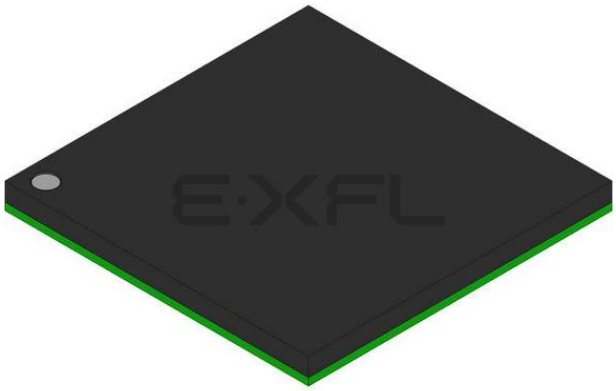

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	Coldfire V2
Core Size	32-Bit Single-Core
Speed	66MHz
Connectivity	EBI/EMI, Ethernet, I²C, SPI, UART/USART, USB
Peripherals	DMA, WDT
Number of I/O	32
Program Memory Size	16KB (4K x 32)
Program Memory Type	ROM
EEPROM Size	-
RAM Size	1K x 32
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	196-LBGA
Supplier Device Package	196-LBGA (15x15)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mcf5272cvm66j

List of Figures (Continued)

Figure Number	Title	Page Number
11-28	Transmit Buffer Descriptor (TxBD)	11-37
12-1	The USB “Tiered Star” Topology	12-2
12-2	USB Module Block Diagram	12-3
12-3	USB Frame Number Register (FNR)	12-9
12-4	USB Frame Number Match Register (FNMR)	12-9
12-5	USB Real-Time Frame Monitor Register (RFMR)	12-10
12-6	USB Real-Time Frame Monitor Match Register (RFMMR)	12-11
12-7	USB Function Address Register (FAR)	12-11
12-8	USB Alternate Settings Register (ASR)	12-12
12-9	USB Device Request Data 1 Register (DRR1)	12-13
12-10	USB Device Request Data 2 Register (DRR2)	12-13
12-11	USB Specification Number Register (SPECR)	12-14
12-12	USB Endpoint 0 Status Register (EP0SR)	12-14
12-13	USB Endpoint 0 IN Configuration Register (IEP0CFG)	12-15
12-14	USB Endpoint 0 OUT Configuration Register	12-16
12-15	USB Endpoint 1–7 Configuration Register	12-16
12-16	USB Endpoint 0 Control Register (EP0CTL)	12-17
12-17	USB Endpoint 1-7 Control Register (EPnCTL)	12-20
12-18	USB Endpoint 0 Interrupt Mask (EP0IMR) and General/Endpoint 0 Interrupt Registers (EP0ISR)	12-22
12-19	USB Endpoints 1–7 Interrupt Status Registers (EPnISR)	12-25
12-20	USB Endpoint 1-7 Interrupt Mask Registers (EPnIMR)	12-26
12-21	USB Endpoint 0-7 Data Registers (EPnDR)	12-27
12-22	USB Endpoint 0-7 Data Present Registers (EPnDPR)	12-28
12-23	Example USB Configuration Descriptor Structure	12-29
12-24	Recommended USB Line Interface	12-36
12-25	USB Protection Circuit	12-37
13-1	PLIC System Diagram	13-2
13-2	GCI/IDL Receive Data Flow	13-3
13-3	GCI/IDL B-Channel Receive Data Register Demultiplexing	13-4
13-4	GCI/IDL Transmit Data Flow	13-4
13-5	GCI/IDL B Data Transmit Register Multiplexing	13-5
13-6	B-Channel Unencoded and HDLC Encoded Data	13-6
13-7	D-Channel HDLC Encoded and Unencoded Data.	13-7
13-8	D-Channel Contention	13-8
13-9	GCI/IDL Loopback Mode	13-9
13-10	Periodic Frame Interrupt	13-10
13-11	PLIC Internal Timing Signal Routing	13-12
13-12	PLIC Clock Generator	13-12
13-13	B1 Receive Data Registers P0B1RR–P3B1RR	13-15
13-14	B2 Receive Data Registers P0B2RR – P3B2RR	13-16
13-15	D Receive Data Registers P0DRR–P3DRR	13-16
13-16	B1 Transmit Data Registers P0B1TR–P3B1TR	13-17
13-17	B2 Transmit Data Registers P0B2TR–P3B2TR	13-17

Table of Contents (Continued)

Paragraph Number	Title	Page Number

Chapter 9
SDRAM Controller

9.1 Overview 9-1
9.2 SDRAM Controller Signals 9-1
9.3 Interface to SDRAM Devices 9-4
9.4 SDRAM Banks, Page Hits, and Page Misses 9-6
9.5 SDRAM Registers 9-6
9.5.1 SDRAM Configuration Register (SDCR) 9-6
9.5.2 SDRAM Timing Register (SDTR) 9-8
9.6 Auto Initialization 9-9
9.7 Power-Down and Self-Refresh 9-9
9.8 Performance 9-10
9.9 Solving Timing Issues with SDCR[INV] 9-12
9.10 SDRAM Interface 9-14
9.10.1 SDRAM Read Accesses 9-15
9.10.2 SDRAM Write Accesses 9-18
9.10.3 SDRAM Refresh Timing 9-20

Chapter 10
DMA Controller

10.1 DMA Data Transfer Types 10-1
10.2 DMA Address Modes 10-2
10.3 DMA Controller Registers 10-2
10.3.1 DMA Mode Register (DMR) 10-2
10.3.2 DMA Interrupt Register (DIR) 10-4
10.3.3 DMA Source Address Register (DSAR) 10-5
10.3.4 DMA Destination Address Register (DDAR) 10-6
10.3.5 DMA Byte Count Register (DBCR) 10-6

Chapter 11
Ethernet Module

11.1 Overview 11-1
11.1.1 Features 11-1
11.2 Module Operation 11-1
11.3 Transceiver Connection 11-3
11.4 FEC Frame Transmission 11-4
11.4.1 FEC Frame Reception 11-5
11.4.2 CAM Interface 11-6
11.4.3 Ethernet Address Recognition 11-6
11.4.4 Hash Table Algorithm 11-8

List of Tables (Continued)

