RadiSys ARTIC960 Co-Processor Platforms

Hardware Technical Reference

RadiSys Corporation
5445 NE Dawson Creek Drive
Hillsboro, OR   97124
(503) 615-1100
FAX: (503) 615-1150
www.radisys.com
September 1999

Before using this information and the product it supports, be sure to read all the information in Appendix D, Notices .

About this Guide

This publication contains technical reference information for the RadiSys ARTIC960 PCI Co-Processor Platform and RadiSys ARTIC960 PCI Co-Processor Platform.

The objectives of this publication are to:

- Describe the various units of the co-processor platform
- Explain the interaction of the units on the co-processor platform
- Explain the technical details of the options and interfaces of the co-processor platform
- Provide descriptions and data related to the co-processor platform configuration, component layout, circuitry, functions, hardware interfaces, and programming considerations.

Detailed programming information for the co-processor platform can be found in the *RadiSys ARTIC960 Co-Processor Platforms Programmer's Guide and Reference*.

Guide contents

The following lists the contents of this Guide.

Chapter		Description
1	Introduction to the RadiSys ARTIC960 Co-Processor Platform	Provides a brief introduction to the co-processor platform and serves as a quick reference to the major hardware components, subsystems, interfaces, and specifications.
2	Co-Processor Platform Components	Covers the co-processor platform hardware in detail. It begins with a high-level discussion and progresses downward to specific registers.
3	Memory Protection Subsystem	Provides an in-depth discussion of the memory protection hardware that is external to the 80960 processor.
4	Error Checking and Correction (ECC) Subsystem	Describes the implementation of the error detection and correction scheme, and provides details of the registers associated with this function.
5	Timer Subsystem	Discusses the interval timers.
6	Interrupt Controller Subsystem	Describes the interaction of the controller with the Application-Specific Integrated Circuits, the AIB, and the 80960 processor.

Chapter		Description
7	CFE Local Bus Subsystem and AIB Interface Description	Provides details of Local Bus operation, specifically addressing the request sequence, bus arbitration, bursting, timeout, parity, and interrupts. This chapter also describes CFE Local Bus and ROM Bus signals, and lists the AIB connector pin assignments.
8	Micro Channel Interface Subsystem	Describes the major functions of the Micro Channel Interface Chip, provides details of the registers associated with the Micro Channel, and discusses overall Bus Master operation.
9	PCI Interface Subsystem	Describes the major functions of the PCI System Bus Interface Chip, provides details of the registers associated with the PCI, and discusses overall Bus Master operation.
10	Miscellaneous Registers	Covers the registers associated with the implementation of miscellaneous co-processor platform functions, such as ROM write enable, memory presence detect, and so forth.
11	Debug Features	Provides details of the tool that is built into the co-processor platform for use during the debug phase of code development.
12	ROM Support	Describes the power-on self-test (POST) process and error reporting for the co-processor platform hardware, as well as initialization, setup, and interface structures for downloading and executing the AIB's ROM POST and supervisor task routines.
13	4-Port Multi-Interface Application Interface board	Describes the architecture for this AIB and contains specifications, interface design criteria, cabling information, and software implementation details.

Appendices

The appendices provide additional information about ARTIC960.

Appendix		Description
A	Acronym Glossary	Contains a listing of the acronyms used in this publication.
B	Co-Processor Platform Registers	Contains a listing of these registers, sorted by abbreviation and by address.
C	4-Port Multi-Interface AIB Registers	Contains a listing of these registers, sorted by abbreviation and by address.

Who Should Read this Book

The information in this publication is both introductory and for reference use. It is intended for hardware and software designers, programmers, engineers, and anyone with a knowledge of electronics and/or programming who wishes to understand the use and the operation of the co-processor platform.

It is assumed that the reader is familiar with the co-processor platform, as well as the computer system in which it is installed, the applications in use, and programming.

Therefore, terminology is not explained herein, except for terms that may be specially implemented.

Notational conventions

This manual uses the following conventions:

- Unless noted otherwise, *RadiSys ARTIC960 Co-Processor Platform* is used to denote both the ARTIC960 and the ARTIC960 PCI adapters. Throughout this publication, the ARTIC960 Co-Processor Platform is also referred to as the *co-processor platform* or the *base card*.
- All counts in this document are assumed to start at zero.
- All bit numbering conforms to the industry standard of the most significant bit having the highest bit number.
- All numeric parameters and command line options are assumed to be decimal values unless otherwise noted.
- To pass a hexadecimal value for any numeric parameter, the parameter should be prefixed by **0x** or **0X**. The numeric parameters **16**, **0x10**, and **0X10** are all equivalent.
- All representations of bytes, words and double words are in the Intel format.
- Utilities all accept the **?** switch as a request for help with command syntax.
- `Screen text and syntax strings appear in this font.`
- All numbers are decimal unless otherwise stated.
- Bit 0 is the low-order bit. If a bit is set to 1, the associated description is true unless otherwise stated.

Notes indicate important information about the product.

Cautions indicate situations that may result in damage to data or the hardware.

Tips indicate alternate techniques or procedures that you can use to save time or better understand the product.

ESD cautions indicate situations that may cause damage to hardware via electro-static discharge.

The globe indicates a World Wide Web address.

Warnings indicate situations that may result in physical harm to you or the hardware.

Where to get more information

You can find out more about ARTIC960 from these sources:

- **World Wide Web**: RadiSys maintains an active site on the World Wide Web. The site contains current information about the company and locations of sales offices, new and existing products, contacts for sales, service, and technical support information. You can also send e-mail to RadiSys using the web site.

  When sending e-mail for technical support, please include information about both the hardware and software, plus a detailed description of the problem, including how to reproduce it.

To access the RadiSys web site, enter this URL in your web browser:
`http://www.radisys.com`

Requests for sales, service, and technical support information receive prompt response.

- **Other**: If you purchased your RadiSys product from a third-party vendor, you can contact that vendor for service and support.

Reference Publications

You may need to use one or more of the following publications for reference:

- Operating and Installation documentation provided with your computer system.
- IBM Operating System/2 (OS/2) documentation
- IBM Advanced Interactive Executive (AIX) documentation
- *RadiSys ARTIC960 Co-Processor Platform Guide to Operations*
- *RadiSys ARTIC960 PCI Co-Processor Platform Guide to Operations*.

  These manuals contain a description of the co-processor platform, instructions for physically installing the adapter, procedures for installing and setting up the operating system, configuration information, parts listings, and a Symptom-to-FRU Index Supplement.
- *Application Interface Board Supplement(s) to RadiSys ARTIC960 Co-Processor Platforms Guide to Operations*. Each Supplement contains a description of an IBM-developed AIB, instructions for installing the AIB Cable, interface connector information, and interface cable details.
- *Application Interface Board Hardware Maintenance Libraries*. Each AIB Library contains one or more wrap plugs that are used during diagnostic testing to wrap selected interface lines of an IBM-developed AIB and the associated Interface Cable. Also included are removal and replacement procedures for field-replaceable units (FRUs), parts listings, and a Symptom-to-FRU Index Supplement. Most AIB Libraries include a diagnostic diskette that is used to isolate a symptom to a failing FRU.

RadiSys ARTIC960 Developer’s Kit—Content

The Developer’s Kit is a set of publications and programs designed to help ARTIC960 AIB software developers develop for the ARTIC960 platform. The following items make up the Developer’s Kit:

- *RadiSys ARTIC960 Co-Processor Platforms Hardware Technical Reference* presents technical details of the adapter’s system, options, and hardware interfaces. It provides descriptions and data related to the card configuration, functions, hardware interfaces, and programming considerations.
- *RadiSys ARTIC960 Co-Processor Platforms Programmer’s Guide and Reference* provides an overview of both the adapter kernel support and the operating system support. It also describes the associated processes and utilities, as well as each of the services provided by the system unit support and the adapter kernel support.

- *RadiSys ARTIC960 Co-Processor Platforms Application Interface Board Developer's Guide* provides the hardware and the software developer with AIB design requirements, and a collection of productivity tools to aid in the development of an AIB.
- A set of operating system packages, each containing sample programs and utilities to support the development of system unit and adapter applications. These packages are to be used with the *RadiSys ARTIC960 Co-Processor Platforms Programmer's Guide* and the *RadiSys ARTIC960 Co-Processor Platform Application Interface Board Developer's Guide*.
- You can obtain these publications in PostScript format from the World Wide Web (WWW) at:

`http://ww.radisys.com/products/artic/`

If you do not have access to the WWW, or you cannot print PostScript files, you can obtain these publications from the no-fee Developer's Assistant Program (DAP), which also provides technical question-and-answer (Q&A) services supported by telephone and E-mail, as explained under Developer's Assistance Program.

Developer's Assistance Program

In addition to the *Developer's Kit*, further programming and hardware development assistance is provided by the RadiSys ARTIC960 Developer's Assistance Program (DAP). The DAP will provide, via phone and electronic communications, on-going technical support—such as sample programs, debug assistance, and access to the latest microcode upgrades.

You can get more information or activate your ***free*** RadiSys ARTIC960 DAP membership by contacting us.

By telephone, call **(561) 454-3200**.

By E-mail, send to **artic@radisys.com**.

Contents

Chapter 13: 4-Port Multi-Interface Application Interface board

Figures

Tables

Introduction to the RadiSys ARTIC960 Co-Processor Platform

1

This chapter provides a brief description of the RadiSys ARTIC960 Co-Processor Platform, and serves as a quick reference guide to the major hardware components, functional subsystems, physical interfaces, and specifications.

Product Description

The RadiSys ARTIC960 Co-Processor Platforms are 1-slot, add-in, I/O adapters for IBM 32-bit Micro Channel or PCI systems. The main engine for the co-processor is the Intel 80960C-Series microprocessor, which serves to off-load the system unit microprocessor of chores typically associated with I/O tasks. On-board storage includes from 1 to 16 megabytes (MB) of 80960 *packet* dynamic random-access memory (DRAM) and 1MB of *instruction* DRAM. The co-processor is capable of supporting up to 32MB of packet DRAM when it becomes available. A special order is required to configure the co-processor with 0 or 4MB of instruction DRAM.

The co-processor is shipped with 1MB of instruction memory and without packet memory. At least 1MB of packet memory must be installed for the platform to function. (Presently, plug-in packet memory modules are available in increments of 1, 4, 8, and 16MB.) If the instruction DRAM is not installed, the packet DRAM acts as both a data buffer and an instruction storage area for the 80960.

Also included are 128 kilobytes (KB) of read-only storage for the platform's power-on self-test (POST) and bootstrap loader. The RadiSys ARTIC960 PCI Co-Processor Platform also uses a 128-byte serial EPROM for the platform's configuration data. (A functional block diagram of the co-processor is shown in Figure 2-1.)

External support functions for the 80960 include five hardware timers, a high-performance Micro Channel interface, and a separate communications I/O bus (CFE Local Bus). This bus is consistent with the definition of the Common Front End (CFE) bus architecture.

The co-processor complies with the SCB (Subsystem Control Block) Move Mode architecture.

Communication-specific support is accomplished by the attachment of an application interface board (AIB). This is the same concept that is used on the IBM Realtime Interface Co-Processor Portmaster Adapter/A product; however, Portmaster Adapter Interface Boards (IBs) and the co-processor AIBs are not plug compatible. The AIB must contain all communication-specific hardware, such as the protocol chip and/or electrical interface drivers. In general, the AIB handles a mix of layer 1 and 2 functions, with a mix of layer 2 and 3 functions being handled by the 80960 base card microprocessor.

Processing Power

The co-processor provides processing power through the on-board 80960C-Series processor. This enables the host processor to off load most processing chores.

Several co-processors, each containing a different type of AIB, can be installed in a computer to support attachment to a variety of devices such as:

- Various serial communications protocols
- Local area network (LAN) attachments
- Direct access storage devices (DASD)
- Small computer system interface (SCSI)
- Specialized application technology
- Realtime multitasking devices.

Software

The software provided with the co-processor is segmented into two parts. The part that runs on the platform itself is called the *kernel*, and the portion that runs on the host computer is called *system unit support*.

The kernel is actually a collection of executables that provide the following capabilities:

- Realtime multi-tasking kernel
- System unit-to-adapter process communications
- Adapter-to-adapter process communications.

The system unit support is a collection of operating system executables that provide:

- Adapter status and configuration information
- System unit-to-adapter process communications
- Application loader
- Adapter dump facility
- Debug facilities.

Compatibility

The RadiSys ARTIC960 Co-Processor Platform is a follow-on product to the IBM Realtime Interface Co-Processor Portmaster Adapter/A, and both operate in a Micro Channel environment. The RadiSys ARTIC960 PCI Co-Processor Platform is a follow-on product to the RadiSys ARTIC960 Co-Processor Platform, but it operates in a PCI environment. Because the micro processor changed in the first case and the system bus changed in the latter case, microcode written for either of the follow-on products may not be 100% compatible.

Some level of compatibility may be possible for those Postmaster applications written in "C" language. Refer to the *RadiSys ARTIC960 Co-Processor Platforms Programmer's Guide and Reference* for more information on programming features.

The RadiSys ARTIC960 Co-Processor Platform is intended to be compatible with 32-bit Micro Channel-based computers systems, including: IBM Personal System/2 Computers (excluding the Model 8570), IBM Industrial Computers (excluding the Model 7561), and RISC System/6000 Computers. The platform is designed to fit into only 32-bit card slots.

The RadiSys ARTIC960 PCI Co-Processor Platform is intended to be compatible with any full-length, 5V PCI slot in PCI-compliant systems; however, the PCI co-processor adapter may not configure in systems with COMPAQ or Award BIOS, and may not configure in IBM systems with previous-level BIOS.

Performance

Memory Performance

For applications requiring up to 5 megabytes-per-second (MB/s) of I/O throughput, a single memory bus architecture is chosen to reduce cost. An application requiring the full 5 MB/s communication bandwidth should consume roughly 30% of the 80960 Local Bus bandwidth. This would reduce the effective million instructions-per-second (MIP) rate of the 80960 from about 15 MIPs to 10 MIPs. This option, which must be special-ordered, requires that the instruction DRAM be removed from the co-processor.

To support data rates from 5MB/s up to 20MB/s and still have sufficient processing bandwidth available to support these rates, the co-processor is configured in a dual memory bus architecture. In this case, the instruction DRAM allows the 80960 to execute in parallel with I/O activity on the platform’s Local Bus. The tradeoff of cost versus performance can be made by the specific application requirement. The application would determine how many MIPs are necessary to support a particular data rate.

On the 80960 bus, the Memory Controller Chip supports 3-1-1-1-0 cycles (36MB-per-second) to either memory. Up to four word bursts (quad word aligned) are supported. The Memory Controller Chip also supports 3-1-1-1-1 read cycles (2-1-1-1-1 write cycles) from the Local Bus to packet memory (4-1-1-1-1 to instruction memory). However, the Local Bus may burst more than four words. This translates to the following rates:

- 33 MB/s for 4-word bursts
- 40 MB/s for 8-word bursts
- 44 MB/s for 16-word bursts.

Micro Channel-to-Local-Bus Performance

The Micro Channel Interface Chip interfaces a high-speed, 80MB/s Micro Channel interface to a high-speed Local Bus. This Local Bus has a **maximum** theoretical bandwidth of 100MB/s. Data rates achieved by the Micro Channel Interface Chip on the Local Bus are dependent on the size of the transfer and the speed of the Local Bus slave. Assuming a 64-byte burst to a zero-wait-state slave, the data rate is 84MB/s. For memory or slaves requiring wait states, the data rate may be much lower.

The data rate sustained on the Micro Channel is also dependent on the speed of the Local Bus slave. With the intermediate data buffering provided internal to the Micro Channel Interface Chip, 80MB/s can be sustained for an absolute minimum of 128 bytes.

Table 1-1 shows the maximum throughput between the Local Bus and the Micro Channel for Bus Master accesses, assuming 40 or 80MB/s streaming data on the Micro Channel, and 50MB/s (1 wait state) or 100MB/s (0 wait states) Local Bus accesses.

Table 1-1. Micro Channel Interface Chip Bus Master Performance (in MB/s)

Micro Channel Speed	Local Bus Speed	Throughput
40	50	32
80	50	41
40	100	40
80	100	55

PCI-to-Local-Bus Performance

The PCI System Bus Interface Chip interfaces a synchronous, 132-MB/sec PCI bus interface to a high-speed local bus. The local bus has a maximum theoretical bandwidth of 100 MB/sec. Data rates achieved by the PCI System Bus Interface Chip on the local bus are dependant on the size of the transfer and the speed of the local bus slave. For memory or slaves requiring wait states, the data rate may be much lower.

The data rate sustained on the PCI bus also is dependant on the speed of the local bus slave. With the intermediate data buffering provided internal to the PCI System Bus Interface Chip, 132MB/sec can be sustained for an absolute minimum of 128 bytes.

Table 1-2 shows the maximum measured throughput between the CFE local bus and the PCI bus for Bus Master channel accesses, assuming a zero wait state target on the PCI bus, for the CFE local bus operating at 25 MHz for 1-wait-state and 0-wait-state slaves.

Table 1-2. PCI Bus Master Performance (MB/sec)

CFE Local Bus Speed (MHz, ws)	Throughput
25,0	90MB/sec
25,1	35MB/sec

- 0 wait states cannot be used when accessing base card memory. Refer to "Memory Performance" for an explanation of memory access rates.
- These numbers are for general reference. Several variables can influence the actual throughput, which may be higher or lower based on these variables.

Diagnostics

The ROM-resident power-on diagnostics are initiated automatically with each power-on sequence of the computer system containing the co-processor. (Refer to *Adapter Initialization and POST* on page 181 for a detailed description of the ROM-resident power-on diagnostics.)

The power-on diagnostics contained on the Micro Channel Startup/Option Diskette or on the PCI OS/2 Support Diskette verify correct operation of the co-processor and perform tests of the following:

- 80960 microprocessor
- Timer
- System Bus
- Memory Protection
- Local Bus
- Memory
- Debug Port.

Hardware Components

- Microprocessor (80960C)
  - 25MHz operation
  - 3-1-1-1 burst operation to DRAM
  - Vectored interrupt support.
- Memory (DRAM and ROM)
  - Packet memory (DRAM SIMM) 1 to 16MB, 1 to 3 wait states
  - Instruction memory (DRAM SOJ modules) 1 or 4MB, 1 to 3 wait states
  - ROM, 128KB, 6 wait states.
- Micro Channel Interface Chip
  - Micro Channel Bus Master interface chip
  - 64-bit, 100-nanosecond streaming data master and slave on Micro Channel
  - 80960 bursting bus interface to CFE Local Bus.
- PCI System Bus Interface Chip
  - Two Bus Master channels, addressable from the CFE Local Bus
  - Slave write buffer (128 bytes) for better PCI utilization
  - Slave prefetch read buffer (128 bytes) for better PCI utilization
  - Support of access to the CFE Local Bus address space from both PCI I/O and memory address spaces
  - PCI data and address parity support
- Memory Controller Chip
  - Two independent DRAM controllers (up to 32MB each)
  - 80960C-Series interface
  - 25MHz operation
  - 32-bit CFE Local Bus interface
  - Local Bus arbiter

  - Memory protection unit
  - Supports 3-1-1-1 memory read cycles; 2-1-1-1 memory write cycles
  - ECC generation and checking for both DRAMs
  - Five hardware timers
  - Asynchronous Serial Debug Port
- Optional AIBs.

Hardware Considerations

RadiSys ARTIC960 Co-Processor Platform (Micro Channel)

The RadiSys ARTIC960 Co-Processor Platform supplies a clock signal (25MHz) to the AIB. This signal **must** be terminated at all times. Normally, the AIB circuitry terminates this signal.

The RadiSys ARTIC960 Co-Processor Platform has a two-pin jumper (J3), which provides an on-card resistance-capacitance (RC) termination for the clock. If you do not have an AIB installed, you should make certain that jumper J3 is between positions 2 and 3 to terminate the clock signal on the RadiSys ARTIC960 Co-Processor Platform.

When an AIB is installed, this jumper must be moved to positions 1 and 2, which is a "no-connect" position for the on-card termination circuitry, and allows the termination to take place using the AIB circuitry.

Damage to the Memory Controller Chip and/or the Micro Channel Interface Chip could occur if the termination is not used when the co-processor is powered up without an AIB attached.

RadiSys ARTIC960 PCI Co-Processor Platform

The RadiSys ARTIC960 PCI Co-Processor Platform supplies a 25 MHz clock to the AIB. This clock signal, at all times, must be terminated. It is the responsibility of the AIB to terminate this signal.

The RadiSys ARTIC960 PCI Co-Processor Platform offers additional write protection for the base card ROM through the use of a 3-pin jumper (P4). When the jumper is in positions 2 and 3, all writes are enabled. When the jumper is in positions 1 and 2, writes are disabled.

To take advantage of the 80960CF data cache, the processor bus is wired to the Memory Controller Chip to allow the use of the 80960CF data cache. See 809ADD for information on how this mapping works.

Functional Subsystems

- Memory Protection
- Error Checking and Correction
- Timer
- Interrupt Controller

- CFE Local Bus
- System Bus Interface
- Debug
- AIB Interface.

Physical Interfaces

- Co-processor platform to host computer
- Co-processor platform to AIB.

Specifications

The following are specifications for the RadiSys ARTIC960 Co-Processor Platform and the RadiSys ARTIC960 PCI Co-Processor Platform, without an AIB attached. For AIB specifications, refer to the appropriate chapter in the *AIB Supplement* that describes the AIB.

RadiSys ARTIC960 Co-Processor Platform (Micro Channel)

Table 1-3. RadiSys ARTIC960 Co-Processor Platform (Micro Channel) Dimensions

Characteristic	Value
Length	282 mm (11.1 inches)
Depth	86 mm (3.4 inches)

Table 1-4. RadiSys ARTIC960 Co-Processor Platform (Micro Channel) Environment

Characteristic	State	Value
Air Temperature	Operating	0°C through 60°C (32°F through 140°F)
	Non-Operating	0°C through 60°C (32°F through 140°F)
Humidity	Operating	5% through 95%
	Wet Bulb Temperature	29.4°C (85°F)

Table 1-5. RadiSys ARTIC960 Co-Processor Platform (Micro Channel) Electrical

Characteristic	Value
Power requirements	+5Vdc, 2.5A (nominal)
	+12 Vdc, 0 mA
	–12 Vdc, 0 mA
Power dissipation	(Typically) 6.5 Watts

RadiSys ARTIC960 PCI Co-Processor Platform (Micro Channel)

Table 1-6. RadiSys ARTIC960 PCI Co-Processor Platform (Micro Channel) Dimensions

Characteristic	Value
Length	312 mm (12.2 inches)
Depth	106 mm (4.2 inches)

Table 1-7. RadiSys ARTIC960 PCI Co-Processor Platform (Micro Channel) Environment

Characteristic	State	Value
Air Temperature	Operating	10°C through 32°C (50°F through 95°F)
	Non-Operating	10°C through 43°C (50°F through 110°F)
Humidity	Operating	8% through 80%
	Wet Bulb Temperature	29.4°C (85°F)

Table 1-8. RadiSys ARTIC960 PCI Co-Processor Platform (Micro Channel) Electrical

Characteristic	Value
Power requirements	+5 Vdc, 2.5 A (nominal)
	+12 Vdc, 0 mA
	–12 Vdc, 0 mA
Power Dissipation	(Typically) 6.5 Watts

.

Notes

1. The operating temperature range and the humidity range for the co-processor are system dependent due to different airflow characteristics.
2. Typical +5 Vdc current usage in Micro Channel systems is 1.3 A, measured in a PS/2 Model 90 with two co-processors and no AIBs attached.

   Typical +5 Vdc current usage in PCI systems is 1.3 A, measured in a ValuePoint Model 6494 with one co-processor platform and no AIBs attached.
3. The voltages shown for "Power Requirements" are not used during normal operation of the co-processor adapter card; however, +12 Vdc is used when reprogramming the co-processor ROM, and ±12 Vdc is used by the Debug Port. The +12 and -12 Vdc lines are provided for use on the AIB.

Co-Processor Platform Components

2

This chapter covers the co-processor platform hardware in detail. It begins with high-level details and then progresses downward to specific registers.

The following terms are used throughout this manual:

AIB	Application Interface Board. This board attaches to the co-processor platform through the AIB Connector. This board also may be referred to as the daughter card.
AIB Interrupts	These are the interrupts from the AIB that go to the co-processor platform's interrupt controller.
DCL BUS	This is the daughter card (AIB) Local Bus. It is the bus that carries the signals that are local to the AIB. These signals are not present at the AIB connector. This bus also be referred to as the DCLB.
CFE Local Bus	This is the bus connecting the ARTIC960 32-Bit Memory Controller Chip, the System Bus Interface Chip, and the AIB connector. This bus supports the Common Front End 32-bit signal set, and also is referred to as the CFE Bus or the Local Bus.
PCI	Peripheral component interconnect.
System Bus Interface Chip (SBIC)	There are two versions of the System Bus Interface Chip. One supports Micro Channel applications; the other supports PCI bus applications. Where possible, *System Bus Interface Chip* is used generically, and specific differences between Micro Channel and PCI applications are noted.

Functional Block Diagram

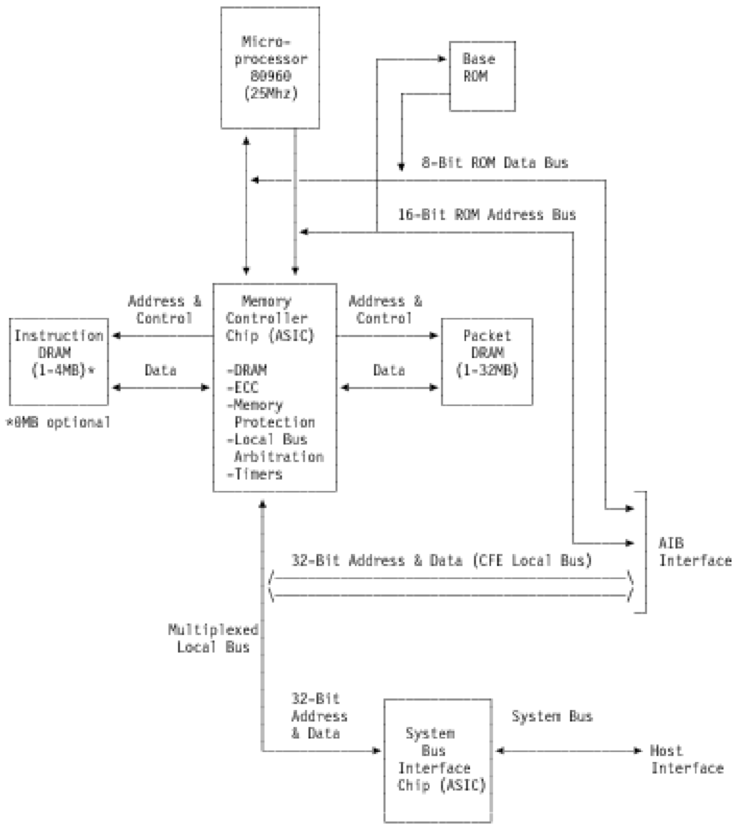


Figure 2-1. Co-Processor Platform Block Diagram

Microprocessor

An Intel 80960C-Series processor is used on the co-processor platform. Earlier co-processor platforms use the CA version and later platforms use the CF version.

Characteristics

- Operation at 25MHz
- CMOS technology
- 1KB, on-chip, two-way set associative instruction cache (CA Series)
- 4KB, on-chip, two-way set associative instruction cache (CF Series)
- 1KB, on-chip, high-speed SRAM (CA and CF Series)
- Vectored interrupt support
- High-bandwidth, 16-byte bursting bus.

Programming Considerations

The *Control Table Structure* on page 179 shows the values programmed into the Control Table, which configures the memory regions.

Reset

On both the Micro Channel and PCI buses, a command reset is passed to the CFE bus through the System Bus Interface Chip; however, there are implementation timing differences. On the Micro Channel bus, both the assertion and deassertion of -CMDRSTOUT (command reset) are synchronized to the CFE bus. On the PCI bus, the assertion of reset is passed through asynchronously; however, the deassertion of reset occurs in an synchronous manner, which will add a PCI 2-clock delay to pass the signal through the chip, and a PCI 2-clock delay to pass the inactive signal to the CFE bus. The following timing diagram illustrates the delay:

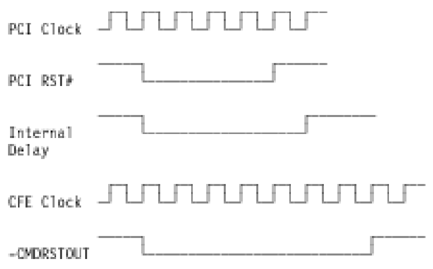


Figure 2-2. Reset Delay Timing

One other difference in reset between Micro Channel and PCI systems is that Ctrl+Alt+Del does not reset adapters in PCI systems.

On the ARTIC960 and ARTIC960 PCI adapters, -CMDRSTOUT is driven to the micro processor. Note that even while reset is active and the processor's outputs are tri-stated, the processor continues to output PCLK.

80960 Addressable Registers

Memory maps of all registers addressable from the 80960 are listed in Table 2-1 through Table 2-8. The registers are grouped by function as follows:

- Memory Interface
- Timers
- Interrupt Control
- Local Bus
- Bus Master Channels (1 and 2)
- SCB Registers
- Serial Debug Port
- Miscellaneous Registers
- Micro Channel-Specific Registers

- PCI-Specific Configuration Registers

Detailed information is located in the column labelled **See Page** in each table. Each register is 4-byte aligned in the address space and should be accessed using 80960 *word* load (ld) and *word* store (st) instructions, unless otherwise noted. In the column labeled **Type**, the abbreviation "ro" is used to denote read-only, and the abbreviation "r/w" is used to denote read/write.

See *Appendix B, Co-Processor Platform Registers* for a list of registers sorted by abbreviation and by address.

The following different types of reset conditions can affect the co-processor platform registers:

1. Reset – A system-unit-initiated reset that pulses a specific Micro Channel or PCI Bus signal. This reset performs the same function as the Power-Up-Reset.
2. Command Reset – A system-unit-generated I/O command initiated under program control, usually by the device driver software. This reset performs a restart of the 80960 processor, as well as setting all registers listed in the following tables into a predefined state. This reset is described in SCP.

The Reset Status Register (RSR) identifies the type of reset that has occurred.

Some register bits affected by Reset are unaffected by Command Reset. Where applicable, these bits are detailed in the individual descriptions for each register. The Reset values and the Command Reset values, labelled Reset and Command Reset, respectively, are shown under the heading **Reset Condition** for each register. This heading also contains the register value presented to the user following POST conditions. Read-only registers or read-only bits in read/write registers will not have a POST initialization value.

Under **Reset Condition**, "U" = Undefined, and "S" = value is the SAME as it was prior to the Reset Command being issued.

Table 2-1. 80960 Addressable Memory Interface Registers

Abbrev	Register Name	Address	Type	See Page
ECCATR	ECC Address Trap Register	1FFB 9034h	ro	35
ECCSTAT	ECC Status Register	1FFB 9038h	ro	35
FEER	Force ECC Error Register	1FFB A010h	r/w	36
LEXATR	Exception Address Trap Register	1FFB 9020h	ro	59
LEXSTAT	Exception Status Register	1FFB 9024h	ro	60
LPATR	Local Bus Protection Addr Trap Register	1FFB 9018h	ro	31
LPSTAT	Local Bus Protection Status Register	1FFB 901Ch	ro	31
MCR	Memory Configuration Register	1FFB A004h	r/w	171
MPER	Memory Protection Enable Register	1FFB 9000h	r/w	27
MPTBR	MC Page Table Base Register	1FFB 9008h	r/w	29
SBATR	Single-Bit ECC Address Trap Register	1FFB 902Ch	ro	34
SBECCR	Single-Bit ECC Error Register	1FFB 9030h	r/w	35
TPATR	Task Protection Address Trap Register	1FFB 9010h	ro	30

Table 2-1. 80960 Addressable Memory Interface Registers

Abbrev	Register Name	Address	Type	See Page
TPSTAT	Task Protection Status Register	1FFB 9014h	ro	30
TPTBR	Task Page Table Base Register	1FFB 9004h	r/w	29

Table 2-2. 80960 Addressable Timer Registers

Abbrev	Register Name	Address	Type	See Page
TCMD	Timer Command Registers	1FFB *x*00Ch	r/w	41
TCR	Timer Control Registers	1FFB *x*000h	r/w	39
TPR	Timer Preset Registers	1FFB *x*004h	r/w	40
TPV	Timer Present Value Registers	1FFB *x*008h	ro	40

x = timer port number (0 to 4)

Table 2-3. 80960 Addressable Interrupt Controller Registers

Abbrev	Register Name	Address	Type	See Page
EDR	Enable/Detect Register	0A00 0004h	r/w	46

Table 2-4. 80960 Addressable Local Bus Interface Registers

Abbrev	Register Name	Address	Type	See Page
LBBAR	Local Bus Base Address Register	1FFA 0024h	r/w	102 (uCh) 149 (PCI)
LBCFG	Local Bus Config Register	1FFB A00Ch	r/w	62
LBPE	Local Bus Parity/Exception Register	1FFA 0020h	ro	58
SPAR	Special Arbitration Register	1FFB A008h	r/w	62

Table 2-5. 80960 Addressable Bus Master Channel X Interface Registers

Abbrev	Register Name	Address	Type	See Page
BCR	Bus Master Channel x Byte Count Register	1FFA x008h	r/w	93 (uCh) 141 (PCI)
BMAR	Bus Master Channel x Address Register	1FFA x010h	r/w	95 (uCh) 143 (PCI)
BMCMD	Bus Master Channel x Command Register	1FFA x01Ch	r/w	99 (uCh) 146 (PCI)
BMSTAT	Bus Master Channel x Status Register	1FFA x018h	ro	97 (uCh) 144 (PCI)
CAR	Bus Master Channel x Card Address Register	1FFA x000h	r/w	92 (uCh) 140 (PCI)
CCR	Bus Master Channel x Control Register	1FFA *x*00Ch	r/w	93 (uCh) 141 (PCI)

Table 2-5. 80960 Addressable Bus Master Channel X Interface Registers

Abbrev	Register Name	Address	Type	See Page
LAP	Bus Master Channel x List Address Pointer	1FFA *x*014h	r/w	96 (uCh) 144 (PCI)
SAR	Bus Master Channel x System Address Register	1FFA *x*004h	r/w	92 (uCh) 140 (PCI)

For Bus Master Channel 1, *x* = 3; for Bus Master Channel 2, x = 4.

Table 2-6. 80960 Addressable SCB Registers

Abbrev	Register Name	Address	Type	See Page
ATTN	SCB Attention Port	1FFA 2004h	ro	107 (uCh) 157 (PCI)
CBSP	SCB Command/Busy Status Port	1FFA 2010h	r/w	112 (uCh) 161 (PCI)
COMMAND	SCB Command Port	1FFA 2000h	ro	107 (uCh) 156 (PCI)
ISP	SCB Interrupt Status Port	1FFA 200Ch	r/w	111 (uCh) 159
SIR	Source Identification Register	1FFA 2014h	r/w	113 (uCh) 162 (PCI)
SCP	SCB Subsystem Control Port	1FFA 2008h	ro	110 (uCh) 158 (PCI)

Table 2-7. 80960 Addressable Serial Debug Port Registers

Abbrev	Register Name	Address	Type	See Page
PCR	Port Configuration Register	1FFB 8008h	r/w	176
RXBUF	Rx Buffer Port	1FFB 8004h	r/w	175
TXBUF	Tx Buffer Port	1FFB 8000h	r/w	174

Table 2-8. 80960 Addressable Miscellaneous Registers

Abbrev	Register Name	Address	Type	See Page
AWE	AIB ROM Write Protect Register	09FF FFF4h	r/w	167
BWE	Base ROM Write Protect Register	09FF FFF0h	r/w	168
EDR	Enable/Detect Register	0A00 0004h	r/w	46
GAID	Micro Channel Interface Chip Gate Array ID Register	1FFA 0008h	ro	169
GAID	PCI Bus Interface Chip Gate Array ID Register	1FFA 0008h	ro	169
GAIDR	Memory Controller Gate Array Register	1FFB A000h	ro	169
IVR	Interrupt Controller Version Register	0A00 0000h	ro	170
MPH	Presence Detect High Register	0A00 000Ch	ro	171
MPL	Presence Detect Low Register	0A00 0008h	ro	170

Table 2-9. Micro Channel 80960 Addressable Registers

Abbrev	Register Name	Address	Type	See Page
CRDID	Card ID Register	1FFA 000Ch	r/w	82
LBPE	Local Bus Parity/Exception Register	1FFA 0020h	ro	58
NMI	NMI Command Register	1FFA 001Ch	ro	114
PROC_CFG	Processor Configuration Register	1FFA 0010h	r/w	78
POS_SETUP1	POS Setup 1 Register	1FFA 0000h	ro	70
POS_SETUP2	POS Setup 2 Register	1FFA 0004h	ro	74
RSR	Reset Status Register	1FFA 0014h	ro	114
XPOS	Extended POS Register	1FFA 0018h	r/w	81

Table 2-10. PCI 80960 Addressable Configuration Registers

Abbrev	Register Name	Address	Type	See Page
CCRID	Class Code/Revision ID Register	1FFA 5008h	ro	126
DEVID	Device ID Register	1FFA 5000h	r/w	124
HMBAR	Host Memory Base Address Register	1FFA 5014h	r/w	128
HMFR	Host Miscellaneous Functions Register	1FFA 500Ch	ro	127
HSCR	Host Status Command Register	1FFA 5004h	ro	125
IOBAR	I/O Base Address Register	1FFA 5010h	ro	128
LGIR	Latency/Grant/Interrupt Register	1FFA 503Ch	r/w	131
MMBAR	Memory Mapped I/O Base Address Register	1FFA 5018h	r/w	129
NMI	NMI Command Register	1FFA 001Ch	ro	164
PROC_CFG	Processor Configuration Register	1FFA 0010h	r/w	78
RSR	Reset Status Register	1FFA 0014h	ro	163
SEER	Serial EPROM Extension Register	1FFA 0028h	r/w	123
SSID	Subsystem ID Register	1FFA 502Ch	r/w	130
XRBAR	Expansion ROM Base Address Register	1FFA 5030h	r/w	130

Memory

Both DRAM and ROM are supported on the co-processor platform. Since the entire address space of the 80960 is memory mapped, all registers appear as memory.

Dynamic Random-Access Memory (DRAM)

The co-processor platform can be configured with either one or two DRAM banks. The execution/packet memory is the default memory bank. The co-processor platform also supports an optional execution-only (instruction memory) DRAM bank for those applications requiring higher I/O throughput and/or processing bandwidth. This memory can have granularities of 1 or 4MB. The starting address for packet memory is 512MB (2000 0000h), and the starting address for instruction memory is 564MB (2200 0000h). This information is stored in the memory configuration register as described in MCR. All

DRAM memory on the adapter is Error Correction Code (ECC) protected. Single-bit errors are corrected and can be counted under program control, while double-bit errors are detected and reported as a non-maskable interrupt (NMI) back to the 80960. Random access cycle time to the DRAM is 200 nanoseconds and page mode cycle time is 80 nanoseconds.

Instruction memory must be used **only** when downloading code to it, or executing code from it. This memory must not be used for storing data; for example, Channel Descriptor Blocks. The reason for this restriction is that long accesses to instruction memory from the Local Bus can result in Micro Channel timeouts if other processes are attempting to use the Local Bus, or poor performance in PCI systems.

Read-Only Memory (ROM)

The co-processor platform supports a minimum of 128KB of ROM memory. This memory contains the power-on self-test (POST) code and the bootstrap loader code (to allow the control program to be loaded to the card from the system unit). An area is reserved in the ROM to support vital product data (VPD) and serialization. This code is RadiSys-supplied.

Throughout this manual, the ROM is also referred to as PROM.

Software Support

The co-processor platform ROM provides the following support functions:

- POST for the co-processor platform
- Interface structures for the AIB ROM POST and supervisor tasks.
- Synchronization between the system unit POST, the co-processor platform POST, and the AIB POST.
- VPD structures for the co-processor platform and the AIB.
- Error reporting structures for the co-processor platform and the AIB POST routines.

See *Adapter Initialization and POST* on page 181 for a detailed description of the power-on diagnostics.

Programming Considerations

The ROM used on the co-processor platform is a “flash” memory, which can be erased and/or written while on the adapter. To prevent accidental erasure, the co-processor platform implements a protection scheme which requires a write enable bit to be set. Once this is done, then the ROM can be written. For details of this protection scheme, see *AIB ROM Write Enable Register (AWE)* on page 167 and/or the *Base ROM Write Enable Register (BWE)* on page 168.

Serial EPROM

The RadiSys ARTIC960 PCI Co-Processor Platform supports the use of a serial EPROM (erasable programmable read-only memory) to supply the PCI system with configuration data. This data is programmed prior to being installed on the base card, at the time of manufacture; it cannot be erased or programmed after this time.

The RadiSys ARTIC960 PCI Co-Processor Platform offers additional write protection for the base card ROM through the use of a 3-pin jumper (P4). When the jumper is in positions 2 and 3, all writes are enabled. When the jumper is in positions 1 and 2, writes are disabled.

Programming Considerations

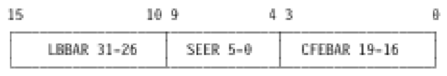
A serial EPROM interface of three pins is provided for allowing the automatic configuration of the SSID register, the CFEBAR, the LBBAR, and the SEER. The System Bus Interface Chip detects the presence or absence of the serial EPROM, and all registers are unaffected by the absence of the serial EPROM.

The serial EPROM interface is an SGS-Thomson ST93C06 or equivalent. The following values are programmed into this chip with the format specified in Table 2-11.

Table 2-11. Serial EPROM Values

Address	Description	Values
B'000000'	SSID 15-0	1014h
B'000001'	SSID 31-16	0043h
B'000010'	Configuration Data	202Ah

The format of the configuration data is provided in the following figure.



DRAM/ROM Memory Map

As shown in Figure 2-3, the 80960 processor supports a 4-gigabyte (GB) linear address space. There are no explicit I/O type instructions within the 80960 instruction set. Hence, all I/O must be memory mapped. On the co-processor platform, the region above the 512MB address boundary is allocated to address DRAM. Also, separate 512KB regions have been allocated for base card memory-mapped I/O and for AIB memory-mapped I/O.

Portions of the 80960 address space are shared with (translated to) the CFE Local Bus address space. A shared address space is defined as an address range where an 80960 access will cause a cycle or access on the CFE Local Bus (for example, the processor reading a System Bus Interface Chip register). For accesses to the shared address space, the Memory Controller Chip simply passes the 80960 access onto the CFE Local Bus. This is true for all 80960 accesses to the shared address space with the exception of DRAM. Although DRAM is located in the shared address range, the 80960 access goes directly to DRAM, and is not passed onto the CFE Local Bus. It should be noted that there is no path for a CFE Local Bus access to be passed onto the 80960 bus in any address range.

Programming Considerations

RadiSys ARTIC960 co-processor adapters EC level E32319 or later, and all RadiSys ARTIC960 PCI co-processor adapters provide hardware support for 80960CF data caching. Since the two adapters implement the caching scheme differently, memory region overlays are different on these adapters, except in both cases, region 10 will overlay region 2. This allows the Memory Controller Chip (region 2) to be accessed starting at address A000 0000h, where the data cache can be enabled.

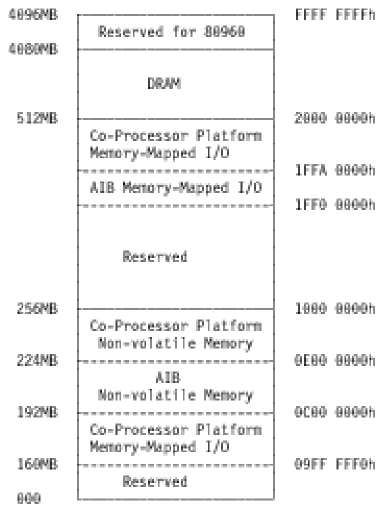


Figure 2-3. Co-Processor Platform Memory Map (as viewed from the 80960)

A more detailed memory map is shown in Table 2-12. Details on how to enable memory protection are described in the *Chapter 3, Memory Protection Subsystem*.

Table 2-12. Detailed Co-Processor Platform Memory Map (as viewed from the 80960)

From (Hex)	To (Hex)	Comments
0000 0000	0000 03FF	80960 Internal Data RAM. Cycles do not appear external to the 80960. Non-burst and address space are not shared with the Local Bus.
0000 0400	09FF FFEF	Reserved. Non-burst and address space are not shared with the Local Bus.
09FF FFF0	09FF FFFF	ROM Write Protect Registers. Non-burst and address space are not shared with the Local Bus.
0A00 0000	0BFF FFFF	Interrupt/Presence Detect Registers. Non-burst and address space are not shared with the Local Bus.
0C00 0000	0DFF FFFF	AIB ROS Chip Select. Non-burst and address space are not shared with the Local Bus.
0E00 0000	0FFF FFFF	Base Card ROS Chip Select. Non-burst and address space are not shared with the Local Bus.
1000 0000	1FEF FFFF	Reserved. Non-burst and address space are shared with the Local Bus.
1FF0 0000	1FF9 FFFF	AIB I/O Area. Non-burst and address space are shared with the Local Bus.
1FFA 0000	1FFA FFFF	System Bus Interface Chip I/O Area. Non-burst and address space are shared with the Local Bus.
1FFB 0000	1FFB FFFF	Memory Controller Chip I/O Area. Non-burst and address space are not shared with the Local Bus.
1FFC 0000	1FFF FFFF	Reserved. Non-burst and address space are shared with the Local Bus.
2000 0000 (See Note)	25FF FFFF	Burst RAM area and address space are shared with the Local Bus. Packet memory can be located from 2000 0000h up to 21FF FFFFh. Instruction memory can be located from 2200 0000h up to 23FF FFFFh. In areas where DRAM is located, the 80960 accesses do not appear on the Local bus and the Memory Controller Chip responds to the memory accesses.
2600 0000 (See Note)	2FFF FFFF	AIB Memory Area. Non-burst and address space are shared with the Local Bus.
3000 0000	9FFF FFFF	Reserved.
A000 0000[1]	AFFF FFFF	Reserved Burst Area RAM. In areas where DRAM is located, the 80960 accesses do not appear on the CFE bus, and the Memory Controller Chip responds to the memory access. Packet memory can be located from A0000000 up to A1FFFFFF. Instruction memory can be located from A2000000 up to A3FFFFFF. All other addresses in this region are reserved.
B000 0000	FFFF FEFF	Reserved.
FFFF FF00	FFFF FF2F	Base ROM Area. Non-burst and address space are **not** shared with the Local Bus.

Table 2-12. Detailed Co-Processor Platform Memory Map (as viewed from the 80960)

From (Hex)	To (Hex)	Comments
FFFF FF30	FFFF FFFF	Reserved. Non-burst and address space are **not** shared with the Local Bus. **Note:** This address map shows shared Bursting ranges with regard to the 80960. Ranges identified with this Note are Non-Bursting on the CFE Local Bus for 80960 accesses, but reside in the architected CFE Burst address range. Accesses from the System Bus Interface Chip in these ranges will be Burst accesses.

1 Maps to region 2 when hardware is enabled.

Memory Controller Chip (ASIC)

The Memory Controller Chip is a RadiSys Application-Specific Integrated Circuit (ASIC), containing over 30000 gates. This chip interfaces to the 80960C-Series processor, the 32-bit CFE Local Bus, and two dynamic random-access memories (DRAMs). The memory interfaces are accessed independently for concurrent operation of the processor and the Local Bus. The chip permits high-speed access from the processor to Local Bus memory, allowing full addressability of both memories from each bus. Memory protection support and Error Checking and Correcting (ECC) ensure data integrity.

Functional Summary

- Two independent DRAM controllers
- 25MHz processor bus-to-Local Bus conversion
- High-speed, 32-bit CFE Local Bus with address and data parity
- Supports 3-1-1-1 burst transfers to either memory
- Flow-through ECC with single-bit correction, double-bit detection on both memories
- Hardware support for memory protection
- Five programmable timers
- I/O mapping for testing in-circuit connections.

System Bus Interface Chip

The following sections show the differences between the two versions of the System Bus Interface Chip. *Micro Channel Interface Chip Application-Specific Integrated Circuit (ASIC)* on page 21 describes the Micro Channel features of the chip. *PCI System Bus Interface Chip* on page 21 describes the PCI features of the chip.

Micro Channel Interface Chip Application-Specific Integrated Circuit (ASIC)

The major functions of the ARTIC 32-bit Micro Channel Interface Chip are as follows.

- Two Bus Master channels, addressable from the local bus
- One 128-byte intermediate data buffer per channel
- Linked List Chaining support for auto-initialization of either channel
- Support of Micro Channel basic transfers as master and slave
- Support of 64-bit, 100 ns streaming data as master and slave
- Slave write buffer (192 bytes) for better Micro Channel utilization
- Slave prefetch read buffer (32 bytes) for better Micro Channel utilization
- Hardware support for Appended I/O operations
- Hardware support for SCB Locate and Move mode
- Support of access to resident memory as both I/O and shared memory window
- Micro Channel data and address parity support
- Support for Micro Channel interrupts and error reporting
- Directly attachable to Micro Channel with 24 mA off-chip drivers
- I/O Mapping for testing of In-Circuit connections

The Micro Channel Interface Chip provides a Micro Channel slave interface, supporting 100 ns streaming data through posted memory writes and prefetched memory reads. Micro Channel Bus Masters access slave resident memory on the Local Bus through a Micro Channel shared memory window.

The Micro Channel host accesses the Local Bus through the shared memory window or through a Micro Channel I/O location.

The Micro Channel Bus Master interface consists of two Bus Master channels addressable from the Local Bus. Each channel utilizes a 128-byte intermediate buffer between resident memory and the Micro Channel. Buffering of data allows the Bus Master channel to sustain 100-ns streaming data on the Micro Channel.

PCI System Bus Interface Chip

The major functions of the PCI System Bus Interface Chip are as follows.

- Operationally compatible with the Micro Channel System Bus Interface Chip
- Two Bus Master channels, addressable from the CFE Local Bus
- One 128-byte intermediate data buffer per channel
- Support for Posting Status from each Bus Master Channel
- Linked List Chaining support for auto-initialization of either channel
- Slave write buffer (128 bytes) for PCI Posted Write Operation
- Slave prefetch read buffer (128 bytes) for PCI Delayed Read Operation

- Master and Slave Support of PCI Bus Commands
- Hardware support for SCB Locate and Move mode
- Support of access to the CFE Local Bus address space from both PCI I/O and memory address spaces
- PCI data and address parity support
- Support for PCI interrupts and error reporting
- Directly attachable to PCI
- JTAG for testing of In-Circuit connections

The PCI System Bus Interface Chip provides a PCI target interface, supporting posted memory writes and delayed/prefetched memory reads to the CFE Local Bus address space. These accesses are mapped to a memory window on the PCI bus. A PCI master, or initiator, can access CFE Local Bus address space through this PCI slave interface. The PCI host accesses the CFE Local Bus through this memory window or through a PCI I/O location known as the Memory Data (MDATA) port. The PCI slave interface also supports a register space, mapped to both memory and I/O space on the PCI Bus.

The PCI Bus Master interface consists of two DMA, or Bus Master channels, addressable from the CFE Local Bus, or indirectly from the PCI Bus through the MDATA port. The term *Bus Master Channel* refers to these two DMA channels’ interface to the PCI System Bus Interface Chip. Each channel uses a 128-byte intermediate buffer between the CFE Local Bus and the PCI Bus.

Memory Protection Subsystem 3

The Memory Controller Chip provides memory protection hardware external to the 80960. This feature allows for 80960 process (task) level, and Local Bus device protection of memory and memory-mapped I/O (MMIO) on 4KB pages. For tasks, the protection covers packet memory, instruction memory, and I/O addresses (1FFx xxxxh). For the conditions where the System Bus Interface Chip issues a slave request on the co-processor platform, protection covers instruction and packet memory only. Turning protection on or off is specified by setting bits in the Memory Protection Enable Register, while the type of protection is specified by entries in memory-resident structures called page tables. The Memory Controller Chip hardware registers provide table base address pointers and exception status information. It should be noted that the management of these registers and tables is typically done by the supervisory level software running on the co-processor platform. For more details on this protection feature, refer to the *RadiSys ARTIC960 Co-Processor Platforms Programmer's Guide and Reference*.

Memory Protection Map

The Memory Controller Chip allows for protection of the entire 80960 4GB memory space. The protection scheme breaks the co-processor platform address space into the following sections:

- Low Memory (LOWMEM) accesses from 0000 0000h to 1FEF FFFFh
- RAM and I/O (RAMIO) accesses to DRAM and 1FFx xxxxh
- Local Bus (LBMEM) accesses from 2000 0000h to 7FFF FFFFh
- Upper Memory (UPMEM) accesses from 8000 0000h to FFFF FFFFh

Memory protection can be enabled separately for each of these regions. The Low Memory region operates without memory protection; however, a range check is done to trap all accesses with this area if the memory protection enable bits are set.

From the 80960 processor, the memory protection map looks as shown in Table 3-1

.

Table 3-1. 80960 Processor Memory Protection Map

From (hex)	To (hex)	MPER Bits	Comments
0000 0000	0000 03FF	- - -	80960 internal data RAM. Cycles are not seen on the Local Bus.
0000 0400	1FEF FFFF	960TSK LOWMEM	Range checking only if protection bits are set.
1FF0 0000	1FF9 FFFF	960TSK RAMIO ALLRD	AIB address area.
1FFA 0000	1FFA FFFF	960TSK RAMIO ALLRD	System Bus Interface Chip address area.
1FFB 0000	1FFB FFFF	960TSK RAMIO ALLRD	Memory Controller Chip address area.
1FFC 0000	1FFF FFFF	960TSK RAMIO ALLRD	Reserved
2000 0000	xxxxxxxx	960TSK RAMIO ALLRD	Packet memory area. Address range is determined by the size of memory (21FF FFFFh for 32MB).
2200 0000	xxxxxxxx	960TSK RAMIO ALLRD	Instruction memory area. Address range is determined by the size of memory (23FF FFFFh for 32MB).
xxxx xxxx	7FFF FFFF	960TSK LBMEM ALLRD	Covers ALL addresses from 2000 0000h to 7FFF FFFFh, except where packet and instruction memory are located.
8000 0000	9FFF FFFF	960TSK UPMEM	None
A000 0000	AFFF FFFF	960TSK RAMID ALLRD	When data cache is enabled.
B000 0000	FFFF FFFF	960TSK UPMEM	None

Local Bus Memory Protection Map

From the Local Bus, the memory protection map looks as shown in Table 3-2.

Table 3-2. Local Bus Memory Protection Map

From (hex)	To (hex)	MPER Bits	Comments
0000 0000	1FEF FFFF	MST0 LOWMEM	Protection checking only.
1FF0 0000	1FFF FFFF	- - -	No protection checking.
2000 0000	xxxx xxxx	MST0	

ALLRDPacket memory area. Address range is determined by the size of memory (21FF FFFFh for 32MB).

ALLRDInstruction memory area. Address range is determined by the size of memory (23FF FFFF for 32MB).

Page Table Entries and Page Tables

All page tables must start on a 64-byte boundary in memory, and are composed of entries covering 64KB areas. Each entry in the page table is subdivided into sixteen, 4KB pages.

Figure 3-1 illustrates a page table entry, and Table 3-3 shows the bit settings for the various protection options. The length of each page table is determined by the amount of memory installed.

Page Table Entries

Each page table entry covers a 64KB area, which is subdivided into sixteen 4KB pages.

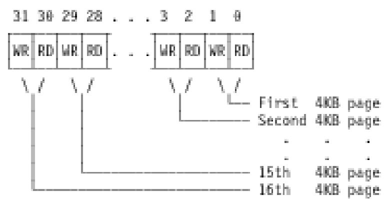


Figure 3-1. Page Table Entry Format

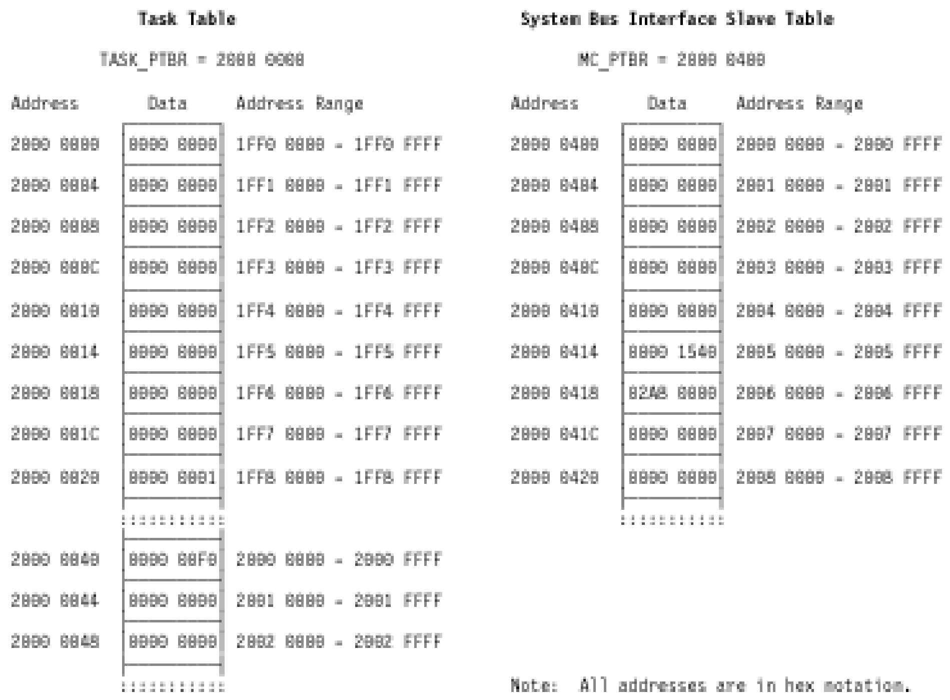
Table 3-3. Page Access Bit Definition

WR	RD	Meaning
0	0	Inaccessible
0	1	Read-Only
1	0	Write-Only
1	1	Read/Write

The access bits are checked by hardware on each protected access, and an interrupt is generated and the violating address latched if a violation is detected. (See *Task Protection Address Trap Register (TPATR)* on page 30 or *Local Bus Protection Address Trap Register (LPATR)* on page 31.) Also, in the case of a write violation, the write to memory will be blocked by the hardware. On 80960 accesses to the Local Bus, the Memory Controller Chip will *prevent* the Local Bus cycle unless the access is allowed by protection.

Page Tables

Page table length is based on the amount of memory installed. Page tables *must* start on a 64-byte boundary in memory. Page table entries are 32-bits wide and cover sixteen, 4KB pages. Examples of a task page table and Local Bus page table are shown in Figure 3-2.



Task Table

TASK_PTBR = 2000 0000

Address	Data	Address Range
2000 0000	0000 0000	1FF0 0000 - 1FF0 FFFF
2000 0004	0000 0000	1FF1 0000 - 1FF1 FFFF
2000 0008	0000 0000	1FF2 0000 - 1FF2 FFFF
2000 000C	0000 0000	1FF3 0000 - 1FF3 FFFF
2000 0010	0000 0000	1FF4 0000 - 1FF4 FFFF
2000 0014	0000 0000	1FF5 0000 - 1FF5 FFFF
2000 0018	0000 0000	1FF6 0000 - 1FF6 FFFF
2000 001C	0000 0000	1FF7 0000 - 1FF7 FFFF
2000 0020	0000 0001	1FF8 0000 - 1FF8 FFFF
	:::::::::::	
2000 0040	0000 00F0	2000 0000 - 2000 FFFF
2000 0044	0000 0000	2001 0000 - 2001 FFFF
2000 0048	0000 0000	2002 0000 - 2002 FFFF
	:::::::::::	

System Bus Interface Slave Table

MC_PTBR = 2000 0400

Address	Data	Address Range
2000 0400	0000 0000	2000 0000 - 2000 FFFF
2000 0404	0000 0000	2001 0000 - 2001 FFFF
2000 0408	0000 0000	2002 0000 - 2002 FFFF
2000 040C	0000 0000	2003 0000 - 2003 FFFF
2000 0410	0000 0000	2004 0000 - 2004 FFFF
2000 0414	0000 1540	2005 0000 - 2005 FFFF
2000 0418	02A0 0000	2006 0000 - 2006 FFFF
2000 041C	0000 0000	2007 0000 - 2007 FFFF
2000 0420	0000 0000	2008 0000 - 2008 FFFF
	:::::::::::	

Note: All addresses are in hex notation.

Figure 3-2. Sample Page Tables

The task table shows a 4KB read-only area starting at 1FF8 0000h and an 8KB read/write area starting at 2000 2000h.

The System Bus Interface Slave table shows a 16KB read-only area starting at 2005 3000h and a 16KB write-only area starting at 2006 9000h. Note that task tables begin with 1FF0 0000h, and local masters start with 2000 0000h. The rest of the entries cover to the top of installed memory. Local Bus memory is not protected by these tables.

Page Table Entry Caching

To limit the number of memory references the hardware must make to check a page table entry, three separate caches are maintained in the Memory Controller Chip chip. An 8-entry, full-way, associative cache is maintained for task page entries. This can cover a total of 512KB in 64KB pieces. If the cache is full and another entry needs to be cached, the Memory Controller Chip will discard the least-recently-used entry to make room for the new one. A 1-entry cache is maintained for AIB Device Master accesses. A 2-entry cache is maintained for the System Bus Interface Chip interface. For the System Bus Interface Chip interface, one entry is cached for writes, and one is cached for reads; however, accesses are checked against both entries.

When the supervisor performs a task switch, it performs a write to the Task Page Table Base Register (TASK_PTBR). This write causes all task page access bytes presently in the cache to be *invalidated*. Similarly, a write to AM_PTBR invalidates its page access byte, and a write to MC_PTBR invalidates the System Bus Interface Chip slave page access byte cache. The Memory Controller Chip does *not* check writes to memory to determine if they are cache entries. In order to be sure a change to the tables will be seen by the Memory Controller Chip, the PTBR *must* be written after the change.

Operating Sequence

1. The Memory Controller Chip determines the accessing unit (80960 task, AIB Device Master Device, or System Bus Interface Slave).
2. The Memory Controller Chip checks if protection is "on" for the accessing unit and the specific address it is accessing (access address). If it is not, no violation occurs.
3. The Memory Controller Chip checks if a page table entry has been cached for this address (see *Page Table Entry Caching* on page 26). If it has been cached, then go to 5.
4. The Memory Controller Chip uses the appropriate PTBR (the upper 16 bits of the access address are used as an index) to fetch the correct page table entry..

   This entry will now be cached.

5. The Memory Controller Chip determines whether this access is a violation, and takes the appropriate action.

Memory Protection Performance

Protection cache misses will cause accesses to be delayed up to 10 extra clocks.

Memory Protection Registers

The following Memory Controller Chip registers are used, in addition to the page tables and Memory Protection Enable Register to implement memory protection.

Memory Protection Enable Register (MPER)

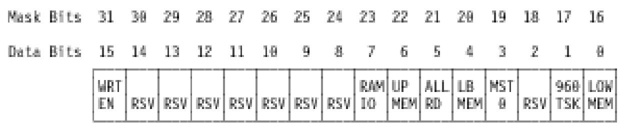
This register provides the capability to turn memory protection on and off separately for the various regions in the memory map. It also provides the capability to turn protection on and off for tasks, and for the System Bus Slave interfaces. A kernel will normally execute without RAM/IO protection; device drivers and interrupt handlers would execute with or without protection at their option.

This register is *not* protected by the hardware. When writing to this register, the upper 16 bits serve as a mask for the lower 16 bits. For example, if you wanted to turn on the 80960 task protection bit, you would store the word 0002 8002h. This tells the Memory Controller Chip to write the register (bit 15 = 1), write bit 1 (bit 17 = 1), and set bit 1 to a **1** (bit 1 = 1). The mask bits always read 0s. Bit 15 must be a 1 to write this register. This register can only be written using 32-bit writes (that is, all bytes are enabled).

For protection to be enabled for a particular access, all bits/signals relating to that access must be in the correct state. (For example, for a DRAM read from the 80960: 960TSK=1, ALLRD=1, RAMIO=1 and the -PDMA signal is high.)

Register Format:

(ADDRESS = 1FFB 9000h) read/write



Mask Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Data Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WRT EN	RSV	RSV	RSV	RSV	RSV	RSV	RSV	RAM IO	UP MEM	ALL RD	LB MEM	MST 0	RSV	960 TSK	LOW MEM

- Bit 15: MPER Write Enable Bit.

  This bit must be a 1 to write this register. This bit always reads 0.

- Bits 14 to 8: Reserved. These bits must be set to 0.
- Bit 7: RAM/IO Protection. (Bit 23 must be a 1 to write this bit.)

  0 – Accesses from 80960 to the Memory Controller Chip DRAM, or 1FFx xxxxh are unprotected. (Out-of-bounds checking can still be on.)

  1 – Accesses to this region are protected.

- Bit 6: Upper Memory (2GB to 4GB) Protection. (Bit 22 must be a 1 to write this bit.)

  0 – Accesses to this region are unprotected.

  1 – Accesses to this region cause a memory protection error.

- Bit 5: All Read Protection. (Bit 21 must be a 1 to write this bit.)

  0 – Turns off read protection for packet and instruction memory, and 1FFx xxxxh area. (This feature is useful for increasing performance for non-debug memory protection users.)

  1 – Reads to protected areas are checked.

- Bit 4: Local Bus Protection Bit. (Bit 20 must be a 1 to write this bit.)

  This bit determines whether the Memory Controller Chip will allow the 80960 to access the Local Bus in the address from 2000 0000h to 7FFF FFFFh, except where packet and/or instruction memory are defined.

  0 – Local Bus memory allowed, no protection.

  1 – Protection on. An attempt to access this region will cause a protection error.

- Bit 3: System Bus Slave Accesses to Local Bus. (Bit 19 must be a 1 to write this bit.)

  0 – System Bus Slave accesses are unprotected.

  1 – System Bus Slave accesses are protected.

- Bit 2: Reserved. This bit must be set to 0.
- Bit 1: 80960 Accesses. (Bit 17 must be a 1 to write this bit.)

  0 – 80960 accesses are unprotected.

  1 – 80960 accesses are protected.

- Bit 0: Low Memory (<511MB) Protection. (Bit 16 must be a 1 to write this bit.)

  0 – Low memory accesses are unprotected.

1 – Access to this area causes protection error.

Reset Conditions:

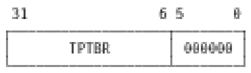
```
               Reset: 0000 0000 0000 0000 0000 0000 0000 0000
ROM Initialization: 0000 0000 0000 0000 0000 0000 0000 0000
```

Task Page Table Base Register (TPTBR)

This register (which is normally changed by supervisory level software at task switch time) is a pointer to the starting address of a task's page table. Logic interprets all writes to this register as a task switch, and invalidates all entries in the task page-access-byte cache. The six, least significant bits of this register will always read 0. See *Chapter 3, Memory Protection Subsystem*. Task protection tables must be in instruction memory if it is installed.

Register Format:

ADDRESS = 1FFB 9004h read/write



Reset Conditions:

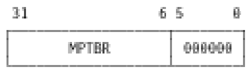
```
               Reset: SSSS SSSS SSSS SSSS SSSS SSSS SS00 0000
ROM Initialization: 0000 0000 0000 0000 0000 0000 0000 0000
```

System Bus Slave Page Table Base Register (MPTBR)

This register (which normally is managed by the supervisory level software) is a pointer to the starting address of the System Bus Slave interface page table. A write to this register invalidates all entries in the System Bus Slave page-access-byte cache. The six, least significant bits of this register will always read 0. See *Chapter 3, Memory Protection Subsystem*. System Bus Slave protection tables must be in packet memory.

Register Format:

(ADDRESS = 1FFB 9008h) read/write



Reset Conditions:

```
                Reset: SSSS SSSS SSSS SSSS SSSS SSSS SS00 00000
ROM Initialization: 0000 0000 0000 0000 0000 0000 0000 00000
```

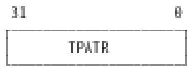
Task Protection Address Trap Register (TPATR)

This register latches the address that caused a memory protection violation by an 80960 task. An interrupt is generated. Once the register has trapped an address, it is locked and will not trap any subsequent failures until the TPSTAT Register is read. The data is valid until the TPSTAT Register is read.

The TPATR Register should be read before reading the TPSTAT Register.

Register Format:

(ADDRESS = 1FFB 9010h) read-only



Reset Conditions:

```
Reset: UUUU UUUU UUUU UUUU UUUU UUUU UUUU UU00
```

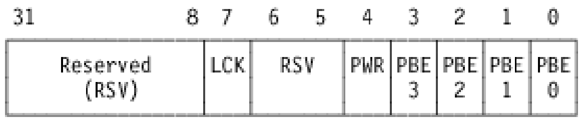
Task Protection Status Register (TPSTAT)

This register latches the status associated with a memory protection violation by an 80960 task. An interrupt is generated. Once the register has trapped status, it is locked and will not trap any subsequent failures until the TPSTAT Register is read.

The TPSTAT Register should be read after reading the TPATR Register.

Register Format:

(ADDRESS = 1FFB 9014h) read-only



Bit Descriptions:

- Bits 31 to 8: Reserved. These bits must be set to 0.
- Bit 7: When this bit is a 1, it indicates that the TPSTAT Register and the TPATR Register are locked, and the interrupt is pending. This bit and the interrupt are cleared when the TPSTAT Register is read.
- Bit 4: A 1 indicates a write operation was trapped. A 0 indicates a read operation.

- Bits 3 to 0: These bits indicate which byte enables were active when the trap occurred (1=enabled).

Reset Conditions:

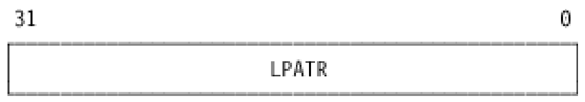
```
Reset: 0000 0000 0000 0000 0000 0000 0UUU UUUU
```

Local Bus Protection Address Trap Register (LPATR)

This register latches the address that caused a Local Bus (System Bus Slave) protection violation. An interrupt is generated. Once the register has trapped an address it is locked and will not trap any subsequent failures until both Local Bus protection interrupts are cleared. Bits in the LPSTAT Register indicate whether the address trapped is for System Bus Slave protection. The LPATR Register is valid until the LPSTAT Register is cleared.

Register Format:

(ADDRESS = 1FFB 9018h) read-only



Reset Conditions:

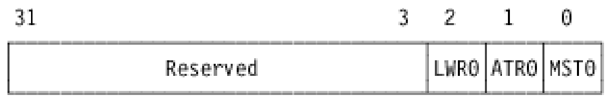
```
Reset: UUUU UUUU UUUU UUUU UUUU UUUU UUUU UUUU
```

Local Bus Protection Status Register (LPSTAT)

This register latches the status associated with a Local Bus (System Bus Slave) protection violation. An interrupt is generated. Once the register has trapped status, it is locked and will not trap any subsequent failures until the LPSTAT Register is read. This register is byte addressable. When clearing interrupts, writes *should* be byte writes. The LPSTAT Register should be cleared *after* reading the LPATR.

Register Format:

(ADDRESS = 1FFB 901Ch) read-only – byte addressable



Bit Descriptions:

- Bits 31 to 3: Reserved. These bits must be set to 0.
- Bit 2: Write/Read bit for System Bus Slave Protection.

  A 1 indicates a write operation was trapped; a 0 indicates a read. This bit is valid only if Bit 0 = 1.
- Bit 1: A 1 indicates the LPATR contains the violating address for this interrupt. This bit is valid only if Bit 0 = 1.

- Bit 0: When this bit is a 1, it indicates that an interrupt for System Bus Slave protection is pending and that the System Bus Slave status bits and the LPATR are locked. Write this bit (using byte operation) to clear the interrupt.

Reset Conditions:

```
Reset: 0000 0000 0000 0000 0000 0UU0 0000 0UU0
```

Error Checking and Correction (ECC) Subsystem

4

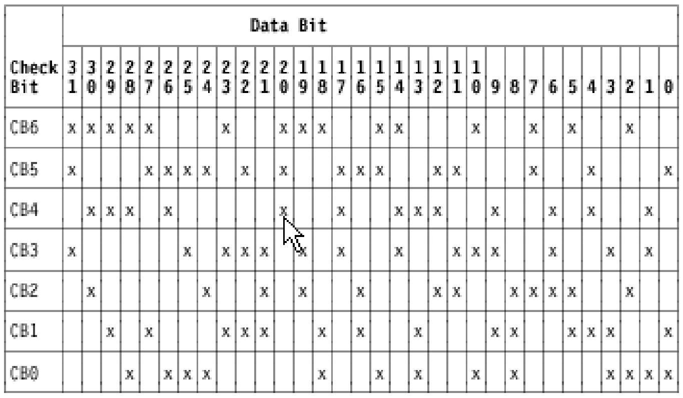
In addition to memory protection, the Memory Controller Chip implements an Error Detection and Correction code (EDC) to provide a one-bit correct, two-bit detect scheme. The data and check bits are grouped as shown below on both the packet and instruction DRAM on the co-processor platform. This grouping provides for chip-kill-detection on the instruction memory and SIMM-kill-detection on the packet memory, when multi-bit errors are detected. The "U" numbers represent the individual memory components that make up the instruction memory..

The following table illustrates that if instruction memory on an ARTIC960 adapter consists of SOJ modules, U2, U4, U6, and U7 will be populated when the instruction memory size equals 1MB.

		ARTIC960 Instruction Memory Components			ARTIC960 PCI Instruction Memory Components	
			SOJ Memory		SOJ Memory	
Data Bits	**Check Bits**	**Zip Memory**	**(1 MB)**	**(4MB)**	**(1 MB)**	**(4MB)**
D2:0	CB0	U2				
D5:3	CB1	U3	U2	U3	U4	U7
D8:6	CB2	U4				
D11:9	CB3	U5				
D14:12	CB4	U6				
D17:15	CB5	U7	U4	U5	U5	U8
D20:18	CB6	U8				
D27:24	---	U10	U6	U6	U3	U3
D31:28	---	U11	U7	U7	U2	U2
D23:21*	---	U9	U4	U5	U5	U8

* One bit of this chip is a spare bit.

The check bits are generated by XNORing the indicated data bits (similar to generating odd parity). The data bits used to generate each check bit are shown in Figure 4-1.



	Data Bit																														
Check Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CB6	x	x	x	x	x				x			x	x	x			x	x				x			x		x			x		
CB5	x				x	x	x	x		x		x			x	x	x			x	x				x			x				x
CB4		x	x	x		x						x			x			x	x	x			x			x		x			x	
CB3	x						x		x	x	x		x		x			x			x	x	x			x			x		x	
CB2		x						x			x		x			x				x	x			x	x	x	x			x		
CB1			x		x				x	x	x			x		x			x				x	x			x	x	x			x
CB0				x		x	x	x						x			x		x			x		x					x	x	x	x

Figure 4-1. Memory Controller Chip Check-Bit Generation

An *x* denotes that this data bit is part of the XNOR tree for that check bit.

The check bits are generated and written to memory at the same time as the data. When the location is read, the data and check bits flow through an XOR tree (same bits as generating logic plus the check bit).

ECC Errors

For single-bit errors, the Memory Controller Chip corrects the data on the fly (with no delay) going to the requesting device. Single-bit errors can be counted and an interrupt generated at terminal count (see *Single-Bit ECC Error Register (SBECCR)* on page 35). For multi-bit errors, the data may be scrambled. Multi-bit errors on 80960 accesses will cause an interrupt. Hardware scrubbing will not be performed. ECC errors can be forced (or turned off) using the FEER (see *Force ECC Error Register (FEER)* on page 36).

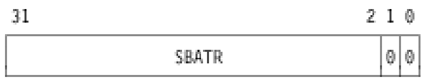
The following paragraphs describe the registers associated with ECC protection.

Single-bit ECC Error Address Trap Register (SBATR)

This register latches the address that caused a single-bit ECC error interrupt. The contents of this register are locked and valid until the interrupt is cleared.

Register Format

(ADDRESS = 1FFB 902Ch) read-only



Reset Conditions

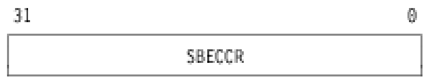
Reset: `UUUU UUUU UUUU UUUU UUUU UUUU UUUU UU00`

Single-Bit ECC Error Register (SBECCR)

This register is used to count the number of single-bit ECC errors detected by the Memory Controller Chip. The SBECCR holds an 8-bit value that is decremented each time a single-bit ECC error is detected by the Memory Controller Chip. When the count reaches zero, an interrupt is generated. To clear the interrupt, a non-zero value must be written to the SBECCR.

Register Format

(ADDRESS = 1FFB 9030h) read/write



FFh = Single-bit detection disabled (errors **are** corrected, but are not trapped or counted).

FEh – 01h = Number of errors to count before generating interrupt.

00h = Single-bit ECC interrupt is pending.

Reset Conditions

Reset: `0000 0000 0000 0000 0000 0000 1111 1111`

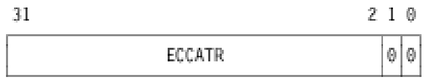
ROM Initialization: `0000 0000 0000 0000 0000 0000 0000 1010`

ECC Address Trap Register (ECCATR)

This register latches the address that caused the last multi-bit ECC error caused by an 80960 or task protection access. The register will be locked and valid until the interrupt is cleared by reading the ECCSTAT Register.

Register Format

(ADDRESS = 1FFB 9034h) read-only



Reset Conditions

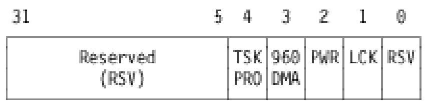
Reset: `UUUU UUUU UUUU UUUU UUUU UUUU UUUU UU00`

ECC Status Register (ECCSTAT)

This register latches the status associated with a multi-bit ECC error during an 80960 or task protection access. An interrupt is generated. Once the register has trapped status for a multi-bit error it is locked and will not trap any subsequent failures until it is read.

Register Format

(ADDRESS = 1FFB 9038h) read-only



Bit Descriptions

- Bits 31-5: Reserved; these bits must be set to 0.
- Bit 4: A 1 indicates the error occurred during an 80960 task protection cache-fill read.
- Bit 3: A 1 indicates the error occurred during an 80960 DMA transfer.
- Bit 2: Indicates whether the access was a processor read (0) or write (1).
- Bit 1: A 1 indicates this register is currently locked and a multi-bit interrupt is pending. It is cleared by a read operation.
- Bit 0: Reserved. This bit must be set to 0.

Reset Conditions

Reset: `0000 0000 0000 0000 0000 0000 UUUU UU00`

Force ECC Error Register (FEER)

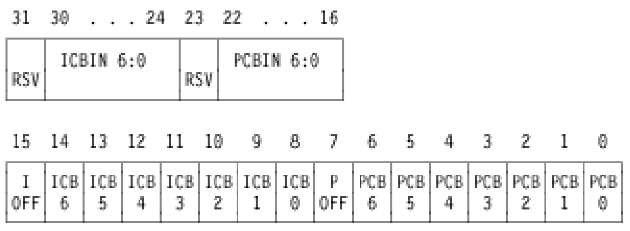
This register is used by **diagnostics** to force bad ECC codes to be written to memory. The FEER should be set to all 0s for normal operation. A 1 in any of the defined bits will cause the corresponding check bit to be inverted (wrong) when a write occurs to memory. For example, to force a single-bit error for data bit 14 (DB14) in packet memory, set PCB(6:0)='1011000'. If you now write 0000 0000h to any location in packet memory, it will read back 0000 4000h (bit 14 has been *corrected*). See Figure 4-2.

This register allows you to turn off ECC (the check bits are still written, but are not checked on reads); it also allows you to read the most recently read check bits..

Setting the PCB or ICB bits can be dangerous. Do not allow interrupts, stack usage, DMA, and so forth, while these bits are set. Also, after clearing these bits, rewrite the memory that was written while they were set.

Register Format

(ADDRESS = 1FFB A010h) read/write – byte addressable



Bit Descriptions

Bit 31: Reserved. This bit must be set to 0.

Bits 30 to 24: ICBIN 6:0. Instruction memory check bits from the most recent access to instruction memory. (Read-Only)

Bit 23: Reserved. This bit must be set to 0.

Bits 22 to 16: PCBIN 6:0. Packet memory check bits from the most recent access to packet memory. (Read-Only)

Bit 15: IOFF. A 0 = normal ECC; a 1 = ECC is off for instruction memory (no checking, no correction)

Bit 14 to 8: Force bad checkbits to be written on writes to instruction memory; see Figure 4-2.

Bit 7: POFF. A 0 = normal ECC; a 1 = ECC is off for packet memory (no checking, no correction)

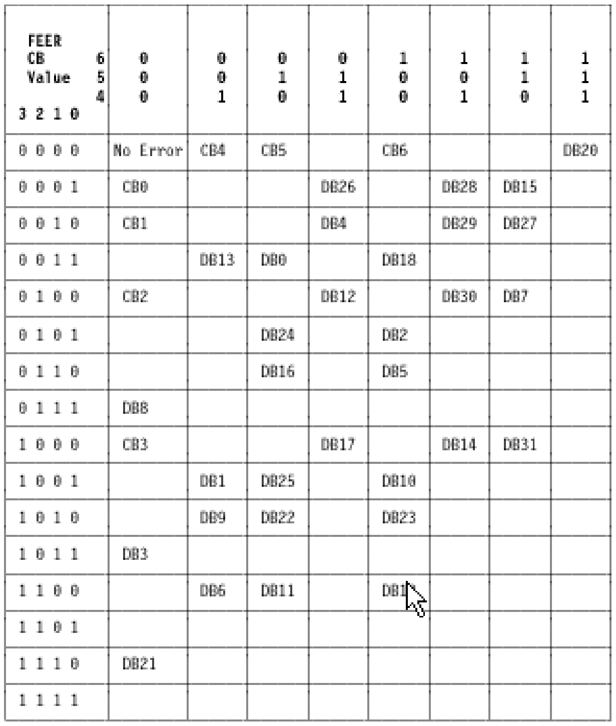
Bits 6 to 0: Force bad checkbits to be written on writes to packet memory; see Figure 4-2.

Reset Conditions

Reset: `0UUU UUUU 0UUU UUUU 0000 0000 0000 0000`

ROM Initialization: `0--- ---- 0--- ---- 0000 0000 0000 0000`

(where - equals read-only)



FEER CB Value 6 5 4 / 3 2 1 0	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0 0 0	No Error	CB4	CB5		CB6			DB20
0 0 0 1	CB0			DB26		DB28	DB15	
0 0 1 0	CB1			DB4		DB29	DB27	
0 0 1 1		DB13	DB0		DB18			
0 1 0 0	CB2			DB12		DB30	DB7	
0 1 0 1			DB24		DB2			
0 1 1 0			DB16		DB5			
0 1 1 1	DB8							
1 0 0 0	CB3			DB17		DB14	DB31	
1 0 0 1		DB1	DB25		DB10			
1 0 1 0		DB9	DB22		DB23			
1 0 1 1	DB3							
1 1 0 0		DB6	DB11		DB1[illegible]			
1 1 0 1								
1 1 1 0	DB21							
1 1 1 1								

Figure 4-2. Memory Controller Chip FEER Values

Each box shows the data bit (DB) or check bit (CB) that will appear to be in error. Blanks in the table indicate multi-bit errors. (Data with multi-bit errors will be unpredictable.)

This table can be used when designing memory tests. By setting the appropriate FEER Register bits, an error can be forced in either the packet memory or the instruction memory. The data read back from memory will indicate if the tested bit was *corrected*.

5

Timer Subsystem

The co-processor platform provides timer support through its integrated timer subsystem. Five hardware interval timers are provided (TMR0-4). Four of the timers (TMR0, 1, 2, 4) have a 1-millisecond interval and a 16-bit resolution to allow for timing events from 1 millisecond to 65 seconds. One of the timers (TMR3) is a 24-bit timer that uses a 333-nanosecond timing interval to allow for timing events from 333 nanoseconds to 6 seconds. (Actually, three ticks of timer TMR3 equal one microsecond.) Each timer’s function can be set up by programming the Timer Control Register (TCR) and by issuing timer-specific commands. The timers are described in the following sections of this chapter.

Programming Considerations

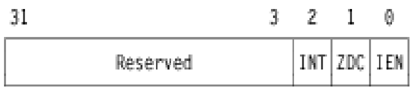
The Memory Controller Chip timers run off a free-running clock; therefore, the first tick could occur anywhere from 0 nanoseconds to 1 millisecond for Timers 0, 1, 2, or 4, and anywhere from 0 nanoseconds to 333 nanoseconds for Timer 3. For example, if Timer 0 is set to a preset value of 1, the zero count is reached after one tick of the clock. The actual time to this tick will be anywhere from 0 nanoseconds to 1 milliseconds.

Timer Control Register (TCR0-4)

Each timer has a Timer Control Register which specifies timer specific parameters.

Register Format

(ADDRESS = 1FFB x000h) read/write; x = timer number 0–4



Bit Descriptions

- Bits 31-3: Reserved. These bits must be set to '0'.
- Bit 2. Interrupt Status. (Read Only)

  This bit indicates the status of the interrupt for the counter.

  INT = 1: Interrupt is active.

  INT = 0: Interrupt is not active.

  If the interrupt is active, the interrupt will be reset when this register is read.
- Bit 1: Zero Detect Control.

This bit determines the action of the counter upon reaching zero count.

ZDC = 1: The timer stops counting without preset.

ZDC = 0: The timer is preset to the value contained in the Timer Preset Register (TPR) at the next clock edge, and continues counting.

- Bit 0: Interrupt Enable.

  IEN = 1: Enables an interrupt to occur when a zero count is reached.

  IEN = 0: Disables this interrupt.

Reset Conditions

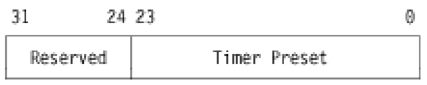
```
                Reset: 0000 0000 0000 0000 0000 0000 0000 0000
ROM Initialization: 0000 0000 0000 0000 0000 0000 0000 0000
```

Timer Preset Register (TPR0-4)

Each timer has a preset register which specifies the initial count value for the timer. The timer is actually a down counter that starts counting when the value of the TPR is loaded into the counter, and a start command is issued. A value of FFFFh causes the maximum of 65535 ticks to occur before zero count. A value of 0000h is not allowed. A value of 0001h causes 1 tick to occur before zero count. Interval timers TMR 0, 1, 2, and 4 are 16-bit timers, whereas TMR 3 is a 24-bit timer. See *Chapter 5, Timer Subsystem* for tick length.

Register Format

(ADDRESS = 1FFB x004h) read/write; x = timer number 0–4



Bit Descriptions

- Bits 31-24: Reserved. These bits must be set to 0.
- Bits 23-0: Timer Preset Value.

  This value is initially loaded by software and then can be reloaded automatically depending on how the TCR is programmed. Only bits 0 to 15 are active for interval timers TMR 0, 1, 2, and 4.

