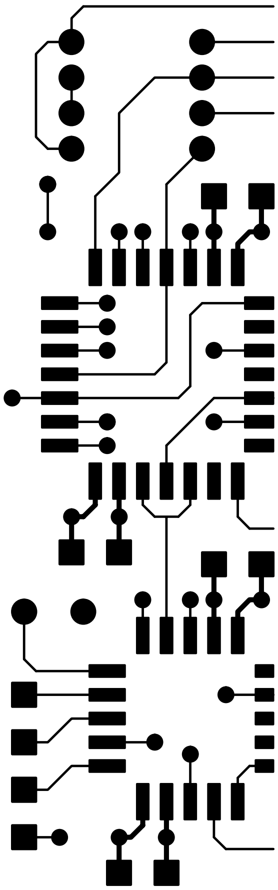
PowerPC

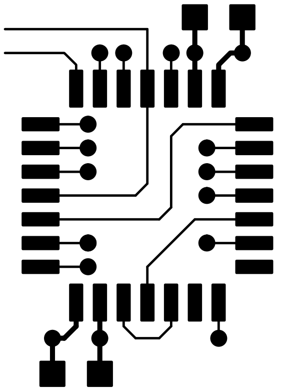
PowerPC™ 604 SMP Reference Design Technical Specification

Release 3.0

This document provides a detailed technical description of the PowerPC 604 SMP Reference Design. It is intended as a first source of information for both hardware and software designers. Where appropriate, other documents are referenced.

Document Number:
MPRZAPTSU-03
December 1995





IBM®

This document contains information which is subject to change by IBM without notice. IBM assumes no responsibility or liability for any use of the information contained herein. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of IBM or third parties. The products described in this document are not intended for use in implantation or other direct life-support applications where malfunction may result in physical harm or injury to persons. NO WARRANTIES OF ANY KIND, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, ARE OFFERED IN THIS DOCUMENT.

Contacts

IBM Microelectronics Division
1580 Route 52, Bldg. 504
Hopewell Junction, NY 12533-6531
Tel: (800) PowerPC
Fax: (800) PowerFax

http://www.chips.ibm.com
http://www.ibm.com
ftp://ftp.austin.ibm.com/pub/PPC_support

ESD Warning

The motherboard, CPU, and memory cards contain CMOS devices which are very susceptible to ElectroStatic Discharge (ESD). DO NOT remove them from the antistatic bags until you have connected yourself to an acceptable ESD grounding strap. Work in a static free environment and be sure any person or equipment coming into contact with the cards does not have a static charge. The cards are particularly susceptible until they are placed in a properly designed enclosure. Bench work should be done by persons connected to ESD grounding straps.

IBM POWERPC 604™ SMP REFERENCE DESIGN AGREEMENT

BEFORE READING THE REST OF THE DOCUMENT, YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS. OPENING THE PACKAGE INDICATES YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THEM, YOU SHOULD PROMPTLY RETURN THE PACKAGE UNOPENED TO YOUR IBM SALES OFFICE.

International Business Machines Corporation (”IBM”) agrees to provide you a PowerPC 604 SMP Reference Design (Reference Design) in return for your promise to use reasonable efforts to develop a system based on the technology in the Reference Design. The Reference Design contains documentation and software listed below:

Documentation
PowerPC 604 SMP Reference Design Technical Specification
PowerPC 604 RISC Microprocessor Hardware Specification
IBM PowerPC 604 SMP Reference Board Design Files (on 8mm tape)
IBM PowerPC 604 SMP Reference Board Mfg. Data Files (in Gerber format)
IBM14N1372 Data Sheet
IBM11D4360B Data Sheet
Selected data sheets from other manufacturers (included with their permission).

LICENSE TO SOFTWARE

The software is licensed not sold. IBM, or the applicable IBM country organization, grants you a license for the software only in the country where you received the software. Title to the physical software and documentation (not the information contained in such documentation) transfers to you upon your acceptance of these terms and conditions. The term ”software” means the original and all whole or partial copies of it, including modified copies or portions merged into other programs. IBM retains title to the software. IBM owns, or has licensed from the owner, copyrights to the software provided under this agreement. The terms of this Agreement apply to all of the hardware, software and documentation provided to you as part of the Reference Design.

With regard to the software provided hereunder, it is understood and agreed that you intend to use the software solely for the purpose of designing PowerPC™ compatible products, testing your designs, and making your own independent determination of whether you wish to eventually manufacture PowerPC compatible products commercially. In accordance with this understanding, IBM hereby grants you the rights to: a) use, run, and copy the software, but only make such number of copies and run on such number of machines as are reasonably necessary for the purpose of designing PowerPC compatible products and testing such designs; and b) copy the software for the purpose of making one archival or backup copy.

With regard to any copy made in accordance with the foregoing license, you must reproduce any copyright notice appearing thereon. With regard to the software provided hereunder, you may not: a) use, copy, modify or merge the software, except as provided in this license; b) reverse assemble or reverse compile it; or c) sell, sublicense, rent. lease, assign or otherwise transfer it. In the event that you no longer wish to use the software, you will return it to IBM.

LICENSE TO DESIGN DOCUMENTATION

With regard to the design documentation provided hereunder, it is understood that you intend to use such documentation solely for the purpose of designing your own PowerPC compatible products, testing your designs, and making your own independent determination of whether you wish to eventually manufacture PowerPC compatible products commercially. In accordance with this understanding, IBM hereby grants you the right to: a) use the design documentation for the purpose of designing PowerPC compatible products and testing such designs; b) make derivative works of the design documentation for the purpose of designing PowerPC compatible products, and testing such designs; and c) make copies of the design documentation and any such derivative works, but only such numbers as are reasonably necessary for designing PowerPC compatible products and testing such designs.

With regard to any copy made in accordance with the forgoing license, you must reproduce any copyright notice appearing thereon. With regard to the design documentation provided hereunder, you may not: a) copy, modify, or merge the design documentation as provided in this license; or b) sell, sublicense, rent, lease, assign, or otherwise transfer it.

In the event you no longer wish to use the design documentation or any derivative versions thereof, you must return them to IBM.

DISCLAIMER OF WARRANTY

IBM does not represent or warrant that the Reference Design (which may contain prototype items): a) meets any particular requirements; b) operates uninterrupted; c) is error free; or d) is non-infringing of any patent, copyright, or other intellectual property right of any third party. IBM makes no representation or warranty regarding the performance or compatibility that may be obtained from the use of the Reference Design or that the Reference Design is adequate for any use. The Reference Design may contain errors and may not provide the level of completeness, functionality, support, performance, reliability, or ease of use available with other products, whether or not similar to the Reference Design. IBM does not represent or warrant that errors or other defects will be identified or corrected.

THE REFERENCE DESIGN IS PROVIDED ”AS IS” WITH ALL FAULTS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE REFERENCE DESIGN IS WITH YOU.

Some jurisdictions do not allow exclusion of implied warranties, so the above exclusions may not apply to you.

LIMITATION OF REMEDIES

IBM's entire cumulative liability and your exclusive remedy for damages for all causes, claims or actions wherever and whenever asserted relating in any way to the subject matter of this agreement including the contents of the Reference Design and any components thereof, is limited to twenty five thousand dollars ($25,000.00) or its equivalent in your local currency and is without regard to the number of items in the Reference Design that caused the damage. This limitation will apply, except as otherwise stated in this Section, regardless of the form of the action, including negligence. This limitation will not apply to claims by you for bodily injury or damages to real property or tangible personal property. In no event will IBM be liable for any lost profits, lost savings, or any incidental damages or economic consequential damages, even if IBM has been advised of the possibility of such damages, or for any damages caused by your failure to perform your responsibilities. In addition, IBM will not be liable for any damages claimed by you based on any third party claim. Some jurisdictions do not allow these limitations or exclusions, so they may not apply to you.

RISK OF LOSS

You are responsible for all risk of loss or damage to the Reference Design upon its delivery to you.

IBM TRADEMARKS AND TRADE NAMES

This Agreement does not give you any rights to use any of IBM's trade names or trademarks. You agree that should IBM determine that any of your advertising, promotional, or other materials are inaccurate or misleading with respect to IBM trademarks or trade names, that you will, upon written notice from IBM, change or correct such materials at your expense.

NO IMPLIED LICENSE TO IBM INTELLECTUAL PROPERTY

Notwithstanding the fact that IBM is hereby providing design information for your convenience, you expressly understand and agree that, except for the rights granted under the sections above, no right or license of any type is granted, expressly or impliedly, under any patents, copyrights, trade secrets, trademarks, or other intellectual property rights of IBM. Moreover, you understand and agree that in the event you wish to be granted any license beyond the scope of the expressly stated herein, you will contact IBM's Intellectual Property Licensing and Services Office (currently located at 500 Columbus Avenue, Thornwood, N.Y.), or such other IBM offices responsible for the licensing of IBM intellectual property, when you seek the license.

YOUR ASSUMPTION OF RISK

You shall be solely responsible for your success in designing, developing, manufacturing, distributing, and marketing any product(s), or portion(s), where use of all or any part of the Reference Design is involved. You are solely responsible for any claims, warranties, representations, indemnities and liabilities you undertake with your customers, distributors, resellers or others, concerning any product(s) or portion(s) of product(s) where use of all or any part of the Reference Design is involved. You assume the risk that IBM may introduce other Reference Designs that are somehow better than the Reference Design which is the subject of this Agreement. Furthermore, you accept sole responsibility for your decision to select and use the Reference Design; for attainment or non-attainment of any schedule, performance, cost, reliability, maintainability, quality, manufacturability or the like, requirements, or goals, self-imposed by you or accepted by you from others, concerning any product(s) or portion(s) of product(s), or for any delays, costs, penalties, charges, damages, expenses, claims or the like, resulting from such non-attainment, where use of all or any part of the Reference Design is involved.

GENERAL

In the event there is a conflict between the terms of this Agreement and the terms printed or stamped on any item or any ambiguities with respect thereto, including documentation, contained in the Reference Design, the terms of this Agreement control to the extent IBM is afforded greater protection thereby. IBM may terminate this Agreement if you fail to comply with the terms and conditions of this Agreement. Upon termination of this Agreement, you must destroy all copies of the software and documentation. You are responsible for payment of any taxes, including personal property taxes, resulting from this Agreement. Neither party may bring an action hereunder, regardless of form, more than one (1) year after the cause of the action arose. If you acquired the Reference Design in the United States, this Agreement is governed by the laws of the State of New York. In the event of litigations, trial shall be in New York without a jury. If you acquired the Reference Design in Canada, this Agreement is governed by the laws of the Province of Ontario; otherwise, this Agreement is governed by the laws of the country in which you acquired the Reference Design. All obligations and duties which, by their nature, survive termination or expiration of this Agreement, shall remain in effect beyond termination or expiration of this Agreement, and shall bind IBM, you and your successors and assigns. If any section or paragraph of this Agreement is found by competent authority to be invalid, illegal or unenforceable in any respect for any reason, the validity, legality, and enforceability of any such section or paragraph in every other respect, and the remainder of this Agreement, shall continue in effect so long as it still expresses the intent of the parties. If the intent of the parties cannot be preserved, the parties will attempt to renegotiate this Agreement and failing renegotiation, this Agreement will then be terminated. The headings in this Agreement shall not affect the meaning or interpretation of this Agreement in any way. No failure by IBM in exercising any right, power or remedy under this Agreement shall serve as a waiver of any such right, power or remedy. Neither this Agreement nor any activities hereunder will impair any right of IBM to develop, manufacture, use or market, directly or indirectly, alone or with others, any products or services competitive with those offered or to be offered by you; nor will this Agreement or any activities hereunder require IBM to disclose any business planning information to you. You agree to comply with all applicable government laws and regulations. Any changes to this Agreement must be in writing and signed by the parties.

Table of Contents

Figures

Tables

About This Book

Audience:

This reference design is designed for engineers and system designers who are interested in implementing PowerPC systems that are compliant with the *PowerPC Reference Platform Specification*. The material requires a detailed understanding of computer systems at the hardware and software level.

Reference Material:

Understanding of the relevant areas of the following documents is required for a good understanding of the reference design:

- *PowerPC 604 User's Manual*, IBM document MPR604UMU-01
- *PowerPC 604 Hardware Specification*, IBM document MPR604HSU-01
- *IBM27-82660 PowerPC to PCI Bridge User's Manual*, IBM document number MPR660UMU-01
- *PCI Local Bus Specification*, Revision 2.1, available from the PCI SIG
- *PowerPC Reference Platform Specification,* Version 1.1, IBM document MPRPRPPKG
- *The Power PC Architecture,* second edition, Morgan Kaufmann Publishers (800) 745-7323, IBM document MPRPPCARC-02
- *Intel 82378ZB System I/O (SIO) Data Book*, Intel order number 290473-004.

The following documents are useful as sources of tutorial and supplementary information about the reference design.

- *PowerPC System Architecture*, Tom Shanley, Mindshare Press (800) 420-2677.
- *IBM27-82650 PowerPC to PCI Bridge User's Manual, IBM document number MPR650UMU-01*

Document Conventions:

Kilobytes, megabytes, and gigabytes are indicated by a single capital letter after the numeric value. For example, 4K means 4 kilobytes, 8M means 8 megabytes, and 4G means 4 gigabytes.

The terms DIMM and SIMM are often used to mean DRAM module.

Hexadecimal values are identified (where not clear from context) with a lower-case letter h at the end of the value. Binary values are identified (where not clear from context) with a lower-case letter b at the end of the value.

In identifying ranges of values *from* and *to* are used whenever possible. The range statement from 0 to 2M means from and including zero up to (but not including) two megabytes. The hexadecimal value for the range from 0 to 64K is: 0000h to FFFFh.

The terms *asserted* and *negated* are used extensively. The term *asserted* indicates that a signal is active (logically true), regardless of whether that level is represented by a high or low voltage. The term *negated* means that a signal is not asserted. The # symbol at the end of a signal name indicates that the active state of the signal occurs with a low voltage level.

Signal ranges are given from MSb (most significant bit) to LSb in the form SIGNAL[MSb:LSb]. For example, CPU_ADDR[0:23] refers to CPU_ADDR signals 0 thru 23, where 0 is the most significant. Note that this convention allows easy identification of the endian mode of the nomenclature. DATA[0:12] is labeled in big endian fashion, and DATA[12:0] is referred to in little endian order.

Section 1
Introduction

This document provides a detailed technical description of the PowerPC™ 604 SMP Reference Design, and is intended to be used by hardware, software, test, simulation, and other engineers as a first source of information. Software developers should read through the entire document because pertinent facts may be located in hardware sections.

The focus of this document is mainly on the motherboard electronics and firmware. Where appropriate, this document references detailed information found in other documents. Consult other documents for information on specific I/O devices (such as hard files, CD-ROMs, L2 cache cards, video cards, etc.) that comprise a total system.

Recommendations for memory mappings, software implementations, and the like are only recommendations and may or may not represent the algorithms implemented in boot code or operating systems.

1.1 IBM Reference Products

IBM offers several different PowerPC reference products for a given PowerPC system.

1.1.1 Reference Design

The PowerPC 604 SMP Reference Design (reference design) is composed of both the intangible design and the documentation describing that design. The reference design documentation addresses the motherboard electronics, firmware, and various system related issues. The reference design is composed of this Technical Specification and, optionally, Gerber format physical design files on an 8mm tape, electrical device model files in Cadence™ format, system firmware guidelines, and contact information for commented boot ROM source code. This is the only component of the reference product array that IBM will offer for the PowerPC 604 SMP Reference Design.

1.1.2 Reference Boards and Systems

The PowerPC 604 SMP Reference Board (reference board) is the physical implementation of the motherboard part of the reference design. The PowerPC 604 SMP Reference System (reference system) consists of a complete 604 PowerPC computer system. IBM will not offer PowerPC 604 SMP reference boards or reference systems for sale. Instead, IBM has enabled other vendors to design and manufacture boards and systems based on the reference design. Contact your IMD Field Marketing Representative for further information.

1.1.3 Reference Firmware

The PowerPC 604 SMP Reference Firmware (reference firmware) is described in the reference design. It consists of the commented source code of the software contained in the boot ROM. This is available from IBM as discussed in section 12.1.

1.2 Purpose

The reference design is intended to help companies develop their own products using the PowerPC architecture. The reference design may be used:

- As a baseline system in order to gauge the effects of changes on the design
- To test new boot code
- To test operating systems and/or applications
- For performance measuring.

The reference board is:

- Produced for internal use only, primarily to verify the design.
- A compliant implementation of the PowerPC Hardware Reference Platform Specification, version 1.1
- Tested for functionality to the level of software available at the time of shipping
- A prototype of a system under development which may have prototype ASICs, errata, and/or wiring changes.

The reference design is not:

- Offered for sale by IBM.
- A complete market ready design
- Tested for compliance to FCC and other regulatory requirements.

1.3 Differences Between Release 1.0 and Release 3.0

This document describes release 3.0 of the reference design, which is associated with pass 3 of the reference board. The previous (preliminary) version was known as release 1.0 of the reference design, which corresponds to pass 2.0 of the reference board. There is no release 2.0 of the reference design.

This release (3.0) of the reference design incorporates the workarounds that were required for the errata found on the previous release (1.0) of the reference design. The schematics and BOM are updated to reflect this incorporation. This release also allows the use of revision 1.2 of the IBM27-82664 (the controller of the 660 Bridge chipset). The schematic and BOM reflect usage of revision 1.2 of the 664. For more information, see Section 20, Errata.

1.4 Reference Design Overview

This section contains an overview of the reference design. The block diagram of the reference design is shown in Figure 1 and Figure 2.

The reference design is compliant with the PowerPC Reference Platform Specification Versions 1.0 and 1.1. It is also complaint with the PCI Local Bus Specification, revisions 2.0 and 2.1.

The reference design is a PowerPC based Symmetrical Multi-Processor (SMP) system using the PowerPC 604 RISC microprocessor and the IBM27-82660 Bridge chipset (660 Bridge). The core of the system is the Central Electronics Complex (CEC) consisting of 1 or 2 CPU daughter cards and a Tag/SRAM daughter card, which is used with the imbedded L2 controller in the 660 Bridge. With these daughter cards, a variety of CEC configurations can be achieved.

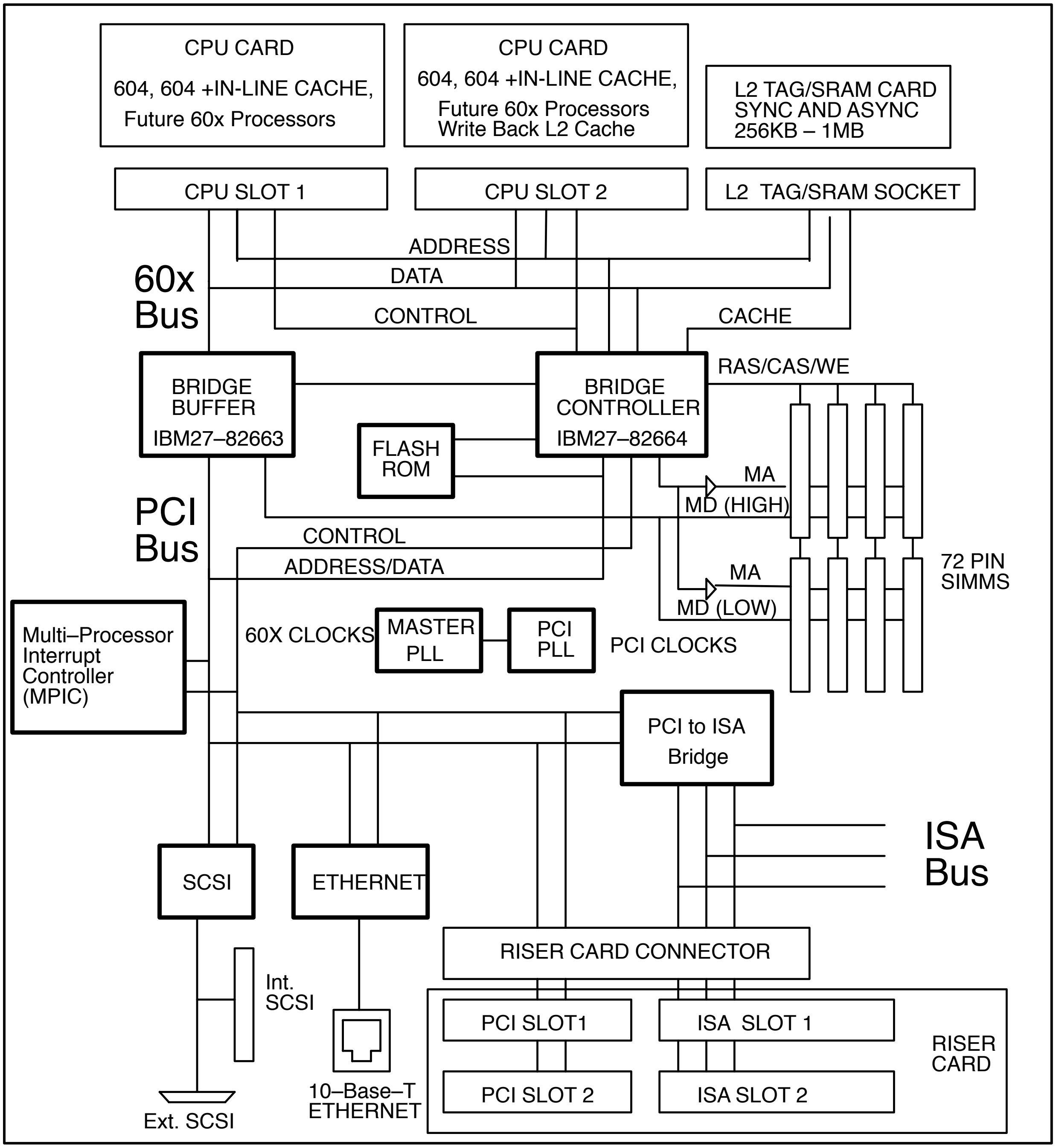


Figure 1. Reference Design Block Diagram (60x and PCI Bus)

The CEC is connected to main memory through the 660 Bridge. All I/O expansion is connected to the CEC through the PCI bus from the 660 Bridge. PCI I/O expansion includes imbedded SCSI HDD and Ethernet controllers, as well as two PCI expansion slots on an LPX riser card. Further I/O expansion is provided via the ISA and XBUS, which are derived from the PCI bus via a PCI/ISA bridge. Imbedded ISA devices include Business Audio and standard PC for Serial, Parallel, and FDD controllers. Two ISA expansion slots are available on the LPX riser slot. XBUS I/O includes these other standard PC functions: Keyboard, Mouse, NVRAM, and RTC (see Figure 2).

System firmware is provided in a FLASH or EPROM device accessed by the 664 via the PCI A/D lines. The 660 bridge direct ROM access does not conform to or violate the PCI protocol because the PCI control signals remain inactive.

The 604 SMP is designed to an industry standard LPX (9” by 13”) outline. Both 3.3v and +5v are required to power most system components. Components that require +3.3v are supported by an additional output connector from a +3.3v power supply. +3.6v is provided to the CPU slots by a 5v to 3.6v regulator on the motherboard. +12v and –12v are required to support some of the peripheral features. –5v is supplied only to the ISA connectors (via the riser card).

