS}$xx Negated	0	20	ns
14	$\overline{CS}$xx Width Negated	0	40	ns
15	CLKO High to $\overline{OE}$ Invalid	0	30	ns
16	CLKO Low to $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$ Negated	0	40	ns
18	$\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$ Negated to $\overline{DTACK}$ Negated	0	65	ns
19	$\overline{DTACK}$ Asserted Data In Valid	—	35	ns
20	Data Input Setup Time to CLKO Low	20	—	ns
21	$\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$ Negated to Data High Impedance	0	30	ns
22	$\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$ Negated to Data Invalid	0	—	ns

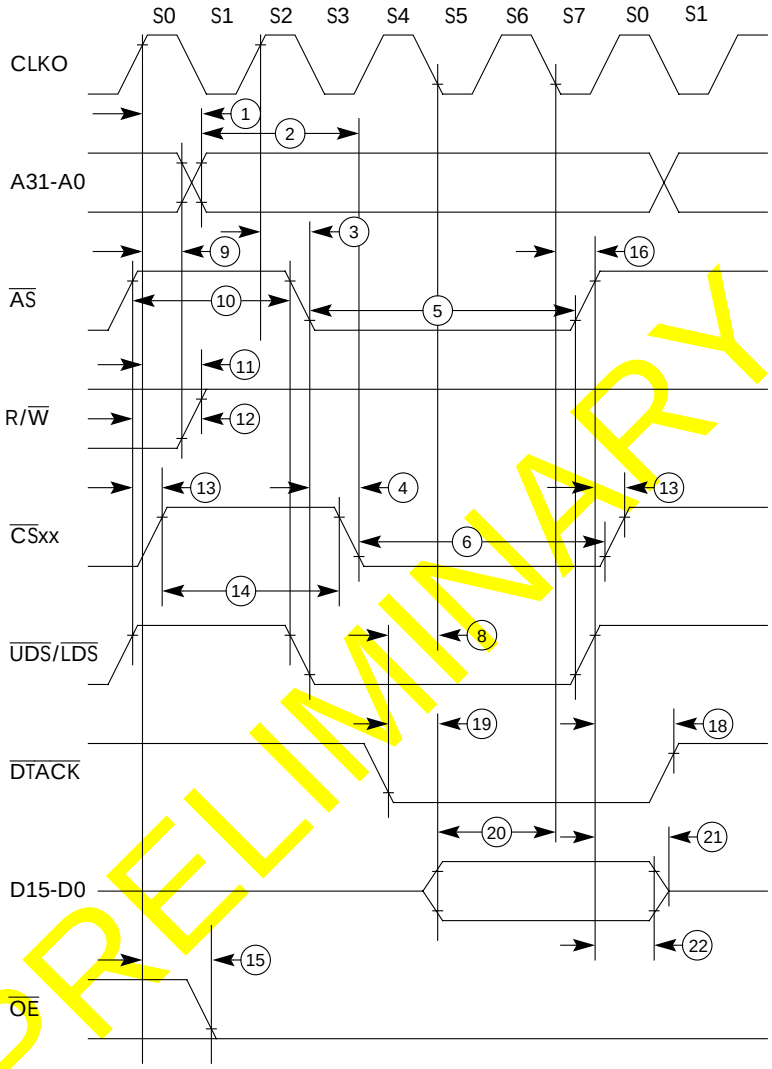


Figure 17-2. Chip-Select Read Cycle Timing (when the CPU is the Bus Master)

Table 17-3. AC Electrical Specifications—LCD-DMA Read Cycle Timing

NUM	CHARACTERISTIC	3.3 V		UNIT
		MIN	MAX	
1	CLKO High to Address Valid	0	30	ns
2	CLKO High to $\overline{CS}$xx Asserted	0	30	ns
3	Data Input Setup Time to CLKO High	20	—	ns
4	$\overline{CS}$xx Width Asserted	40	—	ns
5	$\overline{AS}$ Negated to $\overline{CS}$xx Asserted	T+10	—	ns
6	R/$\overline{W}$ High to $\overline{CS}$xx Asserted	20	—	ns
7	$\overline{AS}$ High to $\overline{CS}$xx Negated	0	20	ns
8	$\overline{CS}$xx Width Negated	0	40	ns
9	$\overline{OE}$ Asserted to $\overline{CS}$xx Asserted	20	—	ns
10	DMAC Address Hold Time	0	—	ns
11	$\overline{CS}$xx Negated to $\overline{AS}$ Asserted	T+10	—	ns
12	CLKO High to $\overline{CS}$xx Negated	0	—	ns
13	$\overline{CS}$xx Negated to $\overline{OE}$, R/$\overline{W}$ Invalid	0	—	ns