Table Number	Title	Page Number
5-10	DBMR Field Descriptions	5-12
5-11	Access Size and Operand Data Location	5-12
5-12	PBR Field Descriptions	5-13
5-13	PBMR Field Descriptions	5-13
5-14	TDR Field Descriptions	5-14
5-15	Receive BDM Packet Field Description	5-18
5-16	Transmit BDM Packet Field Description	5-18
5-17	BDM Command Summary	5-19
5-18	BDM Field Descriptions	5-20
5-19	Control Register Map	5-30
5-20	Definition of DRc Encoding—Read	5-32
5-21	DDATA[3:0]/CSR[BSTAT] Breakpoint Response	5-34
5-22	PST/DDATA Specification for User-Mode Instructions	5-37
5-23	PST/DDATA Specification for Supervisor-Mode Instructions	5-40
6-1	SIM Registers	6-3
6-2	MBAR Field Descriptions	6-4
6-3	SCR Field Descriptions	6-5
6-4	SPR Field Descriptions	6-6
6-5	PMR Field Descriptions	6-8
6-6	USB and USART Power Down Modes	6-9
6-7	Exiting Sleep and Stop Modes	6-10
6-8	DIR Field Descriptions	6-11
6-9	WRRR Field Descriptions	6-12
6-10	WIRR Field Descriptions	6-13
6-11	WER Field Descriptions	6-13
7-1	Interrupt Controller Registers	7-2
7-2	Interrupt and Power Management Register Mnemonics	7-3
7-3	ICR Field Descriptions	7-4
7-4	ISR Field Descriptions	7-6
7-5	PITR Field Descriptions	7-7
7-6	PIWR Field Descriptions	7-8
7-7	PIVR Field Descriptions	7-9
7-8	MCF5272 Interrupt Vector Table	7-10
8-1	CSCR and CSOR Values after Reset	8-2
8-2	CSBRn Field Descriptions	8-3
8-3	Output Read/Write Strobe Levels versus Chip Select EBI Code	8-4
8-4	Chip Select Memory Address Decoding Priority	8-5
8-5	CSORn Field Descriptions	8-5
9-1	SDRAM Controller Signal Descriptions	9-2
9-2	Connecting BS[3:0] to DQMx	9-4
9-3	Configurations for 16-Bit Data Bus	9-4
9-4	Configurations for 32-Bit Data Bus	9-4
9-5	Internal Address Multiplexing (16-Bit Data Bus)	9-5
9-6	Internal Address Multiplexing (32-Bit Data Bus)	9-5

- System integration module (SIM)
  - System configuration including internal and external address mapping
  - System protection by hardware watchdog
  - Versatile programmable chip-select signals with wait-state generation logic
  - Up to three 16-bit parallel input/output ports
  - Latchable interrupt inputs with programmable priority and edge triggering
  - Programmable interrupt vectors for on-chip peripherals
- Physical layer interface controller (PLIC)
  - Allows connection using general circuit interface (GCI) or interchip digital link (IDL) physical layer protocols for 2B + D data
  - Three physical interfaces
  - Four time-division multiplex (TDM) ports
- IEEE 1149.1 boundary-scan test access port (JTAG) for board-level testing
- Operating voltage: 3.3 V ±0.3 V
- Operating temperature: 0 –70°C
- Operating frequency: DC to 66 MHz, from external CMOS oscillator
- Compact ultra low-profile 196 ball-molded plastic ball-grid array package (PGBA)

1.2 MCF5272 Architecture

This section briefly describes the MCF5272 core, SIM, UART, and timer modules, and test access port.

1.2.1 Version 2 ColdFire Core

Based on the concept of variable-length RISC technology, ColdFire combines the simplicity of conventional 32-bit RISC architectures with a memory-saving, variable-length instruction set. The main features of the MCF5272 core are as follows:

- 32-bit address bus directly addresses up to 4 Gbytes of address space
- 32-bit data bus
- Variable-length RISC
- Optimized instruction set for high-level language constructs
- Sixteen general-purpose 32-bit data and address registers
- MAC unit for DSP applications
- Supervisor/user modes for system protection
- Vector base register to relocate exception-vector table
- Special core interfacing signals for integrated memories
- Full debug support

2.7 Instruction Timing

The timing data presented in this section assumes the following:

- The OEP is loaded with the opword and all required extension words at the beginning of each instruction execution. This implies that the OEP spends no time waiting for the IFP to supply opwords and/or extension words.
- The OEP experiences no sequence-related pipeline stalls. For the MCF5272,the most common example of this type of stall involves consecutive store operations, excluding the MOVEM instruction. For all store operations (except MOVEM), certain hardware resources within the processor are marked as busy for two clock cycles after the final DSOC cycle of the store instruction. If a subsequent store instruction is encountered within this two-cycle window, it is stalled until the resource again becomes available. Thus, the maximum pipeline stall involving consecutive store operations is two cycles.
- The OEP can complete all memory accesses without memory causing any stall conditions. Thus, timing details in this section assume an infinite zero-wait state memory attached to the core.
- All operand data accesses are assumed to be aligned on the same byte boundary as the operand size:
  — 16-bit operands aligned on 0-modulo-2 addresses
  — 32-bit operands aligned on 0-modulo-4 addresses
  Operands that do not meet these guidelines are misaligned. Table 2-9 shows how the core decomposes a misaligned operand reference into a series of aligned accesses.

Table 2-9. Misaligned Operand References

A[1:0]	Size	Bus Operations	Additional C(R/W) [1]
x1	Word	Byte, Byte	2(1/0) if read 1(0/1) if write
x1	Long	Byte, Word, Byte	3(2/0) if read 2(0/2) if write
10	Long	Word, Word	2(1/0) if read 1(0/1) if write

[1] Each timing entry is presented as C(r/w), described as follows:
C is the number of processor clock cycles, including all applicable operand fetches and writes, as well as all internal core cycles required to complete the instruction execution.
r/w is the number of operand reads (r) and writes (w) required by the instruction. An operation performing a read-modify write function is denoted as (1/1).

ROMBAR can be configured similarly, as described in Section 4.4.2.2, “Programming ROMBAR for Power Management.”

4.4 ROM Overview

The ROM modules has the following features:

- 16-Kbyte ROM, organized as 4K x 32 bits
- Contains data tables for soft HDLC (high-level data link control)
- The ROM contents are not customizeable
- Single-cycle access
- Physically located on ColdFire core's high-speed local bus
- Byte, word, longword address capabilities
- Programmable memory mapping

4.4.1 ROM Operation

The ROM module contains tabular data that the ColdFire core can access in a single cycle. The ROM can be located on any 16-Kbyte address boundary in the 4-Gbyte address space. Section 4.1, “Interactions Between Local Memory Modules,” describes priorities when a fetch address hits multiple local memory resources.

