COMPACTPCI-810
SYSTEM CONTROLLER
USER'S MANUAL

Revision 1.1, June 2001

Cyclone P/N 800-0810

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
GENERAL INTRODUCTION

1.1 INTRODUCTION

The CPCI-810 is a high-performance CompactPCI System Controller. A block diagram is shown in Figure 1-1.

The board is based on the MPC8240 PowerPC™ integrated processor. The MPC8240 has a processor core based on the PowerPC603e™ low-power microprocessor, and also performs many peripheral functions on chip. The peripheral logic integrates a PCI bridge, memory controller, DMA controller, interrupt controller, I_2O controller, and an I^2C controller. I/O expansion is provided with a PMC Module. The PMC Module location allows the CPCI-810 to be configured for custom applications.

Additionally, the CPCI-810 contains two Intelligent 10/100BaseT Ethernet Controllers and can be used as a processing engine for an embedded systems platform, which requires Ethernet connectivity.

Software development tools for PowerPC processors are available from a variety of vendors, and Board Support Packages (BSPs) for the PSOS operating system is available from Cyclone.

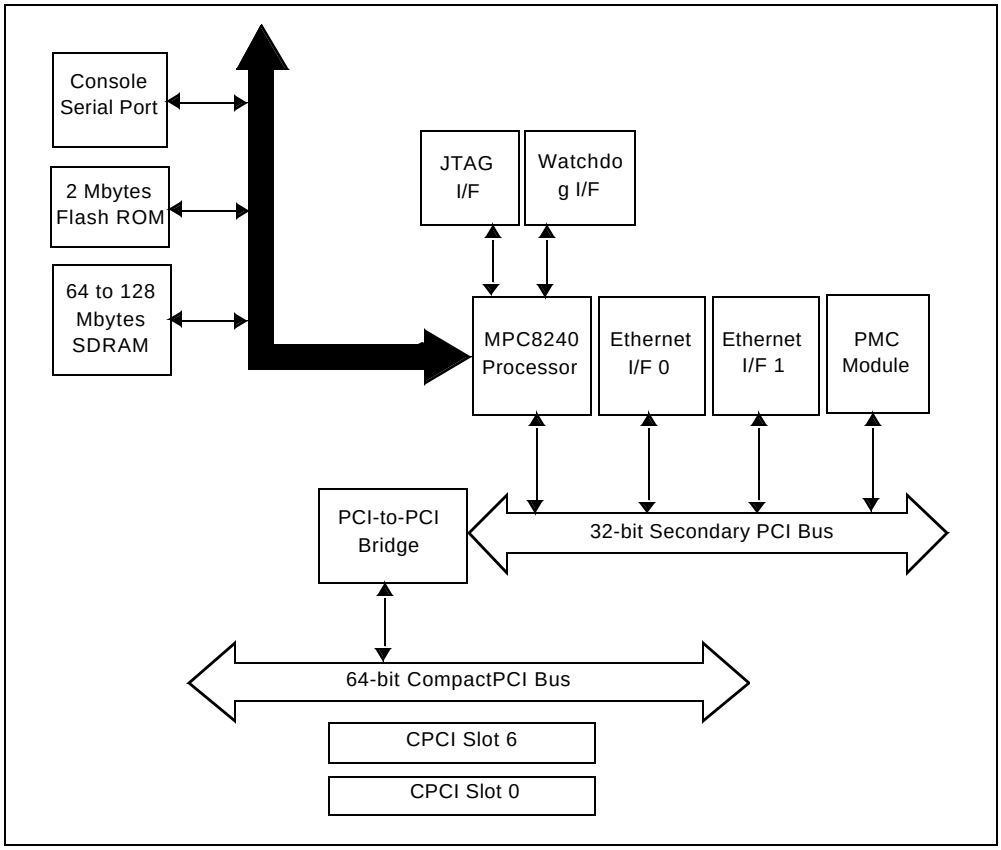

Figure 1-1. CPCI-810 Block Diagram

1.2 FEATURES

• MPC8240 Processor	The microprocessor is Motorola's integrated MPC8240 PowerPC. The device integrates a Motorola 32-bit superscalar PowerPC 603e core, running at 250 MHz internally, and a Peripheral Components Interconnect (PCI). The core boasts a 16 Kbyte instruction cache, a 16 Kbyte data cache and floating-point support. Memory can be accessed through the memory controller to the core processor or from the PCI bus.
• 21154 PCI-to-PCI Bridge	The 21154 has a 64-bit primary bus interface and a 64-bit secondary interface. The 21154 primary bus interfaces with the 64-bit CompactPCI bus and the secondary bus interfaces with the 32-bit PCI bus on the MPC8240. This allows the CPCI-810 to function as a system controller in a CompactPCI system.
• SDRAM	64 MBytes of SDRAM expandable to 192 MBytes via a 144 pin SoDIMM running at 100MHz.
• PMC Module	The CPCI-810 has one location for a 32-bit PMC Module. The module can contain I/O interfaces to customize the CPCI-810 for different applications. The PMC Module resides on the PCI bus of the MPC8240 and can be accessed by the MPC8240 or a host on the *CompactPCI* bus. Devices on the PMC Module can DMA data into local memory or through the bridge to host memory.
• *CompactPCI* Interface	The CPCI-810 meets the PICMG Rev. 1.0 Specification for system slot adapters. The PCI bus runs at 33MHz.
• Flash ROM	2 Mbytes of in-circuit sector-programmable Flash ROM.
• Ethernet Ports	Two 10/100BaseTx ethernet ports are provided. Each port supports up to 100Mbps a uses a RJ45 style module phone jack
• Console Serial Port	An RS-232 serial port is provided for a console terminal or workstation connection. The serial port supports up to 115 Kbps and uses a phone jack to DB25 cable supplied with the CPCI-810 board.
• LM75	Two LM75s are provided for temperature monitoring.
• Fan Monitoring	Fan monitoring is provided for two separate fan frequency inputs.
• Timers	Four 31-bit timers are available to generate interrupts.
• DMA Controller	The MPC8240 supports 2 separate DMA channels for high throughput data transfers between PCI bus agents and the local SDRAM memory.
• I_2O Messaging	The CPCI-810 supports the I_2O specification for interprocessor communication.

1.3 OVERVIEW

The CPCI-810 is 6U CompactPCI System adapter card with support for one 32-bit 3.3V signaling PMC Module. PMC Modules are available from many vendors, including Cyclone and the PMC interface specification is included in Appendix A for those who wish to design their own PMC Modules.

The CPCI-810 is configured as a System controller board. Therefore, the CPCI-810 provides the interrupt, arbitration, clocking and reset function for the other (peripheral) adapters in a CompactPCI system.

The CPCI-810 has two PCI buses. The Primary PCI bus is the CompactPCI bus. The Secondary PCI bus supports the PMC module and the two Ethernet interfaces.

The primary PCI interface is 64-bit data. The CPCI-810 controller also has a 64-bit data path to memory. The secondary (local) PCI bus is a 32-bit data connecting to the PMC Module and also to the 32-bit data Ethernet controllers.

The CPCI-810 interfaces to the *CompactPCI*Ô bus using an Intel 21154 PCI-to-PCI Bridge. This device complies with the PCI Local Bus Specification, revision 2.1, provides concurrent bus operation, allows buffering for both read and write transactions and provides the arbitration for the CompactPCI bus devices.

The CPCI-810 provides for a number of system hardware monitors. There are two circuits provided to monitor the health of power supplies. There are also two circuits provided to monitor a frequency output from two fans. Additionally, two LM75s are provided to monitor temperature.

The Flash ROM on the CPCI-810 can be reprogrammed by software through the JTAG/COP interface. Utilities to perform this programming are available from software development tool vendors. Additional information on the JTAG/COP interface can be found in section 3.8.4

1.4 SPECIFICATIONS

Physical Characteristics

The CPCI-810 is a single slot, double high *CompactPCI*Ô card with a system slot interface. This product is equipped with an Intel i960 microprocessor and 1 PMC location. The PMC has P3 I/O capability.

Height:	9.187” (233.35mm) Double Eurocard (6U)
Depth:	6.299” (160mm)
Width:	.8” (20.32mm)

Power Requirements

The CPCI-810 requires +5V, +12V, -12V and +3.3V from the *CompactPCI*Ô backplane J1 connector.

The following figures do not include the power consumption of any PMC Module installed.

Table 1-1. CPCI-810 Power Requirements

Voltage	Current Typical	Current Maximum
+3.3V	3.10 Amps	4.30 Amps
+5V	0.128 Amps	0.129 Amps
+12V	0 Amps	0 Amps
-12V	0 Amps	0 Amps

1.5 ENVIRONMENTAL

The CPCI-810 should be operated in a CompactPCI card cage with good air flow. The board can be operated at ambient air temperature of 0-55 degrees Celsius, as measure at the board.