Reset Conditions

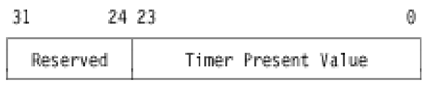
```
                Reset: 0000 0000 0000 0000 0000 0000 0000 0000
ROM Initialization: 0000 0000 0000 0000 0000 0000 0000 0000
```

Timer Present Value Register (TPV0-4)

Each timer can be read to determine its present count value at any time. This can be done without stopping the timer. Timers TPV 0, 1, 2, and 4 are 16-bit read only registers, whereas TPV 3 is a 24-bit read-only value.

Register Format

(ADDRESS = 1FFB x008h) read-only; x = timer number 0–4



Bit Descriptions

- Bits 31 to 24: Reserved. These bits must be set to 0.
- Bits 23 to 0: Timer Present Value.

  This value is latched during the read cycle and indicates the present count for the timer.

Reset Conditions

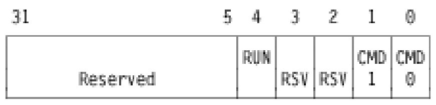
```
Reset: 0000 0000 0000 0000 0000 0000 0000 0000
```

Timer Command Register (TCMD0-4)

Each timer has a set of three defined commands to start, stop, and preset individual timers. Also, this register can be read to determine if the timer is currently running.

Register Format

(ADDRESS = 1FFB x00Ch) read/write x = timer number 0-4



Bit Descriptions

- Bits 31-5: Reserved. These bits must be set to 0.
- Bit 4: Timer Running (read-only).

  When this bit is set, it indicates the timer is running currently. When it is reset, it indicates the timer is stopped.
- Bits 3-2: Reserved. These bits must be set to 0.
- Bits 1-0: Command Bits 0 and 1.

  These two bits are binary encoded to provide four different commands.

  CMD0=0 & CMD1=0: Stop the timer.

  CMD0=1 & CMD1=0: Start a timer.

  CMD0=0 & CMD1=1: Reserved.

  CMD0=1 & CMD1=1: Preset a timer from its TPR and start.

Reset Conditions

```
               Reset: 0000 0000 0000 0000 0000 0000 0000 0000
ROM Initialization: 0000 0000 0000 0000 0000 0000 0000 0000
```

Timer Interrupts

Each of the five timers can generate an interrupt to the 80960 when a zero count has been reached. Each timer interrupts with a separate fixed interrupt vector detailed as follows.

Table 5-1. Timer Interrupt Vector Assignment

Timer #	Vector #		Function
	Dec	Hex	
0	255	FF	Watchdog
1	35	23	Software
2	34	22	Time of Day
3	33	21	Performance
4	32	20	Time Slice

The functions listed are for the ARTIC960 Kernel.

Interrupt Controller Subsystem 6

The general features of the interrupt controller subsystem are as follows:

- Four encoded interrupt inputs from the System Bus Interface Chip.
- Four encoded interrupt inputs from the Memory Controller Chip.
- Support for both vectoring and non-vectoring AIB devices.

The co-processor platform supports 80960 expanded mode interrupting. An 8-bit interrupt vector bus is provided to directly attach to the 80960. The co-processor platform interrupt controller takes the interrupt inputs and translates these into interrupt vectors that it places on the interrupt bus for interpretation by the 80960. The System Bus Interface Chip and the Memory Controller Chip provide four encoded bits to the interrupt controller, which are translated into an interrupt vector. The AIB provides eight separate bits in Direct mode, which the interrupt controller translates into an interrupt vector. Any combination of these eight bits can be used by the AIB.

Support also is provided to allow AIB devices to directly supply the 80960 with an interrupt vector. This is a programmable option, and is referred to as vectored mode. The interrupt controller manages prioritizing the interrupts that these devices generate. The priority is fixed with "single-bit ECC error" being the lowest and the NMI sources being the highest.

Interrupt inputs are not latched by the interrupt controller. The output of the interrupt controller to the 80960 interrupt bus reflects the value of the highest priority input active. Table 6-1 shows the priority scheme that is used to report various classes of interrupts back to the 80960.

Table 6-1. Interrupt Vector Assignment

	Vector Number			
80960 Priority Level	**Dec**	**Hex**	**Interrupt Condition**	**Clearing Register (See page)**
31	255	FF	Watchdog timeout	107
31	254	FE	Multi-bit ECC error	35
31	253	FD	System Bus NMI command	114
31	252	FC	Reserved for co-processor platform	–
31	251-249	FB to F9	Reserved for co-processor platform	–
31	248	F8	AIB Severe Error	See Notes 1 and 2
30	247-244	F7-F4	Reserved for co-processor platform	–
30	243	F3	LBUS exception	59
30	242	F2	LBUS parity error	59
30	241	F1	LBUS parity error with 80960 master	59
30	240	F0	AIB Error	See Notes 1 and 2
29	239 to 236	EF to EC	Reserved for co-processor platform	–
29	235	EB	Memory protect error with 80960 master	30
29	234	EA	Memory protect error with System Bus Slave	31
29	233	E9	Reserved for co-processor platform	–
29	232	E8	AIB	See Note 1
28	231-225	E7 to E1	Reserved for co-processor platform	–
28	224	E0	AIB	See Note 1
27	223 to 220	DF-DC	Reserved for co-processor platform	–
27	219 to 216	DB to D8	AIB	See Note 1
26	215 to 210	D7 to D2	Reserved for co-processor platform	–
26	209	D1	Serial Debug Port - receive	175
26	208	D0	Serial Debug Port - transmit	174
25 to 16	207 to 128	CF to 80	AIB	See Note 1
15	127 to 125	7F to 7D	Reserved for co-processor platform	–
15	124 to 123	7C to 7B	AIB	See Note 1
15	122 to 121	7A to 79	Reserved for co-processor platform	–
15	120	78	AIB	See Note 1
14	119 to 117	77 to 75	Reserved for co-processor platform	–
14	116	74	AIB	See Note 1
14	115 to 113	73 to 71	AIB	–
14	112	70	AIB	See Note 1
13 to 12	111 to 101	6F to 65	Reserved for co-processor platform	–
12	100	64	SCB attention port	107
12	99	63	Bus Master EOT	113

Table 6-1. Interrupt Vector Assignment

12	98	62	General Interrupt	113
12	97	61	Bus Master Channel 2	97
12	96	60	Bus Master Channel 1	97
11 to 10	95 to 84	5F to 54	Reserved for co-processor platform	–
10 to 8	83 to 64	53 to 40	AIB	See Note 1
7	63 to 59	3F to 3B	Reserved for co-processor platform	–
7	58	3A	AIB	See Note 1
7 to 4	57 to 36	39 to 24	Reserved for co-processor platform	–
4	35	23	Timer 1 (software)	39
4	34	22	Timer 2 (time of day)	39
4	33	21	Timer 3 (performance)	39
4	32	20	Timer 4 (time slice)	39
3 to 1	31 to 9	1F to 9	Reserved for co-processor platform	–
1	8	8	Single-bit ECC error	35

1. See AIB documentation for usage.
2. Two interrupt vectors (248 and 240) have been allocated for AIB error conditions. Vector 248 (F8h) should only be used to handle severe errors. The reason for this is that as a priority 31 interrupt (the highest available), this interrupt will be serviced before any other processor function, and also will interrupt the handling of another priority 31 interrupt. Vector 240 (F0h) should be used to handle error conditions only, as its priority is equivalent to exception and parity handling vectors assigned to the co-processor platform.
3. See *Enable/Detect Register (EDR)* on page 46 for a description of the Enable/Detect Register.

Programming Considerations

When the interrupt handler clears the interrupt at the source, it should allow for a delay from the time that this command is issued until the time that the interrupt is actually cleared and interrupts at that level are re-enabled. One way to accomplish this is to read any register on the Local Bus after issuing the command to clear the interrupt.

Figure 6-1 shows the interrupt flow on the co-processor platform.

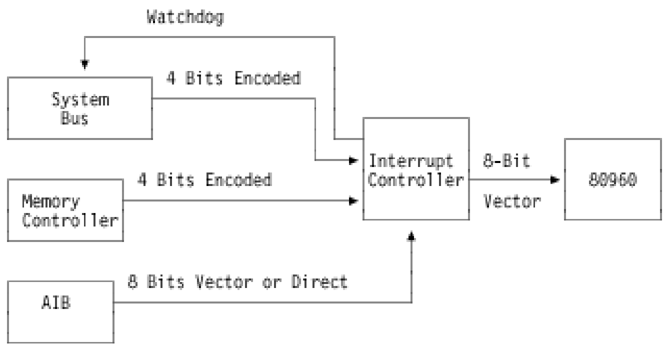


Figure 6-1. Co-Processor Platform Interrupt Controller

Enable/Detect Register (EDR)

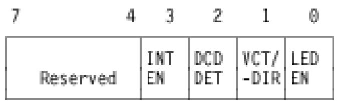
The EDR is used primarily to select the AIB interrupt mode, which may be vectored or direct. It also is used to:

- Enable interrupts from the AIB
- Indicate the presence of an AIB
- Control a light-emitting diode (LED).

A write to this register by the 80960 processor must be done using “byte” instructions or the results will be unpredictable.

Register Format

(80960 ADDRESS = 0A00 0004h) read/write



- Bits 7-4: Reserved.
- Bit 3: AIB Interrupt Enable.

  This bit is used to enable or disable interrupts from the AIB.

  0 = disabled

  1 = enabled.
- Bit 2: AIB Detect.

  This bit indicates whether or not an AIB is connected to the co-processor platform. If this bit reads 1, an AIB is present.
- Bit 1: VCT/-DIR (Vector/-Direct).

  This bit selects the AIB interrupt mode. When this bit is set to 1, the AIB provides an 8-bit, 80960 interrupt vector. This vector is passed through the interrupt controller to the 80960. When this bit is set to 0, each of the eight bits from the AIB is treated as a separate interrupt. The following table shows the vectors associated with the eight AIB interrupt lines when this bit is 0.
- Bit 0: LED Enable. This bit is used to enable or disable the LED.

  0 = disabled

  1 = enabled.

Reset Conditions

```
               Reset: UUUU 0S01
ROM Initialization: ---- nxn0
```

(where - = read-only; *n* = set by the AIB;

x = 0 for AIB not present; and *x* = 1 for AIB present)

CFE Local Bus Subsystem and AIB Interface Description

7

The CFE Local Bus is a 32-bit multiplexed address/data bus, operating at 25MHz. This bus attaches the Memory Controller Chip and the System Bus Interface Chip to the AIB Interface as shown in Figure 2-1. A separate ROM bus is also available at the AIB interface. This interface is designed to be general in nature, and allows a multitude of devices to be attached with minimal logic.

The CFE Local Bus and ROM bus signals are described in Table 7-1 through Table 7-2. The timing relationships of the CFE Local Bus signals are provided in the *ARTIC960 Co-Processor Platforms Application Interface Board Developer's Guide*. Pin assignments for the AIB interface connector are listed in Table 7-7.

Figure 8-3 shows how the host system address space is mapped into the Local Bus address space.

Throughout this manual, the CFE Local Bus is generally referred to as the Local Bus.

CFE Local Bus and ROM Bus Signal Descriptions.

In the following tables, outputs refer to signals driven by the co-processor, while inputs refer to signals received by the co-processor.

CFE Local Bus (AIB Interface) Signals

Descriptions of the CFE Local Bus signals are listed in Table 7-1.

Table 7-1. AIB Interface (CFE Local Bus) Signals

Abbrev	Dir	Description
L_AD31:0	I/O	*32-Bit Multiplexed Address/Data Bus.* These lines contain address information during the address cycle, after which they are used for data. Bit 0 has a special meaning during the address phase. If bit 0=1 and the Micro Channel Interface Chip is the owner, then the AIB protection will function. If bit 0=0, the AIB protection is disabled.
L_ADP3:0	I/O	*Local Bus Parity Bits.* When address or data is present on the AD bus, these lines will contain the odd parity of each byte. L_ADP0 corresponds to AD7:0, L_ADP1 to AD15:8, L_ADP2 to AD23:16, and L_ADP3 to AD31:24.
-L_BE3:0	I/O	*Byte Enables.* The CFE Masters drive these signals and latch them at address time as slaves. For writes, these signals select which bytes to write at the selected address. For reads, these signals determine which bytes are driven onto the data bus. These signals also determine which parity bits are valid.

Table 7-1. AIB Interface (CFE Local Bus) Signals

Abbrev	Dir	Description
-L_READY	I/O	*Ready.* For reads, this signal indicates that data is valid on the rising edge of PCLK. For writes, this signal indicates completion of the write cycle. The Memory Controller Chip and the Micro Channel Interface Chip drive this signal as a slave and monitor it as a master.
-L_ADS	I/O	*Address Strobe.* This signal indicates the beginning of a new address cycle. Address is on the AD bus during this cycle. The Memory Controller Chip and the Micro Channel Interface Chip drive this signal as a master and monitor it as a slave.
L_W/-R	I/O	*Write/Read.* This signal indicates whether a cycle is a write or read access. The Memory Controller Chip and the Micro Channel Interface Chip drive this signal as a master and monitor it as a slave.
-L_BLAST	I/O	*Burst Last.* This signal indicates the last transfer of a burst access. The Memory Controller Chip and the Micro Channel Interface Chip drive this signal as a master and monitor it as a slave. The Micro Channel Interface Chip does not support burst transfers as a slave.
-L_EXCPT	I/O	*Local Bus Exception.* The Memory Controller Chip and the Micro Channel Interface Chip drive this signal as a slave when a data parity error is detected. The Memory Controller Chip also drives this signal when a Local Bus timeout occurs or when a multibit ECC error occurs during a Local Bus access to memory. It will be received by the Memory Controller Chip and the Micro Channel Interface Chip when they are the Local Bus Masters to detect Local Bus exceptions.
-L_REQ1	I	*Local Bus Request 1.* This signal is used to indicate to the Memory Controller Chip that an AIB master is requesting the Local Bus.
-L_GRANT1	O	*Local Bus Grant1.* This signal indicates to an AIB local Bus Master that it has been granted the bus.
-MSTRREQ (-L_REQ2)	o	*Master Request*. This signal is driven by the System Bus Interface Chip to request use of the Local Bus for transfers to, and from, Bus Master Channels 1 and 2. This signal is not driven to the AIB interface connector.
-SLVEREQ (-L_REQ0)	O	*Slave Request.* This signal is driven by the System Bus Interface Chip to request use of the Local Bus for transfers to, and from, the Micro Channel slave buffers. This signal is not driven to the AIB Interface connector.
-MSTRACK (-L_GNT2)	I	*Master Acknowledge*. This signal is received by the System Bus Interface Chip to indicate that a Master Request has been granted. This signal is not driven to the AIB interface connector.
-SLVEACK (-L_GNT0)	I	*Slave Acknowledge*. This signal is received by the System Bus Interface Chip to indicate that a Slave Request has been granted. This signal is not driven to the AIB interface connector.
-WDOG	I	*Watchdog Timeout*. This signal is received by the System BUs Interface Chip asynchronously to detect watchdog timeout exceptions. This singal is not driven to the AIB interface connector.

AIB ROM Bus Signals

Descriptions of AIB ROM Bus signals are listed in Table 7-2.

Table 7-2. AIB ROM Bus Interface Signals

Abbrev	Dir	Description
-D_CD_ROM_CS	O	*AIB ROM Chip Select.* This line is active when the 80960 processor is accessing an address in the 192 to 224MB region. (Refer to the memory map in Table 2-12 for additional information.)
ROM_A15:0	O	*ROM Address Bus.* These lines are used to address the AIB ROM.
ROM_D7:0	I/O	*ROM Data Bus.* These lines are used to send data to, and from, the 80960 processor.
-ROM_OE	O	*ROM Output Enable.* This line is used to gate the data during a read cycle.
-ROM_WR	O	*ROM Write Enable.* This line is used to enable writes to the AIB ROM.

Other Interface Signals

Descriptions of miscellaneous AIB interface signals are listed in Table 7-3.

Table 7-3. Miscellaneous AIB Interface Signals

Abbrev	Dir	Description
-AIB DETECT	I	-*AIB Detect.* This line is driven low when an AIB is attached to the base adapter. The AIB is required to tie this signal to ground.
-CMD_RESET	O	*Power-On-Reset.* This line can be used to reset devices on the AIB to their power-up values.
PCLK_25MHz	O	*Clock Input.* This is the clock input to the AIB. It is driven by the PCLK output of the 80960 processor. Note: This signal must be terminated on the AIB.
-VINT7:0	I	*Vectored Interrupt 7:0.* These lines are used to signal interrupts to the Local Bus Interrupt Controller. See *Chapter 6, Interrupt Controller Subsystem* for more details.

Local Bus Memory Map

As viewed from the Local Bus, the memory map looks as shown in Table 7-4.

Table 7-4. Co-Processor Platform Memory Map (as viewed from the Local Bus)

From (Hex)	To (Hex)	Description
0000 0000	0FFF FFFF	Non-Burst. Address space is **not** shared with the 80960.
1000 0000	1FEF FFFF	Reserved Non-Burst. Address space is shared with the 80960.
1FF0 0000	1FF9 FFFF	Non-Burst. Address space is shared with the 80960. (AIB Register Address Space)
1FFA 0000	1FFA FFFF	Non-Burst. Address space is shared with the 80960. (System Bus Interface Chip Register Address Space)
1FFB 0000	1FFB FFFF	Non-Burst. Address space is shared with the 80960. (Memory Controller Chip Register Address Space)
1FFC 0000	1FFF FFFF	Reserved Non-Burst. Address space is shared with the 80960.
2000 0000 (See Note)	25FF FFFF	Reserved Burst RAM Area. Address space is shared with the 80960. In areas where DRAM is located, the 80960 accesses do not appear on the CFE Local Bus, and the Memory Controller Chip responds to the memory access. Packet memory can be located from 2000 0000h up to 21FF FFFFh, Instruction memory can be located from the 2200 0000h up to 23FF FFFFh.
2600 0000 (See Note)	2FFF FFFF	AIB Burst Relocation Area. (AIBs should locate their region on a boundary equal in size to the region itself.) Address space is shared with the 80960.
3000 0000	FFFF FFFF	Reserved Burst Area. Address space is **not** shared with the 80960.

.

This address map shows shared Bursting ranges with regard to a CFE Master. Ranges identified with this Note are Non-Bursting on the CFE Local Bus for 80960 accesses, but reside in the architected CFE Burst address range. Accesses from the System Bus Interface Chip in these ranges will be Burst accesses.

Local Bus Arbitration

There are four requesters for the Local Bus. Assuming all four are constantly requesting, the service order (in round-robin fashion) is:

- 80960 processor request
- REQ0 – System Bus Slave request
- REQ1 – AIB request
- REQ0 – System Bus Slave request
- REQ2 – System Bus Interface Chip Bus Master DMA request
- REQ0 – System Bus Slave request.

Note that an 80960 processor request is actually the 80960 processor starting an address cycle. All other requesters first request the bus, then are granted the bus, then start their address cycle. Since the 80960 processor is quickest to the bus, it is the default grantee if there are no other requesters.

A master will not be granted the bus twice in a row, unless there are no other requesters.

The System Bus Interface Chip arbitrates for the Local Bus in two ways: (1) as a slave from the System Bus interface or (2) as a Bus Master on Channels 1 and 2. By providing separate requests for master and slave activity, separate arbitration is allowed by a Local Bus arbiter, based on the activity requested. The Memory Controller Chip will request the Local Bus when the 80960 accesses an address from 1000 0000h to 7FFF FFFFh, except for the areas where the Memory Controller Chip's registers (1FFB XXXXh), packet memory, and instruction memory are located. AIB devices will request the Local Bus by driving -REQUEST<1>.

Preemption and Termination of Bus Ownership by the System Bus Interface Chip

The System Bus Interface Chip ownership of the Local Bus can be preempted by removing the active Acknowledge from the Local Bus.

The System Bus Interface Chip will detect the removal of Acknowledge, and remove the associated Request with the assertion of -L_BLAST, signaling the termination of its ownership. Ownership then is relinquished when the slave asserts -L_READY.

ARTIC960 Preemption and Termination

The assertion of another request or an 80960 processor access will cause a preemption to occur.

The time from recognizing the removal of Acknowledge to the removal of Request with -L_BLAST is variable, depending on: (1) which request is active, (2) the type of transfer, and (3) how many transfers have already occurred on the Local Bus. The number of transfers is a factor in termination because the Micro Channel Interface Chip guarantees a minimum number of transfers for each bus ownership. This number is 16 for -SLVEREQ accesses and 4 for -MSTRREQ. The time to relinquish ownership of the bus in number of transfers is given in Table 7-5.

Table 7-5. Local Bus Ownership Termination (number of transfers)

Request Archive	Type of Access	Xfers Prior (XP) to Acknowledge	Xfers Before (XB) Termination
-MSTRREQ	Read or Write	1 to 3	4 minus XP (See Note)
-MSTRREQ	Read or Write	≥4	1
-SLVEREQ	Read	1 to 15	16 minus XP (See Note)
-SLVEREQ	Read	>16	1
-SLVEREQ	Write	1 to 15	16 minus XP (See Note) Note: Where noted, the number of transfers before termination (XB) is equal to the number shown, minus the number of transfers prior (XP) to Acknowledge.

ARTIC960 PCI Preemption and Termination

The PCI System Bus Interface Chip provides an internal preempt counter that provides for the number of clocks before the PCI System Bus Interface Chip relinquishes ownership of the CFE Local Bus after preemption. This counter is programmable by setting the Preempt Latency bits (SEER, Bits 5–4).

By default, the PCI System Bus Interface Chip, when preempted, will complete a minimum of four transfers at 0 wait states before relinquishing the CFE Local Bus. Some transfers of eight or less words are not preemptable.

Local Bus Parity

The System Bus Interface Chip and the Memory Controller Chip support the generation and checking of data and address parity on the Local Bus. Address and data parity are always generated for the Local Bus. The Local Bus address and data parity bits are also sent to the AIB connector. The AIB ROM can set a flag in a RAM table to indicate if parity is supported on the AIB. If parity is not supported, the Base ROM will disable parity checking in the Memory Controller Chip and the System Bus Interface Chip.

The Memory Controller Chip's checking of address and data parity is enabled by setting Bit 0 in the LBCFG Register. See the *Local Bus Configuration Register (LBCFG)* on page 62. The System Bus Interface Chip's checking of address and data parity is enabled by setting Bit 9 in the PROC_CFG Register. See the *Processor Configuration Register (PROC_CFG) (Micro Channel)* on page 78 or *Processor Configuration Register (PROC_CFG) (PCI)* on page 80 for more details.

If the Memory Controller Chip detects a Local Bus parity error while doing a read, status will be captured in the MPE portion of the LEXSTAT Register. Under certain conditions, the address may be captured in the LEXATR Register. See the *Local Bus Exception Status Register (LEXSTAT)* on page 60 and the *Local Bus Exception Address Trap Register (LEXATR)* on page 59 for more details.

The detection of a Local Bus *address* parity error by the System Bus Interface Chip as a slave blocks the current transfer with no interrupt or status provided. If a Local Bus *data* parity error is detected by the System Bus Interface Chip as a Local Bus slave, the -L_EXCPT signal is asserted synchronous to the transfer.

If a Local Bus read data parity error is detected on the System Bus Interface Chip's DMA channel access of the Local Bus, the Bus Master access is terminated and the Local Bus Parity Bit is set in the Bus Master Status Register. The Bus Master termination interrupt for that channel also is asserted.

For address parity errors, the slave System Bus Interface Chip and the slave Memory Controller Chip ignore the address cycle and do not decode the address. This results in a Local Bus timeout, which is detected by the Memory Controller Chip. The Memory Controller Chip then drives the -L_EXCPT signal for the duration of the timeout.

The detection of Local Bus data parity on a System Bus read forces a Local Bus Parity⁄Exception interrupt to the Local Bus and the setting of Bit 0 in the Local Bus Parity⁄Exception Register. This bit, as well as the interrupt, is cleared by reading the Local Bus Parity⁄Exception Register.

Data for the failing transfer is not passed to the System Bus. On the Micro Channel bus, if currently streaming, streaming data is terminated on the transfer prior to the failing transfer in the prefetch buffer. If the Micro Channel master subsequently reaccesses the Micro Channel Interface Chip at the failing address, -CDCHRDY will be asserted. After a timeout of 3.4 microseconds, -CHCK will be asserted by the Micro Channel Interface Chip on the Micro Channel to signal the failing transfer. The assertion of -CHCK on erroneous data transfers is handled in this way to prevent the assertion of -CHCK for prefetched transfers not requested by the Micro Channel master.

The co-processor ROM disables the CD CHRDY timeout logic, (Bit 12, PROC_CFG) because accesses to instruction memory may cause CD CHRDY to be held longer than 3.4 microseconds. With the timeout logic disabled, the -CHCK signal will not occur. This can result in timeout errors on the Micro Channel during parity error or exception conditions.

More information about the CD CHRDY Timeout Disable is provided in the PCFG. On the PCI bus, a PCI Target Abort is issued for the failing transfer.

Local Bus Exceptions

When the Memory Controller Chip detects -L_EXCPT as a Local Bus Master, it will capture status in the MEX portion of the LEXSTAT Register. If the Memory Controller Chip drives -L_EXCPT, it will not generate an interrupt; however, it will latch status in the EXS portion of the LEXSTAT Register. The status indicates which device owned the Local Bus when the exception occurred, and the error condition. Therefore, when an AIB or the Micro Channel Interface Chip receives an exception, these devices should check to see the status in the LEXSTAT Register. If the Memory Controller Chip did not drive -L_EXCPT, the other device may have been the source. See the *Local Bus Exception Status Register (LEXSTAT)* on page 60 and the *Local Bus Exception Address Trap Register (LEXATR)* on page 59 for more information on exception capturing by the Memory Controller Chip.

The System Bus Interface Chip receives notification from two sources of Local Bus exceptions occurring remotely. These sources interrupt the System Bus Interface Chip by two separate exception lines:

- *Watchdog Timeout Exceptions* are received by the -WATCHDOG signal.
- *Local Bus Transfer Exceptions* are received by the -L_EXCPT signal.

These exceptions create different responses by System Bus Interface Chip:

- Watchdog Timeout Exceptions are asserted to the System Bus Interface Chip asynchronously. This exception forces an interrupt to the System Bus and sets the status in the Command Busy Status Port. More information on the handling of the Watchdog interrupt can be found in CBSP for Micro Channel adapters and in CBSP9 for PCI adapters.

  Note that detection of a Watchdog timeout resets all Bus Master activity without interrupt or status. A Watchdog timeout does not affect accesses from the PCI Bus to the PCI System Bus Interface Chip. The interrupt enable bit (EI) in the PROC_CFG register does not affect the assertion of Watchdog interrupts.

- Local Bus Transfer Exceptions are asserted synchronously to the System Bus Interface Chip Local Bus transfer. This exception indicates that a bus error (for example, write data parity error, ECC) has occurred on the current cycle. For Micro Channel read accesses, -CHCK is asserted on the Micro Channel during the failing cycle. For either System Bus, Bus Master, or DMA channel accesses with bus errors, the channel is terminated. Exception status is reported in the BMSTAT Register and a termination interrupt is asserted.

  The detection of the -L_EXCPT on an access to the System Bus Interface Chip read forces a Local Bus Parity/Exception interrupt to the Local Bus, and the setting of Bit 1 in the Local Bus Parity/Exception Register. This bit, as well as the interrupt, is cleared by reading the Local Bus Parity/Exception Register. Data for the failing transfer is not passed to the System Bus. On the Micro Channel bus, if currently streaming, streaming data is terminated on the transfer prior to the failing transfer in the prefetch buffer. If the Micro Channel master subsequently reaccesses the Micro Channel Interface Chip at the failing address, -CDCHRDY will be asserted. After a timeout of 3.4 microseconds, -CHCK will be asserted by the Micro Channel Interface Chip on the Micro Channel to signal the failing transfer. The assertion of -CHCK on erroneous data transfers is handled in this way to prevent the assertion of -CHCK for prefetched transfers not requested by the Micro Channel master.

  Note that this signal is also driven by the Micro Channel Interface Chip if a data parity error is detected on the Local Bus during a write to the Micro Channel Interface Chip as a Local Bus slave.

The co-processor ROM disables the CD CHRDY timeout logic, (Bit 12, PROC_CFG) because accesses to instruction memory may cause CD CHRDY to be held longer than 3.4 microseconds. With the timeout logic disabled, the -CHCK signal will not occur. This can result in timeout errors on the Micro Channel during parity error or exception conditions.

More information about the CD CHRDY Timeout Disable is provided in the *Processor Configuration Register (PROC_CFG) (Micro Channel)* on page 78.

For read access from the PCI Bus, data for the failing transfer is not passed to the PCI Bus. The PCI System Bus Interface Chip issues a PCI Target Abort for the failing transfer. For write accesses, the PCI System Bus Interface Chip issues a PCI Target Disconnect.

Local Bus Timeout

The Memory Controller Chip contains timer circuitry that is intended to prevent the Local Bus from hanging. This chip counts clocks during Local Bus cycles. The timer starts with an ADS or READY, and is reset by READY. If the 64 clocks are counted, the Memory Controller Chip causes a Local Bus Exception. This function can be disabled (see LTO bit, *Special Arbitration Register (SPAR)* on page 62.

Local Bus Exception Handling

The handling of Local Bus exception conditions discussed in the previous paragraphs, Local Bus Parity and the -L_EXCPT signal, are summarized in Table 7-6.

Table 7-6. Local Bus Exception Handling Summary

Function	**Local Bus Parity**	**-L_EXCPT**
	Interface Chip	
Micro Channel Slave Write	Not Applicable	• Sets Status in LBPE • Asserts Local Bus Parity/ Exception Interrupt
Micro Channel Slave Read	• Sets Status in LBPE • Asserts Local Bus Parity/Exception Interrupt • Asserts -CHCK on Micro Channel	• Sets Status in LBPE • Asserts Local Bus Parity/ Exception Interrupt • Asserts -CHCK on Micro Channel
PCI Bus Slave Write	Not Applicable	• Sets Status in LBPE • Asserts CFE Local Bus Parity/ Exception Interrupt • Executes Target Disconnect
PCI Bus Slave Read	• Sets Status in LBPE • Asserts CFE Local Bus Parity/Exception Interrupt • Executes Target Abort	• Sets Status in LBPE • Asserts CFE Local Bus Parity/ Exception Interrupt • Executes Target Abort
Bus Master Channel Read	• Stops channel • Sets Status in BMSTAT • Asserts Bus Master Termination Interrupt	• Stops Channel • Sets Status in BMSTAT • Asserts Bus Master Termination Interrupt
Bus Master Channel Write	Not Applicable	• Stops Channel • Sets Status in BMSTAT • Asserts Bus Master Termination Interrupt
Local Bus Slave Access	Asserts -L_EXCPT	Not Applicable
	Memory Controller Chip	
Local Bus Master Write to Memory Controller Chip	Data Written	• Sets Status in LEXSTAT • Latches Address in LEXATR • Asserts Local Bus Exception Interrupt • Terminates Cycle
Local Bus Master Read to Memory Controller Chip	Not Applicable	• Sets Status in LEXSTAT • Latches Address in LEXATR • Asserts Local Bus Exception Interrupt • Terminates Cycle

Table 7-6. Local Bus Exception Handling Summary

Local Bus Read by 80960	• Sets Status in LEXSTAT • Latches Address in LEXATR • Asserts Local Bus Parity Interrupt • Terminates Cycle	• Local Bus Data Transferred to 80960 • Sets Status in LEXSTAT • Latches Address in LEXATR • Terminates Cycle
Local Bus Write by 80960	Not Applicable	• Sets Status in LEXSTAT • Latches Address in LEXATR • Asserts Local Bus Exception Interrupt • Terminates Cycle

More information on the handling of the Watchdog interrupt can be found in the *Command Busy Status Port (CBSP)* on page 112.

Local Bus Registers

The following paragraphs contain detailed descriptions of the registers associated with the Local Bus. Register descriptions may be repeated, where necessary, to indicate differences between Micro Channel implementation and PCI Bus implementation.

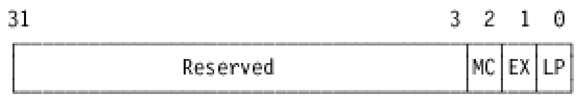
Local Bus Parity/Exception Register (LBPE)

The Local Bus Parity/Exception Register supplies status for Local Bus data parity or exception errors during a non-Bus Master channel access of the Local Bus. Reading this register clears the status. More information on Local Bus interrupts is available in *System Bus Interface Chip Interrupts* on page 115.

Micro Channel

Register Format

(Local Bus Address = 1FFA 0020h) — 32-bit read-only



Bit Descriptions

- Bits 31–3: Reserved. These bits must be set to 0.
- Bit 2: Micro Channel Data Parity Violation.

  This bit indicates a Micro Channel data parity error has been detected by the Micro Channel Interface Chip, while -CMD is active on the Micro Channel during a write to the Micro Channel Interface Chip as a slave. This error reporting is enabled by the Asynchronous Data Parity Condition Disable (POS3B, Bit 5). More information on this condition is provided in *Asynchronous Data Parity Check Description* on page 84.
- Bit 1: Exception.

This bit indicates that -L_EXCPT was received by the Micro Channel Interface Chip as a Micro Channel slave. Reading this register clears the status bit and the associated interrupt.

- Bit 0: Local Bus Parity.

  This bit indicates that a Local Bus data parity error has been detected as a Local Bus slave or as a Micro Channel slave. Reading this register clears the status bit and the associated interrupt.

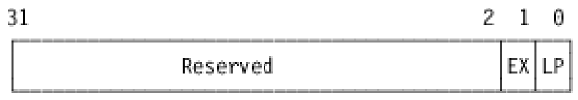
Reset Conditions

```
        Reset: 0000 0000 0000 0000 0000 0000 0000 0000
Command Reset: 0000 0000 0000 0000 0000 0000 0000 0000
```

PCI

Register Format

CFE Local Bus Address = 1FFA0020 h) — 32-bit read only



Bit Descriptions

- Bits 31–2: Reserved.
- Bit 1: Exception. This bit indicates that a CFE Local Bus exception (–L_EXCPT) was received by the PCI System Bus Interface Chip as a PCI target. Reading this register clears the status bit and the associated interrupt.
- Bit 0: Local Bus Parity. This bit indicates that a CFE Local Bus data parity error has been detected during an access to the System Bus Interface Chip as a PCI target. Reading this register clears the status bit and the associated interrupt.

Reset Conditions

```
         RST#: 0000 0000 0000 0000 0000 0000 0000 0000
Command Reset: 0000 0000 0000 0000 0000 0000 0000 0000
```

Local Bus Exception Address Trap Register (LEXATR)

This register latches the address that caused a local bus exception error (or parity error when Memory Controller Chip is the master). Once the register has trapped an address it is locked and will not trap any subsequent failures until the LEXSTAT Register is cleared (all three parts). The address is only valid if the corresponding bits in the LEXSTAT Register are valid, and until the LEXSTAT Register is cleared.

When trapping the address for a parity error (Memory Controller Chip master), the address trapped in the LEXATR will be the address of the error + 4h.

Register Format

(ADDRESS = 1FFB 9020h) read-only

Reset Conditions

```
Reset: UUUU UUUU UUUU UUUU UUUU UUUU UUUU UUUU
```

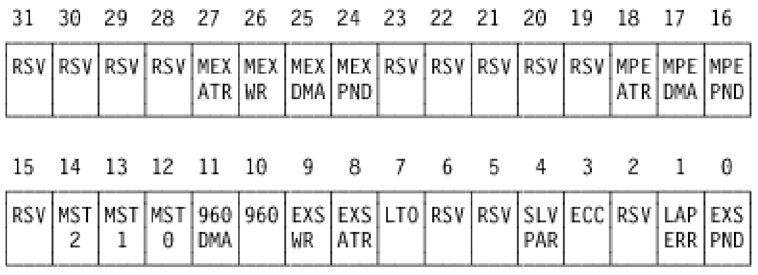
Local Bus Exception Status Register (LEXSTAT)

This register latches the status associated with a local bus exception (or a parity error when Memory Controller Chip is the master). The register is divided into three parts: MEX bits, MPE bits, and EXS bits. The MEX bits provide status for -L_EXCPT detected with the Memory Controller Chip master. The MPE bits provide status if the Memory Controller Chip detects a local bus parity error while doing a read operation. The EXS bits provide status for an exception sourced by the Memory Controller Chip. The three parts operate independently, but share the LEXATR. Once a part has trapped status, it is locked and will not trap any subsequent failures until the interrupt is cleared. If none of the parts were locked at the time of the error, the address will also be trapped in the LEXATR. *Status bits are valid only if the corresponding pending (PND) bit is set***.** To clear pending interrupts and unlock a part, a 1 must be written to the corresponding pending (PND) bit. Byte writes are **not** supported.

The EXS part does not correspond to a Memory Controller Chip interrupt. It is expected that the master that received the exception sourced an interrupt to the processor.

Register Format

(ADDRESS = 1FFB 9024h) read/write



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV	RSV	RSV	RSV	MEX ATR	MEX WR	MEX DMA	MEX PND	RSV	RSV	RSV	RSV	RSV	MPE ATR	MPE DMA	MPE PND

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV	MST 2	MST 1	MST 0	960 DMA	960	EXS WR	EXS ATR	LTO	RSV	RSV	SLV PAR	ECC	RSV	LAP ERR	EXS PND

Bit Descriptions

- Bits 31–28: Reserved. These bits must be set to 0.
- 24: MEX bits; -L_EXCPT was detected with Memory Controller Chip master.
  - Bit 27: If set, LEXATR contains the address of the MEX error.
  - Bit 26: A 0 = error on read; a 1 = error on write.
  - Bit 25: A 0 = error on regular 80960 access; a 1 = error on 80960 DMA access.

  - Bit 24: This interrupt is pending and these status bits are **valid** and locked. A 1 must be written to this bit to clear the interrupt and re-enable trapping.
- Bits 23–19: Reserved. These bits must be set to 0.
- Bits 18-16: MPE bits; Memory Controller Chip detected a local bus parity error on read.
  - Bit 18: If set, LEXATR contains the address of the MPE error.
  - Bit 17: 0 = error on regular 80960 access; a 1 = error on 80960 DMA access.
  - Bit 16: This interrupt is pending and these status bits are **valid** and locked. Write a 1 to this bit to clear the interrupt and re-enable trapping.
- Bit 15: Reserved. This bit must be set to 0.
- Bit 14-0: EXS bits; Memory Controller Chip sourced -L_EXCPT.
  - Bit 14: Indicates if the System Bus (Master 2) was the bus owner when the exception occurred.
  - Bit 13: Indicates if the AIB (Master 1) was the bus owner when the exception .
  - Bit 12: Indicates if the System Bus Slave (Master 0) was the bus owner when the exception occurred.
  - Bit 11: Indicates if 80960 DMA was the bus owner when the exception occurred.
  - Bit 10: Indicates if 80960 (non-DMA) was the bus owner when the exception .
  - Bit 9: 0 = error on read; 1 = error on write.
  - Bit 8: If set, LEXATR contains the address of the EXS error.
  - Bit 7: 1 = a local bus timeout occurred.
  - Bits 6–5: Reserved. These bits must be set to 0.
  - Bit 4: 1 = Memory Controller Chip detected a parity error as a slave.
  - Bit 3: 1 = Memory Controller Chip detected a multibit ECC error as a slave.
  - Bit 2: Reserved. This bit must be set to 0.
  - Bit 1: 1 = Memory Controller Chip detected an address parity error.
  - Bit 0: These status bits are **valid** and locked. Write a 1 to this bit to re-enable trapping. There is not a Memory Controller Chip interrupt associated with this bit. It is expected that the master receiving the exception sourced an interrupt.

Reset Conditions:

```
Reset: UUUU UUU0 UUUU UUU0 UUUU UUUU UUUU UUU0
```

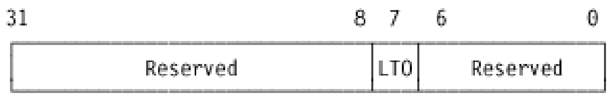
Special Arbitration Register (SPAR)

- This register is used to set the Memory Controller Chip’s Local Bus Timeout function.

The co-processor only supports bit 7 of this register.

Register Format:

(ADDRESS = 1FFB A008h) 32-bit read/write



Bit Descriptions:

- Bits 31–8: Reserved. These bits must be set to 0.
- Bit 7: Local Bus Timeout

  Set to 1 to disable the Memory Controller Chip’s Local Bus timeout function. See *Local Bus Timeout* on page 56.
- Bits 6–0: Reserved. These bits are not supported on the co-processor and must be set to 0.

Reset Conditions:

```
             Reset: 0000 0000 0000 0000 0000 0000 0000 0000
ROM Initialization: 0000 0000 0000 0000 0000 0000 0000 0000
```

Processor Configuration Register (PROC_CFG)

Micro Channel

The Processor Configuration Register is used to enable parity checking on the Local Bus, as well as enabling/disabling the CD CHRDY timeout logic and the Performance Timer. See *Processor Configuration Register (PROC_CFG) (Micro Channel)* on page 78 for detailed information.

PCI

The PROC_CFG register is initialized by the POST code to configure some of the hardware features of the PCI System Bus Interface Chip. See *Processor Configuration Register (PROC_CFG) (PCI)* on page 80 for detailed information.

Local Bus Configuration Register (LBCFG)

This register specifies how certain Local Bus functions operate.

Register Format:

(ADDRESS = 1FFB A00Ch) read/write – byte addressable

Bit Descriptions:

- Bits 31–13: Reserved. These bits must be set to 0.
- Bit 12: Bad Address/Data Parity.

  This bit determines whether BP bits affect data (BAPAR = 0) or address (BAPAR = 1).
- Bits 11–8: Force Bad Parity.

  The Memory Controller Chip will force Local Bus parity errors on the corresponding byte when it is driving address (BAPAR = 1) or when it is driving data (BAPAR = 0) – – Memory Controller Chip master write or slave read; that is, BP3 = 1 causes parity error on byte 3.
- Bits 7–1: Reserved. These bits must be set to 0.
- Bit 0: Parity Check.

  This bit specifies whether or not the Memory Controller Chip will do parity checking (0 = do not check parity; 1 = check parity).

Reset Conditions:

```
             Reset: 0000 0000 0000 0000 0000 0000 0000 0000
ROM Initialization: 0000 0000 0000 0000 0000 0000 0000 000X
(where X = parity checking; it reflects the status of the LBUS
parity bit in the BaseOptions structure)
```

AIB Connector Pin Assignments

The pin assignments for the 140-pin AIB connector are listed in Table 7-7.

Table 7-7. AIB Connector Pin Assignments

Pin	Signal Name	Pin	Signal Name	Pin	Signal Name
001	ROS A(13)	051	L_AD(31)	101	-L_ADS
002	ROS A(10)	052	L_AD(26)	102	VCC
003	ROS A(7)	053	GND	103	-L_EXCEPT
004	GND	054	L_AD(23)	104	Reserved
005	ROS A(2)	055	L_AD(18)	105	Reserved
006	ROS D(7)	056	GND/G	106	GND
007	VCC	057	-12V_TAB	107	ROS A(12)
008	ROS D(2)	058	+12V_PRG_PWR	108	ROS A(9)
009	-ROS_OE	059	VCC	109	VCC
010	GND	060	L_AD(13)	110	ROS A(4)
011	-V_INT(6)	061	L_AD(11)	111	ROS A(1)
012	-V_INT(3)	062	GND	112	GND
013	GND	063	L_AD(2)	113	ROS D(4)
014	-L_BE(1)	064	L_AD(0)	114	ROS D(1)
015	L_PAR(3)	065	GND	115	GND
016	VCC	066	-L_RDY	116	-ROS_WR

Table 7-7. AIB Connector Pin Assignments

Pin	Signal Name	Pin	Signal Name	Pin	Signal Name
017	L_AD(29)	067	-CMD_RESET	117	-D_CD_DET
018	L_AD(27)	068	VCC	118	VCC
019	GND	069	Reserved	119	-V_INT(1)
020	L_AD(21)	070	Reserved	120	-V_INT(0)
021	L_AD(19)	071	ROS A(15)	121	GND
022	-12V_TAB	072	GND	122	L_PAR(1)
023	+12V_PRG_PWR	073	ROS A(8)	123	L_PAR(0)
024	L_AD(15)	074	ROS A(6)	124	GND
025	VCC	075	VCC	125	L_AD(25)
026	L_AD(10)	076	ROS A(0)	126	L_AD(24)
027	L_AD(7)	077	ROS D(6)	127	VCC
028	GND	078	GND	128	L_AD(17)
029	PCLK_25MHz	079	ROS D(0)	129	L_AD(16)
030	GND	080	-D_CD_ROS_CS	130	GND
031	-L_BLAST	081	GND	131	L_AD(9)
032	-GNT(1)	082	-V_INT(7)	132	L_AD(8)
033	Reserved	083	-V_INT(4)	133	GND
034	Reserved	084	VCC	134	L_AD(4)
035	Reserved	085	-L_BE(2)	135	L_AD(3)
036	ROS A(14)	086	-L_BE(0)	136	VCC
037	ROS A(11)	087	GND	137	L_W/-R
038	GND	088	L_AD(30)	138	-REQ(1)
039	ROS A(5)	089	L_AD(28)	139	Reserved
040	ROS A(3)	090	GND	140	Reserved
041	VCC	091	L_AD(22)		
042	ROS D(5)	092	L_AD(20)		
043	ROS D(3)	093	VCC		
044	GND	094	L_AD(14)		
045	-V_INT(5)	095	L_AD(12)		
046	-V_INT(2)	096	GND		
047	GND	097	L_AD(6)		
048	-L_BE(3)	098	L_AD(5)		
049	L_PAR(2)	099	GND		
050	VCC	100	L_AD(1)		

Micro Channel Interface Subsystem

8

The co-processor platform Micro Channel interface subsystem is contained within the Micro Channel Interface Chip. The major functions of this chip are highlighted below:

- Two Bus Master channels, addressable from the Local Bus
- One 128-byte intermediate data buffer per channel
- Linked List Chaining support for auto-initialization of either channel
- Support of Micro Channel basic transfers as master and slave
- Support of 64-bit, 100 ns streaming data as master and slave
- Slave write buffer (192 bytes) for better Micro Channel utilization
- Slave prefetch read buffer (32 bytes) for better Micro Channel utilization
- Hardware support for Appended I/O operations
- Hardware support for SCB Move mode
- Support of access to resident memory as both I/O and shared memory window
- Micro Channel data and address parity support
- Support for Micro Channel interrupts and error reporting
- Directly attachable to Micro Channel with 24 mA off-chip drivers
- I/O mapping for testing of in-circuit connections

The following information is described in this chapter:

- Micro Channel Addressable Registers
- Program Option Select (POS)/Configuration/Subaddressing
- I/O Check Reporting and Parity
- Asynchronous Data Parity Check
- Bus Master Channel
- Slave Memory
- Slave I/O
- Subsystem Control Block Support (SCB)
- Miscellaneous Micro Channel Functions
- Micro Channel Interface Chip Interrupts

Micro Channel Interface Chip Signal Description

The following tables present a description of each signal in the Micro Channel Interface Chip.

Table 8-1. Micro Channel Interface Chip Pin Description – Micro Channel Signals (Alphabetical Order)

Name	Type	Description
A31-A0	I/O	Address bits 31-0 are used by the Bus Master to address memory and I/O slaves. These signals are also used to select the Micro Channel Interface Chip chip for slave operations.
-ADL	I/O	-Address Decode Latch is used by the slave interface to latch the Micro Channel address during a Micro Channel transfer. This signal is driven by the Bus Master interface as an address latching signal for a Micro Channel slave.
APAR3-APAR0	I/O	Address Parity Bits 3-0 are used by the Bus Master to generate address parity on the Micro Channel. They are used by the slave interface to check address parity on the Micro Channel.
-APAREN	I/O	-Address Parity Enable is used by the slave interface to detect the presence of valid address parity on the Micro Channel. This signal is driven by the Bus Master interface to indicate the presence of valid address parity on the Micro Channel.
ARB3-ARB0	I/O	Arbitration Bus is used by the Bus Master interface during arbitration to present its arbitration level to the Micro Channel.
ARB/-GNT	I	Arbitrate/-Grant is used by the Bus Master interface to determine if the Micro Channel is in an arbitration state or if the bus has been granted.
-BE3 - -BE0	I/O	-Byte Enables 3-0 are used by the slave interface to detect which bytes are valid in the 32-bit Micro Channel data bus. This signal is driven by the Bus Master interface to indicate which bytes are valid on the 32-bit Micro Channel data bus.
-BURST	I/O	-Burst is driven by the Bus Master interface to indicate to the bus that the master will perform one or more consecutive data transfer cycles.
CD CHRDY	O	Channel Ready is driven inactive by the slave interface to allow additional time to complete the current data transfer.
-CD DS 16	O	-Card Data Size 16 is driven by the slave interface together with -CD DS 32 to indicate a 16-bit or 32-bit data port.
-CD DS 32	O	-Card Data Size 32 is driven by the slave interface together with -CD DS 16 to indicate a 32-bit data port.
-CD SETUP	I	-Card Setup is used together with A2-A0 to select the POS registers.
-CD SFDBK	O	-Card Selected Feedback is driven by the slave interface to acknowledge that it has received a valid address decode.
-CHCK	I/O	-Channel Check is used by the slave interface to notify the current Micro Channel Bus Master that an exception condition has occurred during the current transfer. This signal is used by the Bus Master interface to detect an exception condition on the Micro Channel.

Table 8-1. Micro Channel Interface Chip Pin Description – Micro Channel Signals (Alphabetical Order)

CHRDYRTN	I	Channel Ready Return is received by the Bus Master interface to detect the need for additional time to complete the current data transfer.
CHRESET	I	Channel Reset is used to reset Micro Channel Interface Chip. from the Micro Channel.
-CMD	I/O	-Command is used by the Bus Master interface to define when data is valid on the Micro Channel for a basic transfer cycle. This signal indicates to the slave interface how long data is valid.
D31-D0	I/O	Data bits 31-0 are used to transfer data between Micro Channel Interface Chip and the Micro Channel.
DPAR3-DPAR0	I/O	Data Parity Bits 3-0 are used by both the slave and Bus Master interface to generate and check data parity on the Micro Channel.
-DPAREN	I/O	-Data Parity Enable is used by the transmitting interface (master or slave) to indicate the presence of valid data parity on the Micro Channel. This signal is used by the receiving interface (master or slave) to detect the presence of valid data parity on the Micro Channel.
-DS 16 RTN	I	-Data Size 16 Return is received by the Bus Master interface to detect that a 16-bit data port has been selected.
-DS 32 RTN	I	-Data Size 32 Return is received by the bus master interface to detect that a 32-bit data port has been selected.
-IRQ A,B,C,D	O	-Interrupt Requests a,b,c,d One of these signals is used by Micro Channel Interface Chip. to interrupt the Micro Channel. The Interrupt Request signal is selected in POS.
MADE 24	I/O	Memory Address Enable 24 is used by the slave interface to distinguish a Micro Channel address below 16MB. This signal is driven by the Bus Master interface to indicate that the current Micro Channel address is below 16MB.
M/-IO	I/O	Memory/-Input Output is used by both the slave and Bus Master interfaces to distinguish memory or I/O cycles on the Micro Channel.
-MSDR	I/O	-Multiplexed Streaming Data Request is driven by the slave interface to indicate the ability to do 64-bit streaming data. This signal is received by the Bus Master interface to detect the ability to do 64-bit streaming data.
-PREEMPT	I/O	-Preempt is used by the Bus Master interface to request an arbitration cycle on the Micro Channel.
-REFRESH	I	-Refresh is used to indicate that a memory refresh operation is in progress on the Micro Channel.
-S0,-S1	I/O	-Status is used by the Bus Master interface to indicate a Bus Master read or write on the Micro Channel. Status is used by the slave interface to determine whether a read or write cycle is taking place on the Micro Channel.
-SBHE	I/O	-System Byte High Enable is used by the Bus Master interface to indicate and enable transfers on D8-D15 of the Micro Channel. This signal is used by the slave interface to determine whether data is enabled on D8-D15 of the Micro Channel.

Table 8-1. Micro Channel Interface Chip Pin Description – Micro Channel Signals (Alphabetical Order)

-SDR(1,0)	I/O	-Streaming Data Requests 1,0 are used by the slave interface to request a streaming data transfer. These signals are received by the Bus Master interface to detect a request for streaming data.
-SD STROBE	I/O	-Streaming Data Strobe is received by the slave interface to clock data on and off the data bus during streaming data transfers. This signal is driven by the Bus Master interface.
-SFDBKRTN	I	-Selected Feedback Return is received by the Bus Master interface to detect that a slave device has been selected.
TR 32	O	Translate 32 is driven inactive by the Bus Master interface to indicate that the Bus Master interface is performing data steering.

Micro Channel Addressable Ports/Registers

The I/O map of the Micro Channel addressable registers is shown in Table 8-2. Detailed information for each register is located on the referenced "See Page".

Table 8-2. Micro Channel Addressable Ports/Registers

Abbreviation	See Page	Micro Channel Address (hex)	Width	Type	Local bus Address (hex)	Data Field Size	Type
COMMAND	107	00-03	32	R/W	1FFA 2000	32	R
ATTN	107	04	8	R/W	1FFA 2004	8	R
SCP	110	05	8	R/W	1FFA 2008	8	R
ISP	111	06	8	R	1FFA 200C	8	R/W
CBSP	112	07	8	R	1FFA 2010	8	R/W
SIR	113	–	–	–	1FFA 2014	16	R/W
GAID	169	0A	8	R	1FFA 0008	8	R
NMI	114	0B	8	R/2	1FFA 001C	1	R
HSBR	102	0C-0F	32	R/2	–	–	–
MDATA	103	10-13	32	R/2	–	–	–
CONF1	77	14-17	32	R	–	–	–
CONF2	77	18-1B	32	R	–	–	–
CONF3	78	1C-1F	32	R	–	–	–

Program Option Select (POS)/Configuration/Subaddressing

POS Registers

The co-processor platform implements eight POS registers and two POS subaddress registers to support automatic configuration. The POS registers are read/write accessible through POS addresses defined for the Micro Channel. In addition, the POS registers are mapped to both the Local/CFE Bus address space and the Micro Channel I/O space. These mappings allow access to POS information for the resident processor and the Micro Channel device driver. POS0 – 1 directly map to the CRDID Register. POS2 – 5 directly map to the POS_SETUP1 Register. POS3A and 3B map to the POS_SETUP2 Register.

Register Format

POS0	POS1	POS2	POS3	POS4	POS5	POS6	POS7	POS3A	POS3B	
		CE	B0	CKE	MW0	SA8 (IBE)	SA0	A24	EN	Low-Order Bit
C	C	W1	B1	A20	MW1	SA9 (CTO)	SA1	A25	AL0	
A	A	R0	B2	A21	MW2	SA10(XST)	SA2	A26	AL1	
R	R	R1	B3	A22	I13	SA11(SDP)	SA3	A27	AL2	
D	D	R2	AS	A23	I14	SA12(BDP)	SA4	A28	AL3	
		R3	PE	A29	I15	SA13	SA5	I10	ADP	
I	I	L0	SF	A30	CS	SA14	SA6	I11	RSV	
D	D	L1	SE	A31	CK	SA15	SA7	I12	RSV	High-Order Bit

- POS 0: Card ID Byte (CRDID). This byte will read ECh.
- POS 1: Card ID Byte (CRDID). This byte will read 8Fh.
- POS 2: CE = Card Enable

  W1 = BIOS Window size

  R0-R3 = ROM/RAM Address Select

  L0,L1 = Interrupt Level
- POS 3: B0-B3 = ARB Level

  AS = Asynchronous/Synchronous Channel Check Enable

  PE = Parity Enable

  SF = Selected Feedback Return Enable

  SE = Streaming Data Enable
- POS 4: CKE = Channel Check Enable

  A20-A23,A29-A31 = Address Bits
- POS 5: MW2-MW0 = Memory Window Size Bits

  I13-I15 = I/O Address Assignment

  CS = Channel Check Status

  CK = Channel Check Indicator
- POS 6: SA15-SA8 = Subaddress Bits
- POS 6 (Status):

  IBE = Invalid Byte Enables

  CTO = Channel Ready Timeout

  XST = Extra Streaming Data Strobes

  SDP = Streaming Data Parity

  BDP = Basic Transfer Data Parity
- POS 7: SA7-SA0 = Subaddress Bits
- POS 3A: A28-A24 = Address Bits

  I10-I13 = I/O Address Assignment

- POS 3B: EN = Arbitration Level Enable

  AL3-AL0 = Second Arbitration Level

  ADP = Asynchronous Data Parity Condition Disable

Programming Considerations

Streaming data is configurable through POS 3, and through the PROC_CFG Register. The data is available in 32-bit or 64-bit (multiplexed) format. The co-processor platform ROM disables the multiplexed streaming data bit in the PROC_CFG Register, so the application code must enable this bit if 64-bit streaming is to be used. POS 3 Bit 7 must also be enabled if either 32-bit or 64-bit streaming is to be used.

When 64-bit streaming is used, the transfers must be 4-byte aligned; that is, the Card Address Register (CAR) and System Address Register (SAR) must have the least two significant address bits set to 0.

When back-to-back memory accesses of the same location (write-read) are performed thorough the memory window, it is possible that the data may be corrupted because the read of the location may be performed before the posted write has completed. To avoid this situation, a delay should be added between the write and the read. A one-millisecond delay should be sufficient to prevent this situation from occurring; however, a shorter delay may be possible..

The following register descriptions are of those registers that are typically accessed by the 80960 processor. Some of these registers also show a Micro Channel I/O address where the same register can be accessed. However, the usual operational mode is for the 80960 processor to control the parameters in these registers. For a description of registers that are typically accessed by the system unit, see *Micro Channel Addressable Ports/Registers* on page 68.

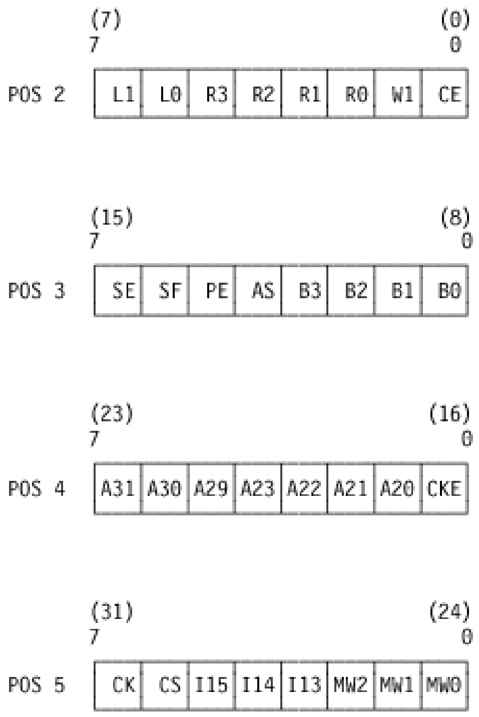
POS_Setup1 Register (POS_SETUP1)

The POS_SETUP1 Register is read accessible by the resident processor and read/write accessible by the Micro Channel. The register is a direct mapping of POS Registers 2–5. Normal operation is for POS to configure the register.

Register Format

(Local Bus Address = 1FFA 0000h)   32-bit read-only

(POS Addresses = 2-5) 8-bit read/write



Bit Descriptions

POS 2:

- Bits 7–6: Interrupt Level. These bits determine the interrupt level in which the Micro Channel Interface Chip interrupts the Micro Channel.

L1 L0	Interrupt Level
0 0	14
0 1	10
1 0	9
1 1	7

- Bits 5–2: ROM/RAM Address Select. These bits are used with the Window Size to place an 8KB shared memory window in one of 16 locations in the Micro Channel

ROM/RAM area, C0000h to DFFFFh. More information on memory window options can be found in the *Shared Memory* on page 100.

R3	R2	R1	R0	Base Address
0	0	0	0	C0000h
0	0	0	1	C2000h *
0	0	1	0	C4000h
0	0	1	1	C6000h *
0	1	0	0	C8000h
0	1	0	1	CA000h *
0	1	1	0	CC000h *
0	1	1	1	CE000h
1	0	0	0	D0000h *
1	0	0	1	D2000h *
1	0	1	0	D4000h
1	0	1	1	D6000h *
1	1	0	0	D8000h
1	1	0	1	DA000h *
1	1	1	0	DC000h
1	1	1	1	DE000h *

* 8KB window only

- Bit 1: BIOS Window Size. This bit selects the Micro Channel BIOS window size. Setting this bit selects a 16KB window; resetting this bit selects an 8KB window. Selecting a 16KB window precludes selecting the starting address at the values marked with an asterisk above. More information on memory window options can be found in the *Shared Memory* on page 100.
- Bit 0: Card Enable. This bit, when reset, disables the Micro Channel Interface Chip on the Micro Channel. When disabled, the Micro Channel Interface Chip only responds to POS accesses. This bit powers up reset.

Reset Conditions

```
        Reset: 0001 0010
Command Reset: SSSS SSSS
```

POS 3

- Bit 7: Streaming Data Enable. This bit, when set, enables the Micro Channel Interface Chip for streaming data as both a Bus Master and a slave. This bit powers up reset. This function should be enabled only if the host system supports streaming data. Typically, this bit should be *disabled* for OS/2 and *enabled* for AIX.
- Bit 6: Selected Feedback Return Enable. This bit, when set, enables the checking and reporting of loss of Selected Feedback Return by the Bus Master. This bit powers up reset. This function determines if a Bus Master should check the Selected Feedback Return Signal. This function should be enabled only if the system and all the slaves support the selected feedback signal. Typically, this bit should be *disabled* for OS/2 and *enabled* for AIX.
- Bit 5: Parity Enable. This bit, when set, enables the address and data parity checking and generation on the Micro Channel. This bit powers up reset. This function should

be enabled only if the host system supports parity checking/generation. Typically, this bit should be *disabled* for OS/2 and *enabled* for AIX.

- Bit 4: +Asynchronous/-Synchronous Bit. This bit, when set, enables the Micro Channel Interface Chip for asynchronous channel checks. When reset, the Micro Channel Interface Chip is set for synchronous reporting. Note that address and data parity exceptions are not affected by this bit. These exceptions always generate a synchronous channel check. This bit powers-up set, or in asynchronous mode. Note that using asynchronous channel check can cause a momentary suspension of all I/O activity, which could require a system reset. This bit should typically be set to the *asynchronous* mode.
- Bits 3–0: Arbitration Level. These bits determine the primary Bus Master arbitration level. They are binary encoded with B3 MSB and B0 LSB. In a PS/2, the primary arbitration level is used by the co-processor platform for bus master DMA transfers. In a RISC System/6000, the primary arbitration level is used for system unit-to-adapter DMA transfers.

  A second arbitration level, AL3–AL0, can be set in POS 3B using POS subaddressing. In a PS/2, this function is generally not used. In a RISC System/6000, the secondary arbitration level is used for adapter-to-adapter transfers.

Reset Conditions

```
        Reset: 0001 1100
Command Reset: SSSS SSSS
```

POS 4

- Bits 7–1: Address Bits. These bits are used to place the starting address of a second shared memory window if all of the adapter’s memory is to be mapped to the Micro Channel. Address bits A28–A24 may be set in POS3A. If the use of POS3A is not supported, these bits are assumed to be set to 0. This allows the starting address of the window to exist on any 1MB boundary in the lower 16MB of each 512MB region of the 4GB address space of the Micro Channel. The window must be placed on a boundary equal to the memory size. Therefore, a 4MB memory window must be on a 4MB boundary. More information on memory window options can be found in *Shared Memory* on page 100.
- Bit 0: Channel Check Enable. This bit is set to enable channel check for sources other than Micro Channel address and data parity.

Reset Conditions

```
        Reset: UUUU UUU0
Command Reset: SSSS SSSS
```

POS 5

- Bit 7: Channel Check Indicator. This bit powers up set to 1. This bit is reset by the assertion of -CHCK by the Micro Channel Interface Chip as a slave. The bit can be reset to 1 by writing a 1 from the Micro Channel. A write of 0 to this bit is ignored, That is, the bit remains a 1, and -CHCK is not asserted.

- Bit 6: Channel Check Status Indicator. This bit indicates that channel check status is available in POS 6. This bit powers up set, and is reset only for channel check conditions. This bit is read-only from both the Micro Channel and the Local Bus registers. This bit is reset to a 1 by resetting the Channel Check Indicator (Bit 7) to 1.
- Bits 5–3: I/O Address Assignment. These bits are used to locate the Micro Channel I/O base address on any 8KB boundary within the 64KB Micro Channel I/O address space. The bits correspond to the three most significant bits of the Micro Channel I/O address space, with I15 = I/O address bit 15 and I13 = I/O address bit 13.

  These bits are used together with POS 3A, Bits 7–5, to locate the I/O address on any 1KB boundary. More information on these additional bits is located in the *POS Setup Register 2 (POS_SETUP2)* on page 74.
- Bits 2–0. Memory Window Size Bits. These bits are written in the PROC_CFG Register by ROM, indicating the size of the memory window. These bits are read-only from POS 5. The intent of the POS mapping is to make this information available to setup routines that can only view the POS registers.

MW2	MW1	MW0	Memory Size
0	0	0	512 KB
0	0	1	1 MB
0	1	0	2 MB
0	1	1	4 MB
1	0	0	8 MB
1	0	1	16 MB
1	1	0	32 MB
1	1	1	64 MB

Reset Conditions

```
        Reset: 11UU UUUU
Command Reset: SSSS SSSS
```

POS Setup Register 2 (POS_SETUP2)

The POS_SETUP Register 2 is read accessible by the resident processor and read/write accessible by the Micro Channel as POS. The register is a direct mapping of POS Registers 3A and 3B (accessible through POS subaddressing), as well as POS Registers 6 and 7. POS Registers 3A and 3B also are mapped to Configuration Register 1, read accessible from the Micro Channel. More information on this register is available in the *Configuration Register 1 (CONF1)* on page 77. POS Registers 6 and 7 are mapped to Configuration Register 3. More information on this register is available in the *Configuration Register 3 (CONF3)* on page 78. More information on the POS subaddressing feature is available in the *Extended POS Base Address Register (XPOS)* on page 81 and *POS Subaddressing* on page 82. The POS_SETUP2 is normally configured through the POS setup routine.

The POS 6 Register also provides status for channel check conditions. This status is reset by writing this register to 00h. This resetting of the register can only be done through POS access.

Register Format

(Local Bus Address = 1FFA 0004h) 32-bit read-only

(POS Addresses = Subaddr 100,101h; POS 6,7) 8-bit read/write

	(7) 7							(0) 0
POS 3A	I12	I11	I10	A28	A27	A26	A25	A24

	(15) 7							(8) 0
POS 3B	RSV	RSV	ADP	AL3	AL2	AL1	AL0	EN

	(23) 7							(16) 0
POS 6	S7	S6	S5	S4	S3	S2	S1	S0

	(31) 7							(24) 0
POS 7	S15	S14	S13	S12	S11	S10	S9	S8

Status Format

	(23) 7							(16) 0
POS 6	RSV	RSV	RSV	BDP	SDP	XST	CTO	IBE

Bit Descriptions

POS 3A

- Bits 7–5: I/O Address Assignment. These bits are used together with POS 5, Bits 5–3, to locate the Micro Channel I/O base address on any 1KB boundary within the 64KB Micro Channel I/O address space. These bits correspond to three bits of the Micro Channel I/O address space, with I12 = I/O address bit 12 and I10 = I/O address bit 10. More information on the primary POS bits associated with I/O Address Assignment is located in the *POS_Setup1 Register (POS_SETUP1)* on page 70.
- Bits 4–0: Address Bits. These bits are used with POS 4, Bits 7–1, to place the starting address of a second shared memory window on any 1MB boundary within the 4GB Micro Channel address space. These bits are set to 0 at reset. More information on memory window options can be found in the *Shared Memory* on page 100.

Reset Conditions

```
        Reset:  1110 0000
Command Reset:  SSSS SSSS
```

POS 3B

- Bits 7–6: Reserved.
- Bit 5: Asynchronous Data Parity Condition Disable. This bit, when set, disables the asynchronous data parity condition. This function is described separately in *Asynchronous Data Parity Check Description* on page 84.
- Bits 4–1: Arbitration Level 2. These bits are used to define a second Micro Channel arbitration level.
- Bit 0: Arbitration Level 2 Enable. This bit is used to enable the second arbitration level. When set, this bit enables the arbitration level defined in POS 3B, Bits 4–1, for use by either Bus Master channel. Assignment of arbitration level to each channel is under control of the Channel Control Register (CCR) of that channel. When this bit is reset, both channels use the arb level defined in POS 3.

Reset Conditions

```
        Reset: 000U UUU0
Command Reset: 000S SSSS
```

POS 6

- Bits 7–0: POS Subaddress Bits 7–0.

Reset Conditions

```
        Reset: UUUU UUUU
Command Reset: SSSS SSSS
```

POS 6 (Status)

- Bits 7–5: Reserved. These bits are set to 0 when a channel check condition occurs.
- Bit 4: Basic Cycle Data Parity Error. This bit, when set, indicates that a data parity error occurred on the Micro Channel during a basic transfer cycle to the Micro Channel Interface Chip.
- Bit 3: Streaming Cycle Data Parity Error. This bit, when set, indicates that a data parity error occurred on the Micro Channel during a streaming data cycle to the Micro Channel Interface Chip.
- Bit 2: Extra Streaming Data Strobe. This bit, when set, indicates that an extra streaming data strobe was active after the Micro Channel Interface Chip as a slave indicated streaming terminations.
- Bit 1: Channel Ready Timeout. This bit, when set, indicates that the Micro Channel Interface Chip as a slave has negated Micro Channel channel ready for more than three microseconds.
- Bit 0: Invalid Byte Enables. This bit, when set, indicates that an invalid combination of byte enables has occurred on the Micro Channel during a transfer to the Micro

Channel Interface Chip as a slave. This error condition is checked for both basic transfer and streaming data transfers.

POS 7

- Bits 7–0: POS Subaddress Bits 15–8.

Reset Conditions

```
        Reset: UUUU UUUU
Command Reset: SSSS SSSS
```

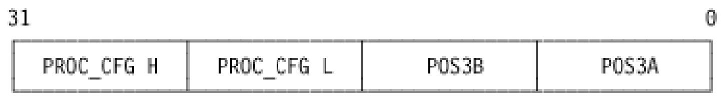
Configuration Register 1 (CONF1)

Configuration Register 1 is read accessible from the Micro Channel. Bits 15–0 of CONF1 are a direct mapping of POS Registers 3A and 3B; Bits 31–16 are a direct mapping of Bits 15–0 of the PROC_CFG Register. POS Registers 3A and 3B are also mapped to the POS_SETUP2 Register. More information on the contents of these registers is available in the *POS Setup Register 2 (POS_SETUP2)* on page 74. More information on Bits 16–31 is available in *Processor Configuration Register (PROC_CFG) (Micro Channel)* on page 78. Normal operation is for POS to alter the contents of POS Registers 3A and 3B, and the resident processor to alter the contents of the PROC_CFG Register.

Register Format

(Micro Channel Address = Base + 14h) 32-bit read-only

(POS 3A,3B Addresses = Subaddress 100,101) 8-bit read/write



Reset Conditions

```
      Reset: 0000 0000 000U UU00 000U UUU0 1110 0000
Command Reset: 0000 0000 000S SS00 000S SSSS SSSS SSSS
```

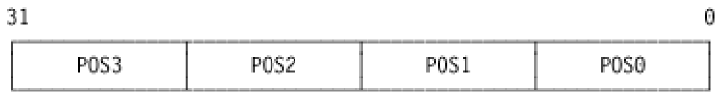
Configuration Register 2 (CONF2)

Configuration Register 2 is read/write accessible from the Micro Channel. The register is read/write accessible from the Micro Channel as four POS Registers. The CONF2 Register is a direct mapping of POS Registers 0–3. POS Registers 0 and 1 are also mapped to the CRDID Register. More information on the contents of these registers is available in *Card Identification (CRDID)* on page 82. POS Registers 2 and 3 are also mapped to the POS_SETUP1 Register. More information on the contents of these registers is available in the *POS_Setup1 Register (POS_SETUP1)* on page 70. Normal operation is for POS to alter the contents of Configuration Register 2.

Register Format

(Micro Channel Address = Base + 18h) 32-bit read-only

(POS Addresses = 0-3) 8-bit read/write



Reset Conditions

```
        Reset: 0001 1100 0001 0010 0000 0000 0000 0000
Command Reset: SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSSS
```

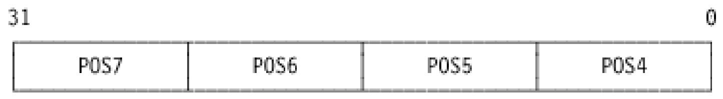
Configuration Register 3 (CONF3)

Configuration Register 3 is read accessible from the Micro Channel. The register also is read/write accessible from the Micro Channel as POS. The register is a direct mapping of POS Registers 4–7. POS Registers 4 and 5 are also mapped to the POS_SETUP1 Register. More information on the contents of these registers is available in the *POS_Setup1 Register (POS_SETUP1)* on page 70. POS Registers 6 and 7 are mapped to the POS_SETUP2 Register. More information on the contents of these registers is available in the *POS Setup Register 2 (POS_SETUP2)* on page 74. Normal operation is for POS to alter the contents of Configuration Register 3.

Register Format

(Micro Channel Address = Base + 1Ch) 32-bit read-only

(POS Addresses = 4-7) 8-bit read/write



Reset Conditions

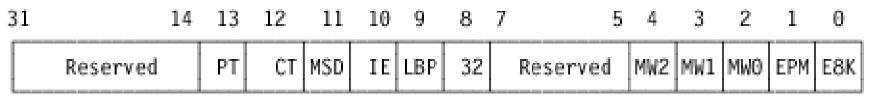
```
        Reset: UUUU UUUU UUUU UUUU 11UU UUUU UUUU UUU0
Command Reset: SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSSS
```

Processor Configuration Register (PROC_CFG) (Micro Channel)

The PROC_CFG register is initialized by the power-on self-test (POST) code to configure some of the hardware features of the Micro Channel Interface Chip. Bits 15–0 of this register also are mapped to Bits 31–16 of Configuration Register 1, read accessible from the Micro Channel. More information on this register is available in the *Configuration Register 1 (CONF1)* on page 77.

Register Format

(Local Bus Address = 1FFA 0010h)   32-bit read/write (bit 8 read-only)



31–14	13	12	11	10	9	8	7–5	4	3	2	1	0
Reserved	PT	CT	MSD	IE	LBP	32	Reserved	MW2	MW1	MW0	EPM	E8K

Bit Descriptions

- Bits 31–14: Reserved. These bits read back zeros.
- Bit 13: Performance Timer Disable. This bit, when set, disables the performance timer function.
- Bit 12: CD CHRDY Timeout Disable. This bit, when set, disables the CD CHRDY timeout logic.
- Bit 11: Multiplexed Streaming Data Enable. This bit, when set, enables the use of 64-bit, or multiplexed, streaming data on the Micro Channel. This bit powers up reset.
- Bit 10: Micro Channel Interrupt Enable. This bit, when set, enables interrupts to the Micro Channel. Interrupts are presented to the Micro Channel by writing the interrupt valid bit (IV) in the Command Busy Status Port. More information on these interrupts is available in ISP and CBSP.
- Bit 9: Local Bus Parity Enable. This bit, when set, enables the checking of Local Bus address and data parity. Local Bus parity generation is always enabled.

  - The Performance Timer extends -BURST for up to 500 nanoseconds after completion of a Micro Channel transfer to allow time for more data to become available in the intermediate buffer. Independent of the state of this bit, the timer is not enabled when the byte count has reached zero, after detection of exception conditions, during reset, or when data is available for the other channel.
  - The CD CHRDY timeout logic provides a mechanism to recover, with synchronous channel check, if a transfer to the Micro Channel Interface Chip as a Micro Channel slave exceeds 3.5 microseconds.

- Bit 8: 32/16 Bit Detect. This bit provides the state of the -16/32 DETECT signal of the Micro Channel Interface Chip. When this bit is set, the chip operates as a 32-bit device; when reset, the chip operates as a 16-bit device. This bit is read-only from both the Micro Channel and the Local Bus.
- Bits 7–5: Reserved. These bits read back 0.

- Bits 4–2: Memory Window Size Bits. These bits are written by ROS, indicating the size of the full shared memory window.

MW2	MW1	MW0	Window Size
0	0	0	512 KB
0	0	1	1 MB
0	1	0	2 MB
0	1	1	4 MB
1	0	0	8 MB
1	0	1	16 MB
1	1	0	32 MB
1	1	1	64 MB

- Bit 1: Full Memory Window Enable. Setting this bit enables the full shared memory window to be gated onto the Micro Channel. Resetting this bit degates the memory window from the Micro Channel. More information on memory window options can be found in the *Shared Memory* on page 100.
- Bit 0: Enable 8KB or 16KB Memory Window. Setting this bit enables the 8KB or 16KB memory window in the system unit ROM area to be gated onto the Micro Channel. Resetting this bit degates the memory window. More information on memory window options can be found in the *Shared Memory* on page 100.

Reset Conditions

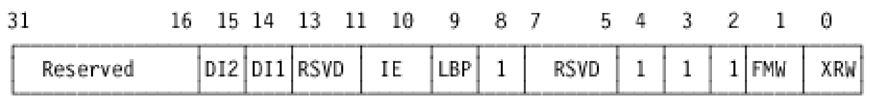
```
             Reset: 0000 0000 0000 0000 0000 0000 000U UU00
     Command Reset: 0000 0000 0000 0000 0000 0000 000S SS00
ROM Initialization: 0000 0000 0000 0000 0011 01-1 000X XX00
                    (where X = the memory window size,
                    and - is as defined on Page 78
                    for PROC_CFG, Bit 9.)
```

Processor Configuration Register (PROC_CFG) (PCI)

The PROC_CFG register is initialized by the power-on self-test (POST) code to configure some of the hardware features of the PCI System Bus Interface Chip. The contents of this register are also mapped to Bits 31–16 of Configuration Register 1, read-accessible from the PCI Bus. Bits 8–0 are read-only from the CFE Local Bus side. More information on this register can be found in *CFE Base Address Register (CFEBAR)* on page 155.

Register Format

(CFE Local Bus Address = 1FFA0010h) 32-bit read/write (bit 8–0 read-only)



31 16	15	14	13 11	10	9	8	7 5	4	3	2	1	0
Reserved	DI2	DI1	RSVD	IE	LBP	1	RSVD	1	1	1	FMW	XRW

Bit Descriptions

- Bits 31–16: Reserved. These bits read back zeros.
- Bit 15–14: DMA Channel 1(2) Interrupt Direction Bit: These bits when set, direct termination interrupts to the CFE Local Bus.
- Bits 13–11: Reserved. These bits read back zeros.
- Bit 10: PCI Bus Interrupt Enable. This bit, when set, enables interrupts to the PCI Bus. Interrupts are presented to the PCI Bus by writing the interrupt valid bit (IV) in the Command Busy Status Port.
- Bit 9: CFE Local Bus Parity Enable. This bit, when set, enables the checking of the CFE Local Bus address and data parity. CFE Local Bus parity generation is always enabled.
- Bit 8: Reserved. This bit reads back 1. This bit is read-only from both the PCI Bus and the CFE Local Bus.
- Bits 7–5: Reserved. These bits read back zeros.
- Bits 4–2: Reserved. These bits read back zeros.
- Bit 1: Full Memory Window Enable. This read-only bit reflects the value of the Memory Space Enable bit (HSCR, Bit 1). This bit can be used to detect whether the Full Memory Window is enabled, bit-compatible with the Micro Channel Interface Chip.
- Bit 0: Expansion ROM Memory Window Enable. This read-only bit reflects the logical **AND** of the Address Decode Enable (XRBAR, Bit 0) and the Memory Space Enable bit (HSCR, Bit 1). This bit can be used to detect whether the Expansion ROM Memory Window is enabled, bit compatible with the Micro Channel Interface Chip.

Reset Conditions

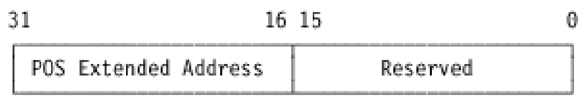
```
        Reset: 0000 0000 0000 0000 0000 0001 0001 1100
Command Reset: 0000 0000 0000 0000 0000 0001 0001 1100
```

Extended POS Base Address Register (XPOS)

The Extended POS Base Address Register supplies the base address in Local Bus address space for the POS subaddress extension. This 16-bit value represents the upper 16-bits of the Local Bus address, defining a 64KB region for POS subaddressing. Addressing within this 64KB region is provided by POS 6 and POS 7. This register must be initialized before writing the CRDID Register. More information on subaddressing is described under *POS Subaddressing* on page 82.

Register Format

(Local Bus Address = 1FFA 0018h) 32-bit read/write



Bit Descriptions

- Bits 31–16: POS Extended Address.
- Bits 15–0: Reserved.

Reset Conditions

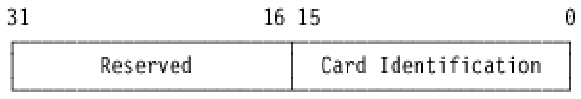
```
               Reset: UUUU UUUU UUUU UUUU 0000 0000 0000 0000
       Command Reset: SSSS SSSS SSSS SSSS 0000 0000 0000 0000
ROM Initialization: 0010 0000 0000 0000 0000 0000 0000 0000
```

Card Identification (CRDID)

The Card Identification Register stores the 16-bit adapter ID, readable from POS Registers 0 and 1. The low-order byte of this register is mapped into POS 0, while the high-order byte is mapped into POS 1. The CRDID is a 16-bit read/write register accessible from the Local Bus. This register also is mapped to Configuration Register 2, read accessible from the Micro Channel. More information on this register is available in *Configuration Register 2 (CONF2)* on page 77.

Register Format

(Local Bus Address = 1FFA 000Ch)   32-bit read/write



Bit Descriptions

- Bits 31–16: Reserved. These bits must be set to 0.
- Bits 15–0: Card ID Bits. These bits store the 16-bit adapter ID. The value for the co-processor platform is 8FECh.

Reset Conditions

```
               Reset: 0000 0000 0000 0000 0000 0000 0000 0000
       Command Reset: 0000 0000 0000 0000 SSSS SSSS SSSS SSSS
ROM Initialization: 0000 0000 0000 0000 1000 1111 1110 1100.
```

Unless the AIB supplies the card ID, the ROM initialization value is the default.

POS Subaddressing

POS subaddressing allows access to additional configuration data. Subaddressing creates a window to additional POS locations through POS Register 3. Locations within this window are indexed from a base address stored in the Extended POS Base Address Register (XPOS). This register stores a 16-bit value, representing the upper 16 bits of a Local Bus base address. This base address defines a 64KB area on the Local Bus. Locations within this 64KB area are addressed by a non-zero value stored in POS Registers 6 and 7.

The first two subaddress locations, 0100h and 0101h in POS 6 and 7, address two registers internally: POS 3A and 3B. POS 6 contains the least-significant byte, and POS 7 the most-significant byte of the subaddress. These two registers are used for extended functions on the Micro Channel. Accesses to subaddress locations from 0001h – 00FFh in POS 6 and 7 generate Local Bus cycles with the Extended POS address. This subaddress area is available for vital product data (VPD) or configuration data, depending on implementation. Subaddress locations above 0101h are out of normal VPD range, but are available as additional subaddress space. Accesses to these subaddresses will generate cycles to the Local Bus. This subaddress mapping is summarized in Table 8-3.

Table 8-3. POS Subaddress Map

POS 7 (hex)	POS 6 (hex)	Function
00	01 – FF	Vital Product Data (VPD)
01	00	POS 3A
01	01	POS 3B
01 – FF	02 – FF	Additional Subaddress Space (0102h – FFFFh)

Note that subaddressing presents a non-bursting interface to the Local Bus, and that prefetching on reads is limited to single transfers. That is, reading subaddress space through POS 3 forces **only** single transfers.

Design Note – Data Steering of Subaddress Cycles to the Local Bus

Local Bus accesses resulting from subaddress cycles assume, by CFE definition, a 32-bit bus width. Therefore, subaddress interfaces to the CFE bus must provide data on all four bytes of the Local Bus although POS accesses, by Micro Channel definition, are eight-bit accesses. The Micro Channel Interface Chip provides an 8-bit interface to the Micro Channel, that is, all odd subaddress accesses are steered to the least-significant byte of the Micro Channel.

Subaddress reads are performed as a 32-bit read on a 4-byte boundary. Odd subaddress reads, therefore, are 32-bit accesses, and the correct byte is selected from the four-byte access and presented to the Micro Channel.

Subaddressing on the Local Bus is implemented as a 32-bit access because all devices on the CFE Local Bus are, by definition, 32-bit devices. Since all subaddress accesses are, by Micro Channel definition, 8-bit accesses, an 8-bit device can be interfaced for purposes of subaddressing, if the read address can be properly incremented, and the data properly steered for odd addressing. The easiest implementation, however, is to provide a 32-bit interface. For example, a 32-bit DRAM or SRAM can be used. POS Setup can be delayed until subaddress data is provided to this memory area by delaying the setting of the CRDID Register until the data transfer to memory is complete. Micro Channel hosts will not perform POS setup, prior to reading a valid POS ID in POS Registers 0 and 1.

I/O Check Reporting and Parity Function

With the -CHCK enable bit set in POS, -CHCK is asserted for the following conditions:

- An exception condition is detected on the Local Bus during a slave read from the Micro Channel. This function is only supported with the CD CHRDY Timeout Disable (Bit 12, PROC_CFG) set with Timeout enabled.

- Local Bus parity is detected during a slave read from the Micro Channel. This function is only supported with the CD CHRDY Timeout Disable (Bit 12, PROC_CFG) set with Timeout enabled.
- A parity error is detected on the Micro Channel during a slave write from the Micro Channel.
- An address parity error is detected on the Micro Channel.
- The Micro Channel Interface Chip asserts CD CHRDY as a Micro Channel slave for longer than three microseconds.
- An invalid byte enable combination is detected on the Micro Channel during an access of the Micro Channel Interface Chip as a slave.
- An extra streaming data strobe is received after the Micro Channel Interface Chip indicates streaming termination as a slave.

The Micro Channel Interface Chip detects -CHCK only when operating as a Bus Master. Additionally, the Bus Master function can only detect those channel checks that occur while it has control of the Micro Channel. The detection of -CHCK stops the Bus Master channel and forces a termination interrupt to the resident processor.

Micro Channel parity detection is also supported when operating as a Bus Master. The detection of parity forces a termination interrupt to the resident processor and stops the Bus Master Channel, but does not assert -CHCK.

Micro Channel parity is controlled in the POS registers by the Data Parity Enable Bit. For Micro Channel parity to be supported during peer-to-peer transfers, parity must be enabled on both the master and the slave.

Asynchronous Data Parity Check Description

A special parity condition exists for non-streaming accesses to the Micro Channel Interface Chip. By Micro Channel definition, data written to the Micro Channel Interface Chip as a slave is set up and held relative to the -CMD signal. If a Bus Master changes data while -CMD is active, resulting in a parity error, this error is normally reported as an asynchronous -CHCK. Since some systems cannot handle asynchronous -CHCKs, a bit is provided in POS 3B, Bit 5 to disable this -CHCK reporting, and alternately report this error through LBPE and its associated interrupt. For this special error, the default operation is for asynchronous -CHCK reporting. More information on the disable bit and alternate reporting mechanism is provided in *POS Setup Register 2 (POS_SETUP2)* on page 74 and LBPR.