4.4.2 ROM Programming Model

The MCF5272 implements the ROM base address register (ROMBAR), shown in Figure 4-2 and described in the following section.

4.4.2.1 ROM Base Address Register (ROMBAR)

ROMBAR determines the base address location of the internal ROM module, as well as the definition of the allowable access types. ROMBAR can be accessed in supervisor mode using the MOVEC instruction with an Rc value of 0xC00. It can also be read when the processor is in background debug mode (BDM). To access the ROM module, ROMBAR should be initialized with the appropriate base address.

	31–14	13–8	7–6	5	4	3	2	1	0
Field	BA	—	—	C/I	SC	SD	UC	UD	V
Reset	—		00	—	—	—	—	—	0
R/W	W for CPU; R/W for debug								
Address	CPU space + 0xC00								

Figure 4-2. ROM Base Address Register (ROMBAR)

ROMBAR fields are described in Table 4-4.

5.2 Signal Description

Table 5-1 describes debug module signals. All ColdFire debug signals are unidirectional and related to a rising edge of the processor core's clock signal. The standard 26-pin debug connector is shown in Section 5.8, "Freescale-Recommended BDM Pinout."

Table 5-1. Debug Module Signals

Signal	Description
Development Serial Clock (DSCLK)	Internally synchronized input. (The logic level on DSCLK is validated if it has the same value on two consecutive rising CLKIN edges.) Clocks the serial communication port to the debug module during packet transfers. Maximum frequency is 1/5 the processor status clock (PSTCLK) speed. At the synchronized rising edge of DSCLK, the data input on DSI is sampled and DSO changes state.
Development Serial Input (DSI)	Internally synchronized input that provides data input for the serial communication port to the debug module.
Development Serial Output (DSO)	Provides serial output communication for debug module responses. DSO is registered internally.
Breakpoint ($\overline{\text{BKPT}}$)	Input used to request a manual breakpoint. Assertion of $\overline{\text{BKPT}}$ puts the processor into a halted state after the current instruction completes. Halt status is reflected on processor status signals (PST[3:0]) as the value 0xF.
Processor Status Clock (PSTCLK)	Delayed version of the processor clock. Its rising edge appears in the center of valid PST and DDATA output. See Figure 5-2. PSTCLK indicates when the development system should sample PST and DDATA values.
Debug Data (DDATA[3:0])	These output signals display the register breakpoint status as a default, or optionally, captured address and operand values. The capturing of data values is controlled by the setting of the CSR. Additionally, execution of the WDDATA instruction by the processor captures operands which are displayed on DDATA. These signals are updated each processor cycle.
Processor Status (PST[3:0])	These output signals report the processor status. Table 5-2 shows the encoding of these signals. These outputs indicate the current status of the processor pipeline and, as a result, are not related to the current bus transfer. The PST value is updated each processor cycle.

Figure 5-2 shows PSTCLK timing with respect to PST and DDATA.

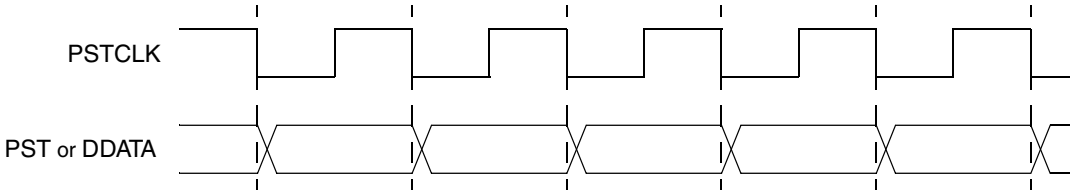

Figure 5-2. PSTCLK Timing

5.4.1 Revision A Shared Debug Resources

In the Revision A implementation of the debug module, certain hardware structures are shared between BDM and breakpoint functionality as shown in Table 5-4.

Table 5-4. Rev. A Shared BDM/Breakpoint Hardware

Register	BDM Function	Breakpoint Function
AATR	Bus attributes for all memory commands	Attributes for address breakpoint
ABHR	Address for all memory commands	Address for address breakpoint
DBR	Data for all BDM write commands	Data for data breakpoint

Thus, loading a register to perform a specific function that shares hardware resources is destructive to the shared function. For example, a BDM command to access memory overwrites an address breakpoint in ABHR. A BDM write command overwrites the data breakpoint in DBR.

5.4.2 Address Attribute Trigger Register (AATR)

The address attribute trigger register (AATR), Figure 5-5, defines address attributes and a mask to be matched in the trigger. The register value is compared with address attribute signals from the processor's local high-speed bus, as defined by the setting of the trigger definition register (TDR).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	RM	SZM		TTM		TMM			R	SZ		TT		TM		
Reset	0000_0000_0000_0101															
R/W	Write only. AATR is accessible in supervisor mode as debug control register 0x06 using the WDEBUG instruction and through the BDM port using the WDMREG command.															
DRc[4–0]	0x06															

Figure 5-5. Address Attribute Trigger Register (AATR)

Table 5-5 describes AATR fields.

Table 5-5. AATR Field Descriptions

Bits	Name	Description
15	RM	Read/write mask. Setting RM masks R in address comparisons.
14–13	SZM	Size mask. Setting an SZM bit masks the corresponding SZ bit in address comparisons.
12–11	TTM	Transfer type mask. Setting a TTM bit masks the corresponding TT bit in address comparisons.
10–8	TMM	Transfer modifier mask. Setting a TMM bit masks the corresponding TM bit in address comparisons.
7	R	Read/write. R is compared with the R/$\overline{W}$ signal of the processor's local bus.
6–5	SZ	Size. Compared to the processor's local bus size signals. 00 Longword 01 Byte 10 Word 11 Reserved

5.4.3 Address Breakpoint Registers (ABLR, ABHR)

The address breakpoint low and high registers (ABLR, ABHR), Figure 5-6, define regions in the processor's data address space that can be used as part of the trigger. These register values are compared with the address for each transfer on the processor's high-speed local bus. The trigger definition register (TDR) identifies the trigger as one of three cases:

1. identically the value in ABLR
2. inside the range bound by ABLR and ABHR inclusive
3. outside that same range

	31	0
Field	Address	
Reset	—	
R/W	Write only. ABHR is accessible in supervisor mode as debug control register 0x0C using the WDEBUG instruction and via the BDM port using the RDMREG and WDMREG commands. ABLR is accessible in supervisor mode as debug control register 0x0D using the WDEBUG instruction and via the BDM port using the WDMREG command.	
DRc[4–0]	0x0D (ABLR); 0x0C (ABHR)	

Figure 5-6. Address Breakpoint Registers (ABLR, ABHR)

Table 5-6 describes ABLR fields.