Table 1-2. ENVIRONMENTAL SPECIFICATIONS

Operating Temperatures	0 to 55 Degrees Celsius
Relative Humidity (non-condensing)	0-95%
Storage Temperatures	-55 to 125 Degrees Celsius

1.6 PHYSICAL ENVIRONMENT

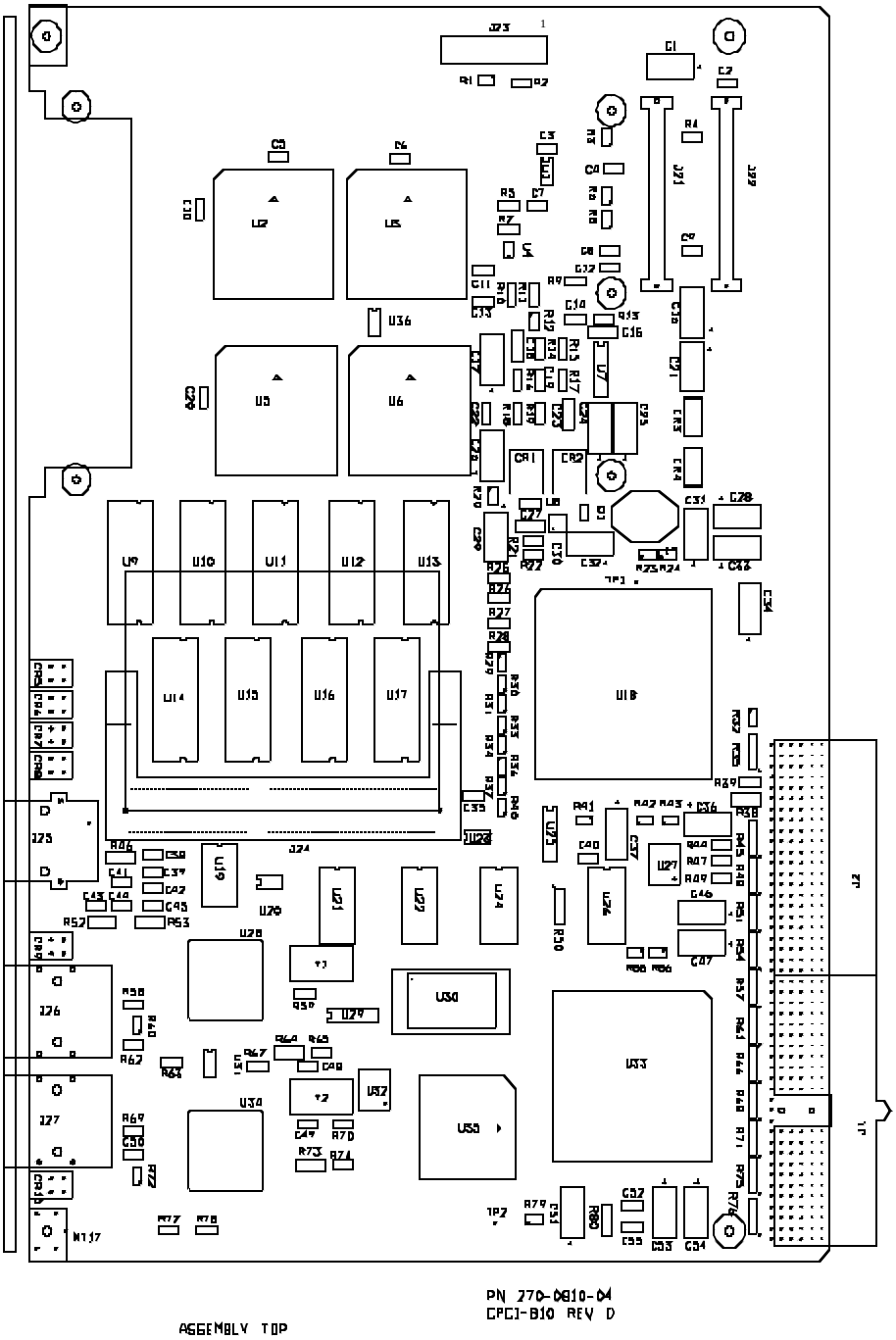

Figure 1-2. Physical Configuration

Figure 1-2 is a physical diagram of the CPCI-810 Adapter, showing the location designators of jumpers, connectors, and ICs. Refer to this figure when component locations are referenced in the manual text.

1.7 REFERENCE MANUALS

MPC8240 Integrated Processor User's Manual
Order Number MPC8240UM/D Rev. 0
Motorola Literature Distribution
P.O. Box 5405
Denver, CO 80217
(800) 441-2447

PowerPC Microprocessor Family:
The Programming Environments for
32-BIT Microprocessors, Rev. 1
Order Number MPCFPE32B/AD
Motorola Literature Distribution
P.O. Box 5405
Denver, CO 80217
(800) 441-2447

TL16C550C UART
Texas Instruments
http://www.ti.com/sc/docs/general/dsmenu.htm

IEEE STD P1386, Draft 2.0
IEEE STD P1386.1, Draft 1.5
Institute of Electrical and Electronics Engineers
PO Box 1331
445 Hoes Lane
Piscataway, NJ 08855-1331

82559 Software Developers Manual
Order Number 743892-002
Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect, IL 60056-7641
(800) 879-4683

LM75 Digital Temperature Sensor and
Thermal Watchdog
National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
(800) 272-9959

CompactPCI Ô Specification
PCI Industrial Computers Manufacturing Group
301 Edgewater Place, Suite 220
Wakefield, MA 01880
(617) 224-1100
(617) 224-1239 Fax

PCI Local BIOS Specification, Revision 2.1
PCI Special Interest Group
2575 NE Kathryn Street #17
Hillsboro, OR 97214
(800) 433-5177 (U.S.)
(503) 693-6232 (International)
(503) 693-8344 (Fax)

I_2O Specification, Revision 1.0
I_2O Special Interest Group
(415) 750-8352
http://www.i2osig.org

CompactPCI Ô Hot Swap Specification, PICMG
2.1, R1.0
PCI Industrial Computers Manufacturing Group
301 Edgewater Place, Suite 220
Wakefield, MA 01880
(617) 224-1100
(617) 224-1239 Fax

PMC on *CompactPCI* Ô Specification, PICMG
2.3, R1.0
PCI Industrial Computers Manufacturing Group
301 Edgewater Place, Suite 220
Wakefield, MA 01880
(617) 224-1100
(617) 224-1239 Fax

CHAPTER 2
MPC8240 PROCESSOR

2.1 MPC8240 PROCESSOR

The MPC8240 contains a PowerPC 603e core processor. The core is configured to run at 250 MHz. This RISC processor utilizes a superscalar architecture that can issue and retire as many as three instructions per clock. The core features independent 16 Kbyte, four-way set-associative, physically addressed caches for instructions and data and on-chip instruction and data memory management units (MMUs).

2.2 BYTE ORDERING

The CPCI-810 is designed to run in big endian mode. The byte ordering determines how the core accessses local memory and the PCI bus. Big endian stores the most significant byte in the lowest address.

2.3 RESET VECTOR

The 8-bit wide Flash ROM is located in the address range FFE0 0000h through FFFF FFFFh. See Figure 2.1, the CPCI-810 memory map. The MPC8240 reset vector is located at address FFF0 0100h. This reset vector location, which contains a branch to the rest of the boot code, is essentially in the middle of the ROM device. This positioning results in a break up of continuous memory space and approximately 50% reduction in usable space for boot code. To better utilize this device, the CPCI-810 re-maps the reset vector to FFE0 0100h by inverting memory address 20 (A20) for the first two processor accesses to memory. These accesses are an absolute jump instruction to the beginning of boot code. After this jump A20 functions normally. Utilizing this method the majority of the 2 Mbyte Flash ROM can be used.

2.4 POWERPC MPC603E CORE CACHE, BUFFERS, ARRAYS

The processor core provides independent on-chip, 16-Kbyte, four-way set-associative, physically addressed caches for instructions and data and on-chip instruction and data memory management units (MMUs). The MMUs contain 64-entry, two-way set associative, data and instruction lookaside buffers (TLB) that provide support for demand-paged virtual memory address translation and variable-sized block translation. The processor also supports block address translation (BAT) arrays of four entries each.

As an added feature to the MPC603e core, the MPC8240 can lock the contentes of one to three ways in the instruction and data cache (or the entire cache).