Bus Master Channel Functional Description

The following sections describe the operation of Bus Master Channels 1 and 2.

Overall Operation

All data transactions, including Bus Master transfers, are subject to intermediate buffering. A diagram of this intermediate buffering is shown in Figure 8-1.

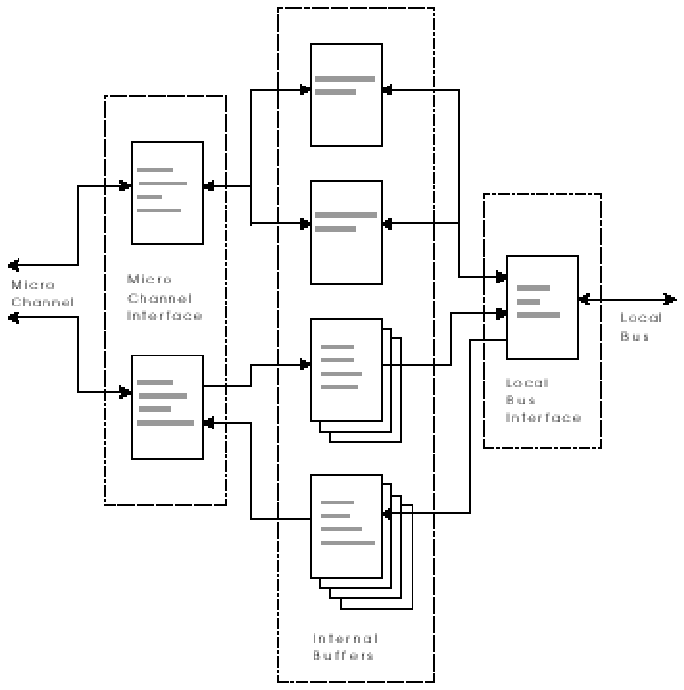


Figure 8-1. Intermediate Buffering

Each Bus Master channel manages data movement through a 128-byte buffer. This buffer is organized as two, 64-byte ping-pong buffers. These two buffers alternately control the Local Bus and Micro Channel interfaces, optimizing the total throughput of the Bus Master transfers.

For a given data transfer, the first access to the Micro Channel is a minimum of 128 bytes. That is, if the direction of the transfer is reading the Micro Channel, the Bus Master channel will read a minimum of 128 bytes (unless reaching a terminal count condition), before it relinquishes the Micro Channel. If direction is writing the Micro Channel, the Bus Master channel will fetch a minimum of 128 bytes from the Local Bus before accessing the Micro Channel. The Bus Master will continue to perform transfers to either interface until it runs out of buffer space. Buffer space is lost when the current 64-byte buffer is filled or flushed, and the other buffer is active on the opposite interface. Access resumes when this other buffer becomes available. For very fast Local Bus interfaces, such as static random access memory (SRAM), a very high bandwidth can be sustained on the Micro Channel.

Additional bandwidth is gained on the Micro Channel by sharing the Micro Channel grant period for the Micro Channel Interface Chip between the two Bus Master channels. That is, if Bus Master Channel 1 has run out of buffer space, and there is additional time on the Micro Channel, Bus Master Channel 2, will perform accesses to its 128-byte buffer. This sharing of the Micro Channel between channels is only possible when the arbitration levels for the channels are the same; that is, the A2 option in the Channel Control Register (CCR) of each channel is set to the same state.

Each Bus Master channel is configured from the Local Bus by programming a Channel Descriptor Block (CDB) resident in the Micro Channel Interface Chip A description of the CDBs for each Channel is given in the *Bus Master Channel Descriptor Block (BMCDB)* on page 92.

Each Bus Master channel is controlled through a CCR, located in each CDB. A channel is enabled by setting the Start/Stop Bit (Bit 0) in the CCR. The direction of the data transfer is set by the Direction Bit (Bit 1) in the CCR. A description of the CCR for each channel is given in the CCR.

When the Start/Stop Bit is set in the Bus Master channel, operation begins in one of two ways, depending on the state of the Direction Bit. If the direction is set to read from the Micro Channel (writing to the Local Bus), the Bus Master channel will immediately arbitrate for the Micro Channel. When granted the Micro Channel, the Bus Master channel will fill its buffer by reading from the address specified in the System Address Register (SAR), located in the CDB. The channel will continue to fill its buffer until 128-bytes are read, or until the Byte Count in the Byte Count Register (BCR) equals zero. Concurrently, once the first 64-byte buffer is full, data is transferred from the buffer into memory resident on the Local Bus at the location specified in the Card Address Register (CAR). In the general case, the channel will read 128 bytes from the Micro Channel, before data is completely flushed to resident memory. At this point, the channel will release its control of the Micro Channel. After the buffered data is flushed to resident memory, the channel re-arbitrates for the Micro Channel. The channel continues this arbitrate/release/flush sequence until the Byte Count equals zero.

If the direction is set to write to the Micro Channel (reading from the Local Bus), the channel fills the buffer from memory resident on the Local Bus from the location loaded in the CAR. The channel continues to fill the buffer until 128 bytes are transferred or the Byte Count equals zero. After 128 bytes are loaded, the channel arbitrates for the Micro Channel. When granted, the channel flushes the buffered data to the Micro Channel at the location loaded in the SAR. Concurrently, after the first 64-byte buffer is completely flushed on the Micro Channel, additional data is loaded in the buffer from the Local Bus. In the general case, the Micro Channel will continue to flush the additional data until the buffer is empty. When there is no more data to be transferred the channel releases its control of the Micro Channel. The channel continues to fill from the Local Bus until the buffer is full or the Byte Count equals zero. The channel will then re-arbitrate for the Micro Channel. The channel continues this fill/arbitrate/flush sequence until the Byte Count equals zero.

Bus Operation

The Micro Channel Interface Chip is capable of performing transfers with any address alignment combination as a Bus Master, and as a Micro Channel slave through the shared memory window. The number of bytes transferred on either the Micro Channel or the Local Bus is controlled by the respective byte enable signals for that bus.

On the Micro Channel, the number of bytes transferred during a cycle is determined by the total number of bytes to be transferred, the Micro Channel address, and the width of both the master and the slave. As a Micro Channel Bus Master, in all cases, the maximum number of bytes are transferred. Therefore, for a transfer between a 32-bit master and a 32-bit slave at an address on an odd-byte boundary, that is, an address with the least significant nibble equal to 1h, the Micro Channel Interface Chip transfers three bytes.

Streaming data transfers, by Micro Channel definition, must be aligned to the width of the transfer. The Micro Channel Interface Chip, as a Micro Channel Bus Master, does this alignment, if starting on an odd boundary, in a minimum number of cycles. Therefore, in the preceding example, if the slave has multiplexed streaming data capability, the Micro Channel Interface Chip transfers three bytes, then performs a four-byte transfer before initiating a streaming cycle.

As a Local Bus Master, the Micro Channel Interface Chip also transfers the maximum number of bytes per transfer. When writing a slave on an odd-byte boundary, the Micro Channel Interface Chip transfers three bytes. In this case, the upper three byte enables (BE3-1), are asserted. When reading a slave on any boundary, the entire 32-bit word associated with that address is read; that is, all four byte enables are asserted and the two least significant bits of the Local Bus address are zero.

Appended I/O Operations

Each channel has the ability to append an I/O write to the Micro Channel upon reaching a terminal count. The intent of this I/O is to create an efficient method for interrupting the Source Identification Register (SIR) through the Attention Port of a slave adapter after a data transfer is complete. The function of the SIR is discussed in the PIIP.

The address and data for an Appended I/O operation are both stored in the Bus Master Address Register (BMAR). When an appended I/O operation is enabled, the channel writes to the Micro Channel I/O address stored in the lowest 16 bits of the BMAR in the CDB. In general, this function is used to write an eight-bit value to the Attention Port of the Micro Channel slave’s Subsystem Control Block (SCB) register map. When a terminal count is reached, the channel appends the I/O write to this address, using the data stored in Bits 16–23 of the BMAR. That is, if the direction of transfer is from the Local Bus to the Micro Channel, the I/O is appended after the last transfer to the Micro Channel. If the direction of transfer is from the Micro Channel to the Local Bus, the I/O is appended after the last transfer to the Local Bus. The BMAR for each channel is discussed in the BMAR.

The channel controls appended I/O operations through its CCR. The Appended I/O (AP) Bit, Bit 7, sets the channel for an Appended I/O operation. The setting and resetting of bits in the slave SIR is discussed in thePIIP.

Setting both the AP and the PS bits in the CCR is an invalid combination. Using this invalid combination will not interfere with the data transfer. A normal termination will occur, but no I/O write will be appended.

Posted Status Operations

Each channel has the ability to perform a posted status operation to a location in Local Bus address space. The intent of this operation is to provide an alternate method of signaling the normal termination of a data transfer, in addition to the terminal count interrupt. Posting status has two applications in conjunction with list chained data transfer elements:

- Normal termination status for each element can be posted in a table in Local Bus address space. A terminal count interrupt can be asserted on the last element in the list chain. The table can be used for identifying a failing element during error recovery.
- The posted write itself can be made to a device or single address location to positively acknowledge completion of a list chain element. This is especially useful for

interfacing to an intelligent device other than the resident processor controlling the enqueuing and dequeuing of elements in the list chain or pipe.

The 32-bit address for this operation is stored in the Bus Master Address Register (BMAR). The appended operation is a 32-bit write to this location of the contents of the Bus Master Status Register (BMSTAT). When a terminal count is reached, the channel posts status to the Local Bus; that is, if the direction of transfer is from the Local Bus to the Micro Channel, status is posted after the last transfer to the Micro Channel. If the direction of transfer is from the Micro Channel to the Local Bus, status is posted after the last transfer to the Local Bus.

The channel controls Posted Status operations through its CCR. The Posted Status Bit, Bit 8, sets the channel for Posted Status operation.

Note that only normal termination status is posted. Status is not posted for exception conditions, but rather the channel is stopped and an interrupt is posted.

Linked List Chaining

Both Bus Master channels support Linked List Chaining (LLC), or scatter/gather DMA. Linked List Chaining provides the ability for each channel to auto-initialize its Channel Descriptor Block (CDB) from a predefined list in memory. Each element in this list consists of a set of new register values. An element is loaded into the CDB when the channel reaches a terminal count for the current data transfer. Since each element represents a separate data transfer, Linked List Chaining allows the channel to interleave the scattering and the gathering of data from different buffer locations in memory, with minimum intervention from the resident processor. In addition, control information in the CCR also can be updated, allowing dynamic changing of DMA parameters.

Within the CDB, a 32-bit address pointer to the list in memory is maintained. At the time of terminal count, if list chaining is enabled, the hardware will fetch six, 32-bit words starting at this memory address and reload the CDB at hardware speeds. List chaining provides a mechanism to offload the resident processor from register initialization after every terminal count. Based on the value programmed in the CCR, the channel can optionally interrupt the resident processor, in addition to List Chaining, on a terminal count.

This list of buffers can occupy any area in free memory. The list chaining function is shown in Figure 8-2.

At the end of this operation, the CDB contains a new pointer to a new list located anywhere else in memory.

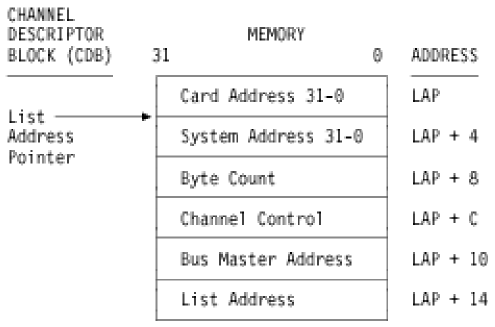


Figure 8-2. Linked List Chaining

Modes of Operation

There are several special modes of operation for terminating the Bus Master channels and interrupting the Local Bus. These different modes involve several bits in the CCR.

It is easiest to think of transfers as broken into three steps:

1. A data transfer
2. An optional appended I/O
3. A list chain operation to bring in a new list chain element.

   A list chain element is a six-word entry in Local Bus address space containing the values for initializing the Channel Descriptor Block (CDB) for a data transfer. It is important to note that the termination interrupt enabled by the Terminal Count Interrupt Enable (Bit 2) in the CCR, represents the normal termination of a data transfer, or of a data transfer plus appended I/O when an appended I/O is enabled, and does not necessarily represent termination or disabling of Bus Master operation.

The beginning of a list chain operation is considered the start of a new data transfer.

- ***Termination Interrupts on the Fly***. Termination interrupts can be enabled for any element in a list chain, not just the last element, associated with the stopping of the channel. Since Normal Termination interrupts are associated with the normal termination of a data transfer or of a data transfer with optional Appended I/O, an interrupt can be asserted after any data transfer or after its Appended I/O, if enabled, by enabling the Terminal Count Interrupt Enable Bit (CCR, Bit 2) for that element in the list chain.

  Although it is possible to take interrupts on the fly, normal operation is to take a single termination interrupt at the end of the entire list chain. A single termination interrupt is desirable for the following two reasons:

  - For list chains involving small transfers, it is difficult to take interrupts for each element, thus some interrupts could be lost without stopping the channel.

  - Since a new list chain element and a new transfer are immediately list chained after the interrupt, there is no status available in the BMSTAT Register representing the termination interrupt. (List Chaining represents the start of a new transfer; therefore, status is cleared).

- ***List Chaining Zero Byte Elements***. If a channel is active with list chaining enabled, and there are currently no transfers to take place, list chain elements containing a byte count of zero can be appended to the list chain. As long as the List Chaining Enable bit (CCR, Bit 3) is enabled in the appended elements, list chaining will continue without data transfers.

- ***Stopping Channel after List Chaining Bit***. The Bus Master Channel can be stopped after the current list chain element is read from memory and prior to performing the data transfer. Upon stopping, the channel can be re-enabled and operation will continue with the current data transfer.

  This mode of operation is especially useful in conjunction with the List Chaining Zero Byte Elements option. In the case where there are currently no transfers to take place, an element of zero byte count can be appended *with the Stop Channel after the List Chaining Bit (CCR, Bit 4) is set in the CCR*. As long as the List Chain Enable Bit is also set in this element, operation will continue upon re-enabling the channel, with a list chain operation taking place to the address pointed to by the current list address pointer. The Stop Channel after List Chaining can be used as a true NOP or Pause function for a given list chain or pipe..

  Since this option terminates after a list chain, and not after a data transfer or appended I/O operation, there is no normal termination interrupt associated with stopping the channel in this mode. The start/stop bit must be polled to determine when the channel is stopped.

- ***Resetting the Start/Stop Bit***. The Bus Master Channel can be stopped non-catastrophically by simply resetting the start/stop bit when list chaining is **not** set for either channel. If list chaining is enabled, the DMA channels cannot be stopped by writing to the CCR or the BMCMD Registers. The DMA channel can be stopped by creating a list element that has list chaining disabled and/or the Stop-on-List-Chain Bit set, and allowing that element to be read normally into the DMA channel’s CDB Register and cause an orderly stop.

  There is some latency between the resetting of the bit and the stopping of the channel. During this period the start/stop bit will not read back 0. The start/stop bit will read back 0 when the channel is stopped, and function as termination status for this mode of operation Operation will resume with the pending data cycles for both the Local Bus and the Micro Channel, upon resetting the start/stop bit.

  Like the **Stopping the Channel after List Chaining** option, this operation is not associated with normal termination of data transfers or Appended I/O, and, after the start/stop bit is reset, there is no termination interrupt associated with this mode of operation.

- ***Resetting the Bus Master Channel***. This function is not supported with List Chaining enabled.

- ***Watchdog Input***. The Watchdog input is an asynchronous input that also forces both channels to terminate non-catastrophically. Although this input is an exception condition, the exception is reported separately to the resident processor, and may be unrelated to the operation of either Bus Master channel. As a result, both channels will terminate without interrupt or status. The Bus Master channels are reset on this error and are not available to complete their current transfers.

Stopping the Channel

The following is a summary of conditions that stop the Bus Master channel with termination interrupt and status:

- Micro Channel Data Parity Error
- -CHCK Detected on Micro Channel
- No Select Feedback Return
- Invalid Combination (-DS16RTN,-DS32RTN)
- Loss of Micro Channel (ARB/-GNT raised by host)
- Resetting of Card Enable (POS 2, Bit 0)
- Resetting of DMA Enable (SCP, Bit 1)
- Local Bus Parity Error
- -L_EXCPT Detected on the Local Bus

The following conditions stop the Bus Master channel without termination interrupt and status:

- All Resets (Command, Micro Channel)
- Termination using the Stop on List Chain Command
- Resetting the Start/Stop Bit in the CCR
- -WATCHDOG Detected on the Local Bus.

Note that all terminations are non-catastrophic; that is, termination of the channel is non-disruptive to either the Micro Channel or the Local Bus.

In all cases, with the exception of resetting the Start/Stop bit, the channel is completely reset on termination. When resetting the Start/Stop bit, the channel terminates with the internal state machines intact, providing the option to restart the channel and complete the current transfer.

Bus Master Registers

Bus Master Channel Descriptor Block (BMCDB)

Each channel has a 6-word channel descriptor block (CDB). This CDB is loadable from the Local Bus. The CDB register map is described in Table 8-7.

Table 8-4. BMCDB Local Bus Register Map

CDB Register	Channel 1	Channel 2
Card Address Register 31-0	1FFA 3000	1FFA 4000
System Address Register 31-0	1FFA 3004	1FFA 4004
Byte Count Register	1FFA 3008	1FFA 4008
Channel Control Register	1FFA 300C	1FFA 400C
Bus Master Address Register 31-0	1FFA 3010	1FFA 4010
List Address Pointer 31-0	1FFA 3014	1FFA 4014

Register descriptions within the CDB are given in the following paragraphs. All channel addresses are Local Bus addresses.

Card Address Register (CAR)

The card address register (CAR) is a 32-bit register that contains the Local Bus address of the next data word in resident memory to be transferred. Depending on the byte count, the last access to resident memory may be one, two, three, or four bytes. This value is initially loaded by a write from the resident processor. Subsequent loading is either by additional writes, or if list chaining is enabled, by hardware accesses to a list in resident memory.

Register Format

```
(Channel 1 Address = 1FFA 3000h)          32-bit read/write
(Channel 2 Address = 1FFA 4000h)          32-bit read/write
```

31
0
Card Address

Reset Conditions

```
             Reset: UUUU UUUU UUUU UUUU UUUU UUUU UUUU UUUU
     Command Reset: SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSSS
ROM Initialization: 0000 0000 0000 0000 0000 0000 0000 0000
```

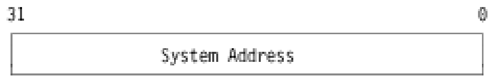
System Address Register (SAR)

The System Address Register (SAR) is a 32-bit register that contains the physical Micro Channel address of the next data word to be transferred. This value is initially loaded by a write from the resident processor. Subsequent loading of this value is either by additional writes, or if list chaining is enabled, by hardware accesses to a list in resident memory.

The contents of this register are altered after an Appended I/O operation to the Micro Channel and are not available for diagnostic purposes after this transfer.

Register Format

```
(Channel 1 Address = 1FFA 3004h)          32-bit read/write
(Channel 2 Address = 1FFA 4004h)          32-bit read/write
```



Reset Conditions

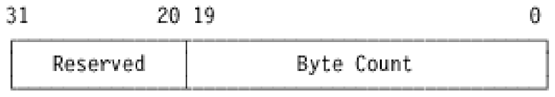
```
             Reset: UUUU UUUU UUUU UUUU UUUU UUUU UUUU UUUU
     Command Reset: SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSSS
ROM Initialization: 0000 0000 0000 0000 0000 0000 0000 0000
```

Byte Count Register (BCR)

This register contains a 20-bit value that signifies the number of bytes that will be transferred before a terminal count is reached.

Register Format

```
(Channel 1 Address = 1FFA 3008h)          20-bit read/write
(Channel 2 Address = 1FFA 4008h)          20-bit read/write
```



Reset Conditions

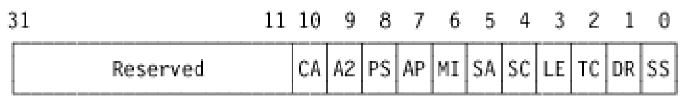
```
             Reset: 0000 0000 UUUU UUUU UUUU UUUU UUUU UUUU
     Command Reset: 0000 0000 SSSS SSSS SSSS SSSS SSSS SSSS
ROM Initialization: 0000 0000 0000 0000 0000 0000 0000 0000
```

Channel Control Register (CCR)

The Channel Control Register (CCR) is the control and command area for a channel. It is initially loaded by a write from the resident processor. The CCR Register is subsequently reloaded either by additional writes or, if linked list chaining is enabled, by the hardware during memory list accesses. In this way, dynamic control of the channel is possible.

Register Format

```
(Channel 1 Address = 1FFA 300Ch)          32-bit read/write
(Channel 2 Address = 1FFA 400Ch)          32-bit read/write
```



Bit Descriptions

This register stores the control/status information for Bus Master Channel 1.

- Bits 31–11: Reserved. These bits must be set to 0.
- Bit 10: Card Address Increment/NoIncrement Bit. This bit, when reset, indicates the channel increments the card address after each Local Bus transfer. This bit, when set, indicates the channel does not increment the card address after each transfer.
- Bit 9: Arb Level 2 Assignment Bit. This bit, when set, assigns the Arbitration Level 2, defined in POS 3B, to the Bus Master channel. Resetting this bit assigns Arbitration Level 1. If the Arbitration Level 2 Enable in POS 3B is not set, this bit is ignored and Arbitration Level 1 is assigned. Note that for the two Bus Master channels to alternate control of the Micro Channel during a single ownership of the Micro Channel by the Micro Channel Interface Chip, this bit must be set to the same value in both Channel CCRs. This restriction is independent of the state of the POS arbitration level initialization. For more information on control of the Micro Channel, see *Bus Master Channel Functional Description* on page 84.
- Bit 8: Post Status Bit. This bit, when set, enables the Bus Master channel to post the contents of the BMSTAT Register to the Local Bus address located in the Bus Master Address Register (BMAR). If the AP bit is set with this bit, both commands are ignored. The data transfer will terminate normally. More information on Posting Status can be found in the *Bus Master Address Register (BMAR)* on page 95.
- Bit 7: Appended I/O Command Bit. Setting this bit enables the channel to append a Micro Channel I/O write operation to an I/O address stored in the Bus Master Address Register. The data for the I/O transfer also is provided in the BMAR as Bits 23–16. 8-bit transfers to any I/O location are supported. If the PS bit is set with this bit, both commands are ignored. The data transfer will terminate normally.
- Bit 6: Memory-I/O Transfer Bit. This bit, when set, indicates the channel performs cycles to Micro Channel memory space. This bit, when reset, indicates the channel performs cycles to Micro Channel I/O space. When setting this bit for streaming to a single I/O location, Bit 5 must also be set.
- Bit 5: System Address Increment/NoIncrement Bit. This bit, when reset, indicates the channel increments the value in the system address register after each transfer. This bit, when set, indicates the channel does not increment the system address register after each transfer. This bit must be set for streaming I/O transfers.
- Bit 4: Stop Channel after List Chaining Bit. This bit is active only when the list chaining enable bit (LE) is set. When active, if this bit is set, the channel will stop after the *current* list chaining operation. That is, after the entire list element is read in which the bit is set. If the bit is reset, the channel will continue to do transfers after the

current list chaining operation. The channel always stops after the current transfer when the list chaining enable (LE) is not set.

- Bit 3: List Chaining Enable Bit. This bit, when set, indicates list chaining is enabled. This bit, when reset, indicates list chaining is disabled. More information is provided in the *Linked List Chaining* on page 88.
- Bit 2: Terminal Count Interrupt Enable Bit. If this bit is set, an interrupt will be presented to the resident processor when the value stored in the Byte Counter Register (BCR) transitions to 00 0000h. If this bit is reset, no interrupt occurs.
- Bit 1: Direction Bit. This bit signifies the direction of data transfer on the Micro Channel. This bit, when set, indicates reading data from the Micro Channel. This bit, when reset, indicates writing data to the Micro Channel.
- Bit 0: Start/Stop bit. Setting this bit initiates a Bus Master transfer. Hardware will reset the bit to indicate the transfer has completed. Writing a 0 to this bit will terminate the present transfer after the cycle in process has completed. No internal counters are reset, so that a subsequent write of 1 will continue the transfer where it left off. Note that writing a 0 to this bit and terminating the transfer is not supported with List Chaining.

  In addition, reading this bit will not read zero unless the cycle has completed and the channel is stopped. The bit, therefore, provides status for the channel, as well as controlling its function.

  When this bit is set, no other bit in the CCR can be updated by a direct I/O write from the resident processor. This feature allows the start/stop bit to be reset without corrupting the transfer currently executing.

  This is the only bit that is not updated during list chaining; all other bits in the CCR are updated. This feature allows the register to be loaded with new values during list chaining, without resetting the start/stop bit. Stopping the channel during list chaining is under control of Bit 4 in the CCR, Stop Channel after List Chaining.

Reset Conditions

```
               Reset: UUUU UUUU UUUU UUUU UUUU UUUU UUUU UUU0
       Command Reset: SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSS0
ROM Initialization: 0000 0000 0000 0000 0000 0000 0000 0000
```

Bus Master Address Register (BMAR)

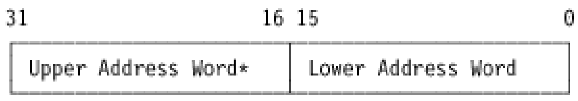
The Bus Master Address Register stores a 32-bit value for support of appended transfers. An appended transfer is an additional transfer appended at the end of a normal data transfer. There are two types of appended transfers: (1) Posting Status to the Local Bus, and (2) Appended I/O transfers to the Micro Channel. Appended transfers are selected in the Channel Command Register (CCR).

For Posting Status, the value in the BMAR represents a location in Local Bus address space. The contents of the BMSTAT for the given channel are written to this location as a 32-bit transfer. Therefore, the address should be 4-byte aligned. Note that the posting of status does not reset the value of the BMSTAT Register. This register must still be read upon termination to clear its contents. When list chaining, the contents of this register are cleared at the beginning of the next list chain operation to reset status for the next data transfer.

For Appended I/O transfers, the lower address word in the BMAR represents a location in Micro Channel I/O address space of a Micro Channel slave. The contents of the register are used by the Bus Master channel to locate the I/O address area of a slave device to access its SIR on appended I/O transfers. The lower eight bits of the upper word, that is, Bits 23–16, store the 8-bit data value for the Appended I/O transfer. Note that only 8-bit I/O transfers are supported. More information on Appended I/O transfers can be found in the *Appended I/O Operations* on page 87.

Register Format

```
(Channel 1 Address = 1FFA 3010h)          16-bit read/write
(Channel 2 Address = 1FFA 4010h)          16-bit read/write
```



* Bits 23-16 of the Upper Address are used for data in Appended I/O operations.

Bit Descriptions

- Bits 31–0: Address Bits. These bits store either the 32-bit address of a Posted Status location or the 16-bit I/O address of a Micro Channel slave.

Reset Conditions

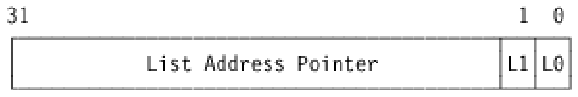
```
                Reset: UUUU UUUU UUUU UUUU UUUU UUUU UUUU UUUU
        Command Reset: SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSSS
ROM Initialization: 0000 0000 0000 0000 0000 0000 0000 0000
```

List Address Pointer (LAP) Register

The List Address Pointer (LAP) is a 32-bit register that contains the address of a resident memory location where a list of CDB information exists. The register is initially loaded by a resident processor write instruction. After initialization, the register can be reloaded by write instruction, or if list chaining is enabled, the register is automatically initialized by hardware from a list table entry in resident memory. The register is always incremented by six transfers, or 24 bits following the movement of one of the list table entries into the Channel Descriptor Block (CDB), but will reflect the value of the new LAP after the list chain is complete. The values written to the register should be 4-byte-aligned; that is, Bits 1 and 0, (L1,L0) are forced to zero. More information on list chaining can be found in *Linked List Chaining* on page 88.

Register Format

```
(Channel 1 Address = 1FFA 3014h)          32-bit read/write
(Channel 2 Address = 1FFA 4014h)          32-bit read/write
```



Reset Conditions

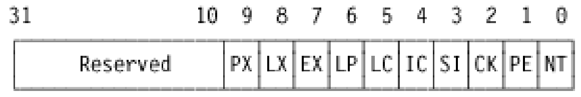
```
             Reset: UUUU UUUU UUUU UUUU UUUU UUUU UUUU UUUU
     Command Reset: SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSSS
ROM Initialization: 0000 0000 0000 0000 0000 0000 0000 0000
```

Bus Master Status Register, BMSTAT1 (2)

This register provides the termination status for each Bus Master channel. The contents of this register are cleared by a read access. More information on Local Bus interrupts is available in System Bus Interface Chip Interrupts on page 8-55.

Register Format

```
(Channel 1 Address = 1FFA 3018h)          32-bit read-only
(Channel 2 Address = 1FFA 4018h)          32-bit read-only
```



Bit Descriptions

- Bits 31–10: Reserved. These bits should always be set to 0.
- Bit 9: Posted Status Exception. This bit is set when a Local Bus exception occurs during a Posted Status operation. It is used together with Bit 7 to determine whether the data transfer was performed successfully, but the status was corrupted. The setting of this bit is mutually exclusive with the setting of Bit 8. Note that data parity detection does not apply to the posting of status. Therefore, only Bit 7 can be set concurrently with this bit.
- Bit 8: List Chaining Exception. This bit is set when a Local Bus exception or data parity error occurs during a List Chaining operation. It is used together with Bits 6 and 7 to determine whether the data transfer was performed successfully, but the list chain was corrupted. The setting of this bit is mutually exclusive with the setting of Bit 9. Note that either Bit 6 or 7 can be set concurrently with this bit.
- Bit 7: Exception. This bit is set when the -L_EXCPT signal is detected during a Bus Master transfer.
- Bit 6: Local Bus Parity. This bit is set when a Local Bus Parity error is detected during a Bus Master read. Note that this bit can never be set when Local Bus Parity is disabled in the PROC_CFG Register.

- Bit 5: Loss of Channel Indicator. This bit is set when the Bus Master loses the Micro Channel. The Micro Channel is lost by: (1) the raising of the ARB/GNT signal on the Micro Channel during Bus Master operation, (2) resetting of the Card Enable (POS 2, Bit 0), or (3) resetting of the DMA Enable (SCP, Bit 1). All of these conditions stop any Bus Master activity. Enabling of the Bus Master channel is also blocked if either the Card Enable or the DMA Enable are reset. Attempting to start the channel; that is, setting Bit 0 in the Channel Control Register (CCR), with either of these bits disabled will force a termination interrupt and the setting of this status bit.
- Bit 4: Invalid Combination (-DS16RTN, -DS32RTN). This bit indicates the invalid combination, -DS32RTN asserted, -DS16RTN negated, is active on the Micro Channel. This condition is checked on the Micro Channel Interface Chip Bus Master transfers.
- Bit 3: Card Selected Feedback Return Indicator. This bit is set when the card selected feedback return signal on the Micro Channel is not detected during a Bus Master cycle. Detection of the -SFDBKRTN signal is enabled by POS 3, Bit 6. Note that the interrupt associated with this bit can never be asserted when Card Selected Feedback Return detection is disabled in the POS registers. However, this bit provides accurate status of the Selected Feedback Return Detection **when -CHCK is asserted during a Bus Master operation**. This status is set **independent of the enabling of -SFDBKRTN detection in POS**.

  The setting of this status for -CHCK provides an indication of whether a slave device was selected during the cycle receiving a -CHCK. This is especially useful for Micro Channel address parity errors. Micro Channel slaves, by definition, do not set -CHCK status in the POS registers when detecting an address parity error. Therefore, this bit can be used in conjunction with the -CHCK Indicator (Bit 2, BMSTAT), to isolate this type of error.
- Bit 2: -CHCK Indicator. This bit is set when -CHCK is asserted during a Bus Master operation on the Micro Channel; that is, while the Bus Master is granted the Micro Channel.
- Bit 1: Data Parity Error.

  This bit is set when a Micro Channel Data Parity Error has occurred. Micro Channel address and data parity detection is enabled by POS 3, Bit 5. Note that this bit can never be set when Micro Channel Parity is disabled in the POS registers.
- Bit 0: Normal Termination. This bit is reset by a read of the BMSTAT Register, or by a list chain operation. Note that if Normal Termination interrupts are enabled, the interrupt associated with the setting of the bit is ***not*** reset by a list chain operation, but can only be reset by a read of the BMSTAT Register.

Reset Conditions

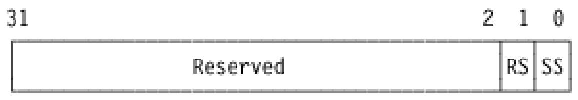
```
        Reset: UUUU UUUU UUUU UUUU UUUU UUUU UUUU UUUU
Command Reset: SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSSS
```

Bus Master Command Register, BMCMD1 (2)

This register is used to start the Bus Master channel or to reset internal Bus Master registers and control logic to a known state. During normal operation it is not necessary to use this register.

Register Format

```
(Channel 1 Address = 1FFA 301Ch)          32-bit read/write
(Channel 2 Address = 1FFA 401Ch)          32-bit read/write
```



Bit Descriptions

- Bits 31–2: Reserved. These bits should always be set to 0.
- Bit 1: Reset State Bit. Setting this bit resets the channel’s internal registers and control logic, and holds the channel in this reset state. Resetting this bit releases the channel from the reset state for further operation. Setting this bit while the start/stop bit is set, will force a non-catastrophic stop of the channel, in addition to resetting the channel. Stopping the channel in this manner is not supported in conjunction with List Chaining.
- Bit 0: Start/Stop Bit. This Start/Stop bit is an alternate access to the Start/Stop bit in the CCR. This bit provides the ability to start the channel without corrupting the contents of the other bits in the CCR. More information concerning the Start/Stop bit is available in the *Channel Control Register (CCR)* on page 93.

Reset Conditions

```
             Reset: 0000 0000 0000 0000 0000 0000 0000 0000
     Command Reset: 0000 0000 0000 0000 0000 0000 0000 0000
ROM Initialization: 0000 0000 0000 0000 0000 0000 0000 0000
```

Shared Memory

The Micro Channel Interface Chip provides a Micro Channel slave interface, supporting 100-nanosecond streaming data through posted memory writes and prefetched memory reads. Micro Channel Bus Masters access slave-resident memory through a Micro Channel shared memory window.

Access to Local Bus address space is available from the Micro Channel memory space. There are two independent windows into Local Bus address space from the Micro Channel memory map. These windows are configured in POS and are separately enabled by the 80960 processor in the PROC_CFG Register. See *Processor Configuration Register (PROC_CFG) (Micro Channel)* on page 78.

ROM/RAM Area

- The first window is available in the ROM/RAM area. The intent of this window is for use by applications requiring BIOS extensions in this area of the Micro Channel address space. The window is configured in POS Register 2, bits 2–5. These bits allow the window to be placed on any 8-KB boundary within the ROM/RAM area. In addition, the size of the window can be selected between 8KB and 16KB. Independent of the size, the first location in this window points to the first location in packet memory, located at the Local Bus address programmed in the Local Bus Base Address Register (LBBAR). The AIX Support Program does not use this hardware feature.

16KB windows must be on 16KB boundaries.

Full Memory Window

In a PS/2, the second window maps up to a 64MB window in Local Bus address space, above the 1MB boundary in Micro Channel address space. Address bits A31–A29 and A23–A20 are configurable in POS to locate the window on various 1MB boundaries above and below the 16MB boundary. The size of the full memory window is set by the Memory Window Size bits (MW2–0) in the PROC_CFG Register.

In a RISC System/6000, the size of the second window defaults to 32MB.

Independent of the size, the first location in this window points to the Local Bus address programmed in the Local Bus Base Address Register (LBBAR).

The memory window must be placed on a boundary equal to the size of the available memory as set in the PROC_CFG Register. For example, for 8MB of memory (MW2–0 = 100), A22–A20 must equal zero.

Address bits A28–A24 are configurable in the POS Subaddress Register 3A, allowing additional options for window placement. These bits are set to zero on a Power-Up Reset, allowing systems that do not support subaddressing to configure the Micro Channel Interface Chip through POS 4 only.

A memory map showing the relationship between the Micro Channel memory windows and Local Bus address space is given in Figure 8-3.

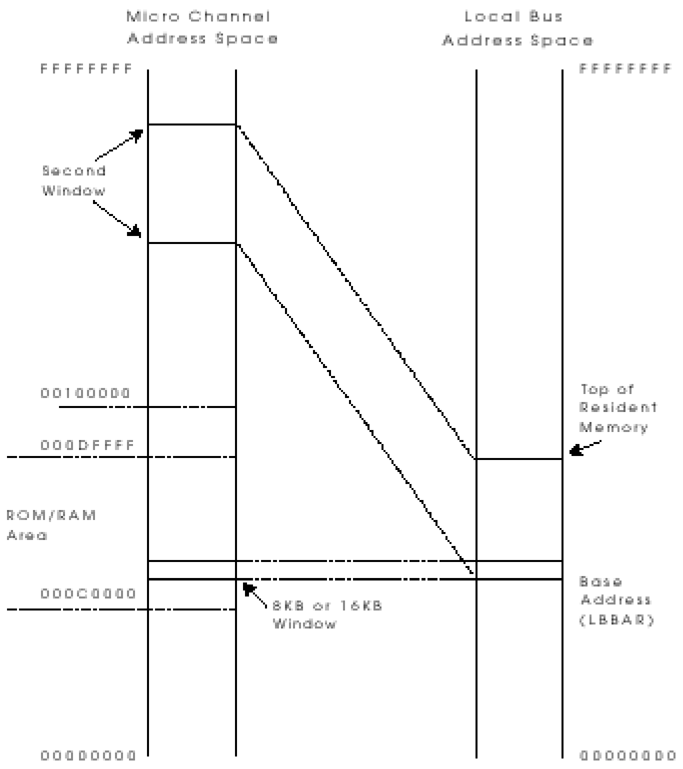


Figure 8-3. Shared Memory Windows

Bursting/Non-Bursting Local Bus Addresses

Transfers on the Local Bus are generally burst accesses. Multiple data transfers occur for one initial address cycle with termination of the transfer indicated by the Local Bus Master’s assertion of the Burst Last (-BLAST) signal. To allow interfacing to devices that do not support this bursting capability, the CFE Architecture provides an area in the Local Bus address space that is reserved for non-burst access. Mapping of separate regions for Bursting and Non-Bursting access is similar to the configuration of the Intel 80960 Bus Controller access into separate 256MB address regions.

From the host, through the Micro Channel Interface Chip, this non-bursting area can be accessed through the two shared memory windows, by the DMA channels, through the MDATA Register, and through POS subaddress space. In addition, for Micro Channel slave read access to this area through the MDATA Register, the prefetch buffer associated with Micro Channel slave read accesses is filled by single transfers; that is, the prefetch mechanism is disabled. This non-bursting, non-prefetching mechanism provides the ability to access Local Bus address areas that are affected by the prefetch mechanism. One example of this type of address area is self-clearing registers that may be corrupted by a prefetch read during an access to a contiguous address. Another example is a protected memory containing boundaries that may be crossed during a prefetch associated with a read from a contiguous location. Note that all POS subaddress accesses are both non-bursting and non-prefetching.

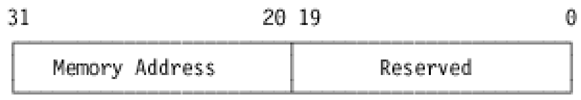
Slave Memory Registers

Local Bus Base Address Register (LBBAR)

The Local Bus Base Address Register supplies the base address in Local Bus address space for the Micro Channel shared memory window options. This register provides the address bits A31–A20, placing the full memory window on any 1MB boundary. This register must be initialized before enabling either the full memory or the 8KB BIOS window in the PROC_CFG Register. More information on shared memory options can be found in *Shared Memory* on page 100.

Register Format

```
(Local Bus Address = 1FFA 0024h)          32-bit read/write
```



Bit Descriptions

- Bits 31–20: Address Bits A31–A20.
- Bits 19–10: Reserved. These bits must be set to 0.

Reset Conditions

```
             Reset: UUUU UUUU UUUU 0000 0000 0000 0000 0000
     Command Reset: SSSS SSSS SSSS 0000 0000 0000 0000 0000
ROM Initialization: 0010 0000 0000 0000 0000 0000 0000 0000
```

Slave I/O Access to Local Bus Address Space

The Host-Slave Base Address and the MDATA Register may be used by the Micro Channel to access co-processor Local Bus address space. A description of this mechanism follows.

Programming Considerations

When reading Local Bus addresses in the non-burst region through the MDATA Port, Local Bus exceptions and read parity errors are not reported with -CHCK. If the CD CHRDY Timeout Disable is disabled, a timeout error can occur on the Micro Channel when one of these exceptions occurs.

Slave I/O Registers

Host-Slave Base Address Register (HSBR)

The Host-Slave Base Address Register stores the physical base address in *slave-resident memory* that is accessed by the host. The host initializes this register before accessing the Memory Data Register (MDATA) for memory transfers. Continuous accesses from the MDATA Register transfer data to, or from, subsequent memory locations based on the value in this register. The value in the HSBR is auto-incremented by the number of bytes transferred. It is possible to write to the Micro Channel Interface Chip's internal registers through this mechanism; these must be 32-bit accesses.

The MDATA Register supports Micro Channel basic transfers only, that is, streaming data is not supported. More information on the Memory Data Register is available in the *Memory Data Register (MDATA)* on page 103.

For example, to read the Memory Controller Chip's Memory Configuration Register at 1FFB A000h using DOS Debug, the following commands would be used:

o 1C0C 00

o 1C0D A0

o 1C0E FB

o 1C0F 1F

And then:

i 1C10 (low-order byte of register)

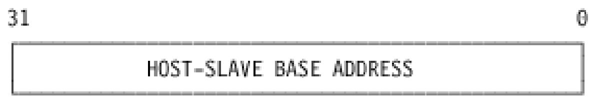
i 1C11

i 1C12

i 1C13 (high-order byte of register)

Register Format

```
(Micro Channel Address = Base + 0C-0Fh)    32-bit read/write
```



Reset Conditions

```
        Reset: UUUU UUUU UUUU UUUU UUUU UUUU UUUU UUUU
Command Reset: SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSSS
```

Memory Data Register (MDATA)

The Memory Data Register is used in conjunction with the Host-Slave Base Address Register (HSBR), for Micro Channel accesses between the Micro Channel host and resident memory. When the Memory Data Register is accessed, depending on the direction of transfer, data is written to, or read from, the location in resident memory stored in the HSBR. For host transfers, subsequent accesses to the MDATA Register after initialization of the HSBR are made to subsequent locations in memory; that is, the HSBR is auto-incremented after every transfer.

Operation

The intended use of the HSBR and the MDATA Register is for small transfers between the host system and the Local Bus. As a result, data steering to the MDATA Register is limited. Although the Host Slave Base Address Register (HSBR) is used for addressing the Local Bus, the MDATA Register does not provide data steering based on the value stored in this register. Rather, the MDATA Register passes data to the Local Bus as it is steered on the Micro Channel, based on the Micro Channel address used to access this register.

The MDATA Register can be accessed on the Micro Channel as a single 32-bit register, two 16-bit registers, or four 8-bit registers. Therefore, to access individual bytes on various address boundaries, the address used to access the MDATA Register must match the alignment of the value stored in the HSBR. That is, the two least significant bits of the MDATA Register's I/O address must match the least significant bits of the HSBR. For example, to access an odd word in Local Bus address space, that is, an address located on an odd-word boundary, the HSBR is first loaded with this odd word address. To access the odd word, the MDATA Register is subsequently accessed as a word register at Micro Channel location 0012h. To access a byte at this Local Bus location, the MDATA Register is accessed as a byte register at location 0012h.

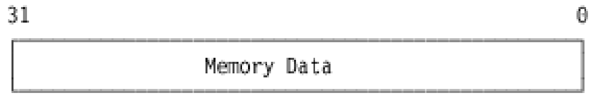
As a result of this limited data steering, the following limitations are placed on MDATA Register access:

- ***Streaming Data Accesses*** – Accessing the MDATA Register does not result in a Micro Channel streaming data request.
- ***Word and Byte Burst Accesses*** – Micro Channel burst access is supported for word and byte access, but the bursting device must be able to alternately access the MDATA Register at word or byte locations to support data steering. For example, to burst word transfers starting on a 4-byte boundary, the bursting master must first address the register as a word at location 0010h and subsequently at location 0012h. The value in the HSBR will be auto-incremented.
- ***Word and Byte String Instructions*** – Related to streaming data and burst access, string instructions such as string OUTs and INs to I/O locations are not supported. These instructions support multiple accesses to a single I/O location, and therefore, cannot support the address changes necessary for data steering.

Note that read accesses through the MDATA Register to the non-bursting area in Local Bus address space limits the prefetch buffer to single transfers. That is, only single transfers will occur on the Local Bus, and -CDCHRDY will be asserted on the Micro Channel for ***each*** slave access. More information on the non-bursting address space can be found on page 101.

Register Format

```
(Micro Channel Address = Base + 10h)           32-bit read/write
(Micro Channel Address = Base + 10,12h)        16-bit read/write
(Micro Channel Address = Base + 10,11,12,13h)  8-bit read/write
```



Reset Conditions

```
        Reset: UUUU UUUU UUUU UUUU UUUU UUUU UUUU UUUU
Command Reset: SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSSS
```

Subsystem Control Block (SCB) Support

The co-processor supports the Move Mode of the Subsystem Control Block (SCB) architecture. The SCB registers and their access are shown in Table 8-5.

Table 8-5. SCB Register Access Summary

SCB Register	Move Mode		
	System Master	Peer Adapter	Local bus
COMMAND	W*	None	R*
ATTN	W	W	R
SCP	W	None	R
ISP	R**	None	W**
CBSP	R	None	W

* Under consideration for Move Mode—not approved

** Currently, no function defined for Move Mode

The overall objective of the SCB architecture is to provide a programming model for the Micro Channel by defining the logical protocols for transferring commands, data, and status between entities on the Micro Channel.

The Subsystem Control Port is accessed by the Micro Channel, and is used to enable various functions of the Micro Channel Interface Chip. This register is used to enable Micro Channel interrupts, to enable DMA operation, to perform a Command Reset of the Micro Channel Interface Chip, and to reset the Reject state of the Command Busy Status Port.

In addition to these registers, the ISP and SIR Registers support Micro Channel and additional Local Bus interrupts.

Move Mode

Move Mode is used for peer-to-peer operation across the Micro Channel. In general, commands are not passed through the Command Port but to a predefined location in a shared memory area. The Attention Port is used to signal an interrupt to the Local Bus. An attention code of ‘D’ hex is used to identify the transfer as a Move Mode operation. A bit in the SIR is set, corresponding to the device number passed with this Move Mode attention code. A full description of the operation of the SIR is available in the *Source Identification Register (SIR)* on page 113.

Interrupts

Micro Channel interrupts (that is, interrupts presented to the Micro Channel from the Micro Channel Interface Chip) are asserted by setting the IV Bit in the CBSP Register from the Local Bus. For signaling interrupts, status is provided by the resident processor in the Interrupt Status Port (ISP). This status is provided prior to setting the Interrupt Valid Bit. For exception conditions, the Reject Bit in the CBSP Register is set along with the IV Bit, and status for the exception is provided in the Status Bits (Bits 5–7) of the CBSP Register. Separation of exception and signaling conditions is thus obtained from one read of the CBSP Register from the Micro Channel.

Micro Channel interrupts are cleared in two different ways, dependent on the state of the Clear on Read Bit (COR, Bit 6) in the SCP Register. When this bit is reset, Micro Channel interrupts are cleared in hardware by writing the End of Interrupt command (E0h) to the Attention Port. This attention code/device number resets the interrupt in hardware, as well as the IV Bit in the CBSP Register. When the COR Bit is set, Micro Channel interrupts are cleared by reading the Command Busy Status Port from the Micro Channel. When read, the Interrupt Valid Bit is reset, as well as the pending interrupt.

Exceptions

Exceptions are reported by setting the Reject Bit, together with the Status Bits in the CBSP Register, from the Local Bus. There is only one Status code for hardware-defined exceptions, '001'b. This exception is asserted for Watchdog Timeout errors, that is, when the Watchdog Timeout signal is detected asserted by the Micro Channel Interface Chip. For this exception, the Reject and Interrupt Valid Bits (as well as Status) are set in hardware, and an interrupt is asserted to the Micro Channel.

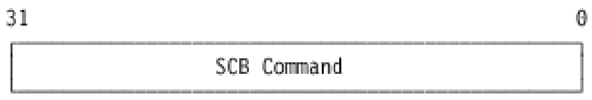
SCB Registers

Command Port (COMMAND)

The Command Port is used to deliver a 32-bit immediate command or the physical address of a control block to a feature adapter on the Micro Channel. The Command Port is protected by both the Reject Bit (Bit 4) and the Busy Bit (Bit 0) in the Command Busy Status Port. If either of these two bits are set, the Command Port write will be ignored. More information on control block architecture can be found in the *Subsystem Control Block (SCB) Support* on page 105.

Register Format

```
(Local Bus Address = 1FFA 2000h)          32-bit read-only
(Micro Channel Address = Base + 00h)     32-bit read/write
```



Bit Descriptions

Bits 31–0: Immediate Command or Address of a Control Block.

Reset Conditions

```
        Reset: UUUU UUUU UUUU UUUU UUUU UUUU UUUU UUUU
Command Reset: SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSSS
```

Attention Port (ATTN)

The Attention Port is used by the Micro Channel to signal, or request, the attention of the resident processor. A Micro Channel write to the Attention Port causes data to be latched in the Attention Port and an attention interrupt to be posted to the resident processor. More information on attention interrupts is available in the *System Bus Interface Chip Interrupts* on page 115. The Attention Port is protected by both the Reject Bit (Bit 4) and the Busy Bit (Bit 0) in the Command Busy Status Port. If either of these two bits is set, the Attention Port write will be ignored.

Attention Code ‘D’

The operation of attention code ‘D’ is different from all other attention codes. This attention code is used in Move Mode to set individual bits in the Source Identification Register (SIR). Either a General Interrupt or an End-of-Data-Transfer Interrupt is issued in place of the Attention Interrupt. The setting of bits in the SIR with this attention code is ***not*** blocked by the Reject or Busy bits. More information on the setting of these bits and the subsequent interrupts is available in the *Source Identification Register (SIR)* on page 113 and *Subsystem Control Block (SCB) Support* on page 105.

Attention Code 'E'

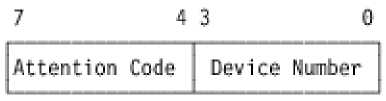
Attention Code ‘E’ is used to clear the IV Bit in the Command Busy Status Port (CBSP) and its Micro Channel interrupt. The operation of the attention code is controlled by the

Clear on Read Bit (Bit 6) in the Subsystem Control Port (SCP). Note that this interrupt also generates an attention interrupt to the resident processor and sets the Busy Bit.

More information on the Command Busy Status Port is available in the *Command Busy Status Port (CBSP)* on page 112. General information on the Subsystem Control Block (SCB) architecture is available in the *Subsystem Control Block (SCB) Support* on page 105.

Register Format

```
(Local Bus Address = 1FFA 2004h)         32-bit read-only;
8-bit field defined by SCB architecture
(Micro Channel Address = Base + 04h)     8-bit read/write
```



Bit Descriptions

- Bits 7–4: Attention Code. The attention code is used to indicate to the resident processor the specific action to be initiated. For SCB Move Mode, the attention code ‘D’ hex is used to set bits in the SIR.

  Valid attention codes are shown in Table 8-6.

Table 8-6. Attention Codes

Attention Code	Device Number	Name	Description
0	X	Reset Command	Request the subsystem to perform Device Reset for the specified device.
1	X	Immediate Command	Requests the subsystem to execute the command contained in the Command port.
2	X	None	Reserved
3	X	Start Control Block Command	Requests the subsystem to process the control block pointed to by the address in the Command port.
4		Device-Dependent	Not Applicable
5–C		None	Reserved
D	0–F	Move Mode Delivery	Used to signal request for Move Mode command delivery. The device number specifies the bit to be set in the SIR.
E	0	End of Interrupt	Requests the subsystem to perform one of the two interrupt resetting commands.
F		Device-Dependent	Not Applicable

X = Do not care. A blank in the Device Number equals unspecified.

- Bits 3–0: Device Number. The device number indicates a specific device to which the attention code is directed. For SCB Move Mode, this device number, together with an attention code of 'D' hex, indicates the bit to be set in the SIR. More information concerning the setting of bits in the SIR is available in the *Source Identification Register (SIR)* on page 113.

Reset Conditions

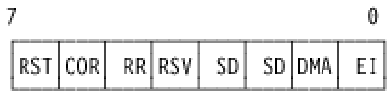
```
        Reset: UUUU UUUU
Command Reset: SSSS SSSS
```

Subsystem Control Port (SCP)

The Subsystem Control Port is used to support several hardware commands from the Micro Channel. Information on the Subsystem Control Block architecture is available in the *Subsystem Control Block (SCB) Support* on page 105.

Register Format

```
(Local Bus Address = 1FFA 2008h)     32-bit read-only;
8-bit field defined by SCB architecture
(Micro Channel Address = Base + 05h)  8-bit read/write
```



Bit Descriptions

- Bit 7: Command Reset. Setting this bit causes a Command Reset. The Command Reset Out signal is asserted. In addition, the Busy Bit in the Command Busy Status Port is active during this reset. This bit should be set for a minimum of 50 microseconds before being reset. Setting this bit does not affect the POS registers. Register values after this reset are provided in the individual register descriptions. Note that while this bit is set, all Micro Channel Interface Chip registers will read back as 0FFh, that is, the chip and all devices dependent on COMMAND RESET OUT are held in a reset state. On the co-processor, these devices are the Interrupt Controller, the processor and the Memory Controller Chip. COMMAND RESET OUT is also driven to the AIB connector.
- Bit 6: Clear on Read Bit. This bit sets the method of clearing the IV Bit. When reset, the IV Bit is cleared by a command (E0h) written to the Attention Port. When set, the IV Bit is cleared by reading the Command Busy Status Port. Although this bit is reserved by SCB architecture, it is read/write accessible from the Micro Channel, and read-only accessible from the Local Bus.
- Bit 5: Reset Reject. Setting this bit resets the Busy Bit and the Reject Bit in the Command Busy Status Port. This bit is self-clearing.
- Bit 4: Reserved. This signal is read/write accessible from the Micro Channel and read-only accessible from the Local Bus.
- Bits 3–2: Device Dependent. This signal is read/write accessible from the Micro Channel and read-only accessible from the Local Bus.
- Bit 1: Enable DMA. Setting this bit enables Bus Master operation.
- Bit 0: Enable Interrupts. Setting this bit enables interrupts to the Micro Channel.

Reset Conditions

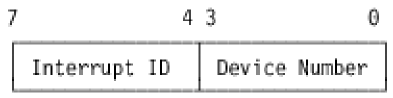
```
        Reset: 0000 0000
Command Reset: 1000 0000
```

Interrupt Status Port (ISP)

The Interrupt Status Port is used to present interrupt data to the Micro Channel. General information on the Subsystem Control Block (SCB) architecture and interrupt handling is available in the *Subsystem Control Block (SCB) Support* on page 105.

Register Format

```
(Local Bus Address = 1FFA 200Ch)     32-bit read/write;
8-bit field defined by SCB architecture
(Micro Channel Address = Base + 06h)  8-bit read-only
```



Bit Descriptions

- Bits 7–4: Interrupt Identifier. These bits identify the cause of the interrupt. Valid interrupt codes are shown in Table 8-7.

Table 8-7. Interrupt Identifier Codes

Hex Value	Interrupt Definition
0	Reset Subsystem/Device Completed with No Error
1	Control Block Command Completed with No Error
2	Notify Event
3	Reserved
4	Reserved
5	Device-Dependent
6	Inform Event
7	Hardware Failure Immediate Command or Hardware Control
8	Hardware Failure Control Block Command
9	Reserved
A	Immediate Command/Hardware Control Completed with No Error
B	Reserved
C	Control Block Command Completed with an Error
D	Immediate Command/Hardware Control Completed with an Error
E	Command Rejected
F	Device-Dependent

- Bits 3–0: Device Number. The device number identifies the specific device providing the interrupt status.

Reset Conditions

```
        Reset: UUUU UUUU
Command Reset: SSSS SSSS
```

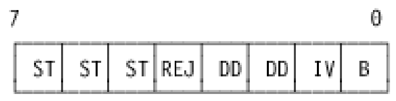
Command Busy Status Port (CBSP)

The Command Busy Status Port has two functions:

- It is read by the Micro Channel after an immediate command or Control Block address is written to the Command Port to determine the status of the command.
- It indicates that an interrupt to the Micro Channel is pending.

Register Format

```
(Local Bus Address = 1FFA2010h)     32-bit read/write;
8-bit field defined by SCB architecture
(Micro Channel Address = Base + 07h) 8-bit read-only
```



Bit Descriptions

- Bits 7–5: Status. These bits are used to encode a rejection code for the current command. When the Reject and Busy Bits are set, these bits reflect status for exception conditions. Only one exception condition is defined in hardware, '001'b. This exception indicates that the Watchdog Timeout signal has been detected by the Micro Channel Interface Chip.
- Bit 4: Reject. This bit indicates that the current command has been rejected. Micro Channel writes to the Attention and Command Ports are blocked when this bit is set, with the exception of the ‘D’ hex attention code. (For more information on attention codes, refer to *Attention Port (ATTN)* on page 107 and *Subsystem Control Block (SCB) Support* on page 105.) If this bit is set, together with the IV Bit, an exception is reported in the Status bits (Bits 7–5).
- Bits 3–2: Device-Dependent. These bits are device-dependent.
- Bit 1: Interrupt Valid. Setting this bit forces an interrupt to the Micro Channel. If this bit is set and the Reject Bit (Bit 4) is reset, the bit indicates that interrupt status is available in the Interrupt Status Port (ISP). If this bit is set and the Reject Bit is set, a status code is available in the status bits of the Command Busy Status Port (CBSP). The ISP is used for normal signaling interrupts, and the CBSP is used under exceptions conditions. Only one exception status code is defined for hardware.

  The resetting of this bit is under control of the Clear on Read Bit (Bit 6, SCP). The definition of this control bit is available in *Subsystem Control Port (SCP)* on page 110. More information on the SCB and interrupt support is available in the *Subsystem Control Block (SCB) Support* on page 105.
- Bit 0: Busy. This bit indicates that Command and Attention Ports are currently being used, or a hardware control Subsystem Reset is either in progress or has just

completed. This bit is set by a write to the Attention Port. Micro Channel writes to the Attention and Command Ports are blocked when this bit is set.

Reset Conditions

```
        Reset: 0000 0001
Command Reset: SSS0 SS01
```

Source Identification Register (SIR)

The Source Identification Register is used in Move Mode only. The register is directly accessible from the Local Bus only. Bits in the SIR are set indirectly through the Attention Port on the Micro Channel. Specifically, a write to the Attention Port of attention code 'D' hex indicates the setting of a bit in the SIR. The device number passed in the Attention Port indicates the bit to be set in the SIR. The bits in the SIR are identified from the least significant bit to the most significant bit as 0–F hex. A write to the Attention Port of ‘DA’ hex would set bit 10 in the SIR.

By Local Bus definition, and independent of SCB architecture, the Source Identification Register is used to issue both general interrupts and end-of-transfer interrupts from the Micro Channel to the resident processor, saving status for these interrupts.

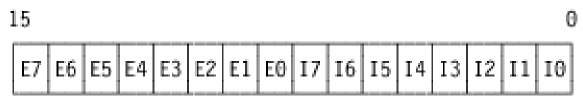
Bits in the SIR can only be set from the Micro Channel. Each bit is set through the Attention Port. Interrupts are generated by the logical OR of the bits in the SIR. A General Interrupt is generated by the logical OR of the lowest eight bits in the register; an End-of-Data-Transfer Interrupt is generated for the upper eight bits in the register.

Bits in the SIR can only be reset from the Local Bus. Each bit is reset by writing a logical 0 to the corresponding bit location. Writes of 1 from the Local Bus to the SIR are ignored.

This manner of resetting bits allows the Micro Channel to generate interrupts to the resident processor through the SIR without losing interrupt status. More information on Local Bus interrupts is available in the *System Bus Interface Chip Interrupts* on page 115. More information on the Subsystem Control Block (SCB) architecture is available in the *Subsystem Control Block (SCB) Support* on page 105.

Register Format

```
(Local Bus Address = 1FFA 2014h)     32-bit read/write;
16-bit field defined by SCB architecture
(Micro Channel Address = Indirect Access through ATTN)
```



Bit Descriptions

- Bits 15–8: End-of-Data-Transfer Interrupt Bits.
- Bits 7–0: General Interrupt Bits.

Reset Conditions

```
               Reset: 0000 0000 0000 0000
       Command Reset: 0000 0000 0000 0000
ROM Initialization: 0000 0000 0000 0000
```

Miscellaneous Micro Channel Registers

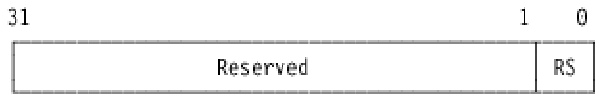
The Micro Channel Interface Chip handles two miscellaneous functions involving the interface between the co-processor and the host system. The Reset Status Register is used to distinguish between Command Resets and other resets. The Non-Maskable Interrupt Register is used to present NMIs from the system to the 80960 processor.

Reset Status Register (RSR)

The Reset Status Register supplies status for a reset of the Micro Channel Interface Chip. A Command Reset sets a bit in this register. This bit is reset by a Micro Channel Reset. The intent of the Reset Status Register is to provide a means in software of distinguishing between Power-Up or Micro Channel Resets, and Command Resets.

Register Format

```
(Local Bus Address = 1FFA 0014h)          32-bit read-only
```



Bit Descriptions

- Bits 31–1: Reserved. These bits must be set to zeros.
- Bit 0: Reset Status.

  This bit indicates that a warm reset has occurred on the Micro Channel.

Reset Conditions

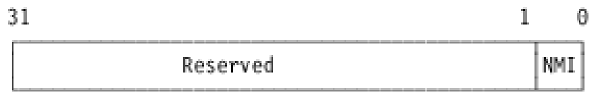
```
        Reset: 0000 0000 0000 0000 0000 0000 0000 0000
Command Reset: 0000 0000 0000 0000 0000 0000 0000 0001
```

Non-Maskable Interrupt Register (NMI)

The Non-Maskable Interrupt Register is used for presenting a Non-Maskable Interrupt from the Micro Channel to the resident processor. Writing the register from the Micro Channel with any non-zero data creates an interrupt of value 0 hex on the encoded interrupt lines, INT 3–0, as well as setting a status bit in Bit 0 of the register. Reading the register clears this status. The intent of the register is to provide a high-priority interrupt separate from the Attention and SIR Interrupts. Any priority could be assigned this interrupt, however, depending on the implementation of the interrupt controller. More information on Local Bus interrupts is available in *System Bus Interface Chip Interrupts* on page 115.

Register Format

```
Local Bus Address = 1FFA 001Ch)          32-bit read-only
(Micro Channel Address = Base + 0Bh)      8-bit read/write
```



Bit Descriptions

- Bits 31–1: Reserved.
- Bit 0: NMI Status Bit. This bit indicates that a high-priority interrupt is pending from the Micro Channel.

Reset Conditions

```
        Reset: 0000 0000 0000 0000 0000 0000 0000 0000
Command Reset: 0000 0000 0000 0000 0000 0000 0000 0000
```

System Bus Interface Chip Interrupts

The Micro Channel Interface Chip interrupts the Local Bus under seven conditions. This chip presents each of these seven interrupt sources as encoded interrupts to the Interrupt Controller.

Each encoded interrupt is held active on the interrupt lines until the clearing procedure for that interrupt is performed. All other interrupts asserted internally are held pending until the current interrupt is cleared. After an interrupt is cleared, all pending interrupts are latched, and the highest priority pending interrupt is asserted on the encoded interrupt lines. After this interrupt is cleared, the next priority interrupt is serviced. This operation continues until all interrupts previously pending are serviced in order of priority. When the last interrupt is cleared, all new interrupts that were asserted internally are latched, while the current interrupts are serviced. Then, the highest-priority interrupt is asserted first. Interrupts for conditions occurring simultaneously are presented in the order of their encoded value. For example, if a General Interrupt occurs simultaneous to a Bus Master Channel interrupt, the Interrupt Command would be presented first. The only exception to this procedure is the NMI command. This interrupt is always presented as the next available interrupt, regardless of the order of occurrence.

The clearing procedure varies with the source of the interrupt:

- ***Non-Maskable Interrupts and Local Bus Parity/Exception Interrupts.*** The interrupt is cleared when the corresponding status register is read.
- ***SCB Attention Port Interrupts.*** The interrupt is cleared when the Attention Port is read.
- ***EOT and General Interrupts.*** The interrupt is cleared when the SIR is written from the Local Bus, resetting a bit. The source of each of these interrupts is the OR of their respective status bits. The interrupt, therefore, will be reasserted after a write from the Local Bus, until the OR of the status bits is equal to zero; that is, all status bits have been reset.
- ***Bus Master Channel 1 and Channel 2 Interrupts.*** The interrupt is cleared by reading the associated channel status register, BMSTAT1(2).

PCI Interface Subsystem

9

The PCI interface subsystem is contained within the PCI System Bus Interface Chip. The major functions of this chip are as follows:

- Is operationally compatible with the Micro Channel Interface Chip
- Features two Bus Master channels, addressable from the CFE Local Bus
- Contains one 128-byte intermediate data buffer per channel
- Supports Posting Status from each Bus Master Channel
- Supports Linked List Chaining for auto-initialization of either channel
- Contains a slave write buffer (128 bytes) for PCI Posted Write Operation
- Contains a slave prefetch read buffer (128 bytes) for PCI Delayed Read Operation
- Contains Master and Slave Support of PCI Bus Commands
- Contains hardware support for SCB Locate and Move mode
- Supports access to the CFE Local Bus address space from both PCI I/O and memory address spaces
- Supports PCI data and address parity
- Supports PCI interrupts and error reporting
- Is directly attachable to PCI
- Supports JTAG for testing of In-Circuit connections.

The PCI System Bus Interface Chip provides a PCI target interface, supporting posted memory writes and delayed/prefetched memory reads to the CFE Local Bus address space. These accesses are mapped to a memory window on the PCI bus. A PCI master, or initiator, can access CFE Local Bus address space through this PCI slave interface. The PCI host accesses the CFE Local Bus through this memory window or through a PCI I/O location known as the Memory Data (MDATA) port. The PCI slave interface also supports a register space, mapped to both memory and I/O space on the PCI Bus.

The PCI Bus Master interface consists of two DMA or Bus Master channels addressable from the CFE Local Bus or indirectly from the PCI Bus through the MDATA port. The term *Bus Master Channel* refers to these two DMA channel interfaces to the PCI System Bus Interface Chip. Each channel utilizes a 128-byte intermediate buffer between the CFE Local Bus and the PCI Bus.

PCI System Bus Interface Chip Signal Description

The following tables present a description of each signal in the PCI System Bus Interface Chip chip. Notation for the PCI bus is consistent with the notation found in the related documentation for the bus. For example, the active low state of a signal is represented by a # following the signal name in PCI notation.

Table 9-1. PCI System Bus Interface Chip Pin Description – PCI Signals (Alphabetical Order)

Name	Type	Description
AD(31::0)	I/O	Address/Data Bits 31-0 are used by the PCI initiator to address memory and I/O targets. These signals are also used to select the PCI System Bus Interface Chip chip for target operations.
C/BE(3::0)#	I/O	Bus Command and Byte Enables are used to define a bus command during the address phase of a PCI cycle, and to select the active bytes during the data phase of PCI cycle. These signals are driven by PCI System Bus Interface Chip as an initiator and received by PCI System Bus Interface Chip as a target.
PAR	I/O	Parity bit provides a single bit of even parity across AD(31::0) and C/BE(3::0). PCI System Bus Interface Chip drives PAR as an initiator for address phases and write data phases, and checks PAR as a target for read data phases. PCI System Bus Interface Chip checks PAR as a target for address phases and write data phases, and drives PAR as a target for read data phases.
FRAME#	I/O	-Cycle Frame indicates the beginning and duration of an access. This signal is driven by PCI System Bus Interface Chip as an initiator and received by PCI System Bus Interface Chip as a target.
IRDY#	I/O	-Initiator Ready indicates the initiator’s ability to complete the current data phase of the transaction. This signal is driven by PCI System Bus Interface Chip as an initiator and received by PCI System Bus Interface Chip as a target.
TRDY#	I/O	-Target Ready indicates the target’s ability to complete the current data phase of the transaction. This signal is driven by PCI System Bus Interface Chip as an target and received by PCI System Bus Interface Chip as a initiator.
STOP#	I/O	-Stop indicates the current target is requesting the initiator to stop the current transaction. This signal is driven by PCI System Bus Interface Chip as an target and received by PCI System Bus Interface Chip as a initiator.
IDSEL	I	Initialization Device Select selects PCI System Bus Interface Chip. during configuration read and write transactions.
DEVSEL#	I/O	-Device Select is driven by PCI System Bus Interface Chip. to indicate that PCI System Bus Interface Chip. is selected as a target. This signal is received by PCI System Bus Interface Chip as an initiator to indicate that a device has been selected.
REQ#	O	-Request is driven by PCI System Bus Interface Chip. to indicate to the system arbiter that PCI System Bus Interface Chip. wants use of the PCI bus.
GNT#	I	-Grant is received by PCI System Bus Interface Chip. to indicate that PCI System Bus Interface Chip has been granted ownership of the PCI bus.

Table 9-1. PCI System Bus Interface Chip Pin Description – PCI Signals (Alphabetical Order)

Name	Type	Description
PERR#	I/O	-Parity Error is used to report data parity errors during all PCI transactions (except Special Cycle). PCI System Bus Interface Chip. asserts this signal when detecting a parity error on received data, and receives this signal when driving data.
SERR#	I/O	-System Error is used to report address parity errors on the Special Cycle command, or for any other catastrophic error. This signal is driven and received by PCI System Bus Interface Chip.
INTA#	O	-Interrupt is used to request an interrupt on the PCI bus.
CLK	I	Clock provides timing for all transactions on the PCI bus.
RST#	I	-Reset is used to bring PCI specific registers, sequencers, and signals to a consistent state.

Table 9-2. PCI System Bus Interface Chip Pin Description—Miscellaneous

Name	Type	Description
SD	I/O	Serial EPROM Data is used to provide the address (Start bit, opcode, address) to a serial EPROM and to receive read data from the EPROM. This pin is tied to both the Data In (DI) and Data Out (DO) of serial EPROM chip (for example, Microchip 93C06). This pin has a weak pullup resistor internal to the chip.
SCS	O	Serial EPROM Chip Select is used to select a serial EPROM.
SCLK	O	Serial EPROM Clock is used to clock a serial EPROM chip.

PCI System Bus Interface Chip Register Description

The following sections provide a description of each register in the System Bus Interface Chip. All registers are shown with their default CFE address. The base CFE address is loaded by the Serial ROM into the CFEBAR register. If the serial ROM is not detected, the CFEBAR defaults to 1FFA0000h as the base address. For more information see, *CFE Base Address Register (CFEBAR)* on page 155.

PCI registers are shown with the addresses as an offset from a base address. The offset for memory and I/O accesses to these registers are indexed from the MMBAR and IOBAR, respectively. Configuration registers are indexed from address 0x00, and are only accessible through PCI bus configuration cycles.

Table 9-3. Miscellaneous Registers

Name	Page Number	PCI Offset (Mem & I/O)	PCI Config Offset	Width	R/W	CFE Address	R/W
GAID	169	–	8	R	1FFA0008	R	
RSR	163	–	–	–	1FFA0014	R	
NMI	164	–	8	R/W	1FFA001C	R	
LBPE	58	–	–	–	1FFA0020	R	
HSBR	153	–	32	R/W	–	–	
MDATA	154	–	32	R/W	–	–	

Table 9-4. CFE Configuration Registers

Name	Page Number	PCI Offset (Mem & I/O)	PCI Config Offset	Width	R/W	CFE Address	R/W
PROC_CFG	78	–	–	–	1FFA0010	R/W	
LBBAR	149	–	–	–	1FFA0024	R/W	
SEER	123	–	–	–	–	1FFA0028	R/W
CFEBAR	155	–	32	R	–	–	

Table 9-5. PCI Configuration Registers

Name	Page Number	PCI Mem Offset	PCI Config Offset	Width	R/W	CFE Address	R/W
DEVID	124	100[1]	00	32	R	1FFA5000	R
HSCR	125	104[1]	04	32	R	1FFA5004	R
CCRID	126	108[1]	08	32	R	1FFA5008	R
HMFR	127	10C[1]	0C	32	R	1FFA500C	R
IOBAR	128	110[1]	10	32	R	1FFA5010	R
HMBAR	128	114[1]	14	32	R	1FFA5014	R
MMBAR	129	118[1]	18	32	R	1FFA5018	R
SSID	130	12C[1]	2C	32	R	1FFA502C	R
XRBAR	130	130[1]	30	32	R	1FFA5030	R
LGIR	131	13C[1]	3C	32	R	1FFA503C	R

1 Memory Offset only. These registers are not accessible in IO space.

Table 9-6. SCB Registers

Name	Page Number	PCI Offset (Mem & I/O)	PCI Config Offset	Width	R/W	CFE Address	R/W
COMMAND	156	–	32	R/W	1FFA20 00	R	
ATTN	157	–	8	R/W	1FFA20 04	R	
SCP	159	–	8	R/W	1FFA20 08	R	
ISP	160	–	8	R	1FFA20 0C	R/W	
CBSP	161	–	8	R	1FFA20 10	R/W	
SIR	162	–	–	–	1FFA20 14	R/W	

Table 9-7. DMA Channel Registers

Name	Page Number	PCI Offset (Mem & I/O)	PCI Config Offset	Width	R/W	CFE Address	R/W
DMA CH1	140	–	–	–	–	1FFA3000-14	R/W
CAR1	140	–	–	–	–	1FFA3000	R/W
SAR1	140	–	–	–	–	1FFA3004	R/W
BCR1	141	–	–	–	–	1FFA3008	R/W
CCR1	141	–	–	–	–	1FFA300C	R/W
BMAR1	143	–	–	–	–	1FFA3010	R/W
LAP1	144	–	–	–	–	1FFA3014	R/W
BMSTAT1	145	–	–	–	–	1FFA3018	R

Table 9-7. DMA Channel Registers

Name	Page Number	PCI Offset (Mem & I/O)	PCI Config Offset	Width	R/W	CFE Address	R/W
BMCMD1	146	–	–	–	–	1FFA301C	R/W
SBCR1	147	–	–	–	–	1FFA3020	R/W
DMA CH2	140	–	–	–	–	1FFA4000-14	R/W
CAR2	140	–	–	–	–	1FFA4000	R/W
SAR2	140	–	–	–	–	1FFA4004	R/W
BCR2	141	–	–	–	–	1FFA4008	R/W
CCR2	141	–	–	–	–	1FFA400C	R/W
BMAR2	143	–	–	–	–	1FFA4010	R/W
LAP2	144	–	–	–	–	1FFA4014	R/W
BMSTAT2	145	–	–	–	–	1FFA4018	R
BMCMD2	146	–	–	–	–	1FFA401C	R/W
SBCR2	147	–	–	–	–	1FFA4020	R/W

.

Notes

1. Registers are referenced by abbreviated names. Numbers associated with DMA CDB register names refer to Bus Master Channels 1 and 2.
2. Detailed information for each register is provided in the section indicated.
3. Data widths for each register are specified for PCI Bus accesses. All Local Bus accesses are 32-bit accesses. Bits not specified in the register definition are reserved. These reserved bits also should be written to zero when writing the register.
4. PCI Addresses refer to offsets from a base address set during configuration. PCI registers are both memory and I/O address mapped. The offsets are the same for both memory and I/O base addresses, set in the MMBAR and the IOBAR respectively.
5. Entries under PCI Config Offset refer to offsets used in conjunction with IDSEL#, and are listed for PCI Configuration registers. The configuration registers are shown as read-only from both the PCI Bus and the CFE Local Bus. Some of these registers are read/write through configuration address space.
6. All unused CFE Local Bus addresses are reserved.
7. CFE Address entries assume a default base address of 1FFA0000 h. This base address may be programmed to any value 1FFx0000 h from the serial EPROM interface.

Programming Considerations

When back-to-back multiple word memory accesses of the same location (write-read) are performed thorough the memory window, it is possible that the data may be corrupted because the read of the location may be performed before the posted write has completed. To avoid this situation, use only single word writes.

According to PCI specification, care should be taken to deal correctly with registers that have reserved bits. On reads, software must use appropriate masks to extract defined bits, and ***cannot rely on reserved bits being any particular value***. On writes, software must ensure that the values of the reserved bit positions are preserved, by first reading the values of these bits and writing these values back when writing the new values of other bit positions.

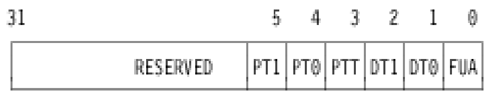
PCI Bus Configuration Registers

Serial EPROM Extension Register (SEER)

The Serial EPROM Extension Register provides additional configuration data to be programmed through the serial EPROM interface. These bits also can be programmed by a direct write from the CFE Local Bus.

Register Format

```
(CFE Local Bus Address = 1FFA0028 h)       32-bit read/write
```



Bit Descriptions

- Bits 31–5: Reserved.
- Bits 5–4: Preempt Timeout Bits. The preempt timeout bits are used to control the CFE bus bandwidth allowed to the PCI System Bus Interface Chip. These bits determine the minimum number of CFE clocks that must occur during a System Bus Interface Chip master transfer before the System Bus Interface Chip will recognize the removal of its acknowledge signals (-MSTRACK or -SLVEACK). The preempt timer starts decrementing from programmed preempt count 2 clocks after ADS. Once the preempt count expires, System Bus Interface Chip will recognize a preempt condition (removal of acknowledge) and relinquish control of the CFE bus a maximum of one data phase later.

PT1	PT0	Preempt Timer (CFE Clocks)
0	0	4
0	1	8
1	0	16
1	1	32

There are certain circumstances where the CFE master may stay on the bus for greater than the time specified in the preempt timer. When this occurs, the master is transferring a fixed number of words on the CFE bus, and is not preemptable. These special conditions are as follow:

- DMA List Chaining Operation – 6-word transfer
- PCI Slave Read using READ or READLINE command – 8-word transfer

- Bit 3: PCI Target Threshold. When set, this bit sets PCI target read prefetch buffer threshold to 4 transfers. When reset, the threshold is 24 transfers. More information on prefetch buffer thresholds can be found in Section *PCI Reads* on page 152.

- Bits 2–1: DMA Threshold bits 1 and 0. These bits set the access threshold of the DMA channels for each bus.

DT1	DT0	PCI / CFE Thresholds
0	0	4 / 4
0	1	4 / 16
1	0	16 / 4
1	1	RSVD

  More information on DMA Thresholds can be found in Section *DMA Thresholds* on page 133.

- Bit 0: Fast Unaligned Transfers. When set, this bit enables fast unaligned transfers on the CFE Local Bus. When reset, this bit disables fast unaligned transfers. Fast unaligned transfers are defined as the ability of the System Bus Interface Chip, as a master on the CFE Local Bus, to burst from an unaligned address through a transfer without requiring an additional address state on the second transfer to align the transfer on a 4-byte boundary. This feature can be used with slaves that support fast unaligned transfers to gain additional performance by eliminating the additional address states. This feature is supported by the Memory Controller Chip. If the PCI System Bus Interface Chip is only transferring to and from the Memory Controller Chip, this bit should be set.

Reset Conditions

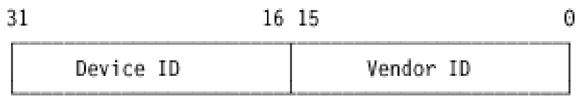
```
        RST# :  0000 0000 0000 0000 0000 0000 0000 0000
Command Reset:  0000 0000 0000 0000 0000 0000 00SS SSSS
```

Device/Vendor ID (DEVID)

The Device/Vendor ID Register identifies IBM as the manufacturer of System Bus Interface Chip. This register is read-only from both the CFE Local Bus and PCI interfaces.

Register Format

```
(Local Bus Address = 1FFA5000 h)         32-bit read-only
(PCI Config Offset = Base + 00 h)        32-bit read-only
(PCI Mem Offset = Base + 100 h)          32-bit read-only
```



Bit Descriptions

- Bits 31-16: Device ID. This read-only field identifies the device as System Bus Interface Chip. Its value is X'0036'.
- Bits 15-0: Vendor ID. This read-only field identifies the manufacturer as RadiSys. Its value is X'1014'.

Reset Conditions

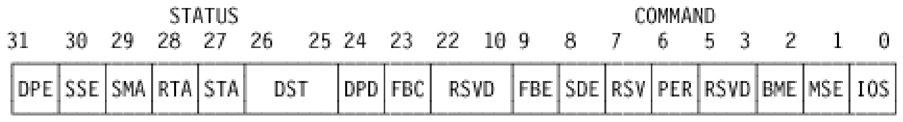
```
Power Up Reset: 0000 0000 0011 0110 0001 0000 0001 0100
Command Reset:  0000 0000 0011 0110 0001 0000 0001 0100
```

Host Status/Command Register (HSCR)

The Host Status/Command Register provides the control and status of the PCI interface in the System Bus Interface Chip. Bits in the status field can be reset from 1 to 0 by writing a 1 to the corresponding bit position.

Register Format

```
(Local Bus Address = 1FFA5004 h)      32-bit read-only
(PCI Config Offset = Base + 04 h)     32-bit read/write
(PCI Mem Offset = Base + 104 h)       32-bit read-only
```



Bit Descriptions

- Bit 31: Detected Parity Error. This bit is set to 1 when the System Bus Interface Chip detects a PCI Bus parity error, regardless of the state of the Parity Error Response Bit (Bit 6). A system configuration write of 1 to this bit will reset it to 0.
- Bit 30: Signalled System Error. This bit is set to 1 whenever the System Bus Interface Chip asserts -SERR on the PCI Bus. A system configuration write of 1 to this bit will reset it to 0.
- Bit 29: Signalled Master Abort. This bit is set to 1 whenever the System Bus Interface Chip terminates a PCI Bus master cycle with a master abort. A system configuration write of 1 to this bit will reset it to 0.
- Bit 28: Received Target Abort. This bit is set to 1 whenever the System Bus Interface Chip receives a target abort as a PCI Bus initiator. A system configuration write of 1 to this bit will reset it to 0.
- Bit 27: Signalled Target Abort Bit. This bit is set to 1 whenever the System Bus Interface Chip signals a target abort. A system configuration write of 1 to this bit will reset it to 0.
- Bits 26-25: DEVSEL Timing Bits. This read-only field indicates the slowest DEVSEL timing of the System Bus Interface Chip on the PCI bus. DEVSEL timing is set at medium, or '01'b.
- Bit 24: Data Parity Detected. The System Bus Interface Chip sets this bit to 1 when all of the following conditions are met: 1) the System Bus Interface Chip is asserted or observed -PERR active; 2) the System Bus Interface Chip is the PCI Bus Master when the error occurs, and 3) the Parity Error Response bit (Bit 6) is set to 1. This bit is reset to 0 with RST#. A system configuration write of 1 to this bit will reset it to 0.
- Bit 23: Fast Back-to-Back Capable. This capability is not implemented in the PCI System Bus Interface Chip; therefore, this bit always reads back a 0.

- Bits 22-10: Reserved. Always read back as 0.
- Bit 9: Fast Back-to-Back Enable. This capability is not implemented in the PCI System Bus Interface Chip; therefore, this bit always reads back a 0.
- Bit 8: -SERR Enable. When set, this bit enables the System Bus Interface Chip to drive -SERR on the system PCI bus. This bit is reset to 0 with RST#.
- Bit 7: Wait Cycle Control. The System Bus Interface Chip does not implement address/data stepping; therefore, this bit is hardwired to a zero, and always reads back as 0.
- Bit 6: Parity Error Response. When set to 1, the System Bus Interface Chip is enabled to check parity during system PCI Bus Master and slave transactions. When reset to 0, parity errors are ignored. Note that the Detected Parity Error bit (Bit 31) is set independent of the state of this bit. This bit is reset to 0 by
- Bit 5: VGA Palette Snoop. This function is not implemented; therefore, this bit always reads 0.
- Bit 4: Memory Write and Invalidate. This function is not implemented; therefore, this bit always reads 0.
- Bit 3: Special Cycles. This function is not implemented; therefore, this bit always reads 0.
- Bit 2: Bus Master Enable. When set to 1, the System Bus Interface Chip is enabled to arbitrate for control of the PCI Bus. When reset to 0, the System Bus Interface Chip is disabled from becoming a PCI Bus Master. This bit is reset to 0 by RST#.
- Bit 1: Memory Space Enable. When set to 1, the System Bus Interface Chip is enabled to respond to PCI Bus memory space accesses. This bit is reset to 0 by RST#.
- Bit 0: I/O Space Enable. When set to 1, the System Bus Interface Chip is enabled to respond to PCI Bus I/O space accesses. This bit is reset to 0 by RST#.

Reset Conditions

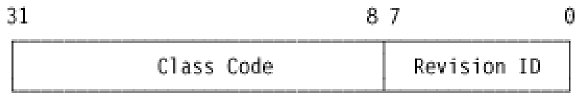
```
Power Up Reset: 0000 0010 0000 0000 0000 0000 0000 0000
Command Reset:  0000 0010 0000 0000 0000 000S 0S00 01SS
```

Class Code/Revision ID Register (CCRID)

The Class Code/Revision ID Register defines the general function and implementation level of the System Bus Interface Chip. This register is read-only from both interfaces.

Register Format

```
(Local Bus Address = 1FFA5008 h)      32-bit read-only
(PCI Config Offset = Base + 08 h)     32-bit read-only
(PCI Mem Offset = Base + 108 h)       32-bit read-only
```



Bit Descriptions

- Bits 31-8: Class code. This field specifies the general function of the System Bus Interface Chip. The value is 068000h, representing a Base Class=Bridge Device, SubClass=other, Programming Interface=not defined.
- Bits 7-0: Revision ID. This field specifies the revision level of the System Bus Interface Chip.

RadiSys Part # for Chip	Revision ID
06H7135	0x01
55H4504	0x02 (when available)

Reset Conditions

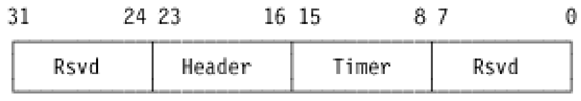
```
Power Up Reset: 0000 0110 1000 0000 0000 0000 0000 0001
Command Reset:  0000 0110 1000 0000 0000 0000 0000 0001
```

Host Miscellaneous Functions Register (HMFR)

The Miscellaneous Functions Register is used to set the PCI bus latency timer in the System Bus Interface Chip.