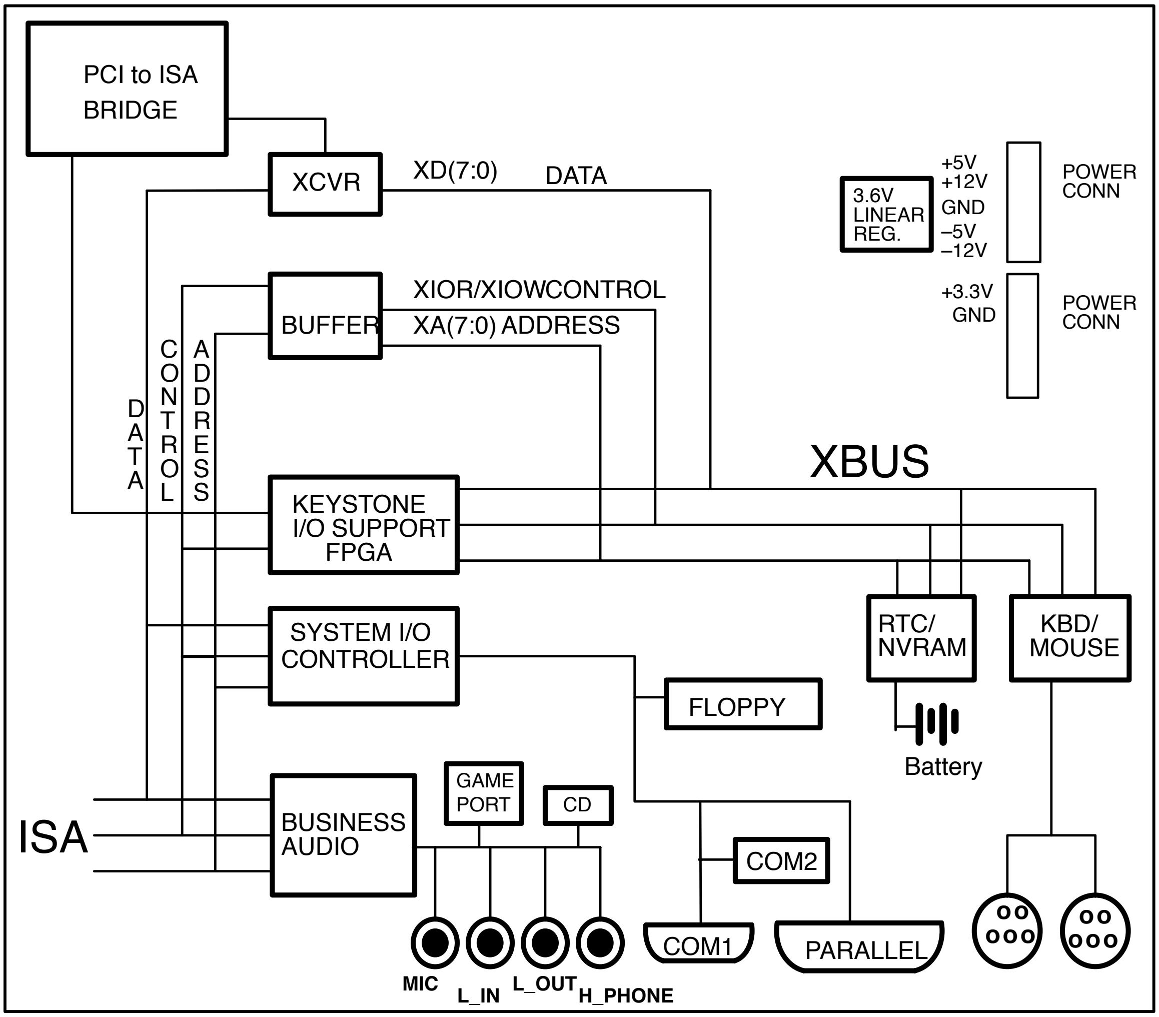


Figure 2. Reference Design Block Diagram (ISA and XBUS)

1.4.1 Processor Cards

The reference motherboard has two 256 pin DIMM connectors for the attachment of daughter cards which provide the 604 microprocessor(s) for the system. These cards interface with the motherboard over a 60x compliant bus having sideband signals for power, resets, interrupts, L2 control, and card ID information.

1.4.2 Other CEC Functions

The reference design provides CPU bus clocks to the CEC at one of two bus frequencies (nominally 60 MHz or 66 MHz as optimized for the installed processor cards). In general, the processors on the daughter cards are capable of running their internal clocks at several different multiples of the bus clock frequency. The reference design runs the PCI bus clock at a fixed frequency multiple of one half the CPU bus clock frequency.

Consult your IBM representative for currently available choices of CPU types and operating frequencies.

The reference design is equipped with an interface to support RISCWatch debugging and monitor systems for PowerPC processors.

1.4.3 IBM27-82660 Bridge

The IBM27-82660 Bridge chipset supplies many of the functions of the reference design. The 660 Bridge interfaces the CPU to system memory, the PCI bus, the L2 TAG and SRAMS, the FLASH/ROM, and other reference design components.

1.4.4 L2 Cache

The reference design supplies an L2 cache controller, located inside the 660 Bridge chipset. The motherboard provides a slot for an TAG/SRAM module. The 660 L2 is a unified, write-through, direct-mapped, look-aside cache that supports 256K-1M of async or sync SRAM to cache the low 1G of CPU memory space. The 660 L2 supplies data to the CPU bus on write hits, and it snarfs the data (updates the SRAM data while the memory controller is accessing DRAM memory) on read/write misses. It snoops PCI to memory transactions. Typical read performance is 3-1-1-1, followed by -2-1-1-1 on pipelined reads.

1.4.5 System Memory

The reference design memory subsystem can support up to 256M of 70ns or faster DRAM memory on eight 72 pin, DRAM (SIMM) modules via sockets. Each SIMM socket can support a 4M, 8M, 16M or 32M, 72 pin SIMM with parity. The DRAM subsystem is 72 bits wide: 64 data bits and eight parity bits. One parity bit is generated for each byte of data written. The 660 Bridge can also be configured to perform ECC memory data checking and correction using standard parity DRAM modules. The 660 Bridge also provides DRAM refresh, and it supports EDO hyper-page mode DRAM.

Memory access performance from the CPU bus at 66MHz with 70ns DRAM is typically:

- Pipelined burst read: 5-4-4-4 CPU bus clocks—16 CPU clocks for 32 bytes of data
- Pipelined burst write: 3-3-4-4 CPU bus clocks—17 CPU clocks for 32 bytes of data.

Memory access performance from the PCI bus at 33MHz with 70ns DRAM is typically:

- Read bursts 5-1-1-1 -1-1-1-1 6-1-1-1 -1-1-1-1 6-1-1-1 -1-1-1-1 ... 6-1-1-1 -1-1-1-1
- Write bursts 5-1-1-1 -1-1-1-1 3-1-1-1 -1-1-1-1 3-1-1-1 -1-1-1-1 ... 3-1-1-1 -1-1-1-1.

1.4.6 PCI Bus

The 660 Bridge includes the interface between the PCI bus and the rest of the system. The reference design allows CPU to PCI access and PCI bus master to memory access (with snooping), and it handles all PCI related system memory cache coherency issues. Two PCI expansion slots are provided.

The reference design also supports memory block locking, types 0 and 1 configuration cycles, and ISA master access to system memory through the ISA bridge.

1.4.7 SCSI Controller

The Symbios™ 53C810 controller attaches directly to the PCI bus on the motherboard and supports the following features:

- Eight-bit SCSI-2 interface
- Variable block size and scatter/gather data transfers
- 32-bit word data bursts with variable burst lengths
- Full 32-bit PCI bus master
- 64-byte FIFO buffer.

1.4.8 Network Support AMD AM79C970A (Ethernet)

This component attaches to the PCI bus on the motherboard and supports the following features:

- ISO8802-3 (IEEE/ANSI 802.3) and Ethernet Standards
- Big endian byte alignment
- Integrated Manchester Encoder-Decoder, integrated AUI and 10-base-T transceiver
- Compliant to PCI 2.0
- 136 byte/128 byte transmit/receive FIFO.

This chip is connected to an RJ-45 connector on the motherboard for 10-base-T Ethernet support.

1.4.9 Multi-Processor Interrupt Controller (MPIC)

This component attaches to the PCI bus on the motherboard and supports the following features:

- 15 individual interrupt inputs with programmable edge or level polarity
- ISA compatible interrupt handling via external 8259 cascaded on INT0
- Distribution of interrupts for symmetric multi-processing (SMP) systems
- One to four processors with interrupts and soft reset functions
- Implementation of OpenPIC IC
- Selectable interrupt vector, priority, level, and routing software
- integrated 32 bit counters.

1.4.10 Flash ROM

The reference design uses an AMD AM29F040-120 Flash™ ROM to contain the POST and boot code. It is recommended that vital product data such as the motherboard speed and native I/O complement be programmed into this device. It is possible to program the Flash ROM before or during or after the motherboard manufacturing process.

After power on, the initial code that is fetched is supplied from this device. The 660 Bridge manages ROM access and control. The reference design supports a 512K Flash ROM.

1.4.11 PCI/ISA Bridge Chip

This device(the Intel 82378ZB SIO) provides the PCI bus arbiter, a PCI to ISA bus bridge, and system services such as DMA and interrupt control. Its major functions are listed below:

- PCI bus arbiter
- Bridge between PCI and ISA
  - Supports 8/16 bit ISA devices
  - Addresses 24 bit on ISA
  - Partially decodes native I/O addresses
  - Forwards unclaimed PCI memory address below 16MB to the ISA bus
  - Forwards unclaimed PCI I/O address below 64KB to the ISA bus
  - Powers up to an ”open” condition (cycles may be passed to the ISA bus)
  - Generates ISA clock with a programmable divide ratio of three or four
  - Allows ISA mastering and has programmable decodes which map ISA memory cycles to the PCI bus
  - Has a 32-bit posted memory write data buffer (has no I/O buffering)
- Seven channel DMA controller
  - Has the function of two 83C37s with 32-bit extensions
  - Supports 8-bit or 16-bit devices on the ISA bus
  - Supports 32-bit addressing for ISA to PCI memory transfers
  - Has an 8-byte bidirectional buffer for DMA data.
- Timer block (function of 82C54 or block with the function)
- Interrupt Controller (function of two 8259s)
- Functions as PCI slave during programming and ISA slave cycles
- Functions as bus master during DMA or ISA master cycles.

1.4.12 Business Audio

Business audio is provided through the Crystal Semiconductor™ CS4232-KS EP stereo audio chip. Conventional (Timer 2) PC speaker functions are also provided (both the Timer 2 signal from the SIO, and the audio chip, drive the speaker).

The system provides for stereo capture and playback. It can play MIDI files, but it is not a full-functioned MIDI system. It has separate DMA channels for record and playback. The system is processor driven and does not include a DSP. Compression and decompression are supported in the hardware. The audio output is to a single speaker mounted in the cabinet. Also supported are four rear-mounted 3.5 mm jacks for:

- Stereo earphones
- Stereo microphone input
- Stereo line in
- Stereo line out.

There are also motherboard connectors for direct playback from the CD-ROM and for an internal fax-modem card.

1.4.13 Native I/O Controller National PC87332 Super I/O

This component is located on the ISA bus and contains:

- Floppy Disk Controller (software compatible with DP8473,765A and NS82077)
- Two serial ports (software compatible with INS8250N-B, PC16550A, and PC16450. Has FIFOs. Decodes COM 1-4)
- One enhanced bidirectional parallel port
- IDE interface (unused).

1.4.14 X Bus

The ISA/XBUS bridge function is supported by the ISA bridge and other devices.

1.4.15 Time of Day Clock

The reference design uses a Dallas Semiconductor™ DS1385S to provide the real time clock (TOD or RTC) function. This device is PC compatible and resides on the X-bus. It features an additional 4K of NVRAM and a replaceable battery.

1.4.16 PS/2 Compatible Keyboard/Mouse Controller

The reference design uses an Intel 8042AH as a keyboard and mouse controller.

The code used is the same version as used in IBM Personal System/2 machines. This microcode may differ from other 8042 type keyboard controllers.

1.4.17 System I/O EPLD

The system I/O EPLD is a programmable logic device that uses the X-bus signals and the partial decode signals from the SIO to decode chip selects for various components. It contains several system planar registers and glue logic for planar subsystems.

1.4.18 System Clocks

The motherboard clocks are provided by two PLL crystals and oscillators. The master PLL provides all the CPU bus clocks from one of two reference sources. At the CPU bus frequency, it also provides a seed clock to the second PLL, which generates the PCI bus clocks. One of the PCI clocks is used by the PCI to ISA bridge controller to generate the ISA bus clock and timings.

I/O clocks (keyboard, mouse, 82376/82378ZB timers, SCSI, etc.) are provided by crystal and oscillators on the motherboard and are located near the intended load.

1.5 Quickstart Peripheral List

The reference design is intended for typical PC peripherals. Products from a large number of manufacturers should work satisfactorily. Reference boards do not come with all of the required peripherals, cables, speaker, indicator LEDs, switches, and such that are needed to configure a properly working system.

Table 1 outlines the generic requirements for peripherals and gives examples of some devices that have been used for testing. It is not a recommendation of any particular vendor. The purpose of this table is to outline at least one set of peripherals that may be used to begin testing.

Table 1 does not include cables for a parallel port, indicators, a switch, or a speaker.

An IBM 3101 asynchronous terminal or equivalent is required for testing with the bring up driver (BUD) code. Settings are 8-bit, no parity, one stop bit, and 9600 baud. VT100 or VT52 emulator terminals may be acceptable. It is desirable to also have a video monitor for BUD tests. The boot code will boot with either an async console, a video on motherboard, or both.

Table 1. Quickstart Peripheral List

Generic Description	Example Device
L2 TAG/SRAM card, 512KB	IBM Corp. P.N. 26H3015
Memory SIMMs 16MB	Hitachi HB56D436B-7, IBM 57G8901
Video adapter card, PCI S3	Diamond Stealth (S3) 864
Floppy disk drive, 3.5”x1.44MB	Alps DFR723F, IBM 73G4514, Mitsubishi MF355F-258UG
Hard disk drive, SCSI-2, 8 bit	Quantum LPS270/5405, Maxtor MXT-540SL, IBM WDS-3200 (79F4042)
Hard disk drive, SCSI 1GB	IBM 94G3187
CD_ROM drive, internal SCSI	Toshiba XM-4101BMY
CD_ROM drive, internal SCSI, 4x	Toshiba 5301-4x
Chassis, LPX	Olsen Metal Products
Power supply, 200W Energy Star	API-3186S, IBM 06H2968
Switch (Power ON/OFF)	IBM 06H3860
Box fan	Panaflow FBA08T12M
Box fan shock mounts	IBM 81F7977
Internal cables, floppy, SCSI, and CD-ROM	Standard cables
Speaker, internal 8Ω .5W 2pin	
LED, 2.5 ma drive	
Asynchronous terminal	IBM 3101
Super VGA monitor	IBM 6324, 6325, 6327, 9524, 9525, 9527, 9521
Keyboard, PS/2 compatible	
Mouse, PS/2 compatible	

Section 2 CPU Bus

This section discusses topics that are directly related to the CPU bus, including how the 660 Bridge decodes CPU initiated transfers as a function of the transfer type and address range. For more information, refer to *The IBM27-82660 PowerPC to PCI Bridge User's Manual* (660 Bridge User's Manual). See Figure 3 for a layout of the CPU bus.

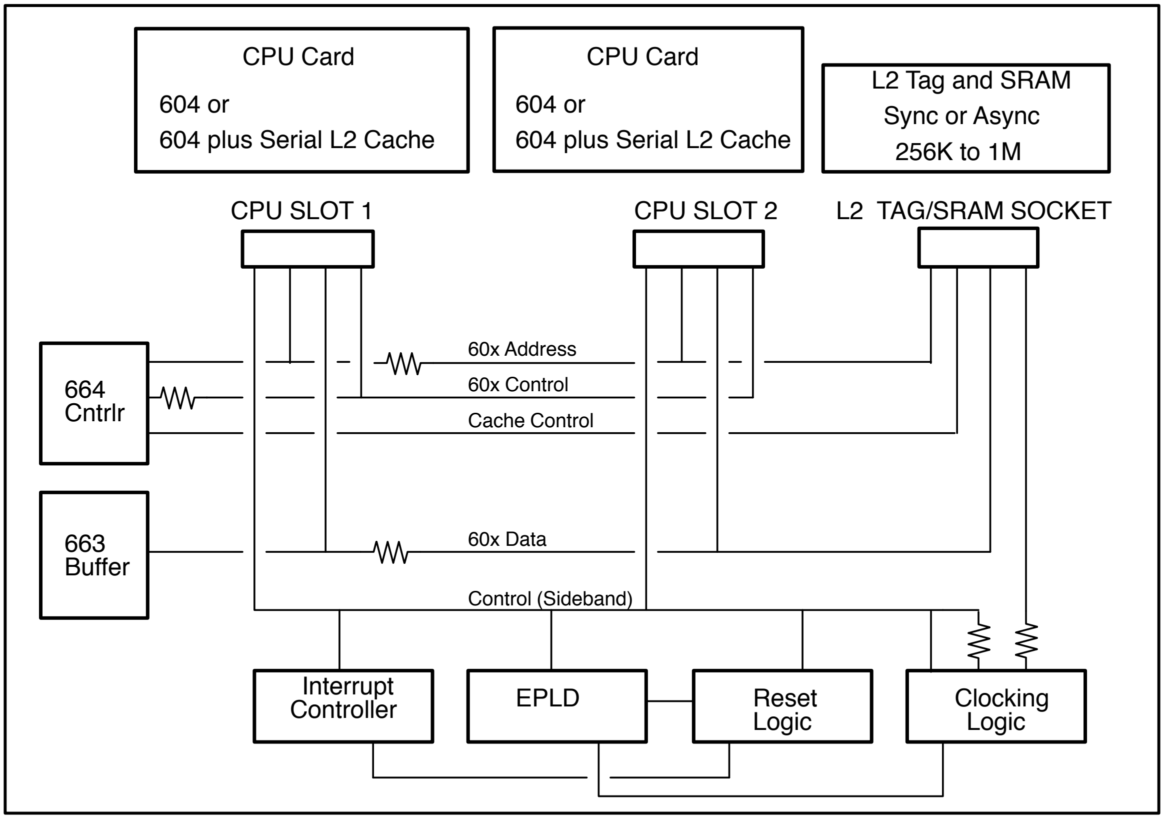


Figure 3. CPU Bus Block Diagram

The term 604, as used herein, refers to the PowerPC 604 family of CPUs.

2.1 CPU Busmasters

The reference design motherboard has no imbedded CPU. Instead, it supports one or two CPU busmaster cards that plug into the system bus via connectors. There are two CPU card connectors or slots. Each CPU busmaster must conform to the 604 bus interface specification. It must look like a 604 to the system bus.

At least one CPU must be installed. This card can feature either a 604 or a 604 plus a serial L2 controller which looks like a 604 to the system bus. Since the 660 Bridge parks the CPU bus on CPU1, uniprocessor systems will perform better while the CPU card is installed in CPU slot 1.

CPU slot 2 can contain either a second CPU card for SMP operation, or a non-604 device (such as an L2 cache or other CPU bus agent) that complies with the 604 bus interface specification. Use only CPU cards of similar configuration and capability to implement SMP.

Both CPU cards are supported by the motherboard over a CPU bus interface that contains full interrupt, clocking, and card ID functions. The reference design offers robust support for both uniprocessor and SMP systems.

One level of address bus pipelining is supported. One level of data write is posted. Precise exceptions are reported via TEA#, and imprecise exceptions are reported via MCP#. PIO or programmed I/O transactions (XATS# type) are not supported.

2.1.1 CPU Bus Arbitration

The reference design supports three busmasters on the CPU bus (two CPU busmaster cards and the 660 Bridge). CPU bus arbitration between the three busmasters is handled by the arbiter in the 660 Bridge, which coordinates the activities of the three CPU bus agents; CPU1 (in CPU card slot 1), CPU2 (in CPU card slot 2), and the 660 Bridge snoop engine. The snoop engine is a conceptual set of logic inside the Bridge which broadcasts snoop cycles to the CPU bus in response to PCI to memory transactions. To minimize CPU1 to memory latency, the 660 Bridge parks the CPU bus on CPU1 while the bus is idle.

For more information on CPU bus arbitration, see *The 660 Bridge User's Manual.*

2.1.2 Fast L2/Data Streaming Mode (No-DRTRY#)

The reference design default configuration is for DRTRY# mode. For use with CPU cards that use no-DRTRY# (fast L2/data streaming) mode, the reference motherboard can be reconfigured. Remove R232 to isolate the DBB# on each CPU card from the other CPU card. Each installed CPU card must provide a pullup on DBB.

In no-DRTRY# mode, each CPU card must also provide correct generation of DRTRY# for the CPU on the card. For the PowerPC 604 CPU card, wire DRTRY# of the CPU to #HRESET of the connector (rather than to connector DRTRY#). This will provide a low level on the CPU DRTRY# at reset and a high level under normal operation, which will place the 604 CPU in fast L2/data streaming mode.

2.1.3 CPU Bus Frequency

The reference design supports CPU bus speeds of 60MHz and 66MHz, and PCI bus speeds of 30MHz and 33MHz. The reference design is initially configured with a CPU:PCI bus clock ratio of 2:1. The CPU:PCI clock ratio can be changed to 1:1 by reconfiguring the PCI PLL and the 660 Bridge, as long as other system considerations are handled correctly. The 3:1 mode is not available due to limitations of the PCI PLL. See Section 7 for more information on clock issues.

2.1.4 Bi-Endian Mode Operation

Bi-endian mode operation is discussed in Section 3.

2.2 System Response by CPU Bus Transfer Type

All access to the rest of the system is provided to the CPU by the 660 Bridge. Table 2 shows 660 Bridge decoding of CPU bus transfer types. Based on TT[0:3], the 660 Bridge responds to CPU bus master cycles by generating a read transaction, a write transaction, or an address-only response. The 660 Bridge ignores TT[4] when it evaluates the transfer type.

The bridge decodes the target of the transaction based on the address range of the transfer as shown in Table 3. The transfer type decoding shown in Table 2 combines with the target decoding to produce one of the following:

- System memory reads and writes
- PCI I/O reads and writes
- PCI configuration reads and writes
- PCI interrupt acknowledge reads
- PCI memory reads and writes
- System ROM reads and writes
- Various bridge control register (BCR) reads and writes.

Table 2. TT[0:3] (Transfer Type) Decoding by 660 Bridge

TT[0:3]	60X Operation	60X Bus Transaction	660 Bridge Operation For CPU to Memory Transfers	660 Bridge Operation For CPU to PCI Transactions
0000	Clean block or lwarx	Address only	Asserts AACK#. No other response. No PCI transaction.	
0001	Write with flush	SBW(1) or burst	Memory write operation.	PCI write transaction.
0010	Flush block or stwcx	Address only	Asserts AACK#. No other response. No PCI transaction.	
0011	Write with kill	SBW or burst	Memory write operation. L2 invalidates addressed block.	PCI write transaction.
0100	sync or tlbsync	Address only	Asserts AACK#. No other response. No PCI transaction.	
0101	Read or read with no intent to cache	SBR(1) or burst	Memory read operation.	PCI read transaction.
0110	Kill block or icbi	Address only	Asserts AACK#. L2 invalidates addressed block.	Asserts AACK#. No other response.
0111	Read with intent to modify	Burst	Memory read operation.	PCI read transaction.
1000	eieio	Address only	Asserts AACK#. No other response. No PCI transaction.	
1001	Write with flush atomic, stwcx	SBW	Memory write operation.	PCI write transaction.
1010	ecowx	SBW	Asserts AACK# and TA# if the transaction is not claimed by another 60X bus device. No PCI transaction. No other response.	
1011	Reserved		Asserts AACK#. No other response. No PCI transaction.	
1100	TLB invalidate	Address only	Asserts AACK#. No other response. No PCI transaction.	
1101	Read atomic, lwarx	SBR or burst	Memory read operation.	PCI read transaction.
1110	External control in, eciwx	Address only	660 asserts all ones on the CPU data bus. Asserts AACK#, and TA# if the transaction is not claimed by another 60X bus device. No PCI transaction. No other response.	
1111	Read with intent to modify atomic, stwcx	Burst	Memory read operation.	PCI read transaction.

Note:
1) As used in this table, SBR means Single-Beat Read, and SBW means Single-Beat Write.

Transfer types in Table 2 that have the same response are handled identically by the bridge. For example, if the address is the same, the bridge generates the same memory read transaction for transfer types 0101, 0111, 1101, and 1111.

The 660 Bridge does not generate PCI or system memory transactions in response to address-only transfers. The bridge does drive all-ones onto the CPU bus and signals TA# during an eciwx if no other CPU bus agent claims the transfer.

References in the remainder of this document to a CPU read, assume one of the transfer types in Table 2 that produce the read response from the 660 Bridge. Likewise, references to a CPU write refer to those transfer types that produce the write response.

2.3 System Response by CPU Bus Address Range

The 660 Bridge determines the target of a CPU bus master transaction based on the CPU bus address range as shown in Table 3. The acronym BCR means a (660) bridge control register.