2.5 MEMORY MAP

Figure 2-1 shows the CPCI-810 memory map.

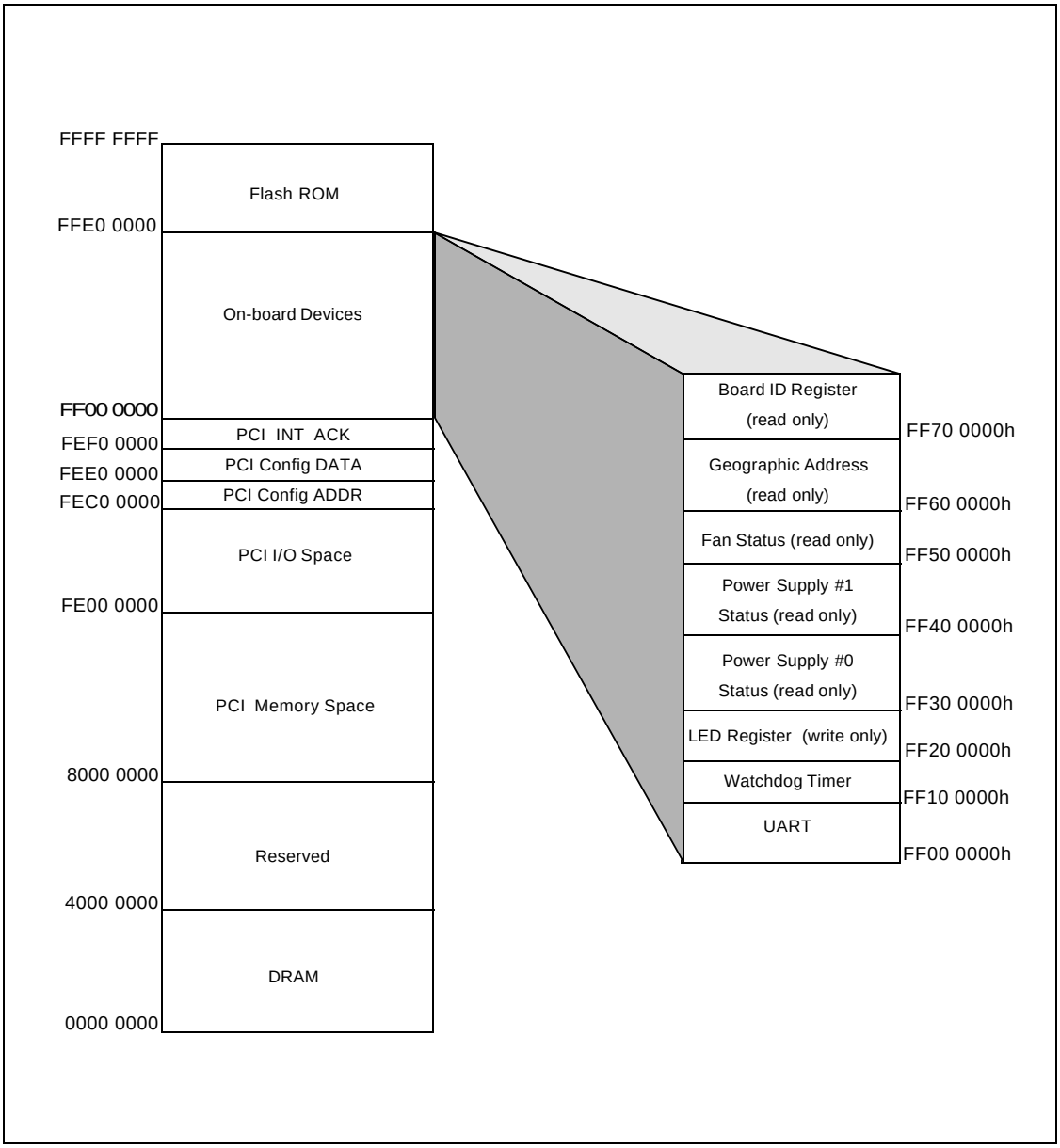

Figure 2-1. CPCI-810 Memory Map

2.6 INTERRUPTS

The CPCI-810 interrupt scheme is based upon the MPC8240 processor's embedded programmable interrupt controller (EPIC). The EPIC unit is set in the serial interrupt mode. The serial interrupt mode allows for a maximum of 16 external interrupts. Table 2-1 shows the assignment for the serial interrupts.

The EPIC interface also contains several internal interrupt sources. These include the four global timers, the two DMA channels, the I^2C bus, and from the Message Unit.

In addition to the EPIC interface, errors detected by the MPC8240 are reported to the processor core by asserting an internal machine check signal Many of the errors detected in the MPC8240 cause exceptions to be taken by the processor core. The error reporting is provided for three of the primary interfaces, processor core interface, memory interface, and the PCI interface.

Table 2-1. Serial Interrupt Assignment

EPIC Serial Interrupt	Interrupt Source	Vector Number	Polarity
0	Primary PCI INTA	0x10	0
1	Primary PCI INTB	0x11	0
2	Primary PCI INTC	0x12	0
3	Primary PCI INTD	0x13	0
4	PMC INTA	0x14	0
5	PMC INTB	0x15	0
6	PMC INTC	0x16	0
7	PMC INTD	0x17	0
8	Ethernet 0 INT	0x18	0
9	Ethernet 1 INT	0x19	0
10	Temperature INT	0x1A	0
11	ENUM	0x1B	0
12	UART INT	0x1C	1
13	Fan INT	0x1D	0
14	Power Supply INT	0x1E	0
15	Not Used	0x1F	x

2.6.1 Shared PMC Interrupts

The PMC interrupts, INTA, INTB, INTC, and INTD, are received on serial interrupts 4 through 7. To remain in compliance with the PCI-to-PCI Bridge Specification, Cyclone has implemented the following SPCI interrupt mapping scheme.

Each PMC module designed by Cyclone Microsystems follows the interrupt mapping scheme described in Table 2-2. Note that each PMC interrupt pin is mapped in a different way to both of the two possible PCI devices on the PMC module. This is to prevent interrupt sharing whenever possible. Users wishing to design their own PMC modules should follow this convention if they would like to take advantage of the PSOS driver interrupt dispatch support.

Table 2-2. PMC Module Interrupt Mapping

PMC Intpin	IDSEL0 Interrupts	IDSEL1 Interrupts
INTA	INTA	INTD
INTB	INTB	INTA
INTC	INTC	INTB
INTD	INTD	INTC

The IDSEL assignments for each device on the PMC module are described in Table 2-3. IDSEL Assignments

Table 2-3. PMC IDSEL Assignments

PMC IDSEL Pin	PMC IDSEL Assignment
IDSEL0	AD12
IDSEL1	AD13

2.6.2 MPC8240 Interrupt Registers

The MPC8240 processor has several different EPIC register maps to facilitate the handling of interrupts which are briefly mentioned below. These registers occupy a 256 Kbyte range of the embedded utilities memory block (EUMB) and can be read and written by software. Please refer to the Motorola MPC8240 User's manual for more detail.

Global EPIC Registers — Provides programming control for resetting, configuration and initialization of the external interrupts. Additionally, a vector register is provided to be returned to the processor during an interrupt acknowledge cycle for a spurious vector.

Global Timer Registers — Each of the four global timers have four individual configuration registers. The registers are the Current Count register, the Base Count register, the Vector/Priority register, and the Destination register.

Interrupt Source Configuration — This group of registers are made up of the vector/priority and destination registers for the serial and internal interrupt sources. This includes the masking, polarity, and sense.

Processor-Related Registers — This group describes the processor-related EPIC registers. They are made up of the Current Task Priority register, the Interrupt Acknowledge register, and the End of Interrupt register.

2.6.3 Error Handling and Exceptions

Errors detected by the MPC8240 are reported to the processor core by asserting an internal machine check signal (mpc#). The MPC8240 detects illegal transfer types from the processor, illegal Flash write transactions, PCI address and data parity errors, accesses to memory addresses out of the range of physical memory, memory parity errors, memory refresh overflow errors, ECC errors, PCI master-abort cycles, and PCI received target-abort errors. Table 2-4 describes the relative priorities and recoverablity of externally-generated errors and exceptions.