Register Format

```
(Local Bus Address = 1FFA500C h)    32-bit read-only
(PCI Config Offset = Base + 0C h)   32-bit read/write
(PCI Mem Offset = Base + 10C h)     32-bit read-only
```



Bit Descriptions

- Bits 31-24: Reserved. This field is reserved for Built-In Self Test (BIST) support. Always read as zeros.
- Bits 23-16: This read-only field specifies the configuration space layout. This field reads back all zeroes.
- Bits 15-8: This read/write field specifies the latency timer value in units of PCI bus clocks. Only the upper five bits of this field are programmable, the other bits read back all zeroes. This field is reset to all zeroes by -RST.
- Bits 7-0: Reserved. This field is reserved for setting the cache line size. This field reads back all zeroes.

Reset Conditions

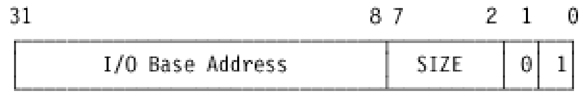
```
Power Up Reset: 0000 0000 0000 0000 0000 0000 0000 0000
Command Reset:  0000 0000 0000 0000 SSSS S000 0000 0000
```

I/O Base Address Register (IOBAR)

The I/O Base Address Register is used to specify the PCI bus starting I/O address and I/O address space size for the System Bus Interface Chip register map. The System Bus Interface Chip requires 256 bytes of I/O address space.

Register Format

```
(Local Bus Address = 1FFA5010 h)      32-bit read-only
(PCI Config Offset = Base + 10 h)     32-bit read/write
(PCI Mem Offset = Base + 110 h)       32-bit read-only
```



Bit Descriptions

- Bits 31-8: I/O Base Address. This read/write field specifies the starting PCI bus I/O address space for the System Bus Interface Chip register space.
- Bits 7-2: These bits are used to determine the I/O size as 256 bytes. These bits always read 0.
- Bit 1: Reserved. This bit always reads 0.
- Bit 0: I/O Space Indicator. This bit always reads 1.

Reset Conditions

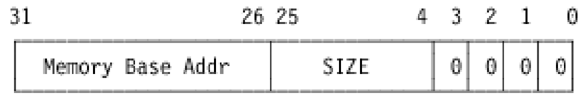
```
Power Up Reset: uuuu uuuu uuuu uuuu uuuu uuuu 0000 0001
Command Reset:  SSSS SSSS SSSS SSSS SSSS SSSS 0000 0001
```

Host Memory Base Address Register (HMBAR)

The Host Memory Base Address Register is used to specify the PCI bus starting memory address and memory space size. The size of the Host memory base address is 64MB.

Register Format

```
(Local Bus Address = 1FFA5014 h)          32-bit read-only
(PCI Config Offset = Base + 14 h)         32-bit read/write
(PCI Mem and I/O Offset = Base + 114 h)   32-bit read-only
```



Bit Descriptions

- Bits 31-26: Memory Base Address. This read/write field specifies the starting PCI bus memory address.
- Bits 25-4: These bits are used to determine the memory size as 64MB. These bits always read 0.
- Bit 3: Prefetchable. This bit always reads 0.
- Bits 2-1: Type. This read-only field always reads 00.
- Bit 0: Memory Space Indicator. This read-only bit is always read as a 0.

Reset Conditions

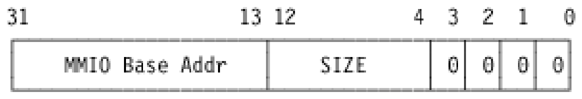
```
Power Up Reset: uuuu uu00 0000 0000 0000 0000 0000 0000
Command Reset:  SSSS SS00 0000 0000 0000 0000 0000 0000
```

Memory Mapped I/O Base Address Register (MMBAR)

The Memory Mapped I/O Base Address Register is used to specify the PCI bus starting memory address and size of memory mapped I/O. Memory mapped I/O is the mapping of all I/O registers addressable from the PCI Bus into memory address space. Memory mapped I/O is required for systems that do not support an I/O address space. The size of the memory mapped I/O address space is 8KB.

Register Format

```
(Local Bus Address = 1FFA5018 h)      32-bit read-only
(PCI Config Offset = Base + 18 h)     32-bit read/write
(PCI Mem Offset = Base + 118 h)       32-bit read-only
```



Bit Descriptions

- Bits 31-13: MM I/O Base Address. This read/write field specifies the starting PCI bus memory address.
- Bits 12-4: These bits are used to determine the memory size as 8KB. These bits always read 0.
- Bit 3: Prefetchable. This bit always reads 0.
- Bit 2-1: Type. This read-only field always reads 00.
- Bit 0: Memory Space Indicator. This read-only bit is always read as a 0.

Reset Conditions

```
Power Up Reset: uuuu uuuu uuuu uuuu uuuu 0000 0000 0000
Command Reset:  SSSS SSSS SSSS SSSS SSSS 0000 0000 0000
```

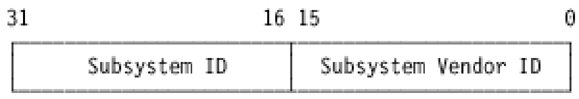
Subsystem ID (SSID)

The Subsystem ID Register identifies the manufacturer and function of the subsystem using the System Bus Interface Chip. This register is divided into two 2-byte registers, located at 2C h and 2E h. These registers are read-only from both the CFE Local Bus and PCI interfaces. The values of these registers are loaded from a serial ROM. If there is no serial ROM present, the value will remain at its reset value of 0000 0000h, indicating the Subsystem ID function is not implemented..

For the serial EPROM values, see Table 9-11.

Register Format

```
(Local Bus Address = 1FFA502C h)       32-bit read-only
(PCI Config Offset = Base + 2C h)      32-bit read-only
(PCI Mem Offset = Base + 12C h)        32-bit read-only
```



Bit Descriptions

- Bits 31-16: Subsystem ID. This read-only field identifies the device as System Bus Interface Chip. Its value powers up to 0000 and is loaded from a serial ROM.
- Bits 15-0: Subsystem Vendor ID. This read-only field identifies the manufacturer of the subsystem. Its value powers up to 0000 and is loaded from a serial ROM.

Reset Conditions

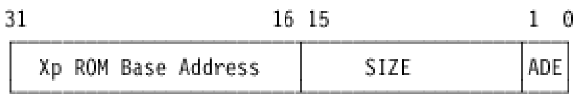
```
Power Up Reset: 0000 0000 0000 0000 0000 0000 0000 0000
Command Reset:  SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSSS
```

Expansion ROM Base Address Register (XRBAR)

The Expansion ROM Base Address Register is used to specify the starting Expansion ROM address of the System Bus Interface Chip in PCI bus address space. The first location in the 64KB window is mapped in CFE Local Bus address space to the value in the LBBAR; that is, the window is mapped to a 64KB region with the LBBAR as the base address.

Register Format

```
(Local Bus Address = 1FFA5030 h)          32-bit read-only
(PCI Config Offset = Base + 30h)          32-bit read/write
(PCI Mem Offset = Base + 130 h)           32-bit read-only
```



Bit Descriptions

- Bits 31-16: Expansion ROM Base Address. This field specifies the starting PCI bus memory address for the System Bus Interface Chip expansion ROM.
- Bits 15-1: These bits indicate the memory space size of 64KB.
- Bit 0: Address Decode Enable. When set to 1, this bit enables access to the System Bus Interface Chip expansion ROM, if the memory space enable bit (Bit 1 in the Host Status/Command Register) also is set to 1.

Reset Conditions

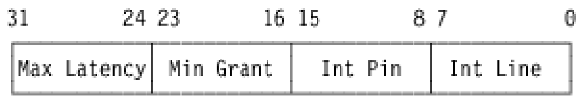
```
Power Up Reset: 0000 0000 0000 0000 0000 0000 0000 0000
Command Reset:  SSSS SSSS SSSS SSSS 0000 0000 0000 000S
```

Latency/Grant/Interrupt Register (LGIR)

The Latency/Grant/Interrupt Register is used to specify the maximum latency, minimum grant time, and interrupt information of the PCI System Bus Interface Chip.

Register Format

```
(Local Bus Address = 1FFA503C h)          32-bit read-only
(PCI Config Offset = Base + 3Ch)          32-bit read/write
(PCI Mem Offset = Base + 13C h)           32-bit read-only
```



Bit Descriptions

- Bits 31-24: Maximum Latency. This read-only field is always read as X'00'.
- Bits 23-16: Minimum Grant. This read-only field is always read as X'00'.
- Bits 15-8: Interrupt Pin. This read-only field is always read as X'01', indicating connection to -INTA on the PCI Bus.
- Bits 7-0. Interrupt Line. This read/write field is used for PCI interrupt priority and vector information. Values in this field are system architecture specific.

Reset Conditions

```
Power Up Reset: 0000 0000 0000 0000 0000 0001 0000 0000
Command Reset:  0000 0000 0000 0000 0000 0001 SSSS SSSS
```

Bus Master Channel Functional Description

The PCI bus master interface consists of two Bus Master channels addressable from the Local Bus. Each channel uses a 128-byte intermediate buffer between resident memory and the PCI. The following sections describe the operation of the Bus Master Channels 1 and 2.

General Operation

All data transactions, including Bus Master transfers, are subject to intermediate buffering. A diagram of this intermediate buffering is shown in Figure 9-1.

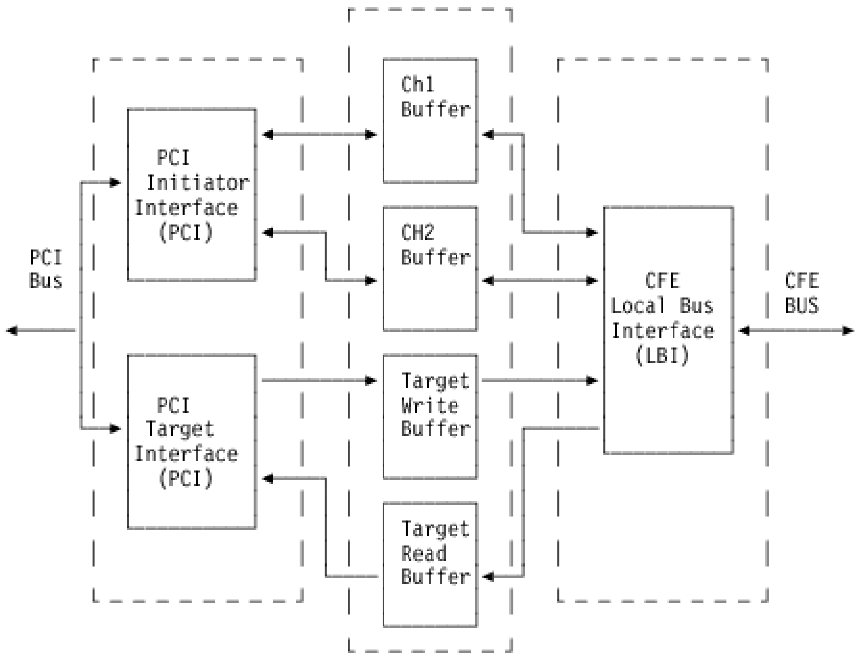

PCI Initiator Interface (PCI)
PCI Target Interface (PCI)
PCI Bus
Ch1 Buffer
CH2 Buffer
Target Write Buffer
Target Read Buffer
CFE Local Bus Interface (LBI)
CFE BUS

Figure 9-1. PCI Bus Intermediate Buffering

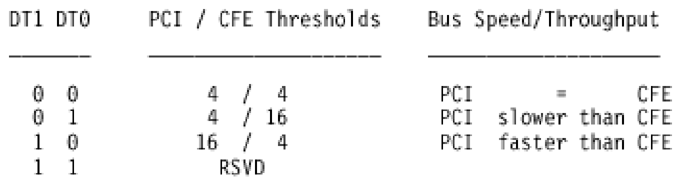
Each Bus Master channel manages data movement through 128 bytes of intermediate buffering. This intermediate buffering is organized as a single dual-ported buffer. Each Bus Master channel uses its buffer for accesses on the CFE Local Bus and PCI Bus interfaces, optimizing the total throughput of its data transfers.

The following discussion pertains to the operation of a single Bus Master channel. For this discussion, the term *bus* refers generically to either the PCI Bus or the CFE Local Bus.

The general operation of a Bus Master channel for large transfers is to fill and flush data to and from its internal buffer. At any given time, data can be flushed from the buffer on one bus, and filled into the buffer from the other bus, approximating a constant throughput between the two buses. When the channel is active, if space in the buffer is available (that is, larger than a minimum requirement to start a transfer for a given bus) and there are no data transfers currently taking place on that bus, the channel will arbitrate immediately for control of the bus.

DMA Thresholds

The DMA channels have an access threshold for each bus. An access threshold is defined as the number of locations in the internal RAM buffers of the Bus Master channels that must be free or filled (depending on the direction of transfer) before the Bus Master channel will request the given bus. The number of free or filled locations corresponds to the number of transfers performed on the opposite bus. For example, if the threshold for the CFE is set to four locations, and the direction of the data transfer is from the PCI Bus to the CFE Local Bus, four transfers must take place on the PCI Bus and four locations must be ***filled*** in the internal RAM buffer before the Bus Master channel will request the CFE Local Bus. In this example, if the data transfer was in the opposite direction, and the internal buffer was full, four transfers must take place on the PCI Bus and four locations must be ***free*** in the internal RAM buffer before the Bus Master channel will request the CFE Local Bus. The threshold for each bus is set by the DMA Threshold bits, (SEER, Bits 21), according to the following table. The intent of these bits is to help compensate for differing speeds and throughput on the CFE and PCI buses. Programming these bits correctly can help to ensure that a burst transfer is sustained for as long as possible when there is a mismatch between the PCI and CFE bus speed.



DT1	DT0	PCI / CFE Thresholds	Bus Speed/Throughput
0	0	4 / 4	PCI = CFE
0	1	4 / 16	PCI slower than CFE
1	0	16 / 4	PCI faster than CFE
1	1	RSVD	

These bits either can be loaded from an external serial EPROM, or directly written from the CFE Local Bus.

Flow of Operation

Each Bus Master channel is configured from the CFE Local Bus by programming a Channel Descriptor Block (CDB) resident in the System Bus Interface Chip. A description of the CDBs for each Channel is given in Section *Bus Master Channel Descriptor Block (BMCDB)* on page 140.

Each Bus Master channel is controlled through a Channel Control Register (CCR) located in each CDB. A channel is enabled by setting the Start/Stop Bit (Bit 0) in the CCR. The direction of the data transfer is set by the Direction Bit (Bit 1) in the CCR. Description of the CCR for each channel is given in *Channel Control Register (CCR)* on page 141.

When the Start/Stop Bit is set in the Bus Master channel, operation begins in one of two ways, depending on the state of the Direction Bit.

- If the direction is set to read from the PCI Bus (writing to the Local Bus), the Bus Master channel will immediately arbitrate for the PCI Bus. When granted the PCI Bus, the Bus Master channel will fill its buffer by reading from the address specified in the System Address Register (SAR), located in the CDB. The channel will continue reading data until its buffer is full or the byte count equals zero. The buffer will be full after a minimum of 128 bytes, but this may be more depending on the speed of the CFE Local Bus.

  At this point, the channel will release its control of the PCI Bus. The channel also may relinquish the bus if the count in the Byte Count Register (BCR) equals zero. The channel may rearbitrate for the PCI Bus once 16(/4) words are available in the internal buffer.

Concurrent to the PCI read operation, and after the first 16(/4) words have been read, data is transferred from the internal buffer to CFE Local Bus address space at the location specified in the Card Address Register (CAR). The channel will continue to fill and flush data until the Byte Count equals zero.

- If the direction is set to write to the PCI Bus (reading from the CFE Local Bus), the channel fills the buffer from CFE address space from the location loaded in the CAR. The channel continues to fill the buffer until the buffer is full or the Byte Count equals zero.

  Concurrent to this operation and after 16(/4) words are loaded, the channel arbitrates for the PCI Bus. When granted, the channel flushes the buffered data to the PCI Bus at the location loaded in the SAR. Concurrently, after the first 16(/4) words are completely flushed on the PCI Bus, the System Bus Interface Chip rearbitrates for the CFE Local Bus, and additional data is loaded in the buffer from the CFE Local Bus.

  In the general case, the PCI Bus will continue to flush the additional data until the buffer is empty. When there is no more data to be transferred, the channel releases its control of the PCI Bus. The channel continues to fill from the Local Bus until the buffer is full or the Byte Count equals zero. The channel then will rearbitrate for the PCI Bus. The channel continues to fill and flush data until the Byte Count equals zero.

Channel Priority on the CFE Local Bus

The two Bus Master channels have equal priority under normal operation, that is, they alternate ownership of the Local Bus equally. If one channel owns the CFE Local Bus and the other channel has a CFE access pending, the System Bus Interface Chip will reissue its request for the CFE Local Bus after the current owner relinquishes ownership. When the System Bus Interface Chip receives its -MSTRACK a second time, ownership of the CFE Local Bus will pass to the pending channel..

A channel could maintain ownership over the CFE Local Bus and lock out the opposite channel for long periods.

Channel Priority on the PCI Bus

Like the CFE Local Bus, the two Bus Master channels alternate ownership of the PCI Bus equally. In addition, the channels alternate ownership for target retry and disconnect, that is, if a Bus Master channel is accessing a target and is retried (or disconnected) the alternate channel gets the next PCI Bus ownership.

Bus Operation

The System Bus Interface Chip is capable of performing transfers with any address alignment combination, as a PCI Bus Master and as a PCI slave through the shared memory window. The number of bytes transferred on either the PCI Bus or the Local Bus is controlled by the respective byte enable signals for that bus.

On the PCI, the number of bytes transferred during a cycle is determined by the total number of bytes to be transferred and the PCI address. As a PCI Bus Master, in all cases, the maximum number of bytes are transferred. Therefore, for a transfer on an odd-byte boundary (that is, an address with the least-significant two bits equal to '01'b), the System Bus Interface Chip transfers three bytes.

As a CFE Local Bus Master, the System Bus Interface Chip also transfers the maximum number of bytes per transfer. When writing a slave on an odd-byte boundary, the System Bus Interface Chip transfers three bytes. In this case, the upper 3-byte enables (BE3-1) are asserted. When reading a slave on any boundary, the entire 32-bit word associated with that address is read (that is, all 4-byte enables are asserted and the two least-significant bits of the CFE Local Bus address are zero).

Posted Status Operations

Each channel has the ability to perform a posted status operation to a location in CFE address space. The intent of this operation is to provide an alternate method of signalling the normal termination of a data transfer, in addition to the terminal count interrupt. Posting status has two applications in conjunction with list chained data transfer elements:

1. Normal termination status for each element can be posted in a table in CFE address space. A terminal count interrupt can be asserted on the last element in the list chain. The table can be used for identifying a failing element during error recovery.
2. The posted write itself can be made to a device or single address location to positively acknowledge completion of a list chain element. This is especially useful for interfacing to an programmable device other than a microprocessor controlling the enqueuing and dequeuing of elements in the list chain or pipe.

The 32-bit address for this operation is stored in the Bus Master Address Register (BMAR). The appended operation is a 32-bit write to this location of the contents of the Bus Master Status Register (BMSTAT). When a terminal count is reached, the channel posts status to this location. If the direction of transfer is from the CFE Local Bus to the PCI Bus, status is posted after the last transfer to the PCI Bus. If the direction of transfer is from the PCI Bus to the CFE Local Bus, status is posted after the last transfer to the CFE Local Bus.

The channel controls Posted Status operations through its CCR. The Posted Status Bit, Bit 8, sets the channel for Posted Status operation.

Exceptions

Note that only normal termination status is posted. Status is not posted for exception conditions; the channel is stopped and an interrupt is posted. Therefore, posted status will always be the value 0000 0001h, indicating normal termination. All other values are reserved.

Linked List Chaining

Both Bus Master channels support Linked List Chaining (LLC), or scatter/gather DMA. Linked List Chaining provides the ability for each channel to auto-initialize its Channel Descriptor Block (CDB) from a predefined list in CFE address space. Each element in this list consists of a set of new register values. An element is loaded into the CDB when the channel reaches a terminal count for the current data transfer. Since each element represents a separate data transfer, Linked List Chaining allows the channel to interleave the scattering and the gathering of data from different buffer locations in memory, with a minimum of direct programming. In addition, control information in the CCR also can be updated, allowing dynamic changing of DMA parameters.

Within the CDB a 32-bit address pointer to the list in memory is maintained. At the time of terminal count, if list chaining is enabled, the hardware will fetch six 32-bit words starting at this memory address and reload the CDB at hardware speeds. List chaining provides a mechanism to off-load the programming of the System Bus Interface Chip after every terminal count. Based on the value programmed in the CCR, the channel can optionally interrupt the CFE Local Bus, in addition to List Chaining, on a terminal count.

This list of buffers can occupy any area in CFE address space. The list chaining function is shown in Figure 9-2.

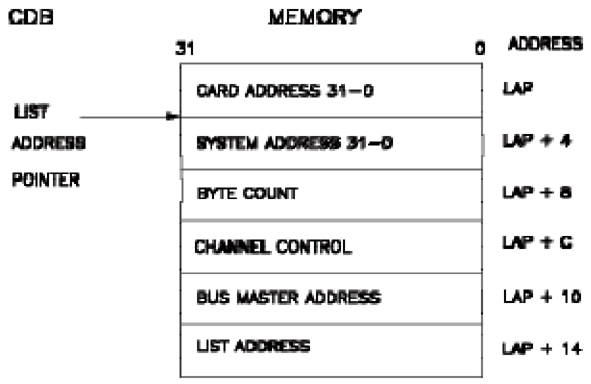


Figure 9-2. Linked List Chaining

At the end of this operation, the CDB contains a new pointer to a new list located anywhere else in memory.

List Chaining from Stop

List Chaining also can be initiated without a data transfer. If the Start/Stop Bit (Bit 0, CCR) and the List Chain Enable Bit (Bit 3, CCR) are set and the byte count is zero (BCR = 0), a list chain operation will start. This list chain operation from a Channel stopped state is a special case of List Chaining Zero Byte Elements, described in Section *Modes of Operation* on page 137.

Bus Master Interrupts to PCI Bus

To assist in designs that require control from the PCI Bus, the Bus Master Channels have the ability to interrupt the PCI Bus on termination of a data transfer. This function is selectable for each channel by selecting the DMA Interrupt Direction Bits (PROC_CFG, Bits 15-14).

When the Interrupt Direction Bit for a given channel is set, the termination interrupt is provided through the INTA# signal to the PCI Bus, rather than through the encoded interrupt of the CFE Local Bus. Conditions and timing for the interrupt are the same as for interrupt the CFE Local Bus. In addition, since only one interrupt signal is presented to the PCI Bus, primary status for distinguishing between channel interrupts and other sources of interrupts is provided in the Command Busy Status Port (CBSP). In addition to the Interrupt Valid Bit (CBSP, Bit 1), the DMA Channel Interrupt Bits are provided (CBSP, Bits 32). These bits operate in the same manner as the IV Bit, and are cleared, together with the IV Bit (if set) when the CBSP is read from the PCI Bus.

The interrupt capability defined by these bits is separate from the SCB architecture. CBSP Bits 32 are defined as device-dependent bits and operate as such when the direction bits are reset in the PROC_CFG.

Modes of Operation

There are several special modes of operation for termination of the Bus Master channels and interrupting the CFE Local Bus. These different modes involve several bits in the CCR.

It is easiest to think of transfers as broken into three steps: a data transfer, an optional posted status operation, and a list chain operation to bring in a new list chain element. A list chain element is a 6-word entry in CFE Local Bus address space containing the values for initializing the Channel Descriptor Block (CDB) for a data transfer. It is important to note that the termination interrupt enabled by the Terminal Count Interrupt Enable (Bit 2) in the CCR, represents the normal termination of a data transfer, a data transfer plus posted status when posted status is enabled, and does not necessarily represent termination or disabling of Bus Master operation.

The beginning of a list chain operation is considered the start of a new data transfer.

- ***Termination Interrupts on the Fly***

  Termination interrupts can be enabled for any element in a list chain, not just the last element, associated with the stopping of the channel. Since Normal Termination interrupts are associated with the normal termination of a data transfer or of a data transfer with posted status, an interrupt can be asserted after any data transfer or after its posted status, if enabled, by enabling the Terminal Count Interrupt Enable Bit (CCR, Bit 2) for that element in the list chain.

  Although it is possible to take interrupts on the fly, normal operation is to take a single termination interrupt at the end of the entire list chain. A single termination interrupt is desirable for the following reasons:

  - For list chains involving small transfers, it is difficult to take interrupts for each element, so some interrupts could be lost without stopping the channel.
  - Since a new list chain element and a new transfer are immediately list chained after the interrupt, there is no status available in the BMSTAT register representing the termination interrupt. (List Chaining represents the start of a new transfer; therefore, status is cleared).

- ***List Chaining Zero Byte Elements***

  If a channel is active with list chaining enabled, and there are currently no transfers to take place, list chain elements containing a byte count of zero can be appended to the list chain. As long as the List Chaining Enable bit (CCR, Bit 3) is enabled in the appended elements, list chaining will continue without data transfers.

- ***Stopping Channel after List Chaining Bit***

  The Bus Master Channel can be stopped after the current list chain element is read from memory and prior to performing the data transfer. Upon stopping, the channel can be re-enabled and operation will continue with the current data transfer.

  This mode of operation is especially useful in conjunction with the List Chaining Zero Byte Elements option. In the case where there are currently no transfers to take place, an element of zero byte count can be appended **with the Stop Channel after List Chaining Bit (CCR, Bit 4) set in the CCR**. As long as the List Chain Enable Bit is also set in this element, operation will continue upon re-enabling the channel, with a list chain operation taking place to the address pointed to by the current list address pointer. The Stop Channel after List Chaining can be used as a true NOP or Pause function for a given list chain or pipe.

  Since this option terminates after a list chain, and not after a data transfer or posted status operation, there is no normal termination interrupt associated with stopping the channel in this mode. The start/stop bit must be polled to determine when the channel is stopped.

- ***Resetting the Start/Stop Bit***

  The Bus Master Channel can be stopped noncatastrophically by simply resetting the start/stop bit. There is some latency between the resetting of the bit and the stopping of the channel. During this period, the start/stop bit will not read back zero. The start/stop bit will read back zero when the channel is stopped, and functions as termination status for this mode of operation. Operation will resume with the pending data cycles for both the CFE Local Bus and the PCI Bus, upon re-setting the start/stop bit.

  Like the **Stopping the Channel after List Chaining** option, this operation is not associated with normal termination of data transfers or posted status, and after the start/stop bit is reset, there is no termination interrupt associated with this mode of operation.

- ***Resetting the Bus Master Channel***

  Like Resetting the Start/Stop Bit, the channel can be stopped noncatastrophically by setting the channel reset in the BMCMD register. This operation functions exactly like resetting the Start/Stop Bit; the Start/Stop Bit functions as status, and no termination interrupt is generated. The Bus Master channel, however, is reset after the operation and not available to complete the current transfer.

- ***Watchdog Input***

  The Watchdog input is an asynchronous input that also forces both channels to terminate noncatastrophically. Although this input is an exception condition, the exception is reported to CFE Local Bus separately from the System Bus Interface Chip, and may be unrelated to the operation of either Bus Master channel. As a result, both channels will terminate without interrupt or status. The Bus Master channels are reset on this error and are not available to complete their current transfers.

Stopping the Channel

The following is a summary of conditions that stop the Bus Master channel with termination interrupt and status.

- PCI Bus Data Parity Error
- Received Target Abort
- Signalled Master Abort
- Detected SERR# or detected PERR#
- Resetting of Bus Master Enable (HSCR, bit 2)
- Resetting of DMA Enable (SCP, bit 1)
- CFE Local Bus Parity Error
- -L_EXCPT Detected on the CFE Local Bus

The following conditions stop the Bus Master channel without termination interrupt and status.

- Bus Master Channel Reset
- All Resets (-RST, Command)
- Termination using the Stop on List Chain Command
- Resetting the Start/Stop Bit in the CCR
- -WATCHDOG Detected on the CFE Local Bus

Note that all terminations are noncatastrophic; that is, termination of the channel is nondisruptive to either the PCI Bus or the CFE Local Bus, with the exception of RST# and Command Reset. Command Reset is catastrophic to CFE Local Bus operations only.

In all cases, with the exception of resetting the Start/Stop bit, the channel is completely reset on termination. When resetting the Start/Stop bit, the channel terminates with the internal state machines intact, providing the option to restart the channel and complete the current transfer.

Bus Master Registers

The following sections describe in detail the registers associated with Bus Master operation.

Bus Master Channel Descriptor Block (BMCDB)

Each channel has a 6-word channel descriptor block (CDB). This CDB is loadable from the Local Bus. The CDB register map is described in Table 9-8.

Table 9-8. BMCDB Local Bus Register Map

CDB Register	Channel 1	Channel 2
Card Addr Register 31-0	1FFA3000	1FFA4000
System Addr Register 31-0	1FFA3004	1FFA4004
Byte Count Register	1FFA3008	1FFA4008
Channel Control Register	1FFA300C	1FFA400C
Bus Master Addr Register 31-0	1FFA3010	1FFA4010
List Addr Pointer 31-0	1FFA3014	1FFA4014

Register descriptions within the CDB are given in the following sections. All channel addresses are Local Bus addresses.

Card Address Register (CAR)

The card address register (CAR) is a 32-bit register that contains the Local Bus address of the next data word in resident memory to be transferred. Depending on the byte count, the last access to resident memory may be one, two, three, or four bytes. This value is loaded initially by a write from the resident processor. Subsequent loading is either by additional writes or, if list chaining is enabled, by hardware accesses to a list in resident memory.

Register Format:

```
(Channel 1 Address = 1FFA3000 h)          32-bit rd/wr
(Channel 2 Address = 1FFA4000 h)          32-bit rd/wr
```

31
0
CARD ADDRESS

Reset Conditions:

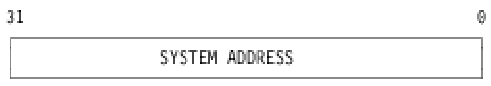
```
RESET:           uuuu uuuu uuuu uuuu uuuu uuuu uuuu uuuu COMMAND
RESET:           SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSSS ROM
INITIALIZATION:  0000 0000 0000 0000 0000 0000 0000 0000
```

System Address Register (SAR)

The System Address Register (SAR) is a 32-bit register that contains the physical PCI Bus address of the next data byte to be transferred. Loading is either by CFE writes or, if list chaining is enabled, by hardware accesses to a list in CFE Local Bus address space.

Register Format:

```
(Channel 1 Address = 1FFA3004 h)          32-bit rd/wr
(Channel 2 Address = 1FFA4004 h)          32-bit rd/wr
```



Bit Descriptions:

Bits 31-0: System Address. These bits contain the system address for the current transfer.

Reset Conditions:

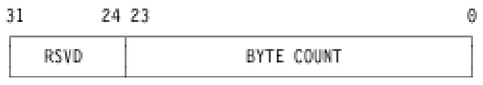
```
RESET:              uuuu uuuu uuuu uuuu uuuu uuuu uuuu uuuu COMMAND
RESET:              SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSSS
ROM INITIALIZATION: 0000 0000 0000 0000 0000 0000 0000 0000
```

Byte Count Register (BCR)

This register contains a 24-bit value that signifies the number of bytes that will be transferred ***on the CFE Bus*** before a terminal count is reached.

Register Format:

```
(Channel 1 Address = 1FFA3008 h)          32-bit rd/wr
(Channel 2 Address = 1FFA4008 h)          32-bit rd/wr
```



Bit Descriptions:

- Bits 31-24: Reserved.
- Bits 23-0: Byte Count. These bits contain the byte count for the current transfer.

Reset Conditions:

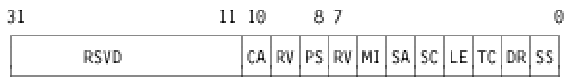
```
RESET:          0000 0000 uuuu uuuu uuuu uuuu uuuu uuuu COMMAND
RESET:          0000 0000 SSSS SSSS SSSS SSSS SSSS SSSS ROM
INITIALIZATION: 0000 0000 0000 0000 0000 0000 0000 0000
```

Channel Control Register (CCR)

The Channel Control Register (CCR) is the control and command area for a channel. It is loaded initially by a write from the CFE Local Bus, and is subsequently reloaded either by additional writes, or if linked list chaining is enabled, it is loaded by the hardware during memory list accesses. In this way, dynamic control of the channel is possible.

Register Format:

```
(Channel 1 Address = 1FFA300C h)          32-bit rd/wr
(Channel 2 Address = 1FFA400C h)          32-bit rd/wr
```



Bit Descriptions:

This register stores the control/status information for Bus Master Channel 1.

- Bits 31-11: Reserved.
- Bit 10: Card Address Increment/NoIncrement Bit. This bit, when reset, indicates the channel increments the card address after each CFE Local Bus transfer. This bit, when set, indicates the channel does not increment the card address after each transfer.
- Bit 9: Reserved. Reads back zero.
- Bit 8: Posted Status Bit. This bit, when set, enables the Bus Master channel to post the contents of the BMSTAT register to the CFE Local Bus address located in the Bus Master Address Register (BMAR). More information on Posting Status can be found in Section *Bus Master Address Register (BMAR)* on page 143 and Section *Posted Status Operations* on page 135.
- Bit 7: Reserved. Reads back zero.
- Bit 6: Memory-I/O Transfer Bit. This bit, when set, indicates the channel performs cycles to PCI Bus memory space. This bit, when reset, indicates the channel performs cycles to PCI Bus I/O space.
- Bit 5: System Address Increment/NoIncrement Bit. This bit, when reset, indicates the channel increments the value in the system address register after each transfer. This bit, when set, indicates the channel does not increment the system address register after each transfer.
- Bit 4: Stop Channel after List Chaining Bit. This bit is active only when the list chaining enable bit (LE) is set. When active, if this bit is set, the channel will stop after the **current** list chaining operation, that is, after the entire list element is read in which the bit is set. If the bit is reset, the channel will continue to do transfers after the current list chaining operation. The channel always stops after the current transfer when the list chaining enable (LE) is not set.
- Bit 3: List Chaining Enable Bit. This bit, when set, indicates list chaining is enabled. This bit, when reset, indicates list chaining is disabled. More information is provided in Section *Linked List Chaining* on page 136.
- Bit 2: Terminal Count Interrupt Enable Bit. If this bit is set, an interrupt will be presented to the CFE Local Bus when the value stored in the byte count transitions to 000000 h, or immediately after a posted status transfer, if enabled. If this bit is reset, no interrupt occurs.

- Bit 1: Direction bit. This bit signifies the direction of data transfer on the PCI Bus. This bit, when set, indicates a transfer from the PCI Bus to the CFE Bus. This bit, when reset, indicates a transfer from the CFE Bus to the PCI Bus.
- Bit 0: Start/Stop bit. Setting this bit initiates a Bus Master transfer. Hardware will reset the bit to indicate the transfer has completed. Writing a 0 to this bit will terminate the present transfer after the cycle in process has completed. No internal counters are reset, so that a subsequent write of 1 will continue the same transfer where it left off. If the channel is stopped by resetting the Start/Stop bit, the channel must be reset with a channel reset prior to initiating a different transfer.

  In addition, reading this bit will not read zero unless the cycle has completed and the channel is stopped. The bit, therefore, provides status for the channel as well as controlling its function.

  When this bit is set, no other bit in the CCR can be updated by a direct write from the CFE Local Bus. This feature allows the start/stop bit to be reset without corrupting the transfer currently executing.

  This bit is not updated during list chaining. All other bits in the CCR are updated during list chaining. This feature allows the register to be loaded with new values during list chaining without resetting the start/stop bit. Stopping the channel during list chaining is under control of Bit 4 in the CCR, Stop Channel after List Chaining. Note that none of the CDB registers other than the CCR return a valid value when read while the channel is running.

Reset Conditions:

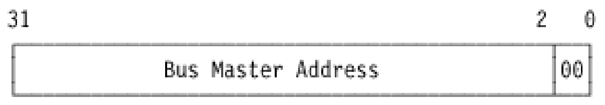
```
RST#:           0000 0000 0000 0000 0000 0u0u 0uuu uuu0
Command Reset:  0000 0000 0000 0000 0000 0S0S 0SSS SSS0
```

Bus Master Address Register (BMAR)

The Bus Master Address Register stores a 32-Bit value for support of posted status transfers. For Posting Status, the value in the BMAR represents a location in CFE Local Bus address space. The contents of the BMSTAT register for the given channel is written to this location as a 32-bit transfer. Therefore, the address should be 4-byte aligned. Note that the posting of status does not reset the value of the BMSTAT register. This register must still be read upon termination to clear its contents. When list chaining, the contents of this register are cleared at the beginning of the next list chain operation to reset status for the next data transfer.

Register Format:

```
(Channel 1 Address = 1FFA3010 h)          32-bit rd/wr
(Channel 2 Address = 1FFA4010 h)          32-bit rd/wr
```



Bit Descriptions:

- Bits 31-2: Address bits. These bits store the 32-bit address of a Posted Status location. Posted Status addresses are located on 4-byte boundaries, and the least significant two bits of this register are forced to zero.
- Bits 1-0: Always read zero.

Reset Conditions:

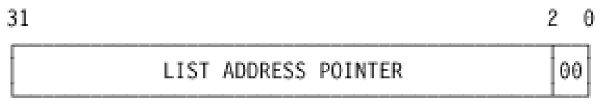
```
RST#:           uuuu uuuu uuuu uuuu uuuu uuuu uuuu uu00
Command Reset:  SSSS SSSS SSSS SSSS SSSS SSSS SSSS SS00
```

List Address Pointer (LAP)

The List Address Pointer (LAP) is a 32-bit register that contains an address in CFE Local Bus address space where a list of CDB information exists. The register is initially loaded by a write from the CFE Local Bus. After initialization, the register can be reloaded by write instruction, or if list chaining is enabled, the register is initialized automatically by hardware from a list table entry in CFE Local Bus address space. List chaining is an atomic operation on the CFE Local Bus consisting of six contiguous transfers. The LAP will contain a new value read from the list chain element after the list chain operation is complete. The values written to the register should be 4-byte aligned; that is, Bits 1 and 0 are forced to zero. More information on list chaining can be found in Section *Linked List Chaining* on page 136.

Register Format:

```
(Channel 1 Address = 1FFA3014 h)          32-bit rd/wr
(Channel 2 Address = 1FFA4014 h)          32-bit rd/wr
```



Bit Descriptions:

- Bits 31-2: Address bits. These bits store the 32-bit address of a list chain element. List Chain addresses are located on 4-byte boundaries, and the least-significant two bits of this register are forced to zero.
- Bits 1-0: Always read zero.

Reset Conditions:

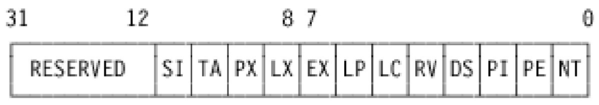
RST#: uuuu uuuu uuuu uuuu uuuu uuuu uuuu uu00
Command Reset: SSSS SSSS SSSS SSSS SSSS SSSS SSSS SS00

Bus Master Status Register (BMSTAT1 (2))

This register provides the termination status for each Bus Master channel. The contents of this register are cleared by a read access. More information on CFE Local Bus interrupts is available in Section *System Bus Interface Chip Interrupts* on page 164.

Register Format:

```
(Channel 1 Address = 1FFA3018 h)          32-bit rd only
(Channel 2 Address = 1FFA4018 h)          32-bit rd only
```



- Bits 31-12: Reserved.
- Bit 1-1: Receipt of SERR#. This bit is set when SERR# is asserted by a target during a Bus Master operation on the PCI Bus, that is, while the System Bus Interface Chip is granted the PCI Bus as a bus initiator.
- Bit 10: Receipt of Target Abort. This bit is set when a target abort is asserted during a Bus Master operation on the PCI Bus, that is, while the Bus Master is granted the PCI Bus.
- Bit 9: Posted Status Exception. This bit is set when a Local Bus exception occurs during a Posted Status operation. It is used together with Bits 7 to determine whether the data transfer was performed successfully, but the status was corrupted. The setting of this bit is mutually exclusive with the setting of Bit 8. Note that data parity detection does not apply to the posting of status. Therefore, only Bit 7 can be set concurrently with this bit.
- Bit 8: List Chaining Exception. This bit is set when a Local Bus exception or data parity error occurs during a List Chaining operation. It is used together with Bits 6 and 7 to determine whether the data transfer was performed successfully, but the list chain was corrupted. The setting of this bit is mutually exclusive with the setting of Bit 9. Note that either Bit 6 and/or 7 can be set concurrently with this bit.
- Bit 7: Exception. This bit is set when the -L_EXCPT signal is detected during a Bus Master transfer.
- Bit 6: Local Bus Parity. This bit is set when a CFE Local Bus parity error is detected during a Bus Master read. Note that this bit can never be set when Local Bus Parity is disabled in the PROC_CFG register.
- Bit 5: Loss of channel indicator. This bit is set when the Bus Master loses the PCI Bus. The PCI Bus is lost by resetting of the DMA Enable (SCP, Bit 1), or by resetting the Bus Master Enable (HSCR, Bit 2). All of these conditions stop any Bus Master activity. Enabling of the Bus Master channel also is blocked if either the DMA Enable or the Bus Master Enable are reset. Attempting to start the channel (that is, setting Bit 0 in the Channel Control Register (CCR)) with either of these bits disabled will force a termination interrupt and the setting of this status bit.
- Bit 4: Reserved. Always read 0.

- Bit 3: DEVSEL# Indicator. This bit is set when DEVSEL# is not detected on the PCI Bus during a Bus Master cycle, and the System Bus Interface Chip performs a master abort on the PCI Bus. This function is analogous and compatible with the Card Selected Feedback Return Indicator Bit in the System Bus Interface Chip.
- Bit 2: Receipt of PERR#. This bit is set when PERR# is asserted by a target during a Bus Master operation on the PCI Bus, that is, while the System Bus Interface Chip is granted the PCI Bus as a bus initiator. This function is analogous and compatible with the -CHCK Indicator function in the System Bus Interface Chip.
- Bit 1: Data Parity Error. This bit is set when a PCI Bus Data Parity Error has been detected. PCI Bus Data parity detection is enabled by HSCR, Bit 6. Note that this bit can never be set when PCI Bus data parity is disabled in the HSCR.
- Bit 0: Normal Termination. This bit is set during normal termination. This bit is reset by a read of the BMSTAT register, or by a list chain operation. Note that if Normal Termination interrupts are enabled, the interrupt associated with the setting of the bit is ***not*** reset by a list chain operation, but can only be reset by a read of the BMSTAT register.

Reset Conditions:

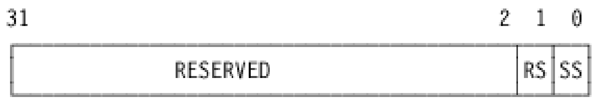
```
RST#:            0000 0000 0000 0000 0000 0000 0000 0000
Command Reset:   0000 0000 0000 0000 0000 0000 0000 0000
```

Bus Master Command Register (BMCMD1 (2))

This register is used to start the Bus Master channel or to reset internal Bus Master registers and control logic to a known state. During normal operation, it is not necessary to use this register.

Register Format:

```
(Channel 1 Address = 1FFA301C H)          32-bit rd/wr
(Channel 2 Address = 1FFA401C H)          32-bit rd/wr
```



Bit Descriptions:

- Bits 31-2: Reserved. Always read zero.
- Bit 1: Reset Bit. Setting this bit resets the channel's internal registers and control logic and holds the channel in this reset state. Resetting this bit releases the channel from the reset state for further operation. Setting this bit while the start/stop bit is set will force a noncatastrophic stop of the channel, in addition to resetting the channel.
- Bit 0: The Start/Stop bit is an alternate access to the Start/Stop bit in the CCR. This bit provides the ability to start the channel without corrupting the contents of the other bits in the CCR. More information concerning the Start/Stop bit is available in Section *Channel Control Register (CCR)* on page 141.

Reset Conditions:

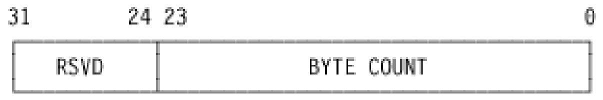
```
RST#:           0000 0000 0000 0000 0000 0000 0000 0000
Command Reset:  0000 0000 0000 0000 0000 0000 0000 0000
```

System Byte Count Register (SBCR)

This read-only register contains a 24-bit value that signifies the number of bytes that will be transferred ***on the PCI Bus*** before a terminal count is reached. This register is used for diagnostics and exception conditions in conjunction with the BCR to determine the number of bytes that have been transferred. This register is loaded from the BCR when a transfer is started and updated when the transfer reaches a normal or exception termination.

Register Format:

```
(Channel 1 Address = 1FFA3020 h)          32-bit read only
(Channel 2 Address = 1FFA4020 h)          32-bit read only
```



Bit Descriptions:

- Bits 31-24: Reserved.
- Bits 23-0: Byte Count. These bits contain the system byte count for the current transfer. These bits are valid after termination of the current transfer.

Reset Conditions:

```
RST#:           0000 0000 uuuu uuuu uuuu uuuu uuuu uuuu
Command Reset:  0000 0000 SSSS SSSS SSSS SSSS SSSS SSSS
```

PCI Memory Windows to CFE Address Space

Access to CFE address space is available from the PCI Bus memory address space. There are two independent windows into CFE address space from the PCI Bus memory map. The base address of each of these windows is set at configuration time in the configuration registers. These windows are enabled by setting the Memory Space Enable bit (Bit 1) in the Host Status/Command Register (HSCR).

Expansion ROM Window

The Expansion ROM window is used by applications requiring BIOS extensions somewhere in PCI Bus address space. The base address in PCI address space is set in the Expansion ROM Base Address Register (XRBAR) in configuration address space. More information on the (XRBAR) is provided inXRBAR9. The first location in this window points to the first location in CFE address space, located at the CFE address programmed in the Local Bus Base Address Register (LBBAR). By PCI definition, the 64KB window will be located on a 64KB address boundary.

Full Memory Window

The Full Memory window maps a 64MB window of CFE Bus address space into PCI memory address space. The base address for this window in PCI memory address space is set in the Host Memory Base Address Register (HMBAR) in PCI configuration address space. More information on the HMBAR is provided in HMBAR9. The first location in this window points to the CFE address programmed in the Local Bus Base Address Register (LBBAR). By PCI definition, the 64MB window will be located on a 64MB address boundary.

A memory map showing the relationship between the PCI memory windows and CFE Local Bus address space is given in Figure 9-3. Note that the two windows defined for the PCI Address space have separate base address registers defined in Configuration space, but map into one base address register (LBBAR) on the CFE Local Bus.

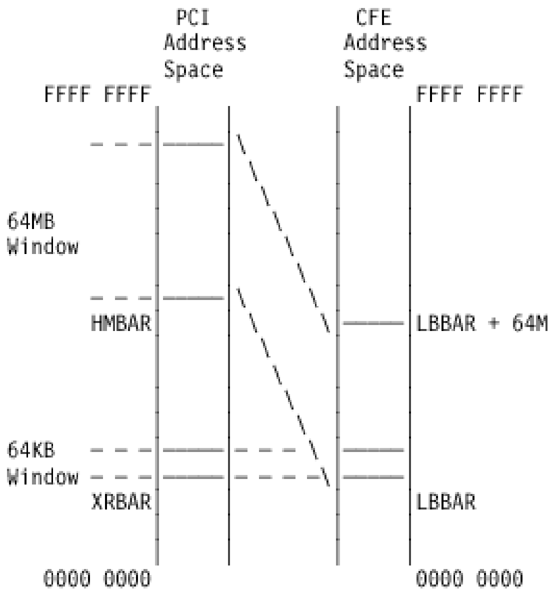


Figure 9-3. Shared Memory Windows

CFE Read Prefetching

The System Bus Interface Chip will prefetch data into the 128 byte slave read prefetch buffer when a PCI read of the System Bus Interface Chip occurs. The number of words that are prefetched varies based on the PCI read command and the logic level of the FRAME# signal on the PCI bus. If the FRAME# signal is at a logic *high* level immediately after the PCI address phase, the PCI master reading the System Bus Interface Chip is performing a single word read and only one word is prefetched. The following table shows the number of words prefetched for the various PCI Commands and the single transfer condition.

Table 9-9. Supported PCI Bus Commands

Read Condition	# of 32-bit Words Prefetched
Memory Read (C/BE# = 0110)	8[2]
Memory Read Line (C/BE# = 1110)	8[2]
Memory Read Multiple (C/BE# = 1100)	Until Buffer Full
Single PCI Transfer Condition	1

2 These transfers cannot be preempted on the CFE bus.

Bursting/Non-Bursting CFE Addresses

Transfers on the CFE Local Bus are generally burst accesses. That is, multiple data transfers occur for one initial address cycle with termination of the transfer indicated by the CFE Master's assertion of the Burst Last (-L_BLAST) signal. To allow interfacing to devices that do not support this bursting capability, the System Bus Interface Chip provides an area in the CFE address space reserved for nonburst access. Mapping of separate regions for Bursting and Non-Bursting access is similar to the configuration of the Intel 80960 Bus Controller access into separate 256MB address regions. This Burst/ Non-Burst CFE Local Bus mapping is summarized in Table 9-10.

Table 9-10. Burst/Non-Burst Address Map

CFE Local Bus Address	Access
0000 0000 - 1FFF FFFF	Non-Burst
2000 0000 - FFFF FFFF	Burst

This nonbursting area can be accessed through the two shared memory windows, by the Bus Master channels, through the MDATA register. In addition, for PCI target read access to this area through the MDATA register, the prefetch buffer associated with PCI target read accesses is filled by single transfers; that is, the prefetch mechanism is disabled. This nonbursting, nonprefetching mechanism provides the ability to access CFE Local Bus address areas that are affected by the prefetch mechanism. One example of this type of address area is self-clearing registers that may be corrupted by a prefetch read during an access to a contiguous address. Another example is a protected memory containing boundaries that may be crossed during a prefetch associated with a read from a contiguous location.

Slave Memory Register

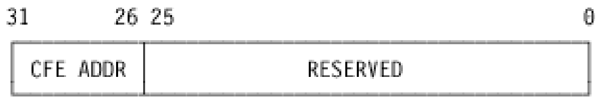
Local Bus Base Address Register (LBBAR)

The Local Bus Base Address Register supplies the base address in CFE Local Bus address space for the PCI Bus memory access options. This register provides the address bits A31A26. This window must be placed on a 64MB boundary.

This window is initialized by using the serial EPROM. More information on the serial EPROM can be found in *Serial EPROM* on page 16. More information on shared memory options can be found in *PCI Memory Windows to CFE Address Space* on page 147.

Register Format

```
(CFE Local Bus Address = 1FFA0024 h)          32-bit read/write
```



Bit Descriptions

- Bits 31-26: Address bits A31A26.
- Bits 25-0: Reserved.

Reset Conditions

```
RST#:            0010 0000 0000 0000 0000 0000 0000 0000
Command Reset:   SSSS SS00 0000 0000 0000 0000 0000 0000
```

Subsystem Control Block (SCB) Support

The System Bus Interface Chip supports both Locate Mode and Move Mode of the Subsystem Control Block (SCB) architecture. The SCB registers and their access for each mode is shown in Table 9-11.

Table 9-11. SCB Register Access Summary

	Normal Access within Operating Modes					
SCB Register	**Locate Mode**			**Move Mode**		
	System Master	Peer Adapter	Local Bus	System Master	Peer Adapter	Local Bus
COMMAND	W	None	R	W	None	R
ATTN	W	None	R	W	W	R
SCP	W	None	R	W	None	R
ISP	R	None	W	R *	None	W *
CBSP	R	None	W	R	None	W

* Currently no function defined for Move Mode

The overall objective of the SCB architecture is to provide a programming model for the PCI Bus, by defining the logical protocols for transferring commands, data and status between entities on the PCI Bus.

Locate Mode

In general, Locate Mode is used by the host to pass either an immediate command or the address of a command to the System Bus Interface Chip. After a command is passed by a PCI Bus write to the Command Port. The Attention Port is used to signal an interrupt to the CFE Local Bus. In addition, the Busy Bit of the Command Busy Status Port is set in the System Bus Interface Chip. The setting of this bit blocks further requests to the Command and Attention Ports. The Command and Attention Ports are serviced by the CFE Local Bus and the command is accessed. If the command is valid, the Busy Bit is reset by a write from the CFE Local Bus. If the command is rejected, the Reject Bit is set in the Command Busy Status Port.

The Subsystem Control Port is accessed by the PCI Bus, and is used to enable various functions of the System Bus Interface Chip. This register is used to enable PCI Bus interrupts, to enable DMA operation, to perform a Command Reset of the System Bus Interface Chip, and to reset the Reject state of the Command Busy Status Port.

In addition to these registers, the ISP and SIR support PCI Bus and additional CFE Local Bus interrupts.

Move Mode

Move Mode is used for peer-to-peer operation across the PCI Bus. In general, commands are not passed through the Command Port, but rather to a predefined location in a shared memory area. The Attention Port is used to signal an interrupt to the CFE Local Bus. An attention code of 'D' hex is used to identify the transfer as a Move Mode operation. A bit in the SIR is set corresponding to the device number passed with this Move Mode attention code. A full description of the operation of the SIR is available in Section *Source Identification Register (SIR)* on page 162.

Interrupts

PCI Bus interrupts (that is, interrupts presented to the PCI Bus from the System Bus Interface Chip) are asserted by setting the IV in the CBSP from the CFE Local Bus, or by Bus Master termination interrupts, if enabled. For signalling interrupts, status is provided by the CFE Local Bus in the Interrupt Status Port (ISP). This status is provided prior to setting the Interrupt Valid bit. For exception conditions, the Reject Bit in the CBSP is set along with the IV bit, and status for the exception is provided in the Status Bits (Bits 57) of the CBSP. Separation of exception and signalling conditions are thus obtained from one read of the CBSP from the PCI Bus.

PCI Bus interrupts are cleared by reading the Command Busy Status Port from the PCI Bus. When read, the Interrupt Valid bit is reset, as well as the pending interrupt.

Exceptions

Exceptions are reported by setting the Reject Bit together with the Status Bits in the CBSP, from the CFE Local Bus. There is only one Status code for hardware defined exceptions, '001'b. This exception is asserted for Watchdog Timeout errors, that is, when the Watchdog Timeout signal is detected asserted by the System Bus Interface Chip. For this exception, the Reject and Interrupt Valid bits, as well as Status, are set in hardware, and an interrupt is asserted to the PCI Bus.

PCI Slave (Target) Features

The PCI slave interface of the System Bus Interface Chip allows for both reads and writes into CFE address space through memory reads and writes on the PCI bus. Also, the CFE address space can be accessed through the MDATA register located in PCI I/O and memory mapped I/O space.

PCI Reads

the System Bus Interface Chip performs Delayed Read Operations and supports the following PCI bus commands for memory read:

- Memory Read (C/BE# = 0110)
- Memory Read Line (C/BE# = 1110)
- Memory Read Multiple (C/BE# = 1100)

When a PCI Bus Master performs a read to the System Bus Interface Chip, the System Bus Interface Chip immediately issues a target RETRY to the PCI Bus Master, and proceeds to prefetch data from the CFE bus into the prefetch buffer. While data is being prefetched into the buffer, and before there is enough data to begin the read transfer, the System Bus Interface Chip will retry all PCI read access attempts by any PCI master. The System Bus Interface Chip maintains a RETRY counter that is incremented each time a PCI master attempts a read access to an address different than the one whose delayed transfer is being prefetched. The retry counter is reset to zero when the original master (determined by address comparison) attempts a read access.

If the RETRY counter reaches a count of 30 RETRY operations, the pending delayed read is discarded and a new delayed read is initiated for the current PCI master. The retry counter is a fail-safe mechanism that prevents a lock-out condition if the original master whose read is pending does not come back.

The amount of data initially prefetched into the prefetch buffer, and the number of words that must be prefetched into the buffer before a read transfer can start, varies depending on the PCI bus read command and the value of the PCI Target Threshold bit in the SEER (see *Serial EPROM Extension Register (SEER)* on page 123). The System Bus Interface Chip also has the ability to detect a single transfer condition when FRAME# is driven high immediately after the address phase of a PCI read access. The following table describes the prefetching operation.

Table 9-12. Prefetch Operation

Read Condition	# of 32-bit Words Prefetched	Target Threshold bit	Start Threshold (# of 32-bit words)
Memory Read	8	-	4
Memory Read Line	8	-	4
Memory Read Multiple	Until Buffer Full	1	24
Memory Read Multiple	Until Buffer Full	0	4
Single PCI Transfer Condition	1	-	1

The Target Threshold bit is intended to provide some control over the prefetch buffer transfer start threshold to account for speed mismatches between the CFE bus and the PCI bus. The Target Threshold bit should be set to a 0 for maximum throughput, and set to a 1 to minimize the read data latency.

Once the read transfer begins, the System Bus Interface Chip has been designed to sustain the burst transfer rate of the CFE bus. If the CFE bus is running slower than the PCI bus, the System Bus Interface Chip will insert wait states on the PCI bus to allow the transfer to continue without termination. The System Bus Interface Chip can throttle down the PCI bus, by inserting TRDY, to approximately a 50MB/s rate (CFE running 25 MHz and 1 wait state). If the CFE bus speed drops below the 50MB/s burst transfer rate, the System Bus Interface Chip terminates the transfer with a DISCONNECT.

PCI Writes

The System Bus Interface Chip performs Posted PCI Write Operations. It accepts data presented by the PCI Bus Master immediately and stores it into the 128-byte slave write buffer. As soon as data arrives into the slave write buffer, the System Bus Interface Chip begins to flush the data onto the CFE bus.

The System Bus Interface Chip has been designed to sustain the burst transfer rate of the CFE bus. If the CFE bus is running slower than the PCI bus, the System Bus Interface Chip will insert wait states on the PCI bus to allow the transfer to continue without termination. The System Bus Interface Chip can throttle down the PCI bus, by inserting TRDY, to approximately a 50MB/s rate (CFE running 25 MHz and 1 wait state). If the CFE bus speed drops below the 50MB/s burst transfer rate, the System Bus Interface Chip terminates the transfer with a DISCONNECT.

Target Abort Conditions

Abort conditions include:

- Exception conditions detected on the CFE Local Bus during a slave read from the PCI
- CFE Local Bus parity errors detected during a slave read from the PCI Bus
- Access to the System Bus Interface Chip during a COMMAND RESET
- Mismatch of PCI byte enables and address bits on I/O transfers.

System Errors (SERR#)

Address(/Command) parity errors force the assertion of SERR#, and the setting of the Signalled System Error Bit (HSCR, Bit 30). The assertion of SERR# and setting of this bit are under control of the System Error Response Bit (HSCR, Bit 8). Responses to parity errors in general are under control of the Parity Error Response Bit (HSCR, Bit 6). Both Bits 6 and 8 must be enabled to signal SERR#. Parity errors also are indicated by the Detected Parity Error Bit (HSCR, Bit 31). This bit is set for all parity errors, regardless of the state of the Parity Error Response Bit. Note that for address parity errors, the System Bus Interface Chip does not assert DEVSEL#, forcing a Master Abort.

Irregular PCI Byte Enables

A PCI Bus Master has the ability to assert irregular byte enables, such as C/BE(3) and C/BE(0) only. For these accesses, the System Bus Interface Chip will complete the transfer with a disconnect, unless the access is a single word transfer.

Host-Slave Base Address Register (HSBR)

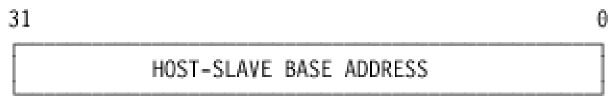
The Host-Slave Base Address register stores the physical base address of the CFE Local Bus address space accessed by the host. The host initializes this register before accessing the Memory Data Register (MDATA) for memory transfers. Continuous accesses from MDATA transfer data to or from subsequent CFE address locations based on the value in this register. The value in the HSBR is auto-incremented by the number of bytes transferred. More information on the Memory Data register is available in *Memory Data Register (MDATA)* on page 154.

Reading and Writing Registers

Note that it is possible to read and write the System Bus Interface Chip internal registers using this mechanism. These accesses must be 32-bit accesses.

Register Format

```
(PCI Mem or I/O Offset = Base + 0C-0F h)    32-bit read/write
```



Reset Conditions

```
         RST#:  uuuu uuuu uuuu uuuu uuuu uuuu uuuu uuuu
Command Reset:  SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSSS
```

Memory Data Register (MDATA)

The Memory Data Register is used in conjunction with the Host-Slave Base Address Register (HSBR), for PCI Bus accesses between the PCI Bus host and the CFE Local Bus. When the Memory Data register is accessed, depending on the direction of transfer, data is written to or read from the location in CFE address space stored in the HSBR. For host transfers, subsequent accesses to MDATA after initialization of the HSBR are made to subsequent locations in memory; that is, the HSBR is auto-incremented after every transfer.

Operation

The intended use of the HSBR and the MDATA register is for small transfers between the host system and the CFE Local Bus. As a result, data steering to the MDATA register is limited. Although the Host Slave Base Address register (HSBR) is used for addressing the CFE Local Bus, the MDATA register does not provide data steering based on the value stored in this register. Rather, the MDATA register passes data to the CFE Local Bus as it is steered on the PCI Bus, based on the PCI Bus address used to access this register.

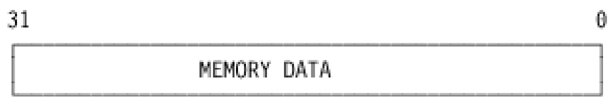
The bytes of the MDATA register are separately addressable; therefore, all byte-enable combinations are supported. Byte accesses are supported, as well as 3-byte contiguous addressing (that is, C/BE(2)C/BE(0) asserted, or C/BE(3)C/BE(1) asserted). For example, to access an odd word in CFE Local Bus address space (an address located on an odd-word boundary), the HSBR is first loaded with this odd-word address. To access the odd word, the MDATA register is subsequently accessed as a word register at PCI Bus location 0012 hex. To access a byte at this CFE Local Bus location, the MDATA register is accessed as a byte register at location 0012 hex.

The byte enables on the PCI Bus must also match the address in the HSBR to guarantee valid data.

Note that the System Bus Interface Chip performs a PCI target disconnect after each transfer from the PCI to the MDATA port, so read accesses limit the prefetch buffer to single transfers. More information on the nonbursting address space can be found in Section *PCI Memory Windows to CFE Address Space* on page 147.

Register Format

```
(PCI Mem or I/O Offset = Base + 10 h)        32-bit read/write
```



Reset Conditions

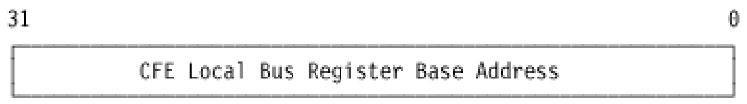
```
         RST#:  uuuu uuuu uuuu uuuu uuuu uuuu uuuu uuuu
Command Reset:  SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSSS
```

CFE Base Address Register (CFEBAR)

This register provides the PCI Bus with the base address of the System Bus Interface Chip register space. This register defaults to a value of 1FFA 0000h on reset, but may be loaded with any value 1FFx 0000h from the serial EPROM register. More information on the serial EPROM interface can be found in *Serial EPROM* on page 16.

Register Format

```
(PCI Mem or I/O Offset =  Base + 14h)        32-bit read-only
```



Bit Descriptions

Bits 31-0: Register Base Address. These bits contain the CFE Local Bus register base address. Bits 1916 default to '1010', but may be updated by serial EPROM.

Reset Conditions

```
         RST#:   0001 1111 1111 1010 0000 0000 0000 0000
Command Reset:   SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSSS
```

PCI Master (Initiator) Features

The System Bus Interface Chip acts as a PCI master for the DMA (Bus Master) channels. The System Bus Interface Chip does not perform master pacing (de-assert IRDY# to insert wait states) on the PCI bus. When the internal DMA buffers do not have adequate data to continue, the System Bus Interface Chip terminates the transfer on the PCI bus and waits for data to accumulate into the DMA buffers (see *DMA Thresholds* on page 133 for more information) before requesting the PCI bus.

In addition to the standard PCI bus commands, the System Bus Interface Chip supports the following PCI bus read commands.

Table 9-13. Read Commands

Read Command	# of Bytes to Be Read
Memory Read	fewer than 16 bytes
Memory Read Line	equal to 16 bytes
Memory Read Multiple	greater than 16 bytes

The support of these commands is intended to allow the PCI slave to control the amount of data it needs to gather (prefetch) when it is accessed.

Target Abort and Parity Function

The System Bus Interface Chip detects Target Abort when operating as a PCI Bus Master. Detection of an Abort stops the Bus Master channel and forces a termination interrupt to the CFE Local Bus.

PCI Bus parity detection is also supported when operating as a Bus Master. The detection of a parity error forces a termination interrupt to the CFE Local Bus and stops the Bus Master channel, as well as the assertion of the -PERR signal.

PCI Parity is controlled by the the Parity Response Bit (Bit 6) in the Host Status/ Command Register (HSCR)

System Errors (-SERR)

The System Bus Interface Chip detects the assertion of SERR# as a Bus Master when it is granted the PCI Bus. Detection of SERR# forces the current transfer to terminate with Receipt of SERR# bit set in the BMSTAT register (BMSTAT, Bit 11).

SCB Registers

Command Port (COMMAND)

The Command Port is used to deliver a 32-bit immediate command or the physical address of a control block to a feature adapter on the PCI Bus. The Command Port is protected by both the Reject Bit (Bit 4) and the Busy Bit (Bit 0) in the Command Busy Status Port. If either of these two bits are set, the Command Port write will be ignored. More information on control block architecture can be found in Section *Subsystem Control Block (SCB) Support* on page 150.

Register Format

```
(CFE Local Bus Address = 1FFA2000 h)      32-bit read-only
(PCI Mem or I/O Offset = Base + 00 h)     32-bit read/write
```

31
0
SCB COMMAND

Bit Descriptions

Bits 31-0: Immediate Command or address of a control block.

Reset Conditions

```
         RST#:  uuuu uuuu uuuu uuuu uuuu uuuu uuuu uuuu
Command Reset:  SSSS SSSS SSSS SSSS SSSS SSSS SSSS SSSS
```

Attention Port (ATTN)

The Attention Port is used by the PCI Bus to signal, or request the attention of the CFE Local Bus. A PCI Bus write to the Attention Port causes data to be latched in the Attention Port and an attention interrupt to be posted to the CFE Local Bus. Data is latched in the Attention Port and the Busy Bit is set for all attention codes, except for the special attention code 'D'. Reading the register from the CFE Local Bus clears the interrupt. More information on attention interrupts is available in Section *System Bus Interface Chip Interrupts* on page 164. The Attention Port is protected by both the Reject Bit (Bit 4) and the Busy Bit (Bit 0) in the Command Busy Status Port. If either of these two bits are set, the Attention Port write will be ignored, and no attention interrupt occurs.

Attention Code 'D'

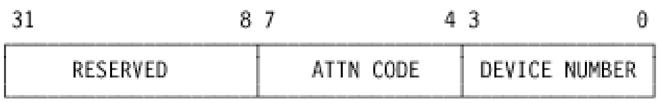
The operation of attention code 'D' is different from all other attention codes. This attention code is used in Move Mode to set individual bits in the Source Identification Register (SIR). Either a General Interrupt or an End-of-Data-Transfer Interrupt is issued in place of the Attention Interrupt. The setting of bits in the SIR with this attention code is ***not*** blocked by the Reject or Busy bits. Note that the busy bit is ***not*** set in the CBSP for this attention code. More information on the setting of these bits and the subsequent interrupts is available in Sections *Source Identification Register (SIR)* on page 162 and *Subsystem Control Block (SCB) Support* on page 150.

General information on the Subsystem Control Block (SCB) architecture is available in Section *Subsystem Control Block (SCB) Support* on page 150.

Register Format

```
(CFE Local Bus Address = 1FFA2004 h)       32-bit read-only
                                           8-bit field defined
```

```
                                      by SCB architecture
(PCI Mem or I/O Offset = Base + 04 h)      8-bit read/write
                                              (Bits 7-0)
```



Bit Descriptions

- Bits 31–8: Reserved.
- Bits 7–4: Attention Code. The attention code is used to indicate to the CFE Local Bus the specific action to be initiated. For SCB Move Mode, the attention code 'D' hex is used to set bits in the SIR.

  Valid attention codes are shown in Table 9-14.

Table 9-14. Attention Codes

Attention Code	Device Number	Name	Description
0	X	Reset Command	Request the subsystem to perform Device Reset for the specified device.
1	X	Immediate Command	Requests the subsystem to execute the command contained in the Command port.
2	X		Reserved
3	X	Start Control Block Command	Requests the subsystem to process the control block pointed to by the address in the Command port.
4		Device Dependent	
5-C			Reserved
D	0F	Move Mode Delivery	Used to signal request for Move Mode command delivery. The device number specifies the bit to be set in the SIR.
E	0	End of Interrupt	Requests the subsystem to perform one of the two interrupt resetting commands.
F		Device Dependent	

X = Do not care. A blank in the Device Number equals unspecified.

- Bits 3-0: Device Number. The device number indicates a specific device to which the attention code is directed. For SCB Move Mode, this device number, together with an attention code of 'D' hex, indicates the bit to be set in the SIR. More information concerning the setting of bits in the SIR is available in Section *Source Identification Register (SIR)* on page 162.

Reset Conditions

```
RST#:          0000 0000 0000 0000 0000 0000 0000 0000
Command Reset: 0000 0000 0000 0000 0000 0000 SSSS SSSS
```

Subsystem Control Port (SCP)

The Subsystem Control Port is used to support commands to the System Bus Interface Chip from the PCI Bus. General information on the Subsystem Control Block (SCB) architecture is available in Section *Subsystem Control Block (SCB) Support* on page 150.

Register Format

```
(CFE Local Bus Address = 1FFA2008 h)         32-bit read-only
                                             8-bit field defined
                                             by SCB architecture
(PCI Mem or I/O Offset = Base + 05 h)        8-bit read/write
                                               (Bits 7-0)
```

Bit Descriptions

- Bits 31–8: Reserved.
- Bit 7: Command Reset. Setting this bit causes a Command Reset. The COMMAND RESET OUT signal is asserted. In addition, the busy bit in the Command Busy Status Port is set (and stays set) during this reset. This bit should be set for a minimum of 50 µsec before being reset. Register values after this reset are provided in the individual register descriptions. Note that while this bit is set, accesses to the System Bus Interface Chip will cause a target abort, except access to this register.
- Bit 6: Clear on Read Bit (Not Implemented). This bit is defined by the SCB architecture to select between two methods of clearing interrupts on the PCI Bus. the System Bus Interface Chip enables the clear on read option as the only method of clearing interrupts on the PCI Bus. Interrupts and their status will be cleared by reading the Command Busy Status Port (CBSP).
- Bit 5: Reset reject. Setting this bit resets the Busy Bit and the Reject Bit in the Command Busy Status Port. This bit is self-clearing.
- Bit 4: Reserved. This signal is read/write accessible from the PCI Bus and read only accessible from the CFE Local Bus.
- Bits 3-2: Device Dependent. This signal is read/write accessible from the PCI Bus and read only accessible from the CFE Local Bus.
- Bit 1: Enable DMA. Setting this bit enables Bus Master operation.
- Bit 0: Enable Interrupts. Setting this bit enables interrupts to the PCI Bus.

Reset Conditions

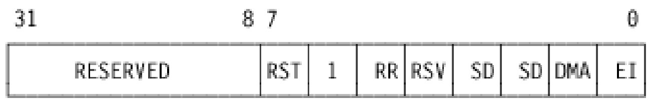
```
RST#:            0000 0000 0000 0000 0000 0000 0100 0000
Command Reset:   0000 0000 0000 0000 0000 0000 1100 0000
```

Interrupt Status Port (ISP)

The Interrupt Status Port is used to present interrupt data to the PCI Bus. General information on the Subsystem Control Block (SCB) architecture and interrupt handling is available in Section *Subsystem Control Block (SCB) Support* on page 150.

Register Format

```
(CFE Local Bus Address = 1FFA200C h)        32-bit read/write
                                            8-bit field defined
                                            by SCB architecture
(PCI Mem or I/O Offset = Base + 06 h        8-bit read-only
                                            (Bits 7-0)
```



Bit Descriptions

- Bits 31-8: Reserved.
- Bits 7-4: Interrupt Identifier. These bits identify the cause of the interrupt.

  Valid interrupt codes are shown in Table 9-15.

Table 9-15. Interrupt Identifier Codes

Hex Value	Interrupt Definition
0	Reset Subsystem/Device Completed No Error
1	Control Block Command Completed No Error
2	Notify Event
3	Reserved
4	Reserved
5	Device Dependent
6	Inform Event
7	Hardware Failure Immediate Command or Hardware Control
8	Hardware Failure Control Block Command
9	Reserved
A	Immediate Command/Hardware Control Completed No Error
B	Reserved
C	Control Block Command Completed w/Error
D	Immediate Command/Hardware Control Completed w/Error
E	Command Rejected
F	Device Dependent

- Bits 3-0: Device Number. The device number identifies the specific device providing the interrupt status.

Reset Conditions

```
RST#:          0000 0000 0000 0000 0000 0000 uuuu uuuu
Command Reset: 0000 0000 0000 0000 0000 0000 SSSS SSSS
```

Command Busy Status Port (CBSP)

The Command Busy Status Port has two functions:

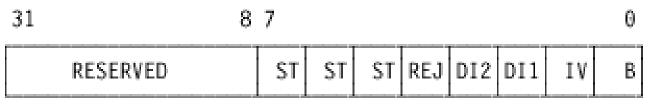
- The Port is read by the PCI Bus after an immediate command or Control Block address is written to the Command Port to determine the status of the command.
- The Port indicates that an interrupt to the PCI Bus is pending.

General information on the Subsystem Control Block (SCB) architecture and interrupt handling is available in Section *Subsystem Control Block (SCB) Support* on page 150.

Register Format

```
(CFE Local Bus Address = 1FFA2010 h)       32-bit read/write
                                           8-bit field defined
                                           by SCB architecture

(PCI Mem or I/O Offset = Base + 07 h)      8-bit read-only
                                              (Bits 7-0)
```



Bit Descriptions

- Bits 31–8: Reserved.
- Bits 7–5: Status. These bits are used to encode a rejection code for the current command. When the Reject and Busy Bits are set, these bits reflect status for exception conditions. Only one exception condition is defined in hardware, '001'b. This exception indicates that the Watchdog Timeout signal has been detected by the System Bus Interface Chip.
- Bit 4: Reject. This bit indicates that the current command has been rejected. PCI Bus writes to the Attention (ATTN) and Command (COMMAND) Ports are blocked when this bit is set, with the exception of the 'D' hex attention code. More information on attention codes is available in Sections *Attention Port (ATTN)* on page 157 and *Subsystem Control Block (SCB) Support* on page 150. If this bit is set, together with the IV Bit, an exception is reported in the Status bits (Bits 75).
- Bits 3–2: DMA Channel 1(2) Interrupt Bits. These bits are enabled by the DMA Channel 1(2) Interrupt Direction Bits (PROC_CFG, Bits 1514). These bits, when enabled, are set upon the termination of their respective DMA channel. They function in the same manner as the IV bit; that is, they generate an interrupt to the PCI Bus, function as primary status for the interrupt, and are cleared by the same mechanism as the IV bit. These bits operate independent of each other and the IV bit. Either or both

of these bits can be set in addition to the IV bit. They will all be cleared by the same clearing operation. These bits are defined as Device Dependent by the SCB architecture. The operation of these bits is not a part of the SCB architecture.

- Bit 1: Interrupt Valid. Setting this bit forces an interrupt to the PCI Bus. If this bit is set and the Reject Bit (Bit 4) is reset, the bit indicates that interrupt status is available in the Interrupt Status Port (ISP). If this bit is set and the Reject Bit is set, a status code is available in the status bits of the Command Busy Status Port. The ISP is used for normal signalling interrupts, and the CBSP is used under exceptions conditions. There is only one exception status code defined for hardware. This bit is reset by reading the CBSP. More information on the SCB and interrupt support is available in Section "Subsystem Control Block (SCB) Support".
- Bit 0: Busy. This bit indicates that Command and Attention ports are currently being used, or a hardware control Subsystem Reset is in progress or has just completed. This bit is set by a write to the Attention Port, (except for the 'D' h attention code). PCI Bus writes to the Attention (ATTN) and Command (COMMAND) Ports are blocked when this bit is set.

Reset Conditions

```
RST#:           0000 0000 0000 0000 0000 0000 0000 0001
Command Reset:  0000 0000 0000 0000 0000 0000 SSS0 0001
```

Source Identification Register (SIR)

The Source Identification Register is used in Move Mode only. The register is directly accessible from the CFE Local Bus only. Bits in the SIR are set indirectly through the Attention Port on the PCI Bus. Specifically, a write to the Attention Port of attention code 'D' hex indicates the setting of a bit in the SIR. The device number passed in the Attention Port indicates the bit to be set in the SIR. The bits in the SIR are identified as bits from the least-significant bit to the most-significant bit as 0F hex. A write to the Attention Port of 'DA' hex would set bit 10 in the SIR.

By CFE Local Bus definition, and independent of SCB architecture, the Source Identification Register is used to issue both general interrupts and end-of-transfer interrupts from the PCI Bus to the CFE Local Bus, saving status for these interrupts.

Bits in the SIR can only be set from the PCI Bus. Each bit is set through the Attention Port. Interrupts are generated by the logical OR of the bits in the SIR. A General Interrupt is generated by the logical OR of the lowest 8 bits in the register; an End-of-Data-Transfer Interrupt for the upper 8 bits in the register.

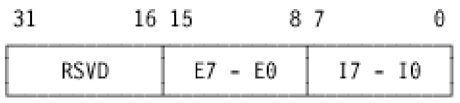
Bits in the SIR can only be reset from the CFE Local Bus. Each bit is reset by writing a logical '0' to the corresponding bit location. Writes of '1' from the CFE Local Bus to the SIR are ignored.

This manner of resetting bits allows the PCI Bus to generate interrupts to the CFE Local Bus through the SIR without losing interrupt status. More information on CFE Local Bus interrupts is available in Section *System Bus Interface Chip Interrupts* on page 164. More information on the Subsystem Control Block (SCB) architecture is available in Section *Subsystem Control Block (SCB) Support* on page 150.

Register Format

```
(CFE Local Bus Address = 1FFA2014 h)        32-bit read/write
                                            16-bit field defined
                                            by SCB architecture

(PCI Mem or I/O Offset = Indirect Access through ATTN)
```



Bit Descriptions

- Bits 31–16: Reserved.
- Bits 15–8: End-of-Data-Transfer Interrupt Bits.
- Bits 7–0: General Interrupt Bits.

Reset Conditions

```
RST#:          0000 0000 0000 0000 0000 0000 0000 0000
Command Reset: 0000 0000 0000 0000 0000 0000 0000 0000
```

Miscellaneous PCI Functions and Registers

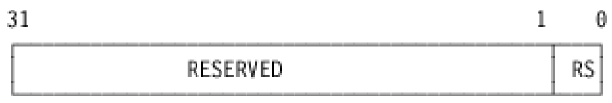
The System Bus Interface Chip handles two miscellaneous functions involving the interface between the co-processor adapter and the host system. The Reset Status Register is used to distinguish between Command Resets, and other resets. The Non-Maskable Interrupt Register is used to present NMIs from the system to the 80960. These registers are described in the following sections.

Reset Status Register (RSR) (PCI)

The Reset Status Register supplies status for a reset of the System Bus Interface Chip. A Command Reset sets a bit in this register. This bit is reset by a RST#. The intent of the Reset Status Register is to provide a means in software of distinguishing between Power Up or RST# and Command Resets.

Register Format

```
(CFE Local Bus Address = 1FFA0014 h)          32-bit read-only
```



Bit Descriptions

- Bits 31-1: Reserved.
- Bit 0: Reset Status. This bit indicates that a Command Reset has occurred.

Reset Conditions

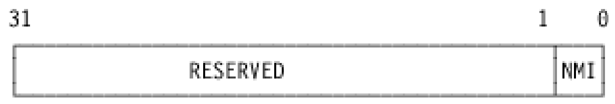
```
RST#:           0000 0000 0000 0000 0000 0000 0000 0000
Command Reset:  0000 0000 0000 0000 0000 0000 0000 0001
```

Non-Maskable Interrupt Register (NMI)

The Non-Maskable Interrupt Register is used for presenting a Non-Maskable Interrupt from the PCI Bus to the CFE Local Bus. Writing the register from the PCI Bus creates an interrupt of value '0' hex on the encoded interrupt lines, INT 30, as well as setting a status bit in Bit 0 of the register. Reading the register from the CFE Local Bus clears this status. The intent of the register is to provide a high-priority interrupt separate from the Attention and SIR interrupts. Any priority could be assigned to this interrupt, however, depending on the implementation of the interrupt controller. More information on CFE Local Bus interrupts is available in Section *System Bus Interface Chip Interrupts* on page 164.

Register Format

```
(CFE Local Bus Address = 1FFA001C h)      32-bit read-only
(PCI Mem or I/O Offset = Base + 0B h)     8-bit read/write
```



Bit Descriptions

- Bits 31-1: Reserved.
- Bit 0: NMI Status Bit. This bit indicates that a high priority interrupt is pending from the PCI Bus.

Reset Conditions

```
RST#:           0000 0000 0000 0000 0000 0000 0000 0000
Command Reset:  0000 0000 0000 0000 0000 0000 0000 0000
```

System Bus Interface Chip Interrupts

The System Bus Interface Chip chip interrupts the CFE Local Bus under seven conditions. The System Bus Interface Chip presents each of these seven interrupt sources as encoded interrupts on the CFE Local Bus interrupt lines, INT(30). These encoded interrupt lines can be used by external hardware to generate interrupts and vectors to a device present on the CFE Local Bus. The lines do not represent a valid vector for any particular device or microprocessor.

Each source of interrupt and its corresponding interrupt code is present in Table 9-16. Note that although the Inactive State for the encoded interrupt signals is 1111, implying that the signals are active low, the signals have been given active high signal designations (INT(03)). The values in this table represent the actual state of the interrupt signals (0 = Low, 1 = High).

Table 9-16. CFE Local Bus Interrupt Sources

Interrupt Source	Encoded Interrupt INT(3-0)
Non-Maskable Interrupt (NMI)	0000
Local Bus Parity/Exception	0001
SCB Attention Port	0010
End-of-Data-Transfer	0011
General Interrupt	0100
Bus Master Channel 2	0101
Bus Master Channel 1	0110
Reserved	0111-1110
Inactive State	1111

Each encoded interrupt is held active on the interrupt lines until the clearing procedure for that interrupt is performed. Until the current interrupt is cleared, all other interrupts asserted internally are held pending. After an interrupt is cleared, the highest priority pending interrupt is asserted on the encoded interrupt lines. After this interrupt is cleared, the next priority interrupt is serviced. This operation continues until all interrupts are serviced in order of priority. Interrupts for conditions occurring simultaneously are presented in the order of their encoded value. For example, if a General Interrupt occurs simultaneous to a Bus Master Channel interrupt, the General Interrupt would be presented first. The only exception to this procedure is the NMI command. This interrupt is always presented immediately.

The clearing procedure varies with the source of the interrupt:

- Non-Maskable Interrupts and CFE Local Bus Parity/Exception Interrupts. The interrupt is cleared when the corresponding status register is read.
- SCB Attention Port Interrupts. The interrupt is cleared when the Attention Port is read.
- End-Of-Data-Transfer and General Interrupts. The interrupt is cleared when the SIR is written from the CFE Local Bus, resetting a bit. The source of each of these interrupts is the OR of their respective status bits. The interrupt, therefore, will be reasserted after a write from the CFE Local Bus, until the OR of the status bits is equal to zero; that is, all status bits have been reset.

  Bus Master Channel 1 and Channel 2 Interrupts: The interrupt is cleared by reading the associated channel status register, BMSTAT1(2).

Miscellaneous Registers

10

A field-programmable gate array is used to implement miscellaneous functions provided by the co-processor platform. These functions include:

- Write enabling of the AIB ROM and the co-processor platform (base) ROM
- Control of a light emitting diode (LED)
- Memory presence detect bits for the instruction and packet memory
- A presence detect bit for the AIB.

Programming Considerations

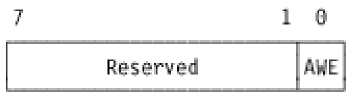
A write command by the 80960 processor to the registers listed in this chapter must be done using “byte” instructions or the results will be unpredictable.

AIB ROM Write Enable Register (AWE)

This register is used to prevent accidental writes to the AIB ROM by forcing a user to set a bit outside of ROM address space in order to enable writes to the ROM address space.

Register Format

```
(80960 ADDRESS = 09FF FFF4h)  8-bit read/write
```



Bit Descriptions

- Bits 7–1: Reserved. These bits must be set to 0.
- Bit 0: AIB ROM Write Enable. This bit enables or prevents writes to the AIB Flash ROM; the bit powers up reset.

  0 = Prohibited

  1 = Enabled

Reset Conditions

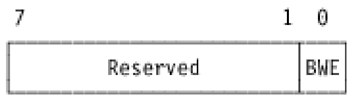
```
                Reset: UUUU UUU0
ROM Initialization: 0000 0000
```

Base ROM Write Enable Register (BWE)

This register is used to prevent accidental writes to the co-processor platform (base card) ROM by forcing a user to set a bit outside of ROM address space in order to enable writes to the ROM address space.

Register Format

```
(80960 ADDRESS = 09FF FFF0h) 8-bit read/write
```



Bit Descriptions

- Bits 7–1: Reserved. These bits must be set to 0.
- Bit 0: Base ROM Write Enable. This bit enables or prevents writes to the base card Flash ROM; the bit powers up reset.

  0 = Prohibited

  1 = Enabled

Reset Conditions

```
             Reset:  UUUU UUU0
ROM Initialization:  0000 0000
```

Enable/Detect Register (EDR)

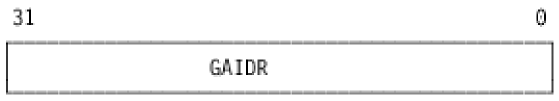
See *Enable/Detect Register (EDR)* on page 46 for a description of each bit in this register.

Memory Controller Chip Gate Array ID Register (GAIDR)

The contents of this register show the version of the Memory Controller Chip. For example, version 2 of the Memory Controller Chip will read 0000 0002h from this register.

Register Format

```
(Address = 1FFB A000h)   32-bit read-only
```



Reset Conditions

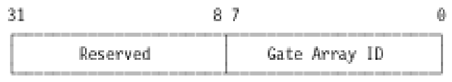
```
Reset: 0000 0000 0000 0000 0000 0000 ---- ----
                       (where - equals the Version)
```

Micro Channel Interface Chip Gate Array Identification (GAID)

The Gate Array Identification Register allows either the Micro Channel or the resident processor to determine the version of the Micro Channel Interface Chip. For example, Version 2 of the Micro Channel Interface Chip will read 0000 0002h from this register. The GAID is a read-only register.