Table 3. 660 Bridge Address Mapping of CPU Bus Transactions

CPU Bus Address	Other Conditions	Target Transaction	Target Bus Address	Notes
0 to 2G 0000 0000h to 7FFF FFFFh		System Memory	0 to 2G 0000 0000h to 7FFF FFFFh	1., 2.
2G to 2G + 8M 8000 0000h to 807F FFFFh	Contiguous Mode	PCI I/O Transaction, BCR Transaction, or PCI Configuration Transaction	0 to 8M 0000 0000h to 007F FFFFh	3.
	Non-Contiguous Mode		0 to 64K 0000 0000h to 0000 FFFFh	4.
2G + 8M to 2G + 16M 8080 0000h to 80FF FFFFh		PCI Configuration (Type 0) Transaction	PCI Configuration Space 0080 0000h to 00FF FFFFh	
2G + 16M to 3G – 8M 8100 0000h to BF7F FFFFh		PCI I/O Transaction	16M to 1G – 8M 0100 0000h to 3F7F FFFFh	
3G – 8M to 3G BF80 0000h to BFFF FFFFh		BCR Transactions and PCI Interrupt Ack. Transactions	1G – 8M to 1G 3F80 0000h – 3FFF FFFFh	3., 6.
3G to 4G – 2M C000 0000h to FFDF FFFFh		PCI Memory Transaction	0 to 1G – 2M 0000 0000h to 3FDF FFFFh	
4G – 2M to 4G FFE0 0000h to FFFF FFFFh	Direct Attach ROM Read, Write, or Write Lockout	BCR Transaction	0 to 2M 0000 0000h to 001F FFFFh (ROM Address Space)	5.
	Remote ROM	PCI Memory Transaction to I/O Bus Bridge	1G – 2M to 1G 3FE0 0000h to 3FFF FFFFh	5.

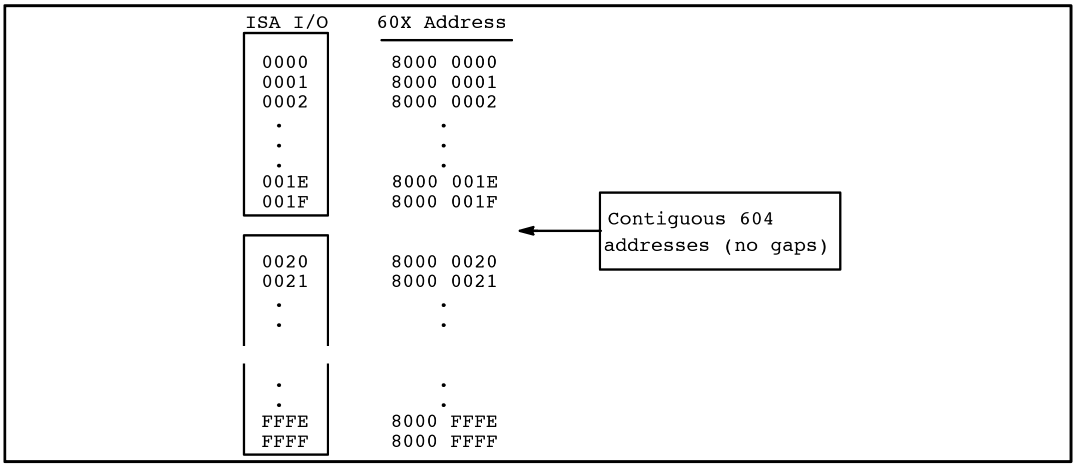
Notes:

1. System memory can be cached. Addresses from 2G to 4G are not cacheable.
2. Memory does not occupy the entire address space.
3. Registers do not occupy the entire address space.
4. In non-contiguous mode, each 4K page in the 8M CPU bus address range maps to 32 bytes in PCI I/O space.
5. Registers and memory do not occupy the entire address space. Accesses to unoccupied addresses result in all one-bits on reads and no-ops on writes.
6. A memory read of BFFF FFF0h generates an interrupt acknowledge transaction on the PCI bus.

2.3.1 Address Mapping for Contiguous I/O

In contiguous I/O mode, CPU addresses from 2G to 2G + 8M generate a PCI I/O cycle on the PCI bus with PCI_AD[29:00] unchanged. The low 64K of PCI I/O addresses are forwarded to the ISA bus unless claimed by a PCI agent.

Memory page protection attributes can only be assigned by 4K groups of ports, rather than by 32-port groups as in the non-contiguous mode. This is the power-on default mode. Figure 4 gives an example of contiguous I/O partitioning.

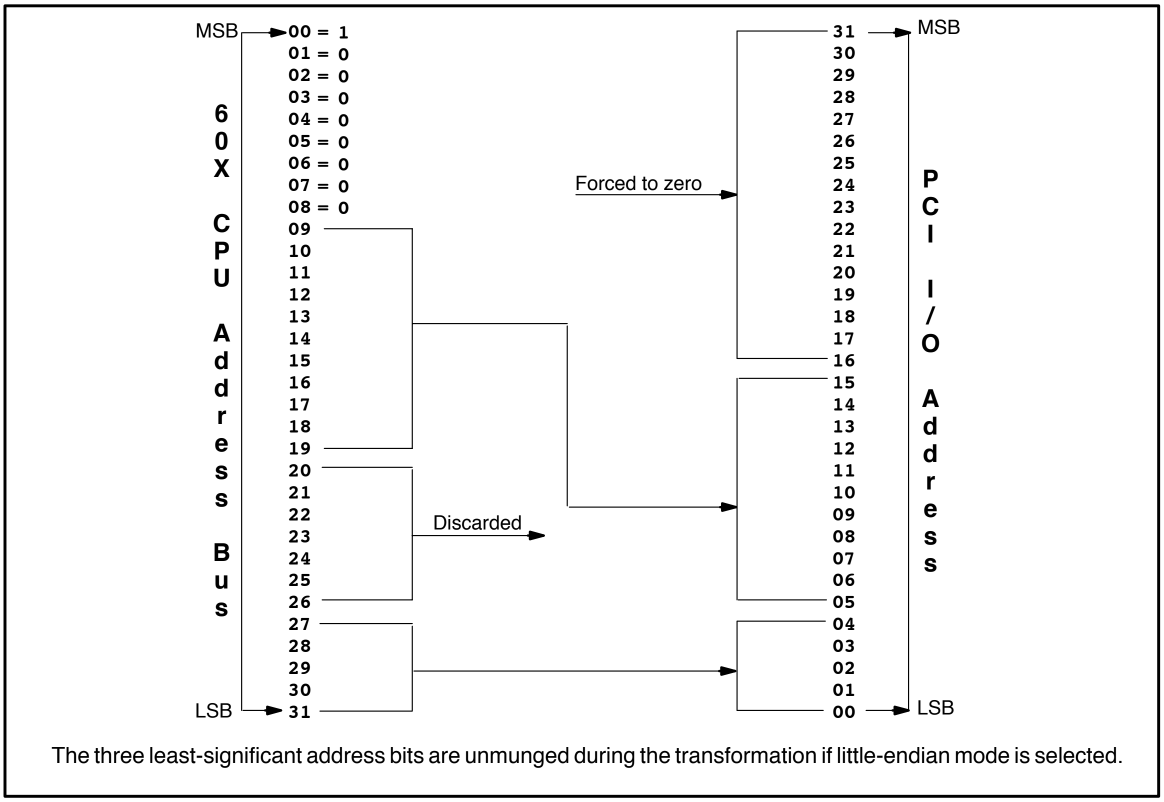


ISA I/O	60X Address
0000	8000 0000
0001	8000 0001
0002	8000 0002
.	.
.	.
.	.
001E	8000 001E
001F	8000 001F
0020	8000 0020
0021	8000 0021
.	.
.	.
.	.
.	.
FFFE	8000 FFFE
FFFF	8000 FFFF

Figure 4. Contiguous PCI I/O Address Translation

2.3.2 Address Mapping for Non-Contiguous I/O

Figure 5 shows the address mapping that the 660 Bridge performs in non-contiguous mode.



The three least-significant address bits are unmunged during the transformation if little-endian mode is selected.

Figure 5. Non-Contiguous PCI I/O Address Transformation

The I/O map type register (address 8000 0850h) and the bridge chip set options 1 register (index BAh) control the selection of contiguous and non-contiguous I/O. In non-contiguous mode, the 8M address space of the 60X bus is compressed into 64K of PCI address space, and the 60X CPU cannot create PCI I/O addresses from 64K to 8M.

In non-contiguous I/O mode, the 660 Bridge partitions the address space such that each 4K page is remapped into a 32-byte section of the 0 to 64K ISA port address space. Thus 60X CPU protection attributes can be assigned to any of the 4K pages. This provides a flexible mechanism to lock the I/O address space from change by user-state code. This partitioning spreads the ISA I/O address locations over 8M of CPU address space.

In non-contiguous mode, the first 32 bytes of each 4K page are mapped to a 32-byte space in the PCI address space. The remainder of the addresses in the 4K page are aliases of the same 32-byte PCI space, and are assigned the same protection attributes in the CPU.

For example, in Figure 6, 60X CPU addresses 8000 0000h to 8000 001Fh are converted to PCI I/O port 0000h through 001Fh. PCI I/O port 0020h starts in the next 4K page at 60X CPU address 8000 1000h.

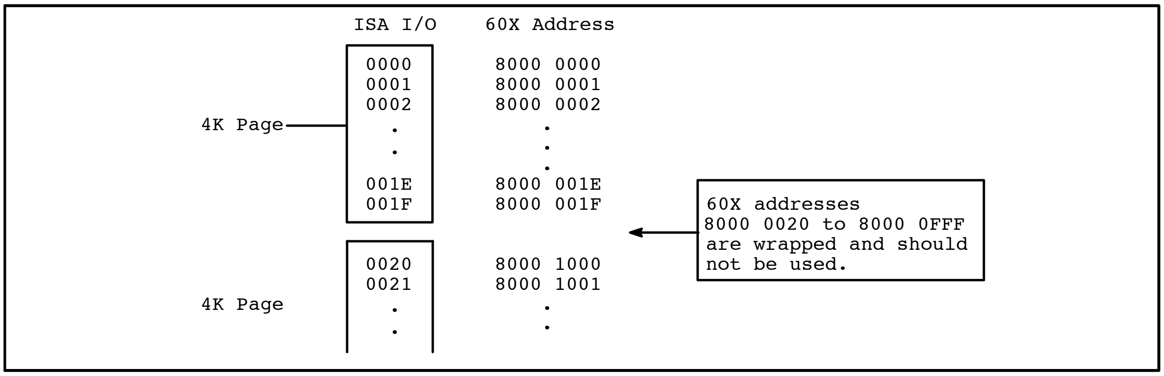


Figure 6. Non-Contiguous PCI I/O Address Translation

2.3.3 PCI Final Address Formation

The 660 Bridge maps 60X CPU bus addresses from 2G to 4G as PCI transactions, error address register reads, or ROM reads and writes. The 660 Bridge manipulates 60X bus addresses from 2G to 4G to generate PCI addresses as follows:

- PCI_AD[31:30] are set to zero.
- PCI_AD[2:0] are unmunged (see Section 3) if little-endian mode is selected.
- After unmunging, PCI_AD[1:0] are set to 00b except for PCI I/O and configuration cycles.

2.4 CPU to Memory Transfers

The system memory address space is from 0 to 2G. Physical memory does not occupy the entire address space. When the CPU reads an unpopulated location, the 660 Bridge returns all-ones and completes the transfer normally. When the CPU writes to an unpopulated location, the Bridge signals normal transfer completion to the CPU but does not write the data to memory. The memory select error bit in the error status 1 register (bit 5 in index C1h) is set in both cases.

All CPU to memory writes are posted and can be pipelined.

The 660 Bridge supports all CPU to memory bursts, and all single-beat transfer sizes and alignments that do not cross an 8-byte boundary, which includes all memory transfers initiated by the 604 CPU.

2.4.1 LE Mode

The bridge supports all transfer sizes and alignments that the CPU can create in LE mode; however, all loads or stores must be at natural alignments in LE mode (or the PowerPC 604 will take an alignment exception). Also, load/store multiple word and load/store string word instructions are not supported in the CPU in LE mode.

2.5 CPU to PCI Transactions

Since all CPU to PCI transactions are CPU memory mapped, software must, in general, utilize the EIEIO instruction which enforces in-order execution, particularly on PCI I/O and configuration transactions. Some PCI memory operations can be sensitive to order of access also. See the *660 Bridge User's Manual*.

All addresses from 2G to 4G (including ROM space) must be marked non-cacheable. See the *PowerPC Reference Platform Specification*. The reference design supports all PCI bus protocols during CPU to PCI transactions.

The reference design supports all CPU to PCI transfer sizes that do not cross a 4-byte boundary. The reference design also supports 8-byte CPU to PCI writes that are aligned on an 8-byte boundary. The bridge does not support CPU bursts to the PCI bus.

When the 660 Bridge decodes a CPU access as targeted for the PCI, the 660 Bridge requests the PCI bus. Once the SIO grants the PCI bus to the 660 Bridge, the bridge initiates the PCI cycle and releases the CPU bus.

CPU to PCI transactions that the PCI target retries, cause the 660 Bridge to deassert its PCI_REQ# (the Bridge follows the PCI retry protocol). The Bridge stays off of the PCI bus for two PCI clocks before reasserting PCI_REQ# (or FRAME#, if the PCI bus is idle and the PCI_GNT# to the Bridge is active).

2.5.1 CPU to PCI Read

If the CPU to PCI cycle is a read, a PCI read cycle is run. If the PCI read cycle completes, the data is passed to the CPU, and the CPU cycle is ended. If the PCI cycle is retried, the CPU cycle is retried. If a PCI master access to system memory is detected before the PCI read cycle is run, then the CPU cycle is retried (and no PCI cycle is generated).

2.5.2 CPU to PCI Write

If the CPU to PCI cycle is a write, a PCI write cycle is run. CPU to PCI I/O writes are not posted, as per the *PCI Local Bus Specification* version 2.1. If the PCI transaction is retried, the Bridge retries the CPU.

CPU to PCI memory writes are posted, so the CPU write cycle is ended as soon as the data is latched. If the PCI cycle is retried, the Bridge retries the cycle until it completes.

2.5.2.1 Eight-Byte Writes to the PCI (Memory and I/O)

The 660 Bridge supports 1-byte, 2-byte, 3-byte, and 4-byte transfers to and from the PCI. The 660 Bridge also supports 8-byte memory and I/O writes (writes only, not reads) to the PCI bus. This enables the use of the 604 store multiple instruction to PCI devices. When an 8-byte write to the PCI is detected, it is not posted initially. Instead, the CPU waits until the first 4-byte write occurs, then the second 4-byte write is posted. If the PCI retries on the first four byte transfer, or a PCI master access to system memory is detected before the first 4-byte transfer, then the CPU is retried. If the PCI retries on the second 4-byte transfer, then the 660 Bridge retries the PCI write.

2.5.3 CPU to PCI Memory Transactions

CPU transfers from 3G to 4G – 2M are mapped to the PCI bus as memory transactions.

2.5.4 CPU to PCI I/O Transactions

CPU transfers from 2G+16M to 3G – 8M are mapped to the PCI bus as I/O transactions. In compliance with the PCI specification, the 660 Bridge master aborts all I/O transactions that are not claimed by a PCI agent.

2.5.5 CPU to PCI Configuration Transactions

The reference design allows the CPU to generate type 0 and type 1 PCI configuration cycles. The CPU initiates a transfer to the appropriate address, the 660 Bridge decodes the cycle and generates a request to the PCI arbiter in the SIO. When the PCI bus is acquired, the 660 Bridge enables its PCI_AD drivers and drives the address onto the PCI_AD lines for one PCI clock before it asserts PCI_FRAME#. Predriving the PCI_AD lines for one clock before asserting PCI_FRAME#, allows the IDSELs to be resistively connected to the PCI_AD[31:0] bus at the system level.

The transfer size must match the capabilities of the target PCI device for configuration cycles. The reference design supports 1-, 2-, 3-, and 4-byte transfers that do not cross a 4-byte boundary, and supports doubleword aligned 8-byte writes to the PCI.

Address unmunging and data byte swapping follows the same rules as for system memory with respect to BE and LE modes of operation. Address unmunging has no effect on the CPU address lines which correspond to the IDSEL inputs of the PCI devices.

See Section 13.2.1 for more information on PCI configuration transactions, including the IDSEL assignment.

2.5.6 CPU to PCI Interrupt Acknowledge Transaction

Reading the interrupt acknowledge address (BFFF FFF0h) causes the bridge to arbitrate for the PCI bus and then to execute a standard PCI interrupt acknowledge transaction. The system interrupt controller in the ISA bridge claims the transaction and supplies the 1-byte ISA interrupt vector. There is no physical interrupt vector BCR in the bridge. Other PCI bus masters can initiate interrupt acknowledge transactions, but this may have unpredictable effects. Also see Section 6, Exceptions, for more information.

2.5.7 PCI Locks and CPU Reservations

The 660 Bridge does not set PCI locks when acting (for the CPU) as the PCI busmaster. The CPU has no mechanism to initiate a PCI lock protocol.

The 660 Bridge allows PCI busmasters to lock one 32-byte cache sector (block) of system memory using the PCI_LOCK# signal. Once a PCI lock is established, the block address is saved. Subsequent accesses to that block from other PCI bus masters or from the CPU bus are retried until the lock is released.

The bridge generates a flush block (see the 660 Bridge User's Manual) snoop cycle on the CPU bus when a PCI bus master sets the PCI lock. The flush block snoop cycle causes the L1 and L2 caches to invalidate the locked block, which prevents cache hits on accesses to locked blocks. If the cache contains modified data, the PCI cycle is retried and the modified data is pushed out to memory.

The 604 does not have a bus locking function. Instead, the 604 uses the *load reserve* and *store conditional* instructions (lwarx and stwcx) to implement exclusive access. This reservation protocol is explained in *The 604 User's Manual*.

The 660 Bridge supports the 604 reservation protocol. The 660 Bridge takes no action on the PCI bus during a CPU request for reservation. Since he 660 Bridge broadcasts the address of all PCI to memory transactions to the CPU, the CPU can monitor PCI memory accesses. If one of the PCI to memory accesses violates the CPU reservation, then the CPU takes appropriate action.

2.6 CPU to ROM Transfers

The *PowerPC Reference Platform Specification* allocates the upper 8M of the 4G CPU address space as ROM space. The 660 Bridge implements a 2M ROM space from 4G – 2M to 4G. The actual ROM is a 512K AMD Flash™ ROM device located at 4G – 2M. The ROM is attached to the 660 Bridge via the PCI_AD lines. This mode is required when using the

Intel SIO. ROM device writes and write-protect commands are supported. See the *660 Bridge User's Manual* for more information.

The ROM device attaches to the 660 Bridge by means of control lines and the PCI_AD[31:0] lines. When a CPU bus master reads from the ROM, the bridge masters a BCR transaction, during which it reads the ROM and returns the data to the CPU. CPU writes to the ROM are also forwarded to the ROM device while the write protect bit in the 660 Bridge is not set.

Although connected to the PCI_AD lines, the ROM is not a PCI agent. The ROM and the PCI agents do not interfere with each other because the ROM is under bridge control, and the bridge does not enable the ROM except during ROM cycles. The bridge accesses the ROM by means of BCR transactions. Other PCI devices cannot read or write the ROM because they cannot generate BCR transactions.

2.6.1 CPU to ROM Read

At power-on, the 604 CPU comes up in BE Mode with the L1 cache disabled, and begins fetching instructions (using 8-byte single beat reads) at address FFF0 0100 (4G – 1M + 100h). The 660 Bridge also resets to BE mode.

The system ROM address space is from 4G – 2M to 4G. Since the size of the installed ROM is less than 2M (512K), it is aliased every 512K throughout the ROM space. Location 0 of the 512K ROM is mapped to CPU bus addresses 4G – 2M, 4G – 1.5M, 4G – 1M, and 4G – 0.5M.

The ROM is located on the PCI bus physically but not logically, and is 8 bits wide. This requires the 660 Bridge to decode ROM address, run 8 cycles to PCI bus without activating FRAME, accumulate the 8 single bytes of read data into an 8-byte group and generate a TA# and an AACK# to complete the cycle. The CPU can also read the ROM using bursts, but it receives the same 2 instructions from the ROM on each beat of the burst. For more information, see the *660 Bridge User's Manual*.

Software can lock out the ROM using a 660 Bridge BCR. When the CPU writes to any ROM location while the ROM is locked out, the bridge signals normal transfer completion to the CPU but does not write the data to the ROM. The CPU bus write to the locked flash (ROM) bit in the 660 Bridge error status 2 register (bit 0 in index C5h) is set.

2.6.2 CPU to ROM Write

Writing to the (flash) ROM is another very specialized cycle. Only one address (FFFF FFF0) is used for writing data to ROM. The ROM address and data are both encoded into four bytes and written using a 4-byte write transfer. Eight byte and burst transfers to the ROM are not supported. See the 660 Bridge User's Manual.

Writes to ROM may be performed in either BE or LE mode. The data byte swapper in the 660 Bridge is gated according to endian mode. Writes in BE mode occur in natural sequence. However, address unmunging in LE mode has no effect on the cycle because the addresses are ignored. Therefore, software must reverse the byte significance of the data and address encoded into the store instructions for LE mode writes to the ROM.

2.6.2.1 ROM Write Protection

ROM write protection must be implemented within software. Port FFFF FFF1 can be used to lock out all ROM writes. Writing any data to this port address locks out all ROM writes until the 660 Bridge is hardware reset. In addition, Flash ROM itself has means to permanently lock out changing certain sectors by writing control sequences. Consult the *Flash ROM Specification* for details.

2.6.3 CPU to BCR Transfers

The 660 Bridge can be extensively programmed by means of the Bridge Control Registers (BCR). See *The 660 Bridge User's Manual* for a description of the operation and programming of the 660 Bridge BCRs.

2.7 CPU Card Interface (CPU Slot)

The CPU card interface (slot) consists of the functional, physical, and electrical interface between the reference design motherboard and the reference design CPU card. The functional interface is described in Section 2.7.1, Signal Descriptions, and the physical and electrical interfaces are described in the remainder of section 2.7. There are two CPU card slots. At least one slot must contain a CPU card containing a 604 CPU. Since the 660 Bridge parks the CPU bus on CPU slot 1, uniprocessor systems can achieve better performance by installing the CPU in that slot.

Information and constraints given for CPU cards also apply to compliant CPU busmaster cards that are not 604 CPU cards. The terms CPU and CPU busmaster are often used interchangeably in this document.

2.7.1 CPU Slot Signal Descriptions

Table 4 describes the signals used to interface the CPU cards to the motherboard. These signals are carried by the main CPU connector associated with each CPU slot. In Table 4, signals are labeled as inputs (I) or outputs (O) as viewed from the motherboard. Thus DBG# is shown as an (O) output because it is an output of the motherboard. For more information on the CPU bus signals, see the *604 and 660 Bridge User's Manuals*.

Table 4. CPU Slot Signal Descriptions

Signal	I/O	Level	Note	Description
604 Bus Signals				
A[0:31]	I/O		1	**CPU Address Bus** Represents the 32 bit physical address of the current transaction. The address is valid from the bus cycle, in which TS# is asserted through the bus cycle in which AACK# is asserted. The address bus is driven by the busmaster. **Motherboard:** These signals are bussed (connected) to each CPU slot and the L2 slot. **CPU Card:**
AACK#	I/O		1	**Address Acknowledge** Assertion Indicates completion of the current address tenure. **Motherboard:** This signal is bussed (connected) to each CPU slot. **CPU Card:**
ABB#	I/O		1	**Address Bus Busy** Indicates that the address bus is in use and cannot be driven (or claimed) by another busmaster, regardless of state of BG# for that master. **Motherboard:** 10K pullup. This signal is bussed (connected) to each CPU slot. **CPU Card:**
AP[0:3]	I/O		1	**Address Bus Parity** Indicates that the parity of each byte of the address bus is odd (1) or even (0). Even parity generates an address parity error in a 604. Driven by the busmaster. **Motherboard:** 10K pullup. The motherboard does not support CPU address bus parity generation or checking on transactions to memory or I/O. These signals are pulled up to force odd parity and avoid CPU address parity errors if no CPU is mastering the bus. These signals are bussed (connected) to each CPU slot. **CPU Card:**
ARTRY#	I/O		1	**Address Retry** Indicates that the current CPU address tenure is to be terminated and retried later. If the data tenure is already in progress, it must be aborted by the master and retried later. This signal allows other busmasters to backoff the current busmaster in order to maintain coherency. **Motherboard:** 10K pullup. This signal is bussed (connected) to each CPU slot. **CPU Card:**
BG#	O		1	**Bus Grant** Indicates that the requesting busmaster can assume ownership of the address bus (with proper qualification by ABB# and ARTRY#). **Motherboard:** 10k pullup. There is an individual BG# for each CPU slot. **CPU Card:**

Table 4. CPU Slot Signal Descriptions (Continued)

Signal	I/O	Level	Note	Description
604 Bus Signals				
BR#	I		1	**Bus Request** Asserted by a busmaster to request mastership of the address bus. **Motherboard:** 10K pullup. There is an individual BR# for each CPU slot. **CPU Card:**
WT#	I		1	**Write Thru** Indicates that the single beat transfer currently on the bus is write through. Second level caches should not post this write operation **Motherboard:** 10K pullup. This signal is bussed (connected) to each CPU slot. **CPU Card:**
CI#	I		1	**Cache Inhibit** Indicates that the single beat transfer currently on the bus is not being cached. **Motherboard:** 10K pullup. This signal is bussed (connected) to each CPU slot. **CPU Card:**
DBB#	I/O		1	**Data Bus Busy** Indicates that the data bus is busy. Used to qualify DBG# for ownership of the data bus. **Motherboard:** 10K pullup. This signal is bussed (connected) to each CPU slot. If fast-L2/data streaming mode is desired, resistor R232 must be removed to avoid bus contention between two processors. **CPU Card:** CPU cards designed to run in fast-L2/data streaming mode may require special wiring of this signal to place the CPU in this mode and to avoid bus contention on this signal. (For a 604 CPU card without a serial L2, fast-L2/data streaming mode can be implemented by adding a 10K pullup to DBB#.) Also see DRTRY# and DBG#.
DBG#	O		1	**Data Bus Grant** Indicates that the CPU or other busmaster can, with proper qualification, assume mastership of the data bus. **Motherboard:** This signal is bussed (connected) to each CPU slot. **CPU Card:** CPU cards designed to run in fast-L2/data streaming mode may require special wiring of this signal. Also see DRTRY# and DBB#.
DRTRY#	O		1	**Data Retry** Indicates that the busmaster must invalidate the data from the previous read data beat. **Motherboard:** 10K pullup. Fast-L2/data streaming mode is supported by the 660 Bridge. The mother board is wired so that it defaults to DRTRY# mode. This signal is bussed (connected) to each CPU slot. **CPU Card:** CPU cards designed to run in fast-L2/data streaming mode may require special wiring of this signal to place the CPU in this mode and avoid bus contention on this signal. (For a 604 CPU card without a serial L2, fast-L2/data streaming mode can be implemented by connecting DRTRY# on the 604 to HRESET# on the CPU card slot. Leave #DRTRY on the CPU card slot unconnected and floating.) Also see DBB# and DBG#.