Table 2-4. Error Priorities

Priority	Exception	Cause
0	Hard reset	Power-on reset, CompactPCI chassis reset switch or via JTAG controller
1	Machine check	Processor transaction error, or Flash error
2	Machine check	PCI address parity error or PCI data parity error when the CPCI-810 is acting as the PCI target
3	Machine check	Memory select error, memory refresh overflow, or ECC error
4	Machine check	PCI address parity error or PCI data parity error when the CPCI-810 is acting as the PCI master, PCI master-abort, or received PCI target-abort

CYCLONE Microsystems

CHAPTER 3
HARDWARE

3.1 SDRAM

The CPCI-810 is equipped with 64 Mbytes of SDRAM. The memory is made up of nine 64 Mbit (8M x 8) devices. The extra eight bits are for ECC data. Memory may be expanded by adding up to 128 Mbytes of SDRAM to the 144 pin SoDIMM socket for a maximum of 196 Mbytes. The SDRAM is accessible by the processor and the PCI bus.

The CPCI-810 uses 72-bit SDRAM with ECC or 64-bit SDRAM without ECC. SDRAM memory bus requires a four-beat data burst. The memory controller unit (MCU) of the CPCI-810 supports SDRAM burst lengths of four. A burst length of four enables seamless read/write bursting of long data streams as long as the MCU does not cross the page boundary. Page boundaries are naturally aligned 2 Kbyte blocks. 72-bit SDRAM with ECC running at 100 MHz allows a maximum throughput of 800 Mbytes per second.

16 Mbyte, 64 Mbyte, and 128 Mbyte SDRAM devices are supported. The MCU keeps four pages open simultaneously. Simultaneously open pages allow for greater performance for sequential access, distributed across multiple internal bus transaction.

3.1.1 Upgrading SDRAM

The CPCI-810 is equipped with 64 Mbytes of SDRAM with ECC mounted on the card. The memory may be expanded by inserting an additional 16 Mbyte to 128 MByte module into the 144 pin SoDIMM socket. Only 144 pin +3.3V SDRAM modules with or without ECC rated for 100 MHz operation should be used on the CPCI-810.

3.1.2 SDRAM Configurations Installation and Removal of Memory Modules

Installation or removal of DIMMs on the CPCI-810 is a simple procedure and requires no special tools. The CPCI-810 should be removed from the host system before its memory configuration is changed, and care must be taken to avoid static discharge while contacting the board. A properly connected grounding strap should be worn while installing or removing memory modules on the CPCI-810 adapter.

Memory modules are removed by rotating the latches located on each end of the SoDIMM socket outward, away from the module. As the latches are moved outward, the module will be ejected from the socket.

To install a memory module, first identify its proper orientation. Each module is keyed with a pair of notches in the card edge of the PC board that correspond to tabs in the socket. With the correct orientation established and the latches in their outward position, begin to slide the module into the socket. The two card edge corners of the module mate with the slots in each latch first. By pressing the module and socket together, the module should snap into the socket. Check that the latches are in their fully closed (inward) position.

3.2 FLASH ROM

The CPCI-810 provides 2 Mbytes of sector-programmable Flash ROM for non-volatile code storage. The Flash ROM is located in local memory space at address FFE0 0000h through FFFF FFFFh. The mapping ensures that, after a reset, the MPC8240 processor can execute the hard reset exception handler located at FFF0 0100h.

3.3 CONSOLE SERIAL PORT

A single console serial port with an RS-232 line interface has been included on the CPCI-810. The port is connected to a RJ-11 style phone jack on the adapter, and can be connected to a host system using the included phone jack to DB-25 cable (Cyclone P/N 530-2006). The pinout of the console connector is as shown in Table 3-1.

Table 3-1. Console Port Connector

Pin	Signal	Description
1		Not Used
2	GND	Ground
3	TXD	Transmit Data
4	RXD	Receive Data
5		Not Used
6		Not Used

The serial port is based on a 16C550 UART clocked at 1.843 MHz. The device may be programmed to use this clock with the internal baud rate counters. The serial port is capable of operating at speeds from 300 to 115200 bps, and can be operated in interrupt-driven or polled mode. The 16C550 register set is shown in Table 3-2. For a detailed description of the registers and device operation refer to the 16C550 databook.

Table 3-2. UART Register Addresses

Address	Read Register	Write Register
FF00 0000H	Receive Holding Register	Transmit Holding Register
FF00 0008H	Unused	Interrupt Enable Register
FF00 0010H	Interrupt Status Register	FIFO Control Register
FF00 0018H	Unused	Line Control Register
FF00 0020H	Unused	Modem Control Register
FF00 0028H	Line Status Register	Unused
FF00 0030H	Modem Status Register	Unused
FF00 0038H	Scratchpad Register	Scratchpad Register

3.4 COUNTER/TIMERS

The MPC8240 processor is equipped with four 31-bit on-chip counter/timers which count at 1/8 the frequency of the SDRAM_CLK signal or 12.5MHz. Users should refer to the Processor User's Manual for the functionality and programming of the counters. The timers can be individually programmed to generate interrupts to the processor when they count down to zero. Two of the timers, timer2 and timer3, can be set up to automatically start periodic DMA operations for DMA channels 0 and 1, respectively, without using the processor interrupt mechanism.

3.5 POWER SUPPLY MONITORING

Two circuits are provided for monitoring the health of power supplies. Additional inputs to the CompactPCI connector define pins for degraded, failed and detected power supplies. The definition for the CompactPCI connector J2 is provided in Appendix B. A failed or degraded power supply, as long as it is detected, will cause an interrupt to the processor. Additionally, the state of the power supply as defined by POWERGOOD, i.e. the power supply is neither degraded or failed, is displayed in a green LED. Figures 3-1 and 3-2 show the register bit definition for the two power supply status registers.

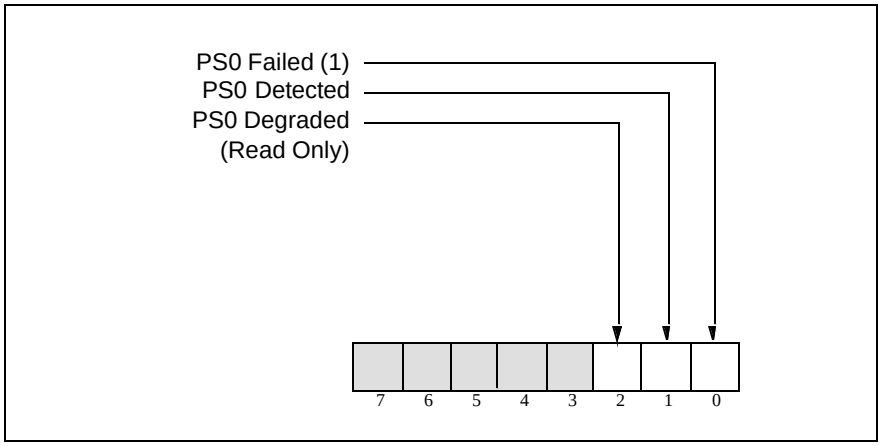

Figure 3-1. POWER SUPPLY #0 STATUS REGISTER (READ ONLY)

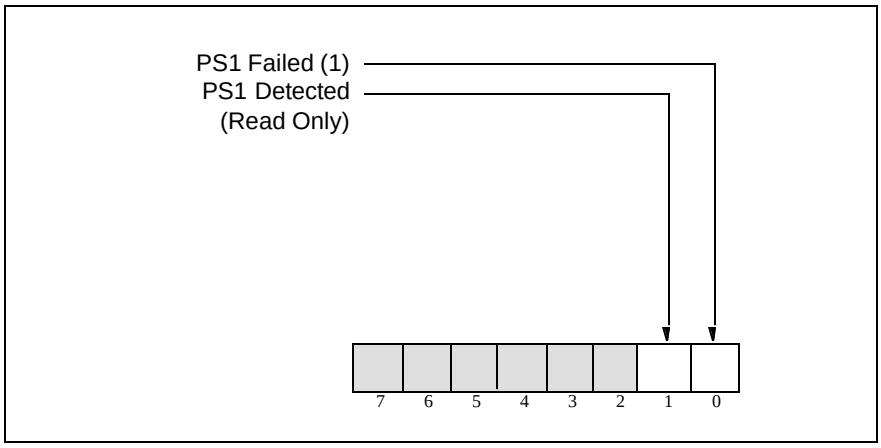

Figure 3-2. POWER SUPPLY #1 STATUS REGISTER (READ ONLY)

3.6 FAN MONITORING

Two circuits are provided for the monitoring of two fan frequencies inputs. As in the case of the power supply monitoring signals, additional inputs to J2 have been defined for the two fan inputs. Refer to Appendix B for their pin locations. The fan monitoring circuits will provide an interrupt to the processor if the frequency of the fan output falls below approximately 8K RPM. Green LEDs are provided for fan interrupt status. If a fan frequency input causes an interrupt, the corresponding LED is turned off. Figure 3-3. shows the register bit definition for the fan status register.