Register Format

```
(Local Bus Address = 1FFA 0008h)         32-bit read-only
(Micro Channel Address = Base + 0Ah)      8-bit read-only
```



Bit Descriptions

- Bits 31–8: Reserved. These bits must be set to 0.
- Bits 7–0: Gate Array Version Identification. (Version 2.0 = 0000 0010)

Reset Conditions

```
        Reset: 0000 0000 0000 0000 0000 0000 ---- ----
Command Reset: 0000 0000 0000 0000 0000 ---- ---- ----
                       (where - equals the Version)
```

System Bus Interface Chip Gate Array Identification (GAID)

The Gate Array Identification Register allows either the PCI or the resident processor to determine the version of the System Bus Interface Chip. For example, Version 2 of the System Bus Interface Chip will read 0000 00F2h from this register. The GAID is a read-only register.

Register Format

```
(Local Bus Address = 1FFA 0008h)         32-bit read-only
(PCI Address = Base + 0Ah)  8-bit read-only (least-significant bit)
```

Bit Descriptions

- Bits 31–8: Reserved. These bits must be set to 0.
- Bits 7–0: Gate Array Version Identification. (Version 1.0 = 0000 00F0)

Reset Conditions

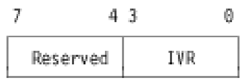
```
        Reset: 0000 0000 0000 0000 0000 0000 ---- ----
Command Reset: 0000 0000 0000 0000 0000 0000 ---- ----
                       (where - equals the Version)
```

Interrupt Controller Version Register (IVR)

The contents of this register show the version of the Interrupt Controller. For example, version 2 of the Interrupt Controller will read 02h from this register.

Register Format

```
(Address = 0A00 0000h)  8-bit read-only
```



Reset Conditions

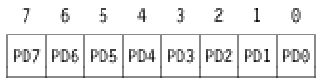
```
Reset: UUUU ----
           (where - equals the Version)
```

Memory Presence Detect Low Register (MPL)

The MPL Register is used in conjunction with the Memory Presence Detect High Register (MPH) to allow two external memory SIMMs to have their presence detect bits be readable by the 80960 processor.

Register Format

```
(80960 ADDRESS = 0A00 0008h)  8-bit read-only
```



Bit Descriptions

- Bits 7–0: SIMM Presence Detect Bits. Presence detect bits 5–0 are wired to the packet memory. Presence detect bits 7–6 are the low-order bits of the instruction memory detect. See *Memory Presence Detect High Register (MPH)* on page 171.

Reset Conditions

```
Reset: ---- ----
             (where - equals the Presence Detect value)
```

Memory Presence Detect High Register (MPH)

The MPH Register is used in conjunction with the Memory Presence Detect Low Register (MPL) to allow two external memory SIMMs to have their presence detect bits be readable by the 80960 processor.

Register Format

```
(80960 ADDRESS = 0A00 000Ch)  8-bit read-only
```

7 4	3	2	1	0
Reserved	PD11	PD10	PD9	PD8

Bit Descriptions

- Bits 7–4: Reserved. These bits must be set to 0.
- Bits 3–0: SIMM Presence Detect Bits. Presence detect bits 11–8 are wired to the instruction memory. Presence detect bits 7–6 are the low order bits of the instruction memory detect. See *Memory Presence Detect Low Register (MPL)* on page 170.

Reset Conditions

```
Reset: UUUU ----
             (where - equals the Presence Detect value)
```

Memory Configuration Register (MCR)

This register tells the Memory Controller Chip the amount of packet memory and/or instruction memory that is installed. This register also selects where instruction memory resides in the memory map and the input oscillator speed.

Register Format

```
(ADDRESS = 1FFB A004h) read/write - byte addressable
```

31 16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV	IM INS	IM LOC	RSV	RSV	RSV	IM SZ2	IM SZ1	IM SZ0	PM INS	REF	OSC 25	RSV	RSV	PM SZ2	PM SZ1	PM SZ0

Bit Descriptions

- Bits 31–16: Reserved. These bits must be set to 0.
- Bit 15: IM INS – Denotes whether memory is installed.

  0 = not installed.

  1 = installed.

  Instruction memory on the co-processor platform **must** start at address 2200 0000h.

- Bit 14: IM LOC – Determines location of instruction memory in memory map.

0 = starting address is 2200 0000h.

1 = starting address is 2400 0000h.

- Bits 13–11: Reserved. These bits must be set to 0.
- Bits 10–8: IM SZ2:0 – Denotes size of instruction memory installed.

SZ2	SZ1	SZ0	Memory Size	Organization	Addressing
0	0	0	1MB	1 Bank	9/9
0	0	1	2MB	2 Banks	9/9
0	1	0	4MB	1 Bank	10/10
0	1	1	8MB	2 Banks	10/10
1	0	0	16MB	1 Bank	12/10 or 11/11
1	0	1	32MB	2 Banks	12/10 or 11/11
1	1	x	Reserved	Reserved	- -

- Bit 7: PM INS – Denotes whether memory is installed.

  0 = not installed.

  1 = installed.
- Bit 6: REF – Denotes rate at which to refresh memory.

  0 = normal refresh rate (row every 15.6 us).

  1 = double refresh rate (row every 7.8 us).
- Bit 5: OSC25 – Tells the Memory Controller Chip the speed at which it is running. This is used for the timers (including refresh timer and debug baud rate generator).

  0 = Reserved.

  1 = 25MHz.
- Bits 4–3: Reserved. These bits must be set to 0.
- Bits 2–0: PM SZ2:0 – Denotes size of packet memory installed. (See previous table for bit descriptions.)

Reset Conditions

```
             Reset: 0000 0000 0000 0000 1000 0101 1100 0101
ROM Initialization: 0000 0000 0000 0000 1000 XXXX 1010 YYYY
             (where X = size of instruction memory and
                    Y = size of packet memory)
```

Debug Features

11

Serial Debug Port

The Memory Controller Chip module has a fixed-function Universal Asynchronous Receiver Transmitter (UART) built into the chip to eliminate the need for a chip of this type during adapter code debug. This is not intended to be used in an operational system environment, but rather only as a tool in a lab environment during the debug phase of code development. The features of this port are:

- Asynchronous operation
- Tx data and Rx data lead support only
- No modem controls
- Two-byte transmitter buffer + shift register
- Two-byte receiver buffer + shift register
- Rx byte available and Tx FIFO empty interrupting
- Rx overrun and Tx FIFO full status reporting
- Eight-bit character length
- No parity
- One start bit, one stop bit
- Programmable data rates of 9600, 19.2K, and 38.4K baud.

The transmitter and the receiver data and status registers are described in the *Transmitter Buffer Port (TxBUF)* on page 174 and the *Receiver Buffer Port (RxBUF)* on page 175. The transmitter and the receiver can be separately disabled and enabled, or the two can be tied together in a wrap mode for test, as described in the *Port Configuration Register (PCR)* on page 176.

Hardware Requirements for Debug Cable

The debug port requires a 3-wire cable. The end of the cable that connects to the co-processor platform uses a 3-pin receptacle assembly (AMP P/N 103959-2). The system end uses a 25-pin, female D-shell (AMP P/N 207463-1). The wiring for the debug cable is as follows:

Pin 1 of receptacle assembly connects to pin 2 of D-shell (TxD).

Pin 2 of receptacle assembly connects to pin 7 of D-shell (GND).

Pin 3 of receptacle assembly connects to pin 3 of D-shell (RxD).

Serial Debug Port Interrupts

Two fixed-value interrupts are assigned to the serial debug port. An interrupt is generated when a receiver byte has been accumulated and is available to be read. This interrupt will remain pending as long as there is more data available to be read. Also, an interrupt is generated when the transmitter FIFO is empty. The status of the interrupt for the transmitter or the receiver can be read in the registers in the “Transmitter Buffer Port (TxBUF)” and the “Receiver Buffer Port (RxBUF)”. The interrupts are cleared when these status registers are read.

Table 11-1. Debug Port Interrupt Vector Assignment

Vector No.	Function
209 (D1 hex)	Serial debug port–receive
208 (D0 hex)	Serial debug port–transmit

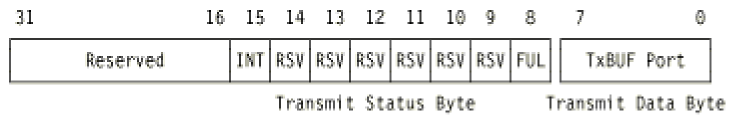
These interrupts can be enabled or disabled as described in *Port Configuration Register (PCR)* on page 176.

Transmitter Buffer Port (TxBUF)

Data that is to be sent out the serial port is loaded in the Transmitter Buffer Port. This port is actually a two-deep FIFO + shift register. When the transmitter is empty, a write to this port sends data immediately to the shift register where the shifting operation begins. Bit 8 of this port can then be read to see if the transmit FIFO is full. If not, another write can be performed to FIFO another byte of data. When Bit 8 is set, it means the FIFO is full. An interrupt will be generated to the 80960 processor when the FIFO again becomes empty (the shift register still has data to shift out at this time).

Register Format

(ADDRESS = 1FFB 8000h) read/write - byte addressable



Bit Descriptions

- Bits 31–16: Reserved. These bits must be set to 0.
- Bit 15: Interrupt Active. (read-only) When set, this bit indicates that the transmitter currently is interrupting the processor. When this bit is read as active, the interrupt to the processor is cleared.
- Bits 14–9: Reserved.
- Bit 8: Transmit FIFO Buffer Full. (read-only)

  When this bit is reset, the FIFO has at least one available byte for data to be written. When set, all FIFO bytes are occupied with data. Further writes to the transmit buffer will be ignored until one or more bytes are transmitted.
- Bits 7–0: Transmitter Buffer Data Port. When read, this byte returns the last byte written to the Transmit FIFO.

Reset Conditions

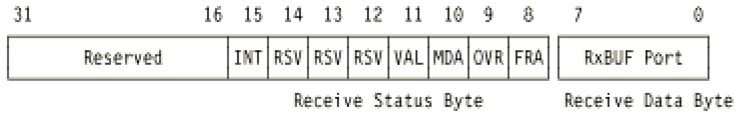
```
        Reset: 0000 0000 0000 0000 0000 0000 0000 0000
ROM Initialization: 0000 0000 0000 0000 0000 0000 0000 0000
```

Receiver Buffer Port (RxBUF)

RxBUF is the port where data can be read that has been received. This port is actually a two-deep FIFO + shift register. Data that comes in from the serial link is accumulated in the shift register and then sent to the first byte of the FIFO. At this point, an interrupt is generated to the 80960 processor. If there is no available position in the FIFO for the accumulated byte to be moved, bit 9 of the RxBUF is set. This is equivalent to an overrun condition.

Register Format

(ADDRESS = 1FFB 8004h) read-only; byte-addressable



Bit Descriptions

- Bits 31–16: Reserved. These bits must be set to 0.
- Bit 15: Interrupt Active. When set, this bit indicates that the receiver is currently interrupting the processor. This bit is cleared when there are no bytes remaining in the FIFO to be read.
- Bits 14–12: Reserved.
- Bit 11: Data Valid. When set, indicates that data in the Receive Data Register is valid (has been received, but has not been read yet.)
- Bit 10: More Data Available. When set, indicates that more data is in the Receiver FIFO.
- Bit 9: Data Overflow. When set, indicates that data was received after the byte in the Receive Data Register, but was lost due to an overflow condition.
- Bit 8: Framing Error. When set, indicates that a framing error occurred while the data byte currently in the Receive Data Register was received.
- Bits 7–0: Receiver Buffer Data Port. Once the data is read, if more data is available in the FIFO (or as more data is collected) the byte in the data port will be replaced with new valid data.

  The status byte will also change when the data byte is read. If the data in the status and data bytes are read by separate byte read operations, the status byte will reflect the status of the byte currently in the data byte.

Reset Conditions

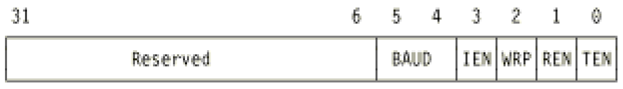
```
Reset: 0000 0000 0000 0000 0000 0000 0000 0000
```

Port Configuration Register (PCR)

The PCR is used to enable operation of the serial debug port and to program its features.

Register Format

(ADDRESS = 1FFB 8008h) read/write



Bit Descriptions

- Bits 31–6: Reserved. These bits must be set to 0.
- Bit 5–4: Baud Rate Select. (Actual rates are approximately 1.5% faster.)

  Baud = 00: 9600 baud

  Baud = 01: 19.2K baud

  Baud = 10: 38.4K baud

  Baud = 11: Reserved.
- Bit 3: Interrupt Enable. (1 = enabled, 0 = disabled) When enabled, an interrupt will be generated on the next occurrence of either of the following conditions:
  - When a byte is collected by the receiver and is ready to be read from the Receiver FIFO buffer
  - When the Transmitter FIFO buffer is empty.
- Bit 2: Wrap Mode Enable. (1 = enabled, 0 = disabled) When wrap mode is enabled, the external chip Rx input and Tx output are disabled, and the output of the transmitter is connected to the input of the receiver.
- Bit 1: Receiver Enable. (1 = enabled, 0 = disabled) When the receiver is enabled, it monitors the Rx input of the Memory Controller Chip (or the transmitter output, if in wrap mode) and collects data that then is stored in the Receiver FIFO Buffer.

  Disabling the receiver aborts any bytes in process and clears bytes left in the FIFO.
- Bit 0: Transmitter Enable. (1 = enabled, 0 = disabled) When the transmitter is enabled, it collects bytes written to the Transmitter FIFO and outputs them on the Tx output of the Memory Controller Chip (or the receiver input, if in wrap mode).

  Disabling the transmitter aborts any bytes in process, clears bytes left in the FIFO, and disables writes to the FIFO.

Reset Conditions

```
                 Reset: 0000 0000 0000 0000 0000 0000 0000 0000
ROM Initialization: 0000 0000 0000 0000 0000 0000 0000 0000
```

ROM Support 12

This chapter describes the co-processor platform's Read-Only Memory (ROM) Version 1.5 on the Micro Channel adapter and Version 2.0.3 on the PCI adapter.

This ROM consists of the Initial Memory Image (IMI) for the 80960Cx microprocessor, Power On Self Test (POST) for the co-processor platform (base card) hardware, initialization, setup, and interface structures for the downloading and execution of AIB ROM POST and supervisor task routines. It also provides for synchronization between the system unit POST, the co-processor platform POST, and the AIB POST.

Refer to Intel User's Manual 270710-001 for 80960CA register and bit descriptions.

Notations and terminology for the Initialization values are as follows:

Value	Description
Hex values	H, h, or 0x
Bit values	Binary
Label or name...	Converted to value by assembler, linker or compiler.
Reserved	Initialized to zeros (thereafter, do not use).
Preserved	Not used by 80960 processor (available for user).
S	Same value as set by prior usage
U	Undefined value
n	Number to be determined by process
IP or (ip)	Instruction pointer
GROUP	Four words to be loaded to 80960 on-chip control registers with SYSCTL instruction, message type 04h.

Main Initialization Structures

When the 80960's RESET pin is deasserted, the processor automatically configures itself with information specified in the Initial Memory Image (IMI).

The co-processor platform ROM provides the IMI for the 80960. The IMI consists of three components:

1. The Initialization Boot Record (IBR)
2. The Process Control Block (PRCB)
3. Secondary Initialization Structures.

These structures are described in the following sections.

Initialization Boot Record (IBR)

The co-processor platform ROM provides the Initialization Boot Record (IBR) on its Programmable Read Only Memory (PROM) at physical address 1FF00h to 1FF30h. When the 80960 reads the IBR for the initialization data at fixed address FFFF FF00h to FFFF FF30h in region 15, it actually reads the data stored in ROM at address 0FFF FF00h to 0FFF FF30h in region 0. This equates to the ROMs' physical address 1FF00h to 1FF30h.

The IBR consists of four components:

- Initial bus configuration data
- First instruction pointer
- PRCB pointer
- Self-test checksum data.

The least-significant byte of the first four words sets the bus configuration for where the ROM IMI is located.

Process Control Block (PRCB)

The PRCB consists of the following components:

- Base pointers for system data structures
- Initial values for Arithmetic_Control, Fault, Instruction, and Register cache configuration words.

The PRCB table is moved into memory to allow for 80960 reinitializing prior to running the Main POST. The reinitialization is required to relocate the interrupt table.

Secondary Initialization Structures

The processor caches the following pointers during its initialization:

- Fault Table
- Control Table
- Interrupt Table.

All of the above tables are located within ROM, with the exception of the interrupt table. The interrupt table is built in RAM after Preliminary POST is complete. For information regarding the interrupt table structure, see the *Base ROM I/F Structures* on page 201.

Fault Table Structure

The fault table structure contains pointers to the fault handling procedures. When the processor detects a fault, it records the fault type and subtype number in a fault record. It then uses the type number to select a fault handling procedure. The fault record is created in either the user or the supervisor stack, depending which stack is active when the fault occurs. If a fault occurs prior to the ROM_TABLE being available for error reporting, the Base ROM places an error code in the Micro Channel Interface Chip's ISP Register, and remains looping forever within the fault handling procedure. If the ROM_TABLE is available to provide ROMStatus error information, then a single error code (PROC_FAULT) is loaded into the ROMStatus word and the processor exits to final adapter initialization.

The fault-procedure-generated error codes to the Micro Channel Interface Chip's ISP Register are as follows:

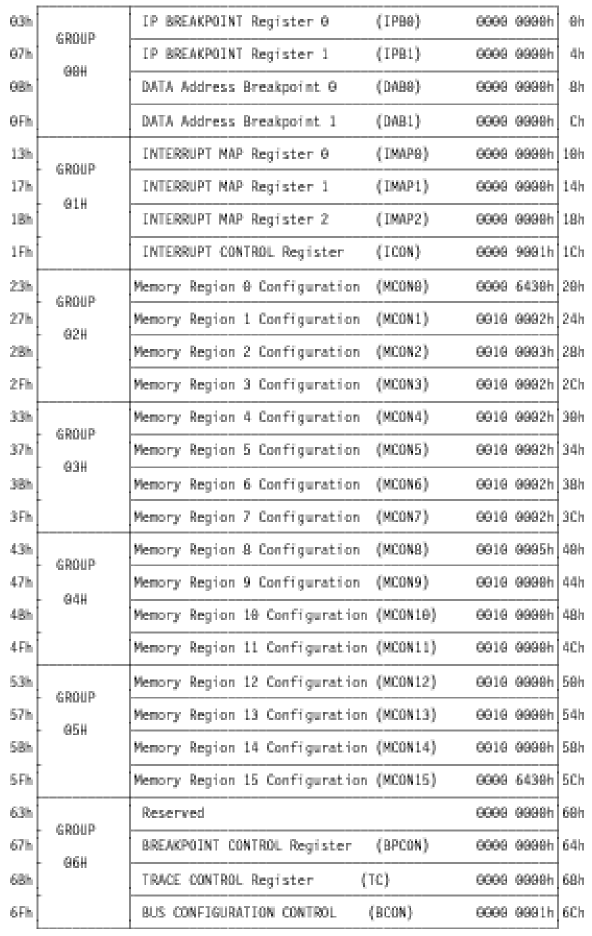
- PAL_FAULT - Parallel fault
- TRA_FAULT - Trace fault
- OPE_FAULT - Operation fault
- ARI_FAULT - Arithmetic fault
- CON_FAULT - Constraint fault
- PRO_FAULT - Protection fault
- TYP_FAULT - Type fault.

See *Error Reporting* on page 189 for more information regarding error codes and reporting.

Control Table Structure

- The control table structure contains the values of the on-chip control registers. At initialization, the processor registers and pointers are automatically loaded. The registers and pointers can be reconfigured and then reinitialized by using the SYSCTL instruction.

On all ARTIC960 PCI adpaters and on ARTIC960 adapters at EC level E32319 or later, region 10 will contain the value 0090 0003h.



03h	GROUP 00H	IP BREAKPOINT Register 0	(IPB0)	0000 0000h	0h
07h		IP BREAKPOINT Register 1	(IPB1)	0000 0000h	4h
0Bh		DATA Address Breakpoint 0	(DAB0)	0000 0000h	8h
0Fh		DATA Address Breakpoint 1	(DAB1)	0000 0000h	Ch
13h	GROUP 01H	INTERRUPT MAP Register 0	(IMAP0)	0000 0000h	10h
17h		INTERRUPT MAP Register 1	(IMAP1)	0000 0000h	14h
1Bh		INTERRUPT MAP Register 2	(IMAP2)	0000 0000h	18h
1Fh		INTERRUPT CONTROL Register	(ICON)	0000 9001h	1Ch
23h	GROUP 02H	Memory Region 0 Configuration	(MCON0)	0000 6430h	20h
27h		Memory Region 1 Configuration	(MCON1)	0010 0002h	24h
2Bh		Memory Region 2 Configuration	(MCON2)	0010 0003h	28h
2Fh		Memory Region 3 Configuration	(MCON3)	0010 0002h	2Ch
33h	GROUP 03H	Memory Region 4 Configuration	(MCON4)	0010 0002h	30h
37h		Memory Region 5 Configuration	(MCON5)	0010 0002h	34h
3Bh		Memory Region 6 Configuration	(MCON6)	0010 0002h	38h
3Fh		Memory Region 7 Configuration	(MCON7)	0010 0002h	3Ch
43h	GROUP 04H	Memory Region 8 Configuration	(MCON8)	0010 0005h	40h
47h		Memory Region 9 Configuration	(MCON9)	0010 0000h	44h
4Bh		Memory Region 10 Configuration	(MCON10)	0010 0000h	48h
4Fh		Memory Region 11 Configuration	(MCON11)	0010 0000h	4Ch
53h	GROUP 05H	Memory Region 12 Configuration	(MCON12)	0010 0000h	50h
57h		Memory Region 13 Configuration	(MCON13)	0010 0000h	54h
5Bh		Memory Region 14 Configuration	(MCON14)	0010 0000h	58h
5Fh		Memory Region 15 Configuration	(MCON15)	0000 6430h	5Ch
63h	GROUP 06H	Reserved		0000 0000h	60h
67h		BREAKPOINT CONTROL Register	(BPCON)	0000 0000h	64h
6Bh		TRACE CONTROL Register	(TC)	0000 0000h	68h
6Fh		BUS CONFIGURATION CONTROL	(BCON)	0000 0001h	6Ch

Interrupt Table Structure

This structure is the 80960 interrupt vector table. This table is used by the Base ROM for POST testing; it may also be used by the AIB POST tests for AIB-related interrupt vectors. An AIB must write valid interrupt handler addresses to the table, prior to enabling and using AIB interrupts.

The lowest-priority vector's (Vector No.8) interrupt handler routine pointer is located at offset 024h from the base of the table. Each vector number (8 through 255) has a full 32-bit address for its respective interrupt routine pointer address. These routine pointers are located in successive locations within the interrupt table starting with Vector No.8. The first 24 bytes in the table are used by the processor for posting software interrupts. This area of the table is initialized to zero, and is not used by the base ROM code.

This table is copied into memory (pointed by VecTabPtr in ROM_TABLE) after Preliminary POST and interrupts are enabled using a special function register operation. All AIB interrupt vectors are initially set to point to the default interrupt service routine. Table 6-1 lists the interrupt vectors associated with the co-processor platform and the AIB.

Some vectors are reserved by the 80960 processor. The pointer addresses for these reserved vectors are set to 0h. For more information on interrupts, consult the 80960CA User's Manual.

80960Cx Reset Conditions

tc	=	initial image ib prbc	==
ac	=	initial image ib prbc	–
pc	=	C01F2002H	==
fp(g15)	=	interr stack ptr	==
pfp(r0)	=	undefined	ssss
sp (r1)	=	interr stack base+64	==
rip(r2)	=	undefined	==
ipnd(sf0)	=	undefined	==
imsk (sf1)	=	00h	==
dmac(sf2)	=	00h	==

After a warm reset, the registers marked by == will contain the cold start value, and the registers marked by ssss will contain the value that was in the register prior to the warm start.

Adapter Initialization and POST

The co-processor platform POST consists of three main routines: Preliminary POST, Main POST, and Final Adapter Initialization.

- Preliminary POST

  During Preliminary POST, the 80960 performs basic integrity tests of the ROM, the Memory Controller Chip and the System Bus Interface Chip registers, and packet memory. If an error occurs during one of these tests: (1) a fatal error code is placed into the ISP Register of the System Bus Interface Chip, (2) the Card ID is written (on ARTIC960 adapters only), and (3) POST exits to a wait loop. Along with the ISP error codes, LED flashing sequences can occur to indicate extreme fatal errors. If the Preliminary POST tests pass, the Base ROM data structures are built in the packet memory.

  For more information regarding error reporting and LED flashing sequences, see *Error Reporting* on page 189.

  After basic integrity tests are performed, *AIB Scan* occurs. During AIB Scan, the 80960 looks for the presence of the AIB. AIB ROM scan is performed and the AIB ROM data structures are moved to packet memory. If an error occurs during AIB Scan, scan testing terminates and the error condition is placed in storage. On completion of Base POST testing, if no base errors have occurred, the ROMStatus and AIBStatus words are set to the appropriate error codes for the scan error previously stored, and final adapter initialization begins.

  Following AIB Scan, the processor is reinitialized using the SYSCTL instruction. Reinitializing is required to relocate the interrupt table from ROM to RAM.

  After reinitializing the processor, the Fixed Stack Procedure is called. The 80960 is moved off the interrupt stack and onto the supervisor stack.

- Main POST

  Main POST performs basic testing of the base card components. If an error occurs during testing, an error code is loaded in the Base ROM Error Status Word and POST exits to a wait loop. Main POST also calls AIB POST and INIT routines.

- Final Adapter Initialization

  During final adapter initialization, the System Bus Interface Chip and the Memory Controller Chip registers are set to the final values, and the error status words are examined to determine if the LED and ROM Busy Bit should be turned off.

Preliminary POST

The Preliminary POST sequence is as follows:

1. 80960 Self Test

   Test the 80960 internal registers and local buses. This test is conducted by the 80960 processor after reset. If a failure occurs, the processor FAIL pin is activated. No recovery is possible.

2. Turn LED on

   Turn LED on by setting EDR Register Bit0 = 1.

3. Base ROM Checksum Test

   Read and sum the 128KB (modulo 100h).

   If the sum is zero, the test passes.

   If the sum is not zero, loop here forever with LED flash sequence 1.

   The checksum test uses a sum field byte in ROM to force the checksum to zero. The value of the sum field is set during programming of the ROM chip. If the ROM is to be flash reprogrammed at a later date, this sum field must be updated to allow proper checksum verification to occur. A second field used for ROM flash programming also is provided. This 1-byte field, when set to x0FFh, indicates that the base ROM checksum test should not be performed. When this field is set to any other value, the base ROM checksum test will be performed.

4. Processor Bus Test

   Write/Read Memory Controller Chip registers

   If no error occurs, the test passes.

   If an error occurs, loop here forever with LED flash sequence 2.

5. Local Bus Test

   Write/Read

   If no error occurs, the test passes.

   If an error occurs, the Base ROM remains looping here forever with LED flash sequence 3.

6. Calculate Packet RAM Memory Size.

   ECC is enabled during this testing.

   - Find the size of packet memory by writing into 32, 16, 8, 4, 2, 1 megabyte boundaries offset from the base address of packet dynamic random access memory (DRAM). These locations are read and verified in the same order to determine the value to write to the MCR Register. The Refresh rate is set to normal (0). The timer speed is set to 25MHz (1).
   - Save the obtained packet memory size in the MCR Register.

     If memory is detected, the test passes.

     If no memory is detected:

     - Set System Bus Interface Chip ISP Register = (NOMEM_FAIL) no packet memory error
     - Write 0x 1FFB 0000h to the Micro Channel Interface Chip XPOS Register
     - Set CBSP Register = interrupt valid bit
     - Write CRDID Register in the Micro Channel Interface Chip = (0x8FECh) default base card ID
     - Loop here forever with LED flash sequence 5.

7. Calculate Instruction RAM Memory Size.

   ECC is enabled during this testing.

   - Find the size of instruction memory by writing into 32, 16, 8, 4, 2, 1 megabyte boundaries offset from the base address of instruction (ZIP) memory. These locations are read and verified in the same order to determine the value to write to the MCR Register.
   - Save the obtained Instruction memory size in the MCR Register.

8. Test First (1MB) Packet RAM

   - Test first 1MB of packet memory using quad word and address in address test.

     If no memory error is detected, initialize 1MB packet memory to zero.

     If memory error is detected:

     - Set
     - Write 0x 1FFB 0000h to the Micro Channel Interface Chip XPOS Register
     - Set CBSP Register = interrupt valid bit.
     - Write CRDID Register in Micro Channel Interface Chip =(0x8FECh) default base card ID
     - Loop here forever with LED flash sequence 6.

9. Build ROM Structures

- Build Base ROM Anchor structure and set pointer values.
- Build ROM_TABLE.
- Build User I/F Structures.
- Set the AIBStatus and ROMStatus words error codes fields to 0xFFFFh.
- If the SSID read from the SSID register does not match what is expected, loop here forever with the LED flash sequence 8 (ARTIC960 PCI only).

10. Base VPD
    - Move the Base VPD data from the Base VPD ROM to the packet memory location 0x 2000 0000h.

11. AIB Scan Testing
    - Test for AIB presence. If AIB is ***not*** present, set ROMStatus and AIBStatus words = 0x0000h and exit AIB scan testing.

      If AIB is present (EDR Register Bit 0 = 1):

      a. Test AIB ROM signature for (0x AA55 AA55h).

         If signature is not correct, store the condition for AIB signature failure and exit scan testing..

         After Base POST testing, if no Base POST test errors have occurred, the AIB_Status and ROMStatus words will be set to AIB_SIGN_FAIL and AIB_ERROR, respectively.

      b. Read AIB ID for non-zero value.

         If AIB ID = 0, Store the condition for AIB ID failure and exit scan testing.

         After Base POST testing, if no Base POST test errors have occurred, the AIB_Status and ROMStatus words will be set to AIB_ID_FAIL and AIB_ERROR, respectively.

      c. Copy AIB ID into the AIBID field of the ROM_TABLE.

      d. Copy AIB RomVersion into the AIB_RomVersion field of the ROM_TABLE.

         Run AIB ROM Checksum Test.

         If the ChkSum field in the AIB ROM Anchor (value used to force the AIB checksum to equal 0) is 0xFF, then the AIB ROM checksum test is bypassed.

         - Set AIB ROM start address = 0x 0C00 0000h.
         - Set AIB ROM end address = base + AIB ROM Length field.
         - Read and sum the ROM, based on the Length field of the AIB ROM Anchor.

If the checksum is not = 0, store the condition for AIB ROM checksum failure and exit scan testing.

After Base POST testing, if no Base POST test errors have occurred, the AIB_Status and ROMStatus words will be set to AIB_ROM_FAIL and AIB_ERROR, respectively.

e. Copy the AIB VPD (if VPD_Ptr not 0) into the VPD area of the Base ROM Anchor.

f. Copy the System x86 ROM Code (if SysROMCode not 0) into the x86 Code area of the Base ROM Anchor.

   If the SysROMLength is greater than the specified 6KB, AIB Scan Testing is exited, NO code will be copied to the system unit executable code window, and an error condition is stored for the SysROMLength failure.

After Base POST testing, if no Base POST test errors have occurred, the AIB_Status and ROMStatus words will be set to AIB_SysCode_FAIL and AIB_ERROR, respectively.

g. Check for an AIB-supplied adapter POS ID. If the POS_ID field of the AIB ROM Anchor is not = 0x FFFF FFFFh then copy the POS_ID value into the Base ROM Anchor structure. (If POS_ID = 0x FFFF FFFFh, then adapter ID = 0x 0000 8FECh)

h. Set AIB_Status = 0x0000h and ROMStatus = 0x0000h; then exit AIB Scan Testing. All AIB tests are OK.

12. Set LBBAR and XPOS Registers
    - Set LBBAR Registers = Base packet memory (0x 2000 0000h).
    - Set XPOS Registers = Base packet memory (0x 2000 0000h)

13. Initialize Processor Configuration Register (PROC_CFG)

    Read MCR Register.

    - If instruction memory is present:

      a. Set PROC_CFG shared memory window = 64MB (full window).

      b. Set PROC_CFG enable ROM window bit.

    - If Instruction memory is not present:

      c. Set PROC_CFG shared memory window = packet memory size (read from MCR).

      d. Set PROC_CFG enable ROM window bit.

14. Card ID (CRDID) Register

    Write the base card ID (from adapter ID field in base ROM Anchor structure) to the System Bus Interface Chip CRDID Register.

Main POST

The Main POST procedure performs Base POST routines, performs tests for AIB POST and INIT routines, calls AIB POST, and calls AIB INIT routines. Each of these called routines performs specific testing or initialization, and can return error codes to the AIBStatus or ROMStatus words. If an error occurs during testing, the Main POST procedure exits to the final adapter initialization sequence. For more information regarding error reporting, see *Error Reporting* on page 189.

After execution of the POST and INIT routines, Main POST performs final adapter initialization. Tasks performed during final adapter initialization are based on the results of prior testing and initialization.

POST Description

1. Initialization Routine

   Initialize ROMStatus = 0xFFFFh;
   Initialize FEER = 0x0000h;
   Initialize (ECC error) SBCCR = 0x00FEh.

2. POST Test Routine

   The following tests are executed in sequence by Test Number.

   A. MemTest. – Read/write and address in address test for packet (> 1MB) and instruction memory.

   B. BRegTest. – Memory Controller Chip register test.

   C. MRegTest. – System Bus Interface Chip register test.

   D. BrTimerFns. – Memory Controller Chip timer tests.

   E. SingleBitEcc. – Single-bit ECC testing for packet (> 1MB) and instruction memory.

   F. MultiBitEcc. – Multi-bit ECC testing for packet (> 1MB) and instruction memory.

   G. LocalBusParity. – Local Bus parity verification test.

   H. LocalBusException. – Local Bus force parity error and Exception test.

   I. TskProtection. – Memory protection testing for packet, instruction, and I/O.

   J. DebugPortFns. – Memory Controller Chip Debug Port testing.

   11. DebugTxferInth – Memory Controller Chip Debug Port testing.

   12. DmaMainTest. – 80960 DMA testing.

   If an error occurs, the program stops testing on the first occurrence of an error and reports the error in the ROMStatus word of the ROM table.

   If no error occurs, the ROMStatus error code = 0x0000h.

AIB POST and INIT

The AIB POST and INIT routines are only executed if no AIB Scan errors occurred during Preliminary POST. If an AIB Scan error has occurred, the ROMStatus and AIBStatus words are set to the appropriate error codes, AIB POST and INIT are bypassed, and the Base ROM begins final adapter initialization.

Prior to executing AIB ROM POST code routines, the Base ROM performs the following AIB ROM code tests.

1. AIB ROM Processor Binding Block Processor ID Test

   The Base ROM scans the binding block headers to look for a match with the correct uP_ID value. The Base ROM, having previously read Register **g0** on Power-Up/Reset knows on which processor type it is executing (that is, 80960CA, 80960CF, and so forth.). If no match is found, an error is logged in the ROMStatus and AIBStatus1 fields to indicate an AIB uP_ID failure.

2. AIB ROM Processor Binding Block Relocation Test

   Once finding the correct processor binding block, the Base ROM reads the ***Location*** field to determine if the AIB POST and INIT code routines are fixed or relocatable. If the Location field is = 0, the AIB ROM code is considered to be position independent. If the Location field is set to a value, the value must be set to the base address of the AIB ROM (0x 0C00 0000h). If the Location field is not 0 and does not equal the correct base address, an error is logged in the ROMStatus and AIBStatus fields to indicate an AIB REL failure.

3. AIB ROM Processor Binding Block POST Presence Test

   The Base ROM reads the ***POST*** field to determine if AIB POST code exists. If POST = 0, then no POST code exists; skip to INIT presence test. If POST exists (POST > 0), the value of POST is used as an offset to the actual POST code based on the previously read Location field.

4. AIB ROM POST Execution (if POST is present)

   When calling the AIB POST routine, the Base ROM places the address of the base ROM Anchor in 80960 Register **g1**.

   POST routines must preserve the contents of all registers except the local registers and Register **g0**, which is used for error status. The AIB POST routine must pass back the error status of its testing in this register. The error status is the same format as the AIBStatus word shown in the *Error Reporting* on page 189.

   Upon return from the AIB POST routine, the Base ROM will examine the returned value and place it in the AIB ROM status word. If any defined bit other than the OEM Bit is set, the AIB Bit in the Base ROM status word should be set. If any one of the RSV Bits or the Base Bit is set, the Base ROM should set the Base and Multi Bits in the AIB ROM status word, along with the FRU for the AIB and an Error Code, indicating an invalid AIB POST status.

5. AIB ROM Processor Binding Block INIT Presence Test

   The Base ROM reads the ***Init*** field to determine if AIB INIT code exists. If Init = 0, no INIT code exists; exit AIB testing and initialization. If INIT exists (Init > 0), the value of Init is used as an offset to the actual INIT code based on the previously read Location field.

6. AIB ROM INIT Execution (if Init is present)

When calling the AIB INIT routine, the Base ROM places the address of the base ROM Anchor in 80960 Register **g1**.

INIT routines must preserve the contents of all registers, except the local registers and Register **g0**. (Register **g0** is used for error status.) The AIB INIT routine must pass back the error status of its testing in this register. The error status is the same format as the AIBStatus word shown in *Error Reporting* on page 189. Generally, no errors are reported by the INIT routines.

Final Adapter Initialization

On entering final adapter initialization, the following initialization sequence occurs. Two separate paths are taken depending on the values of the AIBStatus word and ROMStatus word.

1. The hardware resource table is completed for the base card. If the hardware resource table and number of AIB resources for the AIB are available in AIB ROM, the AIB hardware resource structures are appended to the end of the base card structures and the **NumHWRes** field in the Base ROM_TABLE is updated to reflect the total number of resource structures in the array (AIB and Base). For more information regarding the structure format for hardware resources, see *Struct RIC_RDTEntry (pointed by *HWResTabPtr in ROM_TABLE)* on page 207.
2. (For ARTIC960 adapters only) The Channel Ready timer is disabled in the Micro Channel Interface Chip (PROC_CFG Bit 12 set to 1).
3. (For ARTIC960 adapters only) Multiplexed Streaming is disabled in the Micro Channel Interface Chip (PROC_CFG Bit 11 set to 0).

   Multiplexed Streaming will only be enabled for Micro Channel Interface Chip Pass 3 and higher. For Micro Channel Interface Chip Passes 1 and 2, it is the responsibility of the user to enable multiplexed streaming and adhere to the required address alignment constraints.

4. (For ARTIC960 adapters only) Performance Timer is disabled in the Micro Channel Interface Chip (PROC_CFG Bit 13 set to 1).

   The Performance Timer will only be enabled for Micro Channel Interface Chip Pass 3 and higher.

5. Interrupts are enabled in the System Bus Interface Chip (PROC_CFG Bit 10).
6. NO Errors Occurred

   The AIBStatus and ROMStatus words (Bits 0 to 29) are = 0.

   - The BaseOptions structure is examined to determine if parity checking should be enabled for the System Bus Interface Chip and the Memory Controller Chip (PROC_CFG Bit 9, LBCFG Bit 0, respectively).

     If no AIB exists, the LBUS_Parity field of the BaseOptions structure is set and parity checking will be enabled in the
   - Memory Controller Chip ECC is enabled if not presently enabled.
   - Busy bit in System Bus Interface Chip is reset (CBSP bit 0).

   - Set PostFlag = 0
   - The LED is turned Off.

7. Errors Occurred

   The AIBStatus or ROMStatus word (Bits 0 to 29) is > 0.

   - The BaseOptions structure is examined to determine if parity checking should be enabled for the System Bus Interface Chip and the Memory Controller Chip (PROC_CFG Bit 9 and LBCFG Bit 0, respectively).

     If no AIB exists, the LBUS_Parity field of the BaseOptions structure is set and parity checking will be enabled in the System Bus Interface Chip and the Memory Controller Chip..

     If the error is a parity-related error (0xC0h to 0xC5h), parity checking will not be enabled.

   - Memory Controller Chip ECC is enabled, if not presently enabled.

     If the error is ECC-related (0x81h, 0xB1h to 0xBCh), ECC will not be enabled.

   - The Busy Bit is reset in the System Bus Interface Chip (CBSP and PostFlag bits 0) only if both NLOAD bits in the AIBStatus and ROMSTatus words are = 0. Otherwise, the busy bit remains on, indicating an error has occurred that prohibits loading and operating the Kernel.
   - The LED remains On.

8. The Base ROM code loops waiting for a system unit command. See *Kernel Structures Used by Base ROM (Boot Strap Loader)* on page 214. If an error or unexpected interrupt occurs during the processing of a system unit command, the Base ROM will notify the system unit by loading the exception window with the Invalid_Intr structure. Note that the Base ROM will not generate an interrupt to the system unit when this occurs. It is expected that the system device driver will check the exception window area prior to issuing new commands.

   See *Struct RIC_Except (pointed by ExceptionPtr in ROM_TABLE)* on page 206 for more information regarding the invalid interrupt structure and exception window.

Error Reporting

Errors are classified into three types:

1. Extreme Fatal Errors.

   Extreme fatal errors occur during the very beginning of Preliminary POST. These errors are reported in the ISP Register of the System Bus Interface Chip. It should be noted that some errors are so catastrophic, reporting through the ISP Register is not possible. For diagnostic purposes, specific LED flashing sequences have been defined that correspond to specific extreme fatal error codes. See the following *ISP Error Codes* on page 190 for details of this error-reporting mechanism.

2. Fatal Errors.

Fatal Errors are reported in the ROMStatus and AIBStatus words. Fatal errors generally prevent the adapter from accepting the boot strap load command. See *ROMStatus Word Definitions* on page 191 for details of this error-reporting mechanism. See *Kernel Structures Used by Base ROM (Boot Strap Loader)* on page 214 for more information regarding boot strap loading.

3. General Errors.

   General Errors are reported in the ROMStatus and AIBStatus words. See *ROMStatus Word Definitions* on page 191 for details of this error-reporting mechanism.

ISP Error Codes

There are specific extreme fatal errors that can occur during Preliminary POST, as well as processor faults that cause error codes to be written to the Micro Channel Interface Chip ISP Register. When an error code is written to this register, the IV Bit in the Micro Channel Interface Chip CBSP Register is set to validate the error, and the card ID is written to the Micro Channel Interface Chip CRDID Register to allow the system to view the ISP Register. Table 12-1 describes the ISP error codes and their sources.

Table 12-1. ISP Fatal Error Codes

Error Code	Error Name	Error Source and Description
0x10	Reserved	
0x20	NOMEM_FAIL	No packet memory was detected during the packet memory boundary presence test
0x21	P1MEG_FAIL	Lower 1MB of packet memory failed memory test
0X30	PAL_FAULT	Parallel fault
0X31	TRA_FAULT	Trace fault
0X32	OPE_FAULT	Operation fault
0X33	ARI_FAULT	Arithmetic fault
0X34	CON_FAULT	Constraint fault
0X35	PRO_FAULT	Protection fault
0X36	TYP_FAULT	Type fault

In addition to error codes in the ISP Register, LED flashing sequences are provided for some very basic diagnostic purposes. The LED flashing sequence may be particularly useful for errors that occur very early on in Preliminary POST, thus preventing notification via the ISP Register. The following table describes the LED flashing sequences and the source of the errors. The **No. of Flashes** column defines a period of consecutive visible flashes followed by a pause. This pattern is repeated indefinitely.

Table 12-2. LED Flashing Sequences

Error Name	Error Source and Description	No. of Flashes
BaseCheckSum	Base ROM checksum test failed	1
ProcessorBus	Processor bus test failed	2
LocalBus	Local Bus test failed	3
Reserved		4
NOMEM_FAIL	Packet memory was not detected during the boundary test	5

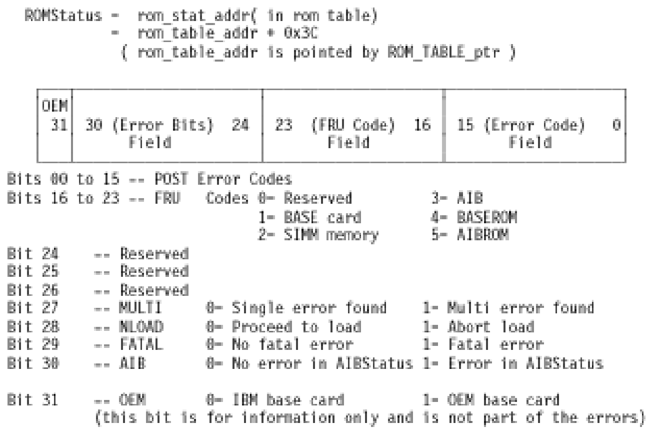
Table 12-2. LED Flashing Sequences

Error Name	Error Source and Description	No. of Flashes
P1MEG_FAIL	Lower 1MB of packet memory failed memory test	6
ProcessorFault	Processor fault occurred	7
SSID Mismatch	The SSID is not as expected	8

ROMStatus Word Definitions

The ROMStatus Word has the following types of error fields:

- Error Code Field
- FRU Error Field
- Error Bits Field



```
ROMStatus =  rom_stat_addr( in rom table)
          =  rom_table_addr + 0x3C
           ( rom_table_addr is pointed by ROM_TABLE_ptr )

 +----+------------------------+---------------------+----------------------+
 |OEM |                        |                     |                      |
 |  31| 30 (Error Bits)  24    | 23  (FRU Code)  16  | 15 (Error Code)    0 |
 |    |        Field           |       Field         |       Field          |
 +----+------------------------+---------------------+----------------------+
Bits 00 to 15 -- POST Error Codes
Bits 16 to 23 -- FRU   Codes 0- Reserved        3- AIB
                             1- BASE card       4- BASEROM
                             2- SIMM memory     5- AIBROM
Bit 24     -- Reserved
Bit 25     -- Reserved
Bit 26     -- Reserved
Bit 27     -- MULTI     0- Single error found    1- Multi error found
Bit 28     -- NLOAD     0- Proceed to load       1- Abort load
Bit 29     -- FATAL     0- No fatal error        1- Fatal error
Bit 30     -- AIB       0- No error in AIBStatus 1- Error in AIBStatus

Bit 31     -- OEM       0- IBM base card         1- OEM base card
           (this bit is for information only and is not part of the errors)
```

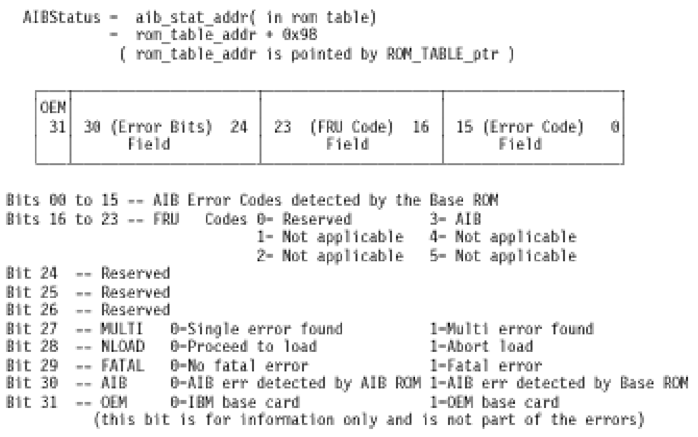
.

Bit 30 = 1 in ROMStatus indicates that an AIB error was detected by the Base ROM and an AIB error code was placed in the AIBStatus.

AIBStatus Word (under Base ROM Control)

The AIBStatus Word has the following types of error fields:

- Error Code Field
- FRU Error Field
- Error Bits Field



```
AIBStatus =  aib_stat_addr( in rom table)
          =  rom_table_addr + 0x98
           ( rom_table_addr is pointed by ROM_TABLE_ptr )

 +---+----------------------+--------------------+------------------+
 |OEM|                      |                    |                  |
 | 31| 30 (Error Bits)   24 | 23  (FRU Code)  16 | 15 (Error Code) 0|
 |   |       Field          |       Field        |      Field       |
 +---+----------------------+--------------------+------------------+

Bits 00 to 15 -- AIB Error Codes detected by the Base ROM
Bits 16 to 23 -- FRU   Codes 0- Reserved        3- AIB
                             1- Not applicable  4- Not applicable
                             2- Not applicable  5- Not applicable
Bit 24  -- Reserved
Bit 25  -- Reserved
Bit 26  -- Reserved
Bit 27  -- MULTI  0-Single error found          1-Multi error found
Bit 28  -- NLOAD  0-Proceed to load             1-Abort load
Bit 29  -- FATAL  0-No fatal error              1-Fatal error
Bit 30  -- AIB    0-AIB err detected by AIB ROM 1-AIB err detected by Base ROM
Bit 31  -- OEM    0-IBM base card               1-OEM base card
          (this bit is for information only and is not part of the errors)
```

Bit 30 = 1 in AIBStatus indicates that an AIB error was detected by the Base ROM (not the AIB ROM).

Error Code Breakdown

The following tables describe the ROMStatus and AIBStatus words:

1. Table 12-3 – ROMStatus Word (Base Card POST Errors)
2. Table 12-4 – ROMStatus Word (AIB Errors found by Base ROM)
3. Table 12-5 – AIBStatus Word

Table 12-3. ROMStatus Word (Base Card POST Errors)

Test No.	Test Name	Error Name	Error Description	OEM Bit 31	AIB Bit 30	FATAL Bit 29	NLOAD Bit 28	MULTI Bit 27	FRU Bits 23-16	Error Code Bits 15-0
–	SUCCESS	Success	All tests pass	IBM	no_AIB error	non_fatal	load	–	–	0000
–	FOXES	NoRun	POST did not execute	IBM	–	fatal	no load	multi	BASE	FFFF
–	–	PROC_FAULT	Processor fault occurred	–	–	fatal	no load	–	–	FFFE
–	–	INVALID_INT	Invalid interrupt detected in IntID structure for current test.	IBM	–	fatal	no load	–	BASE	FFFD
1	MemTestRW	PMemTestRWS_0FAIL	Packet mem byte r/w error	IBM	no_AIB error	fatal	no load	single	SIMM	0050
1	MemTestRW	PMemTestRW_1FAIL	Packet mem short r/w error	IBM	no_AIB error	fatal	no load	single	SIMM	0051
1	MemTestRW	PmemTestRW_2FAIL	Packet mem word r/w error	IBM	no_AIB error	fatal	no load	single	SIMM	0052
1	MemTestRW	PmemTestRW_3FAIL	Packet mem double-word r/w error	IBM	no_AIB error	fatal	no load	single	SIMM	0053
1	MemTestRW	PmemTestRW_4FAIL	Packet mem triple-word r/w error	IBM	no_AIB error	fatal	no load	single	SIMM	0054
1	MemTestRW	PmemTestRW_5FAIL	Packet mem quad-word r/w error	IBM	no_AIB error	fatal	no load	single	SIMM	0055
1	MemTestRW	PmemTestRW_6FAIL	Packet mem 4-byte word alignment error	IBM	no_AIB error	fatal	no load	single	SIMM	0056

Table 12-3. ROMStatus Word (Base Card POST Errors)

Test No.	Test Name	Error Name	Error Description	OEM Bit 31	AIB Bit 30	FATAL Bit 29	NLOAD Bit 28	MULTI Bit 27	FRU Bits 23-16	Error Code Bits 15-0
3	MemTestAA	PMemTestAA_Fail	Packet mem address/address test error	IBM	no_AIB error	fatal	no load	single	SIMM	0060
2	MemTestRW	IMemTestRW_0FAIL	Instruction mem byte r/w error	IBM	no_AIB error	fatal	no load	single	BASE	0070
2	MemTestRW	IMemTestRW_1FAIL	Instruction mem short r/w error	IBM	no_AIB error	fatal	no load	single	BASE	0071
2	MemTestRW	IMemTestRW_2FAIL	Instruction mem word r/w error	IBM	no_AIB error	fatal	no load	single	BASE	0072
2	MemTestRW	IMemTestRW_3FAIL	Instruction mem double-word r/w error	IBM	no_AIB error	fatal	no load	single	BASE	0073
2	MemTestRW	IMemTestRW_4FAIL	Instruction mem triple-word r/w error	IBM	no_AIB error	fatal	no load	single	BASE	0074
2	MemTestRW	IMemTestRW_5FAIL	Instruction mem quad-word r/w error	IBM	no_AIB error	fatal	no load	single	BASE	0075
2	MemTestRW	IMemTestRW_6FAIL	Instruction mem 4-byte word alignment error	IBM	no_AIB error	fatal	no load	single	BASE	0076
4	MemTestAA	IMemTestAA_FAIL	Instruction mem address/address test error	IBM	no_AIB error	fatal	no load	single	BASE	0080
5	ECCCountTest	SBIT_ECC_COUNT_FAIL	Single-bit ECC count > 10 during memory tests	IBM	no_AIB error	fatal	no load	single	SIMM	0081
6	BRegTest	BRegTest_Fail	Memory Controller Chip register test error	IBM	no_AIB error	fatal	no load	single	BASE	0090

Table 12-3. ROMStatus Word (Base Card POST Errors)

Test No.	Test Name	Error Name	Error Description	OEM Bit 31	AIB Bit 30	FATAL Bit 29	NLOAD Bit 28	MULTI Bit 27	FRU Bits 23-16	Error Code Bits 15-0
7	MRegTest	MRegTest_Fail	System Bus Interface Chip register test error	IBM	no_AIB error	fatal	no load	single	BASE	0091
8	BrTimerFns	Inth0_SERV_TIMEOUT	Memory Controller Chip timer0 inth0 service error (vector 255 active)	ibm	no_AIB error	fatal	no load	single	BASE	00A0
8	BrTimerFns	TIMER0_FAIL	Memory Controller Chip timer0 r/w error	IBM	no_AIB error	fatal	no load	single	BASE	00A1
9	BrTimerFns	Inth10_SERV_TIMEOUT	Memory Controller Chip timer1 inth10 service error (vector 35 active)	IBM	no_AIB error	fatal	no load	single	BASE	002A
9	BrTimerFns	TIMER1_FAIL	Memory Controller Chip timer1 r/w error	IBM	no_AIB error	fatal	no load	single	BASE	00A3
A	BrTimerFns	Inth11_SERV_TIMEOUT	Memory Controller Chip timer2 inth11 service error (vector 34 active)	IBM	no_AIB error	fatal	no load	single	BASE	00A4
A	BrTimerFns	TIMER2_FAIL	Memory Controller Chip timer2 r/w error	IBM	no_AIB error	fatal	no load	single	BASE	00A5

Table 12-3. ROMStatus Word (Base Card POST Errors)

Test No.	Test Name	Error Name	Error Description	OEM Bit 31	AIB Bit 30	FATAL Bit 29	NLOAD Bit 28	MULTI Bit 27	FRU Bits 23-16	Error Code Bits 15-0
B	BrTimerFns	Inth12_SERV_TIMEOUT	Memory Controller Chip timer3 inth12 service error (vector 33 active)	IBM	no_AIB error	fatal	no load	single	BASE	00A6
B	BrTimerFns	TIMER3_FAIL	Memory Controller Chip timer3 r/w error	IBM	no_AIB error	fatal	no load	single	BASE	00A7
C	BrTimerFns	Inth13_SERV_TIMEOUT	Memory Controller Chip timer4 inth13 service error (vector 32 active)	IBM	no_AIB error	fatal	no load	single	base	00a8
C	BrTimerFns	TIMER4_FAIL	Memory Controller Chip timer4 r/w error	IBM	no_AIB error	fatal	no load	single	BASE	00A9
D	SingleBitECC	INT14TIMEOUT_PACKET	Inth14 service error packet mem (vector 8 active)	IBM	no_AIB error	fatal	no load	single	SIMM	00B1
D	SingleBitECC	SBIT_PACKMEM_FAIL	sbit correction packet mem error	IBM	no_AIB error	fatal	no load	single	SIMM	00B2
E	SingleBitECC	INT14TIMEOUT_INST	Inth14 service error Inst mem (vector 8 active)	IBM	no_AIB error	fatal	no load	single	BASE	00B3
E	SingleBitECC	SBIT_INSTMEM_FAIL	sbit correction instr mem error	IBM	no_AIB error	fatal	no load	single	BASE	00B4
F	SingleBitECC	SBIT_DIS_PACKET_FAIL	sbit ECC disable packet mem error	IBM	no_AIB error	fatal	no load	single	SIMM	00B5
10	SingleBitECC	SBIT_DIS_INST_FAIL	sbit ECC disable instr mem error	IBM	no_AIB error	fatal	no load	single	BASE	00B6

Table 12-3. ROMStatus Word (Base Card POST Errors)

Test No.	Test Name	Error Name	Error Description	OEM Bit 31	AIB Bit 30	FATAL Bit 29	NLOAD Bit 28	MULTI Bit 27	FRU Bits 23-16	Error Code Bits 15-0
11	MultiBitECC	Inth1TIMEOUT_PACKET	Intr 1 service error packet mem (vector 254 active)	IBM	no_AIB error	fatal	no load	single	SIMM	00B7
11	MultiBitECC	MBIT_PACKMEM_FAIL	mbit correction packet mem error	IBM	no_AIB error	fatal	no load	single	SIMM	00B8
12	MultiBitECC	INT1TIMEOUT_INST	Inth1 service error inst mem (vector 254 active)	IBM	no_AIB error	fatal	no load	single	BASE	00B9
12	MultiBitECC	MBIT_INSTMEM_FAIL	mbit correction instr mem error	IBM	no_AIB error	fatal	no load	single	BASE	00BA
13	MultiBitECC	MBIT_DIS_PACKET_FAIL	mbit ECC disable packet mem error	IBM	no_AIB error	fatal	no load	single	SIMM	00BB
14	MultiBitECC	MBIT_DIS_INST_FAIL	mbit ECC disable instr mem error	IBM	no_AIB error	fatal	no load	single	BASE	00BC
15	LocalBusParity	PARITY_LBWRDATA_FAIL	Local bus data parity error	IBM	no_AIB error	non_fatal	no load	single	BASE	00C0
15	LocalBusParity	PARITY_LBRDDATA_FAIL	Local bus data parity error	IBM	no_AIB error	non_fatal	no load	single	BASE	00C1
15	LocalBusParity	PARITY_LBADDR_FAIL	Local bus address parity error	IBM	no_AIB error	non_fatal	no load	single	BASE	00C2
16	LocalBusException	PARITY_XLBWRDATA_FAIL	Local bus data parity test error (vector 243 active)	IBM	no_AIB error	non_fatal	no load	single	BASE	00C4
16	LocalBusException	PARITY_XLBADDR_FAIL	Local bus address parity test error	IBM	no_AIB error	non_fatal	no load	single	BASE	00C5
17	TskProtection	inth4_SERV_TIMEOUT	Inth4 service error (vector 235 active)	IBM	no_AIB error	fatal	no load	single	BASE	00D0
17	TskProtection	PROT_PACKMEM_FAIL	Task protection packet mem error	IBM	no_AIB error	fatal	no load	single	BASE	00D1
18	Test removed	—	—	–	–	–	–	–	–	–

Table 12-3. ROMStatus Word (Base Card POST Errors)

Test No.	Test Name	Error Name	Error Description	OEM Bit 31	AIB Bit 30	FATAL Bit 29	NLOAD Bit 28	MULTI Bit 27	FRU Bits 23-16	Error Code Bits 15-0
19	TskProtection	PROT_IOMEM_FAIL	Task protection i⁄o error (vector 235 active)	IBM	no_AIB error	fatal	no load	single	BASE	00D3
1A	DebugPortFns	DbugOverflow_FAIL	Rx overflow error	IBM	no_AIB error	non_fatal	load	single	BASE	00E1
1A	DebugPortFns	DbugRxBuf_FAIL	Rx buffer mda/da bits error	IBM	no_AIB error	non_fatal	load	single	BASE	00E2
1B	DebugTxferInth	DbugPortTxRx_FAIL	Tx-Rx buffer error (vector 208, 209 active)	IBM	no_AIB error	non_fatal	load	single	BASE	00E3
1C	DmaMainTest	DmaTimeout_FAIL	DMA timeout error	IBM	no_AIB error	fatal	load	single	BASE	00F0
1C	DmaMainTest	DmaData_FAIL	DMA data error	IBM	no_AIB error	fatal	load	single	BASE	00F1

Table 12-4. ROMStatus Word (AIB Errors found by Base ROM)

Test No.	Test Name	Error Name	Error Description	OEM Bit 31	AIB Bit 30	FATAL Bit 29	NLOAD Bit 28	MULTI Bit 27	FRU Bits 23-16	Error Code Bits 15-0
–	AIB_signature	AIB_SIGN_FAIL	AIB signature error	0	1	0	0	0	0	0000
–	AIB_id	AIB_ID_FAIL	AIB ID error	0	1	0	0	0	0	0000
–	AIB_pos_id	—	—	–	–	–	–	–	–	–
–	AIB_rom_chksum	AIB_ROM_FAIL	AIB_ROM checksum error	0	1	0	0	0	0	0000
–	AIB_SysROMCode	AIB_SysCode_FAIL	AIB-supplied system ROM code length field error (> 6KB)	0	1	0	0	0	0	0000
–	Aib_uPid	AIB_UPID_FAIL	AIB uP not supported by Base ROM	0	1	0	0	0	0	0000
–	Aib_Location	AIB_LOC_FAIL	AIB ROM address error	0	1	0	0	0	0	0000
–	AibPost, AibInit	AIB_INVALID_CODE	Invalid AIB POST code	0	1	0	0	0	0	0000

AIB = 1 defines ABI error in AIBStatus.

Table 12-5. AIBStatus Word

Test No.	Test Name	Error Name	Error Description	OEM Bit 31	AIB Bit 30	FATAL Bit 29	NLOAD Bit 28	MULTI Bit 27	FRU Bits 23-16	Error Code Bits 15-0
–	AIB_signature	AIB_SIGN_FAIL	AIB signature error	IBM	AIB base error	fatal	load	single	AIB	0210
–	AIB_id	AIB_ID_FAIL	AIB ID error	IBM	AIB base error	fatal	load	single	AIB	0220
–	AIB_pos_id	—	—	–	–	–	–	–	–	–
–	AIB_rom_chksum	AIB_ROM_FAIL	AIB ROM checksum error	IBM	AIB base	fatal	load	single	AIB ROM	0230
–	Aib_uPid	AIB_UPID_FAIL	AIB uP not supported by Base ROM	IBM	AIB base error	fatal	load	single	AIB	0250
–	Aib_Location	AIB_LOC_FAIL	AIB ROM address error	IBM	AIB base error	fatal	load	single	AIB	0260
–	AibPost	AIB_INVALID_CODE	Invalid AIB POST code	IBM	AIB base error	fatal	load	multi	AIB	0270
–	AibInit	AIB_INVALID_CODE	Invalid AIB Init code	IBM	AIB base error	fatal	load	multi	AIB	0270
–	aIBiNIT	AIB_INIT_FAIL	AIB initialization error	IBM	AIB base error	fatal	single	load	AIB	0280

AIB base error defines AIB error detected by Base ROM.

Base ROM I/F Structures

Several of the ROM interface structure tables have fields titled ***Filled in By*** and ***Valid***. The ***Filled in By*** column indicates which code device fills in that particular entry in the table.

- BROM - base card ROM code
- AROM - AIB ROM code
- KERNEL - operating system code
- DEVICE - device driver code

The ***Valid*** column indicates when the value of the entity described in that particular row is valid. The intent of this column is to identify when a table entry is available for use, and also when a table entry must be filled in.

- PRE - after Preliminary POST is complete
- BPOST - after Base POST is complete
- AINIT - after AIB POST and Init is complete
- FINAL - after Final Adapter Initialization
- LOAD - during the load process command execution

Packet Memory Map (RadiSys ARTIC960 Co-Processor Platform ROM Anchor)

The ROM assumes 1MB minimum of packet memory installed.

Table 12-6. ROM Usage of Lower 1MB of Packet Memory

Address	Description	# of Hex Bytes
2000 0001	RadiSys ARTIC960 Co-Processor Platform and AIB VPD	FF
2000 0100	reserved = 0	220
2000 0320	Copyright Information	100
2000 0420	Adapter ID (default = 0000 8FECh) for Micro Channel systems SSID (default = 0043 1014h) for PCI systems	4
2000 0424	Processor ID/Level (refer to 80960 processor data sheet for ID description)	4
2000 0428	Base Memory Size (MCR Value)	4
2000 042C	Post Status Flag (1=Post running, 0=In Wait Loop)	4
2000 0430	Pointer to ROM_TABLE structure--->	4
2000 0434	Pointer to Base ROM Work Area --->	4
2000 0438	Reserved = 0	3C8
2000 0800	Reserved = 0, was x86 Code executed by system unit	6KB
2000 2000	Structure-Rom_table, Mem, Io, Hdwr, Mcat, Debug	4KB
2000 3000	Structures-Cwin, Exwin	4KB
2000 4000	Structure-PRCB, Interrupt Table	1.088KB
2000 4440	Interrupt Stack (set by .Id at compile)	2KB
2000 4C40	Supervisor Stack (set by .Id at compile)	4KB
2000 4C40	Supervisor Stack (set by .Id at compile)	4KB
2000 5C40	ROM Work Area (set by .Id at compile)	–
2000 xxxx	Start Free Mem (set by compiler EBSS. See “MemStartFree in structure MEM)	–

Rom_Table Structure

Table 12-7. ROM Table Structure

Offset	Name	Description	Filled in By	Valid
0000	Table_Size;	Size of ROM Table in bytes	BROM	PRE
0004	PageSize;	Size of a page in bytes as operated on by the memory protection hardware	BROM	PRE
0008	Memregions;	Number of memory regions (that is, Packet, Instruction, and so forth)	BROM	PRE
000C	*MemInfoPtr;	Pointer to array of structures---> MEM	BROM	PRE
0010	IOregions;	Number of non-contiguous IO regions	BROM	PRE
0014	*IOInfoPtr;	Pointer to array of structures---> IO	BROM	PRE
0018	CmdStatus;	Status of the system unit command window (0 = window is free for next cmd)	BROM	–
001C	Timeout;	Timeout value for a command	DEVICE	PRE
0020	*CommandPtr;	Pointer to structure---> SUCommands (system unit commands are posted in this window)	BROM	PRE
0024	*ExceptionPtr;	Pointer to structure---> RIC_Except (exception condition information window)	BROM	PRE
0028	NumHWRes;	Number of hardware resources	BROM	PRE
002C	*HWResTabPtr;	Pointer to array of structures---> RIC_RDTEntry (definition of each allocated hardware item)	BROM	PRE
0030	*KRIB_ptr;	Pointer to structure---> KRIB (kernel information block. This pointer is filled in by the kernel)	KERNEL	LOAD
0034	CardNum;	Logical card number for this adapter	DEVICE	LOAD
0038	NumCards;	total number of cards in system (of this type)	DEVICE	LOAD
003C	ROMStatus;	Base ROM error code field	BROM	PRE / FINAL
0040	SupervisorLoaded;	Successful loading of supervisory task	BROM	LOAD
0044	ProcessorLevel;	80960 step number	BROM	PRE
0048	BrightonVersion;	ARTIC 32- bit Memory Controller Chip ID	BROM	PRE
004C	MiamiVersion;	System Bus Interface Chip ID	BROM	PRE
0050	ActelVersion;	FPGA interrupt controller chip ID	BROM	PRE
0054	BaseCardVersion;	RadiSys ARTIC960 Co-Processor Platform card hardware level	BROM	PRE
0058	ROMVersion;	Base ROM level (byte 3=major, byte 2=minor, byte 1,0=revision)	BROM	PRE
005C	CardType;	(RadiSys ARTIC960 Co-Processor Platform = 0)	BROM	PRE
0060	AIBID;	AIB ID (0 indicates no AIB present)	BROM	PRE
0064	*VPD_ptr;	Pointer to Vital Product Data---> VPD (vital product data)	BROM	PRE
0068	AIB_RomVersion;	AIB ROM level (byte 3=major, byte 2=minor, byte 1,0=revision)	BROM	PRE
006C	Kernel;	Kernel version No.	KERNEL	LOAD

Table 12-7. ROM Table Structure

Offset	Name	Description	Filled in By	Valid
0070	BaseSS;	Base subsystem version	KERNEL	LOAD
0074	MChanSS;	System Bus subsystem version	KERNEL	LOAD
0078	SCBSS;	SCB subsystem version	KERNEL	LOAD
007C	DumpHandler	Address of dump handler in the base ROM	BROM	LOAD
0080	SyncFlag;	Synchronization flag used for dump	KERNEL	-
0084	*InMemImage	Pointer to structure---> IBR (IBR structure used for 80960 init)	BROM	PRE
0088	*MCAT_Ptr;	Pointer to structure---> MCAT (System Bus Address Table)	DEVICE	LOAD
008C	*VecTabPtr;	Pointer to interrupt vector table---> VecTab	BROM	PRE
0090	*DB_area	Pointer to Debug scratch area---> DB Start	BROM	PRE
0094	CacheOffsetMask	Cache address offset mask	BROM	PRE
0098	AIBStatus1;	AIB error code field	BROM	PRE / FINAL
009C	AIBStatus2;	Reserved (=0) AIB error code field	BROM	-
00A0	AIBStatus3;	Reserved (=0) AIB error code field	BROM	-
00A4	AIBStatus4;	Reserved (=0) AIB error code field	BROM	-
00A8	*BaseOptions	Pointer to structure--> Base_Options (configurable options on base)	BROM	PRE
00AC	*RomPostStat	Pointer to structure--> RomPostStatus (Post Error Information block)	BROM	PRE
00B0	*IntlD	Pointer to structure--> (Inerrupt ID structure)	BROM	PRE
00B4	CardTypeFlags[1]	Flags for Cache/Base Card Processor Frequency/Channel Type (default=0x00000009). Bits 31-28---> Base Card Processor Frequency, 0000-25Mhz. Bits 27-04---> Reserved. Bit 03=1---> Data Cache enabled. Bits 02-00=001---> PCI Channel	BROM	PRE

[1] This is a new table entry and will not be included in the length field in Micro Channel releases Version 1.5 and below of the ROM_TABLE structure.

Struct MEM (pointed by MemInfoPtr in ROM_TABLE)

One unique structure ***Struct MEM*** exists for every region defined by the ***Memregions*** entry of the ROM_TABLE. Regions are considered to be packet memory and instruction memory.

Table 12-8. MEM Structure

Offset	Name	Description	Filled in By	Valid
0000	*MemBasePtr;	Pointer to start of memory for this region	BROM	PRE
0004	MemTotal;	Total amount memory for this region in bytes	BROM	PRE
0008	*MemStartFreePtr;	Pointer to start of free memory in the region	BROM	PRE
000C	MemFree;	Total amount of free memory for this region in bytes	BROM	PRE
0010	MemType;	Type of memory in this region (0 = packet, 1 = instructino)	BROM	PRE

Struct IO (pointed by IOInfoPtr in ROM_TABLE)

One unique structure ***Struct IO*** exists for every region defined by the ***IOregions*** entry of the ROM_TABLE. Regions are non-contiguous memory segments dedicated as I/O areas. For the co-processor platform, the memory-mapped I/O regions are as follows:

I/O Region	Size in Hex Bytes	Starting Address
I/O Region System Bus Interface Chip & Memory Controller Chip	20000	1FFA 0000h
Low Base I/O 1 (FPGA)	8	09FF FFF0h
Low Base I/O 2 (FPGA)	10	0A000 000h
AIB (when present)	A0000	1FF0 0000h

Table 12-9. IO Structure

Offset	Name	Description	Filled in By	Valid
0000	*IOBasePtr;	Pointer to start of I/O for this region	BROM	PRE
0004	IOSize;	Total amount I/O for this region in bytes **Note:** BROM will set to 0. AIBROM will set to proper size.	BROM/AIBROM	PRE
0008	IOType;	I/O Type (0 = Base card I/O, 1 = AIB I/O)	BROM	PRE

Struct SUCommands (pointed by CommandPtr in ROM_TABLE)

This structure is a 256-byte window used by the system unit to send commands to the adapter.

Table 12-10. SUCommands Structure

Offset	Name	Description	Filled in By	Valid
0000	256-byte window		—	—

Struct RIC_Except (pointed by ExceptionPtr in ROM_TABLE)

This structure is a 64-byte window used to signal exceptions between the adapter and the system unit.

Table 12-11. RIC_Except Structure

Offset	Name	Description	Filled in By	Valid
0000	64-byte window	(See *RadiSys ARTIC960 Platforms Programmer's Guide and Reference* for more information.)	—	—

Structure Definitions

```
struct RIC_Except
{
           RIC_ULONG            ExceptionCode;
           RIC_ULONG            ExceptionDataSize;
           union
           {
              union
              {
                 struct RIC_AsyncEvent      Event;
                 struct RIC_Invalid_Intr    InvIntr;
                 struct RIC_Data_Corrupt    BadData;
                 struct RIC_KernIni         KernIni;
              } KernelInfo;
              struct RIC_MBXErrInfo    MBXInfo;
              struct RIC_SCBErrInfo    SCBInfo;
              struct RIC_MCErrInfo     MCInfo;
          }  ExceptionData;
         };
 -----------------------------------------------------------------
        struct RIC_Invalid_Intr
        {
          RIC_ULONG  VectorNum;
        };
 -----------------------------------------------------------------
```

System Unit Notification of Error

If an error or unexpected interrupt occurs during the processing of a system unit command, the Base ROM will notify the system unit by loading the exception window with the Invalid_Intr structure. Note that the Base ROM will not generate an interrupt to the system unit when this occurs. It is expected that the system device driver will check the exception window area prior to issuing new commands. The following values are used:

- ExceptionCode = TERMERR_INVALID_INTR = 0x0023.

- VectorNum = vector number of interrupt that occurred..

- The interrupt number used is taken form the Interrupt Record located at address FP (Frame Pointer) - 16 on the stack when the interrupt service routine is active.
- The General Interrupt is the only interrupt that is expected during the processing of a system unit command. All other interrupts are considered to be invalid.

Struct RIC_RDTEntry (pointed by *HWResTabPtr in ROM_TABLE)

The Hardware Resource Table is an array of RIC_RDTEntry structures, each 64 bytes in length, allocated in a 2KB memory space. The number of total hardware resources is defined by the ***NumHWRes*** parameter in the ROM_TABLE. Based on the present 2KB allocation, 32 separate hardware resources can be allocated. At present 12 hardware resources are defined for the base card. This allows for 20 separate resources that can be defined by the AIB. During Final Adapter initialization, the base card copies the hardware resources from AIB ROM (the AIB ROM uses the same structure format), appends the AIB resource structures to the end of the base structures, and updates the NumHWRes field of the base ROM_TABLE. For more information regarding final adapter initialization see *Final Adapter Initialization* on page 188. The following is the list of base card hardware resources and their DeviceName field entries in the RIC_RDTEntry:

- System Bus DMA Channel Number 1 - “MICDMA1”
- System Bus DMA Channel Number 2 - “MICDMA2”
- Timer Number 0 - “WDTIMER”
- Timer Number 1 - “SWTIMER”
- Timer Number 2 - “TODTIMER”
- Timer Number 3 - “PERFTIMER”
- Timer Number 4 - “TSTIMER”
- SCB Port - “SCBPORT”
- Error Correction Information - “ECCINFO”
- Serial Port - “SERIALPORT”
- System Bus Interface Chip Base Address - “MASTER1BASE”
- Memory Controller Chip Base Address - “MASTER2BASE”

Devices on the AIB should be named according to the following format:

```
AIBxxx0PORTnn
```

where:

xxx is the 3-digit hexadecimal AIB unique number.

nn is the port number, starting with 00, associated with the device.

If there is more than one device associated with a port, the 0 preceding the ***PORTnn*** can be incremented.

The PostStatus field of the structure indicates if the hardware resource passed or failed POST testing (0=passed; 1=failed). If no unique POST test exists for the specific hardware resource, the status is set to 0.

Table 12-12. RIC_RDTEntry Structure

Offset	Name	Description	Filled in By	Valid
0000	DeviceName;	16-byte character string that names the hardware resource	BROM	FINAL
0014	PostStatus;	POST testing status (0=passed, 1=failed)	BROM	FINAL
0018	DataSize;	Number of bytes in DeviceData array (set to 0 by BROM)	BROM	FINAL
001C	DeviceData(DataSize);	Array device data of size DeviceSize (Max_Data_Size=36)	BROM	—

Struct VPD (pointed by VPD_ptr in ROM_TABLE)

This 255-byte area containing the Vital Product Data (VPD) structures for both the base adapter and the AIB (if present). The VPD structure used is the same for both the base card and the AIB. Each structure begins with the character string **VPD** and is structure type AIB_ROM_VPD.

```
  #define VPD_ROM_COUNT 1
refid=696
  struct AIB_ROM_VPD
  {
    char           VPD[3];                // must contain "VPD"
    unsigned char  Length;                // VPD structure length
    unsigned char  AdapterType[2].;       // currently undefined for adapters
    char           PartNoOfPart[12].;     // part number of product (FRU) -
                                          // assembly part number
    char           FRUNoOfPart[12].;      // FRU part number
    char           SerialNo[8].;          // serial number
    char           ManufactureID[10].;    // manufacturer ID
    char           ECLevel[12].;          // engineering change number
    char           DeviceDriverLevel[2].; // device driver level (RS/6000)
    char          DiagnosticLevel[2].;    // diagnostic level is the version of the
                             // supplemental diskette in RS/6000; the PCI card does
                             // not have a supplemental diskette and is undefined
    char           LoadID[4].;            // ARTIC960 kernel level XY.ZZ, where
                             // X=version, Y=revision, ZZ=modification level
    struct ALT_ROM_ID
    {
      uchar          IDByte;                  // 0xn3 for an alterable ROM

                                          // 'n' (defined as VPD_ROM_COUNT)
                                             // is the count of ROM modules

                                         // and size of the following array
      struct
      {
       char           PartNoOfFRU[12];   // part # of 1st microcode FRU
       char           ECLevel[12];       // engineering change level
       char           FlagByte;          // flag byte
       char           FixID;             // fix ID
      }  ROMPart[VPD_ROM_COUNT];          // array of ROM VPD descriptors
    }       ROM_ID;
    char           MicrocodeLevel[2];    // minimum loadable ROM level
                                         // for functional operation
    char           DisplayMsg[30];       // AIB descriptive message
  };
```

Figure 12-1. AIB ROM VPD Structure

The co-processor platform Base ROM VPD values for the VPD fields are flash-programmed into the ROM at the time of manufacture.

Struct MCAT (pointed by MCAT_Ptr in ROM_TABLE)

This structure describes the System Bus window for each unit in the system. There is one structure for each unit, indexed by the logical unit number.

Table 12-13. MCAT Structure

Offset	Name	Description	Filled in By	Valid
0000	MCMemAddr;	Address of the unit's memory window	DEVICE	LOAD
0004	MCMemSize;	Size of the unit's memory window	DEVICE	LOAD
0008	MCIOAddr;	Address of the unit's I/O window	DEVICE	LOAD
000C	MCIOMapAddr;[2]	Address of the unit's mapped I/O window	DEVICE	LOAD

[2] This is a new field and is valid for PCI systems only. It is not part of the Micro Channel structure.

Struct Debug (pointed by DB_Area in ROM_TABLE)

This area is used as a communication window by the system debugger.

Table 12-14. Debug Structure

Offset	Name	Description	Filled in By	Valid
0000	256-byte window	Debug-specific information	–	–

Struct BaseOptions (pointed by BaseOptions in ROM_TABLE)

The base options structure is used to define dependent hardware options for the adapter. All options are disabled until enabled by the AIB Init code. Therefore, if Local Bus parity checking is desired, it is the AIB's job to enable parity in this structure. The Base ROM reads this structure after POST is complete and enables the options as defined by the AIB Init code. If no AIB is installed, the Base ROM will enable all options and update the structure to show that the options are enabled.

Table 12-15. BaseOptions Structure

Offset	Name	Description	Filled in By	Valid
0000	LBUS_Parity;	Local Bus Parity Checking (0 = no parity, 1 = parity)	AROM	AINIT
0004	Base_Option2;	Reserved = 0	BROM	- -

Table 12-15. BaseOptions Structure

Offset	Name	Description	Filled in By	Valid
0008	Base_Option3;	Reserved = 0	BROM	- -
000C	Base_Option4;	Reserved = 0	BROM	- -

Struct RomPostStatus (pointed by RomPostStat pointer in ROM_TABLE)

This structure is used to log specific information regarding Base POST test failures. This structure is created primarily for use during maintenance effectiveness testing and problem determination.

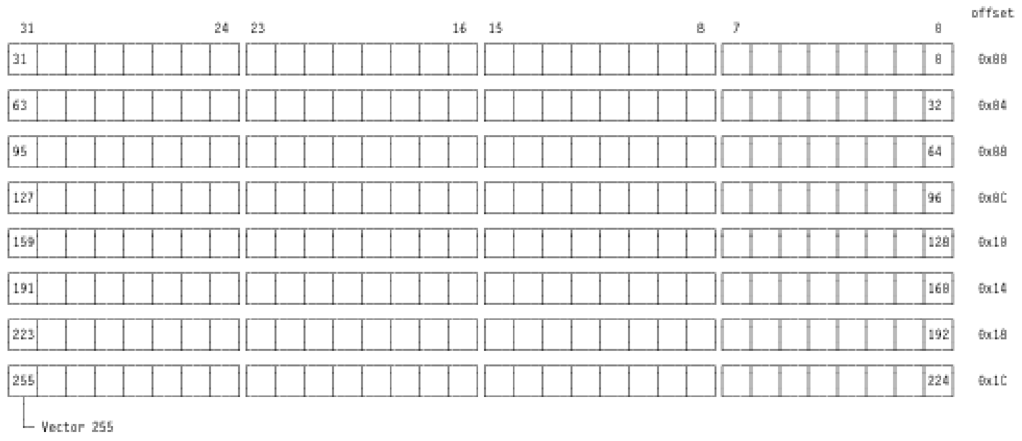
Table 12-16. RomPostStatus Structure

Offset	Name	Description	Filled in By	Valid
0000	TESTNUMBER;	POST Test number of test that failed	BROM	BPOST
0004	RESULT;	Error code	BROM	BPOST
0008	ErrorAddress;	Address of failed register	BROM	BPOST
000C	DataExpected;	Return data expected	BROM	BPOST
0010	DataReceived;	Return data received	BROM	BPOST
0014	InterruptFlag;	0=no interrupt, 1=interrupt	BROM	BPOST
0018	InterruptType;	Vector of interrupt that occurred	BROM	BPOST
001C	ATRRegContent;	Address of working status register	BROM	BPOST
0020	StatRegContent;	Contents of the working status register	BROM	BPOST
0024	SingleBitErr;	Number of single-bit ECC errors	BROM	BPOST

Struct IntID (pointed by IntID pointer in Base ROM_TABLE)

The IntID structure is an 8-word block that identifies the source of interrupts when a POST test is running. Each bit, starting from bit 0, corresponds to a specific interrupt vector number (0–255). Each time a POST test executes, it first resets all bits of the structure, runs its test, then examines the IntID block. If multiple interrupts, or unexpected interrupts occur, the POST test ends with an INVALID_INT error. The AIB POST and Init code should use this structure for its testing when required.

When an interrupt service routine is invoked, it sets the bit corresponding to the vector number of interrupt being serviced.



AIB Structures Used by the Base ROM

In general, the Base ROM has completed all ROM interface structures (with the exception of the hardware resource table) prior to calling AIB POST or AIB Init. The AIB must determine on which type of base card it is installed so the addresses of fixed resources (such as the Memory Controller Chip registers) can be determined. The AIB should use the **CardType** field in the ROM_TABLE to determine the card type. Allocatable resources (such as the ROM_TABLE) can be determined via the Base ROM Anchor address that is passed to the AIB INIT and POST routines in Register **g1**, when called. The AIB code routines can hook into interrupt vectors by moving the address of the AIB service routines into the Interrupt Vector Table stored in memory. The AIB should only hook into vectors that are designated for the AIB (see *Interrupt Table Structure* on page 180). If the AIB code needs to hook into a vector that is not listed as sourced by the AIB, the AIB code must restore the interrupt table to the original service routine address prior to returning back to the Base ROM code..

The AIB must write to the (EDR) Interrupt Select Enable Register on the base card to enable interrupts from the AIB and configure the interrupt controller for encoded or direct interrupts. The ISR is located at address 0x0A000004 (Bit 1: +encoded/-direct; Bit 3: AIB interrupts enabled).

During AIB POST or INIT, the code must preserve all registers, except the local registers and global registers **g0** and **g1**. The global registers (**g0** and **g1**) are used for passing parameters between the AIB code and the Base ROM code. See *Struct BaseOptions (pointed by BaseOptions in ROM_TABLE)* on page 209.

AIB Init code must enable the adapter options it desires in the BaseOptions structure in base card memory. If local bus parity checking is desired, the LBUS_Parity field of the BaseOptions structure must be set to 1; otherwise, parity checking on the adapter will be disabled. See *Struct BaseOptions (pointed by BaseOptions in ROM_TABLE)* on page 209.

A new field for the IBM ARTIC960 PCI ROM table, **CardTypeFlags** at offset 0xB4 of the ROM_TABLE, indicates three values:

- Base card processor frequency
- Data cache enabled
- Channel type

Bits 31–28 = 0000 means the base card processor speed is 25 MHz (the default).

Bits 27–04 are reserved.

Bit 3 = 1 means data cache is enabled.

Bits 02–00 = 001 means this is a PCI adapter.

The AIB developer might want to utilize these values.

AIB POST developers may want to take advantage of the IntID structure during testing. This structure can provide valuable information regarding interrupts that have been serviced during the POST routine. For more information regarding the IntID structure, see *Struct IntID (pointed by IntID pointer in Base ROM_TABLE)* on page 210..

AIB developers who intend to use a different POS ID than the default co-processor platform POS ID must provide a copy of the co-processor platform ADF/ADP file, as well as the base co-processor platform diagnostic file on their specific option disk. This is required for the instances where an adapter ID must be written to POS (for error conditions) prior to accessing the AIB, or when an AIB error exists that prevents the Base ROM from obtaining the POS ID.

AIB ROM Signature Block

The AIB ROM signature block is located at offset zero within the AIB ROM. It is the anchor that allows the Base ROM to find the AIB ROM and locate all important AIB ROM Data.