Table 4. CPU Slot Signal Descriptions (Continued)

Signal	I/O	Level	Note	Description
604 Bus Signals				
DH[0:31]	I/O		1	**Data Bus High** Most significant 4 bytes of the data bus. Data Bus Signals Byte Lane DH[0:7] .... 0 (MSB) DH[8:15] .... 1 DH[16:23] .... 2 DH[24:31] .... 3 **Motherboard:** These signals are bussed (connected) to each CPU slot and the L2 slot. **CPU Card:**
DL[0:31]	I/O		1	**Data Bus Low** Least significant 4 bytes of the data bus. Data Bus Signals Byte Lane DL[0:7] .... 4 DL[8:15] .... 5 DL[16:23] .... 6 DL[24:31] .... 7 (LSB) **Motherboard:** These signals are bussed (connected) to each CPU slot and the L2 slot. **CPU Card:**
DP[0:7]	I/O		1	**Data Bus Parity** Indicates the parity by byte of the CPU data bus (1 = odd, 0 = even). Parity Bit Byte Lane Parity Bit Byte Lane DP[0] .... 0 DP[4] .... 4 DP[1] .... 1 DP[5] .... 5 DP[2] .... 2 DP[6] .... 6 DP[3] .... 3 DP[7] .... 7 **Motherboard:** The 660 Bridge checks data bus parity (using DP[0:7], DH[0:31], and DL[0:31]) during CPU writes. During CPU read transfers, the 660 Bridge drives DP[0:7] with the (odd) parity information stored in the DRAM (if there is an L2 hit, the L2 supplies the stored parity information). These signals are bussed (connected) to each CPU slot and to the L2 slot. **CPU Card:**
DPE#	I		1	**Data Parity Error** Indicates that the busmaster has detected a parity error. **Motherboard:** During CPU to memory reads that result in L2 hits, the 660 Bridge monitors DPE# to detect CPU data bus parity errors. This signal is bussed (connected) to each CPU slot. **CPU Card:** If parity is not supported by the CPU card, this signal should be pulled up by a 10K ohm resistor on the CPU card.
GBL#	I/O		1	**Global** Indicates that the current transaction should be snooped by other busmasters. **Motherboard:** This signal is bussed (connected) to each CPU slot. **CPU Card:**

Table 4. CPU Slot Signal Descriptions (Continued)

Signal	I/O	Level	Note	Description
604 Bus Signals				
SHD#	I/O		1	**Shared** Indicates a cache hit on a shared block. If asserted with ARTRY#, then the asserting busmaster will perform a snoop push of modified data. **Motherboard:** 10K pullup. This signal is bussed (connected) to each CPU slot. **CPU Card:**
TA#	O		1	**Transfer Acknowledge** Asserted by the target of the CPU transfer to indicate that the current data beat has been accepted. For each CPU clock that TA# is asserted, a data beat completes. For a single-beat (8-byte) data tenure, TA# is only one clock. For a four-beat (32-byte) burst, the data tenure competes on the fourth cycle in which TA# is asserted. **Motherboard:** 10K pullup. This signal is bussed (connected) to each CPU slot. **CPU Card:**
TBST#	I/O		1	**Transfer Burst** Indicates that a burst (32-byte) transfer is in progress. **Motherboard:** 10k pullup. This signal is bussed (connected) to each CPU slot. **CPU Card:**
TEA#	O		1	**Transfer Error Acknowledge** Indicates that an exception has occurred and that the CPU should take a machine check exception. Assertion of TEA# terminates the current data beat and tenure. **Motherboard:** 10K pullup, TEA# can be masked on the motherboard via the bridge chipset control registers. This signal is bussed (connected) to each CPU slot. **CPU Card:**
TS#	I/O		1	**Transfer Start** Indicates the start of a memory or memory mapped I/O transaction by a busmaster and that the address bus and address transfer attributes are valid. **Motherboard:** 10k pullup. This signal is bussed (connected) to each CPU slot. **CPU Card:**
TSIZ[0:2]	I/O		1	**Transfer Size** This 3 bit transfer size encoding, in conjunction with TBST#, indicates the size of the current CPU data bus beat. TBST# TSIZ(0..2) Transfer Size 0 010 Burst (32 bytes) 1 000 8 bytes 1 001 1 byte 1 010 2 bytes 1 011 3 bytes 1 100 4 bytes 1 101 5 bytes 1 110 6 bytes 1 111 7 bytes **Motherboard:** These signals are bussed (connected) to each CPU slot. **CPU Card:**

Table 4. CPU Slot Signal Descriptions (Continued)

Signal	I/O	Level	Note	Description
604 Bus Signals				
TT[0:4]	I/O		1	**Transfer Type** 5 bit encoded transfer type of the current bus transaction. See the *604 User's Manual* and Table 2 for more information. **Motherboard:** See the 660 Bridge User's Manual for more information. These signals are bussed (connected) to each CPU slot. **CPU Card:**
XATS#	I		1	**Extended Address Transfer Start** Indicates the start of a direct store operation on the bus. **Motherboard:** 10K pullup. Unsupported operation on the motherboard and will result in an exception (TEA# asserted). This signal is bussed (connected) to each CPU slot. **CPU Card:** Do not assert XATS#.
L2 Controller				
L2_BR#	I		2	**L2 Bus Request** Indicates that the L2 cache controller or second master is requesting mastership of the address bus. **Motherboard:** No connect. The motherboard does not support a second master in the same CPU card slot. For each CPU slot, BR# is hardwired to the arbiter, and L2_BR# is unconnected. The L2_BR# for the slot can be connected to the BR# for the slot by installing a 0Ω resistor (R76 for CPU slot 1 and R20 for CPU slot 2). **CPU Card:** CPU cards must contain only one busmaster. This busmaster must use the primary bus request for the slot, BR#.
L2_BG#	O		2	**L2 Bus Grant** Indicates the L2 cache controller or second master can assume ownership of the address bus (given proper qualification by ABB# and ARTRY#). **Motherboard:** No connect. The motherboard does not support a second master in the same card slot. For each CPU slot, BG# is hardwired to the arbiter, and L2_BG# is unconnected. The L2_BG# for the slot can be connected to the BG# for the slot by installing a 0Ω resistor (R66 for CPU slot 1 and R83 for CPU slot 2). **CPU Card:** CPU cards must contain only one busmaster. This busmaster must use the primary bus grant for the slot, BG#.
L2_CLAIM#	I		2	**L2 CPU Bus Claim** Indicates that a CPU bus target (e.g., an L2 cache controller) is claiming the CPU bus transfer and will supply the data, the address bus transfer signals, and the data bus transfer signals, as appropriate, for a target. The 660 Bridge aborts the memory controller cycle and tristates its AACK#, TA#, TEA#, and CPU data bus drivers. **Motherboard:** 10K pullup. This signal is bussed (connected) to each CPU slot. **CPU Card:** The address range in which this signal can be asserted depends on the state of the 660 Bridge. While the 660 Bridge internal L2 is enabled, L2_CLAIM# can be asserted for accesses from the top of memory up to 2G (e.g., if 8M of DRAM is installed, L2_CLAIM# can be asserted from 8M to 2G). While the 660 Bridge internal L2 is disabled, L2_CLAIM# can be asserted for accesses from 0 to 2G.

Table 4. CPU Slot Signal Descriptions (Continued)

Signal	I/O	Level	Note	Description
L2 Controller				
L2_CLR#	O		2	**L2 Cache Clear** Indicates that all of the L2 tags are to be invalidated or cleared. **Motherboard:** This signal is shared with the imbedded L2 cache in the 660 Bridge. An active low pulse is generated on this signal for 1 CPU clock whenever a CPU bus write is performed to 8000 0814h. This signal is bussed (connected) to each CPU slot and the L2 slot. **CPU Card:**
L2_INH#	O		4	**L2 Cache Miss Inhibit** Indicates that the L2 cache should be inhibited from updating the SRAM on all L2 misses. This allows the L2 cache to retain its contents during memory accesses, while maintaining coherency (snooping and tag updates continue to occur). **Motherboard:** This signal is not shared with the imbedded L2 cache on the 660 Bridge. It can only be accessed when the 660 Bridge is in external register mode (see Bridge Chip Set Options 3 BCR in the *660 User's Manual*). In this mode the signal follows the value written to System Control BCR 0 (address 8000 081Ch bit 7. This signal is bussed (connected) to each CPU slot. **CPU Card:**
L2_FLUSH#	O		4	**L2 Cache Flush** Indicates that the L2 in this slot should write all modified lines to memory and mark all lines as invalid. **Motherboard:** This signal is not shared with the imbedded L2 cache on the 660 Bridge. It can only be accessed when the 660 Bridge is in external register mode (see Bridge Chip Set Options 2 BCR in the *660 User's Manual*). In this mode, the signal follows the value written to System Control BCR 0 (address 8000 081Ch bit 4). This signal is bussed (connected) to each CPU slot. **CPU Card:**
L2_DISABLE#	O		4	**L2 Cache Disable** Indicates that the L2 in this slot should be disabled. This signal can be used by the system to inhibit all operations of the cache without invalidating the data in the cache. It is the responsibility of the system to maintain coherency in this case. **Motherboard:** This signal is not shared with the imbedded L2 cache on the 660 Bridge. It can only be accessed when the 660 Bridge is in external register mode (see Bridge Chip Set Options 2 BCR in the *660 User's Manual*). In this mode the signal follows the value written to System Control BCR 0 address 8000 081Ch bit 6. This signal is bussed (connected) to each CPU slot. **CPU Card:**
CONFIG#	O		4	**Configuration Enable** A low on this signal enables a serial L2 cache to decode ranges of CPU bus addresses as configuration transfers. These transfers will not be forwarded to the system bus. **Motherboard:** This signal is a programmable I/O and follows the value of the L2 Control Register located at 8000 086Bh bit 2. This signal is bussed (connected) to each CPU slot. **CPU Card:** These transfers should be intercepted by the serial cache and not forwarded to the system bus. When two CPU cards with serial caches are installed, software must prevent CPU_x from performing unintended configuration cycles while CPU_y is using the signal to configure its serial cache. Software must sequence L2 configuration cycles.

Table 4. CPU Slot Signal Descriptions (Continued)

Signal	I/O	Level	Note	Description
L2 Controller				
L2_WT#	O		4	**L2 Cache Write Thru Only** A low Indicates that the L2 cache in this slot should operate in write thru mode only. **Motherboard:** This signal is a programmable I/O and follows the value of the L2 Control Register located at 8000 086Bh bit 1. This signal is bussed (connected) to each CPU slot. **CPU Card:**
L2_AACK_EN	O		4	**L2 AACK Bi-Directional enable** A high on this signal enables the L2 in this slot to drive AACK#. A low indicates that the L2 should treat this signal as input only. **Motherboard:** Always pulled up to 3.3v via a 10 K pullup. On the motherboard, the pullup resistors (R19 for slot 1 and R7 for slot 2) can be removed and pulldown resistors added (R12 for slot 1 and R86 for slot 2). There is an individual L2_AACK_EN for each CPU slot. **CPU card:** L2 cards designed to drive AACK# must also drive L2_CLAIM# to avoid bus contention on AACK#.
Clock/Interrupts/Resets				
BUS_CLK[0:2]	O		1	**System Bus Clocks** Bus Clocks for the CPU bus. **Motherboard:** Nominally 66MHz or 60MHz, depending on the FREQ_ID[0:3] of both slots. If no card is present, system firmware may disable these clocks via the Freeze Clock Registers at ports 8000 0860h and 8000 0862h. See Section 10 (ELPD) for information on the operation of these ports. There are three individual BUS_CLKs for each CPU slot. **CPU Cards:** To minimize skew and other clock problems, make each clock trace 3.75" +/– .1". Place exactly one load on each clock line that is used.
INT_A#	O		1,3	**CPU Interrupt** Active low, level sensitive interrupt signal to the CPU. **Motherboard:** Slot and CPU specific interrupt output to the CPU in the card slot. These outputs come from the MPIC device on the motherboard. There is an individual INT_A# for each CPU slot. INT_A# from CPU slot 1 connects to the INT0# out of MPIC. INT_A# from CPU slot 2 connects to the INT1# output of MPIC (see Section NO TAG). **CPU Card:**
CHECKSTOP#	I		1	**Checkstop Out** Indicates that the CPU in this slot has detected a check stop condition and has ceased operation. **Motherboard:** 10K pullup. There is an individual CHECKSTOP# for each CPU slot. The RISCWatch system uses this signal to monitor the status of the CPUs. **CPU Card:**

Table 4. CPU Slot Signal Descriptions (Continued)

Signal	I/O	Level	Note	Description
Clock/Interrupts/Resets				
MCP_A#	I/O		1,3,4	**Machine Check Interrupt** Initiates a machine check interrupt to the CPU. May be asserted asynchronously. **Motherboard:** 330Ω pullup to 3.3v. This signal is used to report system errors to the CPU. The 660 Bridge asserts this signal for two CPU clock cycles in the event of a catastrophic or unrecoverable system error. The MCP_A# signal is not processor or slot specific. **CPU Cards:** The MCP_A# signal should be wire OR'd to any device on the CPU card which monitors or asserts MCP#. Devices which assert MCP_A# must use open drain outputs to avoid contention with other devices. No pullup or pulldown resistors should be connected.
SRESET_A#	O		1,3,4	**Soft Reset** Asserting initiates a soft reset exception in the CPU. **Motherboard:** The SRESET_A# signal is driven by logic on the motherboard which allows three methods of asserting SRESET_A#. A CPU specific soft reset can be issued by the MPIC to any individual CPU in the system. A global soft reset can be generated by writing a 0 to port 8000 0092h bit 0, or by the RISCWatch interface. There is an individual SRESET_A# for each CPU slot. **CPU Card:**
HRESET_A#	O		1,4	**Hardware Reset** Asserting initiates a hard reset operation in the CPU. **Motherboard:** The HRESET_A# signal is driven by logic on the motherboard which allows three methods of asserting HRESET_A#. A CPU specific hard reset can be issued via the Processor Enable Register at port 8000 0871h (see Section 6). A global hard reset can be generated by the power supply power good indicator or via the RISCWatch interface. There is an individual HRESET_A# for each CPU slot. **CPU Card:**
HALT_A/ **QREQ_A**	I		1,4	**HALTED** Indicates that the internal clocks have been stopped on the CPU due to entering a power management state. **Motherboard:** 10K pullup. Logic on the motherboard monitors the HALT_A/QREQ_A inputs from each CPU slot, and deasserts the RUN signal to both CPUs when they both have halted. There is an individual HALT_A/QREQ_A for each CPU slot. **CPU Card:**
RUN/QACK	O		1,4	**Run** When asserted, RUN/QACK indicates that the CPU must force internal clocks to run during power managed Nap mode, thus allowing bus transactions to be snooped. When deasserted, it allows the internal CPU clocks to be stopped, which will suspend CPU snooping of CPU bus operations. **Motherboard:** Common to all CPU slots, RUN is used in conjunction with the HALT signal to to manage the entry of the CPUs into the power managed modes. **CPU Card:**

Table 4. CPU Slot Signal Descriptions (Continued)

Signal	I/O	Level	Note	Description
Clock/Interrupts/Resets				
SMI#	O		1,4	**System Management Interrupt** Asynchronous interrupt that indicates that the CPU should initiate an SMI operation. Often used by the system to force the CPU out of power managed states. **Motherboard:** This signal is held deasserted by the motherboard, and is bussed (connected) to each CPU slot. **CPU Card:**
TBEN	O		1	**Timebase Enable** Indicates that the CPU time base counter should continue clocking. **Motherboard:** 10K pullup. This signal is bussed (connected) to each CPU slot. **CPU Card:**
TCK	O		1	**JTAG Clock** Serial clock for JTAG interface **Motherboard:** 10K pullup. This signal is connected to the RISCWatch connector pin 5, and is bussed (connected) to each CPU slot. **CPU Card:** CPU cards with more than one JTAG capable device should connect this signal to all device shift clock inputs.
TRST#	O		1	**JTAG TEST Reset** Resets JTAG logic. **Motherboard:** Logic on the motherboard asserts TRST# during hardware reset of the CPU to ensure proper resetting of the CPU. The motherboard also asserts TRST# if the RISCWatch interface has asserted OCS_OVERRIDE (J18 pin 15). There is an individual TRST# for each CPU slot. **CPU Card:** CPU cards with more than one JTAG capable device should connect this signal to all JTAG reset inputs.
TDI	O		1	**JTAG Test Data Input** This is the scan string data connection to the first JTAG device input in the CPU card JTAG chain. **Motherboard:** 10K pullup. There is an individual TDI for each CPU slot. The motherboard connects this signal to the RISCWatch connector J18 pin 7 (SCAN_IN) via the J81 jumpers, which can be set to configure the scan chain as shown: Install Jumper Scan Chain J81 1–3 & 4–6 .... Slot 1 only 1 ** 2 2–4 & 3–5 .... Slot 2 only 3 ** 4 1–3, 2–4, & 5–6 .. Slot 1 —> Slot 2 5 ** 6 1–2, 3–5, & 4–6 .. Slot 2 —> Slot 1 **CPU Card:** CPU cards with more than one JTAG capable device should connect this signal to the first JTAG device input in the loop.
TDO	I		1	**JTAG Test Data Output** This is the scan string data connection to the last JTAG device output in the loop. **Motherboard:** There is an individual TDO for each CPU slot. The motherboard connects this to the RISCWatch connector J18 pin 8 (SCAN_OUT) via the J81 jumpers, which can be used to configure the scan chain as shown for the TDI signal. **CPU Card:** CPU cards with more than one JTAG capable device should connect this signal to the last JTAG device output in the loop.

Table 4. CPU Slot Signal Descriptions (Continued)

Signal	I/O	Level	Note	Description
Clock/Interrupts/Resets				
TMS	I		1	**JTAG Test Mode Select** JTAG interface signal **Motherboard:** 10k pullup. This signal is bussed (connected) to each CPU slot and is connected to the RISCWatch connector J18 pin 4. **CPU Card:** CPU cards with more than one JTAG capable device should connect this signal to the Test Mode Select input of all devices.
Configuration				
DVR_MOD[0:1]	O			**CPU Output Drive Mode Select** Select the capacity of the bus drivers on CPUs that support this function. **Motherboard:** These signals are bussed (connected) to each CPU slot and are set by resistors on the motherboard. Normal drive mode (0,1) is the default value, and is recommended for the reference design. DRV_MOD[0] DRV_MOD[1] 0 . . . . 0 . . . . Disabled 0 . . . . 1 . . . . Normal (Default) 1 . . . . 0 . . . . Strong 1 . . . . 1 . . . . Herculean **CPU Card:**
FREQ_ID[0:3]	O			**Frequency selection** These outputs determine the PLL settings of the CPU on the CPU card installed in the slot. **Motherboard:** These signals are bussed (connected) to both CPU slots. Logic on the motherboard uses the PD[0:3] inputs to determine the best PLL configurations for the CPUs in the system (see Section 7, Clocking, for details). **CPU Card:** These signals should be connected to the PLL[0:3] inputs of the CPU.

Table 4. CPU Slot Signal Descriptions (Continued)

Signal	I/O	Level	Note	Description
Configuration				
PD[0:3]	I			**CPU ID (and presence detect) inputs** These inputs indicate the desired operating frequency of the CPU on the CPU card. **Motherboard:** 10K pullup. There is an individual PD[0:3] for each CPU slot. Logic on the motherboard uses these inputs to determine the correct operating frequency for the CPU bus and the values that will be presented to the CPUs via the FREQ_ID[0:3] outputs (see Section 7, Clocking, for details). These bits also indicate the absence of a CPU card if all PD bits are high and the software determines that no serial ROM configuration device is present. Also see L2_PD[0:1]. The status of PD[0:3] can be read via the L2 PD Register 8000 080Dh bits [3:0], and have the following meanings. Bit 0 Bit 1 Bit 2 Bit 3 Slot contains 0 0 0 0 No CPU. This is an L2-only card. 0 0 0 1 100 MHz CPU 0 0 1 0 120 MHz CPU 0 0 1 1 132 MHz CPU 0 1 0 0 150 MHz CPU 0 1 0 1 167 MHz CPU 0 1 1 0 180 MHz CPU 0 1 1 1 Reserved 1 0 0 0 Reserved 1 0 0 1 Reserved 1 0 1 0 Reserved 1 0 1 1 Reserved 1 1 0 0 Reserved 1 1 0 1 Reserved 1 1 1 0 Reserved 1 1 1 1 No CPU Card Present or Serial ROM Present * * If bits 3:0 of port 80D are 1, then either no L2 card is present or a serial ROM with configuration information is present. The motherboard can read the serial ROM by driving PD[0] with the serial clock and reading or writing data on PD[1]. This interface is controlled via the Serial ROM control Register 8000 0868h. **CPU Card:** Use a 100Ω, 10% pulldown resistor on those lines that are to be set to 0. Do not connect lines which are to be set to 1.

Table 4. CPU Slot Signal Descriptions (Continued)

Signal	I/O	Level	Note	Description
Configuration				
L2_PD[0:1]	I/0			**L2 Presence Detect 0 and 1** As inputs, these signals tell the motherboard if an L2 cache is present on the CPU card. They can also be used as I/O pins to read a serial ROM that can be attached to L2_PD[0:1], and which contains configuration information about the card in the slot. Also see Section NO TAG, CPU Card. **Motherboard:** 10K pullup. There is an individual L2_PD[0:1] for each CPU slot. The status of L2_PD[0:1] can be read via the CPU 1 PD Register 8000 0866h bits 4, and 5, and CPU 2 PD Register 8000 0867h bits 4 and 5. These bits have the following encoding: bit 4 bit 5 slot contains 0 . . . . . . 0 . . . . . . Reserved 1 . . . . . . 0 . . . . . . TBD 0 . . . . . . 1 . . . . . . TBD 1 . . . . . . 1 . . . . . . No Cache or Serial ROM present * * If bits 5:0 of port 866 or port 867 are 1, then either no board is present or a serial ROM with configuration information is present. The motherboard can read the serial ROM by driving L2_PD[0] with the serial clock and reading or writing data on L2_PD[1]. This interface is controlled via the serial ROM control register 8000 0868h. **CPU Card:** A 100Ω, 10% resistor should be used when required to to pull down these signals. The presence and type of SLC can be determined by CPU by using the CONFIG# signal (SLC information can not be determined by examining the PD bits).
GND				**Logic Ground** **Motherboard:** These are the digital logic ground of the motherboard. They provide the return path for all power supplied to the CPU card.
VCC5				**+5.0v supply** **Motherboard:** These pins are connected to the +5v pins of the motherboard power connector and to the 5v logic on the motherboard.
3.3/3.6V				**+3..3/3.6v supply pins** Can be either 3.3v or 3.6v depending on system implementation. **Motherboard:** these pins are connected to a 3.6v linearly regulated supply on the motherboard. **CPU card: Do not connect devices that cannot tolerate 3.6v supply tolerance.**
3.3V				**3.3v supply** **Motherboard:** These pins are connected to the 3.3v pins of the motherboard power supply connector and to the 3.3v logic on the motherboard.