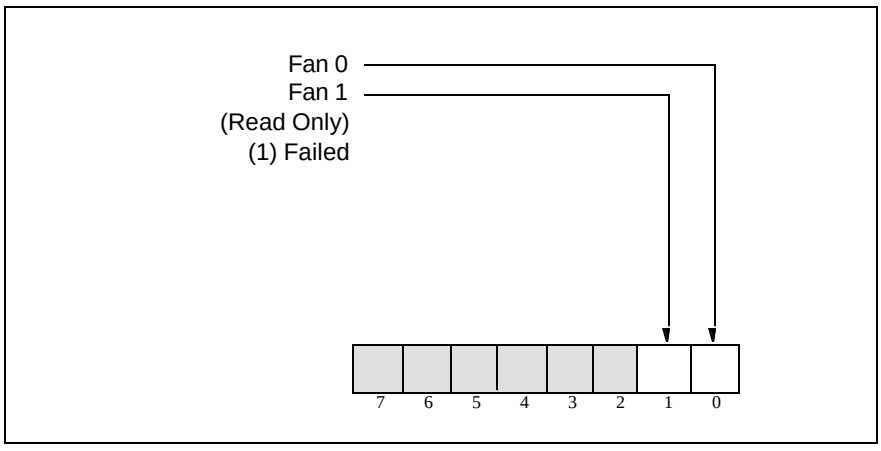

Figure 3-3. FAN STATUS REGISTER (READ ONLY)

3.7 LEDS

The CPCI-810 has twelve green LEDs. The four green LEDs labeled, IOP, ACT, STAT0, and STAT1 are under software control. The LEDs are controlled by a write-only register which is located at address FF20 0000H. The LED Register bitmap is shown in Figure 3-4.

As mentioned previously, two LEDs are provided for the POWERGOOD power supply status and two LEDs are for fan interrupt status.

The remaining four LEDs are associated with the two ethernet circuits and indicated link integrity and network activity for each port. Refer to section 3.7.2 for additional information.

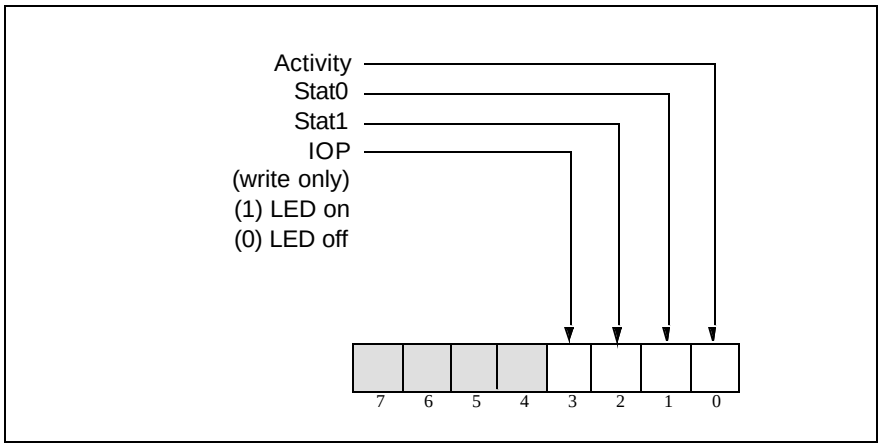

Figure 3-4. LED Register Bitmap, FF20 0000H

3.8 PCI INTERFACE

The CPCI-810 contains a primary 64-bit PCI bus and a secondary 32-bit PCI bus. Both buses are clocked at 33 MHz. The primary PCI bus interfaces the 64-bit CompactPCI bus to the 21154 PCI-to-PCI bridge. The secondary side of the 21154 interfaces a 32-bit PCI bus to the MPC8240 with the PMC module and the two ethernet ports.

3.8.1 Primary PCI Arbitration

The primary PCI bus arbitration is provided by the 21154 PCI-to-PCI bridge. The arbiter arbitrates between the CPCI-810 primary PCI bus requests and the bus requests from the seven peripheral CompactPCI locations. The arbiter supports a programmable 2-level rotating algorithm. Two groups of masters are assigned, a high priority group and a low priority group. The low priority group as a whole represents one entry in the high priority group; that is, if the high priority group consists of n masters, then in at least every n + 1 transactions the highest priority is assigned to the low priority group. Priority rotates evenly among the low priority group. In the case of the CPCI-810, the default grouping, all peripheral slots are assigned to the low priority group and the 21154 is assigned to the high priority group is maintained.

3.8.2 Secondary PCI Arbitration

Secondary bus arbitration logic, between the MPC8240 processor, the 21154 bridge, the two ethernet interfaces and the two PMC devices, is contained within the MPC820. The bus arbitration unit allows fairness as well as a priority mechanism. A two-level round-robin scheme is used in which each device can be programmed within a pool of high- or low-priority arbitration. One member of the low-priority pool is promoted to the high-priority pool. As soon as it is granted the bus it returns to the low-priority pool.

3.8.3 DMA Channels

The MPC8240 processor features two DMA channels. Data movement occurs on the PCI and/or memory bus. Each channel has a 64-byte queue to facilitate the gathering and sending of data. Both the local processor and PCI masters can initiate a DMA transfer. Some of the features of the MPC8240 DMA unit include: misaligned transfer capability, scatter gather DMA chaining and direct DMA modes, and interrupt on completed segment, chain, and error. Figure 3-5 provides a block diagram of the MPC8240 DMA unit.

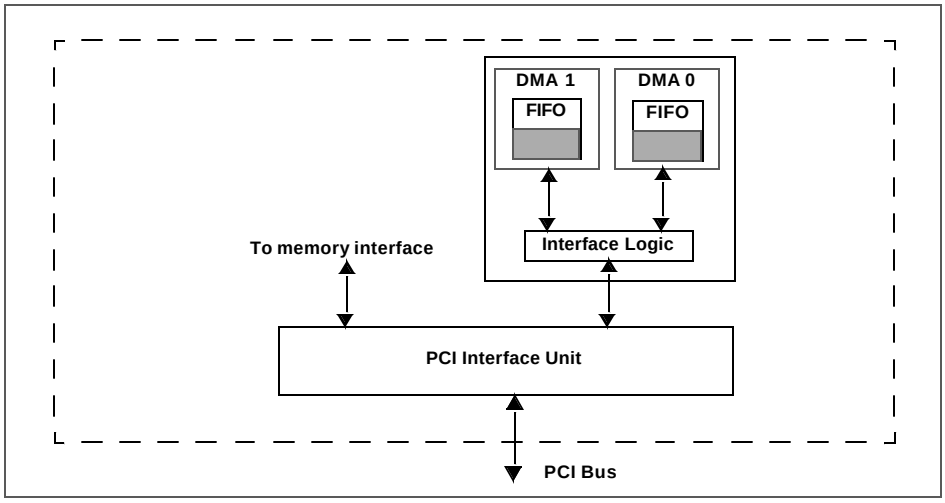

Figure 3-5. MPC8240 Processor DMA Controller

3.8.4 Message Unit

The MPC8240 provides a message unit (MU) to facilitate communications between the host processor and peripheral processors. The MPC8240's MU can operate with generic messages and doorbell registers, and also implements an I_2O compliant interface.

The Intelligent Input Output (I_2O) specification allows architecture-independent I/O subsystems to communicate with an OS through an abstraction layer. The specification is centered around a message-passing scheme. An I_2O-compliant peripheral (IOP) is comprised of memory, processor, and input/output devices. The IOP dedicates a certain space in its local memory to hold inbound (from the remote processor) and outbound (to the remote processor) messages. The space is managed as memory-mapped FIFOs with pointers to this memory maintained through the MPC8240 I_2O registers. Please refer to the MPC8240 User's Manual for I_2O register descriptions, FIFO descriptions and an I_2O message queue example.

3.8.5 JTAG/COP Support

The MPC8240 provides a joint test action group (JTAG) interface. Additionally, the JTAG interface is also used for accessing the common on-chip processor (COP) function of PowerPC processors. The COP function of PowerPC processors allows a remote computer system (typically a PC with dedicated hardware and debugging software) to access and control the internal operations of the processor. The COP interface connects primarily through the JTAG port of the processor. The 16 pin COP header (sample part is Samtec # HTSW-108-07-S-S) is located at J23. The COP header adds many benefits including breakpoints, watchpoints, register and memory examination/modification and other standard debugger features are possible. The COP header definition is shown if Figure 3-6.