Struct AIB_ROM_ANCHOR

ULONG	Signature; (0x AA55 AA55)
ULONG	Reserved;
ULONG	Length; (AIB ROM Length)
ULONG	VPD_Ptr; (Pointer to VPD struc--->)
ULONG	uPBind_Ptr; (Pointer to BIND struc-->)
ULONG	User_Ptr; (Pointer for AIB Structs)
ULONG	RomVersion; (AIB ROM level)
ULONG	ChkSum; (Value forces AIB ROM to 0)
ULONG	AIB_ID; (AIB Identification)
ULONG	POS_ID; (Reserved field. This field must be set to 0xFFFFFFFF = default IBM ARTIC960 Co-Processor Platform card ID)
char	ROMDate(8); (ROM Version Date)
ULONG	Resfield1; (Reserved)
ULONG	Resfield2; (Reserved)
ULONG	HWRscTableCount; (Count of entries in H/W resource table)
ULONG	HWRscTable; (Offset of AIB H/W resource table)

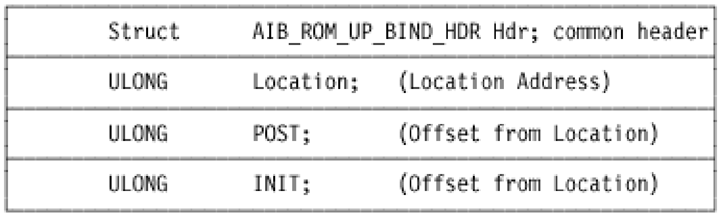
Vital Product Data Block

The AIB VPD is read by the Base ROM and placed in the packet memory under the Base ROM VPD. See *Struct VPD (pointed by VPD_ptr in ROM_TABLE)* on page 208 for a definition of the fields in this structure.

AIB Processor Binding Block

The processor binding block contains an ID (uP_ID) that identifies the processors bound by the block. The block contains the location of the AIB ROM, the offset where POST starts, and the offset where INIT starts.

Struct AIB_ROM_960_BIND



Struct	AIB_ROM_UP_BIND_HDR Hdr; common header	
ULONG	Location;	(Location Address)
ULONG	POST;	(Offset from Location)
ULONG	INIT;	(Offset from Location)

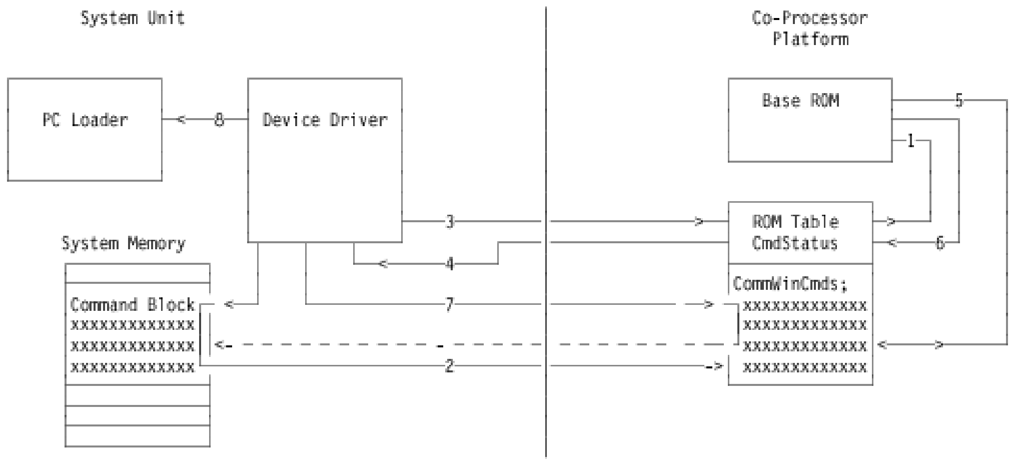
```
struct  AIB_ROM_UP_BIND_HDR
{
  ULONG   Reserved;   must be zero
  ULONG   NextLink;   offset to next uP binding
  ULONG   UP_ID;      Processor ID
};
```

AIB Hardware Resource Table

Entries in the AIB Hardware Resource table are of the same format as the entries in the Base Card Hardware Resource Table. See *Struct RIC_RDTEntry (pointed by *HWResTabPtr in ROM_TABLE)* on page 207 for a description of the fields in this table.

Kernel Structures Used by Base ROM (Boot Strap Loader)

Once the co-processor platform has completed its initialization, POST, and enabled the ROM Ready Bit, the platform remains in a WAITLOOP routine, waiting for a command from the system unit to load a supervisor task. The following flow diagram shows the handshaking process between the device driver and the co-processor platform.



Flow Sequence

1. Base ROM loops on CmdStatus looking for a non-zero value.
2. The device driver copies the command block from system memory to the command window pointed to by the SUCommands pointer in the ROM_TABLE.
3. The device driver sets CmdStatus to 0xFF.
4. The device driver waits for CmdStatus to be returned to zero.
5. The Base ROM gets the input parameters for the command buffer area, processes the command, fills in the return parameters, (if required), and sets the return code.

   If an error or unexpected interrupt occurs during the processing of a system unit command, the Base ROM will notify the system unit by loading the exception window with the Invalid_Intr structure. Note that the Base ROM will not generate an interrupt to the system unit when this occurs. It is expected that the system device driver will check the exception window area prior to issuing new commands. See *Struct RIC_Except (pointed by ExceptionPtr in ROM_TABLE)* on page 206 for more information regarding the invalid interrupt structure and exception window.

6. The Base ROM resets the CmdStatus to zero to signal device driver.
7. The device driver copies the command window back to the location in system memory.

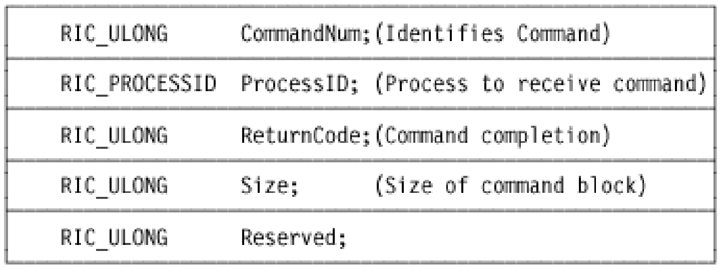
8. The device driver will return control to the system unit.

The SupervisorLoaded field in the ROM_TABLE will be set to 1 to indicate successful loading of the SUPERVISOR task.

The system unit will issue a command through the shared storage window to the co-processor platform in a command block. The command blocks have a common header with variant parts unique to each command. The command block format is described in the following sections.

System Unit Commands

Struct RICKERN_CMD



RIC_ULONG	CommandNum;	(Identifies Command)
RIC_PROCESSID	ProcessID;	(Process to receive command)
RIC_ULONG	ReturnCode;	(Command completion)
RIC_ULONG	Size;	(Size of command block)
RIC_ULONG	Reserved;	

```
struct ROMCMDSTRUCT
{
     struct RICKERN_CMD    Header;
     union
     {
            struct LoadProcessCmd         Cmd2;
            struct FillCmd                Cmd9;
            struct SetProcessEntryCmd     Cmd7;
            struct RIC_StartProcessCmd    Cmd6;
     }Cmds;
}
```

The sequence to download and start a supervisor task from the system unit to the co-processor platform is as follows.

1. Loader issues a LOAD_PROCESS command using the struct LoadProcessCmd (Cmd2=0x03). The Base ROM Checks for ProcName = RIC_KERN.REL for a valid download task. The Base ROM, using the MEM struct and either Instruction or packet memory, determines where to place the Kernel's download code, data, parameters, stack, and so forth. The Base ROM builds the KernInpInfo struct with all the memory addresses the information necessary for loading a process.
2. Loader issues a FILLCOMMAND using the structure FillCmd (Cmd9=0x0a). This command passes the paramenter to initialize memory to zero before use.
3. Loader issues an ENTRY_POINT command using the struct SetProcessEntryCmd (Cmd7=0x08). This command passes the EntryPoint vector to the Base ROM. The Base ROM uses this vector to start the process.

4. Loader Issues START_PROCESS command using the struct RIC_StartProcessCmd (Cmd6=0x07). This command passes the address of the KenInpInfo (Kernel input information) struct to the Kernel. The Kernel uses this information to build its startup Kernel structures.

Load Process Command

This command gets the memory addresses for loading a process.

Struct LoadProcessCmd

RIC_ULONG	Priority;	(Priority task will execute at)	Passed
RIC_ULONG	Textsize;	(Size of code area)	
RIC_ULONG	Datasize;	(Size of data area)	
RIC_ULONG	Stacksize;	(Size of stack for process)	
RIC_ULONG	ParamLength;	(Size of parameter area)	
Char	ProcName[SUPERVISOR];	(16 char max)	
Void	*Textaddr;	(Address of code area)	Returned
Void	*Dataaddr;	(Address of data area)	
Void	Paramaddr;	(Address for the parameter area)	
Void	*Stackaddr;	(Address of stack area)	
RIC_PROCESSID	ProcessID;	(Process number = 0)	
RIC_ULONG	Reserved;	(Bit 0 = 1 Cache Stack request) (Bit 1 = 1 Data Cache request)	Passed

Fill Command

This command fills an area of memory with the given character.

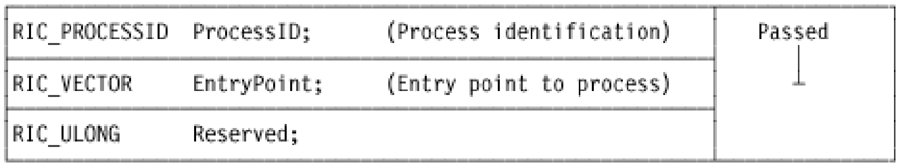
Struct FillCmd

RIC_ULONG	StartAddress;	(Address to fill)	Passed
RIC_ULONG	Length;	(Byte to write)	
RIC_ULONG	CharToFill;	(Character)	
RIC_ULONG	Reserved;		

Set Process Entry Command

This command sets the entry point for a process given the process identification.

Struct SetProcessEntryCmd

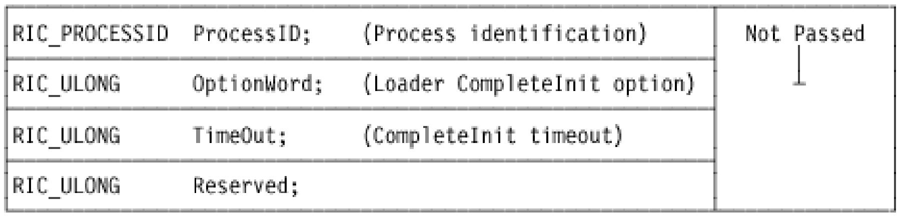


RIC_PROCESSID	ProcessID;	(Process identification)	Passed
RIC_VECTOR	EntryPoint;	(Entry point to process)	
RIC_ULONG	Reserved;		

Start a Process Command

This command starts a process given the process identification.

Struct RIC_StartProcessCmd

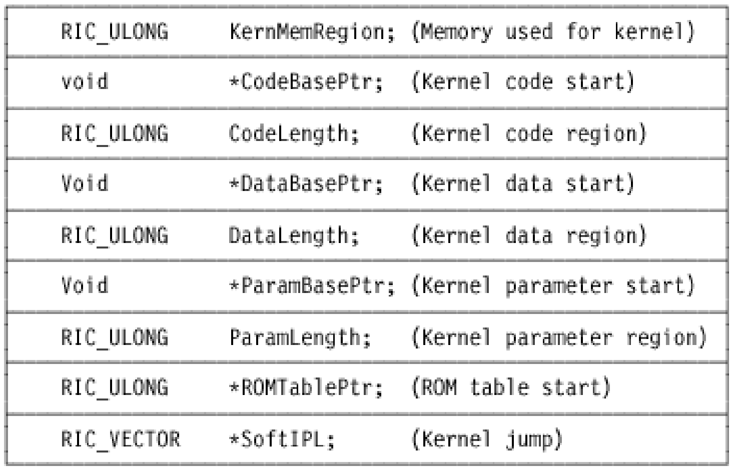


RIC_PROCESSID	ProcessID;	(Process identification)	Not Passed
RIC_ULONG	OptionWord;	(Loader CompleteInit option)	
RIC_ULONG	TimeOut;	(CompleteInit timeout)	
RIC_ULONG	Reserved;		

Kernel Input Structure

Before passing control to the Kernel, on successful load, the ROM builds the Kernel input struct in memory and passes the pointer (address of the struct) to the Kernel.

Struct KernInp



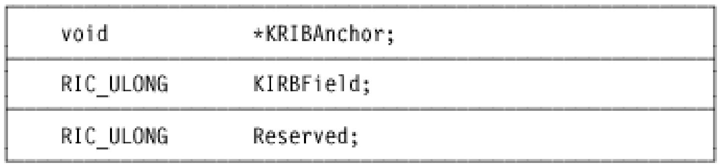
RIC_ULONG	KernMemRegion;	(Memory used for kernel)
void	*CodeBasePtr;	(Kernel code start)
RIC_ULONG	CodeLength;	(Kernel code region)
Void	*DataBasePtr;	(Kernel data start)
RIC_ULONG	DataLength;	(Kernel data region)
Void	*ParamBasePtr;	(Kernel parameter start)
RIC_ULONG	ParamLength;	(Kernel parameter region)
RIC_ULONG	*ROMTablePtr;	(ROM table start)
RIC_VECTOR	*SoftIPL;	(Kernel jump)

Kernel Structure Used by the Kernel through the ROM Table

Set Control Block Anchor

This call sets a control block address for a device driver or subsystem in the Kernel resource interface block (KRIB). The KRIB allows utilities executing in the system unit to access and display adapter structures.

Struct SetControlBlockAnchor



void	*KRIBAnchor;
RIC_ULONG	KIRBField;
RIC_ULONG	Reserved;

4-Port Multi-Interface Application Interface board

13

This chapter contains both hardware and software introductory and reference information for the IBM 4-Port Multi-Interface Application Interface Board (AIB), and is intended for hardware and software designers, programmers, engineers, and anyone who needs to understand the use and operation of this AIB.

The 4-Port Multi-Interface AIB:

- Attaches to the RadiSys ARTIC960 Co-Processor Platform, and through optional cables, provides up to four ports (network links) of high-speed interface to any **one** of the following electrical interfaces:
  - EIA-232-D
  - EIA-530 (RS-422)
  - ISO 4902 (V.36/V.35 compatible)
  - ISO 4903 (X.21)
- Connects to the RadiSys ARTIC960 Co-Processor Platform by a 140-pin connector.
- Once attached to the Co-Processor Platform, occupies just one 32-bit expansion slot in IBM Personal System/2 (PS/2) Computers and IBM Industrial Computers that are Micro Channel compatible.
- Interface signals exit the AIB through a 100-pin connector at the rear of the board.

Data Rates

External Clocking: When clocks are supplied by an external device, the 4-Port Multi-Interface AIB supports four ports running simultaneously at a maximum data rate of 2.048M bits per second (bps), duplex, and synchronous, except for the EIA-232 interface.

The maximum speed supported for each electrical interface is as follows.

Table 13-1.

Electrical Interface	Maximum Speed (per port)
EIA-232-D	38.4K bps (U.S. only) 19.2K bps (EMEA only)
EIA-530	2.048M bps
ISO 4902 (V.36)	2.048M bps (U.S. only) 64K bps (EMEA only)
ISO 4903 (X.21)	2.048M bps

The 64K bps limit on the V.36 interface for EMEA is due to the CCITT maximum speed limit of 72K bps for the V.36 interface.

Internal Clocking: When configured for an **outbound** transmit clock, the 4-Port Multi-Interface AIB provides the clock from one of two sources:

- A programmable clock from 9.6 KHz to 460.8 KHz can be generated by the Dual Universal Serial Communications Controllers (DUSCCs). For more details, refer to the *Signetics SC26C562/SC68C562 CMOS DUSCC User's Guide*.
- An on-card circuit can provide a 1.544 MHz or 2.048 MHz clock for each port. See *Transmit Clock* on page 238 and *Port 0–3 X.21 Enable Registers (X21EN0–3)* on page 286.

This clock can be driven out to external circuitry. The line interface signals supported per electrical interface are shown in Table 13-10.

Optional Cable Assemblies

Four optional cable assemblies are available for use with the 4-Port Multi-Interface AIB. Each cable assembly has four ports and is for a single electrical interface—for example, EIA-232, EIA-530, ISO 4902 (V.36), or ISO 4903 (X.21).

The individual cables are 1.8 meters (6 feet) long and have a 100-pin, male D-shell connector on the end that attaches to the AIB. The 100-pin end of each cable branches into four individual cables, each of which is terminated with a male D-shell connector—as illustrated under *Cable Configurations* on page 248.

The type of connector provided on the multiconnector end of each AIB cable conforms with the following electrical interfaces.

Table 13-2.

Electrical Interface	Connector Type
EIA-232-D	25-pin D-shell
EIA-530	25-pin D-shell
ISO 4902 (V.36)	37-pin D-shell
ISO 4903 (X.21)	15-pin D-shell

Additional information about the cables is under *AIB Cable Assemblies* on page 247.

Hardware Implementation

This section describes the hardware implementation of the 4-Port Multi-Interface AIB. The software implementation is described under *Software Implementation* on page 253.

Card Architecture

A block diagram of the 4-Port Multi-Interface AIB is shown in Figure 13-1. Each of the blocks is briefly described in the paragraphs that follow.

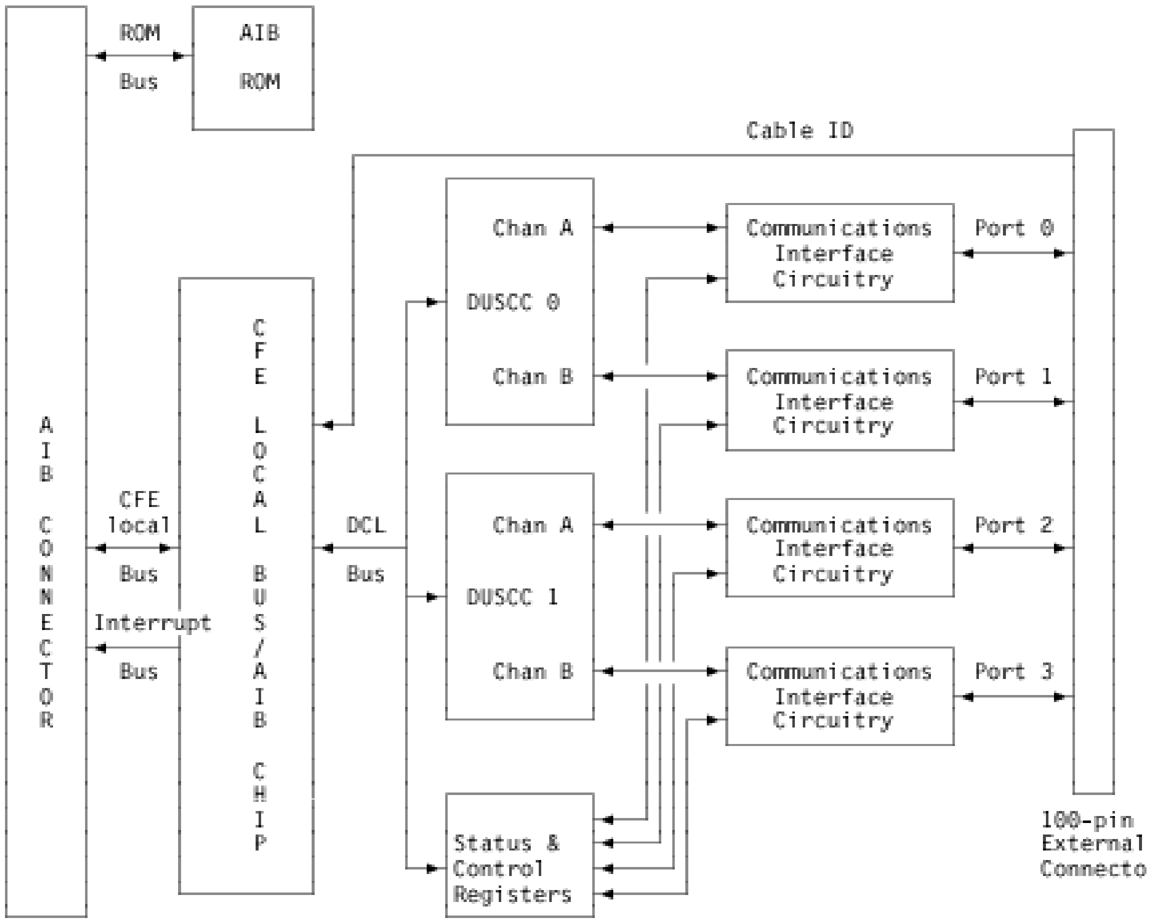

ROM
Bus
AIB
ROM
Cable ID
AIB CONNECTOR
CFE local Bus
Interrupt Bus
CFE LOCAL BUS/AIB CHIP
DCL Bus
Chan A
DUSCC 0
Chan B
Chan A
DUSCC 1
Chan B
Status & Control Registers
Communications Interface Circuitry
Communications Interface Circuitry
Communications Interface Circuitry
Communications Interface Circuitry
Port 0
Port 1
Port 2
Port 3
100-pin External Connecto

Figure 13-1. Card Architecture

Legend

AIB = Application Interface Board
ROM = Read-only memory
CFE = Common Front End architecture
DCL = Daughter Card Local
DUSCC = Dual Universal Serial Communications Controller

AIB Connector

The 140-pin AIB Connector connects the 4-Port Multi-Interface AIB to the RadiSys ARTIC960 Co-Processor Platform. The pin assignments for this connector are listed on Page 63. The AIB connector contains:

- 5V and +/-12V power
- Grounds (signal and power)

- 3 interface buses

The three interface buses are:

- The read-only memory (ROM) Bus
- The common front end (CFE) of the co-processor platform Local Bus
- The CFE Local Bus/AIB Interface Chip Interrupt Bus.

ROM Bus

The ROM Bus is an 8-bit data and 16-bit address bus with control lines to handle reading and writing the ROM on the 4-Port Multi-Interface AIB.

It is used by the co-processor platform to access the AIB ROM as a byte-wide device arranged as a 64KB area of memory.

CFE Local Bus

The co-processor platform CFE Local Bus contains a 32-bit multiplexed address/data bus, operating at 25MHz, and provides an interface between the co-processor platform and the AIB Connector.

Interrupt Bus

The CFE Local Bus/AIB Interface Chip interrupt vector bus forms an 8-bit interrupt vector that is passed to the co-processor platform. The expanded mode interrupt structure of the Intel 80960 processor is an example of this interrupt structure.

The CFE Local Bus/AIB Interface Chip internally manages the prioritization of interrupt input signals and translates those inputs into specific interrupt vectors.

AIB ROM

The 4-Port Multi-Interface AIB ROM is a 32Kx8 Flash ROM, which contains code and identification information to be read or executed by the co-processor platform. It is an Intel 28F256 device with a 200 ns cycle time and is connected to the co-processor platform through the ROM Bus. It is writable by the co-processor platform.

No write protection is implemented on the AIB, but the co-processor platform provides a mechanism to prevent accidental corruption of the ROM data. This is detailed in the *AIB ROM Write Enable Register (AWE)* on page 167.

The AIB ROM provides:

- Card identification
- Self-test of card features
- Initialization of card features

The information provided for card identification and the operations performed in self-test are described under *AIB ROM Description* on page 296.

The ROM is programmed during card manufacture. Programming by the end user is not supported, except for field updates provided by RadiSys.

CFE Local Bus/AIB Interface Chip

The CFE Local Bus/AIB Interface Chip module is the interface between the co-processor platform Local (CFE) Bus and the Daughter Card Local (DCL) Bus. The DMA Subsystem contained on this chip is described, starting on page 226.

DCL Bus

The Daughter Card Local (DCL) Bus is the interface between the CFE Local Bus/AIB Interface Chip module and its peripherals. These peripherals are the DUSCCs and the Status and Control Registers.

DUSCC 0 and 1

The Signetics SC26C562 Dual Universal Serial Communications Controller (DUSCC) is the Communication Controller for serial data communication. This is a CMOS version of the DUSCC used on previous ARTIC products, such as the Multiport Adapter/A. The oscillator frequency supplied to the DUSCCs is 14.745 MHz. For complete details on the Signetics SC26C562 DUSCC, see the *Signetics SC26C562/SC68C562 CMOS DUSCC User’s Guide*.

Status and Control Registers

Control of individual port enables and other logic is implemented in Field Programmable Gate Arrays (FPGAs). These functions include:

- Output driver enables
- Input receiver selects
- On-card clock generation
- X.21 pattern detection
- External clock detection

Details of the content and usage of these registers are under *FPGA Registers* on page 282.

Electrical Communications Interface Circuitry

The Electrical Communications Interface Circuitry consists of receivers, drivers, and signal-conditioning circuitry required to meet the Electrical Interfaces in Table 13-10. They are the interface between the DCL Bus peripherals and the 100-pin external connector.

External Connector

The 100-pin External Connector connects the 4-Port Multi-Interface AIB to external devices—through one of four AIB cable assemblies described under *AIB Cable Assemblies* on page 247. This connector contains all the signals and grounds for each of four output ports. See Page 225 for more information about the 100-pin External Connector.

Cable ID

Four pins are brought from the external connector to inputs of the CFE Local Bus/AIB Interface Chip module. These signals are pulled up through resistors on the card, and can be connected to ground in the cable or wrap plug to form a 4-bit ID. The state of these pins is readable through the Presence Detect Register (PDR) of the CFE Local Bus/AIB Interface Chip to determine the type of cable or wrap plug that is connected to the card. Additional information is under *Presence Detect Register (PDR)* on page 281.

AIB Specifications

The 4-Port Multi-Interface AIB specifications (electrical, environmental, and physical) are listed in Table 13-3 through Table 13-9.

Electrical

Table 13-3. Electrical Characteristics

	No Ports Wrapped		All Ports Wrapped	
Voltage	**Typical**	**Maximum**	**Typical**	**Maximum**
+5V DC	0.5 A	0.9 A	0.6 A	1.2 A
+12V DC	12 mA	25 mA	50 mA	75 mA
-12V DC	12 mA	25 mA	50 mA	75 mA
Total Power	2.8 W	5.1 W	3.7 W	7.8 W

Environmental

Table 13-4. Environmental Characteristics

Test	Specification
Temperature	16 to 32°C
PS/2[1]	0 to 60°C
Industrial Computer[1,2]	0 to 60°C
Storage	–40 to 60°C
Shipping	
Humidity (Non-Condensing)	8–80%
PS/2	5–95%
Industrial Computer	5–80%
Storage	5–100%
Shipping	
Wet Bulb	23°C
PS/2	29°C
Industrial Computer	29°C
Storage	29°C
Shipping	
Altitude (Operating)	0–7000 ft.
PS/2	0–10,000 ft.
Industrial Computer	

1. System ambient temperatures.

2. The operating temperatures of Industrial Computers are dependent on the system hard files and adapters being used.

Physical

Table 13-5. Physical Characteristics

Dimensions	216 mm (8.5 inches) long x 80 mm (3.15 inches) high

The 4-Port Multi-Interface AIB (Figure 13-2) connects to the co-processor platform through a 140-pin connector. The mating height of this connector is 0.495 (±0.005) inches.

Four standoffs are provided to secure the AIB to the co-processor platform. Eight 4-40 screws are provided—four to fasten the standoffs to the AIB and four to secure the co-processor platform to the standoffs.

The 100-pin External Connector (J1) connects the AIB to external devices—through one of four 4-Port Multi-Interface AIB cable assemblies described under *AIB Cable Assemblies* on page 247. The external connector contains all the signals and grounds for each of four output ports. The pin assignment for this connector is under *100-Pin External Connector* on page 225.

An insulating shield is provided to cover the back of the AIB and prevent it from contacting cards in adjacent slots while installed in a system.

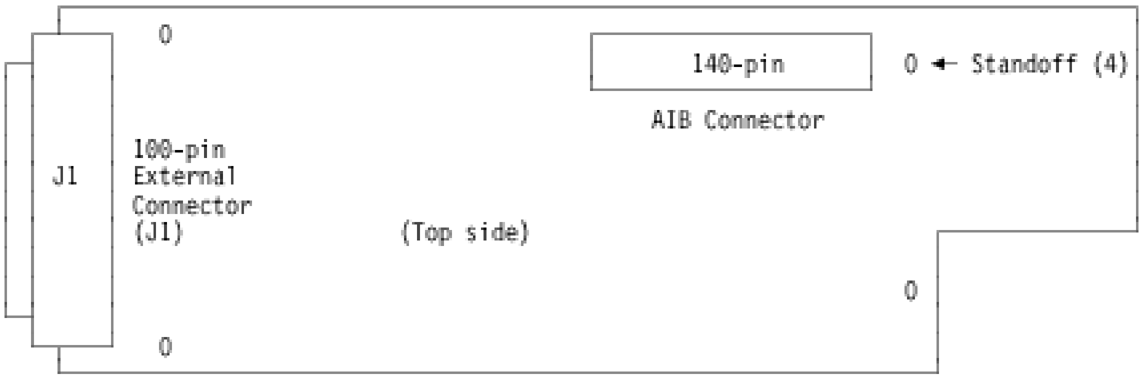


Figure 13-2. Card Layout of the 4-Port Multi-Interface AIB—Top Side

100-Pin External Connector

The pinout and pin assignment for the 100-pin external connector (J1) for the 4-Port Multi-Interface AIB are shown in Figure 13-3 and listed in Table 13-6.

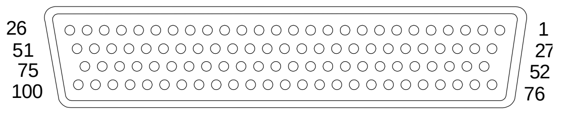


Figure 13-3. 100-Pin Connector Pinout

Table 13-6. 100-Pin External Connector (J1)

		J1 Pin Assignments			
Signal Name	**I/O**	**Port 0**	**Port 1**	**Port 2**	**Port 3**
TxDn(A)	O	1	9	15	21
TxDn(B)	O	27	34	41	47
TxD*n*	O	5	12	18	24
RxD*n*(A)	I	52	58	64	95
RxD*n*(B)	I	76	82	88	70
RxD*n*	I	80	86	92	99
TCLKI*n*(A)	I	53	59	65	96
TCLKI*n*(B)	I	77	83	89	71
TCLKO*n*(A)	O	2	10	16	22
TCLKO*n*(B)	O	28	35	42	48
RCLKI*n*(A)	I	54	60	66	97
RCLKI*n*(B)	I	78	84	90	72
RTS*n*(A)	O	3	11	17	23
RTS*n*(B)	O	29	36	43	49
RTS*n*	O	31	37	68	74
CTS*n*(A)	I	30	61	67	98
CTS*n*(B)	I	4	85	91	73
CTS*n*	I	32	39	45	51
DTR*n*	O	7	13	19	26
DSR*n*	I	57	63	69	50
CD*n*(A)	I	55	33	40	20
CD*n*(B)	I	79	8	14	46
CD*n*	I	81	62	94	100
Ground	–	6	38	44	25
Cable ID 3	I	75			
Cable ID 2		93			
Cable ID 1		87			
Cable ID 0		56			

The *n* represents the port number.

CFE Local Bus/AIB Interface Chip DMA Controller

The functions of the DMA controller portion of the CFE Local Bus/AIB Interface Chip are as follows:

- Eight independent DMA channels
- Split-bus implementation (32-bit DCL Bus to 32-bit CFE Local Bus)
- Support of 8/16/32 bit DCL Bus devices
- 16-byte buffer per channel
- 16-byte burst capability on CFE Local Bus (57 MB/s peak bandwidth)
- 32-bit 4GB addressability on CFE Local Bus
- 16-bit 64KB addressability on DCL Bus for AIB operations (OPs)

- Separate DREQ and DACK signals for each DMA channel
- Programmable DACK cycle time
- 64KB byte-count capability
- 8-word descriptor block for each channel
- Linked list chaining of buffers on all channels
- Chaining support for end-of-process and terminal count condition
- Buffer queuing (up to 4K–1 buffers) on all channels
- Automatic storage of residual transfer count on chain event
- Automatic storage of AIB OP reads on chain event
- Up to two programmable auto I/O operations on chain event
- Seven interrupt sources for each channel.

Each of the eight DMA channels are of equal function. A channel can be programmed to support data transfer from: (1) the DCL Bus to the CFE Local Bus (receive), or (2) the CFE Local Bus to the DCL Bus (transmit). The service priority is fixed, with CH0 having the highest priority and CH7 having the lowest priority within the round robin.

Each channel is controlled by an 8-word Channel Descriptor Table (CDT), as described on Page 228. The program normally writes the desired CDT register values to memory resident structures called Channel Descriptor Blocks (CDBs). The program then writes the first CDB values directly into the CDR registers. With the linked-list-chain option, once the DMA channel is started, the CDBs can be automatically fetched by the DMA channel’s state machine.

Once enabled, a channel will service DMA requests from the DCL Bus until one of a number of programmable conditions is reached. If the DMA channel is programmed to stop on one of these conditions, the channel can be re-enabled with a write to the Channel Control Register (CCR), or by queuing another buffer. Any condition that can stop the channel can also interrupt the RadiSys ARTIC960 Co-Processor Platform. Also, the channel has the ability to interrupt, without stopping the channel. All of the options are programmable in the CCR.

Two types of DMA requests originate from the DCL Bus; these are: Transmit Request (TR) and Receive Request (RR).

Once a TR is received, the DMA controller arbitrates for control of the CFE Local Bus. When granted control, the DMA controller bursts 16 bytes of data from memory into that channel’s FIFO buffer. The data is then transferred to the requesting device across the DCL Bus until the FIFO buffer is empty. This allows data transfer to the AIB device to occur in the background of CFE Local Bus activity.

For a receive DMA channel, the DMA controller will receive up to 16 bytes of data across the DCL Bus. It then arbitrates for control of the CFE Local Bus. Once granted, the DMA controller bursts all 16 bytes into memory. This allows all data transfer from the AIB device to occur in the background of CFE Local Bus activity.

The data transfer from the CFE Local Bus to the DCL Bus is not direct; it is buffered in a set of FIFOs. There is a 16-byte-deep FIFO associated with each DMA channel; this FIFO acts as an intermediate storage area for data. The FIFOs support high-speed access so that all accesses to the CFE Local Bus move data into and out of the FIFOs, instead of directly to the DCL Bus, which might support only slower devices. This feature allows all CFE Local Bus DMA data transfer initiated by CFE Local Bus/AIB Interface Chip to be high-speed bursted accesses.

The FIFOs are configured such that DCL Bus accesses and CFE Local Bus accesses can occur simultaneously. One DMA channel can be performing a read or write on the DCL Bus, while at the same time, another channel is performing a read or write on the CFE Local Bus.

Channel Descriptor Table (CDT)

Table 13-7 shows the organization of the channel descriptor table for one of the DMA channels. This table is duplicated for each of the eight DMA channels. The table exists in the address space of the 80960 processor, and is accessible using 32-bit load- and store-type instructions.

Table 13-7. CFE Local Bus/AIB Interface Chip Channel Descriptor Table (CDT)

Register Name	CFE Local Bus Address	Valid Bits	See Page
Channel Control Register (CCR)	1FF8 x018h	21 – 0	262
Chain Pointer Register (CPR)	1FF8 x014h	31 – 0	265
Transfer Count Register (TQC)	1FF8 x010h	15 – 0	270
Memory Pointer Register (MPR)	1FF8 x00Ch	31 – 0	269
AIB_OP2 Data Register (AIB_OP2)	1FF8 x008h	31 – 0	261
AIB_OP1 Data Register (AIB_OP1)	1FF8 x004h	31 – 0	261
AIB Address 1/2 Register (AIB_ADDR1/2)	1FF8 x000h	31 – 0	260

Linked List Chaining (LLC)

The CFE Local Bus/AIB Interface Chip DMA controller supports a feature known as *Linked List Chaining (LLC)*, sometimes called *scatter/gather DMA*. This feature allows the programmer to create, in memory, lists of different data buffer areas and buffer counts (CDBs) that can be automatically reloaded to the CDT after a terminal count or end-of-process condition for the present data buffer has occurred. The channel can optionally interrupt or not interrupt the 80960 processor when any of these conditions has occurred. Also, the DMA channel can independently be stopped or allowed to run at this point. In addition, certain control information for the channel can be changed during the chaining operation.

Within the CDT, a 32-bit address pointer (CPR) is maintained. This address points to a CDB in memory. At the time of terminal count or end-of-process detection, if linked list chaining is enabled, the hardware will do three memory writes, followed by seven memory reads, starting at this chain pointer address. It stores into memory the present DMA transfer byte count, and any data read from the DCL Bus by AIB_OP1 or AIB_OP2. The hardware then fetches seven new words of CDB information as detailed in Figure 13-4. At the end of this operation, the CPR in the CDT will be the CPR value that was just read from the CDB in memory. The address is the starting address of the next CDB in the chain. This provides a mechanism to off-load the 80960 processor from handling the DMA terminal count or the end-of-process interrupts. In theory, this list of buffers could occupy any amount of free memory. This concept is detailed in Figure 13-4.

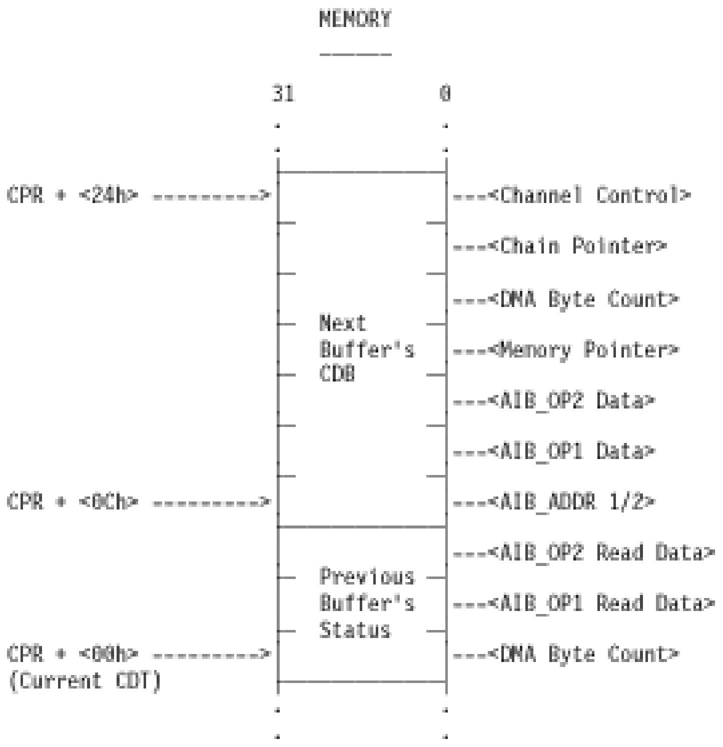


Figure 13-4. Linked List Chaining

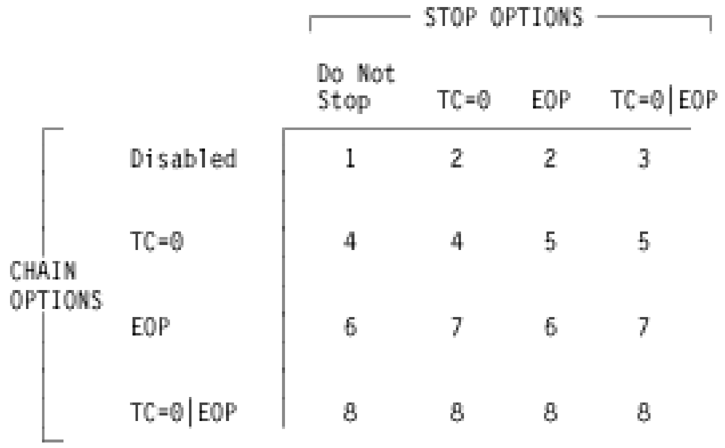
Note that the hardware will always perform the three memory writes, regardless of whether the option to do reads from the AIB is chosen. If this option is not chosen, the data written to these two fields will be indeterminate.

Linked List Chaining/Stopping Matrix

The CCR provides many different options for linked list chaining and stopping DMA channels. These options are described in CCR1. The chaining options and the stopping options must be looked at together to determine the action a DMA channel takes when these conditions are encountered during DMA channel execution. The following matrix describes how a DMA channel operates for all possible combinations of chaining and stopping options being programmed into the CCR.

Summary

If chaining is disabled, the DMA channel is stopped by the programmed option. If chaining is enabled, the stop programming options are ignored unless the programmed stop option is different from the programmed chaining option. The state of the CCR enable bit in a memory CDB determines whether a DMA channel remains enabled or disabled after the chaining operation occurs.



		STOP OPTIONS			
		Do Not Stop	TC=0	EOP	TC=0|EOP
CHAIN OPTIONS	Disabled	1	2	2	3
	TC=0	4	4	5	5
	EOP	6	7	6	7
	TC=0|EOP	8	8	8	8

1. DMA channel detects TC=0, but continues DMAing data to the next MPR address.
2. DMA channel stops servicing DMA requests when condition is detected.
3. DMA channel stops servicing DMA requests when either condition is detected.
4. DMA channel stops servicing DMA requests when TC=0 is detected, performs chaining operation, then is either re-enabled or stays disabled based on new CCR enable bit that is chained in.
5. DMA channel operates as in (4), but will stop servicing DMA requests, without chaining, if EOP is detected.
6. DMA channel stops servicing DMA requests when EOP is detected, performs chaining operation, then is either re-enabled or stays disabled based on new CCR enable bit that is chained in.
7. DMA channel stops servicing DMA requests when EOP is detected, performs chaining operation, then is either re-enabled or stays disabled based on new CCR enable bit that is chained in. If TC=0 occurs, the DMA channel stops servicing DMA requests with no chaining.

DMA channel stops servicing DMA requests when either condition is detected, performs chaining operation, then is either re-enabled or stays disabled, based on new CCR

Figure 13-5. CCR Chaining/Stopping Matrix

DMA Interrupts

Each of the eight DMA channels has seven possible sources of interrupts to the 80960 processor. Of these seven, two are classified as *normal termination* interrupts and five are classified as *error* interrupts. Normal termination interrupts are maskable in the CCR and error interrupts are non-maskable.

Each DMA channel has a fixed interrupt vector assigned to it. These eight interrupt vectors are at priority level 8 within the 80960 processor interrupt model. The following vectors are assigned to the eight DMA channels.

Table 13-8. CFE Local Bus/AIB Interface Chip DMA Channel Vector Assignment

DMA Channel No.	Interrupt Vector No.
0	71
1	70
2	69
3	68
4	67
5	66
6	65
7	64

Each DMA channel has an interrupt status register associated with it that is read during the interrupt subroutine to determine the pending interrupt status bits for that channel. This read clears all active status bits.

DMA Performance

The CFE Local Bus/AIB Interface Chip acts as a bridge between the bursting co-processor platform CFE (Local) Bus and the non-bursting DCL Bus. See Figure 13-5.

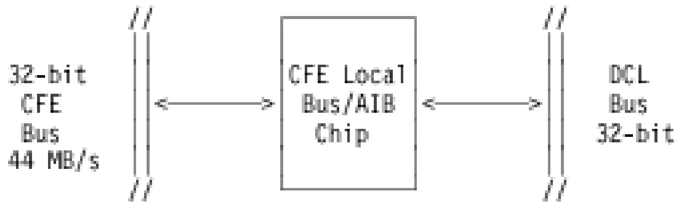


Figure 13-5. CFE Local Bus/AIB Interface Chip Bus Interface

The CFE Local Bus/AIB Interface Chip buffers both inbound (receive) and outbound (transmit) data 16-bytes deep for each DMA channel. This allows most accesses by the CFE Local Bus/AIB Interface Chip to the co-processor platform's CFE Local Bus to be bursted accesses of 44MB-per-second (MB/s), consisting of a first access of 182 ns, followed by three accesses of 61 ns each, with all four accesses being 4 bytes wide). Since the DCL Bus is a non-bursting bus, it is this bus that will limit the DMA bandwidth. The DMA controller will therefore consume some portion of the total 44MB/s CFE Bus bandwidth, with the remainder of the bandwidth being available to other master devices on the co-processor platform.

Factors that influence DCL Bus bandwidth are as follows:

- DMA cycle time, or more precisely, the DACK pulse width
- Non-DMA (CFE Bus random accesses) cycles to the DCL Bus
- DCL device cycle time
- DCL device bus width
- DRAM to Micro Channel throughput.

The following performance example assumes that the DCL device has a cycle time less than the cycle time capability of the CFE Local Bus/AIB Interface Chip DMA controller:

- Given a DACK cycle time of 121 ns (programmable in the DAPW Register) and a 32-bit-wide device, a 33MB/s DCL capability is realized.
- By subtracting from this, 3MB/s for the random CFE Bus accesses to the DCL Bus for status checking and control, a 30MB/s DCL capability is possible. (A 30MB/s capability would use approximately 70% of the 44MB/s CFE Local Bus capability; this is impractical since this just moves data to DRAM and not all the way to system unit memory.)
- Using no more than 50% of the CFE Local Bus bandwidth (this still allows for approximately 7 MIPS of 80960 processor compute power) to get data from the DCL device to the system unit, translates to about 20MB/s of CFE Local Bus activity. This means that 10MB/s of CFE Local Bus bandwidth can be used to get data from 4-Port Multi-Interface AIB to memory, and the other 10MB/s of CFE Local Bus bandwidth can be used to get data from memory to the system unit.

4-Port Multi-Interface AIB Interrupt Controller Subsystem

The general features of the interrupt controller subsystem are as follows:

- Two separate interrupt inputs, called *CFE Local Bus Interrupts* (reserved)
- Five separate interrupt inputs from the DCL (1 high priority)
- Support for both vectoring and non-vectoring DCL devices
- Programmable single or double (8259 type) IACK pulse
- Programmable IACK pulse width
- Programmable interrupt masking of DCL Bus interrupt inputs.

The RadiSys ARTIC960 Co-Processor Platform supports expanded mode interrupting, as implemented in the 80960 processor. An 8-bit interrupt vector bus is provided to directly attach to the 80960 processor. The CFE Local Bus/AIB Interface Chip takes the interrupt inputs and translates these into interrupt vectors that it places on the interrupt bus for interpretation by the co-processor platform. However, support is provided to allow two DCL devices to directly supply an interrupt vector for the co-processor platform. This is a programmable option.

Table 13-9 shows the priority scheme that is used to report various classes of interrupts back to the co-processor platform.

Table 13-9. Interrupt Vector Assignment

80960 Priority Level	Vector Number	Interrupt Condition	Interrupt Source
25-21	207-168	AIB INT0 (vector provided by DCL device)	DCL
20-16	167-128	AIB INT1 (vector provided by DCL device)	DCL
15	120	AIB INT2	DCL
14	112	AIB INT3	DCL
8	71-64	CFE Local Bus/AIB Interface Chip DMA channel 0-7	CFE Local Bus/AIB Interface Chip

The interrupt controller manages prioritizing the interrupts that these devices generate.

Interrupt inputs are not latched by the interrupt controller. The output of the interrupt controller to the interrupt bus reflects the value of the highest priority input active.

Electrical Interface Design

This section contains information about the following:

- Drivers/receivers signals
- Drivers/receivers controls
- Homologation design
- Driver/receiver block diagrams
- X.21 state detection

Drivers/Receivers Signals

Table 13-10 shows the numbers and types of signals required to support one port for each of the electrical interfaces. The controls to the drivers and receivers are illustrated under *Driver/Receiver Block Diagrams* on page 235.

Table 13-10. Drivers/Receivers

Signal Name	I/O	Voltage	EIA-232-D	EIA-530	X.21	V.36
TxD (T)	O	12V 5V Balanced	1 -	- 2	- 2 H	- 2 H
RxD (R)	I	12V 5V Balanced	1 -	- 2	- 2 H	- 2 H
TCLK IN	I	5V Balanced	-	2	-	2 H
TCLK OUT (X)	O	5V Balanced	-	2	2 H	2 H
RCLK IN (S)	I	5V Balanced	-	2	2 H	2 H
RTS (C)	O	12V 5V Balanced	1 -	- 2	- 2 H	1 -
CTS (I)	I	12V 5V Balanced	1 -	- 2	- 2 H	1 -
CD	I	12V 5V Balanced	1 -	- 2	-	1 -
DSR	I	12V	1 -	- -	- -	1 -
DTR	O	12V	1 -	- -	- -	1 -
Ground			1	1	1	1

- Signal names shown in parentheses are the X.21 names.
- The letter *H* in the table indicates drivers/receivers requiring homologation circuitry.

Drivers/Receivers Controls

The drivers and receivers on the 4-Port Multi-Interface AIB are controlled by software through the Control FPGA. The registers used are described under *FPGA Registers* on page 282.

In addition, the hardware will automatically disable all drivers when the -A_RESET signal from the CFE Local Bus/AIB Interface Chip module is activated, or when the +5V supply to the card drops below +4.25V (nominal).

Homologation Design

For the X.21 and V.36 interfaces:

Every balanced driver has the line impedance circuitry illustrated in Figure 13-6 to satisfy the homologation specifications.

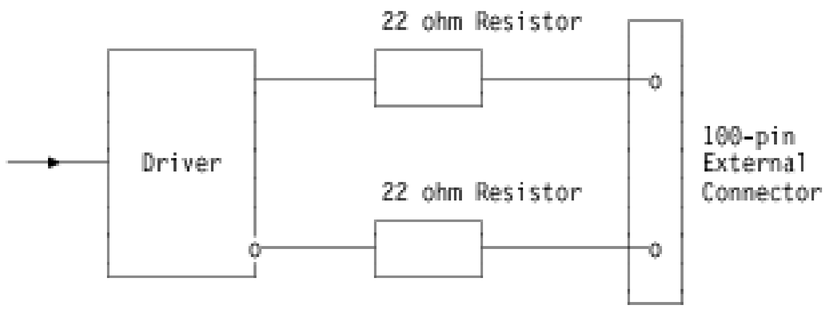


Figure 13-6. Homologation Circuit for Balanced Drivers

Every balanced receiver has the line impedance circuitry illustrated in Figure 13-6 to satisfy the homologation specifications.

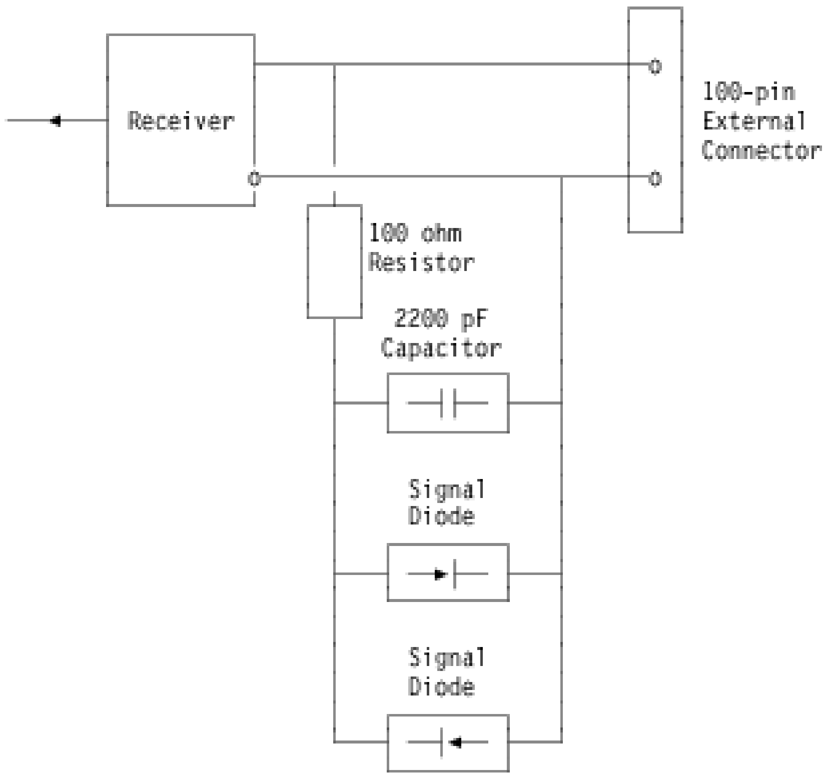


Figure 13-7. Homologation Circuit for Balanced Receivers

Driver/Receiver Block Diagrams

The following block diagrams (Figure 13-8 through Figure 13-15) illustrate the controls to the drivers and receivers on the 4-Port Multi-Interface AIB.

The diagrams show DUSCC 0 and its drivers and receivers to Ports 0 and 1 (Ports A and B of DUSCC 0). The circuitry for DUSCC 1 and the drivers and receivers for Ports 2 and 3 are not shown, but they are the same as shown for DUSCC 0..

Homologation circuitry is shown where required as blocks marked *H*.

Transmit Data

The following diagram illustrates the Transmit Data (TxD) signal.

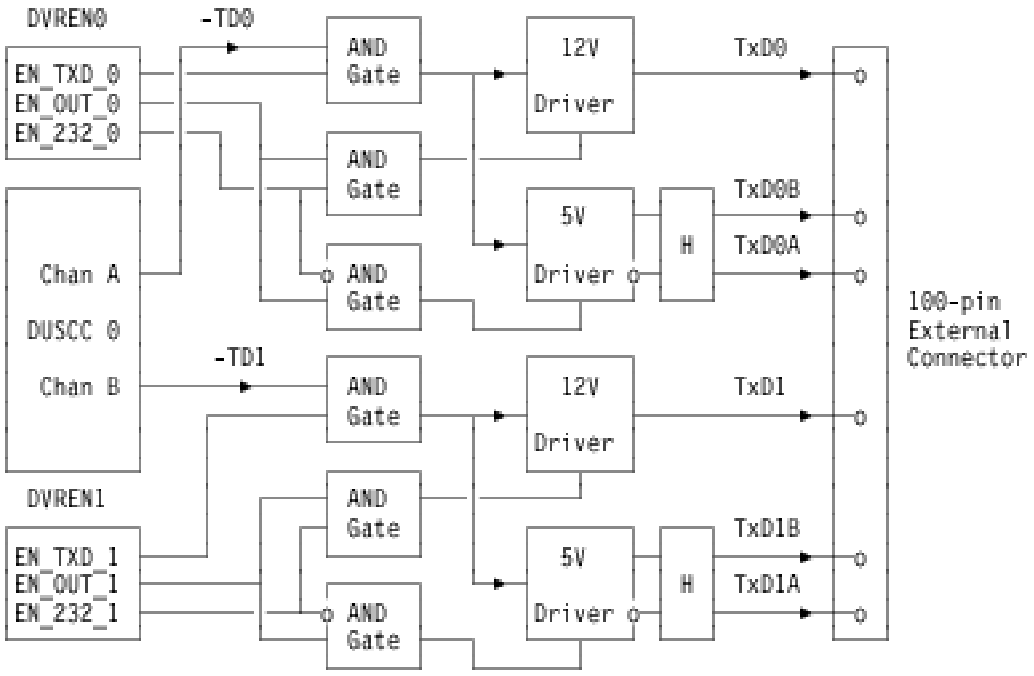


Figure 13-8. TxD Block Diagram

.

1. The output driver tri-state controls are controlled by a bit in DVRENn called EN_OUT_n. At power-on, the driver is disabled to prevent any data transmission from the DUSCC onto the line. This state is active until the software enables data communication by setting the EN_OUT_n bit to a logic '1'. (See *Port 0–3 Driver Enable Registers (DVREN0–3)* on page 283.)
2. The Xmt Data Enable gates block the data transmission from the DUSCCs to the transmit data lines (TxD) for transmission of all 0's (for use by X.21 applications). Each of these gates is controlled by a bit in DVRENn called EN_TXD_n. At power-on, the gates are blocking. This state is active until the software enables data communication by setting the EN_TXD_n bit to a logic '1'. (See *Port 0–3 Driver Enable Registers (DVREN0–3)* on page 283.)

Receive Data.

The “B” inputs of these signals are pulled to +5V through 10K resistors. The “A” inputs are pulled to GND through 10K resistors.

The following diagram illustrates the Receive Data (RxD) signal.

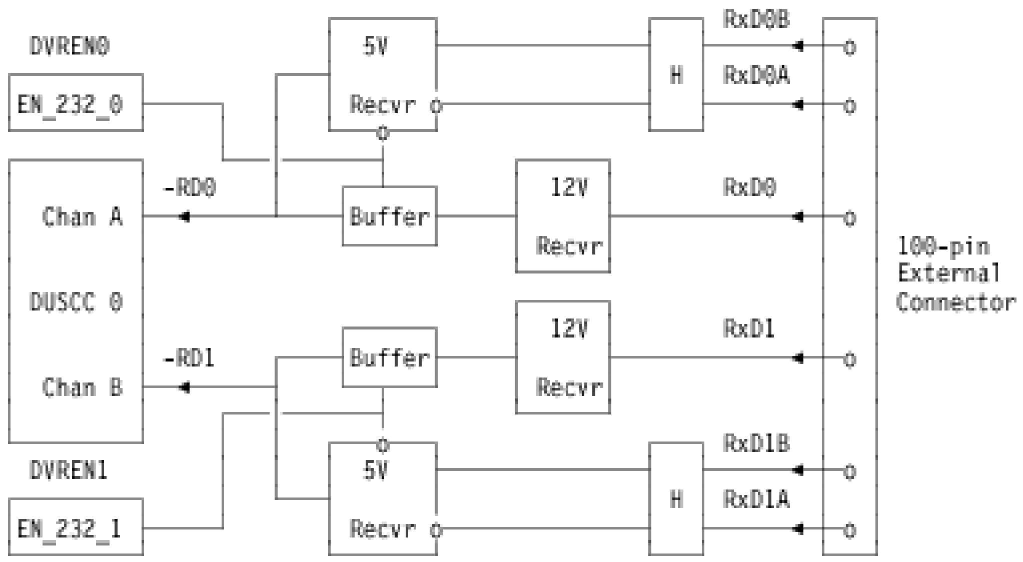


Figure 13-9. RxD Block Diagram

Transmit Clock

The following diagram illustrates the Transmit Clock (TxCLK) signal.

The “B” inputs of these signals are pulled to +5V through 10K resistors. The “A” inputs are pulled to GND through 10K resistors.

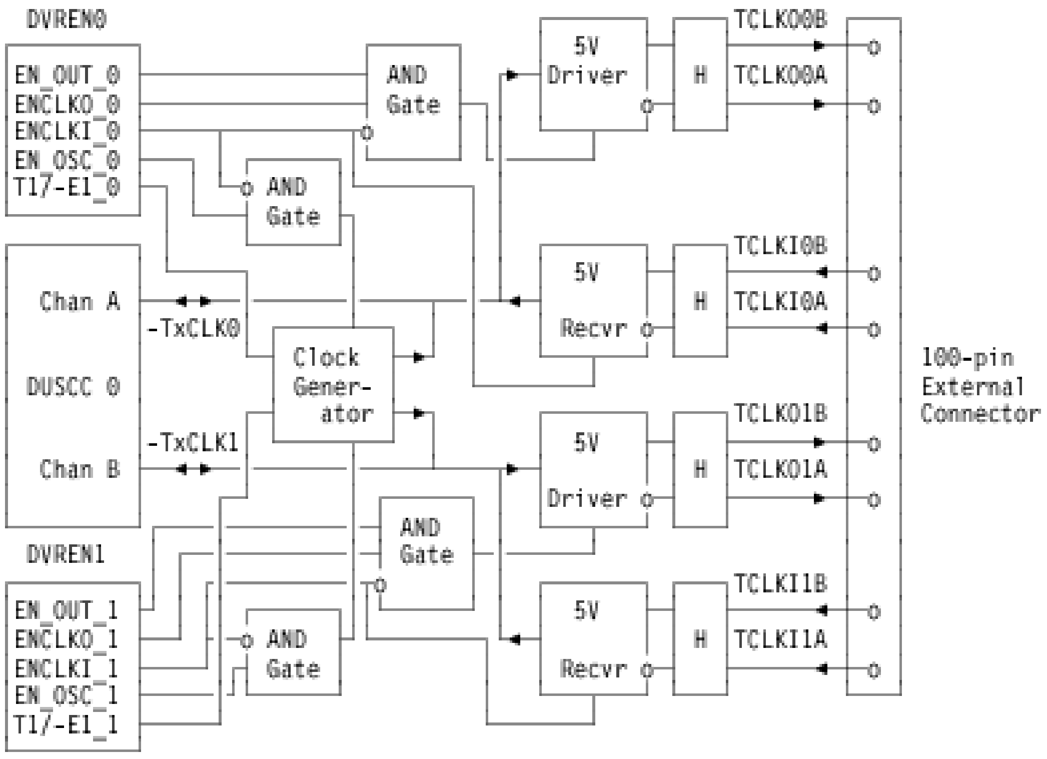


Figure 13-10. TxCLK Block Diagram

Receive Clock.

The “B” inputs of these signals are pulled to +5V through 10K resistors. The “A” inputs are pulled to GND through 10K resistors.

The following diagram illustrates the Receive Clock (RxCLK) signal.

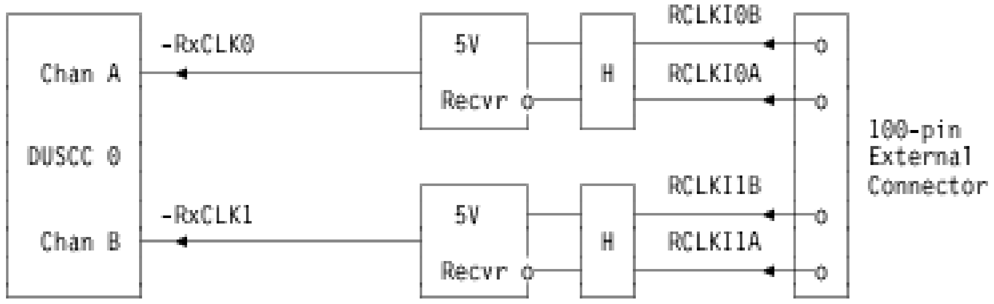


Figure 13-11. RxCLK Block Diagram

Request to Send

The following diagram illustrates the Request to Send (RTS) signal.

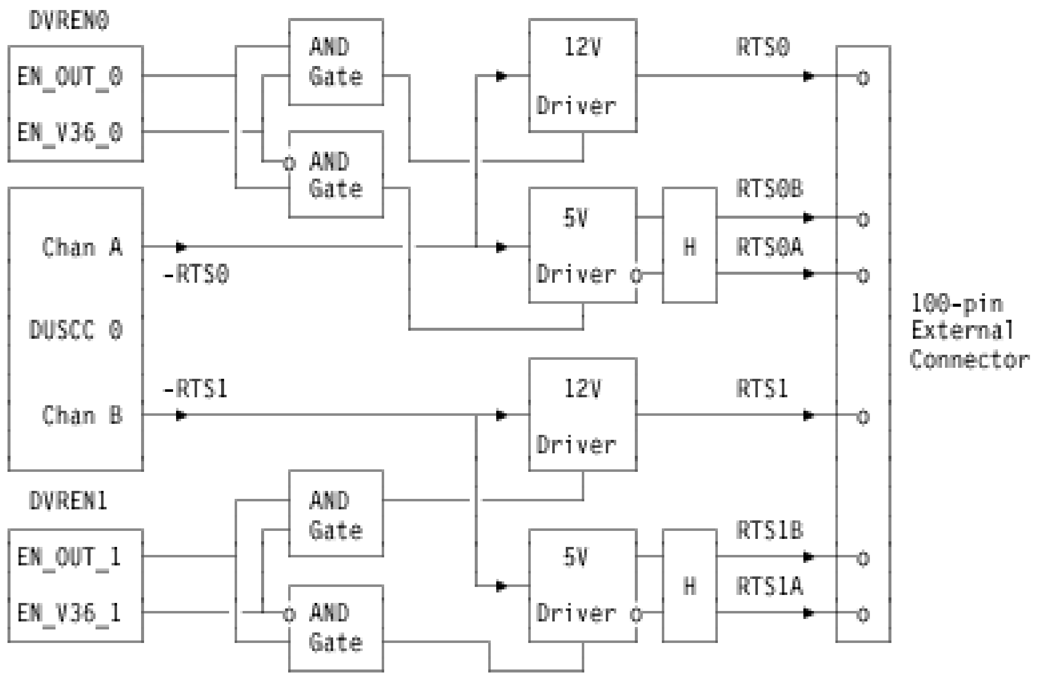


Figure 13-12. RTS Block Diagram

Clear to Send and Carrier Detect

The following diagram is illustrated for Clear to Send (CTS), but it also applies to the Carrier Detect (CD) signal—with the exception of the homologation circuitry, which is not present on the CD circuitry..

The “B” inputs of these signals are pulled to +5V through 10K resistors. The “A” inputs are pulled to GND through 10K resistors.

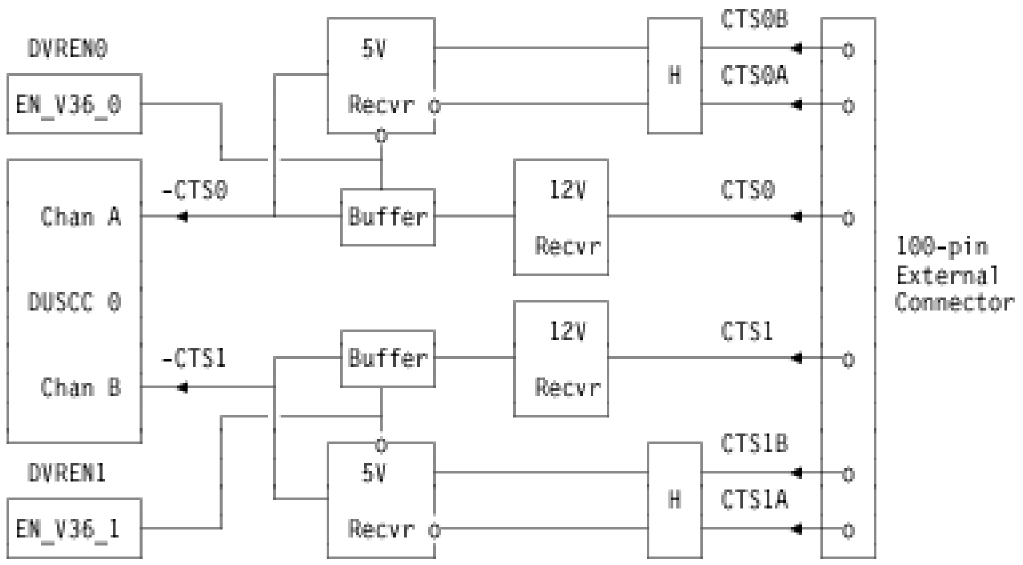


Figure 13-13. CTS Block Diagram

Data Terminal Ready

The following diagram illustrates the Data Terminal Ready (DTR) signal.

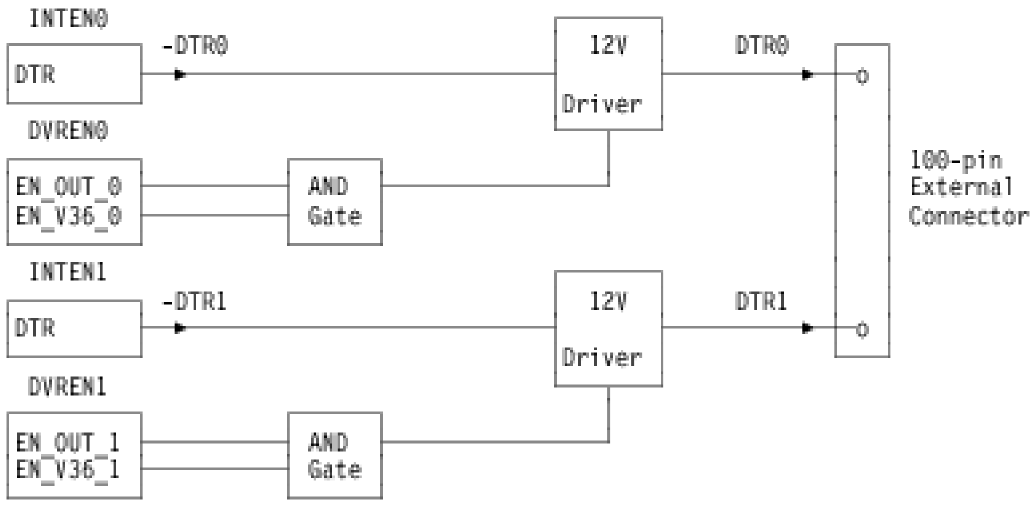


Figure 13-14. DTR Block Diagram

Data Set Ready

The following diagram illustrates the Data Set Ready (DSR) signal.

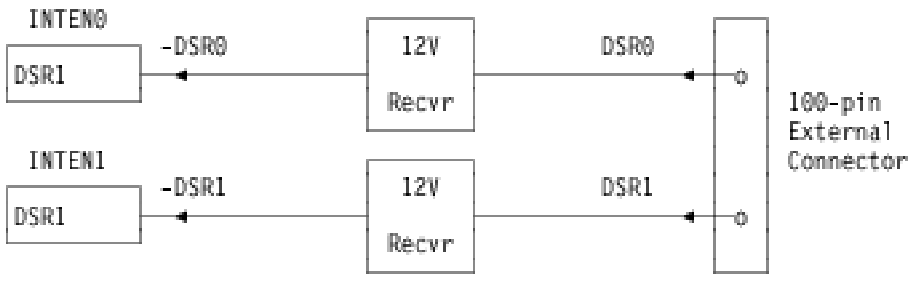


Figure 13-15. DSR Block Diagram

X.21 State Detection

This section describes circuit operation during X.21 state changes. Figure 13-16 shows the block diagram.

The function described in this section is implemented in the hardware on the 4-Port Multi-Interface AIB in an FPGA (Field Programmable Gate Array) device contained on the card. The registers and controls of this FPGA are detailed under *X.21 Pattern Detect Registers* on page 285 and *X.21 State Detection Circuit Description* on page 242.

The appropriate software responses are protocol dependent and are the user's responsibility.

Block Diagram

The following block diagram illustrates the X.21 State Detection circuitry..

- The letter *n* represents the port number (0, 1, 2 or 3).
- A detailed explanation of the operation of the X.21 State Detection circuitry follows.

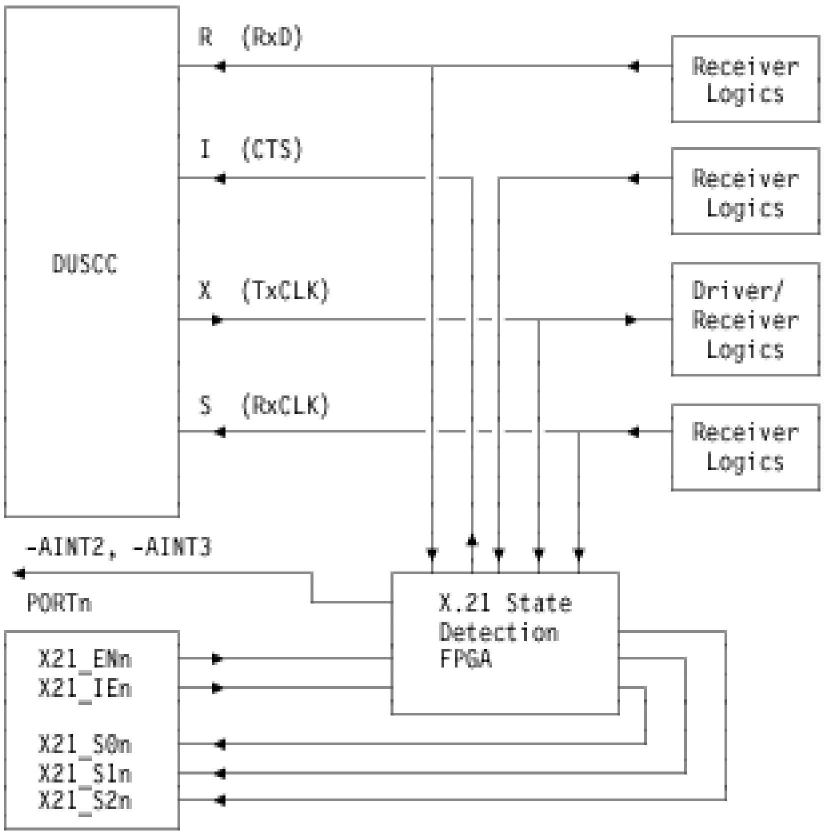


Figure 13-16. X.21 State Detection Block Diagram

X.21 State Detection Circuit Description

This section describes the circuit implemented in the X.21 State Detection FPGA on the 4-Port Multi-Interface AIB.

The CCITT X.21 recommendation defines a steady state condition of the I (CTS) signal as one that lasts for at least 16 bit times.

The hardware can be programmed to intercept the I (CTS) input and hold it in its present state until it is held in the opposite state for 16 S (RxCLK) pulses, and is steady state. This interception is controlled by the I/-CTSn select bit (bit 2 of the X21ENn register).

A state condition is defined by the CCITT to be steady if it lasts 16 contiguous bit intervals. The following signal conditions are decoded by hardware:

- All 1s on R (RxD) during 16 S (RxCLK) pulses and I (CTS) in the OFF state.
- All 0s on R (RxD) during 16 S (RxCLK) pulses and I (CTS) in the OFF state.
- Alternating 0s and 1s on R (RxD) during 16 S (RxCLK) pulses ('010101...' or '101010...') and I (CTS) in the OFF state.

Three status bits are used by the hardware to report the detection of the preceding states:

- X21_SPn(3–0)

  0000 = pattern not detected

0001 = pattern of 16 0s detected

0010 = pattern of 16 1s detected

0011 = pattern of 16 alternating 0s and 1s detected

- X21_OVn — 16 alternating 0s and 1s

  0 = no overflow

  1 = overflow — at least one pattern was detected before X21_SPn was cleared to not detected from the previous detection.

  Where *n* is the port number (0–3).

The X21_SPn bits report that a valid pattern was detected for 16 bit times after the detect enable (X21_ENn) was active (1). The detection hardware samples signals during the on-to-off transition of S (RxCLK). It remains in that state until it is cleared or another pattern is detected.

The status bits are cleared by writing a 1 to any of the status bits or by turning off the detect enable (X21_EN) for that port.

If the status is cleared before the pattern that was detected ends, the circuitry will not report another occurrence of a pattern until it ends and then it (or another pattern) is detected for 16 bit times. (Therefore only one pattern is reported, no matter how long it is held active.)

The detection counter is **not** reset by a write to these status bits; so, if a pattern is in progress on the port, the pattern detection will not be affected by clearing the status.

If the interrupt enable (X21_IEn) is active (1), the hardware activates an AIB interrupt when one of the patterns is detected.

AIB Interrupt 2 will be activated for a state detected on Port 0 or 1 (Channel A or B of DUSCC 0). AIB Interrupt 3 will be activated for a state detected on Port 2 or 3 (Channel A or B of DUSCC 1).

The interrupt is cleared when its status bits are cleared or when the interrupt is disabled by writing a 0 to the interrupt enable (X21_IE)..

These interrupts are shared with the DSR interrupt for the same ports.

The registers associated with this circuitry are described under *X.21 Pattern Detect Registers* on page 285.

Quiescent Phase

Figure 13-17 through Figure 13-19 illustrate the quiescent phase detection states.

Detection of Remote Ready

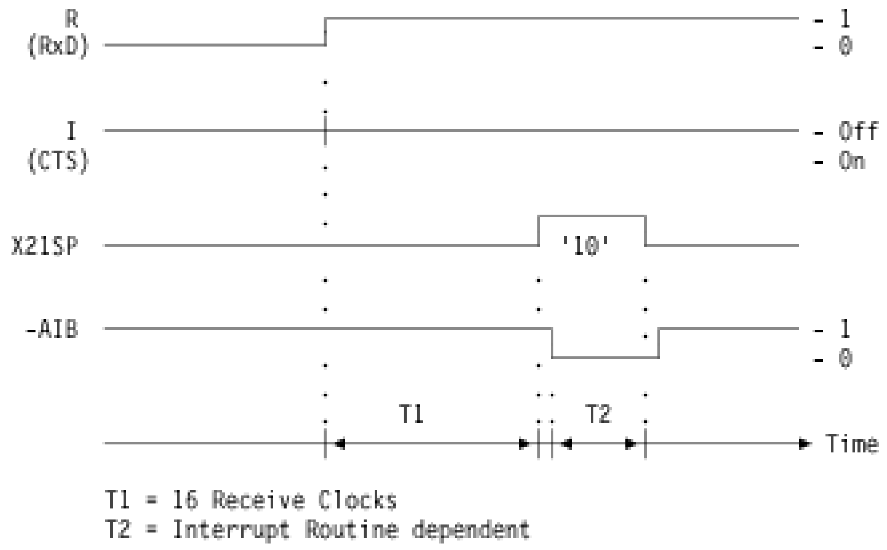


Figure 13-17. Detection of Remote Ready

Detection of Remote Uncontrolled Not Ready

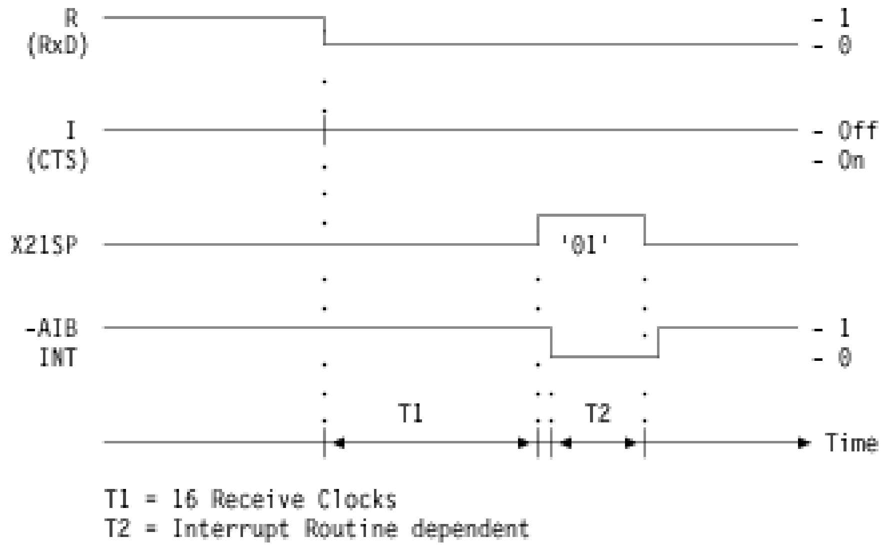


Figure 13-18. Detection of Remote Uncontrolled Not Ready

Detection of Remote Controlled Not Ready:

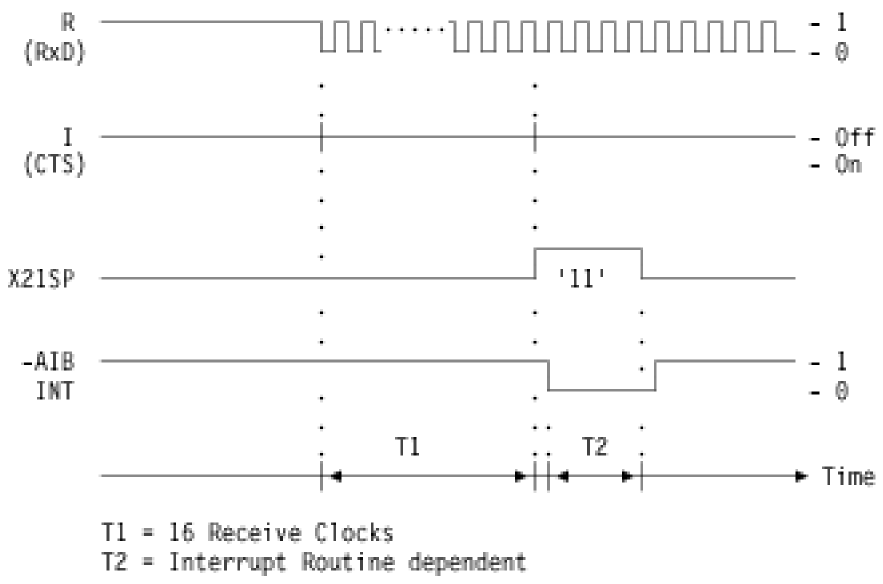


Figure 13-19. Detection of Remote Controlled Not Ready

Call Control Phase for Data Terminal Equipment (DTE)

This section applies to a port configured as a DTE when it is connected to a Circuit-Switched Service. Figure 13-20 through Figure 13-21 illustrate the circuitry.

Detection of DCE Connection in Progress

The data circuit-terminating equipment (DCE) connection in progress state is characterized by I=Off and R=1, which is the same pattern as DCE Ready. This is a case where it is necessary to detect the same pattern twice in a row.

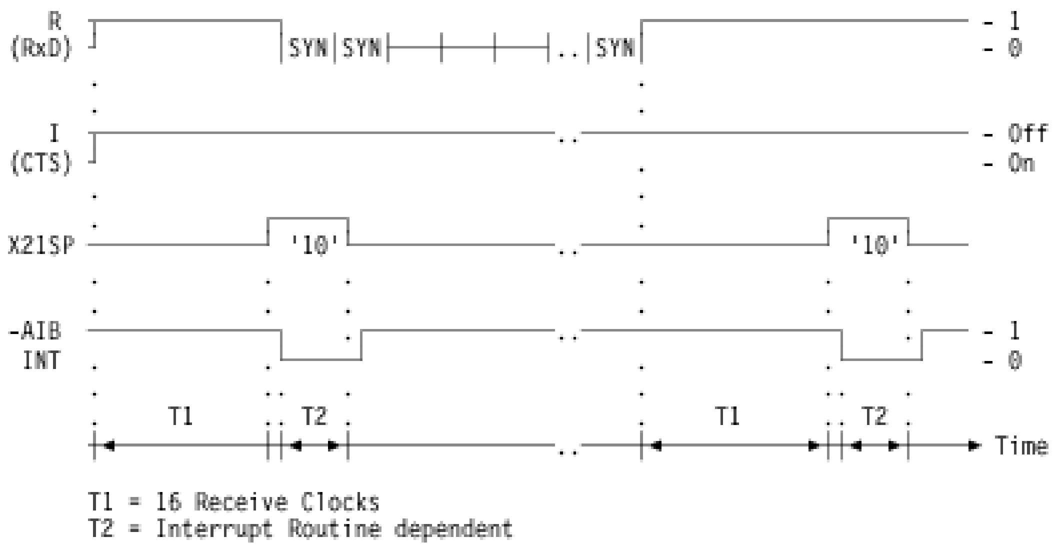


Figure 13-20. Detection of DCE Connection in Progress

Detection of DCE Ready for Data

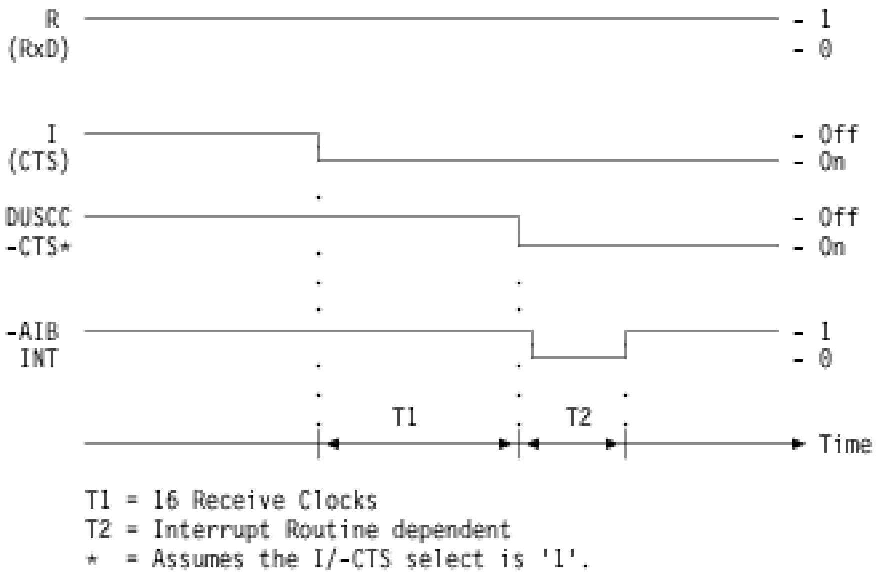


Figure 13-21. Detectino of DCE Ready for Data

Call Control Phase for Data Circuit-Terminating Equipment (DCE)

This section applies to a port configured as a DCE when it is connected to a Circuit-Switched Service.

Incoming Call (Calling DCE)

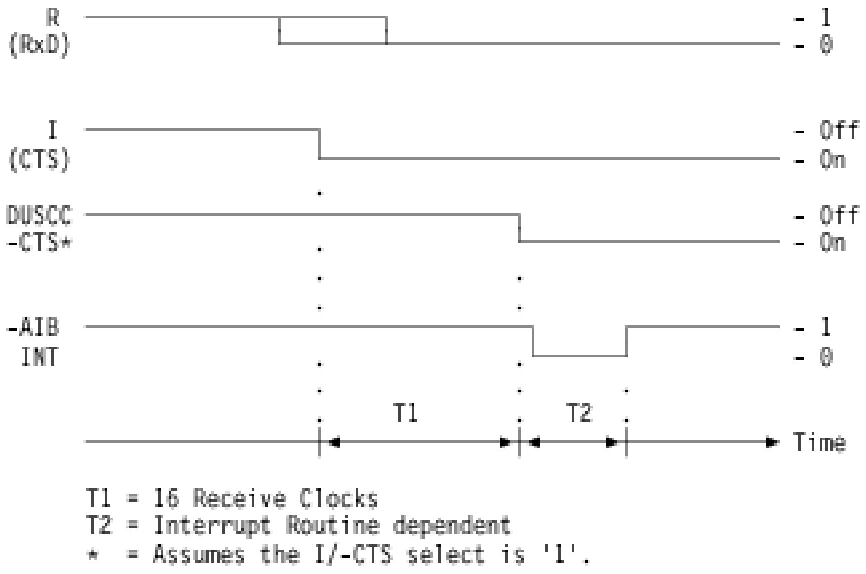
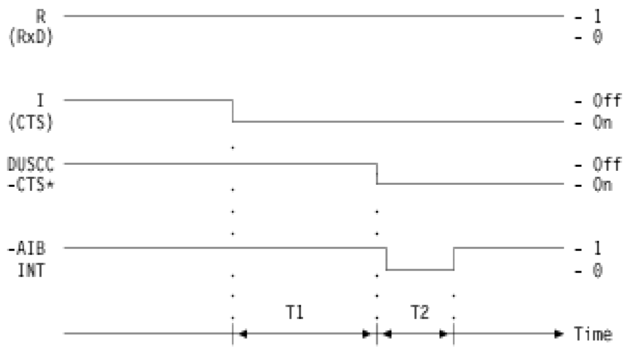


Figure 13-22. Detection of DTE Call Request

Outgoing Call (Calling DCE)



T1 = 16 Receive Clocks
T2 = Interrupt Routine dependent
* = Assumes the I/-CTS select is '1'.

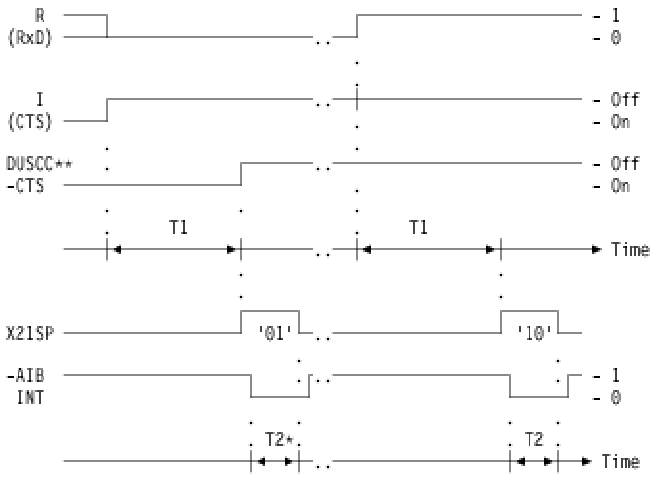
Figure 13-23. Detection of DTE Call Accepted

Clearing Phase

The following detections are valid for a Circuit-Switched Service.