Table 5-6. ABLR Field Description

Bits	Name	Description
31–0	Address	Low address. Holds the 32-bit address marking the lower bound of the address breakpoint range. Breakpoints for specific addresses are programmed into ABLR.

Table 5-7 describes ABHR fields.

Table 5-7. ABHR Field Description

Bits	Name	Description
31–0	Address	High address. Holds the 32-bit address marking the upper bound of the address breakpoint range.

Command Sequence:

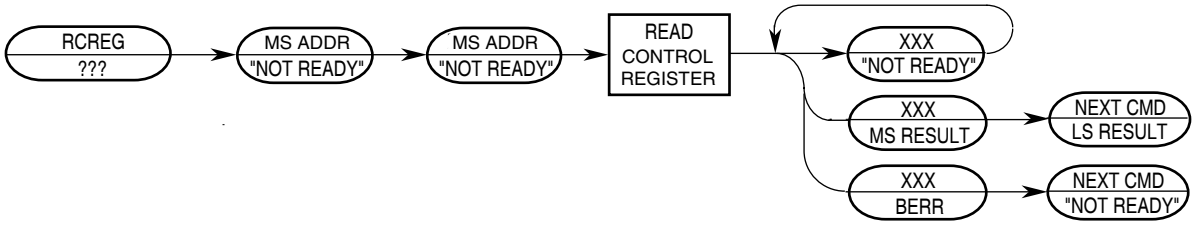

Figure 5-34. RCREG Command Sequence

Operand Data: The only operand is the 32-bit Rc control register select field.

Result Data: Control register contents are returned as a longword, most-significant word first. The implemented portion of registers smaller than 32 bits is guaranteed correct; other bits are undefined.

5.5.3.3.10 Write Control Register (WCREG)

The operand (longword) data is written to the specified control register. The write alters all 32 register bits.

Command/Result Formats:

	15–12	11–8	7–4	3–0
Command	0x2	0x8	0x8	0x0
	0x0	0x0	0x0	0x0
	0x0	Rc		
Result	D[31:16]			
	D[15:0]			

Figure 5-35. WCREG Command/Result Formats

Command Sequence:

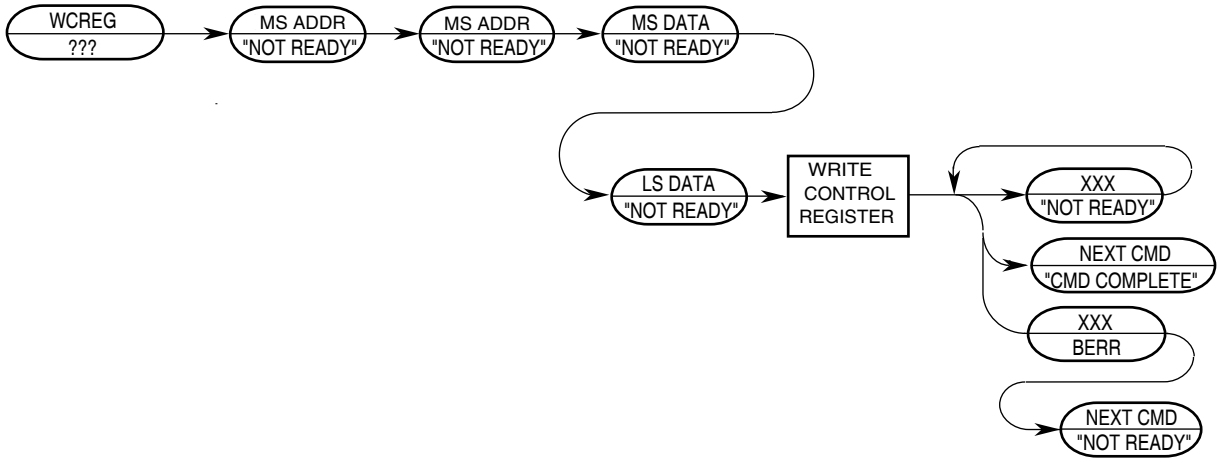

Figure 5-36. WCREG Command Sequence

Chapter 2, “ColdFire Core.” Pending interrupts from external sources ($\overline{\text{INT}}$[6:1]) can be cleared using the ICRs.

For an interrupt to be successfully processed, stack RAM must be available. A programmable chip select is often used for the RAM, in which case, the RAM is not immediately available at startup. Thus, no interrupts are recognized until PIVR is initialized. The RAM chip select and system stack should be set up before this initialization.

If more than one interrupt source has the same interrupt priority level (IPL), the interrupt controller daisy chains the interrupts with the priority order following the bit placement in the PIWR, with $\overline{\text{INT1}}$ having the highest priority and SWTO having the lowest priority, as shown in Figure 7-8.

7.2.1 Interrupt Controller Registers

This section describes the registers associated with the interrupt controller. Table 7-2 gives the nomenclature used for the interrupt and power management registers.

Table 7-2. Interrupt and Power Management Register Mnemonics

Mnemonic or Portion Thereof	Description
INT1, INT2, INT3, INT4, INT5, INT6	External interrupt signals 1–6.
TMR0, TMR1, TMR2, TMR3	Timers 3–0 from timer module
USB0, USB1, USB2, USB3, USB4, USB5, USB6, USB7	USB endpoint 0–7
UART1, UART2	UART1, UART2 modules
PLIP	PLIC 2-KHz periodic interrupt, 2B+D data
PLIA	PLIC asynchronous and maintenance channels interrupt
DMA	DMA controller interrupt
ETx	Ethernet module transmit data interrupt
ERx	Ethernet module receive data interrupt
ENTC	Ethernet module non-time-critical interrupt
QSPI	Queued serial peripheral interface
IPL2, IPL1, IPL0	Interrupt priority level bits 2–0
PI	Pending interrupt
PDN	Power down enable
WK	Wakeup enable
SWTO	Software watchdog timer time out

9.5.2 SDRAM Timing Register (SDTR)

The SDTR is used to configure SDRAM controller refresh counters for the type of SDRAM devices used and the number of clocks required for each type of SDRAM access. The reset value is 0x2115. For lower CPU clock frequencies, precharge and activate times can be reduced to eliminate up to 2 clock cycles from the read and write accesses. Consult the data sheets of the SDRAMs being considered.