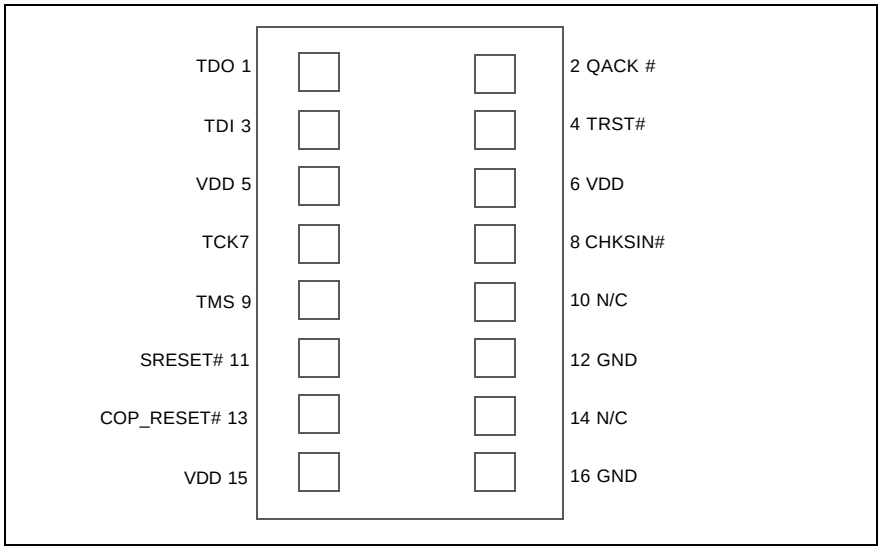

Figure 3-6. COP Header

3.9 GEOGRAPHIC ADDRESSING

CompactPCI backplanes that support 64-bit connector pin assignments are required to provide a unique differentiation based upon which physical slot the board has been inserted. The CPCI-810 makes this definition available to the software. The definition for GA[4:0] is shown in Figure 3-7.

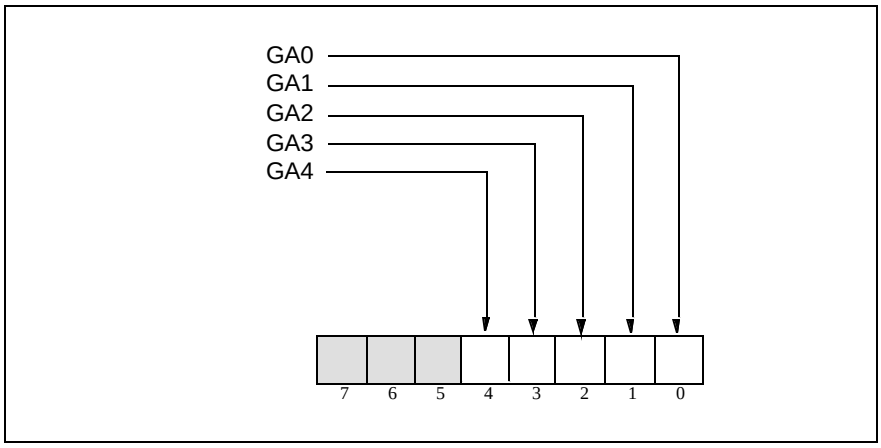

Figure 3-7. Geographic Addressing Register, FF60 0000H.

3.10 ETHERNET PORTS

The two Ethernet ports are based on the Intel 82559 Fast Ethernet PCI Bus LAN Controller with Integrated PHY and support 10BaseT or 100BaseTx signaling. If 100BaseTx signaling is negotiated with its link partner, the port will perform serial transfer at 100Mbps. The 82559 is the core component of the Ethernet interface. It uses a 32-bit PCI interface to communicate with the host and has an integrated PHY (physical layer interface) that connects via an isolation transformer to the network.

3.10.1 82559 Ethernet Controller

The 82559 is an integrated 32-bit PCI bus LAN controller for 10/100Mbps Fast Ethernet networks. It consists of both the Media Access Controller (MAC) and the 10/100Mbps physical layer interface (PHY).

The integrated PHY supports 10BaseT and 100BaseTx operation. The PHY performs digitally controlled receive line equalization and transmit waveform generation for 10Mbps and 100Mbps ethernet networks.

The MAC is a 32-bit PCI bus master with enhanced scatter-gather memory operations. Four DMA channels support high speed data transfers on the PCI interface. A microcode-based engine enables the 82559 to process high level commands and perform multiple operations without CPU intervention. Three kilobyte transmit and receive FIFOs provide storage of multiple transmit and receive frames.

3.10.2 Ethernet Port LEDs

The Ethernet ports on the CPCI-810 each have two LEDs driven by the associated 82559 that provide a visual indication of network status:

The LEDs labeled "LNK1" indicates link integrity for Ethernet port 1 and "LNK2" for Ethernet port 2. The "LNK1" and "LNK2" LEDs will be on continuously if the port is connected to a functional network or Ethernet port.

The LEDs labeled “ACT1” and “ACT2” indicate network activity for the corresponding port. The “ACT1” and “ACT2” LEDs will blink during transmit or receive activity.

3.10.3 Ethernet Port Connector

Each Ethernet port is connect to a shielded RJ45 (modular phone type) connector. The connector conforms to the 10/100BaseTx specification. The connector exits the front panel of the CPCI-810. The front panel has the ports (and LEDs) labeled. The pinout of each port is shown in Table 3-3. Pin one is to the extreme left as one looks into the connector opening with the tab notch down.

Table 3-3. 100BaseTx Connector

Pin	Signal	Description
1	TX+	Output
2	TX-	Output
3	RX+	Input
4		Not Used
5		Not Used
6	RX-	Input
7		Not Used
8		Not Used

3.11 I²C BUS

The CPCI-810 has three components attached to the Inter-Integrated Circuit (I2C) bus interface of the MPC8240 processor: the SoDIMM SDRAM EEPROM and two temperature sensors. The I²C addresses of the devices are shown in Table 3-4.

Table 3-4. I²C Device Addresses

Designator	Device	Function	Address
J24	SoDIMM EEPROM	Memory Configuration	1010001
U23	LM75	Temperature Sensor	1001000
U1	LM75	Temperature Sensor	1001001

3.11.1 SDRAM EEPROM

The EEPROM located on the SoDIMM SDRAM module contains identification and configuration information. Initialization code should read this information on power-up and properly configure the memory controller.

3.11.2 Temperature Sensors

The LM75 temperature sensors have overtemperature trip points that will trigger an interrupt when crossed. The sensors have been placed on the board at U1 and U23, and share serial interrupt #10. The sensors can be read for a temperature reading at any time, and reading after and interrupt clears the interrupt. The sensor will not interrupt again until the temperature has dropped below the hysteresis. Consult the LM75 data sheet for more details on programming the temperature sensors.

3.11.3 Watchdog Timer

The CPCI-810 contains a watchdog timer circuit. The time-out period for a watchdog failure is programmable for 250ms, 500ms or 1s. The watchdog must be enabled to function and can be disabled at any time. A watchdog keep alive must take place before the time-out period is reached. Failure to do so will result in a reset to the processor and both the Primary and Secondary PCI buses.

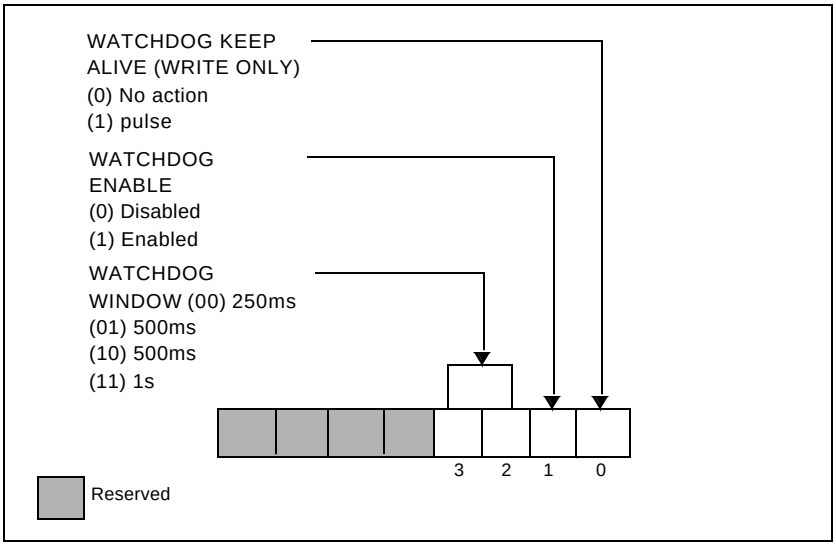

Figure 3-8. Watchdog Register, FF10 0000h

3.12 BOARD ID REGISTER

The Board ID Register is a read-only register that can be used to differentiate between the CPCI-810 and other Cyclone Microsystems MPC8240-based CompactPCI cards. It is located at address FF70 0000h on all such cards, with each card returning a unique ID value. Figure 3-9 shows the board ID for the CPCI-810.