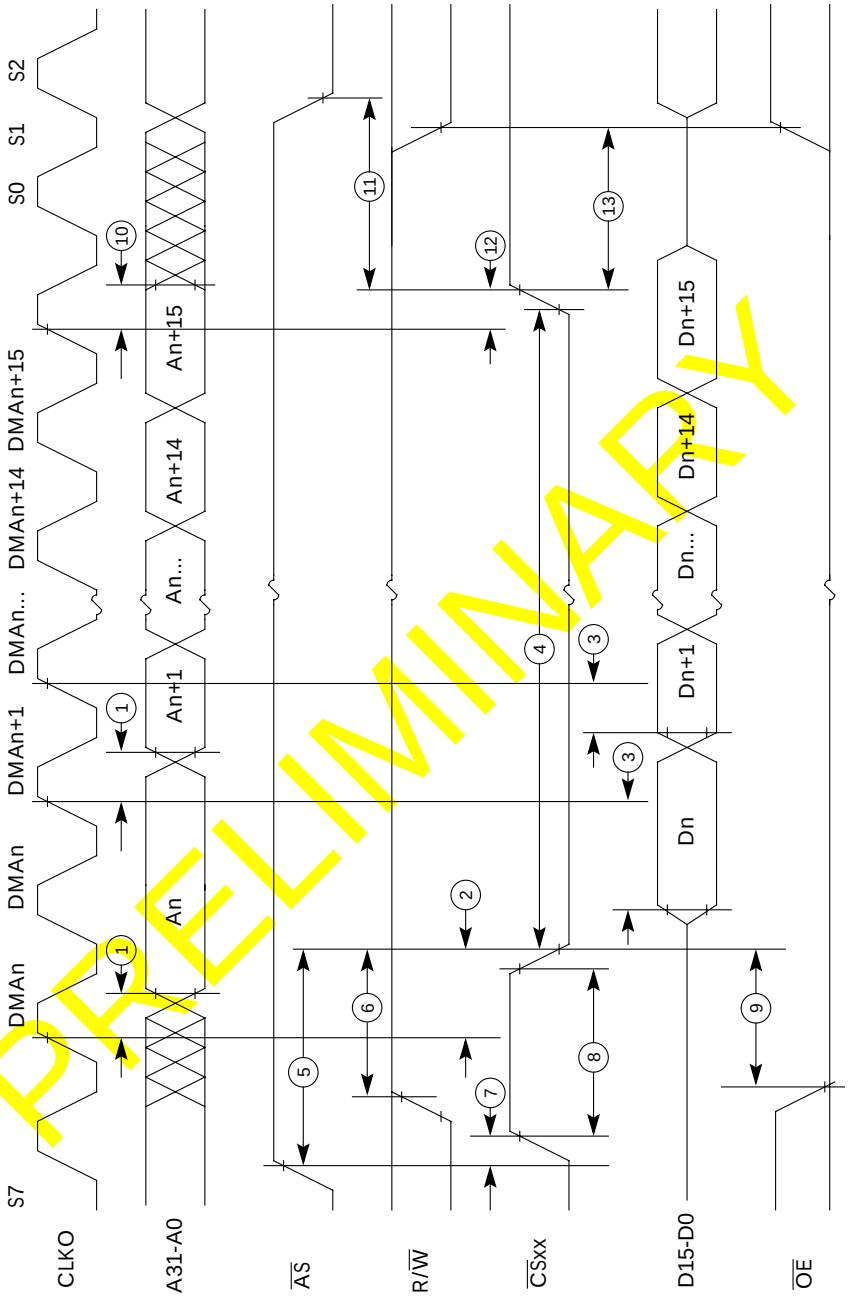


Figure 17-3. LCD-DMA Read Cycle Timing (when the DMAC is the Bus Master)

Table 17-4. AC Electrical Specifications—PCMCIA Write Cycle Timing

NUM	CHARACTERISTIC	3.3 V		UNIT
		MIN	MAX	
1	Write Cycle Time	150	—	ns
2	Addr Setup Time to $\overline{CE1}$, $\overline{CE2}$ Asserted	40	—	ns
3	Addr Valid to $\overline{WE}$ Asserted	0	30	ns
4	$\overline{WE}$ Negated to Address Invalid	90	—	ns
5	Data Setup to $\overline{WE}$ Negated	0	30	ns
6	$\overline{WE}$ Negated to Data Invalid (Hold Time)	30	—	ns