	15–10	9–8	7–6	5–4	3–2	1–0
Write	RTP	RC	—	RP	RCD	CLT
Reset	0010_00	01	00	01	01	01
R/W	R/W					
Addr	MBAR + 0x0186					

Figure 9-4. SDRAM Timing Register (SDTR)

Table 9-8 describes SDTR fields.

Table 9-8. SDTR Field Descriptions

Bits	Name	Description
15–10	RTP	Refresh timer prescaler. Determines the number of clock cycles x 16 between refreshes. The following table describes different recommended prescaler settings for different clock frequencies including a margin of 1.2 μS. Recommended values are as follows: **RTP** / **15.6 μs = 1/f*RTP*16** / **System Clock** 111101 / 61 / 66 MHz 101011 / 43 / 48 MHz 011101 / 29 / 33 MHz 010110 / 22 / 25 MHz 000100 / 4 / 5 MHz (emulator)
9–8	RC	Refresh count. Indicates the number of clock cycles spent in refresh state (RC + 5). Refresh occurs during the first of these clock cycles; the rest of the time is the delay that must occur before the SDRAM is ready to do anything else. 00 5 cycles 01 6 cycles (default) 10 7 cycles 11 8 cycles
7–6	—	Reserved, should be cleared.
5–4	RP	Precharge time. Specifies number of clock cycles taken for a precharge (RP + 1). 00 1 cycle 01 2 cycles (default) 10 3 cycles 11 4 cycles

- Supports remote wakeup
- Detects start-of-frame and missed start-of-frame for isochronous endpoint synchronization
- Notification of start-of-frame, reset, suspend, and resume events

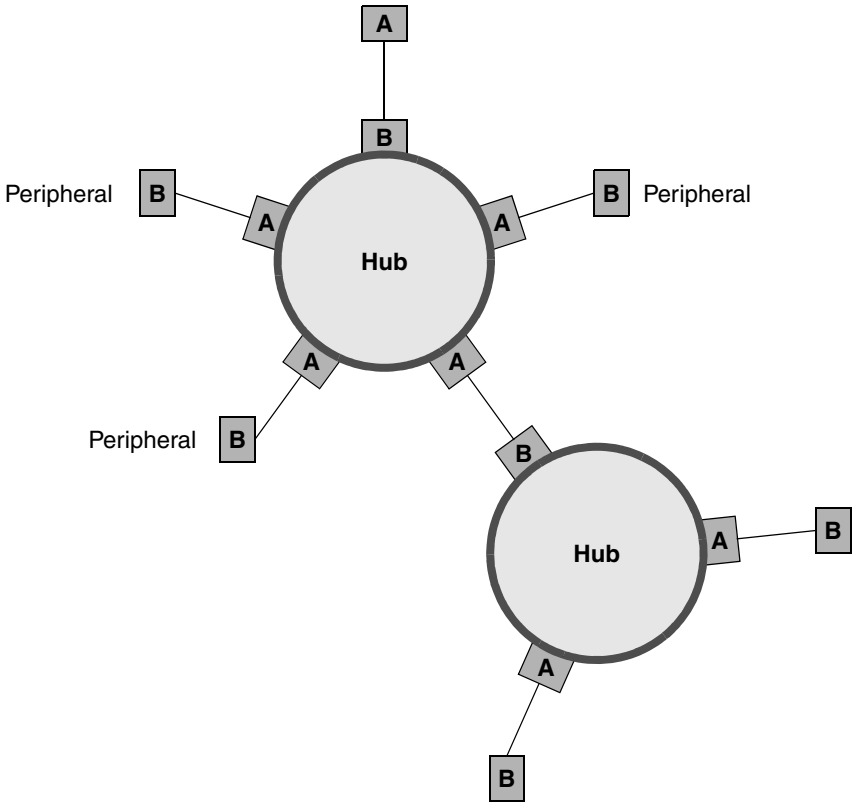

Figure 12-1. The USB "Tiered Star" Topology

12.2 Module Operation

The MCF5272 USB system consists of a protocol state machine which controls the transmitter and receiver modules. The state machine implements only the USB function state diagram. The MCF5272 USB controller can serve as a USB function endpoint, but cannot serve as a USB host.

12.2.1 USB Module Architecture

A block diagram of the USB module is shown in Figure 12-2. The module is partitioned into five functional blocks. These blocks are USB internal transceiver, clock generator, USB control logic, USB request processor, and endpoint controllers.

Table 12-12. EP0CTL Field Descriptions (continued)

Bits	Name	Description
7	CRC_ERR	CRC error generation enable. This bit enables CRC error generation for debug and test purpose. In order to use this feature, the DEBUG bit must be set. Enabling this bit causes a CRC error on the next data packet transmitted. The CRC_ERR bit must be set again in order to generate another CRC error. This bit only applies to IN transfers. This command bit is write-only and always returns 0 when read. 1 CRC error generation if DEBUG = 1 0 default value
6	—	Reserved, should be cleared.
5–4	OUT_LVL	Endpoint 0 OUT FIFO level for interrupt. This field selects the FIFO level to generate an OUT_LVL interrupt. The OUT_LVL interrupt is generated when the FIFO fills above the selected level. 00 FIFO **25%** Full 01 FIFO 50% Full 10 FIFO 75% Full 11 FIFO 100% Full
3–2	IN_LVL	Endpoint 0 IN FIFO level for interrupt. This field selects the FIFO level to generate an IN_LVL interrupt. The IN_LVL interrupt is generated when the FIFO falls below the selected level. 00 FIFO **25%** Empty 01 FIFO 50% Empty 10 FIFO 75% Empty 11 FIFO 100% Empty
1	IN_DONE	This bit controls the USB's response to IN tokens from the host. This bit is set at Reset and must be cleared by software when the last byte of a transfer has been written to the IN-FIFO. This bit is then subsequently set by the USB core when an end of transfer (EOT) event occurs indicating that the transfer has been completed. An end of transfer (EOT) event is indicated by one of the following: a) An IN packet is transmitted that contains less than the maximum number of bytes defined at endpoint configuration. b) A zero length IN packet is transmitted. This occurs when the previously transmitted IN packet was full, and no more data remains in the IN-FIFO. Hence a single zero length packet must be sent to indicate EOT. 0 CPU has completed writing to the IN-FIFO and transfer is in progress. The USB module will send any amount of data in the FIFO or a zero-length packet when the FIFO is empty. 1 Transfer completed or CPU Busy writing transfer into the IN-FIFO. The USB module will only send maximum size packets or NAK responses if the FIFO contains less than a maximum size packet. This bit is set at Reset and on an EOT event.
0	—	Reserved, should be cleared.

frame sync (offset with respect to the port 1 GCI/IDL block). Port 3 can also have dedicated data in and data out pins, DIN3 and DOUT3 of pin set 3 (see Section 13.5.7, “Port Configuration Registers (P0CR–P3CR)”). This allows the MCF5272 to connect to ISDN NT1s that have a common frame sync and clock, but two sets of serial data-in and data-out pins.