Notes:

1) Refer to the *PowerPC 604 Users Manual* and *The IBM27-82660 PowerPC to PCI Bridge User's Manual* for the details of definitions and timing of the signal. These signals conform to the function defined in these specifications.
2) Refer to *The IBM27-82660 PowerPC to PCI Bridge User's Manual* for the details of definitions and timing of the signal. These signals conform to the function defined in this specifications.
3) Some signal names are shown as SIG_A. The initial configuration of the reference design included support for 2 CPUs on each CPU card. Thus SIG_A was associated with the first CPU on a particular card, and SIG_B was associated with the second CPU. Signals associated with the second CPU flow over the auxiliary CPU connector. The second CPU on each CPU card and the auxiliary connector are not supported by the reference design at this time.
4) These signals may be asserted asynchronously by the motherboard logic.
5) These signals are unsupported on the CPU slot: APE#, TC[0:3], DBDIS#, CSE[0:1], RSRV#, and DBWO#.

2.7.2 CPU Slot DC Characteristics

Table 5. CPU Slot DC Characteristics

Symbol	Parameter	Min	Max	Unit
V_{IH}	Input High Voltage	2.0	3.78	v[1]
V_{IL}	Input Low Voltage	0.0	0.8	v
I_{IH}	Input High Current		100	uA
I_{IH}	Input Low Current		100	uA
V_{OH}	Output High Voltage	2.4	3.78	v[1]
V_{OL}	Output Low Voltage	0.0	0.4	v
I_{OH}	Output High Current		1.0	mA
I_{OL}	Output Low Current		1.0	mA
I_{TS}	Tristate Leakage Current		100	uA
C_F	Signal Pin Capacitance		20	pF

Note:

1) All devices on CPU Bus must be 5.0v tolerant.
2) These specifications are for the CPU slot envelope. They describe the resources that are supplied to the CPU card by the system, and the constraints placed on the CPU card by the system.

2.7.3 CPU Slot AC Timing

Table 6. CPU Slot AC Timing (5)

Parameter	CPU Bus Signal	60 MHz (1)		66MHz (1)		Unit
		min	Max	min	Max	
Clock Period	BUS_CLK(n)		16.6	15		nS
Clock Duty Cycle	BUS_CLK(n)	40	60	40	60	%
Clock Skew (2)	BUS_CLK(n)		.75		.75	nS
Clock Trace Length	BUS_CLK(n)	3.65	3.85	3.65	3.85	inch (3)
Clock Net Loading	BUS_CLK(n)		10		10	pF (3)
Trace Impedance	All Critical Nets (4)	63	82.5	65	83	Ohms
Critical Signal Trace Length	All Critical Nets (4)		3		3	inch

Notes:

1. CPU bus speed is determined by the capabilities of the 660 Bridge and the installed CPU cards (see Section 7 for determination of bus speed).
2. This is the (supplied) maximum skew between BUS_CLK(n) and any of the other devices on the CPU bus (given that each CPU card meets the requirements of note 3).
3. Wire BUS_CLK(n) point to point between connector and load. Allow a maximum of one load per clock net. Make these nets exactly the trace length specified to match other clock nets on CPU bus, or adjust the trace length to adjust the clock timing with respect to the other devices on the CPU bus.
4. Critical nets should be wired point to point on internal planes. These nets include: TS#, TA#, TBST#, TSIZ[0:2], TT[0:4], DPE#, ABB#, AACK#, ARTRY#, DRTRY#, DBB#, DBG#, BG#, BR#, GBL#, SHD#, DP[0:7], A[0:31], DL[0:31], DH[0:31]. See Section 2.10 for information on the motherboard implementation of these nets.
5. These specifications are for the CPU slot envelope. They describe the resources that are supplied to the CPU card by the system, and the constraints placed on the CPU card by the system.

2.7.4 CPU Slot Power Supplies

Table 7 shows the power supplies that are available to each individual CPU card. Total power available for all CPU cards is also shown.

Table 7. CPU Slot Power Supplies (2)

Voltage	Regulation	Maximum Current Per CPU Slot	Maximum Current Total, Both Slots
+ 5v	+/–5%	2.5 A	5 A
+3.3v	+3.6v / –3.0v	7.5 A	15 A
+3.6v	+/–5%	1 A	2 A

Note:

1. Combined power consumption must be less than 25W per slot.
2. These specifications are for the CPU slot envelope. They describe the resources that are supplied to the CPU card by the system, and the constraints placed on the CPU card by the system.

2.7.5 CPU Slot Thermal Envelope

The reference design CPU card is defined to dissipate a maximum of 25W. The reference design is intended for a broad range of applications. The particular physical implementation of the motherboard, the CPU card, and the enclosure allows the designers to supply air to the CPU card at an average of 200 linear feet per minute. This requires that the ambient temperature, as measured at the leading edge of the card, be maintained at a maximum of 34°C in order to adequately cool the CPU card. Figure 7 shows the airflow direction in the reference system.

The information presented here is intended only for reference, and is not presented as a solution to the thermal challenges of any particular application. Conduct independent thermal design, analysis, and testing of the particular system implementation.

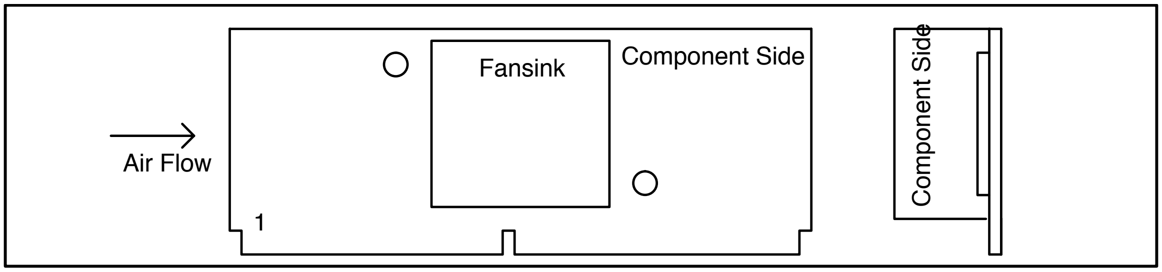


Figure 7. CPU Card Physical Envelope

2.7.6 CPU Slot Card Connector

The CPU card connector is a 2x128 pin DIMM type connector with 1 mm pin spacing. Figure 8 shows an end view of the connector. Figure 8 shows the connector pin assignments.

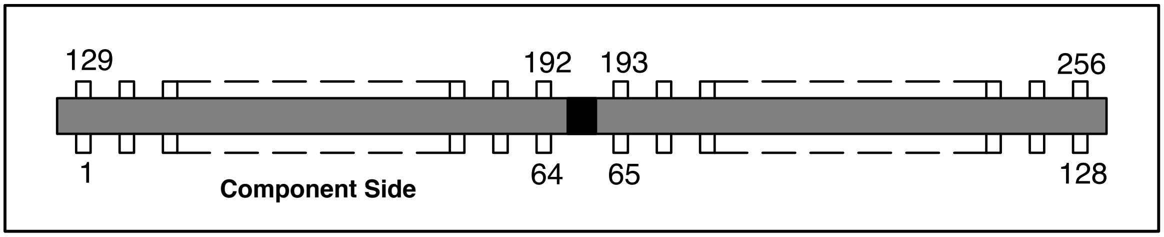


Figure 8. CPU Slot Card Connector (Socket)

Table 8. Main CPU Slot Connector Pin Assignments

Pin No.	Signal Name
1	GND
2	DVR MOD 0
3	PD0
4	PD2
5	GND
6	DH30
7	3.3/3.6V
8	DH28
9	DH26
10	DH24
11	GND
12	DH22
13	DH20
14	DH19
15	DH17
16	DH15
17	GND
18	3.3/3.6V
19	DH12
20	DH11
21	DH9
22	DH7
23	GND
24	DH5
25	DH3
26	DH2
27	DH0
28	GND
29	3.3V
30	3.3V
31	DL28
32	DL26
33	DL24
34	GND
35	DL22
36	3.3V
37	DL20
38	DL18
39	DL16
40	DL14
41	GND
42	3.3V
43	DL12
44	DL11
45	DL9
46	DL7
47	GND
48	3.3V
49	DL4
50	DL3
51	DL1
52	DL0
53	DP5
54	3.3V
55	3.3V
56	DP2
57	DP0
58	GND
59	BUS_CLK 2
60	GND
61	A29
62	A28
63	A26
64	A25
65	A23
66	A21
67	GND
68	DRTRY#
69	GND
70	A19
71	3.3V
72	A17
73	GND
74	A13
75	A12
76	A11
77	A9
78	GND
79	A7
80	A5
81	A3
82	3.3V
83	A0
84	3.3V

Pin No.	Signal Name
85	GND
86	AACK#
87	GND
88	DBG#
89	TA#
90	DBB#
91	TEA#
92	5.0V
93	GND
94	XATS#
95	WT#
96	TT1
97	ARTRY#
98	SHD#
99	GND
100	5.0V
101	ABB#
102	5.0V
103	TT2
104	GND
105	TCK
106	TDI
107	SRESET_A#
108	L2_AACK_EN #
109	CONFIG#
110	GND
111	TRST#
112	5.0V
113	HALT_A / QREQ_A
114	L2_WT#
115	RUN / QACK
116	GND
117	5.0V
118	AP0
119	GND
120	AP1
121	VREF
122	L2_BR#
123	L2_CLAIM#
124	L2_CLR#

Pin No.	Signal Name
125	PWR_DWN
126	FREQ_ID0
127	FREQ_ID2
128	GND
129	DVRMOD1
130	GND
131	PD1
132	PD3
133	DH31
134	DH29
135	GND
136	DH27
137	DH25
138	DH23
139	3.3/3.6V
140	DH21
141	GND
142	DH18
143	DH16
144	DH14
145	DH13
146	3.3/3.6V
147	GND
148	DH10
149	DH8
150	DH6
151	3.3V
152	DH4
153	GND
154	DH1
155	DL31
156	DL30
157	DL29
158	GND
159	DL27
160	DL25
161	DL23
162	3.3V

Pin No.	Signal Name
163	DL21
164	GND
165	DL19
166	DL17
167	DL15
168	DL13
169	3.3V
170	3.3V
171	GND
172	DL10
173	DL8
174	DL6
175	3.3V
176	DL5
177	GND
178	DL2
179	DP7
180	DP6
181	DP4
182	3.3V
183	DP3
184	DP1
185	A31
186	A30
187	GND
188	BUS_CLK 0
189	GND
190	A27
191	3.3V
192	A24
193	A22
194	GND
195	BUS_CLK 1
196	GND
197	3.3V
198	A20
199	A18
200	A16

Pin No.	Signal Name
201	A15
202	A14
203	GND
204	3.3V
205	A10
206	A8
207	A6
208	A4
209	GND
210	A2
211	A1
212	GND
213	3.3V
214	BG#
215	GBL#
216	TSIZ0
217	3.3V
218	GND
219	TSIZ2
220	TSIZ1
221	TS#
222	TT3
223	TMS
224	GND
225	5.0V
226	TT0
227	CHECKSTOP#
228	TT4
229	CI#
230	GND
231	BR#
232	TBST#
233	DPE#
234	TDO
235	HRESET A#
236	Reserved
237	GND
238	5.0V

Pin No.	Signal Name
239	SMI#
240	INT A#
241	L2_PD0
242	GND
243	MCP A#
244	L2_PD1
245	5.0V
246	GND
247	TBEN

Pin No.	Signal Name
248	AP2
249	AP3
250	L2_FLUSH#
251	L2_BG#
252	L2_INH#
253	L2_DISABLE#
254	FREQ_ID1
255	GND
256	FREQ_ID3

2.7.7 Auxiliary CPU Slot Connectors

The auxiliary CPU connectors J15 and J16 are 24 pin headers which provide additional power and sideband signals for supporting a second CPU on the CPU cards. The headers are 2x12 pin on .100 in. centers. J15 is located near and supports CPU slot P1 (J8), and J16 is located near and supports CPU slot P2 (J9).

This interface is for evaluation purposes only, and is not a supported part of the reference design. These connectors may not be implemented on the motherboard or the CPU cards. Figure 9 shows the auxiliary connector, and Table 9 shows the pinout.

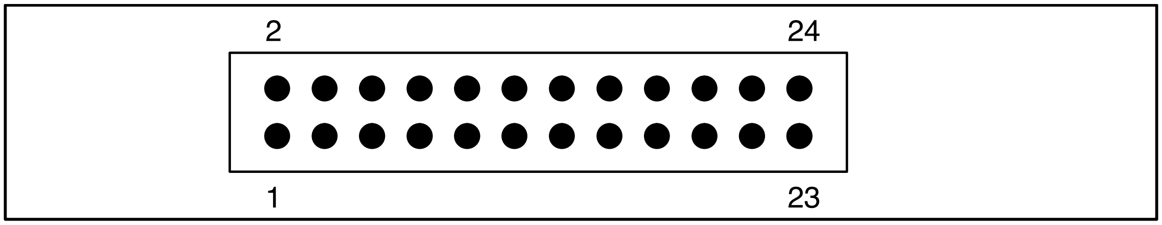


Figure 9. Auxiliary CPU Slot Connector

Table 9. Aux CPU Slot Connector Pinout

Pin No.	Signal Name	Pin No.	Signal Name
1	SRESET_B#	13	+3.3V
2	GND	14	GND
3	HRESERT_B#	15	+3.3V
4	GND	16	+3.3V
5	HALT_B/QREQ_B	17	+3.3V
6	GND	18	GND
7	MCP_B#	19	+3.3V
8	GND	20	+3.3V
9	INT_B#	21	+3.3V
10	GND	22	GND
11	+3.3V	23	Reserved
12	+3.3V	24	+3.3V

2.8 L2 Tag/SRAM Interface (L2 Slot)

There are several types of level 2 (L2) caches that can be used with the reference design. A serial (inline) L2 cache can be added to the CPU card to provide caching and bus traffic filtering for the CPUs. A CPU busmastering look-aside L2 can be installed in one of the CPU card slots to provide write-thru or write-back caching. The L2 card described in this section is installed in the L2 tag/SRAM slot of the reference design motherboard. In conjunction with the L2 cache controller located in the 660 Bridge, the tag/SRAM card provides unified, write-thru, direct-mapped, look-aside level 2 caching for the system 604 bus over the address range from 0 to 1G of the CPU memory space.

The tag/SRAM card provides the tagRAM and SRAM components of the L2. By changing the cards in the slot, the cache can be changed from no cache to an asynchronous cache, to a synchronous cache type. The size of the cache can be 256K, 512K, or 1MB. The size and type of the cache are sensed by the firmware through four presence detect bits defined on the interface.

The size of the tagRAM and data SRAM in the L2 depends on the devices and configuration of the tag/SRAM card. Since the different sizes and configurations are transparent to the motherboard, the L2 size can be changed without changing system configuration. The type (synchronous or asynchronous) of the SRAM and tagRAM affect the number of clock cycles required to access the cache, so the tag/SRAM type may require the firmware to reconfigure the 660 Bridge L2 controller.

The L2 supplies data to the CPU bus on read hits and snarfs the data (updates the SRAM data while the memory controller is accessing DRAM memory) on read/write misses. It snoops PCI to memory transactions. Typical synchronous SRAM read performance with 9ns SRAM is 3-1-1-1, followed by -2-1-1-1 on pipelined reads. Typical asynchronous SRAM read performance with 15ns SRAM is 3-2-2-2, followed by -3-2-2-2 on pipelined reads.

For more information on the operation, capabilities, and configuration of the L2 cache, see the 660 Bridge User's Manual. For information on the particular tag/SRAM card (if any) supplied with the reference design, see the appropriate data sheet in Section NO TAG.

2.8.1 L2 Slot ID ROM

The L2 card ID ROM is a serial EEPROM device which can be installed on certain proposed L2 cards. It is intended to allow systems to support caches which span the full range of performance and features. The ID ROM contains information which may be used by the POST to fully test the cache SIMM. It also contains timing information for the SIMM. This information is specified by the SIMM designer and placed in the EEPROM. At power on, system performance can be tuned for best performance, given the exact capabilities of the SIMM. Diagnostics may test the entire tag/SRAM for verification of its function. In the event that a given card is not a supported configuration for the system in which it is installed, the cache may be disabled without any adverse side effects (e.g., placing a writethrough-only SIMM in a system which requires writeback capability).

The EEPROM contains 128 bits which are organized into16 bytes. Functional capabilities and diagnostic information are typically defined by bytes 0:3. Timing information is contained in bytes 4:7. Bytes 8:15 are reserved for future use. The L2 card ID ROM contains the fields found in Table 10. Note that ID ROM signals are numbered in big endian fashion.

Table 10. L2 Cache ID ROM

Byte Addr	Bit Field	Parameter	Description	Units
0	0	PD	Presence detect. Programmed to zero. The serial data pin should be pulled-up on the CPU motherboard so that cacheless systems return a one when attempting to access the EEPROM.	
0	1	Parity	Parity detect. Identifies that a cache module implements parity RAM.	
0	2	Copyback	Copyback capable. Identifies that a cache module supports copyback by implementing a dirty RAM and having a tag that can drive the appropriate address bits onto the bus.	
0	3	Cache Type	Set if the cache module contains an onboard cache controller. Reset if cache timing is performed by the motherboard.	
0	4		Reserved	
0	5:7	REV	Revision number	
1	0:4	n	Cache size=$2^{(n+1)}$. Identifies the size of the cache. Used to determine the number of tag entries. This information is helpful for diagnostic testing of the tag RAM.	Bytes
1	5:7	p	Cache line size=$2^{(p+1)}$. Used to determine the number of tag entries. This information is helpful for diagnostic testing of the tag RAM.	Bytes
2:3	0:7 0:7	ID	Two character ASCII encoding for manufacturer identification.	
4	0:1	Tag Type	Tag RAM types as follows: 0 – Asynchronous tag RAM 1 – Synchronous tag RAM 2 – Reserved 3 – Reserved.	
4	2		Reserved	
4	3:7	Tag Speed	Binary value encoding the tag RAM match time.	NS
5	0:1	SRAM Type	SRAM pipeline latency as follows: 0 – Asynchronous RAM 1 – Synchronous burst RAM 2 – Synchronous pipeline burst RAM 3 – Reserved.	
5	2		Reserved	
5	3:7	SRAM Speed	Binary value encoding the SRAM read data access time.	NS
6	0:7	FMAX	Synchronous RAM maximum clock frequency.	MHz
7	0:1	SRAM Voltage	SRAM signalling voltage as follows: 0 – 5v signals 1 – 3.3v signals 2 – 2.5v signals 3 – Reserved.	

Table 10. L2 Cache ID ROM (Continued)

Byte Addr	Bit Field	Parameter	Description	Units
7	2:3	Tag Voltage	Tag RAM signalling voltage as follows: 0 – 5v signals 1 – 3.3v signals 2 – 2.5v signals 3 – Reserved.	
7	4		Reserved	
7	5:7	Data Type	Identifier for implementation dependent data.	
8:15		Data Field	Implementation dependent data. For example, this field could contain a heavily compressed low-res JPEG image of the design team.	

2.8.1.1 L2 Cache ID ROM Signaling Interface

The serial communication protocol used by the L2 cache ID ROM is defined as follows. One start bit followed by an eight bit control byte then a returned eight bits of data. The control byte consists of a two bit read command (1,0), a four bit address, and two don't care bits. The four bit address selects which of the 16 bytes to read. The entire sequence is shown in Figure 10 and Table 11.

Table 11. Serial Communications Protocol Sequence

Start	1	0	A3	A2	A1	A0	X	X	D7	D6	D5	D4	D3	D2	D1	D0

The start bit consists of a falling edge on SDA while SCLK is held high. With the exception of the start bit, all transitions on SDA occur while SCLK is low. The read cycle is driven on the bus as shown in Figure 10. For more information see Section 10.2.8, Serial ROM Control Register.

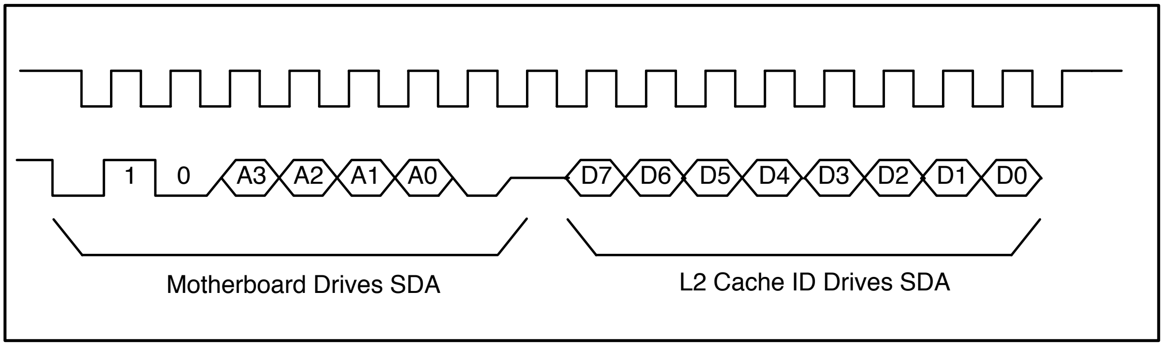


Figure 10. Read Cycle On the Bus

2.8.2 L2 Slot Signal Descriptions

Table 12 describes the signals used to interface the L2 tag/SRAM card to the motherboard. These signals are carried by a 182 pin connector, which provides the 72-bit data and parity bus from the CPU bus, the tagRAM and SRAM address bus, and the tagRAM and SRAM control signals. In Table 12, signals are labeled as inputs (I) or outputs (O) as viewed from the motherboard. Thus ADDR1 is shown as an (O) output because it is an output of the motherboard.

Table 12. L2 Slot Signal Descriptions

Signal	I/O	Level	Notes	Description
60X Bus				
A[0:28]	I/O		1	**Address Bus** Represents a portion of the 32 bit physical address of the current transaction. The address is valid from the bus cycle in which TS# is asserted through the bus cycle in which AACK# is asserted. These signals are big-endian, even though ADDR0 and AADR1 use little endian nomenclature. **Motherboard:** These signals are connected to A[0:28] on the CPU slots. **L2 Card:**
ADDR1	O		2	**Address Bus Bit 1** Represents the LSB+1 bit physical address for asynchronous cache SRAM. **Motherboard:** ADDR1, SRAM_CNT_EN0#, and SRAM_CNT_EN1# are tied together on the motherboard (and connected to RESERVED 2 through a 0Ω resistor). These signals are connected to SRAM_CNT_EN#/ADDR1 on the 660 Bridge. While configured for asynchronous SRAM, the 660 Bridge drives this pin with the second-to-least significant address bit. While configured for synchronous SRAM, the 660 Bridge drives this pin with the SRAM count enable signal. **L2 Card:**
ADDR0	O		2	**Address Bus Bit 0** Represents the LSB bit physical address for asynchronous cache SRAM. **Motherboard:** ADDR0, SRAM_ADS0#, and SRAM_ADS1# are tied together on the motherboard (and connected to RESERVED 1 through a 0Ω resistor). These signals are connected to SRAM_ADS#/ADDR0 on the 660 Bridge. While configured for asynchronous SRAM, the 660 Bridge drives this pin with the least significant address bit. While configured for synchronous SRAM, the 660 Bridge drives this pin with the SRAM address strobe. **L2 Card**:
CLK[0:4]	O		1,2	**System Bus Clocks** Bus clocks for the CPU bus. Tag/SRAM cards will typically use CLK[2] for tagRAM, CLK[0,1] for 512K SRAM cache, and CLK[0,1,3,4] for 1M SRAM cache **Motherboard:** Nominally 66MHz or 60MHz depending on the FREQ_ID[0:3] of both CPU slots. If no L2 card is present, system firmware may disable these clocks via the Freeze Clock Registers at 8000 0860h and 8000 0862h. See section 7 for details on the operation of these ports. **L2 Cards:** To ensure minimal clock skew with respect to the other CPU bus clocks, make these nets 3 in. +/– 0.1 in.