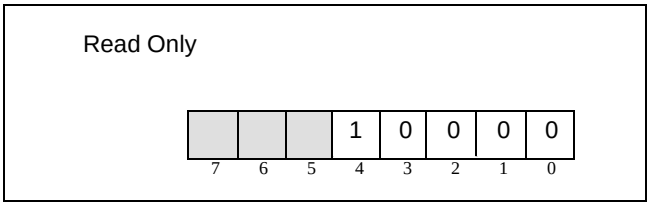

Figure 3-9. Board Identification Registers, FF70 0000h

CHAPTER 4
pSOS SOFTWARE DEVELOPMENT

4.1 INTRODUCTION

This chapter is dedicated to aiding pSOS application development using the Cyclone CPCI-810 pSOS BSP. It contains information specific to the Cyclone BSP, and is intended to be used in conjunction with the pSOS documentation provided by ISI / Wind River Systems. Note that there are many items within the BSP that a user may want to self configure, so users should be readily able to modify and rebuild the BSP when necessary.

Once an application has been built and linked with the CPCI-810 BSP, the image can be downloaded to DRAM via Ethernet using the Cyclone TFTP Bootloader (see documentation on this procedure), or downloaded to Flash ROM or DRAM using a JTAG-emulator such as Wind River's Visionprobe tool (see section 3.10).

This chapter is divided into the following sections:

- Embedded Utilities Memory Block
- Endian Considerations
- PCI Configuration
- EPIC Interrupt Programming
- LM75 Temperature Sensors

4.2 EMBEDDED UTILITIES MEMORY BLOCK

The Embedded Utilities Memory Block (EUMB) is a relocatable memory block that contains the registers for several of the MPC8240's embedded features, including the Messaging Unit, DMA Controller, Address Translation Unit (ATU), I^2C Controller, and Embedded Programmable Interrupt Controller (EPIC). Figure 4-1 shows the EUMB memory offsets for each of these embedded devices.

The base of the EUMB is software programmable by setting the EUMB Base Address Register (EUMBBAR) in the MPC8240's PCI Configuration Space (offset 0x78). pSOS initialization sets this value at startup. Users should never modify this value, and should read this value when necessary using a local PCI configuration read cycle (see section 4.4).

For further information on MPC8240 address maps and the EUMB, consult chapter 4 of the MPC8240 User's Manual.

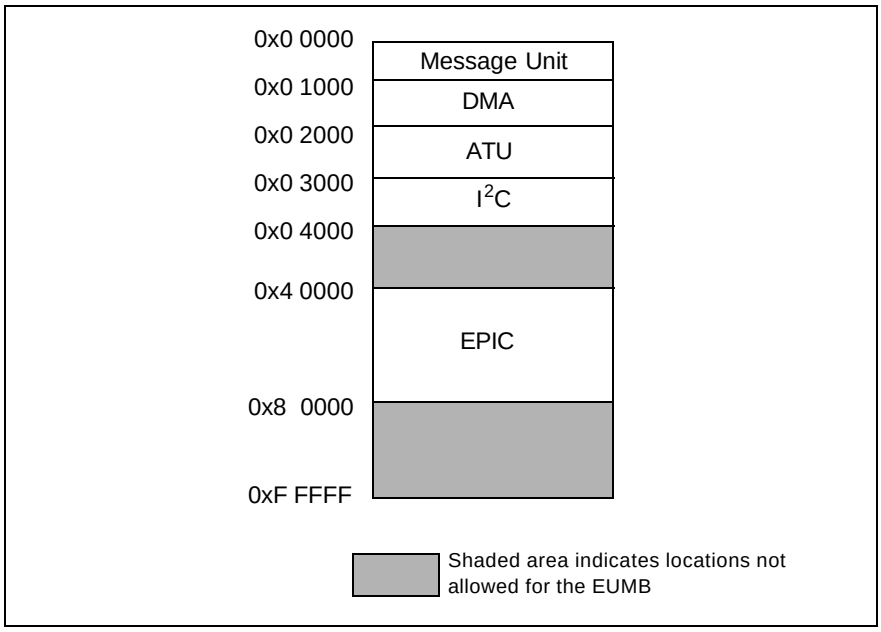

Figure 4-1. Embedded Utilities Memory Block

4.3 ENDIAN CONSIDERATIONS

The MPC8240 on the CPCI-810 stores data in local memory in a big endian manner (most significant byte in the lowest memory address). However, the PCI bus is a little endian bus (least significant byte in the lowest byte lane), including access to all registers in the EUMB. Care must be taken to byte swap data transferred from memory to the PCI bus or EUMB registers.

pSOS provides the following functions to read and write data to/from the PCI bus and EUMB. They perform all required byte swapping. The following function declarations are from pci/pcihdr.h:

```
void  PciWrite32(ULONG addr, ULONG value);
void  PciWrite16(ULONG addr, ULONG value);
void  PciWrite8 (ULONG addr, ULONG value);
ULONG PciRead32(ULONG addr);
ULONG PciRead16(ULONG addr);
ULONG PciRead8(ULONG addr);
```

4.4 PCI CONFIGURATION

The CPCI-810 is a CompactPCI system controller, responsible for the configuration of all PCI devices in the system. This includes PCI initialization of the two 82559 Ethernet devices and PMC module devices resident on PCI bus 0, and all other CPCI devices resident behins the 21154 PCI-to-PCI bridge on the CompactPCI bus (bus 1).

4.4.1 pSOS PCI Device Driver Interface

Once initialization is complete, a pSOS PCI device list is maintained containing pertinent information on PCI devices found and configured in the system. Thus, device driver developers can use any of the functions described in chapter 6 of the pRISM+ Advance Topics Guide to find, claim, or access any local PCI or CompactPCI devices. In addition, the PCI header file (\pci\pcihdr.h) provides useful protypes and important macros for dealing with PCI devices.

One important definition in this header file is the PCI_LOC structure, which is used to define the PCI location of a particular device. This structure is defined below in Figure 4-2. At PCI configuration, a list of PCI_LOC structures is created by the pSOS Auto-Configuration. This list is defined in pcicfg.c by:

PCI_LOC pci_dev_list[PCI_DEV_LIST_SIZE];

This list is important, as it contains a PCI_LOC structure for all of the PCI devices on the local PCI bus. Many pSOS PCI functions, such as those to find a particular device in the list, require a pointer to this list and the list length as arguments. Others, such as those that access a particular PCI device, require the PCI_LOC element from the list, which indicates which device the transaction is to occur on.

```
typedef struct pciloc {
    short          bus;          /* bus number */
    char           device;       /* device number */
    char           function;     /* function number */
    char           hostBridge;   /* Host/PCI bridge number*/
    unsigned char  cfgFlags;     /* Configuration flags */
    unsigned char  claimed;      /* Claimed Status */
    unsigned char  cfgStat;      /* Configuration Status */
    unsigned long  dev_vend;     /* Devices and Vendor ID */
    unsigned long  keyValue;     /* pSOS Key Value */
    unsigned long  intrVec;      /* Interrupt vector number */
} PCI_LOC;
```

Figure 4-2. PCI_LOC Structure Definition

For further information on the PCI configuration, consult the Intel 21154 PCI-to-PCI Bridge Data Sheet, chapter 8 of the MPC8240 User's Manual, and chapter 6 of the pRISM+ Advance Topics Guide.

4.5 EPIC INTERRUPT PROGRAMMING

The Embedded Programmable Interrupt Controller (EPIC) is the general-purpose interrupt controller internal to the MPC8240. EPIC control and status registers are located in the EUMB.

CPCI-810 hardware is configured to provide fifteen dedicated external hardware interrupts, which are time-division multiplexed onto one serial input on the MPC8240. The EPIC controller also provides four internal timers that can be interrupt sources, and handles internal interrupts from the I^2C, I_2O, and 2 DMA channels.

Table 4-1 shows the EPIC hardware interrupts, and the assigned default interrupt priorities. These interrupt priorities can be modified by the application programmer by changing the vector priority values in the table called priTable_810 in the BSP file epic.c, and recompiling the BSP. Priority values are in the range of 15 to 0, with 15 being the highest priority (0 inhibiting the interrupt altogether).

For further information on the MPC8240 EPIC, consult chapter 4 of the MPC8240 User's Manual.

Table 4-1. CPCI-810 Interrupt Vectors.