The MCF5272 PLIC provides two sets of D-channel arbitration control pins:

- DREQ0 and DGNT0 for pin set 0
- DREQ1 and DGNT1 for pin set 1

Because pin set 1 connects ports 1, 2, and 3, these ports do not have D-channel arbitration control signals.

13.2 GCI/IDL Block

This section describes the GCI/IDL block.

13.2.1 GCI/IDL B- and D-Channel Receive Data Registers

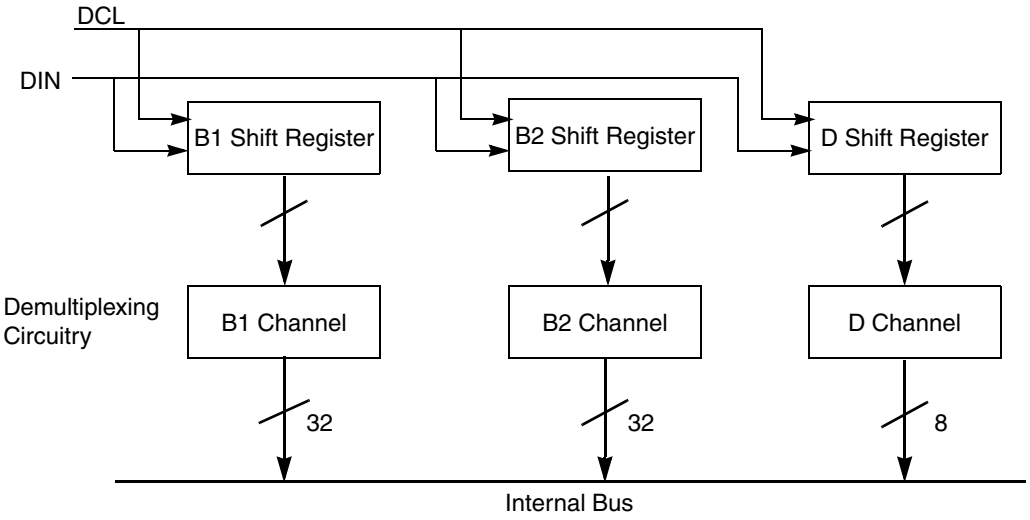

Figure 13-2. GCI/IDL Receive Data Flow

The maximum data rate received for each GCI/IDL port is 144 Kbps: the sum of two 64-Kbps B channels and one 16-Kbps D-channel. Frames of B_1 and B_2 channels are packed together to form longwords (32 bits). Frames of D-channels are packed together to form bytes. For channels B and D, this requires CPU service at a 2-KHz rate, because it requires four frames to fill the 32-bit B-channel register and the 8-bit D-channel register.

The CPU should service the B1 and B2 registers once every 500 μS. Overrun conditions can be avoided only if the CPU services these registers in a timely manner.

The MCF5272 has 4 GCI/IDL interfaces. Thus the theoretical maximum is twelve 32-bit data registers to be read. For most applications the typical number is less.

13.5.6 D Data Transmit Registers (P0DTR–P3DTR)

All bits in these registers are read/write and are set on hardware or software reset.

The PLTD registers contain four frames of D-channel transmit data, packed from lsb to msb, for each of the four physical ports on the MCF5272. P0DTR is the D-channel byte for port 0, P1DTR the D channel for port 1, and so on.

The four byte-addressable 8-bit registers, P0DTR–P3DTR, are packed to form one 32-bit register, PLTD. PLTD is aligned on a long-word boundary at MBAR + 0x348 and can be read as a single 32-bit register. P0DTR is located in the MSB of the PLTD register, P3DTR is located in the LSB of the PLTD register.

	31 – 24	23 – 16
Field	P0DTR	P1DTR
Reset	1111_1111	1111_1111
R/W	Read/Write	

	15 – 8	7 – 0
Field	P2DTR	P3DTR
Reset	1111_1111	1111_1111
R/W	Read/Write	
Addr	MBAR + 0x348 (P0DTR); 0x349 (P1DTR); 0x34A (P2DTR); 0x34B (P3DTR)	

Figure 13-18. D Transmit Data Registers P0DTR–P3DTR

13.5.7 Port Configuration Registers (P0CR–P3CR)

	15	14–12	11	10	9	8	7	6	5	4	3	2	1	0
P0CR	ON/OFF	M	—	G/S	—	ACT	—	—	—	—	SHB2	SHB1	ENB2	ENB1
P1CR	ON/OFF	— M	M/S	G/S	FSM	ACT	—	—	—	—	SHB2	SHB1	ENB2	ENB1
P2CR	ON/OFF	— — —	—	—	—	—	—	—	—	—	SHB2	SHB1	ENB2	ENB1
P3CR	ON/OFF	— — —	—	—	—	—	DMX	—	—	—	SHB2	SHB1	ENB2	ENB1
Reset	0000_0000_0000_0000													
R/W	Read/Write													
Addr	MBAR + 0x350 (P0CR); 0x352 (P1CR); 0x354 (P2CR); 0x356 (P3CR)													

Figure 13-19. Port Configuration Registers (P0CR–P3CR)

P*n*CR are registers containing configuration information for each of the four ports on the MCF5272.

All bits in these registers are read/write and are cleared on hardware or software reset.