Table 12. L2 Slot Signal Descriptions (Continued)

Signal	I/O	Level	Notes	Description
60X Bus				
DH[0:31]	I/O		1	**Data Bus High** Most significant 4 bytes of the data bus. Data Bus Signals Byte Lane DH[0:7] .... 0 (MSB) DH[8:15] .... 1 DH[16:23] .... 2 DH[24:31] .... 3 **Motherboard:** These signals are connected to DH[0:31] on the CPU slot. **CPU Card:**
DL[0:31]	I/O		1	**Data Bus Low** Least significant four bytes of the data bus. Data Bus Signals Byte Lane DL[0:7] .... 4 DL[8:15] .... 5 DL[16:23] .... 6 DL[24:31] .... 7 (LSB) **Motherboard:** These signals are connected to DL[0:31] on the CPU slot. **L2 Card:**
DP[0:7]	I/O		1	**Data Bus Parity** Indicates the parity by byte of the CPU data bus (1 = odd, 0 = even). Parity Bit Byte Lane Parity Bit Byte Lane DP[0] .... 0 DP[4] .... 4 DP[1] .... 1 DP[5] .... 5 DP[2] .... 2 DP[6] .... 6 DP[3] .... 3 DP[7] .... 7 **Motherboard:** The 660 Bridge checks data bus parity (using DP[0:7], DH[0:31], and DL[0:31]) during CPU writes. During CPU read transfers, the 660 Bridge drives DP[0:7] with the (odd) parity information stored in the DRAM (if there is an L2 hit, the L2 supplies the stored parity information). These signals are connected to DP[0:7] on the CPU slot. **L2 Card:**

Table 12. L2 Slot Signal Descriptions (Continued)

Signal	I/O	Level	Notes	Description
60X Bus				
PD[0:3]	I/0			**Presence Detect** As inputs, these signal tell the motherboard firmware what type of tag/SRAM card is present in the L2 slot. They can also be used to read a serial ROM that can be attached to PD[0:1], and which contains configuration information about the card in the slot (see Section 4, CPU Card). **Motherboard:** 10K pullup. The status of PD[0:3] can be read via the L2 PD Register 8000 080Dh bits [3:0], and have the following meanings. Bit 3 Bit 2 Bit 1 Bit 0 Slot contains 0 0 0 0 Burst 256K No Parity 0 0 0 1 Burst 512K No Parity 0 0 1 0 Burst 1M No Parity 0 0 1 1 Reserved 0 1 0 0 Burst 256K w/Parity 0 1 0 1 Burst 512K w/Parity 0 1 1 0 Burst 1M w/Parity 0 1 1 1 Reserved 1 0 0 0 Asynch 256K No Parity 1 0 0 1 Asynch 512K No Parity 1 0 1 0 Asynch 1M No Parity 1 0 1 1 Reserved 1 1 0 0 Asynch 256K w/Parity 1 1 0 1 Asynch 512K w/Parity 1 1 1 0 Asynch 1M w/Parity 1 1 1 1 No Cache Present or Serial ROM Present * If bits 3:0 of port 80D are 1, then either no L2 card is present or a serial ROM with configuration information is present. The motherboard can read the serial ROM by driving PD[0] with the serial clock and reading or writing data on PD[1]. This interface is controlled via the Serial ROM control Register 8000 0868h. **L2 Card:** 100Ω resistor is used when required to pull down these signals.
SRAM_ADS[1:0]#	O		2	**SRAM Address Strobes** (Synchronous SRAM) Address strobes for Burst SRAM. While asserted, burst SRAM latches in the address on the address lines. **Motherboard:** See ADDR0. **L2 Card:**
SRAM_ALE	O		2	**SRAM Address Latch Enable** (Asynchronous SRAM) Enables latching of address for the SRAMs to support address pipelining when asynchronous SRAMS are used. Always high for burst SRAM **Motherboard:** **L2 Card:**
SRAM_CNT_EN [1:0]#	O		2	**SRAM Count Enables** Enables burst SRAM to increment the burst address on the clock. **Motherboard:** See ADDR1. **L2 Card:**
SRAM_OE[1:0]#	O		2	**SRAM Output Enables** Enables the SRAM output drivers. **Motherboard:** **L2 Card:**

Table 12. L2 Slot Signal Descriptions (Continued)

Signal	I/O	Level	Notes	Description
60X Bus				
SRAM_WE[0:7]#	O		2	**SRAM Write Enables** Enables updating of SRAM with data from the CPU data bus **Motherboard:** All eight WE# pins are wired to the same net on the motherboard. Write updates of less than 8 bytes are not supported. **L2 Card:**
TAG_CLR#	O		2	**TagRAM Clear** Indicates that all entries in the tagRAM should be invalidated. **Motherboard:** This signal is also connected to (shared with) the CPU slot L2 signals. An active low pulse is generated on this signal for one CPU clock whenever a write is performed to 8000 0814h. **L2 Card:**
TAG_MATCH	I		2	**Tag Match** Indicates that a hit in the tagRAM has occurred. **Motherboard:** 200 ohm pullup **L2 Card:**
TAG_VALID	O		2	**Tag Valid** Indicates that the current block should be marked valid in the tagRAM. **Mother board:** **L2 Card:**
TAG_WE#	O		2	**TagRAM Write Enable** Enables the tagRAM to be updated with the current address tag. **Motherboard:** **L2 Card:**
TAG_OE#	O			**TagRAM Output Enable** Enables the tagRAM output to drive the address bus. **Motherboard:** 10K pullup. Only for write back caches, which are not supported by the motherboard. **L2 Card:**
DIRTYIN	O			**Dirty In** **Motherboard:** 1K pull-down. Unsupported function on motherboard. **L2 Card:**
DIRTYOUT	I			**Dirty Output** **Motherboard:** No connection. Unsupported function on motherboard. **L2 Card:**
STANDBY	I			**Standby Power Mode** Indicate that the TAG and SRAMS are to be placed in a low power state. **Motherboard:** 1K pulldown. **L2 Card:**

Table 12. L2 Slot Signal Descriptions (Continued)

Signal	I/O	Level	Notes	Description
60X Bus				
GND				**Logic Ground** **Motherboard:** These pins are connected to the +5v return pins of the power connector, and the the devices on the motherboard.
VCC5				**+5.0v supply** **Motherboard:** These pins are connected to the +5v pins of the power connector and the 5v logic on the motherboard. **L2 Card:**
3.3V				**3.3v supply** **Motherboard:** These pins are connected to the 3.3v pins of the power supply connector and the 3.3v logic on the motherboard **L2 Card:**

Notes

1. Refer to the *PowerPC 604 Users Manual* and *The IBM27-82660 PowerPC to PCI Bridge User's Manual* for the details of definitions and timing of the signal. These signals conform to the function defined in these specifications.
2. Refer to *The IBM27-82660 PowerPC to PCI Bridge User's Manual* for the details of definitions and timing of the signal. These signals conform to the function defined in this specifications.

2.8.3 L2 Slot DC Characteristics

Table 13. L2 Slot DC Characteristics (2)

SYM	Parameter	Min	Max	Unit
V_{IH}	Input High Voltage	2.0	3.78	v[1]
V_{IL}	Input Low Voltage	0.0	0.8	v
I_{IH}	Input High Current		100	uA
I_{IL}	Input Low Current		100	uA
V_{OH}	Output High Voltage	2.4	3.78	v[1]
V_{OL}	Output Low Voltage	0.0	0.4	v
I_{OH}	Output High Current		1.0	mA
I_{OL}	Output Low Current		1.0	mA
I_{TS}	Tristate Leakage Current		100	uA
C_F	Signal Pin Capacitance		20	pF

Notes:

1. All devices on CPU Bus must be 5.0v tolerant. Use of 5.0v SRAM is not recommended due to the possibility that additional slewing time required for 5.0v operation may limit maximum frequency.
2. These specifications are for the L2 slot envelope. They describe the resources that are supplied to the L2 card by the system, and the constraints placed on the L2 card by the system.

2.8.4 L2 Slot AC Timing

Table 14. L2 Slot AC Timing (5)

Parameter	CPU Bus Signal	60 MHz[1]		66MHz[1]		Unit
		Min	Max	Min	Max	
Clock Period	BUS_CLK(n)		16.6	15		nS
Clock Duty Cycle	BUS_CLK(n)	40	60	40	60	%
Clock Skew [2]	BUS_CLK(n)		.75		.75	nS
Clock Trace Length	BUS_CLK(n)	2.9	3.1	2.9	2.9	inches[3]
Clock Net Loading	BUS_CLK(n)		10		10	pF[3]
Trace Impedance	All Critical Nets [4]					Ohms
Critical Signal Trace lengths	All Critical Nets [4]		2		2	inches

Notes:

1) CPU Bus speed is determined by the capabilities of processor(s) cards installed in the system (see Section 7 for determination of bus speed.
2) Skew between BUS_CLKs of other devices on the CPU bus when L2 Tag/SRAM card meet requirements of note 3.
3) BUS_CLK(n) should be wire point to point between connector and single load at the trace length specified to match other clock nets on CPU bus, trace length maybe varied to adjust clock skew of devices on CPU cards.
4) Critical nets should be wired point to point. These nets include: DP[0:7], A[0:31], DL[0:31], DH[0:31]. See Section 2.10 for information on the motherboard implementation of these nets.
5) These specifications are for the L2 slot envelope. They describe the resources that are supplied to the L2 card by the system, and the constraints placed on the L2 card by the system.

2.8.5 L2 Slot Power Supplies

Table 15 shows the power supplies that are available to the L2 card.

Table 15. L2 Slot Power Supplies (2)

Voltage	Regulation	Maximum Current
+ 5v	+/–5%	2.0A
+3.3v	+3.6v / +3.0v	2.0A

Notes:

1) Combined power consumption must be less than 9W.
2) These specifications are for the L2 slot envelope. They describe the resources that are supplied to the L2 card by the system, and the constraints placed on the L2 card by the system.

2.8.6 L2 Slot Thermal Envelope

The reference design L2 tag/SRAM card is defined to dissipate a maximum of 9W. The reference design is intended for a broad range of applications. The particular physical system implementation of the motherboard, the CPU board, the tag/SRAM card, and the enclosure that was tested in our lab delivers air to the L2 card at an average of 25 linear feet per minute (50 to 66 fpm on the side of the L2 card nearest the CPU cards, and 6 to 23 fpm (reverse flow) on the other side of the L2 card). This requires the ambient temperature to be maintained at a maximum of 34°C in order to adequately cool the CPU card. Figure 11 shows the airflow direction in the reference system.

The information presented here is intended only for reference, and is not presented as a solution to the thermal challenges of any particular application. Conduct independent thermal design, analysis, and testing of the particular system implementation.

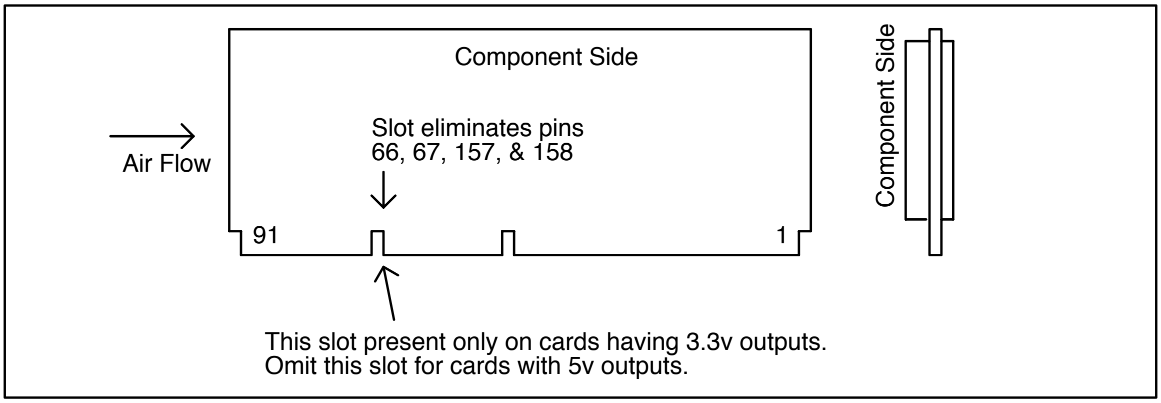

Component Side
Air Flow
Slot eliminates pins 66, 67, 157, & 158
91
1
Component Side
This slot present only on cards having 3.3v outputs.
Omit this slot for cards with 5v outputs.

Figure 11. L2 Tag/SRAM Card Airflow and Keying

2.8.7 L2 Slot Dual Voltage Capability

The L2 Tag/SRAM cards designed for this connector may use either 5v or 3.3v data outputs. The motherboard has been designed to tolerate 5v signals on the CPU bus and the tag/SRAM interface; however, due the increase in slewing time for 5v signals, the CPU bus maximum operating frequency can be reduced when 5v data SRAM is used. It is recommended that 3.3v output data SRAM be used.

Because some motherboards may be designed which are not 5v tolerant, the L2 card design allows the motherboard connector to be keyed such that it will not accept tag/SRAM cards that require V_{DD} = 5v. See Figure 11.

Tag/SRAM connector pins 66, 67, 157, and 158 are assigned as +5v power pins. Motherboards that are not 5v tolerant will use a connector which replaces these pins with a blocking key. All 3.3v output tag/SRAM cards will have a notch cut at the corresponding position. This will allow a tag/SRAM card with 3.3v data outputs to be plugged into either a 5v tolerant board or one that is only 3.3v tolerant. Tag/SRAM cards using 5v outputs must not have the notch cut at this position. The blocking key on the motherboard connector prevents the 5v tag/SRAM card from plugging into a motherboard which cannot tolerate their output voltage level.

Even though the above four +5v pins are removed, other +5v pins remain available for use. This allows 3.3v cards to use 5v tagRAM and SRAM which has 3.3v output levels.

Pin numbering is consistent between the 5v and 3.3v tag/SRAM cards. This means that on 3.3 volt cards, pins 66, 67, 157, and 158 are missing.

See Section 16.5 for more information.

2.8.8 L2 Slot Connector

The L2 tag/SRAM connector is a 2X92 pin type connector with 1.27mm pin spacing. Figure 12 shows an end view of the connector (socket). Table 16 shows the connector pin assignments. This is a 5v tolerant connector.

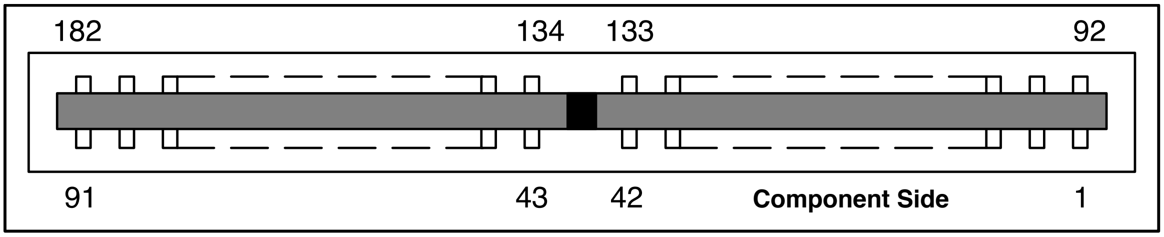


Figure 12. L2 Slot Card Connector (Socket)

2.8.9 L2 Slot Pin Assignments

Table 16. L2 Slot Pinout

Pin No.	Signal Name
1	GND
2	PD0/IDS_CLK
3	PD2
4	DH30
5	DH28
6	DH26
7	DH24
8	VCC3.3
9	DP3
10	DH22
11	DH20
12	DH19
13	GND
14	DH17
15	DP2
16	DH15
17	DH12
18	VCC5
19	DH11
20	DH9
21	DP1
22	DH7
23	VCC3.3
24	DH5
25	DH3
26	DH2
27	DH0
28	DP0
29	GND
30	CLK1
31	GND
32	DL28
33	DL26
34	DL24
35	DP7
36	VCC5
37	DL22
38	DL20
39	DL18
40	DL16
41	GND
42	DP6
43	DL14
44	DL12
45	DL11
46	GND
47	DL9
48	DP5
49	DL7
50	DL4
51	VCC3.3
52	DL3
53	DL1
54	DL0
55	GND
56	CLK2 (for TagRAM)
57	GND
58	DP4
59	SRAM_OE0#
60	SRAM_OE1#
61	VCC3.3
62	ADDR0
63	RESERVED
64	SRAM_ADS0#
65	SRAM_ADS1#
66	VCC5
67	VCC5
68	A28
69	A26
70	A25

Pin No.	Signal Name
71	A23
72	GND
73	A21
74	A19
75	A17
76	A13
77	VCC3.3
78	A12
79	A11
80	A9
81	GND
82	A7
83	A5
84	A3
85	A0
86	VCC5
87	TAG_CLR#
88	TAG_MATCH
89	TAG_OE#
90	DIRTYIN
91	GND
92	GND
93	PD1/IDS_DATA
94	PD3
95	DH31
96	DH29
97	DH27
98	DH25
99	VCC3.3
100	SRAM_WE3#
101	DH23
102	DF21
103	DH18
104	GND
105	DH16
106	SRAM_WE2#
107	DH14
108	DH13

Pin No.	Signal Name
109	VCC5
110	DH10
111	DH8
112	SRAM_WE1#
113	DH6
114	VCC3.3
115	DH4
116	GND
117	CLK0
118	GND
119	DH1
120	SRAM_WE0#
121	DL31
122	DL30
123	GND
124	DL29
125	DL27
126	DL25
127	VCC5
128	SRAM_WE7#
129	DL23
130	DL21
131	DL19
132	GND
133	DL17
134	SRAM_WE6#
135	DL15
136	DL13
137	GND
138	DL10
139	DL8
140	SRAM_WE5#
141	DL6
142	142
143	VCC3.3
144	DL2
145	GND
146	CLK3

Pin No.	Signal Name
147	GND
148	CLK4
149	GND
150	SRAM_WE4#
151	SRAM_ALE
152	VCC3.3
153	ADDR1
154	RESERVED
155	SRAM_CNT_EN0#
156	SRAM_CNT_EN1#
157	VCC5
158	VCC5
159	A27
160	A24
161	A22
162	A20
163	GND
164	A18

Pin No.	Signal Name
165	A16
166	A15
167	A14
168	VCC3.3
169	A10
170	A8
171	A6
172	GND
173	A4
174	A2
175	A1
176	BURST_MODE
177	VCC5
178	TAG_VALID
179	TAG_WE#
180	STANDBY
181	DIRTYOUT
182	GND

2.9 JTAG/RISCWatch Interface

The PowerPC family of processors provides a diagnostic interface for the hardware and software developer. This interface uses standard JTAG connections to allow the RISC-Watch development tool access to the processor. In systems with more than one device, a multiple device JTAG loop can be created. Figure 13 shows the single RISCWatch interface connector (J18) on the motherboard, which can be connected to each CPU card slot individually or to both CPU card slots in series by changing Jumper positions on J81. Table 17 shows the pin assignments.

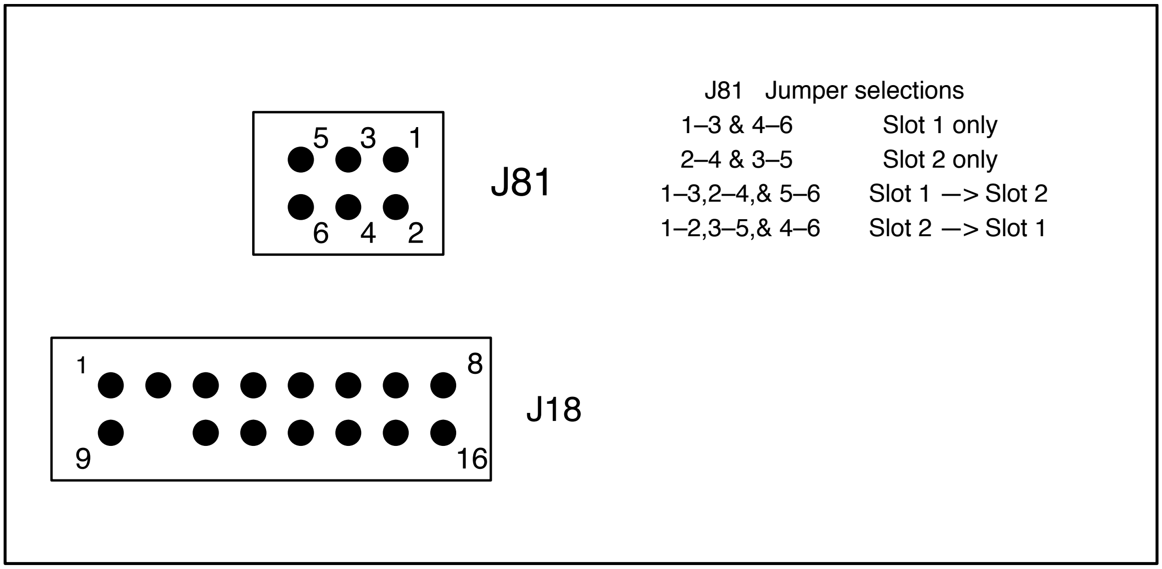


Figure 13. JTAG/RISCWatch Connector and Configuration Jumpers

Table 17. J18 Pin Assignments

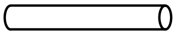
Pin No.	Signal Name	Pin No.	Signal Name
1	CHECK_STOP#	9	GND
2	HRESET_ESP#	10	KEY
3	SRESET_ESP#	11	GND
4	CNTL/SCAN_DATA (TMS)	12	RESERVED1
5	SHIFT_CLOCK (TCK)	13	RESERVED2
6	RUN/BREAKPOINT	14	+3.3V
7	SCAN_IN (TDI)	15	OCS_OVERRIDE (TRST)
8	SCAN_OUT (TDO)	16	RESERVED3

2.10 Electrical Model of Major Signal Groups

This section describes an electrical model of the critical signal paths in the reference design. The complete model of a path consists of the 660 Bridge model, the model of the signal paths on the motherboard, the CPU and L2 slot connector models, the model of the L2 card, and the model of the CPU card(s).

2.10.1 Motherboard Electrical Model

The physical signal paths (traces) of some major signal groups are modeled in the following figures, where:

-  indicates a trace length that is modeled as a transmission line, where
  $Z_o = 70\Omega$ . . . . Characteristic Impedance,
  $T_D = 177$ps/in . . . . Time Delay, and
  $R_S = 159\Omega$/in . . . . DC Resistance.
  L gives the maximum length of the trace in inches.
- The resistors shown are components that are included in the net.

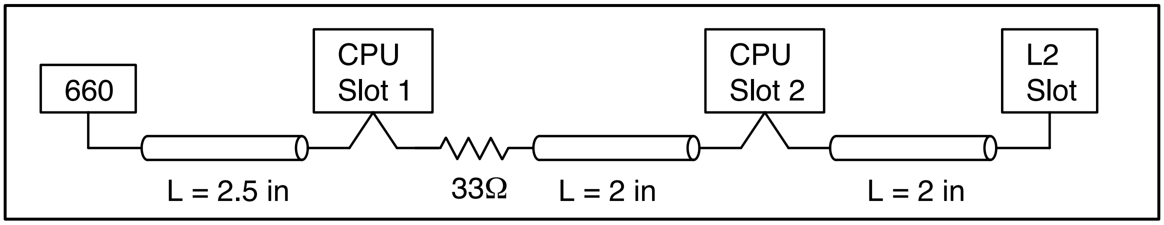


Figure 14. Data and Data Parity Path

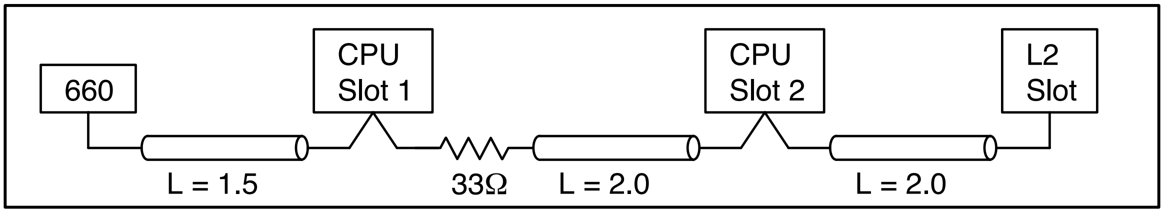


Figure 15. Address Path

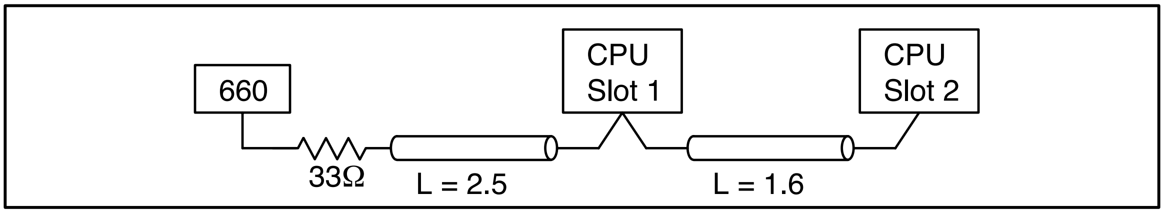


Figure 16. Control Path (TA#,TS#, AACK#, ARTRY#, TT[0:3])

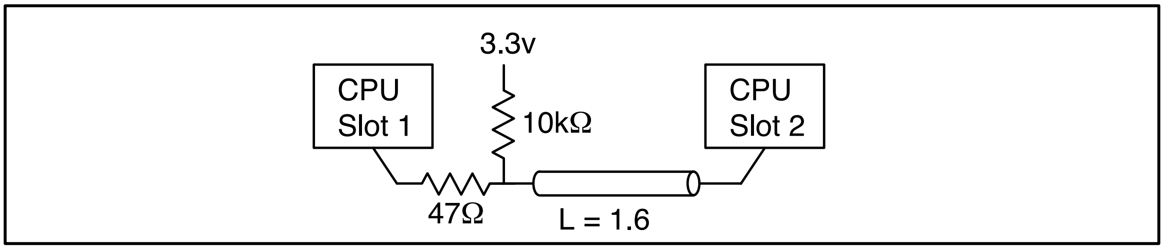


Figure 17. DBB# Path

2.10.2 CPU Card and L2 Card Interface Models

Figure 18 shows the model of the CPU slot interfaces, and Figure 19 shows the model of the L2 slot interface. Both interfaces are modeled in the mated condition, and the values shown are typical.