INT NUMBER	INT VECTOR	SOURCE	PRIORITY LEVEL	DESCRIPTION
0	0x10	PINTA	8	INTA on PCI Bus
1	0x11	PINTB	8	INTB on PCI Bus
2	0x12	PINTC	8	INTC on PCI Bus
3	0x13	PINTD	8	INTD on PCI Bus
4	0x14	PMCINTA	6	INTA from PMC Module
5	0x15	PMCINTB	6	INTB from PMC Module
6	0x16	PMCINTC	6	INTC from PMC Module
7	0x17	PMCINTD	6	INTD from PMC Module
8	0x18	ENETINT0	12	Interrupt from 82559 Unit 0
9	0x19	ENETINT1	12	Interrupt from 82559 Unit 1
10	0x1a	TEMP_INT	4	LM75 Temperature Interrupt
11	0x1b	ENUM_INT	10	Hot Swap ENUM Interrupt
12	0x1c	UART_INT	10	UART Interrupt
13	0x1d	FANINT	14	Loss of Fan Interrupt from Chassis
14	0x1e	POWERINT	14	Loss of Power Interrupt from Chassis
15	0x1f	UNUSED	0	UNUSED
16	0x20	TIMER0	12	EPIC Internal Tick Timer 0
17	0x21	TIMER1	12	EPIC Internal Tick Timer 1
18	0x22	TIMER2	2	EPIC Internal Tick Timer 2
19	0x23	TIMER3	2	EPIC Internal Tick Timer 3
20	0x24	I^2C	4	Interrupt from I^2C Controller
21	0x25	DMA0	7	Interrupt from DMA Channel 0
22	0x26	DMA1	7	Interrupt from DMA Channel 1
23	0x27	MSG_UNIT	7	Interrupt from Messaging Unit

4.5.1 Connecting and Disconnecting Interrupt Handlers in pSOS

PSOS utilities for connecting and disconnecting interrupt handlers to these interrupts can be found in the BSP file isr.c. The function PssSetIntHandler is used to connect and enable an interrupt handler:

```
 long PssSetIntHandler(
   ULONG Level,            /* Interrupt vector number */
   void *handler,          /* Pointer to handler function */
   void *arg,              /* Optional argument to handler */
   ULONG type              /* Optional wrapper type */
)
```

The function PssUnSetIntHandler is used to disconnect an interrupt handler:

```
long PssUnSetIntHandler(
   ULONG Level,        /* Interrupt vector number */
   void *handler,      /* Pointer to handler function */
   void *arg           /* Optional argument to handler */
)
```

4.6 LM75 TEMPERATURE SENSORS

The two on-board LM75 devices can be used to detect possible temperature problems in the system, such as overheating. The BSP file lm75.c contains a collection of routines that simplify the use of the devices. Included are functions to read and write to registers on the LM75, including the temperature, trip, and hysteresis registers.

Before use, the LM75 should be placed in interrupt mode. When in this mode an LM75 will interrupt if the temperature goes above the value in the trip register, and will interrupt again when the temperature falls back below the value in the hysteresis register.

Also included in the lm75.c file is the function lm75_test, which is a simple diagnostic that uses useful LM75 routines to test the devices. Developers can use this as an example for writing their own utilities to operate the LM75, and can also call this test in their application to ensure that the devices are operating correctly.

For further information consult the National Semiconductor LM75 Data Sheet.

APPENDIX A
PMC MODULE INTERFACE

A.1 INTRODUCTION

The PMC Module Interface allows PCI devices to be connected to the secondary PCI interface of the CPCI-810 Adapter. The IEEE STD P1386.1, PCI Mezzanine Card (PMC), provides for one set of clocking and arbitration signals per PMC Module. Cyclone Microsystems has expanded this to two sets for the PMC Module on the CPCI-810. Otherwise, with a few exceptions, the standard signals defined for 32-bit CPCI connectors are used for the PMC Modules. The exceptions are noted in section A.3. The timing for devices on PMC Modules is the same as the timing for any other PCI device; see the *PCI Local Bus Specification* revision 2.1 for details.

A number of PMC Modules are available from Cyclone Microsystems. This section is intended for users interested in developing their own modules.

A.2 PHYSICAL ATTRIBUTES

Please refer to IEEE P1386/Draft 2.0 for the physical dimensions of PMC modules.

A.3 PMC MODULE SIGNAL DEFINITIONS

PMC Modules use the signals defined in the IEEE STD P1386.1. The following four signals are added to this definition to handle the expansion from one to three devices per PMC module:

- GNT1#
- REQ1#
- CLK1
- IDSEL1

Please note that the added signals used the PMC-RSVD signals as defined in IEEE STD P1386.1. The PCI-RSVD remain untouched.

Also, note that GNT1# follow the description for GNT#, REQ1# follow the description for REQ#, CLK1 follow the description for CLK, and IDSEL1 follow the description for IDSEL. When the appropriate signals are connected to PCI devices on a PMC Module, each device has the full complement of PCI signals defined in the specification.

Table A-1 shows the IDSELx# routing and table A-2 shows the interrupt routing on the CPCI-810.

Table A-1. PMC Clock & Arbitration Assignment

IDSEL	ADDR	IDSEL#	CLOCK	ARBITRATION
IDSEL#	AD16	J12.25	CLKA	REQ0#,GNT0#
IDSEL1#	AD17	J12.34	CLKB	REQ1#,GNT1#

Table A-2. PMC Interrupt Assignment

DEVICE INTx#	1ST DEVICE	2ND DEVICE
INTA#	INTA#	INTD#
INTB#	INTB#	INTA#
INTC#	INTC#	INTB#
INTD#	INTD#	INTC#

A.4 PMC MODULE CONNECTOR

PMC Modules use three board-to-board connectors (plug) with 64 pins each. The receptacles (AMP P/N 120521-2) is located on the host platform and attaches to the plugs (AMP P/N 120527-2). This connector combination allows for a 10 mm board-to-board spacing. See IEEE P1386/Draft 2.0 for dimensions and component clearance details.

Table A-3. P21 PMC Module Connector Pinout

Pin	Signal	Pin	Signal
1	TCK	2	-12V
3	GND	4	INTA#
5	INTB#	6	INTC#
7	BUSMODE1#	8	+5V
9	INTD#	10	PCI-RSVD
11	GND	12	PCI-RSVD
13	CLK	14	GND
15	GND	16	GNT#
17	REQ#	18	+5V
19	V(I/O)	20	AD31
21	AD28	22	AD27
23	AD25	24	GND
25	GND	26	C/BE3#
27	AD22	28	AD21
29	AD19	30	+5V
31	V(I/O)	32	AD17
33	FRAME#	34	GND
35	GND	36	IRDY#
37	DEVSEL#	38	+5V
39	GND	40	LOCK#
41	SDONE#	42	SBO#
43	PAR	44	GND
45	V(I/O)	46	AD15

47	AD12	48	AD11
49	AD09	50	+5V
51	GND	52	C/BE0#
53	AD06	54	AD05
55	AD04	56	GND
57	V(I/O)	58	AD03
59	AD02	60	AD01
61	AD00	62	+5V
63	GND	64	REQ64#

Table A-4. P22 PMC Module Connector Pinout

Pin	Signal	Pin	Signal
1	+12V	2	TRST#
3	TMS	4	TDO
5	TDI	6	GND
7	GND	8	PCI-RSVD
9	PCI-RSVD	10	PCI-RSVD
11	BUSMODE2#	12	+3.3V
13	RST#	14	BUSMODE3#
15	+3.3V	16	BUSMODE4#
17	PCI-RSVD	18	GND
19	AD30	20	AD29
21	GND	22	AD26
23	AD24	24	+3.3V
25	IDSEL	26	AD23
27	+3.3V	28	AD20
29	AD18	30	GND
31	AD16	32	C/BE2#
33	GND	34	PMC+IDSEL1
35	TRDY#	36	+3.3V
37	GND	38	STOP#
39	PERR#	40	GND
41	+3.3V	42	SERR#
43	C/BE1#	44	GND
45	AD14	46	AD13
47	GND	48	AD10
49	AD08	50	+3.3V
51	AD07	52	PMC+REQ1

53	+3.3V	54	PMC+CLK1
55	PMC+GNT1#	56	GND
57	PMC-RSVD	58	PMC-RSVD
59	GND	60	PMC-RSVD
61	ACK64#	62	+3.3V
63	GND	64	PMC-RSVD

APPENDIX B
CPCI J2 DEFINITION

B.1 INTRODUCTION

The CPCI-810 utilizes some of the reserved pins in J2 for FAN and Power Supply status information. Differences from the CPCI specification are shown in table B-1.

Table B-1. CPCI-810 J2 Definition

21	FAN1	FAN0	DET1#
20	FAL1#	GND	DET0#
Pin	C	D	E