Table 13-2. P0CR–P3CR Field Descriptions

Bits	Name	Description
15	ON/OFF	0 Port is off and in a steady state condition. In this state, the B and D channels on the transmit side are high impedance when in GCI/IDL. The receive registers are all set. In IDL and GCI modes with the port in this state, all periodic and aperiodic interrupts associated with the port are disabled. 1 Switches on the port for operation in the configured mode.
14–12	M	Mode. Selects between various modes of operation as described below. Note: bit 14 is relevant to port 0 only. The IDL modes on the PLIC only support short frame sync. **Port 1-3 Port 0** 000 IDL8IDL8 001 IDL10IDL10 010 GCIGCI 011 ReservedReserved 10*x* ReservedReserved 11*x* ReservedReserved
11	M/S	Master/Slave. Defines the direction of the DCL1 and FSC1 pins. 0 DCL1 and FSC1 are inputs and are sourced from an external master. Note: This bit is relevant to port 1 only, as port 0 is always in slave mode. 1 enables DCL1 and FSC1 to be outputs, that is, the MCF5272 drives DCL1 and FSC1.
10	G/S	GCI/SCIT. 0 The normal mode of GCI is used (i.e. no D-channel contention control). 1 Selects SCIT mode of operation for the GCI interfaces.
9	FSM	Frame Sync Master. 0 Default reset value. 2-KHz interrupt is generated from port 0. 1 Port 1 FSC/FSR is used to generate the 2-KHz interrupt.
8	ACT	GCI Activation. 0 Default reset value. 1 Causes Dout to transition to a logic low for the respective port. This bit is only operational when the port is in GCI mode. Setting the ACT bit in any other mode has no effect. It is the responsibility of the CPU to clear the ACT bit when normal operation on Dout is required. This bit is intended to be used to request activation from the upstream DCL/FSC driver. Periodic interrupts commence as soon as the upstream device generates DCL, provided the appropriate interrupts, such as IE, B1RIE, and so on, are enabled for the port.
7	DMX	Data multiplex. 0 port 3 Dout and Din are multiplexed onto Dout1 and Din1. 1 enables port 3 Dout and Din to be connected to dedicated output and input pins, DOUT3 and DIN3.
3	SHB2	B2 channel shift direction. 0 B2 channel data is received/transmitted msb first. The msb-first convention is often used for communication with PCM CODECs and converters. 1 B2 channel data is received/transmitted lsb first. The lsb-first convention is used when the data is to be HDLC encoded.
2	SHB1	B1 channel shift direction. See SHB2.
1	ENB2	Enable B2 data channel. 0 The B2 channel is disabled and all periodic interrupts in both receive and transmit directions are disabled. The behavior of Din and Dout in this state is shown below. **Mode / Din / Dout** IDL / All 1s / High Impedance GCI / Operational (data on Din visible) / Open drain 1 Enables the B2 data channel for the respective port.
0	ENB1	Enable B1 data channel. See ENB2.

16.3.5 UART Command Registers (UCR*n*)

The UART command registers (UCR*n*), Figure 16-6, supply commands to the UART. Only multiple commands that do not conflict can be specified in a single write to a UCR*n*. For example, RESET TRANSMITTER and ENABLE TRANSMITTER cannot be specified in one write.

	7	6	4	3	2	1	0
Field	ENAB	MISC		TC		RC	
Reset	0000_0000						
R/W	Write only						
Address	MBAR + 0x108, 0x148						

Figure 16-6. UART Command Registers (UCR*n*)

Table 16-6 describes UCR*n* fields and commands. Examples in Section 16.5.2, “Transmitter and Receiver Operating Modes,” show how these commands are used.

Table 16-6. UCR*n* Field Descriptions

Bits	Value	Command	Description
7	ENAB	—	Enable autobaud 0 Autobaud disabled. 1 Autobaud enabled. The transmission rate is calculated from the first received character. If the rate must be recalculated, ENAB must first be cleared and reset. UISR*n*[ABC] indicates a transmission rate has been calculated and loaded into the UART divider registers. UDU*n* and UDUL*n* must be initialized to 0x00 before enabling autobaud.
6–4	**MISC Field** (This field selects a single command.)		
	000	no command	—
	001	reset mode register pointer	Causes the mode register pointer to point to UMR1*n*.
	010	reset receiver	Immediately disables the receiver, clears USR*n*[FFULL,RxRDY], and reinitializes the receiver FIFO pointer. No other registers are altered. Because it places the receiver in a known state, use this command instead of RECEIVER DISABLE when reconfiguring the receiver.
	011	reset transmitter	Disables the transmitter and clears USR*n*[TxEMP,TxRDY]. No other registers are altered. Because it places the transmitter in a known state, use this command instead of TRANSMITTER DISABLE when reconfiguring the transmitter.
	100	reset error status	Clears USR*n*[RB,FE,PE,OE]. Also used in block mode to clear all error bits after a data block is received.
	101	reset break– change interrupt	Clears the delta break bit, UISR*n*[DB].
	110	start break	Forces TxD low. If the transmitter is empty, the break may be delayed up to one bit time. If the transmitter is active, the break starts when character transmission completes. The break is delayed until any character in the transmitter shift register is sent. Any character in the transmitter holding register is sent after the break. The transmitter must be enabled for the command to be accepted. This command ignores the state of $\overline{\text{CTS}}$.
	111	stop break	Causes TxD to go high (mark) within two bit times. Any characters in the transmitter buffer are sent.