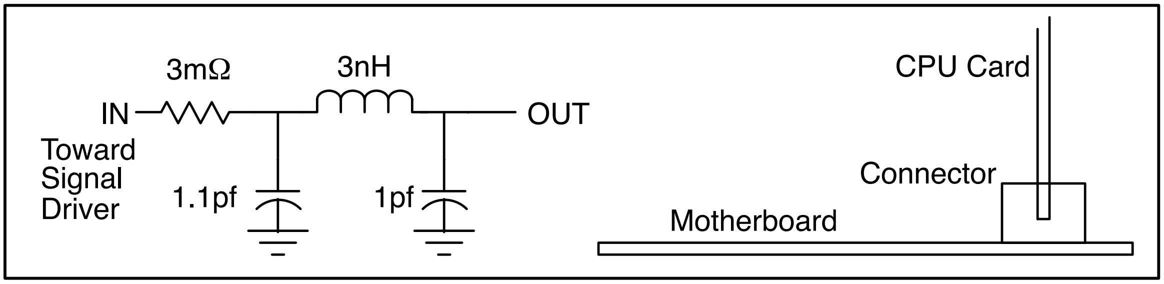


Figure 18. Model of CPU Card to Motherboard Interface (CPU Slot)

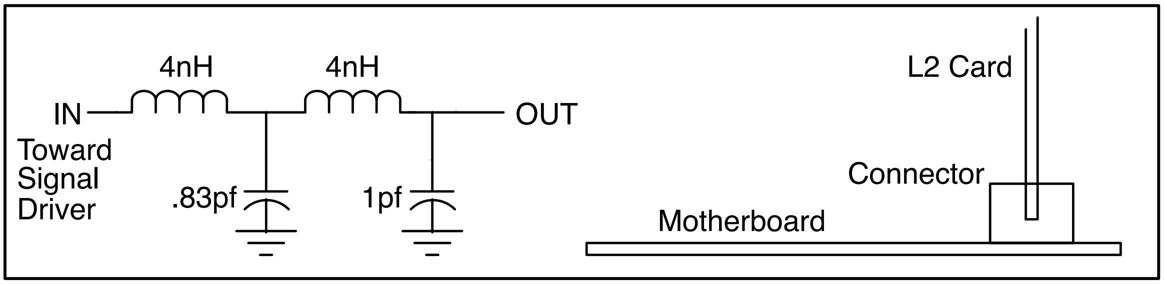


Figure 19. Model of L2 Card to Motherboard Interface (L2 Slot)

2.10.3 CPU Card Model

See Section 4.4 for the electrical model of the Cheetah0 PowerPC 604 CPU Card. For other CPU cards, see their data sheet for model information.

2.10.4 L2 Card Model

See the data sheet of the L2 Card (if any) for electrical model information.

2.10.5 660 Bridge Electrical Model

For the 660 Bridge electrical model, see the 660 Bridge User’s Manual.

2.10.6 Model Building

When constructing the combined model of the net with the interfaces, connect the IN node of the interface model to the motherboard if the signal is an output from the motherboard. On the other hand, if the signal is an output of the plug-in card, connect the IN node of the interface model to the plug-in card. The model of the CPU (or L2) card is placed on the other side of the interface model.

For example, Figure 20 models the address lines during a CPU bus operation mastered by the CPU card in slot 1. The IN node of the slot 1 model connects to the CPU card. The IN node of the other slot models connect to the motherboard.

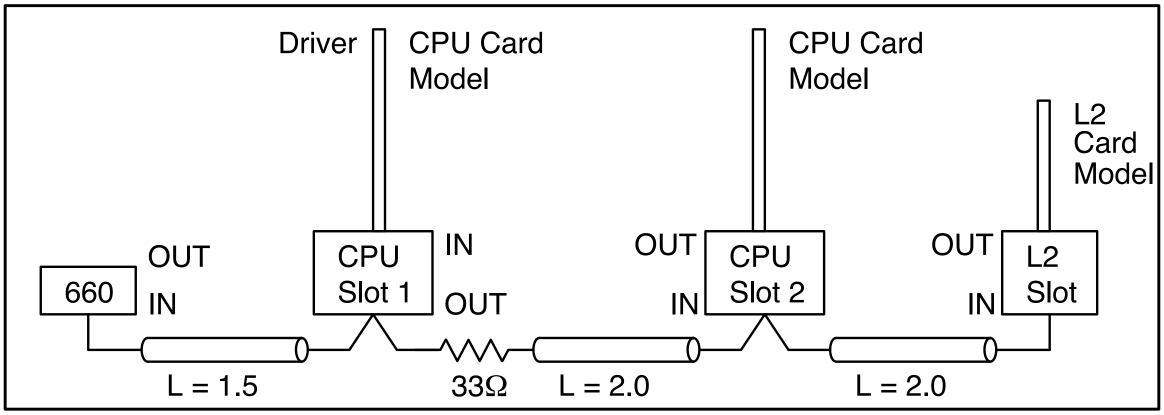


Figure 20. Address Path

If a particular card is not installed, the electrical model of that slot is still to be used. Connect the IN node of the slot model to the motherboard net, and leave the OUT node unconnected.

Section 3
Endian Mode Considerations

Data represented in memory or media storage is said to be in big endian (BE) order when the most significant byte is stored at the lowest numbered address, and less significant bytes are at successively higher numbered addresses.

Data is stored in little endian (LE) order when it is stored with the order of bytes reversed from that of BE order. In other words, the most significant byte is stored at the highest numbered address. The endian ordering of data never extends past an 8-byte group of storage.

The reference design normally operates with big endian (BE) byte significance, which is the native mode of the PowerPC 604 CPU. Internally, the CPU always operates with big endian addresses, data, and instructions, which is ideal for operating systems such as AIX™, which store data in memory and on media in big endian byte significance. In BE mode, neither the CPU nor the 660 Bridge perform address or data byte lane manipulations that are due to the endian mode. Addresses and data pass 'straight through' the CPU bus interface and the 660 Bridge.

The CPU also features a mode of operation designed to efficiently process code and operating systems such as WindowsNT™, which store data in memory and on media in LE byte significance. The reference design also supports this mode of operation.

When the reference design is in little endian mode, data is stored in memory with LE ordering. The 660 Bridge has hardware to select the proper bytes in the memory and on the PCI bus (via address transforms), and to steer the data to the correct CPU data lane (via a data byte lane swapper). Also, see the *604 CPU* and *660 Bridge User's Manuals*.

Table 18 summarizes the operation of the reference design in the two different modes.

Table 18. Endian Mode Operations

Mode	What the 604 Does	What the 660 Bridge Does
Big Endian (BE)	No munge, no shift	No unmunge, no swap
Little Endian (LE)	Address Munged & Data Shifted	Address Unmunged & Data Swapped

In BE mode, the CPU emits the address unchanged, and does not shift the data. This is the native mode of the 604 CPU. In BE mode, the 660 Bridge passes the address and data through to the target without any changes (that are due to endian mode).

In LE mode, the CPU transforms (munges) the three least significant address bits, and shifts the data on the byte lanes to match the munged address. In LE mode, the 660 Bridge unmunges the address and swaps the data on the byte lanes.

3.1 What the 604 CPU Does

3.1.1 The 604 Address Munge

The 604 CPU assumes that the significance of memory is BE. When it operates in LE mode, it internally generates the same effective address as the LE code would generate. Since it assumes that the memory is stored with BE significance, it transforms (munges) the three low order addresses when it activates the address pins. For example, in the 1-byte transfer case, address 7 is munged to 0, 6 to 1, 5 to 2, and so on. Table 19 shows the address transform rules for the allowed LE mode transfer sizes.

Table 19. 604 LE Mode Address Transform

Transfer Size	Address Transform
8	None
4	Physical Address[29:31] XOR 100 => A[29:31]
2	Physical Address[29:31] XOR 110 => A[29:31]
1	Physical Address[29:31] XOR 111 => A[29:31]

3.1.2 The 604 Data Shift

The data transfer occurs on the byte lanes identified by the address pins and transfer size (TSIZ) pins in either BE or LE mode. In LE mode, the CPU shifts the data from the byte lanes pointed to by the unmunged address, over to the byte lanes pointed to by the munged address. This shift is linear in that it does not rotate or alter the order of the bytes, which are now in the proper set of byte lanes. Note that the individual bytes are still in BE order.

3.2 What the 660 Bridge Does

While the reference design is operating properly, data is stored in system memory in the same endian mode as the mode in which the CPU operates. That is, the byte significance in memory is BE in BE mode and it is LE in LE mode. Because of this, hardware is included in the 660 Bridge that (in LE mode) will swap the data bytes to the correct byte lanes, and that will transform (or un-munge) the address coming from the 604.

3.2.1 The 660 Bridge Address Unmunge

In LE mode, the 660 Bridge unmunges address lines A[29:31]. This unmunge merely applies the same XOR transformation to the three low-order address lines as did the CPU. This effectively reverses the effect of the munge that occurs within the CPU. For example, if the CPU executes a one-byte load coded to access byte 0 of memory in LE mode, it will munge its internal address and emit address A[29:31] = 7h. The 660 Bridge will then unmunge the 7 on A[29:31] back to 0, and use this address to access memory.

3.2.2 The 660 Bridge Data Swapper

The 660 Bridge contains a byte swapper. As shown in Figure 21, the byte swapper is placed between the CPU data bus and the memory and PCI data busses. This allows the byte lanes to be swapped between the CPU bus and the PCI bus, or between the CPU bus and memory, but not between the PCI bus and memory. Thus, when a PCI busmaster accesses memory, the reference design does not change either the address or the data location to adjust for endian mode. In either mode, data is stored or fetched from memory at the address presented on the PCI bus.

The 660 Bridge cannot tell the endian mode of the CPU directly, and so cannot automatically change endian mode to match the CPU. There is a control bit located in ISA I/O space (port 0092) that the CPU can write to in order to set the endian mode of the motherboard.

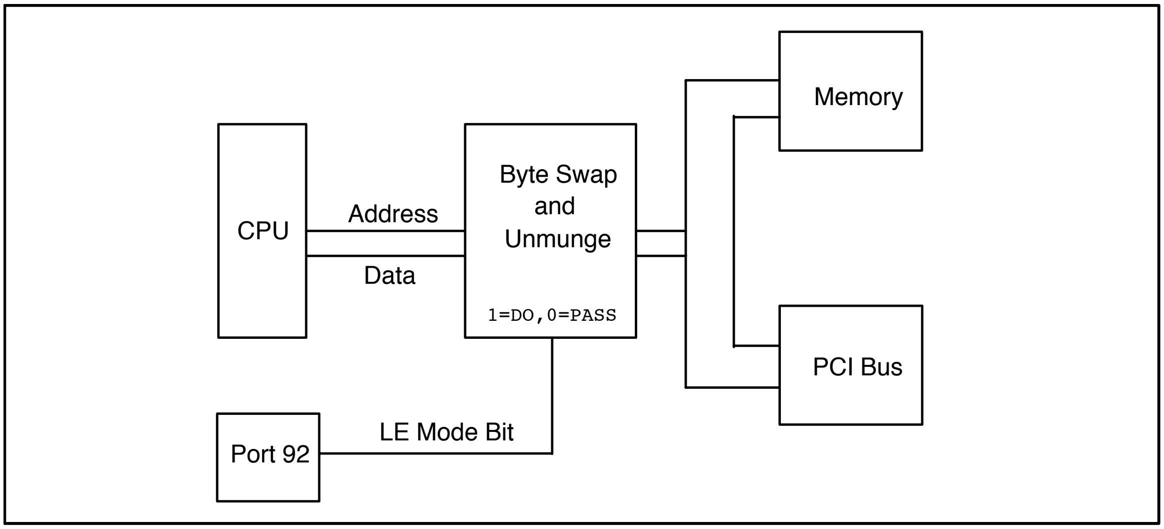


Figure 21. Endian Mode Block Diagram

In BE mode, the 660 Bridge byte swapper is off, and data passes through it with no changes. In LE mode, the byte swapper is on, and the order of the byte lanes is rotated (swapped) about the center. As shown in Table 20, the data on CPU byte lane 0 is steered to memory byte lane 7, the data on CPU byte lane 1 is steered to memory byte lane 6, and so on. During reads, the data flows in the opposite direction over the same paths.

Table 20. 660 Bridge Endian Mode Byte Lane Steering

CPU Byte Lane	BE Mode Connection	LE Mode Connection
CPU byte lane 0 (MSB)	Memory byte lane 0, PCI lane 0	Memory byte lane 7, PCI lane 7*
CPU byte lane 1	Memory byte lane 1, PCI lane 1	Memory Byte lane 6, PCI lane 6*
CPU byte lane 2	Memory byte lane 2, PCI lane 2	Memory byte lane 5, PCI lane 5*
CPU byte lane 3	Memory byte lane 3, PCI lane 3	Memory byte lane 4, PCI lane 4*
CPU byte lane 4	Memory byte lane 4, PCI lane 4*	Memory byte lane 3, PCI lane 3
CPU byte lane 5	Memory byte lane 5, PCI lane 5*	Memory byte lane 2, PCI lane 2
CPU byte lane 6	Memory byte lane 6, PCI lane 6*	Memory byte lane 1, PCI lane 1
CPU byte lane 7 (LSB)	Memory byte lane 7, PCI lane 7*	Memory byte lane 0, PCI lane 0

Note: * In this table, PCI byte lanes 3:0 refer to the data bytes associated with PCI_C/BE[3:0]# when the third least significant bit of the target PCI address (PCI_AD[29]) is 0, as coded in the instruction. PCI byte lanes [7:4] refer to the data bytes associated with PCI_C/BE[3:0]# when PCI_AD[29] is a 1.

3.3 Bit Ordering Within Bytes

The LE convention of numbering bits is followed for the memory and PCI busses, and the CPU busses are labeled in BE nomenclature. The various busses are connected to the 660 Bridge with their (traditional) native significance maintained (BE for CPU, and LE for PCI and memory), so that MSb connects to MSb and so on. The bit paths between the CPU and memory data busses are shown in Table 21 for both BE and LE mode operation.

Table 21. 660 Bit Transfer

CPU_DATA[ ]	BE Mode MEM_DATA[ ]	LE Mode MEM_DATA[ ]
0	7	63
1	6	62
2	5	61
3	4	60
4	3	59
5	2	58
6	1	57
7	0	56
8	15	55
9	14	54
10	13	53
11	12	52
12	11	51
13	10	50
14	9	49
15	8	48
16	23	47
17	22	46
18	21	45
19	20	44
20	19	43
21	18	42
22	17	41
23	16	40
24	31	39
25	30	38
26	29	37
27	28	36
28	27	35
29	26	34
30	25	33
31	24	32
32	39	31
33	38	30
34	37	29
35	36	28
36	35	27
37	34	26
38	33	25
39	32	24
40	47	23
41	46	22
42	45	21
43	44	20
44	43	19
45	42	18
46	41	17
47	40	16
48	55	15
49	54	14
50	53	13
51	52	12
52	51	11
53	50	10
54	49	9
55	48	8
56	63	7
57	62	6
58	61	5
59	60	4
60	59	3
61	58	2
62	57	1
63	56	0

3.4 Byte Swap Instructions

The Power PC architecture defines both word and halfword load/store instructions that have byte swapping capability. Programmers will find these instructions valuable for dealing with the BE nature of this architecture. For example, if a 32-bit configuration register of a typical LE PCI device is read in BE mode, the bytes will appear out of order unless the ”load word with byte swap” instruction is used. The byte swap instructions are:

- lhbrx (load half word byte-reverse indexed)
- lwbrx (load word byte-reverse indexed)
- sthbrx (store half word byte-reverse indexed)
- stwbrx (store word byte-reverse indexed)

The byte-reverse instructions should be used in BE mode to access LE devices and in LE mode to access BE devices.

3.5 604 CPU Alignment Exceptions In LE Mode

The CPU does not support a number of instructions and data alignments in the LE mode that it supports in BE mode. When it encounters an unsupportable situation, it takes an internal alignment exception (machine check) and does not produce an external bus cycle. See the latest 604 CPU documentation for details. Examples include:

- LMW instruction
- STMW instruction
- Move assist instructions (LSWI, LSWX, STSWI, STWX)
- Unaligned loads and stores.

3.6 Single-Byte Transfers

Figure 22 is an example of byte write data a at address xxxx xxx0.

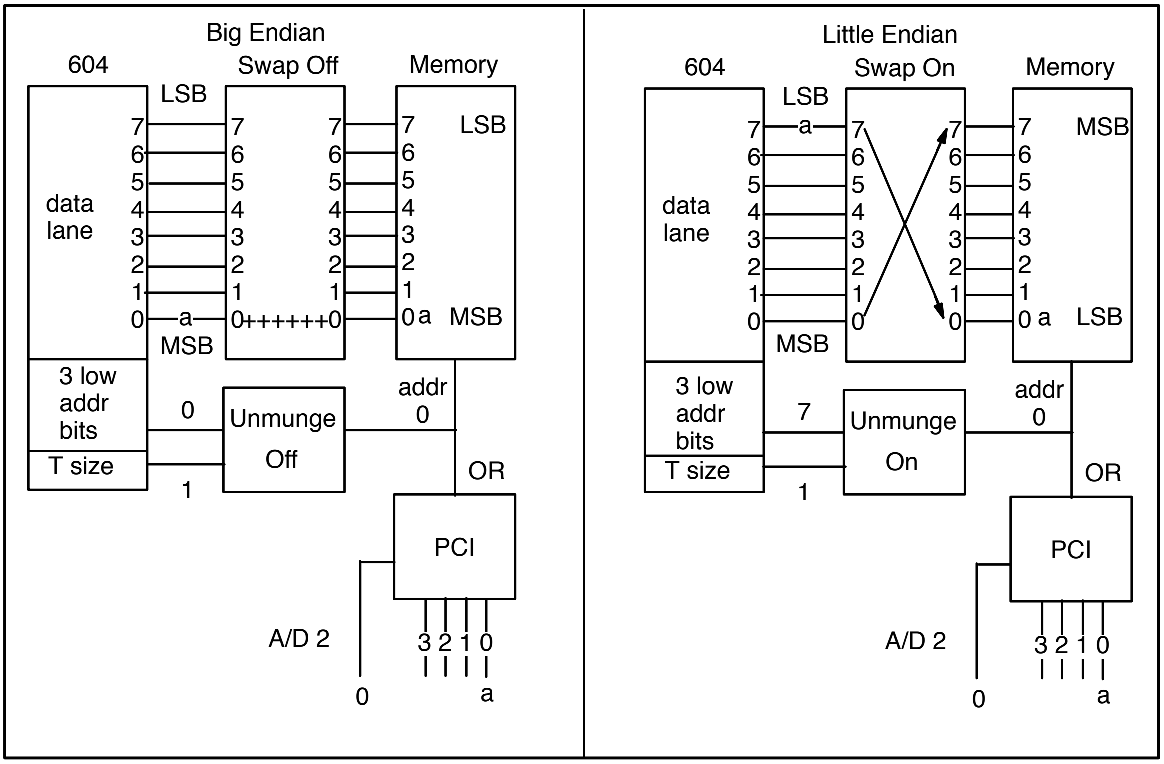


Figure 22. Example at Address xxxx xxx0

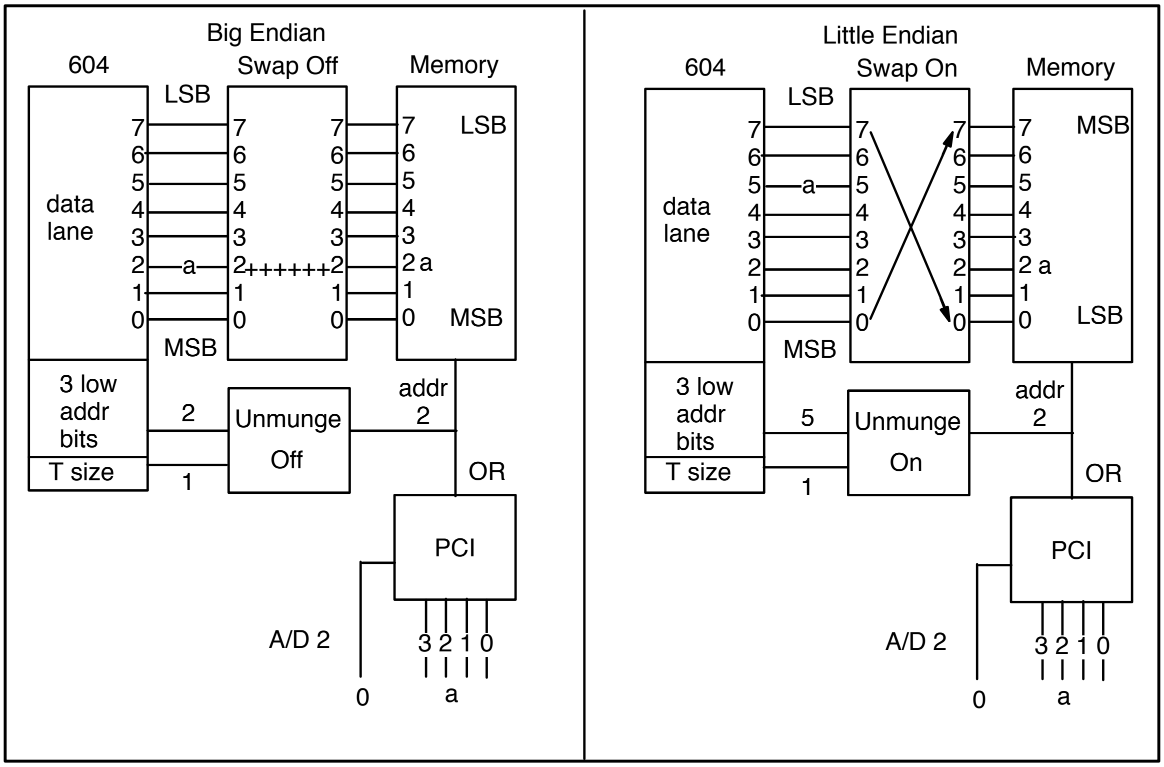


Figure 23. Example at Address xxxx xxx2

Figure 23 is an example of byte write data a at address xxxx xxx2.

For single byte accesses to memory in BE mode, Table 22 applies.

Table 22. Memory in BE Mode

604			604	BYTE		BYTE	MEM BYTE	CAS
A31	30	29	add	LANE		LANE*	LANE	ACTIVE
0	0	0	0	0	MSB	0	0	0
1	0	0	1	1		1	1	1
0	1	0	2	2		2	2	2
1	1	0	3	3		3	3	3
0	0	1	4	4		4	4	4
1	0	1	5	5		5	5	5
0	1	1	6	6		6	6	6
1	1	1	7	7	LSB	7	7	7
NOT MUNGED							SWAP OFF	NOT UNMUNGED

Note:
*At the CPU side.

For single byte accesses to memory in LE mode, Table 23 applies.