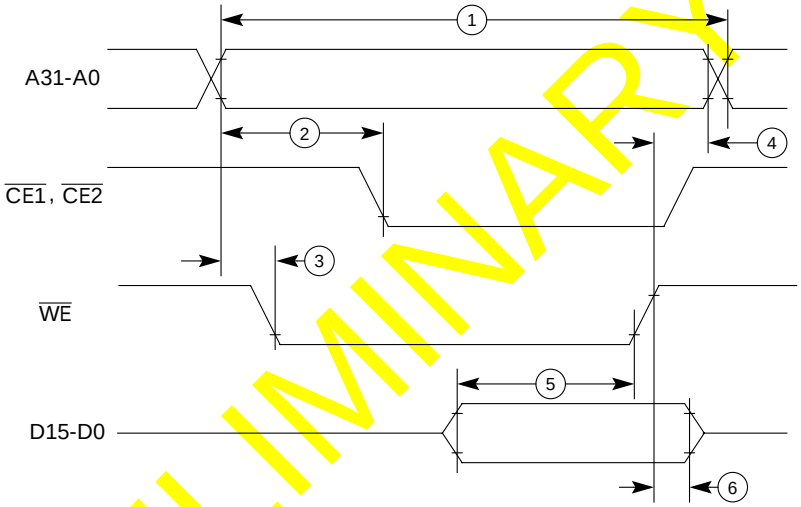


Figure 17-4. PCMCIA Write Cycle Timing

Table 17-5. AC Electrical Specifications—PCMCIA Read Cycle Timing

NUM	CHARACTERISTIC	3.3 V		UNIT
		MIN	MAX	
1	Read Cycle Time	150	—	ns
2	Addr Access Time	—	250	ns
3	Addr Valid to $\overline{OE}$ Asserted	0	30	ns
4	$\overline{CE1}$, $\overline{CE2}$ Access Time	—	250	ns
5	$\overline{OE}$, $\overline{CE1}$, $\overline{CE2}$ Negated to Data High Impedance	0	30	ns
6	Addr Invalid to Data Invalid (Hold Time)	—	0	ns

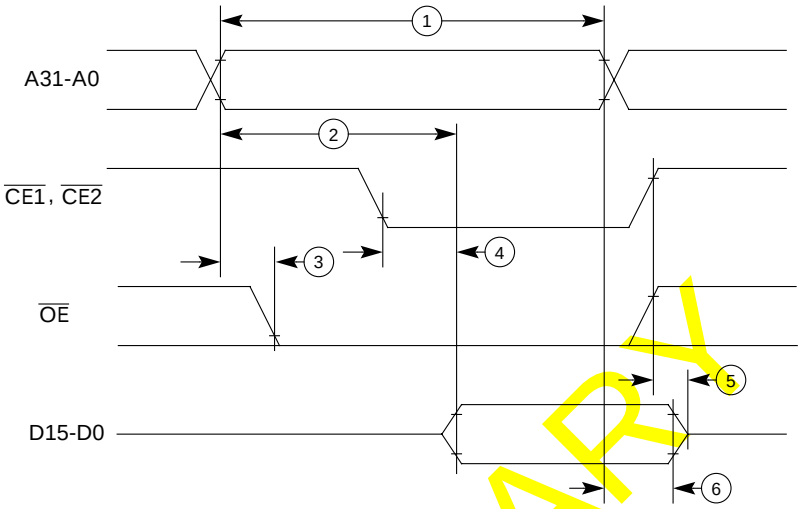


Figure 17-5. PCMCIA Read Cycle Timing

17.5 DC ELECTRICAL SPECIFICATIONS

CHARACTERISTIC	SYMBOL	MIN	MAX	UNIT
Input high voltage (except EXTAL)	V_{IH}	2.0	V_{DD}	V
Input low voltage	V_{IL}	GND – 0.3	0.8	V
Clock input high voltage (EXTAL)	VIHC	0.7 V_{DD}	V_{DD}+ 0.3	V
Input leakage current @3.3V (all input-only pins)[1]	I_{IN}	—	-1.0	uA
Three-state (off state) input current @2.4V/0.4V	I_{TSI}	—	-5.0	uA
Output high voltage (I_{OH} = 1.5mA)	V_{OH}	0.8V_{DD}	—	V
Output low voltage (I_{OL} = 2mA)	V_{OL}	—	0.4	V
Input capacitance[3] All input-only pins All I/O pins	C_{IN}	— —	7 14	pF
Load capacitance [3]	C_L	—	50	pF

NOTES:
1. Not including internal pull-up.
2. Currents listed are with no loading.
3. Capacitance is periodically sampled rather than 100% tested.