Detection of Remote Clear

The timing, illustrated in Figure 13-24, applies to the detection of either Remote Clear Indication or Remote Clear Confirmation.



T1 = 16 Receive Clocks
T2 = Interrupt Routine dependent
* = Two interrupts are generated at this time (if enabled).
One for 'I' transition and one for the pattern detected.
** = Assumes the I/-CTS select is '1'.

Figure 13-24. Detection of Remote Clear

AIB Cable Assemblies

This section contains the following information about the AIB cable assemblies:

- Cable configurations

- D-shell connectors
- User cables.

Cable Configurations

Four cable assemblies are available as options to support the 4-Port Multi-Interface AIB. The cable assemblies convert the 100-pin, female D-shell connector (J1) on the AIB to four 15, 25, or 37-pin, male D-shell connectors for external cabling (P0–P3).

The cable assemblies are 1.8 meter (6 feet) long. They have a 100-pin, male D-shell connector (P4) on the end that attaches to the connector on the AIB—as illustrated in Figure 13-25. Four cable sections extend to D-shell connectors that match industry standards for the electrical interfaces provided by the cable assembly.

Each cable assembly has four ports, supporting a single electrical interface standard (protocol). Four different cable assemblies are provided, as listed in Table 13-11.

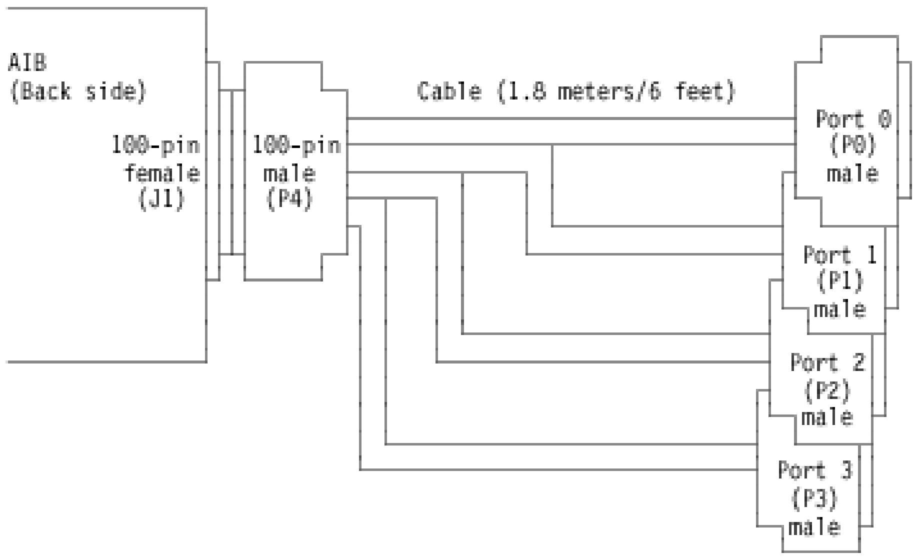


Figure 13-25. AIB Cable Assembly

Table 13-11. D-Shell Connectors Pin Assignment

Cable Assembly	Protocol	Connector Size (Pins)	Pinout
EIA-232-D	EIA-232-D	25	25pin
EIA-530	RS-422-A	25	25pin
ISO 4903	X.21	15	15pin
ISO 4902	V.36	37	37pin

D-Shell Connectors

Table 13-12 lists the pin assignments for the male D-shell connectors of the four AIB cable assemblies. The connector pinouts are shown in Figure 13-26 through Figure 13-28.

The 100-pin connector on the other end of the cable assemblies is shown in Figure 13-3. The pin assignments of the 100-pin connector are listed in Table 13-6.

Pin Assignments (Listed by Cable Assembly)

Table 13-12. D-Shell Connectors Pin Assignments for the AIB Cable

Pin	EIA-232-D	EIA-530 (RS-422-A)	ISO 4903 (X.21)	ISO 4902 (V.36)
1	Shield	Shield	Shield	Shield
2	TxD	TxD(A)	T(A)	
3	RxD	RxD(A)	C(A)	
4	RTS	RTS(A)	R(A)	TxD(A)
5	CTS	CTS(A)	I(A)	TCLKI(A)
6	DSR	S(A)	RXD(A)	
7	GND	GND	X(A)	
8	CD	CD(A)	GND	RCLKI(A)
9	RCLKI(B)	T(B)		
10	CD(B)	C(B)		
11	TCLKO(B)	R(B)		
12	TCLKI(B)	I(B)		
13	CTS(B)	S(B)		
14	TXD(B)	X(B)		
15	TCLKI(A)			
16	RXD(B)			
17	RCLKI(A)	TCLKO(A)		
18				
19	RTS(B)	GND		
20	DTR	DCE Rtn (GND)		
21				
22	TXD(B)			
23	TCLKI(B)			
24	TCLKO(A)	RXD(B)		
25	RTS			
26	RCLKI(B)			
27	CTS			
28				
29	DSR			
30	DTR			
31	CD			
32				
33				
34				
35	TCLKO(B)			
36				
37	DTE Rtn (GND)			

Connector Pinouts

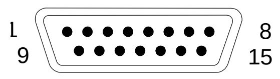


Figure 13-26. 15-Pin D-Shell Connector

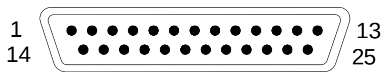


Figure 13-27. 25-Pin D-Shell Connector

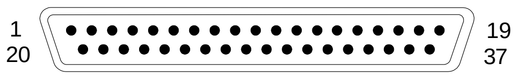


Figure 13-28. 37-Pin D-Shell Connector

User Cables

Maximum Length Supported

For data transfer at speeds up to 64K bps, the maximum cable length from the system is 122 meters (440 feet).

For data transfer at speeds up to 2.048M bps, the maximum cable length from the system is 7 meters (21 feet).

Recommendations

The following cabling recommendations apply for all cables that are to be constructed. Correct operation of the interface depends on several factors that should be taken into consideration during installation.

- Unshielded twisted-pair wire should be adequate for most low-noise installations.
- In areas of high electrical noise, consider using shielded twisted-pair cable to increase noise immunity. Shielded twisted-pair cable also helps eliminate interference from the co-processor platform.
- Metal D-shell connectors should be used to terminate the shield and provide a low-impedance path to ground for noise.

Characteristics

Typical physical characteristics for this cable are:

- 24 AWG, copper conductor, twisted-pair telephone cable, approximately 100 ohms .
- DC resistance (single conductor) is 23.7 ohms/1000 meters.
- Shunt capacitance is 16 pF/foot.

AIB Wrap Plugs

The following tables provide information about the 4-Port Multi-Interface AIB’s 100-pin wrap plug and the cable wrap plugs.

100-Pin Wrap Plug

Table 13-13 lists the signal connections inside the AIB 100-pin wrap plug.

The cable ID of the 100-pin wrap plug is 0 (zero).

Table 13-13. AIB 100-Pin Wrap Plug

Wrapped Signals	Port 0 Pins	Port 1 Pins	Port 2 Pins	Port 3 Pins
TxD(A) - RxD(A)	1 - 52	9 - 58	15 - 64	21 - 95
TxD(B) - RxD(B)	27 - 76	34 - 82	41 - 88	47 - 70
TxD - RxD	5 - 80	12 - 86	18 - 92	24 - 99
RTS(A) - CTS(A) - TCLKI(A)	3 - 30 - 53	11 - 61 - 59	17 - 67 - 65	23 - 98 - 96
RTS(B) - CTS(B) - TCLKI(B)	29 - 4 - 77	36 - 85 - 83	43 - 91 - 89	49 - 73 - 71
TCLKO(A) - RCLKI(A) - CD(A)	2 - 54 - 55	10 - 60 - 33	16 - 66 - 40	22 - 97 - 20
TCLKO(B) - RCLKI(B) - CD(B)	28 - 78 - 79	35 - 84 - 8	42 - 90 - 14	48 - 72 - 46
RTS - CTS - CD	31 - 32 - 81	37 - 39 - 62	68 - 45 - 94	74 - 51 - 100
DTR - DSR	7 - 57	13 - 63	19 - 69	26 - 50
CABLE IDs - GND	87 - 38, 56 - 6, 75 - 25, 93 - 44			

Cable Wrap Plugs

Table 13-14 lists all wrapped signal connections inside the wrap plugs for connectors P0-P3 of each AIB cable.

Table 13-14. Wrap Plugs for AIB Cable Connectors P0-P3

Wrapped Signals	EIA-232-D	EIA-530	ISO 4903	ISO 4902

TxD(A)
RxD(A)
TCLKI(A)
	2
	3
		2
		4
			4
			6
			5	

TxD(B)
RxD(B)
TCLKI(B)
	14
	16
		9
		11
			22
			24
			23	

TxD RxD	2 3			

TCLKO(A)
RCLKI(A)
CD(A)
	24
	17
	8
		7
		6
			17
			8	

TCLKO(B)
RCLKI(B)
CD(B)
	11
	9
	10
		14
		13
			35
			26	

RTS(A)
CTS(A)
TCLKI(A)
	4
	5
	15
		3
		5		

Table 13-14. Wrap Plugs for AIB Cable Connectors P0-P3

Wrapped Signals	EIA-232-D	EIA-530	ISO 4903	ISO 4902
RTS(B)	19	10		
CTS(B)	13	12		
TCLKI(B)	12			
RTS	4	25		
CTS	5	27		
CD	8	31		
(NO CONNECT)	15			
DTR	20	30		
DSR	6	29		
(NO CONNECT)	22			
(NO CONNECT)	23			
(NO CONNECT)	17			
(NO CONNECT)	24			

Software Implementation

Co-Processor Platform Resources Used by the AIB

This section contains information about the co-processor platform resources used by the 4-Port Multi-Interface AIB:

- Interrupts and direct memory access (DMA)
- AIB addressable memory and registers

Interrupts and Direct Memory Access (DMA)

The interrupt and DMA assignments for the devices on the 4-Port Multi-Interface AIB are listed in Table 13-15.

The interrupt mode for the AIB on the co-processor platform must be set to Vectored mode. Interrupts from the AIB can be masked by the co-processor platform in the enable/ detect register (EDR), and by the CFE Local Bus/AIB Interface Chip module in the interrupt mask register (IMR).

The AIB ROM enables interrupts in the EDR, but leaves them masked in the IMR.

AIB interrupts 0 and 1 can be programmed in the CFE Local Bus/AIB Interface Chip module to generate default values of Intel 80960-type interrupt vectors, or interrupt acknowledge cycles can be generated on the DCL bus to fetch vectors from the DUSCCs. The CFE Local Bus/AIB Interface Chip module should be programmed to generate the interrupt acknowledge cycles. Furthermore, the CFE Local Bus/AIB Interface Chip should be set up to provide two INTA pulses for the DUSCCs, and the DUSCCs should be programmed to respond to a 2-pulse interrupt acknowledge.

The DMA on the DUSCCs should be set for duplex, single-address mode, if used.

For details on setting these options, see *Programming Restrictions* on page 289.

Table 13-15. 4-Port Multi-Interface AIB Interrupts and DMA

Device	AIB Interrupt	DMA Level	Interrupt Vector (Dec)	Comments
DUSCC 0	0	- 0 1 2 3	207 - 168 71 70 69 68	Supports Ports 0 and 1 –Port 0 Rcv –Port 1 Rcv –Port 0 Xmt –Port 1 Xmt
DUSCC 1	1	- 4 5 6 7	167 - 128 67 66 65 64	Supports Ports 2 and 3 - Port 2 Rcv - Port 3 Rcv - Port 2 Xmt - Port 3 Xmt
Control FPGA	2		120	DSR for Ports 0 and 1
Control FPGA	3		112	DSR for Ports 2 and 3
X.21 Detect FPGA	2		120	State change for Ports 0 and 1
X.21 Detect FPGA	3		112	State change for Ports 2 and 3

AIB Addressable Memory and Registers

The memory map of all memory and registers on the 4-Port Multi-Interface AIB that are addressable from the co-processor platform follows. These addressable devices are grouped by functions as follows:

- AIB Read-Only Memory (ROM)
- DCL Bus Devices
  - FPGA Status and Control Registers
  - DUSCC Registers
- CFE Local Bus/AIB Interface Chip Registers
  - DCL DMA Channels (0–7)
  - DCL Interrupt Control
  - DCL Bus Control
  - CFE Local Bus/AIB Interface Chip Miscellaneous Registers

AIB ROM

The 4-Port Multi-Interface AIB ROM is an 8-bit (1 byte) wide device that is connected to the co-processor platform ROM bus. It should be accessed using byte-type instructions. (For example, the processor on the RadiSys ARTIC960 Co-Processor Platform is an Intel 80960 and is accessed using byte load (ldob) and byte store (stob) instructions.)

The AIB ROM provides:

- Card identification

- Self-test of card features
- Initialization of card features

The details of this information are described under *AIB ROM Description* on page 296.

The actual memory map of the AIB ROM is dependent on the co-processor platform on which the AIB is installed. For example, Table 13-16 lists the location of the AIB ROM in the 80960's memory on the RadiSys ARTIC960 Co-Processor Platform.

Table 13-16. AIB ROM Memory Map on the RadiSys ARTIC960 Co-Processor Platform

Device	CS Range (Bytes)	Processor Address Range (Hex)
Flash ROM	32K	0C00 0000 – 0C00 8FFF[1]

1 This is the primary address range for the ROM. The ROM will be selected throughout the region 0C01 0000 – 0DFF FFFF as well.

DCL Bus Devices

All devices on the DCL Bus are 8-bits (1 byte) wide and should be accessed using byte-type instructions. (For example, the processor on the RadiSys ARTIC960 Co-Processor Platform is an Intel 80960 and is accessed using byte load (ldob) and byte store (stob) instructions.)

Only 19 address bits are significant on the DCL Bus. The upper 13 address bits from the CFE Bus (A31–19) are used by the CFE Local Bus/AIB Interface Chip module to decode the DCL Bus address space, and are passed to the DCL Bus as 0s.

Table 13-17 lists the assignments given to the DCL Chip Selects for the devices on the 4-Port Multi-Interface AIB DCL Bus.

Table 13-17. DCL Bus Memory Map

Device	CFE Address Range (Hex)	Chip Select	CS Range (Bytes)	DCL Address Range (Hex)	Reference Section
X.21 Pattern Detect FPGA	1FF2 0000 – 1FF2 3FFF	-CS0	16K	00000 – 03FFF	285
DUSCC 0	1FF0 0000 – 1FF0 1FFF	-CS1	8K	00000 – 01FFF	256
DUSCC 1	1FF0 2000 – 1FF0 3FFF	-CS2	8K	00000 – 01FFF	256
Control FPGA (Ports 0 thru 3)	1FF1 0000 – 1FF1 3FFF	-CS3	16K	00000 – 03FFF	282

FPGA Registers

Table 13-18 lists the assignments given to the FPGA registers. These registers are described under *FPGA Registers* on page 282.

Table 13-18. FPGA Memory Map

Register	CFE Address (Hex)	DCL Chip Select	DCL Address (Hex)	See Page
DVREN0	1FF1 0000	-CS3	00000	283
DVREN1	1FF1 1000	-CS3	01000	283
DVREN2	1FF1 2000	-CS3	02000	283
DVREN3	1FF1 3000	-CS3	03000	283
INTEN0	1FF1 0800	-CS3	00800	284
INTEN1	1FF1 1800	-CS3	01800	284
INTEN2	1FF1 2800	-CS3	02800	284
INTEN3	1FF1 3800	-CS3	03800	284
X21EN0	1FF2 0800	-CS0	00800	286
X21EN1	1FF2 1800	-CS0	01800	284
X21EN2	1FF2 2800	-CS0	02800	284
X21EN3	1FF2 3800	-CS0	03800	284
X21ST0	1FF2 0000	-CS0	00000	284
X21ST1	1FF2 1000	-CS0	01000	284
X21ST2	1FF2 2000	-CS0	02000	284
X21ST3	1FF2 3000	-CS0	03000	284

DUSCC Register

Table 13-19 and Table 13-20 list the DUSCC registers. The Signetics SC26C562 DUSCC registers are described in the *Signetics SC26C562/SC68C562 CMOS DUSCC User's Guide*. See also *Hardware Notes* on page 288 and *Signetics SC26C562 DUSCC* on page 290.

Table 13-19. CMOS DUSCC DCL Memory Map (Part 1 of 2)

			Register (See Note)	
DUSCC 0 CFE Address (CS1)	DUSCC 1 CFE Address (CS2)	DCL Address (Both)	(A7 = 0)	(A7 = 1)
1FF0 0000	1FF0 2000	00000	CMR1 (Chan A)	
1FF0 0080	1FF0 2080	00080	CMR2 (Chan A)	
1FF0 0100	1FF0 2100	00100	S1R (Chan A)	IER1 (Chan A)
1FF0 0180	1FF0 2180	00180	S2R (Chan A)	IER2 (Chan A)
1FF0 0200	1FF0 2200	00200	TPR (Chan A)	
1FF0 0280	1FF0 2280	00280	TTR (Chan A)	IER3 (Chan A)
1FF0 0300	1FF0 2300	00300	RPR (Chan A)	
1FF0 0380	1FF0 2380	00380	RTR (Chan A)	TRCR (Chan A)
1FF0 0400	1FF0 2400	00400	CTPRH (Chan A)	
1FF0 0480	1FF0 2480	00480	CTPRL (Chan A)	
1FF0 0500	1FF0 2500	00500	CTCR (Chan A)	

Table 13-19. CMOS DUSCC DCL Memory Map (Part 1 of 2)

			Register (See Note)	
DUSCC 0 CFE Address (CS1)	**DUSCC 1 CFE Address (CS2)**	**DCL Address (Both)**	**(A7 = 0)**	**(A7 = 1)**
1FF0 0580	1FF0 2580	00580	OMR (Chan A)	
1FF0 0600	1FF0 2600	00600	CTH (Chan A)	
1FF0 0680	1FF0 2680	00680	CTL (Chan A)	
1FF0 0700	1FF0 2700	00700	PCR (Chan A)	RFLR (Chan A)
1FF0 0780	1FF0 2780	00780	CCR (Chan A)	
1FF0 0800	1FF0 2800	00800	TxFIFO (Chan A)	
1FF0 0A00	1FF0 2A00	00A00	RxFIFO (Chan A)	
1FF0 0C00	1FF0 2C00	00C00	RSR (Chan A)	
1FF0 0C80	1FF0 2C80	00C80	TRSR (Chan A)	
1FF0 0D00	1FF0 2D00	00D00	ICTSR (Chan A)	
1FF0 0D80	1FF0 2D80	00D80	GSR	
1FF0 0E00	1FF0 2E00	00E00	IER (Chan A)	FTLR (Chan A)
1FF0 0E80	1FF0 2E80	00E80	REA / CID	
1FF0 0F00	1FF0 2F00	00F00	IVR	TRMSR (Chan A)
1FF0 0F80	1FF0 2F80	00F80	ICR	TELR (Chan A)

A7 is an address extension that is internal to the DUSCC; it is:

- Set to 1 by writing to the SEA Register.
- Cleared to 0 by writing to the REA Register.

The value of the data written to the SEA Register and the REA Register is ignored.

Table 13-20. CMOS DUSCC DCL Memory Map (Part 2 of 2)

			Register (See Note)	
DUSCC 0 CFE Address (CS1)	**DUSCC 1 CFE Address (CS2)**	**DCL Address (Both)**	**(A7 = 0)**	**(A7 = 1)**
1FF0 1000	1FF0 3000	01000	CMR1 (Chan B)	
1FF0 1080	1FF0 3080	01080	CMR2 (Chan B)	
1FF0 1100	1FF0 3100	01100	S1R (Chan B)	IER1 (Chan B)
1FF0 1180	1FF0 3180	01180	S2R (Chan B)	IER2 (Chan B)
1FF0 1200	1FF0 3200	01200	TPR (Chan B)	
1FF0 1280	1FF0 3280	01280	TTR (Chan B)	IER3 (Chan B)
1FF0 1300	1FF0 3300	01300	RPR (Chan B)	
1FF0 1380	1FF0 3380	01380	RTR (Chan B)	TRCR (Chan B)
1FF0 1400	1FF0 3400	01400	CTPRH (Chan B)	
1FF0 1480	1FF0 3480	01480	CTPRL (Chan B)	
1FF0 1500	1FF0 3500	01500	CTCR (Chan B)	
1FF0 1580	1FF0 3580	01580	OMR (Chan B)	
1FF0 1600	1FF0 3600	01600	CTH (Chan B)	
1FF0 1680	1FF0 3680	01680	CTL (Chan B)	
1FF0 1700	1FF0 3700	01700	PCR (Chan B)	RFLR (Chan B)
1FF0 1780	1FF0 3780	01780	CCR (Chan B)	

Table 13-20. CMOS DUSCC DCL Memory Map (Part 2 of 2)

			Register (See Note)	
DUSCC 0 CFE Address (CS1)	**DUSCC 1 CFE Address (CS2)**	**DCL Address (Both)**	**(A7 = 0)**	**(A7 = 1)**
1FF0 1800	1FF0 3800	01800	TxFIFO (Chan B)	
1FF0 1A00	1FF0 3A00	01A00	RxFIFO (Chan B)	
1FF0 1C00	1FF0 3C00	01C00	RSR (Chan B)	
1FF0 1C80	1FF0 3C80	01C80	TRSR (Chan B)	
1FF0 1D00	1FF0 3D00	01D00	ICTSR (Chan B)	
1FF0 1D80	1FF0 3D80	01D80	GSR	
1FF0 1E00	1FF0 3E00	01E00	IER (Chan B)	FTLR (Chan B)
1FF0 1E80	1FF0 3E80	01E80	SEA	
1FF0 1F00	1FF0 3F00	01F00	IVRM	TRMSR (Chan B)
1FF0 1F80	1FF0 3F80	01F80	MRR	TELR (Chan B)

7 is an address extension that is internal to the DUSCC; it is:

- Set to 1 by writing to the SEA Register.
- Cleared to 0 by writing to the REA Register.

The value of the data written to the SEA Register and the REA Register is ignored.

CFE Local Bus/AIB Interface Chip Registers

Each CFE Local Bus/AIB Interface Chip register is 4-byte aligned in the address space and should be accessed using word-type instructions. (For example, the processor on the RadiSys ARTIC960 Co-Processor Platform is an Intel 80960 and is accessed using word load (ld) and word store (st) instructions.)

All registers, when read, will return 0 values in those bits that are undefined for a specific register.

At the end of each register description section, the register's "Reset Conditions" are shown. Reset Command refers to the value that the register bits assume when a DMA channel is reset using either the DMA Channel Command Register (DCCR) or the Global DMA Command Register (GDCR).

A U in the Reset Conditions section means undefined. An S means the value stays the same as before the reset command.

Table 13-31 through Table 13-32 list the bus address/access type of the CFE Local Bus/ AID Interface Chip registers.

Table 13-21. CFE Local Bus/AIB Interface Chip DMA Registers

Abbreviation	Register Name	CFE Bus Address (Hex)	Accses Type	See Page
AIB_ADDR 1/2	DMA Channel x AIB_ADDR 1/2	1FF8 x000	r/w	260
AIB_OP1	DMA Channel x AIB_OP1 Data	1FF8 x004	r/w	261
AIB_OP2	DMA Channel x AIB_OP2 Data	1FF8 x008	r/w	261
CCR	DMA Channel x Channel Control	1FF8 x018	r/w	262
CPR	DMA Channel x Chain Pointer	1FF8 x014	r/w	265

Table 13-21. CFE Local Bus/AIB Interface Chip DMA Registers

Abbreviation	Register Name	CFE Bus Address (Hex)	Accses Type	See Page
DCCR	DMA Channel *x* Command	1FF8 x020	r/w	266
DFRC	DMA Channel *x* FIFO Residual Count	1FF8 x02C	ro	266
DISR	DMA Channel *x* Interrupt Status	1FF8 x028	ro	267
GDCR	Global DMA Command	1FF8 8000	r/w	268
LBCR	CFE Local Bus Configuration	1FF8 8008	r/w	269
MPR	DMA Channel *x* Memory Pointer	1FF8 x00C	r/w	269
TQC	DMA Channel *x* Transfer Count	1FF8 0x10	r/w	270
	Reserved	1FF8 x01C		
	Reserved	1FF8 x024		
	Reserved	1FF8 8004		
	Reserved	1FF8 800C		

x = DMA channel number 0 to 7.

Table 13-22. CFE Local bus/AIB Interface Chip Interrupt Registers

Abbreviation	Register Name	CFE Bus Address (Hex)	Access Type	See Page
EOI0	ABI INT0 End-of-Interrupt	1FF8 9000	wo (command)	271
EOI1	AIB INT1 End-of-Interrupt	1FF8 A000	wo (command)	271
IIR	Interrupt Initialization	1FF8 8010	r/w	271
IMR	Interrupt Mask	1FF8 8014	r/w	274
ISR	Interrupt Status	1FF8 8018	ro	275

Table 13-23. CFE Local Bus/AIB Interface Chip DCL Registers

Abbreviation	Register Name	CFE Bus Address (Hex)	Access Type	See Page
ACSR	DCL Command/Status	1FF8 C000	r/w	275
CSD0	Chip Select Definition 0	1FF8 B000	r/w	276
CSD1	Chip Select Definition 1	1FF8 B004	r/w	276
CSD2	Chip Select Definition 2	1FF8 B008	r/w	276
CSD3	Chip Select Definition 3	1FF8 B00C	r/w	276
CSD4	Chip Select Definition 4	1FF8 B010	r/w	276
DAPW0	DMA Acknowledge Pulse Width 0	1FF8 B030	r/w	279
DAPW1	DMA Acknowledge Pulse Width 1	1FF8 B034	r/w	279
DAPW2	DMA Acknowledge Pulse Width 2	1FF8 B038	r/w	279
DAPW3	DMA Acknowledge Pulse Width 3	1FF8 B03C	r/w	279
DAPW4	DMA Acknowledge Pulse Width 4	1FF8 B040	r/w	279
DAPW5	DMA Acknowledge Pulse Width 5	1FF8 B044	r/w	279
DAPW6	DMA Acknowledge Pulse Width 6	1FF8 B048	r/w	279
DAPW7	DMA Acknowledge Pulse Width 7	1FF8 B04C	r/w	279

Table 13-24. CFE Local Bus/AIB Interface Chip Miscellaneous Registers

Abbreviation	Register Name	CFE Bus Address (Hex)	Access Type	See Page
LER	LED Enable	1FF8 D004	r/w	281
PDR	Presence Detect	1FF8 D000	ro	281

Software Interface

This section contains the following information about the software interface.

- CFE Local Bus/AIB Interface Chip registers
- FPGA registers

CFE Local Bus/AIB Interface Chip Registers

The CFE Local Bus/AIB Interface Chip registers are categorized as follows: DMA, interrupt, DCL, and miscellaneous.

DMA Registers

The following paragraphs describe the DMA registers contained on the CFE Local Bus/ AIB Interface Chip:

- AIB Address 1/2 Register (AIB_ADDR 1/2)
- AIB_OP1 Data Register (AIB_OP1)
- AIB_OP2 Data Register (AIB_OP2)
- Channel Control Register (CCR)
- Chain Pointer Register (CPR)
- DMA Channel Command Registers (DCCR)
- DMA FIFO Residual Count Register (DFRC)
- DMA Interrupt Status Registers (DISR)
- Global DMA Command Register (GDCR)
- CFE Local Bus Configuration Register (LBCR)
- Memory Pointer Register (MPR)
- Transfer Count Register (TQC)

AIB Address 1/2 Register (AIB_ADDR 1/2)

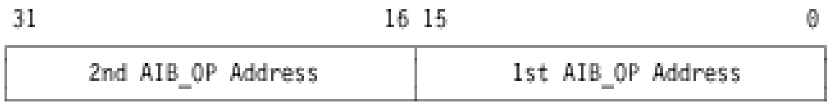
The AIB_ADDR 1/2 register contains two separate 16-bit address fields. Bits 0–15 define the first AIB address of two possible I/O operations that the hardware will perform when a chain event occurs. Bits 16–31 define the second AIB address of two possible I/O operations that the hardware will perform when a chain event occurs.

The first operation (read or write) is directed at the address defined by bits 0–15 of this register. For a write operation, the data is defined by the value in the AIB_OP1 Data register. For read operations, the data read is stored in memory at the CPR address (the first address of the next CDB). Also, because the AIB supports a 19-bit (512KB) address space, devices which use this feature must reside within the lower 64KB of the 512KB AIB address space.

The second operation (read or write) is directed at the address defined by bits 16–31 of this register. This operation is the same as the first except that the data written to the AIB is that contained in the AIB_OP2 Data register, or may be a pre-defined fixed value, as defined in CCR bits 20–21.

Register Format

```
(80960 Address = 1FF8 x000h) read/write; x = channel number
```



Reset Conditions

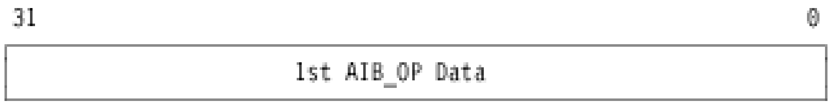
```
Reset: UUUU UUUU UUUU UUUU UUUU UUUU UUUU UUUU
```

AIB_OP1 Data Register (AIB_OP1)

The AIB_OP1 register contains the data field of the first of two operations upon a chain event. When a chain event occurs, the DMA controller can be programmed to perform up to two I/O operations to any DCL Bus address in the lower 64K of the AIB address space. The first operation (read or write) is directed at the address defined by bits 0–15 of the AIB_ADDR 1/2 register. For a write operation, the data is defined by this register.

Register Format

```
(80960 Address = 1FF8 x004h) read/write; x = channel number
```



Reset Conditions

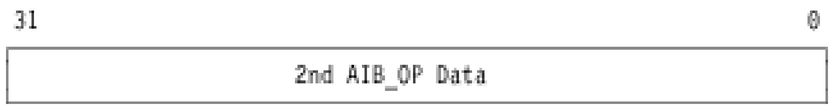
```
Reset: UUUU UUUU UUUU UUUU UUUU UUUU UUUU UUUU
```

AIB_OP2 Data Register (AIB_OP2)

The AIB_OP2 register contains the data field of the second of two operations upon a chain event. When a chain event occurs, the DMA controller can be programmed to perform up to two I/O operations to any DCL Bus address in the lower 64KB of the AIB address space. The second operation (read or write) is directed at the address defined by bits 16–31 of the AIB_ADDR 1/2 register. For a write operation, the data is defined by this register or may be a pre-defined fixed value, as defined by bits 20–21 of the CCR.

Register Format

`(80960 Address = 1FF8 x008h) read/write; x = channel number`



Reset Conditions

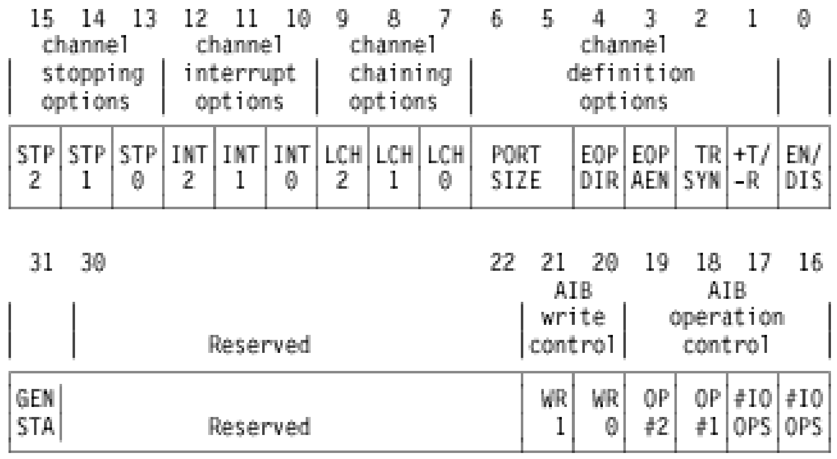
`Reset: UUUU UUUU UUUU UUUU UUUU UUUU UUUU UUUU`

Channel Control Register (CCR)

The CCR register controls the operational personality for a DMA channel.

Register Format

`(80960 Address = 1FF8 x018h) read/write; x = channel number`



Bit Descriptions

- Bit 31: General Purpose Status. This bit can be used for any application-dependent purpose. It has no function internal to the chip. It is updated by I/O writes, and during list chaining fetches.
- Bits 30–22: Reserved. Always program these bits to '0'.
- Bits 21–20: These two bits are encoded to indicate what data will be written to the DCL Bus on the second of two AIB OPs, if the first of the two AIB OPs is a read.

Table 13-25.

B21	B20		
0	0	–	Write the value of AIB_OP2 Data Register
0	1	–	Write the value that was read on first OP
1	0	–	Write the complement of value read on first OP
1	1	–	Reserved

- Bit 19: AIB OP #2, Read or Write. This bit is coupled with bits 16 and 17. It defines the second IO operation to the DCL Bus after a chain event to be a read or write; 0 = read, 1 = write. If the operation is a write, the data written is defined by bits 20 and 21 of the CCR. If the operation is a read, the data will be stored in memory.
- Bit 18: AIB OP #1, Read or Write. This bit is coupled with bits 16 and 17. It defines the first IO operation to the DCL Bus after a chain event to be a read or write; 0 = read, 1 = write. If the operation is a write, the data written is defined by bits of the AIB_OP1 register. If the operation is a read, the data will be stored in memory.
- Bits 17–16: Number of IO operations to DCL Bus upon chain event. These two bits are encoded to provide the DMA controller with the ability to execute up to 2 discrete IO operations to the DCL Bus following the recognition of a chain event. The address to which the first IO operation occurs is defined by bits 0–15 in the AIB_ADDR 1/2 register. The address to which the second IO operation occurs is define by bits 16–31 in the AIB_ADDR 1/2 register.

  00 = No IO operation upon chain event.
  01 = One IO operation upon chain event.
  10 = Two IO operations upon chain event.
  11 = Reserved.
- Bits 15–13: Encoded DMA Channel Stopping Options.

Table 13-26.

STP2	STP1	STP0	Stopping Option
0	0	0	Do not stop
0	0	1	TC=0
0	1	0	EOP
0	1	1	TC=0 “or” EOP
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

When a channel is stopped, the CCR enable/disable bit is reset. Also, a chaining condition takes precedence over a stopping condition if both occur at the same time because the chaining condition causes a new CCR enable/disable bit to be fetched from memory.

- Bits 12–10: Encoded DMA Interrupt Options.

Table 13-27.

INT1	INT2	INT0	Interrupt Option
0	0	0	Disabled
0	0	1	TC=0
0	1	0	EOP
0	1	1	TC=0 “or” EOP
1	0	0	Reserved
1	0	1	Reserved

Table 13-27.

INT1	INT2	INT0	Interrupt Option
1	1	0	Reserved
1	1	1	Reserved

- Bits 9–7: Encoded List Chaining Enabling Options.

Table 13-28.

LCH2	LCH1	LCH0	Chaining Option
0	0	0	Disabled
0	1	1	TC=0
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	N0P

The chaining NOP selection, when detected, immediately causes the next CDB in memory to fetched without processing the CDB containing the NOP.

- Bits 6–5: DMA Port Size. These two bits define the data port size of the DMA device.

Table 13-29.

PS1	PS0	Data Port Size
0	0	8-bit
0	1	16-bit
1	0	32-bit
1	1	Reserved

- Bit 4: End-of-Process Direction Control. If this bit is set to a 1, the EOP signal is an input. If this bit is set to a 0, the EOP signal is an output.

  If the channel is a transmit DMA channel, and this bit is 0, the EOP signal is driven synchronously with the last byte transferred when TC=0. For a transmit channel, if this bit is 1, EOP is an asynchronous input that can causing chaining based on how bits 7–9 are programmed. Asynchronous EOPs must only occur on 16-byte boundaries for Tx DMA.

  If the channel is a receive DMA channel, and this bit is 0, the EOP signal is disabled as an input. For a receive channel, if this bit is 1, EOP is an input that can cause chaining, based on how bits 7–9 are programmed.

- Bit 3: EOP Asynchronous Input Enable. When this bit is set to 1, the CFE Local Bus/AIB Interface Chip is enabled to detect an asynchronous EOP input. When reset, the chip will not detect asynchronous EOP inputs. The CFE Local Bus/AIB Interface Chip is **always** enabled to detect synchronous EOP inputs.

- Bit 2: DMA Transfer Synchronization Option. When this bit is set to 0, all DMA transfers are synchronized by the DREQ input pins of the CFE Local Bus/AIB Interface Chip. The DREQ signal must become active for a DMA transfer to occur. When this bit is set to 1, DMA transfers are not synchronized to the DREQ input pins.

DMA transfers will begin as soon as the CCR bit 0 is enabled and will continue until a channel stopping condition is reached.

- Bit 1: -Receive/+Transmit Indicator. When this bit is set to 0, the channel is servicing receive DMA requests. When this bit is set to 1, the channel is servicing transmit DMA requests. For receive requests, data transfer is FROM DCL Bus TO Local Bus. For transmit requests, data transfer is FROM CFE Local Bus TO DCL Bus.
- Bit 0: +Enable/-Disable Channel. When this bit is set to 0, the channel is disabled to service DMA requests. When this bit is set to 1, the channel is enabled to service DMA requests. This bit can be used to disable the channel at any time. Re-enabling the channel will cause the DMA operation to start where it left off. Also, any programming option that disables the channel will be reflected in this bit. This bit is updated with its corresponding bit during list chaining reads from memory. This bit can also be set/reset via the DCCR. See *DMA Channel Command Registers (DCCR)* on page 266.

This bit or bit 0 of the DCCR registers alone should not be used as a status bit to determine if the channel is disabled, stopped, or reached the end of a chain. Instead, the user should use both bit 31 and bit 0 of the CCR to determine the status of the channel during a linked list chaining.

- The following is an example assuming that bit 31 and bit 0 of the CCR are set in the main CDBs and are reset in the dummy CDBs:

Table 13-30.

Bit 31	Bit 0	CFE Local Bus/AIB Interface Chip status
1	1	Running
1	0	Chaining
0	0	Stopped

Reset Conditions

```
            Reset:  U000 0000 00UU UUUU 0000 0000 0UU1 0UU0
ROM Initialization:  0000 0000 0000 0000 0000 0000 0000 0000
```

Chain Pointer Register (CPR)

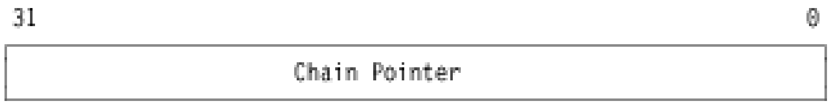
The CPR contains the 32-bit address pointer that points to the memory location where the DMA controller will fetch the CDB for the next buffer when a chain event occurs. The lower two bits of this register should always be programmed to 00 so the chain pointer is 4-byte aligned. The following conditions define a chaining event:

- Terminal count has been reached and list chaining for terminal count is enabled in the CCR.
- An End-of-Process condition has occurred and list chaining for end-of-process is enabled in the CCR.

The CPR should always point to a valid memory location. This location should contain a valid dummy CDB with the channel control set to stop the channel and disable list chaining.

Format

(80960 Address = 1FF8 x014h) read/write; x = channel number



Reset Conditions

```
Reset:  UUUU UUUU UUUU UUUU UUUU UUUU UUUU UUUU
```

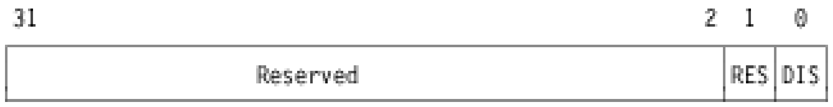
DMA Channel Command Registers (DCCR)

The DCCR supports two commands. One command allows for the bit manipulation of the CCR enable/disable bit. This bit allows CCR bit 0 to be set/reset without affecting the other CCR bits. A second command allows internal logic to be reset to a known state if an error condition causes a channel to stop before completion. A separate command register exists to control each DMA channel.

The command to enable a channel should not be issued with the same write that releases the channel from a reset command.

Register Format

(80960 Address = 1FF8 x020h) read/write; x = channel number



Bit Descriptions

- Bits 31–2: Reserved.
- Bit 1: Reset Channel command. Setting this bit to 1 causes a channel to be reset as shown by the Reset Command for each register. The channel is held in the reset state until the bit is set back to 0.
- Bit 0: Disable Channel command. Bit 0 = 0 defines the command to disable the channel. Bit 0 = 1 defines the command to enable the channel. Bit 0 of the CCR, if read, will reflect a disable command if it is issued through the DCCR.

  When a channel is disabled, any internally pending operations are performed before the channel stops servicing DMA requests.

Reset Conditions

```
             Reset:  0000 0000 0000 0000 0000 0000 0000 0000
ROM Initialization:  0000 0000 0000 0000 0000 0000 0000 0000
```

DMA FIFO Residual Count Register (DFRC)

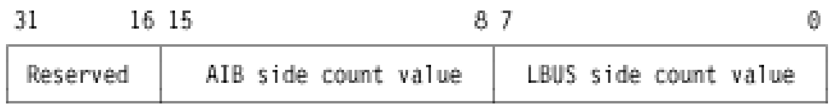
The DFRC register can be read when a DMA channel is stopped or known to be inactive to determine how many bytes of data are in the FIFO. The register contains two 8-bit fields, one of which is the value of the AIB side counter, and the other is the value of the LBUS side counter.

For a transmit DMA channel, the LBUS side counter increments as bytes are loaded into the FIFO from the Local Bus, and the AIB side counter increments as these bytes are sent to the DCL Bus. Software must subtract the AIB side count value from the LBUS side count value to obtain a residual count value of how many bytes are still in the FIFO waiting to be sent to the DCL Bus.

For a receive DMA channel, the AIB side counter increments as bytes are loaded into the FIFO from the DCL Bus, and the LBUS side counter increments as these bytes are sent to the Local Bus. Software must subtract the LBUS side count value from the AIB side count value to obtain a residual count value of how many bytes are still in the FIFO waiting to be sent to the Local Bus.

Register Format

```
(80960 Address = 1FF8 x02Ch) read-only; x = channel number
```



Reset Conditions

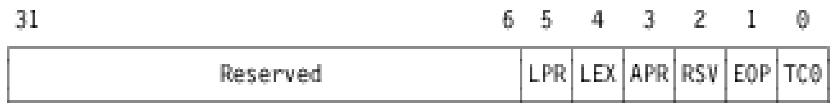
```
Reset:  0000 0000 0000 0000 0000 0000 0000 0000
```

DMA Interrupt Status Registers (DISR)

Each DMA channel has a DMA Interrupt Status register (DISR) associated with it. This status register is cleared when read. Because seven different conditions can cause the interrupt, this means it is possible for multiple conditions to be present when the status register is read. However, because the register is cleared by the read, only one interrupt will be presented to the 80960 processor. Therefore, the interrupt service routine must be capable of servicing all of the pending status bits.

Register Format

```
(80960 Address = 1FF8 x028h) read-only; x = channel number
```



Bit Descriptions

- Bits 31–6: Reserved.
- Bit 5: LPR. This bit, when set, indicates a CFE Local Bus parity error has occurred when the DMA channel was reading data from the Local Bus. In this case, the channel is automatically stopped.
- Bit 4: LEX. This bit, when set, indicates a CFE Local Bus exception error occurred. This happens when a CFE Local Bus slave drives the Exception signal while the CFE

Local Bus/AIB Interface Chip is the CFE Local Bus master. In this case, the CFE Local Bus/AIB Interface Chip DMA channel is automatically stopped.

- Bit 3: APR. This bit, when set, indicates an DCL Bus parity error has occurred when the DMA channel was reading data from the DCL Bus. In this case, the channel is automatically stopped.
- Bit 2: Reserved.
- Bit 1: EOP. This bit, when set, indicates an end-of-process condition has occurred for the DMA channel. This bit is set upon detecting an EOP, regardless of the state of its corresponding interrupt enable bit.
- Bit 0: TC0. This bit, when set, indicates a terminal count condition has been reached for the DMA channel. This bit is set upon detecting a TC, regardless of the state of its corresponding interrupt enable bit.

Reset Conditions

```
             Reset:  0000 0000 0000 0000 0000 0000 0000 0000
ROM Initialization:  0000 0000 0000 0000 0000 0000 0000 0000
```

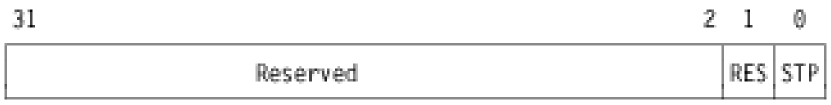
Global DMA Command Register (GDCR)

The GDCR allows two commands to be issued globally to all DMA channels at once. This register allows all DMA channels to be stopped with one command or all DMA channels to be reset with one command. The stopped state of each channel is reflected in the EN/DIS bit of the CCR.

The command to re-start all channels should not be issued with the same write that releases all channels from a reset command.

Register Format

```
(80960 Address = 1FF8 8000h) read/write
```



Bit Descriptions

- Bits 31–2: Reserved.
- Bit 1: Reset All Channels command. Setting this bit to 1 causes all the channels to be reset as shown by the Reset Command for each register. All channels are held in the reset state until the bit is set back to 0.
- Bit 0: Stop All Channels command. If written to a 1, all channels are stopped. If written to a 0, all channels are re-started.

  When a channel is stopped, all internally pending operations are performed before the channel stops servicing DMA requests.

Reset Conditions

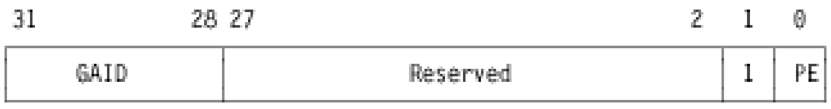
```
             Reset:  0000 0000 0000 0000 0000 0000 0000 0000
ROM Initialization:  0000 0000 0000 0000 0000 0000 0000 0000
```

CFE Local Bus Configuration Register (LBCR)

The LBCR enables various options associated with the CFE Local Bus operation.

Register Format

```
(80960 Address = 1FF8 8008h) read/write
```



Bit Descriptions

- Bits 31–28: Gate Array ID (Read-Only). Write 0s to maintain compatibility with possible future versions.
- Bits 27–2: Reserved.
- Bit 1: **Always** set this bit to 1.
- Bit 0: Address/Data Parity Enable. When set, address/data parity checking on the CFE Local Bus is enabled. When reset, address/data parity checking on the CFE Local Bus is disabled. Address and data parity are always generated on the Local Bus.

Reset Conditions

```
Reset:  0000 0000 0000 0000 0000 0000 0000 0000.
```

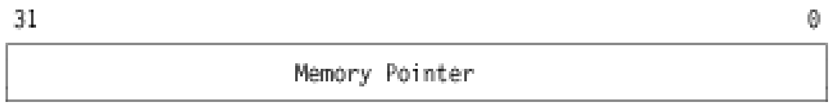
The 4-Port AIB ROM code will initialize this register to the proper value. *Do not modify the value programmed by the ROM.*

Memory Pointer Register (MPR)

The MPR contains the 32-bit CFE Local Bus address of the next data byte to be DMAed. There are no alignment restrictions on the address programmed into this register.

Register Format

`(80960 Address = 1FF8 x00Ch) read/write; x = channel number`



Reset Conditions

`Reset:  UUUU UUUU UUUU UUUU UUUU UUUU UUUU UUUU`

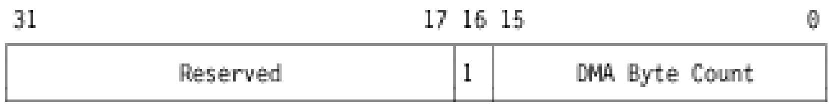
Transfer Count Register (TQC)

The TQC register contains the current DMA transfer count in bytes. Values from 0001h to FFFFh can be programmed. This allows DMA transfers of from 1 to 64KB-1 to be transferred. This register is automatically saved into memory when a chain event occurs. When a receive DMA channel has been programmed with a port size of 32-bits, transfer count values 0001h, 0002h and 0003h are illegal and should not be programmed. When a receive DMA channel has been programmed with a port size of 16-bits, a transfer count value of 0001h is illegal and should not be programmed.

For a transmit DMA channel, a TC=0 condition is always defined as the transfer count value going from 0001h to 0000h. For a receive DMA channel, a TC=0 condition depends on the port size. For a 32-bit port, a TC=0 condition is when the transfer count value is less than four. For a 16-bit port, a TC=0 condition is when the transfer count value is less than two. For an 8-bit port, a TC=0 condition is when the transfer count value is 0000h.

Register Format

`(80960 Address = 1FF8 x010h) read/write; x = channel number`



Bit Descriptions

- Bits 31–17: Reserved. These bits must be set to 0.
- Bit 16: Always set this bit to 1.
- Bits 15–0: DMA Byte Count.

Reset Conditions

`Reset:  0000 UUUU UUUU UUUU UUUU UUUU UUUU UUUU.`

Never write a zero value to this register.

Interrupt Registers

The following paragraphs describe the interrupt registers contained on the CFE Local Bus/ AIB Interface Chip:

- AIB INT0 End-of-Interrupt (EOI0) Command
- AIB INT0 End-of-Interrupt (EOI1) Command
- Interrupt Initialization Register (IIR)
- Interrupt Mask Register (IMR)
- Interrupt Status Register (ISR)

AIB INT0 End-of-Interrupt (EOI0) Command

The EOI0 command is used to inform the interrupt controller that the 80960 processor has cleared the AIB INT0 source. This command must be issued for all interrupting AIB devices that use the Interrupt Acknowledge option in “Interrupt Initialization Register (IIR)”. This command should be issued after the command that goes directly to the AIB to clear the interrupting condition. The EOI0 command is issued by writing any data value to address 1FF8 9000h.

Command Format

```
80960 ADDRESS = 1FF8 9000h
80960 DATA    = don't care
```

Reset Conditions

Not applicable.

AIB INT1 End-of-Interrupt (EOI1) Command

The EOI1 command is used to inform the interrupt controller that the 80960 has cleared the AIB INT1 source. This command must be issued for all interrupting AIB devices that use the Interrupt Acknowledge option in "Interrupt Initialization Register (IIR)". This command should be issued after the command that goes directly to the AIB to clear the interrupting condition. The EOI1 command is issued by writing any data value to address 1FF8 A000h.

Command Format

```
80960 ADDRESS = 1FF8 A000h
80960 DATA    = don't care
```

Reset Conditions

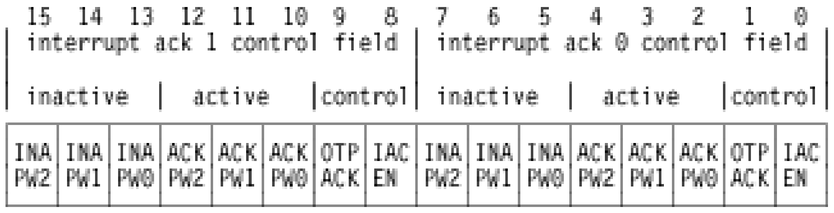
Not applicable.

Interrupt Initialization Register (IIR)

The IIR register controls the initialization parameters for the CFE Local Bus/AIB Interface Chip interrupt controller logic. This register must be programmed before interrupts are generated from the AIB.

Register Format

(80960 Address = 1FF8 8010h) read/write



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
interrupt ack 1 control field								interrupt ack 0 control field							
inactive			active			control		inactive			active			control	
INA PW2	INA PW1	INA PW0	ACK PW2	ACK PW1	ACK PW0	OTP ACK	IAC EN	INA PW2	INA PW1	INA PW0	ACK PW2	ACK PW1	ACK PW0	OTP ACK	IAC EN

Bit Descriptions

- Bits 31–16: Reserved.
- Bits 15–8: Interrupt Level 1 Control Field.
- Bits 7–0: Interrupt Level 0 Control Field.

Each 8-bit control field defines the interrupt controller operation for that interrupt level. All control fields are equivalent.

- IAC EN: Interrupt Acknowledge Enable. When set to 0, the INT0 (or INT1) input functions as a direct interrupt input from the AIB. No interrupt acknowledge cycle will be run. When using this mode, an interrupt will generate the lowest vector in the defined range for the interrupt (168 for INT0, 128 for INT1). When set to 1, an interrupt will cause an interrupt acknowledge cycle to be run on the DCL Bus. If the vector is in the defined range during the interrupt acknowledge cycle, it will be presented to the 80960 processor. If the vector is not within the defined range, the highest vector in the defined range (207 for INT0, 167 for INT1) will be given instead.
- OTP ACK: One or Two Pulse Interrupt Acknowledge Signal. Some devices use an 8259 type interrupt acknowledge cycle (two pulse). Others require external logic to eliminate the first pulse. This bit should reduce or eliminate the need for that external logic.

  OTP ACK=0: one pulse.
  OTP ACK=1: two pulses.

  If two pulses are selected, an idle time is inserted between the two pulses for device recovery, the length of which is controlled by INA PW2–0.
- ACK PW2–0: Interrupt Acknowledge Cycle Active Pulse Width. These three bits provide eight options for the interrupt acknowledge cycle active length. Some devices respond slower than others and would require external logic to insert wait states into the interrupt acknowledge cycle. These bits should eliminate the need for this external logic. Also, these bits allow a programmable active time from 120 ns to 680 ns in 80 ns increments for the interrupt acknowledge pulse.

ACK PW2	**ACK PW1**	**ACK PW0**	**INTACK 'low' (active) pulse width**
0	0	0	120ns
0	0	1	200ns
0	1	0	280ns
0	1	1	360ns
1	0	0	440ns
1	0	1	520ns
1	1	0	600ns
1	1	1	680ns

- INA PW2–0: Interrupt Acknowledge Recovery (inactive) Width. These three bits provide eight options for the interrupt acknowledge recovery length. Some devices are slower than others to release (float) data signals on the DCL Bus, and thus require a certain amount of bus idle time after the interrupt acknowledge cycle is completed. These bits allow a programmable inactive time from 120 ns to 680 ns in 80 ns increments.

INA PW2	**INA PW1**	**INA PW0**	**INTACK 'high' (inactive) pulse width**
0	0	0	120ns
0	0	1	200ns
0	1	0	280ns
0	1	1	360ns
1	0	0	440ns
1	0	1	520ns
1	1	0	600ns
1	1	1	680ns

If the OTP ACK bit is 1 (two pulses), both interrupt acknowledge pulses will be the length set by ACK PW2–0, with an inactive time between and after them determined by INA PW2–0.

Either or both interrupt acknowledge cycles can be extended by the AIB READY signal in increments of 40ns.

Reset Conditions

```
Reset:  0000 0000 0000 0000 UUUU UUU0 UUUU UUU0
```

The 4-Port AIB ROM code will initialize this register to the proper value. *Do not modify the value programmed by the ROM.*

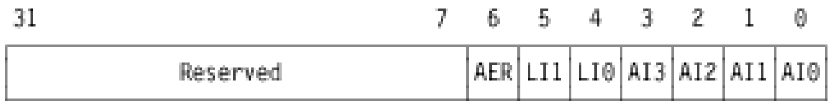
Interrupt Mask Register (IMR)

The IMR allows the four AIB interrupt inputs, the two CFE Local Bus interrupt inputs, and the AIB Error input to be enabled and disabled from causing interrupts to the 80960 processor without programming the interrupting device directly. This allows for a point of synchronization between the 80960 and the interrupting device.

Typically, devices contain their own interrupt enable functions. However, these are sometimes not usable on a realtime basis because spurious interrupts can result if an interrupt is disabled within a device immediately after its interrupt output has been asserted. This register prevents the possibility of these spurious interrupts if the device takes no precautions to do this.

Register Format

```
(80960 Address = 1FF8 8014h) read/write
```



Bit Descriptions

- Bits 31–7: Reserved.
- Bits 6: AIB_ERROR Input Mask. This bit can mask the AIB_ERROR input signal from generating an interrupt to the 80960 processor. The disabling of these bits is synchronized with the interrupt input signal to prevent spurious interrupts when disabling. Interrupts are enabled when a bit is set to 1 and disabled when set to 0.
- Bits 5–4: LB_INT1–0 Mask. These bits can individually mask the two CFE Local Bus interrupt inputs from generating an interrupt to the 80960. The disabling of these bits is synchronized with the interrupt input signal to prevent spurious interrupts when disabling. Interrupts are enabled when a bit is set to 1 and disabled when set to 0.
- Bits 3–0: AIB_INT3–0 Mask. These bits can individually mask the four AIB interrupt inputs from generating an interrupt to the 80960. The disabling of these bits is synchronized with the interrupt input signal to prevent spurious interrupts when disabling. Interrupts are enabled when a bit is set to 1 and disabled when set to 0.

Reset Conditions

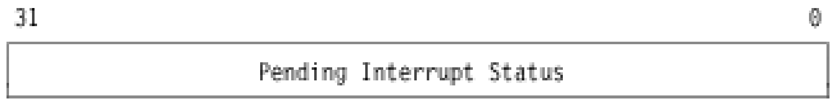
```
               Reset:  0000 0000 0000 0000 0000 0000 0000 0000
ROM Initialization:  0000 0000 0000 0000 0000 0000 0000 0000
```

Interrupt Status Register (ISR)

The ISR allows all CFE Local Bus/AIB Interface Chip interrupt input sources to read through a single status register.

Register Format

(80960 Address = 1FF8 8018h) read-only



Bit Descriptions

- Bits 31–24: Reserved.
- Bits 23–20: AIB_INT 3–0 Status.
- Bit 19: AIB Error Input Status.
- Bit 18: AIB Parity Error Status.
- Bits 17–16: CFE Local Bus Interrupt 1–0 Status.
- Bits 15–12: Memory Controller Chip Interrupt 3–0 Status.
- Bits 11–8: Micro Channel Interface Chip Interrupt 3–0 Status.
- Bits 7–0: CFE Local Bus/AIB Interface Chip DMA Channel 7–0 Status.

Reset Conditions

Not applicable.

DCL Registers

The following paragraphs describe the DCL registers contained on the CFE Local Bus⁄AIB Interface Chip:

- DCL Command/Status Register (ACSR)
- Chip Select Definition Registers (CSD0–4)
- DMA Acknowledge Pulse Width Registers (DAPW0–7)

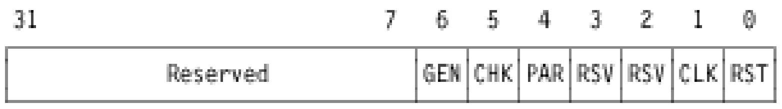
DCL Command/Status Register (ACSR)

The ACSR allows the AIB reset signal to be toggled under program control of the 80960. It should be written to a 1 to activate the reset signal and written back to 0 to deactivate it. The register also allows data parity checking to the AIB to be enabled and disabled.

Reading this register will clear interrupt vector number 240. (Vector 240 is an DCL Bus read, from Local Bus, parity error condition.)

Register Format

`(80960 Address = 1FF8 C000h) read/write`



Bit Descriptions

- Bits 31–7: Reserved.
- Bit 6: Parity Generate Polarity.

  0 = odd parity
  1 = even parity
- Bit 5: Parity Check Polarity.

  0 = odd parity
  1 = even parity
- Bit 4: Address/Data Parity Enable. This bit should be set to enable address/data parity checking of the AIB data bus.
- Bits 3–2: Reserved.
- Bit 1: Clock Enable. This bit is reset to enable the AIB_CLK signal (25MHz oscillator) to be presented to the AIB. It is set to disable the clock from propagating to the AIB.
- Bit 0: Reset. This bit is written to a 1 to activate the AIB reset signal and is written to a 0 to deactivate the signal.

Reset Conditions

Reset: 0000 0000 0000 0000 0000 0000 0000 0000

The 4-Port AIB ROM code will initialize this register to the proper value. *Do not modify the value programmed by the ROM.*

Chip Select Definition Registers (CSD0–4)

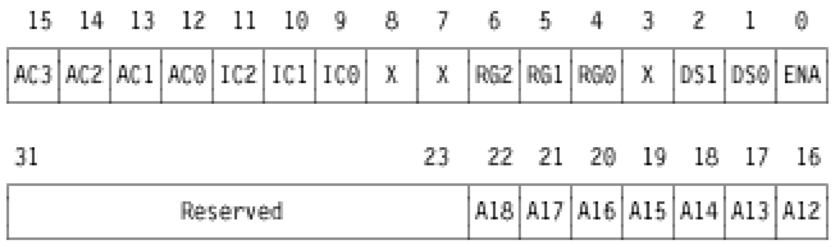
Each of the five chip select signals can be programmed for the following:

- A base starting address on any 4KB boundary within the 512KB address space of the AIB
- To decode a range of addresses from 4KB to 512KB
- To run a cycle from 240 ns to 1120 ns (in 40 ns increments)

In addition, both active and inactive (restore) portions of the cycle can be varied. This should be suitable to accommodate most devices without having to design external READY circuitry. However, an external READY signal is also available to further extend cycles or to run variable length cycles based on other real-time changing conditions that might exist on the AIB. There is a separate CSD register for each chip select signal (total of five).

Register Format

```
(80960 ADDRESS = 1FF8 B000h) CSD0 read/write
(80960 ADDRESS = 1FF8 B004h) CSD1 read/write
(80960 ADDRESS = 1FF8 B008h) CSD2 read/write
(80960 ADDRESS = 1FF8 B00Ch) CSD3 read/write
(80960 ADDRESS = 1FF8 B010h) CSD4 read/write
```



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AC3	AC2	AC1	AC0	IC2	IC1	IC0	X	X	RG2	RG1	RG0	X	DS1	DS0	ENA

31–23	22	21	20	19	18	17	16
Reserved	A18	A17	A16	A15	A14	A13	A12

Bit Descriptions

- Bits 31–23: Reserved.
- Bits 22–16: Chip Select Base Address. These seven bits determine the base address of the chip select address range. This document will only reference the primary addresses for the CFE Local Bus/AIB Interface Chip AIB devices selected by CSD0–4. The primary and secondary addresses maps for the CFE Local Bus/AIB Interface Chip AIB devices are:

  Primary Address Map: 1FF0 000h to 1FF7 FFFFh

  Secondary Address Map: 1FE0 000h to 1FE7 FFFFh

  Starting addresses can be on any 4KB boundary of the AIB address space. A value of 0000000b chooses the starting address for a chip select as 1FF0 0000h in the 80960 address space. A value of 1111111b chooses the starting address for a chip select as 1FF7 F000h in the 80960 address space. These bits can be thought of as replacing bits 18–12 of the physical 80960 address.
- Bits 15–12: Active Cycle Time. These four bits determine the active cycle time for a particular chip select. Active time is defined as the time when the -A_RD or -A_WR signal is low (0 volts)..

AC3	AC2	AC1	AC0	CS ‘low’ (active) AC0 pulse width
0	0	0	0	200ns
0	0	0	1	240ns
0	0	1	0	280ns
0	0	1	1	320ns
0	1	0	0	360ns
0	1	0	1	400ns
0	1	1	0	440ns
0	1	1	1	480ns
1	0	0	0	520ns
1	0	0	1	560ns

AC3	AC2	AC1	AC0	CS ‘low’ (active) AC0 pulse width
1	0	1	0	600ns
1	0	1	1	640ns
1	1	0	0	680ns
1	1	0	1	720ns
1	1	1	0	760ns
1	1	1	1	800ns

- Bits 11–9: Inactive Cycle Time. These three bits determine the inactive cycle time for a particular chip select. Inactive time is defined as the time when the -A_RD or -A_WR signal is high (5 volts).

IC2	IC1	IC0	CS ‘high’ (active) pulse width
0	0	0	80ns
0	0	1	120ns
0	1	0	160ns
0	1	1	200ns
1	0	0	240ns
1	0	1	280ns
1	1	0	320ns
1	1	1	360ns

- Bits 8–7: Reserved.
- Bits 6–4: Chip Select Address Range. These bits are binary encoded to provide the range of addresses that are decoded for each chip select signal. The eight binary possibilities give decode ranges of 4KB, 8KB, 16KB, 32KB, 64KB, 128KB, 256KB, and 512KB addresses. There is a restriction that the base starting address selected in bits 22–16 must be on a boundary that is equal to the decode range selected by bits 6–4. For instance, if a decode range of 32KB is selected, the base starting address can be any 32KB boundary.

RG2	RG1	RG0	Chip Select Decode Range
0	0	0	4K
0	0	1	8K
0	1	0	16K
0	1	1	32K
1	0	0	64K
1	0	1	128K
1	1	0	256K
1	1	1	512K

- Bits 2–1: Device Size. These bits are encoded with the data bus size of the device in this chip select range. Device size options of 8, 16, and 32 bits are possible.

DS1	DS0	Device size
0	0	8-bit
0	1	16-bit
1	0	32-bit
1	1	Reserved

- Bit 0: Chip Select Enable. When set to 0, this bit disables this chip select; when set to 1, this bit enables the chip select.

Reset Conditions

CSD0 ----

```
Reset:  UUUU UUUU UUUU UUUU UUUU UUU0 0UUU 0UU0
```

CSD1 ----

```
Reset:  UUUU UUUU UUUU UUUU UUUU UUU0 0UUU 0UU0
```

CSD2 ----

```
Reset:  UUUU UUUU UUUU UUUU UUUU UUU0 0UUU 0UU0
```

CSD3 ----

```
Reset:  UUUU UUUU UUUU UUUU UUUU UUU0 0UUU 0UU0
```

CSD4 ----

```
Reset:  UUUU UUUU UUUU UUUU UUUU UUU0 0UUU 0UU0
```

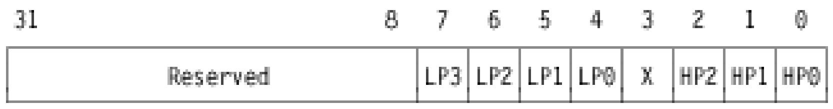
The 4-Port AIB ROM code will initialize these registers (CSD0 – CSD4) to their proper values. *Do not modify the values programmed by the ROM.*

DMA Acknowledge Pulse Width Registers (DAPW0–7)

These registers are used to program the pulse width of the DMA acknowledge (DACK) signal that the CFE Local Bus/AIB Interface Chip sends to the DMA requesting device. Different devices require this signal to be a certain duration depending on how fast the device responds. These registers allow the DACK cycle to be optimized relative to the specific DMA device. The timings given below are nominal and can vary by ± 3ns.

Register Format

```
(80960 ADDRESS = 1FF8 B030h) read/write     DACK0 pulse width
(80960 ADDRESS = 1FF8 B034h) read/write     DACK1 pulse width
(80960 ADDRESS = 1FF8 B038h) read/write     DACK2 pulse width
(80960 ADDRESS = 1FF8 B03Ch) read/write     DACK3 pulse width
(80960 ADDRESS = 1FF8 B040h) read/write     DACK4 pulse width
(80960 ADDRESS = 1FF8 B044h) read/write     DACK5 pulse width
(80960 ADDRESS = 1FF8 B048h) read/write     DACK6 pulse width
(80960 ADDRESS = 1FF8 B04Ch) read/write     DACK7 pulse width
```



Bit Descriptions

- Bits 31–8: Reserved.
- Bits 7–4: DACK 'low' Pulse Width. These four bits are binary encoded to provide ten options for setting the duration that the DACK signal will be 'low' (0 volts).

LP3	LP2	LP1	LP0	DACK 'low' (active) pulse width
0	0	0	0	80ns
0	0	0	1	120ns
0	0	1	0	160ns
0	0	1	1	200ns
0	1	0	0	240ns
0	1	0	1	280ns
0	1	1	0	320ns
0	1	1	1	360ns
1	0	0	0	400ns
1	0	0	1	440ns

- Bit 3: Reserved. Always program this bit to a 0.
- Bits 2–0: DACK 'high' Pulse Width. These three bits are binary encoded to provide eight options for setting the duration that the DACK signal will be 'high' (+5 volts).

HP2	HP1	HP0	DACK 'high' (inactive) pulse width
0	0	0	Invalid (Do not use)
0	0	1	120ns
0	1	0	160ns
0	1	1	200ns
1	0	0	240ns
1	0	1	280ns
1	1	0	320ns
1	1	1	360ns

Reset Conditions

```
Reset:  0000 0000 0000 0000 0000 0000 UUUU 0UUU
```

The 4-Port AIB ROM code will initialize these registers to the proper values. *Do not modify the values programmed by the ROM.*

Miscellaneous Registers

The following paragraphs describe these two registers contained on the CFE Local Bus/AIB Interface Chip:

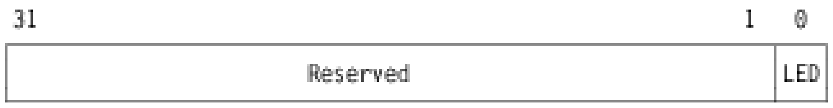
- LED Enable Register (LER)
- Presence Detect Register (PDR)

LED Enable Register (LER)

The LER is used to allow an external LED to be enabled and disabled with a write operation by the 80960. It could also be used as a general-purpose output.

Register Format

```
(80960 Address = 1FF8 D004h) read/write
```



Bit Descriptions

- Bits 31–1: Reserved.
- Bit 0: LED Enable. When set to 0, the LED_EN signal will be low. When set to 1, the LED_EN signal will be high. This is a 4-milliampere driver.

Reset Conditions

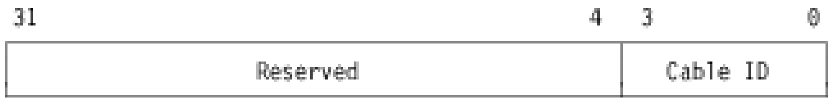
```
Reset:  0000 0000 0000 0000 0000 0000 0000 0000
```

Presence Detect Register (PDR)

The PDR can be used to read static logic levels attached to the CFE Local Bus/AIB Interface Chip. These bits are non-inverted when read.

Register Format

```
(80960 Address = 1FF8 D000h) read-only
```



Bit Descriptions

- Bits 31–4: Reserved.
- Bits 3–0: Cable ID. The IDs of the cables and wrap plug can be read through bits 3–0 and will indicate the following connections:

Value (hex)	Connected
F	Nothing (or cable ID not supported)
E–5	Reserved
4	EIA-530 Cable
3	ISO-4902 (V.36) Cable
2	EIA-232 Cable
1	ISO 4903 (X.21) Cable
0	100-pin wrap

Reset Conditions

```
Reset:  0000 0000 0000 0000 000U UUUU UUUU UUUU
```

FPGA Registers

There are two Field-Programmable Gate Array devices on the card.

The first device is called the "Control FPGA" and is used to control:

- The enables for the drivers and receivers for the electrical interfaces
- The on-card clock generation circuitry (1.544 or 2.048 MHz)

The second device is called the “X.21 Pattern Detect FPGA”. This device is used to implement X.21 Pattern Detection until the implementation in the Signetics SC26C562 DUSCC is verified.

The registers implemented in the FPGAs are described in the following paragraphs.

FPGA Status and Control Registers

The eight registers that make up the FPGA Status and Control Register set are:

- DVREN0–3, the driver/receiver enable controls, and
- INTEN0–3, the interrupt enable and status, and DTR and DSR controls.

The locations of the FPGA registers are listed underDCLMAP ; descriptions of the registers follow.

Port 0–3 Driver Enable Registers (DVREN0–3)

Figure 13-29 shows the driver enable register. This is a read/write register accessed by a co-processor platform CFE Bus Master, as shown in Table 13-18.

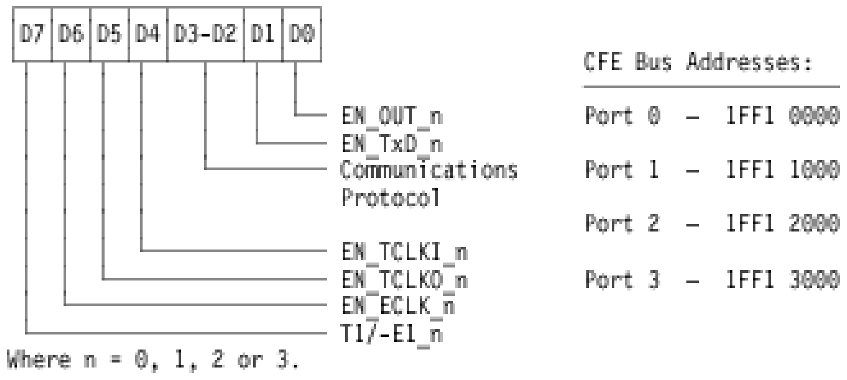


Figure 13-29. Port n Driver Enable Register Description

Bit Descriptions

- Bit 7: T1/-E1 0–3 (Read/Write). This bit selects the frequency of the on-card clock generator.

  0 = 2.048 MHz
  1 = 1.544 MHz

- Bit 6: EN_CLK 0–3 (Read/Write).

  This bit enables the on-card clock generator to drive the TxCLKn signal on the card.

  0 = disabled
  1 = enabled

  EN_TCLKIn must be inactive and the DUSCC must not be driving the TxCLK output before enabling this bit. See *Pin Configuration Register (PCRA/B)* on page 291.

- Bit 5: EN_TCLK0 0–3 (Read/Write).

  This bit must be programmed to a 1 to enable the outbound transmit clock driver (DCE function) on Ports 0–3.

  Bit 4: EN_TCLKI 0–3 (Read/Write). This bit must be programmed to a 1 to enable an inbound transmit clock receiver (DTE function) on Ports 0–3.

  EN_CLKn must be inactive and the DUSCC must not be driving the TxCLK output before enabling this bit. See *Pin Configuration Register (PCRA/B)* on page 291.

- Bits 3–2: Communications Protocol.

  These bits determine the electrical interface used by the port, as follows:

  '00' — EIA-530 or ISO 4903 (X.21)

  '01' — ISO 4902 (V.36)

'10' — reserved

'11' — EIA-232-D

- Bit 1: EN_TXD 0–3 (Read/Write). This bit must be programmed to a 1 to allow the data from the DUSCCs to the transmit data lines (TxD) for that port. It should be programmed to 0 for transmission of all 0s (for use by X.21 applications).
- Bit 0: EN_OUT 0–3 (Read/Write). This bit must be programmed to a 1 to enable the output driver for that port.

  The 4-Port Multi-Interface AIB's ROM initialization code maintains the reset value of these registers.

Reset Conditions

```
DVREN0-3 = 0000 0000
```

Port 0–3 Interrupt Enable Registers (INTEN0–3)

Figure 13-30 shows the interrupt enable register. This is a read/write register accessed by a co-processor platform CFE Bus Master, as shown in Table 13-18.

Interrupts for -DSR0 and -DSR1 are combined onto AIB INT2. Interrupts for -DSR2 and -DSR3 are combined onto AIB INT3.

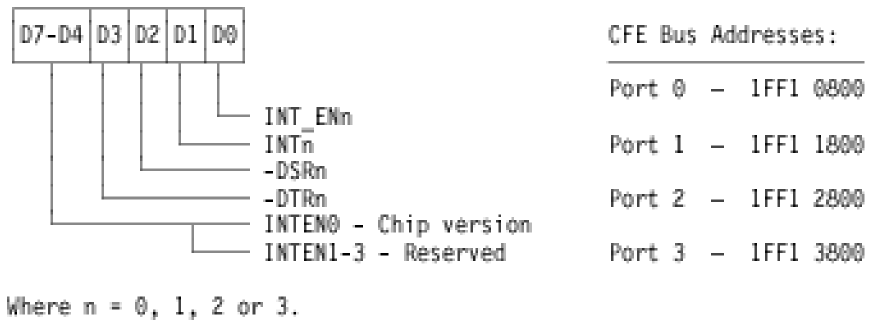


Figure 13-30. Port n Interrupt Enable Register Description

Bit Descriptions

- Bits 7–4:
  - Port 0: Chip Version (Read-Only). These four bits contain the version level of the chip. Write 0s to maintain compatibility with future versions.
  - Ports 1–3: Reserved. These bits are read as '0000' (binary). Write 0s to maintain compatibility with future versions.
- Bit 3: -DTR 0–3. Data Terminal Ready Signal for Ports 0 through 3 (Read/Write).

  0 = inactive (high)
  1 = active (low)
- Bit 2: -DSR 0–3. Data Set Ready Signal for Ports 0 through 3 (Read-Only).

  0 = inactive (high)
  1 = active (low)

- Bit 1: INT 0–3. Interrupt Active for -DSR_n (Read-Only). When read, this bit indicates whether there is currently an interrupt active corresponding to -DSR_n.

  0 = interrupt not active
  1 = interrupt active (see below)

  If the interrupt is active when this register is read, the interrupt is cleared at the end of the read operation. The interrupt will also be cleared by disabling the interrupt (bit 0) or by a reset of the 4-Port Multi-Interface AIB.

- Bit 0: INTEN 0–3. Interrupt Enable for -DSR_n (Read/Write). This bit controls whether -DSR_n can generate an interrupt to the co-processor platform.

  0 = interrupt not enabled
  1 = interrupt enabled

  If enabled, an interrupt will be generated on a transition (positive **or** negative) of the -DSR_n signal.

  If the port is selected for operation in either EIA-530 or X.21 modes (in the DVRENn register), the interrupt is automatically disabled. The state of this bit will have no effect.

The 4-Port Multi-Interface AIB's ROM initialization code maintains the reset value of these registers.

Reset Conditions

```
INTEN0   = xxxx 0x00
INTEN1-3 = 0000 0x00
```

X.21 Pattern Detect Registers

The eight registers that make up the X.21 Pattern Detect Register set are:

- X21ST0–3, the pattern detect status, and
- X21EN0–3, the enables.

The operation of the X.21 pattern detection is described under *X.21 State Detection Circuit Description* on page 242.

The Clock Presence Detect circuitry is used to detect whether or not a cable is connected to the port. Pulling the input line states to a known level is not sufficient, because the states must be clocked as valid for X.21 protocol, and the clock would not be working if the cable was disconnected.

This circuitry detects that the frequency of the receive clock is below 12 KHz (12000 bits/second) and can supply an interrupt or a status bit to inform the application of a possible line-break condition.

The locations of the registers are shown under *DCL Bus Devices* on page 255 and detailed in the following sections.

Port 0–3 X.21 Enable Registers (X21EN0–3)

Figure 13-31 shows the X.21 enable register. This is a read/write register accessed by a co-processor platform CFE Bus Master, as shown in Table 13-18.

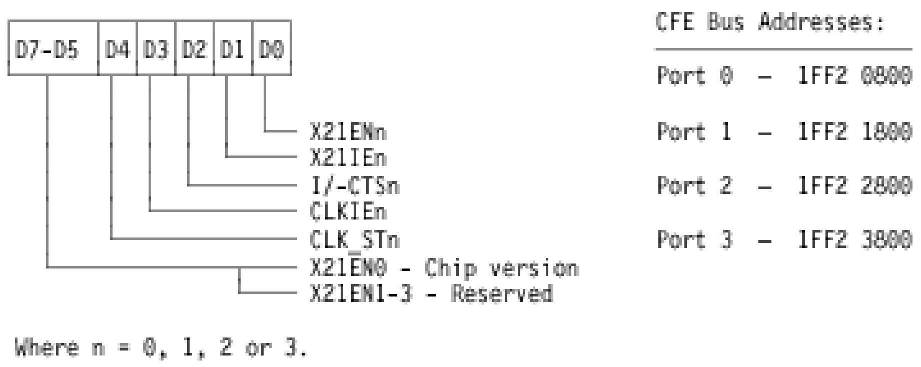


Figure 13-31. Port n X.21 Enable Register Description

Bit Descriptions

- Bits 7–5:
  - Port 0: Chip Version (Read-Only). These three bits contain the version level of the chip. Write 0s to maintain compatibility with future versions.
  - Ports 1–3: Reserved. These bits are read as '000' (binary). Write 0s to maintain compatibility with future versions.
- Bit 4: CLK_STn. Clock Presence Detect Status. This bit indicates the presence (or lack of) an RxCLKn frequency above about 12 KHz.

  0 = RxCLK frequency is below 12 KHz or completely absent
  1 = RxCLK frequency is 12 KHz or above
- Bit 3: Reserved. Clock Presence Detect Interrupt Enable (Read/Write).

  0 = disabled
  1 = enabled

  When enabled, an interrupt is generated when the Clock Presence Detect Status changes state.
- Bit 2: I/-CTS 0–3. X.21 I or CTS Select (Read/Write). This bit controls the timing of the CTS input to the DUSCC.

  0 = CTS mode — signal is passed to the DUSCC without modification.

  1 = X.21 I mode — signal is held at one state until it is valid (steady at the opposite state for 16 bit times).
- Bit 1: X21IE 0–3. Pattern Detect Interrupt Enable (Read/Write).

  0 = disabled
  1 = enabled
- Bit 0: X21EN 0–3. Pattern Detect Enable (Read/Write). This bit will allow the circuitry to search for patterns. When disabled, all status bits for the corresponding port are cleared, the overflow bit is cleared, and the pattern counter is set to 0.

0 = disabled
1 = enabled

The 4-Port Multi-Interface AIB's ROM initialization code maintains the reset value of these registers.

Reset Conditions

```
X21EN0   = xxx0 0000
X21EN1-3 = 0000 0000
```

Port 0–3 X.21 Pattern Status Registers (X21ST0–3)

Figure 13-32 shows the pattern status register. This is a read-only register accessed by a co-processor platform CFE Bus Master, as shown in Table 13-18.

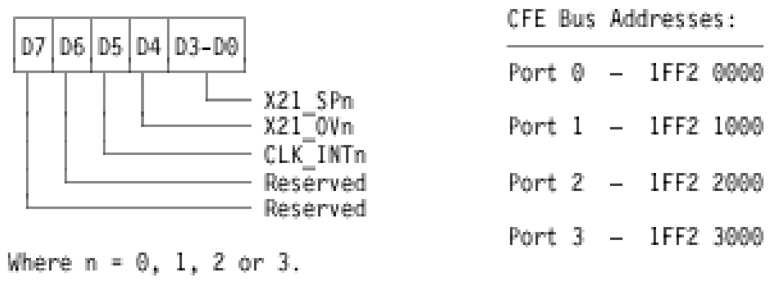


Figure 13-32. Port n X.21 Pattern Status Register Description

Bit Descriptions

- Bit 7–6: Reserved. These bits are read as '00' (binary).
- Bit 5: CLK_INTn. Clock Presence Detect Interrupt Active. This bit indicates if an interrupt is active due to a change in the status of CLK_STn, if enabled in X21ENn register.

  0 = interrupt not active
  1 = interrupt active

  If active when read, the interrupt is cleared at the end of the read.
- Bit 4: X21_OVn. X.21 Pattern Detect Overflow. On read, this bit indicates the occurrence of a pattern that was not cleared by software before the next pattern was detected.

  0 = no overflow
  1 = overflow — at least one pattern was missed

  Reading this register will clear the status to no overflow and clear the X21_SPn to not detected.
- Bits 3–0: X21_SPn(0–3). X.21 State Pattern. On read, this bit indicates the occurrence of a 16-bit pattern.

  0000 = pattern not detected
  0001 = pattern of 16 0s detected
  0010 = pattern of 16 1s detected

0011 = pattern of 16 alternating 0s and 1s detected
0100–1111 = reserved

Reading this register will clear the status to not detected and clear the X21_OVn to no overflow. Clearing the status of these bits will also clear the interrupt output to -AINT2 or -AINT3, if enabled.

Reset Conditions

```
X21ST0-3 = 0000 0000
```

Hardware Notes

Signetics SC26C562 DUSCCD Interrupt Timings

The Signetics SC26C562 can generate interrupts from its -CTS and -CD inputs that can be used by software to detect state changes, especially in the case of X.21 with -CTS.

However, the sampling rate the DUSCC uses to generate the interrupts for these signals is relatively slow—6.8 microseconds. The DUSCC also must sample the signal at the new level twice before it recognizes it as valid.

This means the interrupt will not be generated for up to 17 microseconds after the signal changes state. See *DCL INT0 and INT1 End-of-Interrupt Registers (EOI0 and EOI1)* on page 290 for further information on DUSCC interrupt handling.

Signetics SC26C562 DUSCC CCR Register

The *Signetics SC26C562/SC68C562 CMOS DUSCC User's Guide* says a minimum of three X1 clocks with the chip enable inactive are necessary between two writes to the CCR of the same port.

The X1 clock is the 14.745 MHz oscillator, and 3 clocks correspond to 226 ns. The application can either:

- Read an FPGA register on the DCL Bus after every write to the CCR

  or

- Change the inactive CS time for the DUSCCs to 240 ns in the CFE Local Bus/AIB Interface Chip CSD1 and CSD2 registers.

The chip selects for the FPGA registers are programmed for an active time of longer than 300 ns by the AIB ROM initialization code, so the first option will ensure no accesses to the DUSCC for that period. However, there is no hardware control to ensure that the FPGA register is written; the software must account for it.

The second option has the disadvantage of slowing down **all** consecutive accesses to the DUSCC, and it affects performance. However, it has the advantage of being completely transparent to software once the inactive time has been changed.

Signetics SC26C562 DUSCC RTS Output in X.21 Applications

The -RTS output of the DUSCC, which is output in ISO 4903 (X.21) applications as the Control (C) signal, is not synchronized with the TxCLK (X) clock output.

The X.21 protocol specifies that it should be synchronized.

CFE Local Bus/AIB Interface Chip AIB Reset Output

The -A_RESET output of the CFE Local Bus/AIB Interface Chip module does not go active when the -RESET on the CFE bus is activated, but does go active for 60 ns when the -RESET on the CFE bus is removed.

The Signetics SC26C562 DUSCC specifications state that the minimum reset should be 200 ns.

The 4-Port Multi-Interface AIB ROM code on the card instructs the processor to activate the -A_RESET in the CFE Local Bus/AIB Interface Chip module through the ACSR register at CFE address 1FF8 C000h, bit 0, for that length of time.

Programming Restrictions

Critical Registers

The following registers on the 4-Port Multi-Interface AIB are initialized to proper operational values by the 4-Port AIB ROM, and **must not** be modified by other code.

- CFE Local Bus/AIB Interface Chip
  - LBCR (Page 269)
  - IIR (Page 271)
  - ACSR (Page 275)
  - CSD0–4 (Page 276)
  - DAPW0–4 (Page 279)
- Signetics SC26C562 DUSCC
  - IVR (Signetics SC26C562/SC68C562 CMOS DUSCC User’s Guide)

In addition, the restrictions below **must** be followed to ensure proper operation of the card.

RadiSys ARTIC960 Co-Processor Platform

The following register, located on the co-processor platform, must be programmed correctly for proper operation of the 4-Port Multi-Interface AIB.

Interrupt Controller Enable/Detect Register (EDR)

The EDR register is used to configure the interrupt format received from the AIB. For operation with the AIB, this register must be set for *vectored* AIB interrupts, and the enable active. Vectored AIB interrupts are selected by writing a 1 to bit 1 of EDR. AIB interrupts are enabled by writing a 1 to bit 3 of EDR.

The 80960 address of this register is 0A00 0004h.

The AIB’s ROM initialization code ORs the value 0A (hex) to this register to configure it for proper operation.

For more details on this register, see *Enable/Detect Register (EDR)* on page 46.

CFE Local Bus/AIB Interface Chip

The following registers, located in the CFE Local Bus/AIB Interface Chip module, must be programmed correctly for proper operation of the 4-Port Multi-Interface AIB.

Interrupt Mask Register (IMR)

This register is used to enable or disable interrupt inputs to the CFE Local Bus/AIB Interface Chip module. It must be programmed to enable the 4-Port Multi-Interface AIB interrupts 0–3 as required by the software.