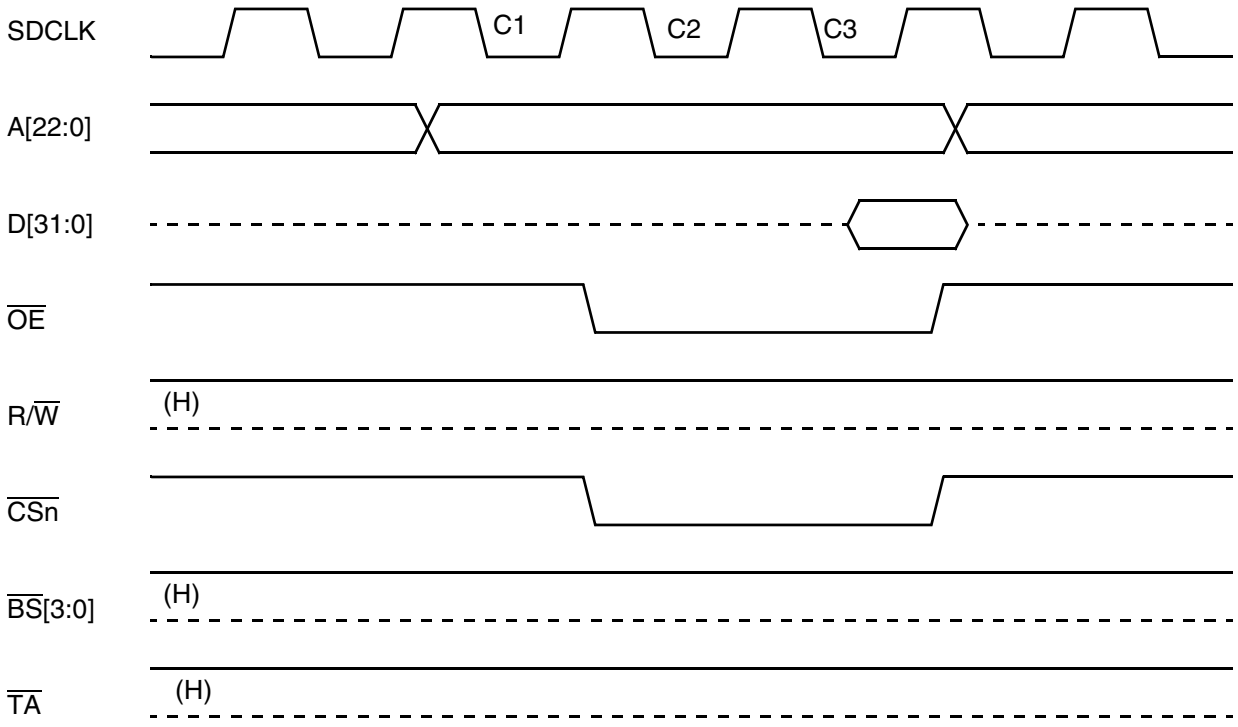

Figure 20-12. Read with Address Setup; EBI=11; 32-Bit Port; Internal Termination

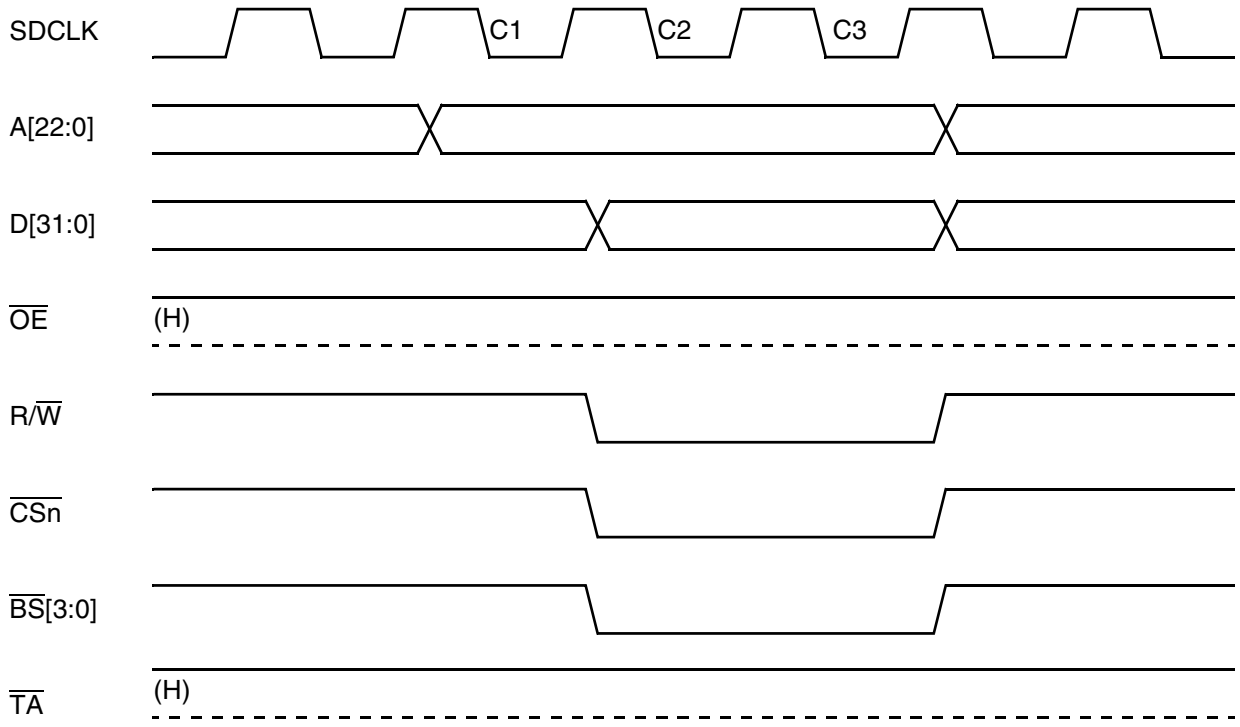

Figure 20-13. Longword Write with Address Setup; EBI=11; 32-Bit Port; Internal Termination

Table A-14. PLIC Module Memory Map (continued)

MBAR Offset	[31:24]	[23:16]	[15:8]	[7:0]
0x038C	Aperiodic Interrupt Status Register (PASR)		Reserved	Loop back Control (PLCR)
0x0392	Reserved		D Channel Request (PDRQR)	
0x0394	Port0 Sync Delay (P0SDR)		Port1 Sync Delay (P1SDR)	
0x0398	Port2 Sync Delay (P2SDR)		Port3 Sync Delay (P3SDR)	
0x039C	Reserved		Clock Select (PCSR)	

Table A-15. Ethernet Module Memory Map

MBAR Offset	[31:24]	[23:16]	[15:8]	[7:0]
0x0840	Ethernet Control Register (ECR)			
0x0844	Ethernet Interrupt Event Register (EIR)			
0x0848	Ethernet Interrupt Mask Register (EIMR)			
0x084C	Ethernet Interrupt Vector Status (IVSR)			
0x0850	Ethernet Rx Ring Updated Flag (RDAR)			
0x0854	Ethernet Tx Ring Updated Flag (TDAR)			
0x0880	Ethernet MII Data Register (MMFR)			
0x0884	Ethernet MII Speed Register (MSCR)			
0x08CC	Ethernet Receive Bound Register (FRBR)			
0x08D0	Ethernet Rx FIFO Start Address (FRSR)			
0x08E4	Transmit FIFO Watermark (TFWR)			
0x08EC	Ethernet Tx FIFO Start Address (TFSR)			
0x0944	Ethernet Rx Control Register (RCR)			
0x0948	Maximum Frame Length Register (MFLR)			
0x0984	Ethernet Tx Control Register (TCR)			
0x0C00	Ethernet Address (Lower) (MALR)			
0x0C04	Ethernet Address (Upper) (MAUR)			
0x0C08	Ethernet Hash Table (Upper) (HTUR)			
0x0C0C	Ethernet Hash Table (Lower) (HTLR)			
0x0C10	Ethernet Rx Descriptor Ring (ERDSR)			
0x0C14	Ethernet Tx Descriptor Rin (ETDSR)			
0x0C18	Ethernet Rx Buffer Size (EMRBR)			
0x0C40–0x0DFF	FIFO RAM (EFIFO)			