The AIB’s ROM initialization code writes '00' to this register, disabling all interrupt inputs from the CFE Local Bus/AIB Interface Chip.

AIB Address 1/2 Register (AIB_ADDR 1/2)

This register contains two separate 16-bit address fields for possible I/O operations that the hardware will perform when a chain event occurs. It must be programmed so that both operations access locations defined within the same AIB chip select; that is, either CS1 **or** CS2, but not both.

DCL INT0 and INT1 End-of-Interrupt Registers (EOI0 and EOI1)

These registers are written to clear the interrupt vector that was latched in the CFE Local Bus/AIB Interface Chip module after an interrupt acknowledge cycle was run for either -AINT0 or -AINT1 (the Signetics SC26C562 DUSCC interrupts). The software routine that services the interrupt writes to the EOI register to clear the interrupt before returning from the service routine.

However, because the processor on the base card (such as the RadiSys ARTIC960 Co-Processor Platform) may be capable of exiting from its service routine before the write cycle to the CFE Local Bus/AIB Interface Chip module is complete, a read of a register on the CFE bus is recommended after the write to EOI0 or EOI1 and before the service routine is exited.

Signetics SC26C562 DUSCC

The locations of the DUSCC registers are listed under *FPGA Registers* on page 282.

For further details on these registers, see the *Signetics SC26C562/SC68C562 CMOS DUSCC User’s Guide*.

All possible functions and features available with the DUSCC registers are not available. Following are the known programming restrictions on the DUSCC chip. Failure to comply with these restrictions may lead to unpredictable results.

Channel Mode Register 2 (CMR2)

- Bits 5–3:

'010' — Duplex, single-address mode (DMA operation)
'111' — Polled or interrupt

These are the only two valid bit combinations for bits 5–3 of CMR2.

Pin Configuration Register (PCRA/B)

Program these registers as follows:

- Bit 7: '1' — External Clock. (Channel A only; this bit is undefined for Channel B.)
- Bit 6: '0' — Select the GPO2 function (if not in duplex DMA).
- Bit 5: '1' — Select the RTS function.
- Bits 4–3: '00' — Select the RTxC pin as an input.
- Bits 2–0: '000' — Select the TRxC pin as an input. This can be changed to select this pin as an output, but only **after** the corresponding DVRENn register has been set to disable the other TRxC sources. See *Port 0–3 Driver Enable Registers (DVREN0–3)* on page 283.

The 4-Port Multi-Interface AIB's ROM initialization code writes A0 (hex) to PCRA and 20 (hex) to PCRB on both DUSCCs.

Output and Miscellaneous Register (OMRA/B)

Program these registers as follows:

- Bit 4: '0' — FIFO not full.
- Bit 3: '0' — FIFO not empty.
- Bit 2: '0' — GPO2 = high (unused function).
- Bit 1: '0' — GPO1 = high (unused function).
- Bit 0: '0' — RTS = high (inactive). Program this bit to a 0 for RTS inactive or program to a 1 for RTS active.

The 4-Port Multi-Interface AIB's ROM initialization code writes 00 to these registers on both DUSCCs.

Interrupt Control Register (ICR)

Program these registers as follows:

- Bits 7–6: '10'. This sets the DUSCC's interrupt priorities to be 'Interleaved A', in that priorities are interleaved between channels. Channel A has the highest priority between events of equal channel priority as follows (in order of descending priority):

  A(0) — Channel A receiver ready
  B(0) — Channel B receiver ready
  A(1) — Channel A transmitter ready
  B(1) — Channel B transmitter ready
  A(2) — Channel A Rx/Tx status
  B(2) — Channel B Rx/Tx status
  A(3) — Channel A external or C/T status
  B(3) — Channel B external or C/T status.

- Bits 5–4: '00'. This puts the DUSCC in 'two IACK pulse' mode.
- Bit 3: This bit **must** be set to 0 for DUSCC 0. It can be set to either 0 or 1 for DUSCC 1; 0 is recommended. This bit selects which vector bits are modified by interrupt status; this will result in illegal vectors if set to 1 for DUSCC 0.
- Bit 2: '1'. This causes the interrupt status to modify the vector given to the CFE Local Bus/AIB Interface Chip.
- Bits 1–0: '00'. This disables the interrupts from both DUSCC channels.

The 4-Port Multi-Interface AIB’s ROM initialization code writes 84 (hex) to this register on both DUSCCs, which results in the preceding settings and disables interrupts for both channels.

Other Restrictions

A channel cannot be dynamically reconfigured.

- Do not write to CMR1, CMR2, S1R, S2R, or PCR when the channel is in use (receiver or transmitter enabled).
- Do not write to RPR or RTR if the receiver is enabled.
- Do not write to TPR or TTR if the transmitter is enabled.
- Do not write to CTCR, CTPRH, or CTPRL if the counter/timer is enabled.

X.21 Interface

X.21 Protocol

The X.21 protocol is *not* provided in hardware. It is the user’s responsibility to provide software support to implement it.

DUSCC Signal Assignment

The following DUSCC signals are assigned per port:

- The TxD pin is assigned to TRANSMIT (T).
- The RTS pin is assigned to CONTROL (C).
- The RxD pin is assigned to RECEIVE (R).
- The CTS pin is assigned to INDICATE (I).
- The RCLK pin is assigned to SIGNAL ELEMENT TIMING (S).
- The TCLK pin is assigned to DTE SIGNAL ELEMENT TIMING (X).

DUSCC Pin Programming

The following DUSCC pins are programmed as follows.

- RTS pin
  - RTS is an output pin.

  - RTS is *not* affected by the status of the transmitter.
  - The logical state of the pin is controlled by the OMR register.
  - CTS pin
  - CTS is an input pin.
  - CTS has *no* affect on the transmitter.
  - An interrupt is generated each time a logical transition occurs.
- The Transmit Clock (TCLK) and Receive Clock (RCLK) pins
  - **DTE Operation** (clock timings are provided by the DCE device at the other end of the cable). Each port is typically configured to act as a DTE (Data Terminal Equipment) device.

    In this mode, RCLKIn is input from a source external to the card on the Receive Clock (S) line of the cable and is used for the receive clock on the DUSCC.

    There is no Transmit Input signal defined for a DTE device using the X.21 protocol, so the DUSCC should be programmed to use its receive clock input (RCLK) for transmit timings.

    If the DCE device supports an input Transmit Clock (X), this clock should be driven back out of the DUSCC on its transmit clock (TCLK) pin, and the TXCLKOn signal should be driven from the port to the Transmit Clock (X) line of the cable.

    This configuration is shown in Figure 13-33.

    In applications using higher speeds than 64K bits/second, it is strongly recommended that the card supply the Transmit Clock (X) line to the DCE equipment. This is because the clocking of the transmitted data by the DCE may be unreliable due to the cumulative effect of:

    - Delays of the DCE and DTE drivers and receivers
    - Internal DUSCC delay
    - Propagation delay of the cable in both directions.

    If the DCE does not support an input Transmit Clock (X), cable lengths should be kept as short as possible to minimize the total delay.

    Older DCE equipment may drive the same lines on the cable for other purposes that the 4-Port Multi-Interface AIB uses for the Transmit Clock (X) (pins 7 and 14 of the 15-pin X.21 connector). If this is the case, the TCLKOn output should not be driven by the 4-Port Multi-Interface AIB. (See Table 13-12 for a complete listing of the pin assignments for the D-Shell Connectors.

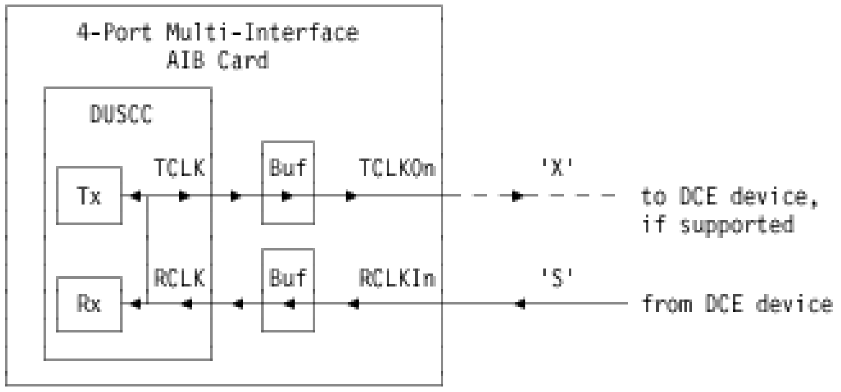


Figure 13-33. X.21 DTE Operation

- **DCE Operation** (clock timings are provided to the DTE device at the other end of the cable)

  If the port is to be directly connected to a DTE device, the port can be configured to act as a DCE (Data Communications Equipment) device with the addition of an external null modem connection.

  In this mode, the transmit clock for the port must be supplied by either the baud rate generator in the DUSCC or the on-card oscillator, and must be driven out through the cable on TCLKOn. The null modem then should make the proper connections so it gets to the DTE on the Receive Clock (S) line.

  If the DTE device does not supply the Transmit Clock (X), the null modem must wrap the Receive Clock (S) back to the RCLKIn input as shown in Figure 13-34.

  If RCLKIn is *not* provided by the DTE device or the null modem, the X.21 Pattern Detection Circuitry and the Clock Detection Circuitry on the card will not work, since they are both based off of the RCLKIn input.

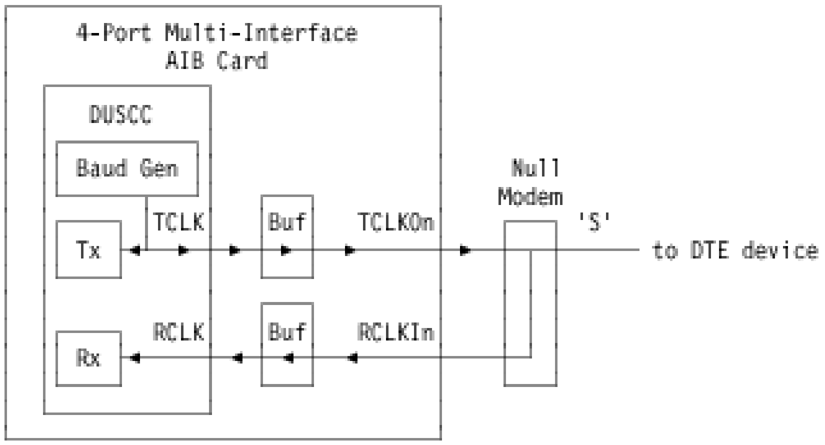


Figure 13-34. X.21 DCE Operation — Low Speed

In applications where the DTE device supplies the Transmit Clock (X), the receive clock should be programmed to use the DUSCC receive clock pin. This uses the Transmit Clock (X) line from the DTE device, which the null modem will connect to RCLKIn. This configuration is shown in Figure 13-35.

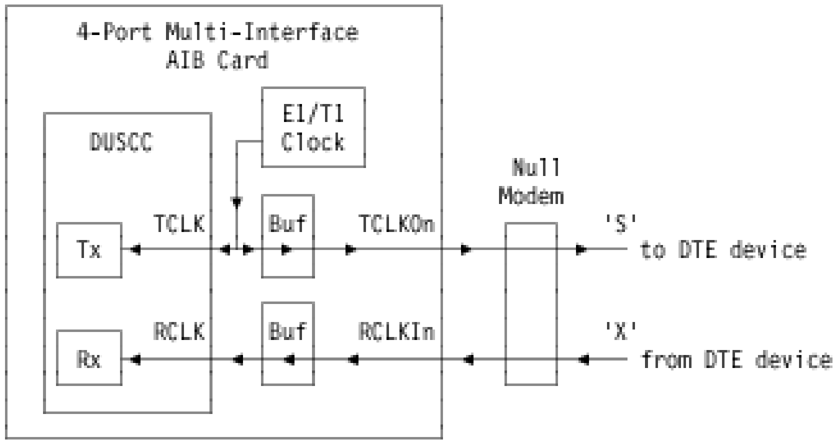


Figure 13-35. X.21 DCE Operation — High Speed

The complete null modem connections are shown in Figure 13-36. (The cable signals in this figure are connected between the port and the DTE.)

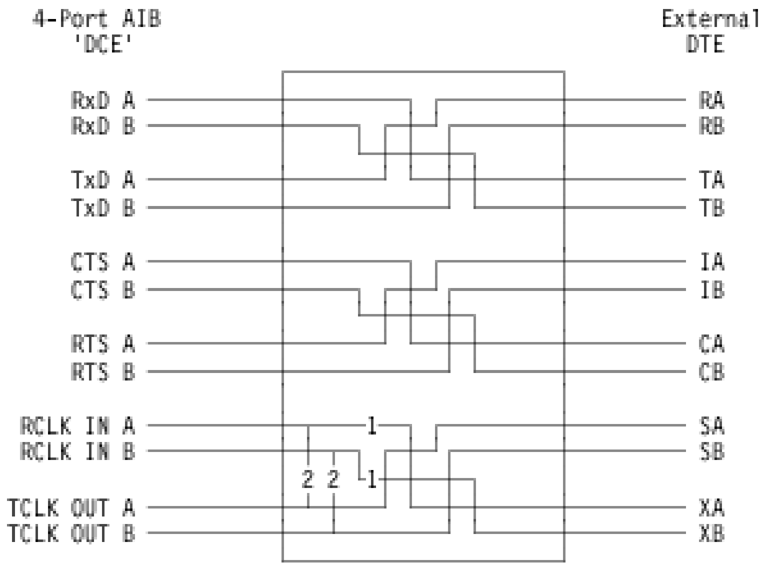


Figure 13-36. Null Modem Connections

.

The connections marked 1 should be made if the DTE device supplies the Transmit Clock (X), and the connections marked 2 should be made if it does not.

X.21 Pattern Detection

The circuitry implemented in the X.21 State Detection FPGA should be used, instead of the X.21 function implemented on the Signetics SC26C562 DUSCC. See *X.21 State Detection Circuit Description* on page 242 and *X.21 Pattern Detect Registers* on page 285.

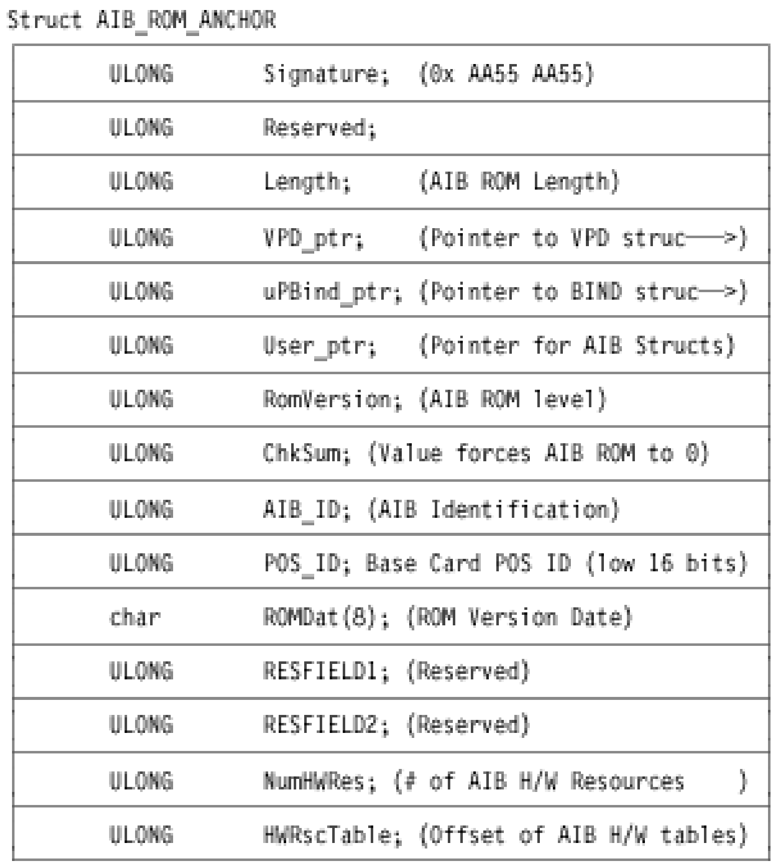
AIB ROM Description

This section describes the basic structures, power-on self test (POST) routines, as well as specific product data and error codes for the 4-Port Multi-Interface AIB.

AIB ROM Structures

AIB ROM Signature Block

The AIB ROM signature block is located at offset zero within the AIB ROM. It is the anchor that allows the Base ROM to find the AIB ROM and locate all important AIB ROM Data.



Struct AIB_ROM_ANCHOR

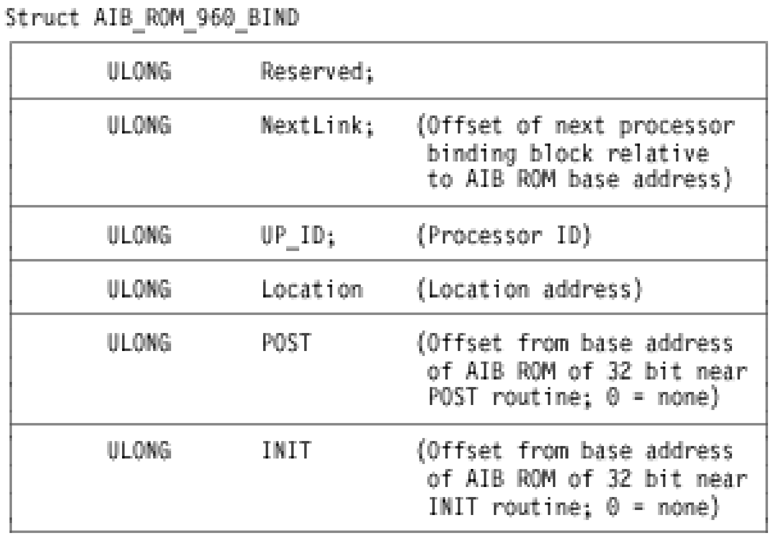
ULONG	Signature; (0x AA55 AA55)
ULONG	Reserved;
ULONG	Length; (AIB ROM Length)
ULONG	VPD_ptr; (Pointer to VPD struc—>)
ULONG	uPBind_ptr; (Pointer to BIND struc—>)
ULONG	User_ptr; (Pointer for AIB Structs)
ULONG	RomVersion; (AIB ROM level)
ULONG	ChkSum; (Value forces AIB ROM to 0)
ULONG	AIB_ID; (AIB Identification)
ULONG	POS_ID; Base Card POS ID (low 16 bits)
char	ROMDat(8); (ROM Version Date)
ULONG	RESFIELD1; (Reserved)
ULONG	RESFIELD2; (Reserved)
ULONG	NumHWRes; (# of AIB H/W Resources)
ULONG	HWRscTable; (Offset of AIB H/W tables)

AIB ID

The ROM ID is 001x xxxxh for the 4-Port Multi-Interface AIB.

AIB ROM Binding Block

Currently, the 4-Port Multi-Interface AIB’s ROM supports the Intel 80960CA processor. The 80960CA processor binding block follows the following format:



Struct AIB_ROM_960_BIND

ULONG	Reserved;	
ULONG	NextLink;	(Offset of next processor binding block relative to AIB ROM base address)
ULONG	UP_ID;	(Processor ID)
ULONG	Location	(Location address)
ULONG	POST	(Offset from base address of AIB ROM of 32 bit near POST routine; 0 = none)
ULONG	INIT	(Offset from base address of AIB ROM of 32 bit near INIT routine; 0 = none)

AIB ROM Power-On Self Test (POST)

The following sections outline the execution and testing of the AIB ROM POST.

AIB ROM POST Execution

When calling the AIB POST routine, base ROM places the address of the base ROM Anchor in the 80960 register **g1**.

AIB POST preserves the contents of all registers except the local registers and **g0**. AIB POST returns its error status in register **g0** in the same format as AIBStatus word, as described under *Error Reporting* on page 311.

The following operations are performed by the AIB ROM power-on self test:

- Miscellaneous Test
- General Register Test
- Asynchronous Polled Transfer Mode
- Asynchronous Interrupt Transfer Mode
- Asynchronous Direct Memory Access (DMA) Transfer Mode
- HDLC/SDLC Transfer Mode
- BISYNC Transfer Mode
- DMA Start/Stop Using the DMA Channel Command Registers (DCCRs)
- DMA Start/Stop Using the Global DMA Command Register (GDCR)

Miscellaneous Test

This test verifies the ID of the base card and the ID of the CFE Local Bus/AIB Interface Chip. The rest of the POST tests are skipped if the ID for the CFE Local Bus/AIB Interface Chip is invalid.

Output Parameters

```
SUCCESS             –   Test pass
UNKNOWN_BASE_CARD   –   Error in Base Card ID test
INVALID_INTERFACE   –   Invalid Interface Chip ID
NO_FREE_MEMORY      –   Error allocating free memory for testing
```

Method of Verification

The base card ID is read from the base card ANCHOR structure. This read value is then compared with the RadiSys ARTIC960 card ID. The CFE Local Bus/AIB Interface Chip ID is determined by reading the Presence Detect Register (PDR) in the CFE Local Bus/AIB Interface Chip. If the AIB card contains an invalid CFE Local Bus/AIB Interface Chip, none of the POST tests are performed.

Sequence of Steps

The sequence of steps for this test is as follows:

1. Read the base card ID from the base card ROM ANCHOR structure.
2. Verify that it matches with the RadiSys ARTIC960 ID.
3. Return error code if it does not match.
4. Read the CFE Local Bus/AIB Interface Chip ID by reading the PDR register.
5. if it is invalid, then skip all the remaining POST tests.
6. Return SUCCESS.

The NO_FREE_MEMORY error is returned if a problem is encountered while trying to allocate memory for testing. The memory is allocated from pointers passed from the co-processor platform base card.

General Register Test

This test verifies the writing and reading of all the registers in the CFE Local Bus/AIB Interface Chip, the two DUSCC chips, as well as the FPGA modules. After all the verification tests are done, the POST code reads the Base ROM INTID structure to check if any unexpected interrupts occurred.

Output Parameters

```
Success             Test pass
VERO_REG_FAIL       CFE Local Bus/AIB Interface Chip register
                    verification test failed
DUSCC_REG_FAIL      DUSCC chip register verification test
                    failed
CTRL_FPGA_REG_FAIL  Control FPGA register verification test
                    failed
X21_FPGA_REG_FAIL   X21 FPGA register verification test failed
INVALID_AIB_INT     Unexpected interrupt occurred
```

Method of Verification

The General Register test writes hex string '5'('0101' in binary) in each nibble for most of the registers and reads it back along with a suitable mask. The next verification test is performed by writing hex string 'A'('1010' in binary) in each nibble for most of the registers and reading them back along with a suitable mask.

Sequence of Steps

The sequence of steps for this test is as follows:

1. Write test patterns in the register.
2. Read that register and logically 'AND' the read value with the appropriate mask.
3. Compare the read value with the expected value.
4. If they do not compare, return the appropriate error code.
5. Repeat these steps for all the registers on the AIB.
6. Check for any unexpected interrupts.
7. If any unexpected interrupts occurred, then return the error code.
8. Return SUCCESS.

Asynchronous Polled Transfer Mode Test

This test verifies the asynchronous transfer mode of each port on the AIB. This test primarily exercises the Philips CDUSCC.

Output Parameters

```
SUCCESS              – Test pass
ERROR_POL.Dx_TIMEOUT – Transfer not done on Port x (0–3)
ERROR_POL Dx_DATA    – Tx/Rx data mismatch on Port x (0–3)
```

Method of Verification

The asynchronous test transfers eight bytes through each port in Local Loopback Mode with interrupts disabled. The test will poll the status register of each receiver for every byte of receive data. A timeout is provided in software if the data is not received.

Sequence of Steps

The sequence of steps for this test is as follows:

1. Initialize both DUSCCs (four ports) for asynchronous transfer as shown in the following table:

Register	Data	Description
Channel Command Register	00 h	Reset Transmitter
Channel Command Register	40 h	Reset Receiver
Channel Mode Register 2	B8 h	Set Channel Connection for Local Loopback and Data Transfer for Polled or Interrupt.
Channel Mode Register 1	07 h	Set to asynchronous mode with NO PARITY
Transmitter Parameter Register	73 h	Set 8-bit character with 1 stop bit
Receiver Parameter Register	03 h	Set 8-bit character
Receiver Timing Register	2F h	Set the Receiver BRG at 38.4k
Transmitter Timing Register	3F h	Set the Transmitter BRG at 38.4k
Channel Command Register	02 h	Enable the Transmitter
Channel Command Register	42 h	Enable the Receiver

2. Define variable VALUE of type char and set to 08 h.
3. Write VALUE to Transmitter (all four ports).
4. Wait 1 millisecond.
5. Check the General Status Register for RxRDY (all four ports).
6. Read the contents of Receiver and check against VALUE.
7. Return SUCCESS for test pass and port status for test fail.
8. Decrement VALUE.
9. Continue above steps until VALUE=0.

Asynchronous Interrupt Transfer Mode Test

This test verifies the asynchronous transfer interrupt for each port on the AIB. This test primarily exercises the Philips CDUSCC.

Output Parameters

```
SUCCESS              - TEST PASS
ERROR_INTx_TIMEOUT   - Failed to receive interrupt on Port x (0-3)
ERROR_INTx_DATA      -Tx / Rx data mismatch on Port x (0–3)
ERROR_INTx_DATA      -Tx Method of Verification
```

The asynchronous test transfers eight bytes through each port in Local Loopback Mode with interrupts enabled. A timeout will be provided in software if the receiver interrupt does not occur.

Sequence of Steps

The sequence of steps for this test is as follows.

1. Initialize both DUSCCs (four ports) for asynchronous transfer as shown in the following table:

Register	Data	Description
Channel Command Register	00 h	Reset Transmitter
Channel Command Register	40 h	Reset Receiver
Channel Mode Register 2	B8 h	Set Channel Connection for Local Loopback and Data Transfer for Polled or Interrupt.
Channel Mode Register 1	07 h	Set to asynchronous mode with NO PARITY
Transmitter Parameter Register	73 h	Set 8-bit character with 1 stop bit
Receiver Parameter Register	03 h	Set 8-bit character
Receiver Timing Register	2F h	Set the Receiver BRG at 38.4k
Transmitter Timing Register	3F h	Set the Transmitter BRG at 38.4k
Interrupt Control Register	87 h	Enable master interrupt
Channel Command Register	02 h	Enable the Transmitter
Channel Command Register	42 h	Enable the Receiver

2. Enable AIB interrupts in the CFE Local Bus/AIB Interface Chip by writing (0F h) to IMR register.
3. Write one character to transmitter (all four ports).
4. The remaining characters are put in the transmitter in the interrupt service routine. All characters are received in receiver interrupt service routine.
5. Wait for 2 milliseconds to receive all 8 characters.
6. Compare Tx and Rx data.
7. If the data matches, then return SUCCESS.
8. If timeout or data mismatch happens, return error code.

Asynchronous DMA Transfer Mode Test

This test verifies the function of the AIB in asynchronous transfer mode. It also tests the CFE Local Bus/AIB Interface Chip’s DMA function with stop on Terminal Count feature on receiver as well as transmitter.

Output Parameters

```
SUCCESS                 – Test pass
ERROR_ADMAx_TIMEOUT     – Transfer not done on port x (0–3)
ERROR_ADMAx_DATA        - Tx/Rx data mismatch on port x (0–3)
ERROR_ADMAx_NOSTOP      - CFE Local Bus/AIB Interface Chip DMA did
                          not stop on port x (0–3)
ERROR_ADMAx_NOINT       – CFE Local Bus/AIB Interface Chip DMA did
                          not interrupt on port x (0–3)
```

Method of Verification

The ADMA transfer test programs the CFE Local Bus/AIB Interface Chip's DMA transmit and receive channel for transfer count of 40. Both the DUSCCs are programmed for asynchronous transfer mode. The CFE Local Bus/AIB Interface Chip's transmitter and receiver channels are programmed for stop and interrupt on Terminal Count. At the end of the transfer, data is compared at the transmit and receive buffers.

Sequence of Steps

The sequence of steps for this test is as follows:

1. Initialize both DUSCCs (four ports) for asynchronous transfer as shown in the following table:

Register	Data	Description
Channel Command Register	00 h	Reset Transmitter
Channel Command Register	00 h	Reset Receiver
Channel Mode Register 2	090 h	Set Channel Connection for Local Loopback and Data Transfer for full-duplex, single-address DMA
Channel Mode Register 1	07 h	Set to asynchronous mode with NO PARITY
Transmitter Parameter Register	73 h	Set 8-bit character, 1 stop bit, EOM indicator on
Receiver Parameter Register	03 h	Set 8-bit character
Receiver Timing Register	2F h	Set the Receiver BRG at 38.4k
Transmitter Timing Register	3F h	Set the Transmitter BRG at 38.4k
Interrupt Control Register	87 h	Enable master interrupt
Channel Command Register	02 h	Enable the Transmitter
Channel Command Register	42 h	Enable the Receiver

2. Initialize the CFE Local Bus/AIB Interface Chip's DMA Transmit registers as shown in the following table:

Register	Data	Description
Channel Command Register	0x 0000 2402h	Stop and interrupt on TC=0
Memory Pointer Register	TXBUF address	Transmitter buffer address
TQC Register	00FF 0028h	Set transfer count of 39 with queuing disabled
Chain Pointer Register	0	No chain

3. Initialize the CFE Local Bus/AIB Interface Chip's DMA Receive registers as shown in the following table:

Register	Data	Description
Channel Command Register	0x 0000 2400h	Stop and interrupt on TC=0
Memory Pointer Register	RXBUF address	Receiver buffer address
TQC Register	00FF 0028h	Set transfer count of 40 with queuing disabled
Chain Pointer Register	0	No chain

4. Enable AIB interrupts in the CFE Local Bus/AIB Interface Chip by writing (0F h) to IMR register.
5. Start the transfer by enabling the CFE Local Bus/AIB Interface Chip's receive channel CCR bit 0 and the transmit channel CCR bit 0 for each port (0–3).
6. Wait for 2 milliseconds.
7. Compare Rx and Tx data buffers. Check if interrupt occurred.
8. If the data values are correct, return SUCCESS.
9. Otherwise, return the appropriate error code.

HDLC/SDLC Transfer Mode Test

This test verifies the function of the AIB in synchronous transfer mode. It also verifies the link list chaining feature of the CFE Local Bus/AIB Interface Chip.

Output Parameters

```
SUCCESS              –  Test pass
ERROR_HDLCx_TIMEOUT  –  Transfer not done on port x (0–3)
ERROR_HDLCx_DATA     –  Tx/Rx data mismatch on port x (0–3)
ERROR_HDLCx_NOCHAIN  –  CFE Local Bus/AIB Interface Chip DMA did
                        not chain on port x (0–3)
ERROR_HDLCx_APPIO    –  CFE Local Bus/AIB Interface Chip DMA
                        appended I/O error on port x (0–3)
```

Method of Verification

The HDLC transfer test programs the CFE Local Bus/AIB Interface Chip's DMA transmit channel for transfer count of 39 and the receive channel for transfer count of 40. By doing this, it verifies transmitter list chain function on Terminal Count and receiver list chain function on EOP. Chaining is verified by comparing the data at transmit buffer and receive buffer and by reading the appended I/O data in memory and DUSCC.

Sequence of Steps

The sequence of steps for this test is as follows.

1. Initialize both DUSCCs (four ports) for synchronous transfer as shown in the following table.

Register	Data	Description
Channel Command Register	00 h	Reset Transmitter
Channel Command Register	40 h	Reset Receiver
Channel Mode Register 2	090 h	Set Channel Connection for Local Loopback and Data Transfer for full-duplex, single-address DMA
Channel Mode Register 1	00 h	Set to synchronous mode with NO PARITY,BOP primary,NRZ
Transmitter Parameter Register	33 h	Set 8-bit character, EOM indicator on
Receiver Parameter Register	03 h	Set 8-bit character
Receiver Timing Register	6F h	Set the Receiver BRG at 38.4k
Transmitter Timing Register	3F h	Set the Transmitter BRG at 38.4k
Interrupt Control Register	87 h	Enable master interrupt
Output and Misc. register	E0 h	Residual char length = 8
Syn registers (SYN1 and SYN2)	AA h	Data for appended I/O
Channel Command Register	02 h	Enable the Transmitter
Channel Command Register	42 h	Enable the Receiver

2. Initialize the CFE Local Bus/AIB Interface Chip's DMA Transmitter registers as shown in the following table.

Register	Data	Description
Channel Command Register	0x 000A 0082h	Chain on TC=0, first I/O is read, second I/O is write
AIB OP2 & OP1 address register	Address of DUSCC SYN1 register	Set appended I/O address
Memory Pointer Register	TXBUF address	Transmitter buffer address
TQC Register	00FF 0027h	Set transfer count of 39 with queuing disabled
Chain Pointer Register	DummyTxCDB	Set address of dummy CDB

3. Initialize the CFE Local Bus/AIB Interface Chip's DMA Receiver registers as shown in the following table.

Register	Data	Description
Channel Command Register	0x 0006 0110h	Chain on EOP, first I/O is write, second I/O is read
AIB OP2 & OP1 address register	Address of DUSCC SYN2 register	Set appended I/O address
Memory Pointer Register	RXBUF address	Receiver buffer address
TQC Register	00FF 0028h	Set transfer count of 40 with queuing disabled
Chain Pointer Register	DummyRxCDB	Set address of dummy CDB

4. Issue the TSOM command to the DUSCC transmitter.
5. Enable AIB interrupts in the CFE Local Bus/AIB Interface Chip by writing (0F h) to IMR register.
6. Start the transfer by enabling the CFE Local Bus/AIB Interface Chip's receive channel CCR bit 0 and the transmit channel CCR bit 0 for each port (0–3).
7. Wait for 2 milliseconds.
8. Compare Rx and Tx data buffers, appended I/O values.
9. If the data and appended I/O values are correct, return SUCCESS.
10. Otherwise, return the appropriate error code.

BISYNC Transfer Mode Test

This test verifies the function of the AIB in synchronous transfer mode for BISYNC protocol. It also tests the ability of the CFE Local Bus/AIB Interface Chip to stop on TC=0 on Transmitter channel and stop on EOP for Receiver channel.

Output Parameters

```
SUCCESS              –  Test pass
ERROR_BSCx_TIMEOUT   –  Transfer not done on port x (0–3)
ERROR_BSCx_DATA      –  Tx/Rx data mismatch on port x (0–3)
ERROR_BSCx_NOSTOP    –  CFE Local Bus/AIB Interface Chip DMA did
                        not stop on port x (0–3)
ERROR_BSCx_NOINT     –  CFE Local Bus/AIB Interface Chip DMA did
                        not interrupt on port x (0–3)
```

Method of Verification

The BISYNC transfer test programs the CFE Local Bus/AIB Interface Chip's DMA transmit channel for transfer count of 39 and the receive channel for transfer count of 40. By doing this, it verifies transmitter stop and interrupt function on Terminal Count and receiver stop and interrupt function on EOP.

Sequence of Steps

The sequence of steps for this test is as follows.

1. Initialize both DUSCCs (four ports) for synchronous transfer as shown in the following table.

Register	Data	Description
Channel Command Register	00 h	Reset Transmitter
Channel Command Register	40 h	Reset Receiver
Channel Mode Register 2	90 h	Set Channel Connection for Local Loopback and Data Transfer for full-duplex, single-address DMA
Channel Mode Register 1	25 h	Set to BISYNC, ASCII, no parity, NRZ
Transmitter Parameter Register	13 h	Set 8-bit character, EOM indicator on
Receiver Parameter Register	03 h	Set 8-bit character, no sync stripping
Receiver Timing Register	6F h	Set the Receiver for BRG at 38.4k
Transmitter Timing Register	3F h	Set the Transmitter for BRG at 38.4k
Interrupt Control Register	87 h	Enable master interrupt
Output and Misc. register	00 h	Residual char length = 1
Syn registers (SYN1 and SYN2)	32 h	Sync character
Channel Command Register	02 h	Enable the Transmitter
Channel Command Register	42 h	Enable the Receiver

2. Initialize the CFE Local Bus/AIB Interface Chip’s DMA Transmitter registers as shown in the following table.

Register	Data	Description
Channel Command Register	0x 0000 2402h	Stop and interrupt on TC=0
Memory Pointer Register	TXBUF address	Transmitter buffer address
TQC Register	00FF 0027h	Set transfer count of 39 with queuing disabled
Chain Pointer Register	0	No chaining

3. Initialize the CFE Local Bus/AIB Interface Chip’s DMA Receiver registers as shown in the following table.

Register	Data	Description
Channel Command Register	0x 0000 4810h	Stop and chain on EOP
Memory Pointer Register	RXBUF address	Receiver buffer address
TQC Register	00FF 0028h	Set transfer count of 40 with queuing disabled
Chain Pointer Register	0	No chain

4. Issue the TSOM command to DUSCC transmitter.
5. Enable AIB interrupts in the CFE Local Bus/AIB Interface Chip by writing (0F h) to IMR register.

6. Start the transfer by enabling the CFE Local Bus/AIB Interface Chip's receive channel CCR bit 0 and the transmit channel CCR bit 0 for each port (0–3).
7. Wait for 2 milliseconds.
8. Compare Rx and Tx data buffers.
9. If the data values are correct, return SUCCESS.
10. Otherwise, return the appropriate error code.

DMA Start/Stop Test Using DCCR

This test verifies that the CFE Local Bus/AIB Interface Chip's DMA can be stopped, restarted, and then reset by setting appropriate bits in DCCR register for all the channels.

Output Parameters

```
SUCCESS             –  Test pass
ERROR_SS_TIMEOUT    –  Timeout on transfer start
ERROR_SS_NOSTOP     –  DMA did not stop
ERROR_SS_RESTART    –  DMA did not restart
ERROR_SS_RESET      –  DMA did not reset
```

Method of Verification

CFE Local Bus/AIB Interface Chip and DUSCC registers are programmed as per the values described for the ADMA test. The DMA transfer is started by enabling the CCR bit 0 for all the channels. The DMA stop command is issued through DCCR register for each of the channels. The stop condition is verified by reading the CCR bit 0 of each of the channels. Then DMA is restarted by issuing the start command through DCCR for each of the channels. The restart of DMA is verified by reading the CCR bit 0 of each channel. The DMA channel is reset by setting the bit in DCCR register for each of the channels. CCR bit 0 is read to verify that the channel is stopped.

Sequence of Steps

The sequence of steps for this test is as follows.

1. Initialize both DUSCCs (four ports) for asynchronous transfer as shown in the following table.

Register	Data	Description
Channel Command Register	00 h	Reset Transmitter
Channel Command Register	00 h	Reset Receiver
Channel Mode Register 2	090 h	Set Channel Connection for Local Loopback and Data Transfer for full-duplex, single-address DMA
Channel Mode Register 1	07 h	Set to asynchronous mode with NO PARITY
Transmitter Parameter Register	73 h	Set 8-bit character, 1 stop bit, EOM indicator on
Receiver Parameter Register	03 h	Set 8-bit character
Receiver Timing Register	2F h	Set the Receiver BRG at 38.4k
Transmitter Timing Register	3F h	Set the Transmitter BRG at 38.4k
Interrupt Control Register	87 h	Enable master interrupt
Channel Command Register	02 h	Enable the Transmitter
Channel Command Register	42 h	Enable the Receiver

2. Initialize the CFE Local Bus/AIB Interface Chip’s DMA Transmitter registers as shown in the following table:

Register	Data	Description
Channel Command Register	0x 0000 2402h	Stop and interrupt on TC=0
Memory Pointer Register	TXBUF address	Transmitter buffer address
TQC Register	00FF 0028h	Set transfer count of 39 with queuing disabled
Chain Pointer Register	0	No chain

3. Initialize the CFE Local Bus/AIB Interface Chip’s DMA Receiver registers as shown in the following table.

Register	Data	Description
Channel Command Register	0x 0000 2400h	Stop and interrupt on TC=0
Memory Pointer Register	RXBUF address	Receiver buffer address
TQC Register	00FF 0028h	Set transfer count of 40 with queuing disabled
Chain Pointer Register	0	No chain

4. Enable AIB interrupts in the CFE Local Bus/AIB Interface Chip by writing (0F h) to IMR register.
5. Start the transfer by enabling the CFE Local Bus/AIB Interface Chip’s receive channel CCR bit 0 and the transmit channel CCR bit 0 for each port (0–3).
6. Issue the Stop DMA command to each channel by writing 0 to the DCCR register.
7. Verify it by reading bit 0 of all the CCR registers of the CFE Local Bus/AIB Interface Chip. It should be 0. If it is not 0, return an error code.

8. Issue the Restart DMA command to each channel by writing 1 to DCCR.
9. Verify it by reading bit 0 of all the CCR registers of the CFE Local Bus/AIB Interface Chip. It should be 1. If it is not 1, return an error code.
10. Issue the Reset DMA command to each channel by writing 2 to DCCR.
11. Verify it by reading bit 0 of all the CCR registers of the CFE Local Bus/AIB Interface Chip. It should be 0. If it is not 0, return an error code.
12. Clear the reset state by writing 0 to DCCR register of all the channels.

DMA Start/Stop Test Using GDCR

This test verifies that the CFE Local Bus/AIB Interface Chip’s DMA can be stopped, restarted, and then reset globally by setting appropriate bits in GDCR register.

Output Parameters

```
SUCCESS              –  Test pass
ERROR_GLOBAL_TIMEOUT –  Timeout on transfer start
ERROR_GLOBAL_NOSTOP  –  DMA did not stop
ERROR_GLOBAL_RESTART –  DMA did not restart
ERROR_GLOBAL_RESET   –  DMA did not reset
```

Method of Verification

CFE Local Bus/AIB Interface Chip and DUSCC registers are programmed as per the values described for the ADMA test. The DMA transfer is started by enabling the CCR bit 0 for all the channels. The DMA stop command is issued through GDCR register. The stop condition is verified by reading the CCR bit 0 of each of the channels. Then DMA is restarted by issuing the start command through GDCR. The restart of DMA is verified by reading the CCR bit 0 of each channel. The DMA channel is reset by setting the bit in GDCR. CCR bit 0 is read to verify that all the channels are stopped.

Sequence of Steps

The sequence of steps for this test is as follows.

1. Initialize both DUSCCs (four ports) for asynchronous transfer as shown in the following table.

Register	Data	Description
Channel Command Register	00 h	Reset Transmitter
Channel Command Register	00 h	Reset Receiver
Channel Mode Register 2	090 h	Set Channel Connection for Local Loopback and Data Transfer for full-duplex, single-address DMA
Channel Mode Register 1	07 h	Set to asynchronous mode with NO PARITY
Transmitter Parameter Register	73 h	Set 8-bit character, 1 stop bit, EOM indicator on
Receiver Parameter Register	03 h	Set 8-bit character
Receiver Timing Register	2F h	Set the Receiver BRG at 38.4k
Transmitter Timing Register	3F h	Set the Transmitter BRG at 38.4k
Interrupt Control Register	87 h	Enable master interrupt
Channel Command Register	02 h	Enable the Transmitter
Channel Command Register	42 h	Enable the Receiver

2. Initialize the CFE Local Bus/AIB Interface Chip’s DMA Transmitter registers as shown in the following table.

Register	Data	Description
Channel Command Register	0x 0000 2402h	Stop and interrupt on TC=0
Memory Pointer Register	TXBUF address	Transmitter buffer address
TQC Register	00FF 0028h	Set transfer count of 39 with queuing disabled
Chain Pointer Register	0	No chain

3. Initialize the CFE Local Bus/AIB Interface Chip’s DMA Receiver registers as shown in the following table.

Register	Data	Description
Channel Command Register	0x 0000 2400h	Stop and interrupt on TC=0
Memory Pointer Register	RXBUF address	Receiver buffer address
TQC Register	00FF 0028h	Set transfer count of 40 with queuing disabled
Chain Pointer Register	0	No chain

4. Enable AIB interrupts in the CFE Local Bus/AIB Interface Chip by writing (0F h) to IMR register.
5. Start the transfer by enabling the CFE Local Bus/AIB Interface Chip’s receive channel CCR bit 0 and the transmit channel CCR bit 0 for each port (0–3).
6. Issue the Stop DMA command to each channel by writing 1 to GDCR register.
7. Verify it by reading bit 0 of all the CCR registers of the CFE Local Bus/AIB Interface Chip. It should be 0. If it is not 0, return an error code.

8. Issue the Restart DMA command to each channel by writing 0 to GDCR.
9. Verify it by reading bit 0 of all the CCR registers of the CFE Local Bus/AIB Interface Chip. It should be 1. If it is not 1, return an error code.
10. Issue the Reset DMA command to each channel by writing 2 to GDCR.
11. Verify it by reading bit 0 of all the CCR registers of the CFE Local Bus/AIB Interface Chip. It should be 0. If it is not 0, return an error code.
12. Clear the reset state by writing 1 to GDCR register of all the channels.

AIB ROM INIT Execution

When calling the AIB INIT routine, base ROM places the address of the base ROM Anchor in the 80960 register **g1**.

AIB INIT preserves the contents of all registers except the local registers and **g0**. AIB INIT returns its error status in register **g0** in the same format as AIBStatus word, as described under Error Reporting.

Error Reporting

Errors are classified into two types:

- Fatal Errors—errors that generally prevent the adapter from accepting the boot strap load command.
- General Errors—errors that are reported but do not prevent the adapter from accepting the boot strap load command.

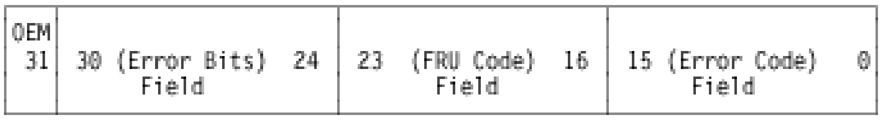
The errors are reported in the AIBStatus word.

AIBStatus Word

The AIBStatus Word has three types of error fields:

- Error Code Field
- FRU Error Field
- Error Bits Field

```
AIBStatus =  aib_stat_addr( in rom table)
          =  rom_table_addr + 0x98
           ( rom_table_addr is pointed by ROM_TABLE_ptr )
```



OEM 31	30 (Error Bits) 24 Field	23 (FRU Code) 16 Field	15 (Error Code) 0 Field

Bit 30 = 1 in AIBStatus indicates that an AIB error was detected by the Base ROM (not the AIB ROM).

Error Code Breakdown

Error codes for the 4-Port Multi-Interface AIB are shown in Table 13-31. Note that in the AIBStatus Word, bits 15–8 represent the error number and bits 7–0 represent the Port number 0–3. This same error code format can be used for similar function cards supporting up to 255 ports.

Table 13-31. AIBStatus Errors (AIB POST)

Test No.	Test Name	Error Name	Error Description	OEM Bit 31	AIB Bit 30	FATAL/ NON-FATAL Bit 29	NLOAD Bit 28	MULTI Bit 27	FRU Bits 23-16	Error Code Bits 15-0
–	SUCCESS	success	All tests pass	–	–	–	–	–	–	0000
–	FOXES	NoRun	POST did not execute	–	–	–	–	–	BASE	FFFF
–	Misc. TEST	UNKNOWN_BASE_CARD	Base card not RadiSys ARTIC960	IBM	AIB	fatal	load	single	AIB	0001
–	Misc. TEST	INVALID_INTERFACE	Invalid Interface Chip ID	IBM	AIB error	fatal	load	single	AIB	0002
–	Misc. TEST	NO_FREE_MEMORY	Insufficient memory available to execute tests	IBM	AIB error	fatal	load	single	AIB	0003
1	REG TEST	INVALID_AIB_INT	Unexpected interrupt occurred	IBM	AIB error	fatal	load	single	AIB	FFFD
1	REG TEST	VERO_REG_FAIL	CFE Local Bus/AIB Interface Chip register verification failed	IBM	AIB error	fatal	load	single	AIB	1001
1	REG TEST	DUSCC_REG_FAIL	DUSCC register verification failed	IBM	AIB error	fatal	load	single	AIB	1002
1	REG TEST	CTRL_FPGA_REG_FAIL	Control FPGA register verification failed	IBM	AIB error	fatal	load	single	AIB	1003
1	REG TEST	X21_FPGA_REG_FAIL	X21 FPGA register verification failed	IBM	AIB error	fatal	load	single	AIB	1004
2	Async Poll	ERROR_POLD0_TIMEOUT	Port 0 transfer not done	IBM	AIB error	non-fatal	load	single	AIB	3001
2	Async Poll	ERROR_POLD0_DATA	Port 0 Tx/Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	3002
2	Async Poll	ERROR_POLD1_TIMEOUT	Port 1 transfer not done	IBM	AIB error	non-fatal	load	single	AIB	3011
2	Async Poll	ERROR_POLD1_DATA	Port 1 Tx/Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	3012

Table 13-31. AIBStatus Errors (AIB POST)

Test No.	Test Name	Error Name	Error Description	OEM Bit 31	AIB Bit 30	FATAL/ NON-FATAL Bit 29	NLOAD Bit 28	MULTI Bit 27	FRU Bits 23-16	Error Code Bits 15-0
2	Async Poll	ERROR_POLD2_TIMEOUT	Port 2 transfer not done	IBM	AIB error	non-fatal	load	single	AIB	3021
2	Async Poll	ERROR_POLD2_DATA	Port 2 Tx/Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	3022
2	Async Poll	ERROR_POLD3_TIMEOUT	Port 3 transfer not done	IBM	AIB error	non-fatal	load	single	AIB	3031
2	Async Poll	ERROR_POLD3_DATA	Port 3 Tx/Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	3032
3	Async INT	ERROR_INT0_TIMEOUT	Port 0 transfer not done	IBM	AIB error	non-fatal	load	single	AIB	3101
3	Async INT	ERROR_INT0_DATA	Port 0 Tx/Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	3102
3	Async INT	ERROR_INT1_TIMEOUT	Port 1 transfer not done	IBM	AIB error	non-fatal	load	single	AIB	3111
3	Async INT	ERROR_INT1_DATA	Port 1 Tx/Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	3112
3	Async INT	ERROR_INT2_TIMEOUT	Port 2 transfer not done	IBM	AIB error	non-fatal	load	single	AIB	3121
3	Async INT	ERROR_INT2_DATA	Port 2 Tx/Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	3122
3	Async INT	ERROR_INT3_TIMEOUT	Port 3 transfer not done	IBM	AIB error	non-fatal	load	single	AIB	3131
3	Async INT	ERROR_INT3_DATA	Port 3 Tx/Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	3132
4	Async DMA	ERROR_ADMA0_TIMEOUT	Port 0 DMA transfer not done	IBM	AIB error	non-fatal	load	single	AIB	2201
4	Async DMA	ERROR_ADMA0_DATA	Port 0 DMA Tx/Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	2202

Table 13-31. AIBStatus Errors (AIB POST)

Test No.	Test Name	Error Name	Error Description	OEM Bit 31	AIB Bit 30	FATAL/ NON-FATAL Bit 29	NLOAD Bit 28	MULTI Bit 27	FRU Bits 23-16	Error Code Bits 15-0
4	Async DMA	ERROR_ADMA0_NOSTOP	Port 0 DMA transfer did not stop	IBM	AIB error	non-fatal	load	single	AIB	2203
4	Async DMA	ERROR_ADMA0_NOINT	Port 0 No DMA interrupt	IBM	AIB error	non-fatal	load	single	AIB	2204
4	Async DMA	ERROR_ADMA1_TIMEOUT	Port 1 DMA transfer not done	IBM	AIB error	non-fatal	load	single	AIB	2211
4	Async DMA	ERROR_ADMA1_DATA	Port 1 DMA Tx/Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	2212
4	Async DMA	ERROR_ADMA1_NOSTOP	Port 1 DMA transfer did not stop	IBM	AIB error	non-fatal	load	single	AIB	2213
4	Async DMA	ERROR_ADMA1_NOINT	Port 1 No DMA interrupt	IBM	AIB error	non-fatal	load	single	AIB	2214
4	Async DMA	ERROR_ADMA2_TIMEOUT	Port 2 DMA transfer not done	IBM	AIB error	non-fatal	load	single	AIB	2221
4	Async DMA	ERROR_ADMA2_DATA	Port 2 DMA Tx/Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	2222
4	Async DMA	ERROR_ADMA2_NOSTOP	Port 2 DMA transfer did not stop	IBM	AIB error	non-fatal	load	single	AIB	2223
4	Async DMA	ERROR_ADMA2_NOINT	Port 2 No DMA interrupt	IBM	AIB error	non-fatal	load	single	AIB	2224
4	Async DMA	ERROR_ADMA3_TIMEOUT	Port 3 DMA transfer not done	IBM	AIB error	non-fatal	load	single	AIB	2231
4	Async DMA	ERROR_ADMA3_DATA	Port 3 DMA Tx/Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	2232
4	Async DMA	ERROR_ADMA3_NOSTOP	Port 3 DMA transfer did not stop	IBM	AIB error	non-fatal	load	single	AIB	2233
4	Async DMA	ERROR_ADMA3_NOINT	Port 3 No DMA interrupt	IBM	AIB error	non-fatal	load	single	AIB	2234

Table 13-31. AIBStatus Errors (AIB POST)

Test No.	Test Name	Error Name	Error Description	OEM Bit 31	AIB Bit 30	FATAL/ NON-FATAL Bit 29	NLOAD Bit 28	MULTI Bit 27	FRU Bits 23-16	Error Code Bits 15-0
5	Bisync DMA	ERROR_BSC0_TIMEOUT	Port 0 BISYNC DMA transfer not done	IBM	AIB error	non-fatal	load	single	AIB	2301
5	Bisync DMA	ERROR_BSC0_DATA	Port 0 BISYNC DMA Tx/ Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	2302
5	Bisync DMA	ERROR_BSC0_NOSTOP	Port 0 DMA transfer did not stop	IBM	AIB error	non-fatal	load	single	AIB	2303
5	Bisync DMA	ERROR_BSC0_NOINT	Port 0 No DMA interrupt	IBM	AIB error	non-fatal	load	single	AIB	2304
5	Bisync DMA	ERROR_BSC1_TIMEOUT	Port 1 BISYNC DMA transfer not done	IBM	AIB error	non-fatal	load	single	AIB	2311
5	Bisync DMA	ERROR_BSC1_DATA	Port 1 BISYNC DMA Tx/ Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	2312
5	Bisync DMA	ERROR_BSC1_NOSTOP	Port 1 DMA transfer did not stop	IBM	AIB error	non-fatal	load	single	AIB	2313
5	Bisync DMA	ERROR_BSC1_NOINT	Port 1 No DMA interrupt	IBM	AIB error	non-fatal	load	single	AIB	2314
5	Bisync DMA	ERROR_BSC2_TIMEOUT	Port 2 BISYNC DMA transfer not done	IBM	AIB error	non-fatal	load	single	AIB	2321
5	Bisync DMA	ERROR_BSC2_DATA	Port 2 BISYNC DMA Tx/ Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	2322
5	Bisync DMA	ERROR_BSC2_NOSTOP	Port 2 DMA transfer did not stop	IBM	AIB error	non-fatal	load	single	AIB	2323
5	Bisync DMA	ERROR_BSC2_NOINT	Port 2 No DMA interrupt	IBM	AIB error	non-fatal	load	single	AIB	2324
5	Bisync DMA	ERROR_BSC3_TIMEOUT	Port 3 BISYNC DMA transfer not done	IBM	AIB error	non-fatal	load	single	AIB	2331
5	Bisync DMA	ERROR_BSC3_DATA	Port 3 BISYNC DMA Tx/ Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	2332

Table 13-31. AIBStatus Errors (AIB POST)

Test No.	Test Name	Error Name	Error Description	OEM Bit 31	AIB Bit 30	FATAL/ NON-FATAL Bit 29	NLOAD Bit 28	MULTI Bit 27	FRU Bits 23-16	Error Code Bits 15-0
5	Bisync DMA	ERROR_BSC3_NOSTOP	Port 3 DMA transfer did not stop	IBM	AIB error	non-fatal	load	single	AIB	2333
5	Bisync DMA	ERROR_BSC3_NOINT	Port 3 No DMA interrupt	IBM	AIB error	non-fatal	load	single	AIB	2334
6	HDLC DMA	ERROR_HDLC0_TIMEOUT	Port 0 HDLC DMA transfer not done	IBM	AIB error	non-fatal	load	single	AIB	2401
6	HDLC DMA	ERROR_HDLC0_DATA	Port 0 HDLC DMA Tx/Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	2402
6	HDLC DMA	ERROR_HDLC0_NOCHAIN	Port 0 DMA did not chain	IBM	AIB error	non-fatal	load	single	AIB	2405
6	HDLC DMA	ERROR_HDLC0_APPIO	Port 0 Appended I/O error	IBM	AIB error	non-fatal	load	single	AIB	2406
6	HDLC DMA	ERROR_HDLC1_TIMEOUT	Port 1 HDLC DMA transfer not done	IBM	AIB error	non-fatal	load	single	AIB	2411
6	HDLC DMA	ERROR_HDLC1_DATA	Port 1 HDLC DMA Tx/Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	2412
6	HDLC DMA	ERROR_HDLC1_NOCHAIN	Port 1 DMA did not chain	IBM	AIB error	non-fatal	load	single	AIB	2415
6	HDLC DMA	ERROR_HDLC1_APPIO	Port 1 Appended I/O error	IBM	AIB error	non-fatal	load	single	AIB	2416
6	HDLC DMA	ERROR_HDLC2_TIMEOUT	Port 2 HDLC DMA transfer not done	IBM	AIB error	non-fatal	load	single	AIB	2421
6	HDLC DMA	ERROR_HDLC2_DATA	Port 2 HDLC DMA Tx/Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	2422
6	HDLC DMA	ERROR_HDLC2_NOCHAIN	Port 2 DMA did not chain	IBM	AIB error	non-fatal	load	single	AIB	2425
6	HDLC DMA	ERROR_HDLC2_APPIO	Port 2 Appended I/O error	IBM	AIB error	non-fatal	load	single	AIB	2426

Table 13-31. AIBStatus Errors (AIB POST)

Test No.	Test Name	Error Name	Error Description	OEM Bit 31	AIB Bit 30	FATAL/ NON-FATAL Bit 29	NLOAD Bit 28	MULTI Bit 27	FRU Bits 23-16	Error Code Bits 15-0
6	HDLC DMA	ERROR_HDLC3_TIMEOUT	Port 3 HDLC DMA transfer not done	IBM	AIB error	non-fatal	load	single	AIB	2431
6	HDLC DMA	ERROR_HDLC3_DATA	Port 3 HDLC DMA Tx/Rx data mismatch	IBM	AIB error	non-fatal	load	single	AIB	2432
6	HDLC DMA	ERROR_HDLC3_NOCHAIN	Port 3 DMA did not chain	IBM	AIB error	non-fatal	load	single	AIB	2435
6	HDLC DMA	ERROR_HDLC3_APPIO	Port 3 Appended I/O error	IBM	AIB error	non-fatal	load	single	AIB	2436
7	Stop/Start DMA	ERROR_SS_TIMEOUT	Timeout on transfer start	IBM	AIB error	non-fatal	load	single	AIB	2501
7	Stop/Start DMA	ERROR_SS_NOSTOP	DMA did not stop	IBM	AIB error	non-fatal	load	single	AIB	2503
7	Stop/Start DMA	ERROR_SS_RESTART	DMA did not restart	IBM	AIB error	non-fatal	load	single	AIB	2507
7	Stop/Start DMA	ERROR_SS_RESET	DMA did not reset	IBM	AIB error	non-fatal	load	single	AIB	2508
8	Global DMA	ERROR_GLOBAL_TIMEOUT	Timeout on transfer start	IBM	AIB error	non-fatal	load	single	AIB	2601
8	Global DMA	ERROR_GLOBAL_NOSTOP	DMA did not stop	IBM	AIB error	non-fatal	load	single	AIB	2603
8	Stop/Start DMA	ERROR_GLOBAL_RESTART	DMA did not restart	IBM	AIB error	non-fatal	load	single	AIB	2607
8	Global DMA	ERROR_GLOBAL_RESET	DMA did not reset	IBM	AIB error	non-fatal	load	single	AIB	2608

Specifications Relevant to Supported Interfaces

Table 13-32 lists the signal set, electrical, and mechanical characteristics of the communication interfaces supported by the 4-Port Multi-Interface AIB card and the AIB cable assemblies.

The specifications shown in parentheses are equivalent specifications from other sources.

Table 13-32. Communication Standards Specifications

Interface	Signal Set Supported	Electrical Interface	Mechanical Interface
EIA-232-D	EIA-232-D (X.24)	EIA-232-D (V.10, X.26)	EIA-232-D (ISO 2110)
EIA-530	EIA-530	EIA-422-A	EIA-530
X.21	X.21	X.27 (V.11)	ISO 4903
V.36	V.36	V.10 (X.26) and V.11 (X.27)	ISO 4902

Acronym Glossary

A

AIB:	Application interface board
ASIC:	Application-specific integrated circuit
BOP:	Bit-oriented protocol
CCR:	Channel Control Register
CDB:	Channel Descriptor Block
CDT:	Channel Descriptor Table
CFE:	Common front end
CMOS:	Complementary metal-oxide semiconductor
COP:	Character-oriented protocol
CPR:	Chain Pointer Register
CPU:	Central processing unit
CSU:	Channel service unit
DACK:	DMA acknowledgment
DCL:	Daughter card local
DMA:	Direct memory access
DRAM:	Dynamic random-access memory
DREQ:	DMA request
DSU:	Data service unit
EPLD:	Electrically programmable logic device
EPROM:	Erasable programmable read-only memory
I/O:	Input/output
IOCHCK:	I/O channel check
KB:	Kilobyte
LEPB:	Low-end parallel bus
LLC:	Linked list chaining
LSSD:	Level-sensitive scan design
MB:	Megabyte
MB/s:	Megabytes-per-second
MC:	Micro Channel

MMIO:	Memory-mapped I/O
MPR:	Memory Pointer Register
NMI:	Non-maskable interrupt
PCI:	Peripheral Component Interface
POS:	Programmable option select
POST:	Power-on self-test
RAM:	Random access memory
ROM:	Read-only memory
ROS:	Read-only storage
RR:	Receive request
SCB:	Subsystem Control Block
SIMM:	Single in-line memory module
SOJ:	Small outline J-lead
SRAM:	Static random access memory
TR:	Transmit request
TQC:	Transfer Count Register
VLSI:	Very-large-scale integration
VPD:	Vital product data
ZIP:	Zig-zag in-line package

Co-Processor Platform Registers

B

Sorted by Abbreviation

Table B-1. Co-Processor Registers (Sorted by Abbreviation)

Abbrev	Register Name	Address	Type	See Page
ATTN[2]	SCB Attention Port (uCh)	1FFA 2004h	ro	107
ATTN[6]	SCB Attention Port (PCI)	1FFA 2004h	ro	157
AWE	AIB ROM Write Enable Register	09FF FFF4h	rw	167
BCR[3]	Bus Master Channel x Byte Count Register	1FFA x008h	r/w	93 (uCh) 141 (PCI)
BMAR[3]	Bus Master Channel x Address Register	1FFA x010h	r/w	95 (uCh) 143 (PCI)
BMCMD[3]	Bus Master Channel x Command Register	1FFA x01Ch	r/w	99 (uCh) 146 (PCI)
BMSTAT[3]	Bus Master Channel x Status Register	1FFA x018h	ro	97 (uCh) 144 (PCI)
BWE	Base ROM Write Enable Register	09FF FFF0h	rw	168
CAR[3]	Bus Master Channel x Card Address Register	1FFA x000h	r/w	92 (uCh) 140 (PCI)
CBSP[2]	SCB Command/Busy Status Port (uCh)	1FFA 2010h	r/w	112
CBSP[6]	SCB Command/Busy Status Port (PCI)	1FFA 2010h	r/w	161
CCR[3]	Bus Master Channel x Control Register	1FFA x00Ch	r/w	93 (uCh) 141 (PCI)
CCRID[7]	Class Code/Revision ID Reg	1FFA 5008h	ro	126
CFEBAR	CFE Base Address Register	14 – 17h	ro	155
COMMAND[2]	SCB Command Port (uCh)	1FFA 2000h	ro	156
COMMAND[6]	SCB Command Port (PCI)	1FFA 2000h	ro	107
CONF1[1]	Configuration Register 1	14 – 17h	r/o	77
CONF2[1]	Configuration Register 2	18 – 1Bh	r/o	77
CONF3[1]	Configuration Register 3	1C – 1Fh	r/o	78
CRDID	Card ID Register	1FFA 000Ch	r/w	82
DEVID[7]	Device/Vendor ID Register	1FFA 5000h	ro	124
ECCATR	ECC Address Trap Register	1FFB 9034h	ro	35
ECCSTAT	ECC Status Register	1FFB 9038h	ro	35
EDR	Enable/Detect Register	0A00 0004h	r/w	46
FEER	Force ECC Error Register	1FFB A010h	r/w	36
GAID[2]	Gate Array ID Register (uCh)	1FFA 0008h	ro	169
GAID[6]	Gate Array ID Register (PCI)	1FFA 0008h	ro	169

Table B-1. Co-Processor Registers (Sorted by Abbreviation)

Abbrev	Register Name	Address	Type	See Page
GAIDR	Memory Controller Gate Array ID Register	1FFB A000h	ro	168
HMBAR[7]	Host Memory Base Address Register	1FFA 5014h	ro	128
HMFR[7]	Host Misc Function Register	1FFA 500Ch	ro	127
HSBR[1]	Host-Slave Base Address Register (uCh)	0C – 0Fh	r/w	103
HSBR[5]	Host-Slave Base Address Register (PCI)	0C – 0Fh	r/w	153
HSCR[7]	Host Status/Command Register	1FFA 5004h	ro	125
IOBAR[7]	I/O Base Address Register	1FFA 5010h	ro	128
ISP[2]	SCB Interrupt Status Port (uCh)	1FFA 200Ch	r/w	111
ISP[6]	SCB Interrupt Status Port (PCI)	1FFA 200Ch	r/w	159
IVR	Interrupt Controller Version Register	0A00 0000h	ro	170
LAP[3]	Bus Master Channel x List Address Pointer	1FFA x014h	r/w	96 (uCh) 144 (PCI)
LBBAR	Local Bus Base Address Register	1FFA 0024h	r/w	102 (uCh) 149 (PCI)
LBCFG	Local Bus Configuration Register	1FFB A00Ch	r/w	62
LBPE	Local Bus Parity/Exception Register	1FFA 0020h	ro	58
LEXATR	Exception Address Trap Register	1FFB 9020h	ro	59
LEXSTAT	Exception Status Register	1FFB 9024h	ro	60
LGIR[7]	Latency/ Grant/ Interrupt Register	1FFA 503Ch	ro	131
LPATR	Local Bus Protection Address Trap Register	1FFB 9018h	ro	31
LPSTAT	Local Bus Protection Status Register	1FFB 901Ch	ro	31
MCR	Memory Configuration Register	1FFB A004h	r/w	171
MDATA[1]	Memory Data Register (uCh)	10 – 13h	r/w	103
MDATA[5]	Memory Data Register (PCI)	10 – 13h	r/w	154
MMBAR[7]	Memory Mapped I/O Base Address Register	1FFA 5018h	ro	129
MPER	Memory Protection Enable Register	1FFB 9000h	r/w	27
MPH	Presence Detect High Register	0A00 000Ch	ro	171
MPL	Presence Detect Low Register	0A00 0008h	ro	170
MPTBR	MC Page Table Base Register	1FFB 9008h	r/w	29
NMI[2]	NMI Command Register (uCh)	1FFA 001Ch	ro	114
NMI[6]	NMI Command Register (PCI)	1FFA 001Ch	ro	164
PCR	Port Configuration Register	1FFB 8008h	r/w	176
POS_SETUP1	POS Setup 1 Register	1FFA 0000h	ro	70
POS_SETUP2	POS Setup 2 Register	1FFA 0004h	ro	74
PROC_CFG	Processor Configuration Register (uCh)	1FFA 0010h	r/w	78
PROC_CFG	Processor Configuration Register (PCI)	1FFA 0010h	r/w	80
RSR	Reset Status Register	1FFA 0014h	ro	114 (uCh) 163 (PCI)
RXBUF	Rx Buffer Port	1FFB 8004h	r/w	175

Table B-1. Co-Processor Registers (Sorted by Abbreviation)

Abbrev	Register Name	Address	Type	See Page
SAR[3]	Bus Master Channel x System Address Register	1FFA x004h	r/w	92 (uCh) 140 (uCh)
SBATR	Single-Bit ECC Address Trap Register	1FFB 902Ch	ro	34
SBCR[3]	System Byte Count Channel X Register (PCI)	1FFA x020h	ro	147
SBECCR	Single-Bit ECC Error Register	1FFB 9030h	r/w	35
SCP[2]	SCB Subsystem Control Port (uCh)	1FFA 2008h	ro	110
SCP[6]	SCB Subsystem Control Port (PCI)	1FFA 2008h	ro	158
SEER	Serial EPROM Extension Register	1FFA0028h	r/w	123
SIR[2]	Source Identification Register (uCh)	1FFA 2014h	r/w	113
SIR[6]	Source Identification Register (PCI)	1FFA 2014h	r/w	162
SPAR	Special Arbitration Register	1FFB A008h	r/w	62
SSID[7]	Sub System ID Register	1FFB 502Ch	r/w	130
TCMD	Timer Command Registers	1FFB x00Ch	r/w	41
TCR	Timer Control Registers	1FFB x000h	r/w	39
TPATR	Task Protection Address Trap Register	1FFB 9010h	ro	30
TPSTAT	Task Protection Status Register	1FFB 9014h	ro	30
TPR	Timer Preset Registers	1FFB x004h	r/w	40
TPTBR	Task Page Table Base Register	1FFB 9004h	r/w	29
TPV	Timer Present Value Registers	1FFB x008h	ro	40
TXBUF	Tx Buffer Port	1FFB 8000h	r/w	174
XPOS	Extended POS Register	1FFA 0018h	r/w	81
XRBAR[7]	Expansion ROM Base Address Register	1FFA 5030h	ro	130

1 Offset from Base I/O address. See POS 5 and 3A for a description of the Base I/O address.

2 This register is also accessible from the Micro Channel. See the register description for the Micro Channel address.

3 For Bus Master Channel 1, ‘x’ = 3; for Bus Master Channel 2, ‘x’ = 4.

4 The term ‘ro’ is an abbreviation for read-only; the term ‘r/w’ is an abbreviation for read/write.

5 Offset from Base address for Memory and I/O access. See the MMBAR and IOBAR register descriptions for the PCI address.

6 This register is also accessible in PCI Memory space and in PCI IO space. See the register description for these address offsets.

7 This register is also accessible in PCI Memory space and in PCI Configuration space. See the register description for these address offsets.

Sorted by Address

Table B-2. Co-Processor Registers (Sorted by Address)

Address	Abbrev	Register Name	Type	See Page
0C – 0Fh	HSBR[1]	Host-Slave Base Address Register (uCh)	r/w	103
0C – 0Fh	HSBR[5]	Host-Slave Base Address Register (PCI)	r/w	153
10 – 13h	MDATA[1]	Memory Data Register (uCh)	r/w	103
10 – 13h	MDATA[5]	Memory Data Register (PCI)	r/w	154
14 – 17h	CONF1[1]	Configuration Register 1[1]	r/o	77
14h – 17h	CFEBAR[5]	CFE Base Address Register (PCI)	ro	155
18 – 1Bh	CONF2[1]	Configuration Register 2	r/o	77
1C – 1Fh	CONF3[1]	Configuration Register 3	r/o	78
09FF FFF0h	BWE	Base ROM Write Enable Register	rw	168
09FF FFF4h	AWE	AIB ROM Write Enable Register	rw	167
0A00 0000h	IVR	Interrupt Controller Version Register	ro	170
0A00 0004h	EDR	Enable/Detect Register	r/w	46
0A00 0008h	MPL	Presence Detect Low Register	ro	170
0A00 000Ch	MPH	Presence Detect High Register	ro	171
1FFA 0000h	POSSET1	POS Setup 1 Register	ro	70
1FFA 0004h	POSSET2	POS Setup 2 Register	ro	74
1FFA 0008h	GAID[2]	Gate Array ID Register (uCh)	ro	169
1FFA 0008h	GAID[6]	Gate Array ID Register (PCI)	ro	169
1FFA 000Ch	CRDID	Card ID Register	r/w	82
1FFA 0010h	PROC_CFG	Processor Configuration Register (uCh)	r/w	78
1FFA 0010h	PROC_CFG	Processor Configuration Register (PCI)	r/w	80
1FFA 0014h	RSR	Reset Status Register	ro	114 (uCh) 163 (PCI)
1FFA 0018h	XPOS	Extended POS Register	r/w	81
1FFA 001Ch	NMI[2]	NMI Command Register (uCh)	ro	114
1FFA 001Ch	NMI[6]	NMI Command Register (PCI)	ro	164
1FFA 0020h	LBPE	Local Bus Parity/Exception Register	ro	58
1FFA 0024h	LBBAR	Local Bus Base Address Register	r/w	102 (uCh) 149 (PCI)
1FFA 0028h	SEER	Serial EPROM Extension Register	r/w	123
1FFA 2000h	COMMAND[2]	SCB Command Port (uCh)	ro	107
1FFA 2000h	COMMAND[6]	SCB Command Port (PCI)	ro	156
1FFA 2004h	ATTN[2]	SCB Attention Port (uCh)	ro	107
1FFA 2004h	ATTN[6]	SCB Attention Port (PCI)	ro	157
1FFA 2008h	SCP[2]	SCB Subsystem Control Port (uCh)	ro	110
1FFA 2008h	SCP[6]	SCB Subsystem Control Port (PCI)	ro	158
1FFA 200Ch	ISP[2]	SCB Interrupt Status Port (uCh)	r/w	111
1FFA 200Ch	ISP[6]	SCB Interrupt Status Port (PCI)	r/w	159
1FFA 2010h	CBSP[2]	SCB Command/Busy Status Port (uCh)	r/w	112
1FFA 2010h	CBSP[6]	SCB Command/Busy Status Port (PCI)	r/w	161

Table B-2. Co-Processor Registers (Sorted by Address)

Address	Abbrev	Register Name	Type	See Page
1FFA 2014h	SIR[2]	Source Identification Register (uCh)	r/w	113
1FFA 2014h	SIR[6]	Source Identification Register (PCI)	r/w	162
1FFA x000h	CAR[3]	Bus Master Channel x Card Address Register	r/w	92
1FFA x004h	SAR[3]	Bus Master Channel x System Address Register	r/w	92 (uCh) 140 (PCI)
1FFA x008h	BCR[3]	Bus Master Channel x Byte Count Register	r/w	93 (uCh) 141 (PCI)
1FFA x010h	BMAR[3]	Bus Master Channel x Address Register	r/w	95 (uCh) 143 (PCI)
1FFA x00Ch	CCR[3]	Bus Master Channel x Control Register	r/w	93 (uCh) 141 (PCI)
1FFA x014h	LAP[3]	Bus Master Channel x List Address Pointer	r/w	96 (uCh) 144 (PCI)
1FFA x018h	BMSTAT[3]	Bus Master Channel x Status Register	ro	97 (uCh) 144 (PCI)
1FFA x01Ch	BMCMD[3]	Bus Master Channel x Command Register	r/w	146
1FFA x020h	SBCR[3]	System Byte Count Channel X Register (PCI)	ro	147
1FFA 5000h	DEVID[7]	Device/Vendor ID Register	ro	124
1FFA 5004h	HSCR[7]	Host Status/Command Register	ro	125
1FFA 5008h	CCRID[7]	Class Code/Revision ID Reg	ro	126
1FFA 500Ch	HMFR[7]	Host Misc Function Register	ro	127
1FFA 5010h	IOBAR[7]	I/O Base Address Register	ro	128
1FFA 5014h	HMBAR[7]	Host Memory Base Address Register	ro	128
1FFA 5018h	MMBAR[7]	Memory Mapped I/O Base Address Register	ro	129
1FFB 502Ch	SSID[7]	Sub System ID Register	r/w	130
1FFA 5030h	XRBAR[7]	Expansion ROM Base Address Register	ro	130
1FFA 503Ch	LGIR[7]	Latency/Grant/Interrupt Register	ro	131
1FFB 8000h	TXBUF	Tx Buffer Port	r/w	174
1FFB 8004h	RXBUF	Rx Buffer Port	r/w	175
1FFB 8008h	PCR	Port Configuration Register	r/w	176
1FFB 9000h	MPER	Memory Protection Enable Register	r/w	27
1FFB 9004h	TPTBR	Task Page Table Base Register	r/w	29
1FFB 9008h	MPTBR	MC Page Table Base Register	r/w	29
1FFB 9010h	TPATR	Task Protection Address Trap Register	ro	30
1FFB 9014h	TPSTAT	Task Protection Status Register	ro	30
1FFB 9018h	LPATR	Local Bus Protection Address Trap Register	ro	31
1FFB 901Ch	LPSTAT	Local Bus Protection Status Register	ro	31
1FFB 9020h	LEXATR	Exception Address Trap Register	ro	59
1FFB 9024h	LEXSTAT	Exception Status Register	ro	60

Table B-2. Co-Processor Registers (Sorted by Address)

Address	Abbrev	Register Name	Type	See Page
1FFB 902Ch	SBATR	Single-Bit ECC Address Trap Register	ro	34
1FFB 9030h	SBECCR	Single-Bit ECC Error Register	r/w	35
1FFB 9034h	ECCATR	ECC Address Trap Register	ro	35
1FFB 9038h	ECCSTAT	ECC Status Register	ro	35
1FFB A000h	GAIDR	Memory Controller Gate Array ID Register	ro	168
1FFB A004h	MCR	Memory Configuration Register	r/w	171
1FFB A008h	SPAR	Special Arbitration Register	r/w	62
1FFB A00Ch	LBCFG	Local Bus Configuration Register	r/w	62
1FFB A010h	FEER	Force ECC Error Register	r/w	36
1FFB x000h	TCR	Timer Control Registers	r/w	39
1FFB x004h	TPR	Timer Preset Registers	r/w	40
1FFB x008h	TPV	Timer Present Value Registers	ro	40
1FFB x00Ch	TCMD	Timer Command Registers	r/w	41

1 Offset from Base I/O address. See POS 5 and 3A for a description of the Base I/O address.

2 This register is also accessible from the Micro Channel. See the register description for the Micro Channel address.

3 For Bus Master Channel 1, ‘x’ = 3; for Bus Master Channel 2, ‘x’ = 4.

4 The term ‘ro’ is an abbreviation for read-only; the term ‘r/w’ is an abbreviation for read/write.

5 Offset from Base address for Memory and I/O access. See the MMBAR and IOBAR register descriptions for the PCI address.

6 This register is also accessible in PCI Memory space and in PCI IO space. See the register description for these address offsets.

7 This register is also accessible in PCI Memory space and in PCI Configuration space. See the register description for these address offsets.

4-Port Multi-Interface AIB Registers

C

Sorted by Abbreviation

The following list does not include the DUSCC registers listed in Table 13-19 and Table 13-20.

Abbreviation	Register Name	CFE Bus Address (Hex)	Access Type	See page
ACSR	DCL Command/Status	1FF8 C000	r/w	275
AIB_ADDR 1/2	DMA Channel 'x' AIB_ADDR 1/2	1FF8 x000	r/w	260
AIB_OP1	DMA Channel 'x' AIB_OP1 Data	1FF8 x004	r/w	261
AIB_OP2	DMA Channel 'x' AIB_OP2 Data	1FF8 x008	r/w	261
CCR	DMA Channel 'x' Channel Control	1FF8 x018	r/w	262
CPR	DMA Channel 'x' Chain Pointer	1FF8 x014	r/w	265
CSD0	Chip Select Definition 0	1FF8 B000	r/w	276
CSD1	Chip Select Definition 1	1FF8 B004	r/w	276
CSD2	Chip Select Definition 2	1FF8 B008	r/w	276
CSD3	Chip Select Definition 3	1FF8 B00C	r/w	276
CSD4	Chip Select Definition 4	1FF8 B010	r/w	276
DAPW0	DMA Acknowledge Pulse Width 0	1FF8 B030	r/w	279
DAPW1	DMA Acknowledge Pulse Width 1	1FF8 B034	r/w	279
DAPW2	DMA Acknowledge Pulse Width 2	1FF8 B038	r/w	279
DAPW3	DMA Acknowledge Pulse Width 3	1FF8 B03C	r/w	279
DAPW4	DMA Acknowledge Pulse Width 4	1FF8 B040	r/w	279
DAPW5	DMA Acknowledge Pulse Width 5	1FF8 B044	r/w	279
DAPW6	DMA Acknowledge Pulse Width 6	1FF8 B048	r/w	279
DAPW7	DMA Acknowledge Pulse Width 7	1FF8 B04C	r/w	279
DCCR	DMA Channel 'x' Command	1FF8 x020	r/w	266
DFRC	DMA Channel 'x' FIFO Residual Count	1FF8 x02C	ro	266
DISR	DMA Channel 'x' Interrupt Status	1FF8 x028	ro	267
DVREN0	Driver Enable 0	1FF1 0000	r/w	283
DVREN1	Driver Enable 1	1FF1 1000	r/w	283
DVREN2	Driver Enable 2	1FF1 2000	r/w	283
DVREN3	Driver Enable 3	1FF1 3000	r/w	283
EOI0	AIB INT0 End-of-Interrupt	1FF8 9000	wo (command)	271
EOI1	AIB INT1 End-of-Interrupt	1FF8 A000	wo (command)	271

Abbreviation	Register Name	CFE Bus Address (Hex)	Access Type	See page
GDCR	Global DMA Command	1FF8 8000	r/w	268
IIR	Interrupt Initialization	1FF8 8010	r/w	271
IMR	Interrupt Mask	1FF8 8014	r/w	274
INTEN0	Interrupt Enable 0	1FF1 0800	r/w	284
INTEN1	Interrupt Enable 1	1FF1 1800	r/w	284
INTEN2	Interrupt Enable 2	1FF1 2800	r/w	284
INTEN3	Interrupt Enable 3	1FF1 3800	r/w	284
ISR	Interrupt Status	1FF8 8018	ro	275
LBCR	CFE Local Bus Configuration	1FF8 8008	r/w	269
LER	LED Enable	1FF8 D004	r/w	281
MPR	DMA Channel 'x' Memory Pointer	1FF8 x00C	r/w	269
PDR	Presence Detect	1FF8 D000	ro	281
TQC	DMA Channel 'x' Transfer Count	1FF8 x010	r/w	270
X21EN0	X.21 Enable 0	1FF2 0800	r/w	286
X21EN1	X.21 Enable 1	1FF2 1800	r/w	286
X21EN2	X.21 Enable 2	1FF2 2800	r/w	286
X21EN3	X.21 Enable 3	1FF2 3800	r/w	286
X21ST0	X.21 Pattern Status 0	1FF2 0000	ro	287
X21ST1	X.21 Pattern Status 1	1FF2 1000	ro	287
X21ST2	X.21 Pattern Status 2	1FF2 2000	ro	287
X21ST3	X.21 Pattern Status 3	1FF2 3000	ro	287

Sorted by Address

The following list does not include the DUSCC registers listed in Table 13-19 and Table 13-20.

CFE Bus Address (Hex)	Abbreviation	Register Name	Access Type	See Page
1FF1 0000	DVREN0	Driver Enable 0	r/w	286
1FF1 0800	INTEN0	Interrupt Enable 0	r/w	284
1FF1 1000	DVREN1	Driver Enable 1	r/w	283
1FF1 1800	INTEN1	Interrupt Enable 1	r/w	284
1FF1 2000	DVREN2	Driver Enable 2	r/w	283
1FF1 2800	INTEN2	Interrupt Enable 2	r/w	284
1FF1 3000	DVREN3	Driver Enable 3	r/w	283
1FF1 3800	INTEN3	Interrupt Enable 3	r/w	284
1FF2 0000	X21ST0	X.21 Pattern Status 0	ro	287
1FF2 0800	X21EN0	X.21 Enable 0	r/w	286
1FF2 1000	X21ST1	X.21 Pattern Status 1	ro	287
1FF2 1800	X21EN1	X.21 Enable 1	r/w	286
1FF2 2000	X21ST2	X.21 Pattern Status 2	ro	287
1FF2 2800	X21EN2	X.21 Enable 2	r/w	286
1FF2 3000	X21ST3	X.21 Pattern Status 3	ro	287
1FF2 3800	X21EN3	X.21 Enable 3	r/w	286
1FF8 x000	AIB_ADDR 1/2	DMA Channel 'x' AIB_ADDR 1/2	r/w	260
1FF8 x004	AIB_OP1	DMA Channel 'x' AIB_OP1 Data	r/w	261
1FF8 x008	AIB_OP2	DMA Channel 'x' AIB_OP2 Data	r/w	261
1FF8 x00C	MPR	DMA Channel 'x' Memory Pointer	r/w	269
1FF8 x010	TQC	DMA Channel 'x' Transfer Count	r/w	270
1FF8 x014	CPR	DMA Channel 'x' Chain Pointer	r/w	265
1FF8 x018	CCR	DMA Channel 'x' Channel Control	r/w	262
1FF8 x01C		Reserved		
1FF8 x020	DCCR	DMA Channel 'x' Command	r/w	266
1FF8 x024		Reserved		
1FF8 x028	DISR	DMA Channel 'x' Interrupt Status	ro	267
1FF8 x02C	DFRC	DMA Channel 'x' FIFO Residual Count	ro	266
1FF8 8000	GDCR	Global DMA Command	r/w	268
1FF8 8004		Reserved		
1FF8 8008	LBCR	CFE Local Bus Configuration	r/w	269
1FF8 800C		Reserved		
1FF8 8010	IIR	Interrupt Initialization	r/w	271
1FF8 8014	IMR	Interrupt Mask	r/w	274
1FF8 8018	ISR	Interrupt Status	ro	275
1FF8 9000	EOI0	AIB INT0 End-of-Interrupt	wo (command)	271
1FF8 A000	EOI1	AIB INT1 End-of-Interrupt	wo (command)	271

CFE Bus Address (Hex)	Abbreviation	Register Name	Access Type	See Page
1FF8 B000	CSD0	Chip Select Definition 0	r/w	276
1FF8 B004	CSD1	Chip Select Definition 1	r/w	276
1FF8 B008	CSD2	Chip Select Definition 2	r/w	276
1FF8 B00C	CSD3	Chip Select Definition 3	r/w	276
1FF8 B010	CSD4	Chip Select Definition 4	r/w	276
1FF8 B030	DAPW0	DMA Acknowledge Pulse Width 0	r/w	279
1FF8 B034	DAPW1	DMA Acknowledge Pulse Width 1	r/w	279
1FF8 B038	DAPW2	DMA Acknowledge Pulse Width 2	r/w	279
1FF8 B03C	DAPW3	DMA Acknowledge Pulse Width 3	r/w	279
1FF8 B040	DAPW4	DMA Acknowledge Pulse Width 4	r/w	279
1FF8 B044	DAPW5	DMA Acknowledge Pulse Width 5	r/w	279
1FF8 B048	DAPW6	DMA Acknowledge Pulse Width 6	r/w	279
1FF8 B04C	DAPW7	DMA Acknowledge Pulse Width 7	r/w	279
1FF8 C000	ACSR	DCL Command/Status	r/w	275
1FF8 D000	PDR	Presence Detect	ro	281
1FF8 D004	LER	LED Enable	r/w	281

‘*x*’ = DMA channel number 0 to 7.

D Notices

Index