Table 23. Memory in LE Mode

					663		
604			604	BYTE	BYTE	MEM BYTE	CAS
A31	30	29	add	LANE	LANE*	LANE	ACTIVE
0	0	0	0	0 MSB	0	7	7
1	0	0	1	1	1	6	6
0	1	0	2	2	2	5	5
1	1	0	3	3	3	4	4
0	0	1	4	4	4	3	3
1	0	1	5	5	5	2	2
0	1	1	6	6	6	1	1
1	1	1	7	7 LSB	7	0	0
MUNGED						SWAP ON	UNMUNGED

Note:
*At the CPU side.

For single byte accesses to PCI in BE mode, Table 24 applies.

Table 24. PCI in BE Mode

604			604	BYTE	BYTE	PCI BYTE	A/D**	BE#
A31	30	29	add	LANE	LANE	LANE	2 1 0	3 2 1 0
							(0=active byte enable)	
0	0	0	0	0 MSB	0	0	0 0 0	1 1 1 0
1	0	0	1	1	1	1	0 0 1	1 1 0 1
0	1	0	2	2	2	2	0 1 0	1 0 1 1
1	1	0	3	3	3	3	0 1 1	0 1 1 1
0	0	1	4	4	4	0	1 0 0	1 1 1 0
1	0	1	5	5	5	1	1 0 1	1 1 0 1
0	1	1	6	6	6	2	1 1 0	1 0 1 1
1	1	1	7	7 LSB	7	3	1 1 1	0 1 1 1
NOT MUNGED						SWAP OFF	NOT UNMUNGED	

Note:
**AD[0:1] set to 00 for all PCI transactions except I/O cycles.

For single byte accesses to PCI in LE mode, Table 25 applies.

Table 25. PCI in LE Mode

			663*			
604	604	BYTE	BYTE	PCI BYTE	A/D **	BE#
A31 30 29	add	LANE	LANE	LANE	2 1 0	3 2 1 0
						(0=active byte enable)
0 0 0	0	0 MSB	0	3	1 1 1	0 1 1 1
1 0 0	1	1	1	2	1 1 0	1 0 1 1
0 1 0	2	2	2	1	1 0 1	1 1 0 1
1 1 0	3	3	3	0	1 0 0	1 1 1 0
0 0 1	4	4	4	3	0 1 1	0 1 1 1
1 0 1	5	5	5	2	0 1 0	1 0 1 1
0 1 1	6	6	6	1	0 0 1	1 1 0 1
1 1 1	7	7 LSB	7	0	0 0 0	1 1 1 0
MUNGED				SWAP ON	UNMUNGED	

Notes:
*At the CPU side.
**AD[0:1] set to 00 for all PCI transactions except I/O cycles.

3.7 Two-Byte Transfers

Figure 24 gives an example of double byte write data ab at address xxxx xxx0.

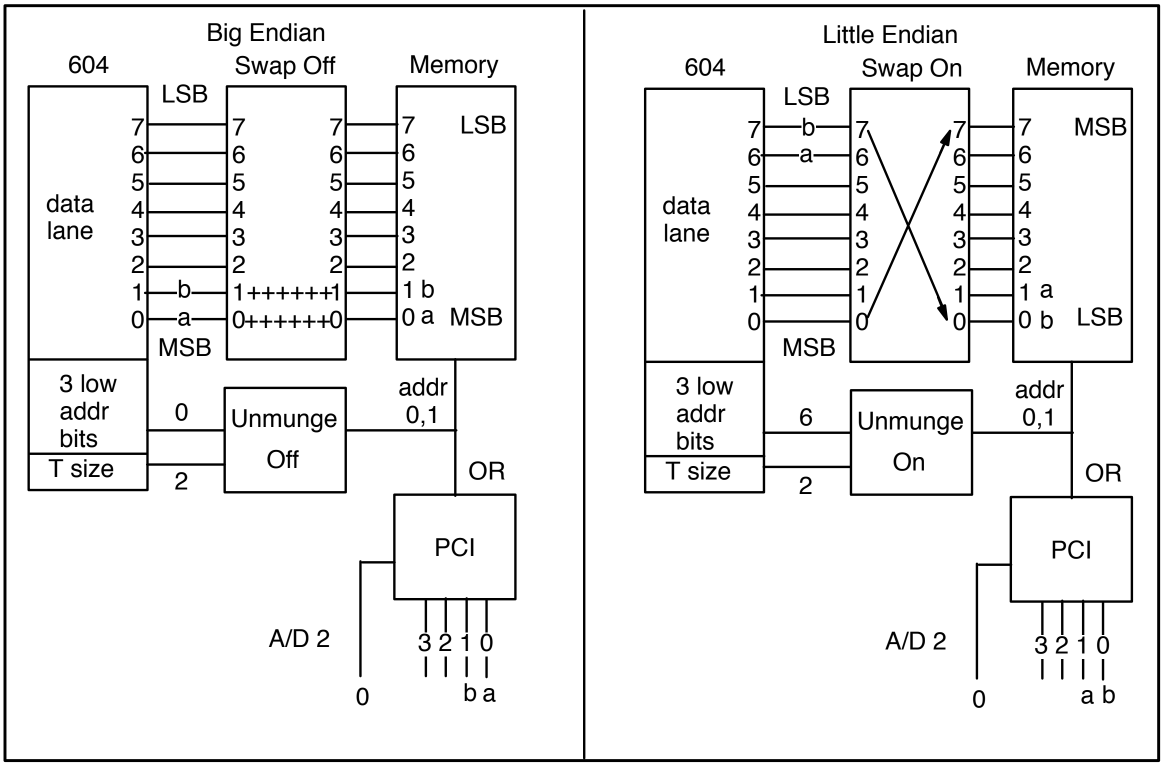


Figure 24. Double Byte Write Data ab at Address xxxx xxx0

Table 26 and Table 27 illustrate all cases that can occur. The columns of Table 26 have these meanings:

- The first column indicates target address (e.g. the address of the byte coded into a store half-word instruction).
- The next two columns show the state of the address pins for BE mode.
- The next two columns show the state of the address pins for the same target data when the machine is in LE mode.
- The remaining columns show the CASs and the PCI byte enables associated with the target data.
- The notes indicate which combinations either do not occur at the pins because of internal exceptions, or are not supported externally.

For 2-byte transfers, Table 26 holds:

Table 26. Two Byte Transfer Information

PROG	BE MODE		LE MODE		BE OR LE	BE OR LE		BE OR LE	
TARG	604	BE	(x or w 110)		Target	CAS# 0:7		PCI CBE#	
ADDR	add	a29:31	Add	a29:31	bytes	0	7	AD2	3210
0	0	000	6	110	0—1	0011	1111	0	1100
1	1	001	7	E 111	1—2	E 1001	1111	0	E 1001
2	2	010	4	100	2—3	1100	1111	0	0011
3	3	011	5	E 101	3—4	E 1110	0111	1	E PPPP
4	4	100	2	010	4—5	1111	0011	1	1100
5	5	101	3	E 011	5—6	E 1111	1001	1	E 1001
6	6	110	0	000	6—7	1111	1100	1	0011
7	N	NNN	1	E 001	NNN	E NNNN	NNNN	N	E NNNN

Notes:
N= not emitted by 60X because it crosses 8 bytes (transforms to 2 singles in BE, machine CH in LE)
P= not allowed on PCI (crosses 4 bytes)
E= causes exception (does not come out on 604 bus) in LE mode

Table 27 contains the same information as found in Table 26, but it is arranged to show the CAS and PCI byte enables that activate as a function of the address presented at the pins of the 604 and as a function of BE/LE mode.

Table 27. Rearranged 2-Byte Transfer Information

2 BYTE XFERS		BE		BE		LE		LE	
60X ADDRESS PINS		CAS#0:7		PCI	CBE#	CAS#0:7		PCI	CBE#
		0	7	A2	3210	0	7	AD2	3210
0	000	0011	1111	0	1100	1111	1100	1	0011
1	001	1001	1111	0	1001	E NNNN	NNNN	N	E NNNN
2	010	1100	1111	0	0011	1111	0011	1	1100
3	011	1110	0111	0	PPPP	E 1111	1001	1	E 1001
4	100	1111	0011	1	1100	1100	1111	0	0011
5	101	1111	1001	1	1001	E 1110	0111E	0	E PPPP
6	110	1111	1100	1	0011	0011	1111	0	1100
7	111	NNNN	NNNN	N	NNNN	E 1001	1111E	0	E 1001

Notes:
N= not emitted by 60X because it crosses 8 bytes (transforms to 2 singles in BE, machine CH in LE)
P= not allowed on PCI (crosses 4 bytes)
E= causes exception (does not come out on 604 bus) in LE mode

3.8 Four-Byte Transfers

Figure 25 gives an example of Word (4-BYTE) Write of 0a0b0c0dh AT ADDRESS xxxx xxx4.

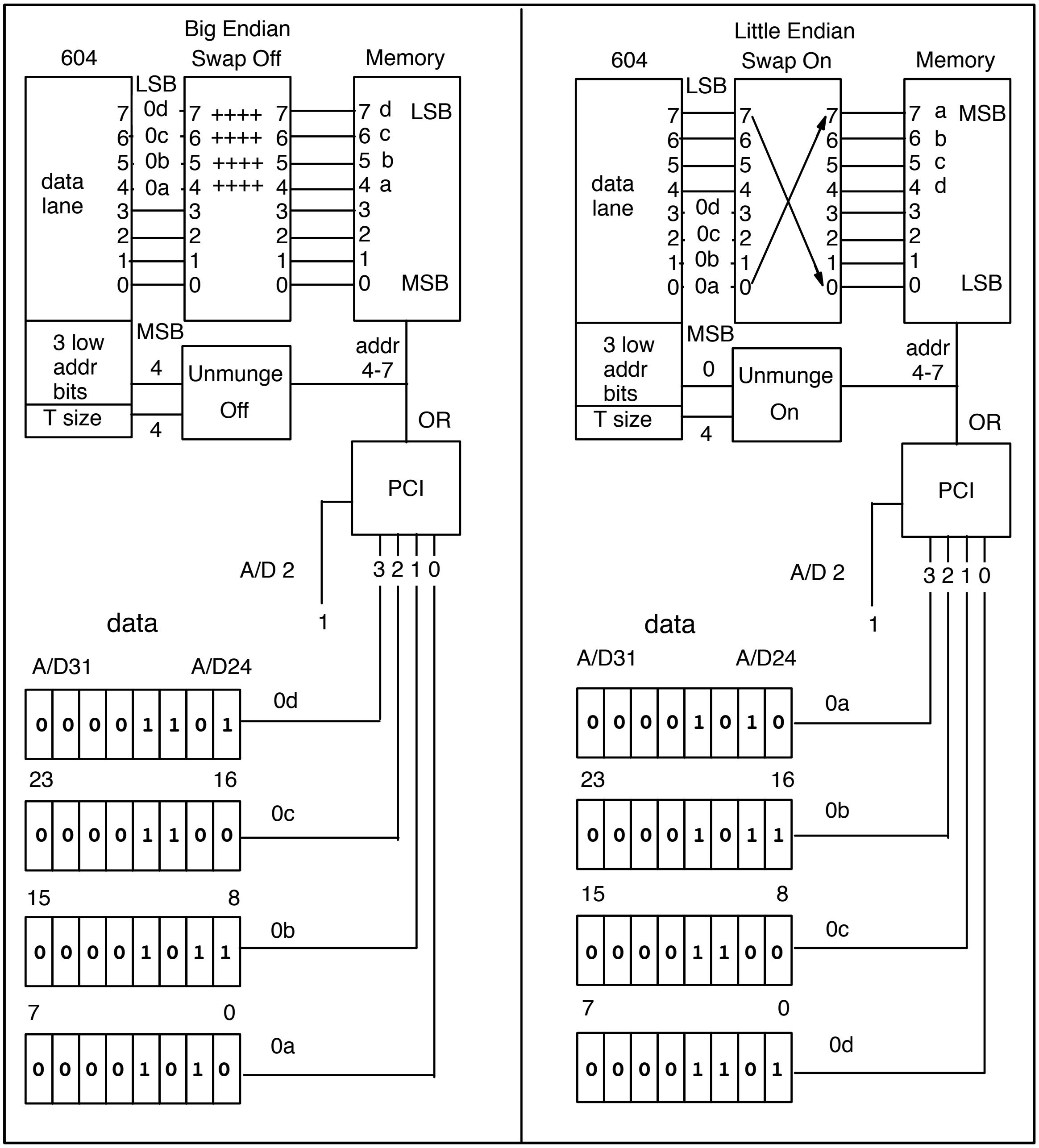


Figure 25. Word (4-Byte) Write of 0a0b0c0dh at Address xxxx xxx4

Table 28 and Table 29 illustrate the cases that can occur. The columns of Table 28 have these meanings:

- The first column indicates the target address (e.g. the address of the byte coded into a store word instruction).
- The next two columns show the state of the address pins for BE mode.
- The next two columns show the state of the address pins for the same target data when the machine is in LE mode.
- The remaining columns show the CASs and the PCI byte enables associated with the target data.
- The notes indicate which combinations either do not occur at the 604 pins because of internal exceptions, or are not supported externally.

Table 28. 4-Byte Transfer Information

PROG	BE MODE		LE MODE		BE OR LE	BE OR LE	BE OR LE
TARG	604 BE		(x or w 100)		Target	CAS# 0:7	PCI CBE#
ADDR	add	a29:31	add	a29:31	bytes	0 7	AD2 3210
0	0	000	4	100	0–3	0000 1111	0 0000
1	1	001	5	E 101	1–4	E 1000 0111	0 E PPPP
2	2	010	6	E 110	2–5	E 1100 0011	0 E PPPP
3	3	011	7	E 111	3–6	E 1110 0001	1 E PPPP
4	4	100	0	000	4–7	1111 0000	1 0000
5	5	NNN	1	E NNN	N–N	NNNN NNNN	1 E NNNN
6	6	NNN	2	E NNN	N–N	NNNN NNNN	1 E NNNN
7	7	NNN	3	E NNN	N–N	NNNN NNNN	1 E NNNN

Notes:
N= not emitted by 60X because it crosses 8 bytes (transformed into 2 bus cycles)
P= not allowed on PCI (crosses 4 bytes)
E= causes exception (does not come out on 604 bus) in LE mode

Table 29 contains the same information as found in Table 28, but it is arranged to show the CAS and PCI byte enables that activate as a function of the address presented at the pins of the 604 and as a function of BE/LE mode.

Rearranging Table 29 for 4-byte transfers:

Table 29. Rearranged 4-Byte Transfer Information

4 BYTE XFERS		BE	BE		LE	LE	
60X ADDRESS PINS		CAS#0:7	PCI	CBE#	CAS#0:7	PCI	CBE#
		0 7	A2	3210	0 7	AD2	3210
0	000	0000 1111	0	0000	1111 0000	0	0000
1	001	1000 0111	0	PPPP	E NNNN NNNN	O	E NNNN
2	010	1100 0011	0	PPPP	E NNNN NNNN	O	E NNNN
3	011	1110 0001	0	PPPP	E NNNN NNNN		E NNNN
4	100	1111 0000	1	0000	0000 1111	1	0000
5	101	NNNN NNNN	1	NNNN	E 1000 0111	1	E PPPP
6	110	NNNN NNNN	1	NNNN	E 1100 0011	1	E PPPP
7	111	NNNN NNNN	1	NNNN	E 1110 0001	1	E PPPP

Notes:
N= not emitted by 60X because it crosses 8 bytes (transformed into 2 bus cycles)
P= not allowed on PCI (crosses 4 bytes)
E= causes exception (does not come out on 604 bus) in LE mode
X= not supported in memory controller (crosses 4-byte boundary

3.9 Three byte Transfers

There are no explicit Load/Store three-byte instructions; however, three-byte transfers occur as a result of unaligned four-byte loads and stores as well as a result of move multiple and string instructions.

The TSIZ=3 transfers with address pins = 0, 1, 2, 3, 4, or 5 may occur in BE. All of the other TSIZ and address combinations produced by move multiple and string operations are the same as those produced by aligned or unaligned word and half-word loads and stores.

Since move multiples, strings, and unaligned transfers cause machine checks in LE mode, they are not of concern in the BE design.

3.10 Instruction Fetches and Endian Modes

Most instruction fetching is with cache on. Therefore memory is fetched eight bytes wide. Figure 26 shows the instruction alignment.

Example: 8 byte instruction fetch I1=abcd, I2=efgh at address xxxx xxx0

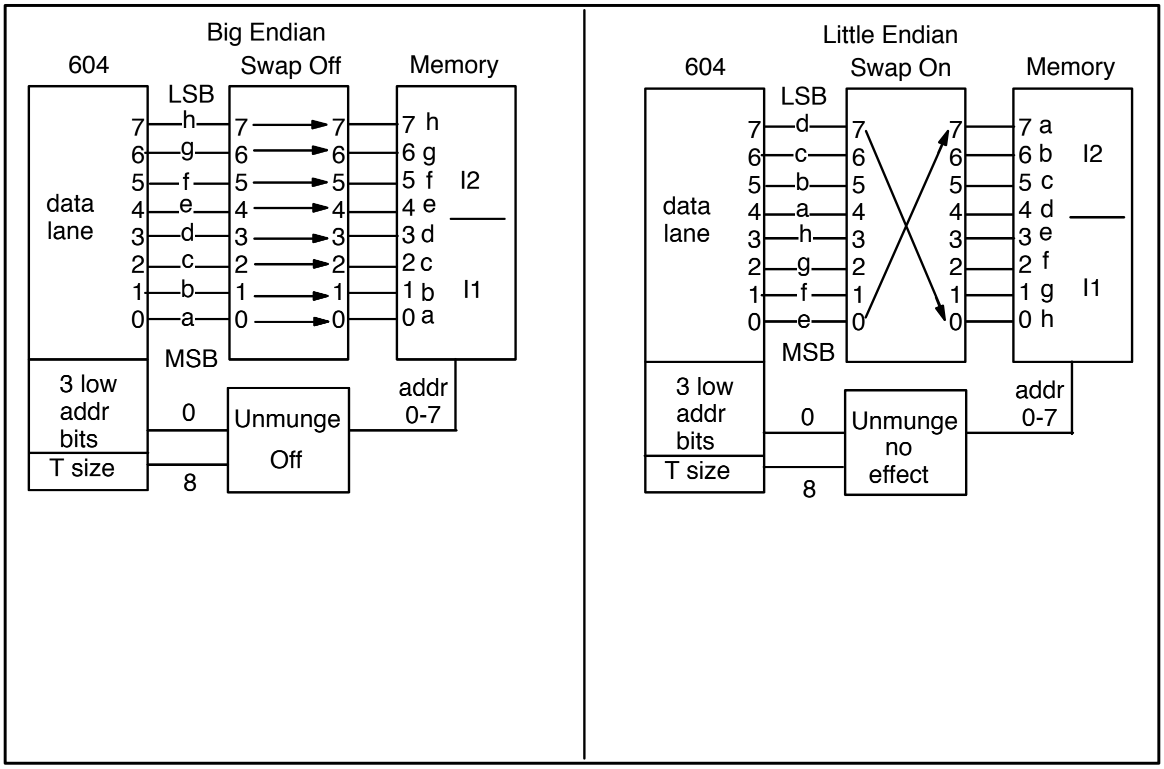


Figure 26. Instruction Alignment Example

It is possible to fetch instructions with 4 byte aligned transfers when the cache is turned off. In that case, the 604 does not munge the address in LE mode. The memory controller does not differentiate between instruction and data fetches, but the unmunger is ineffective because the memory is always read 8 byte wide, and data is presented on all 8 byte lanes. If the unmunger were used, the wrong instruction would be read. The net result is illustrated in Figure 27.

Example: 4 byte instruction fetch, I2=efgh at address xxxx xxx4

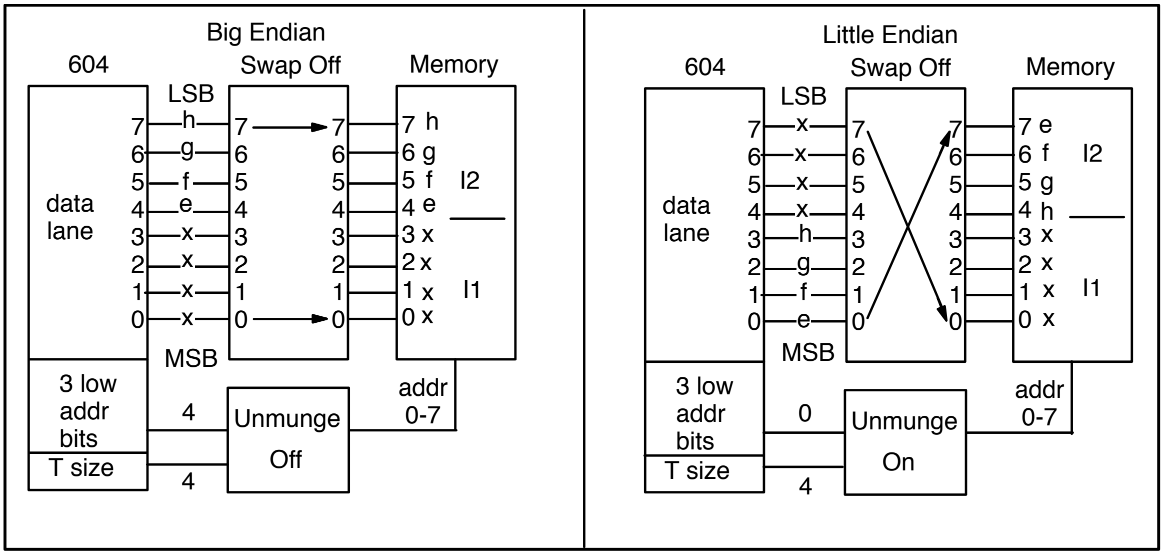


Figure 27. Wrong Instruction Read When Unmunger is used

3.11 Changing BE/LE Mode

There are two BE/LE mode controls. One is inside the 604 CPU and the other is a register bit on the motherboard. The 604 CPU interior mode is not visible to the motherboard hardware. The BE mode bit referred to in this document is the register bit on the motherboard. It is a bit in I/O space which is memory mapped just like other I/O registers. It defaults to BE mode.

The 604 CPU always powers up in the BE mode and begins fetching to fill its cache. Consequently, at least the first of the ROM code must be BE code. It is beyond the scope of this document to define how the system will know to switch to LE mode; however, great care must be made during the switch in order to synchronize the internal and external mode bits, to flush all caches, and to avoid executing extraneous code.

The following process switches the system from BE to LE mode when used in this system:

1. Disable L1 caching.
2. Disable L2 caching.
3. Flush all system caches.
4. Turn off interrupts immediately after a timer tick so no timer interrupts will occur during the next set of cycles.
5. Mask all interrupts.

6. Set the CPU state and the motherboard to LE (see Figure 28). Note that CPU is now in LE mode. All instructions must be in LE order.
7. Put interrupt handlers and CPU data structures in LE format.
8. Enable caches.
9. Enable Interrupts.
10. Start the LE operating system initialization.

Figure 28 shows the instruction stream to switch endian modes.

```
        x       mfspr   R2,1008         ;Load the HDO register
;Instructions to set the Little-Endian bit in R2
        0       sync
        4       sync
        8       sync
        C       mtspr   1008,R2         ;Moves to HID0 register
        10      sync
        14      sync
        18      sync
        1c      sync
        20      Store to external Endian control port (X8000 0092)
;The above instruction must be on a double word boundary
;So the following instruction is executed first (due to pipeline)
        24      eieio
; To this point all instructions are in Big Endian format
; The following instructions look the same in either Endian mode
        28      X38010138
        2C      X38010138
        ...     ;Enough of these instructions must be executed
        ...     ;to guarantee the above store has occurred.
;
;before any memory or I/O cycles are listed.
        xx      X38010138
```

Figure 28. Instruction Stream to Switch Endian Modes

3.12 Summary of Bi-Endian Operation and Notes

- When the 604 CPU is in BE mode, the memory is in BE mode, and data flowing on the PCI is in BE order so that it is recorded on the media in BE order. Byte 0 is the most significant byte.
- When the 604 CPU is in LE mode, the memory is in LE mode, and data flowing on the PCI is in LE order so that it is recorded on the media in LE order. Byte 0 is the least significant byte.
- The PCI bus is addressed in the same manner that memory is when the 604 CPU runs a cycle. The unmunging in LE mode changes the effective low-order address bits (the byte enables and A/D 2). On all but I/O cycles, the two low-order A/D lines are set to zero. On PCI I/O cycles, A/D 1,0 are also transformed by the unmunge operation
- No translations are made when PCI accesses memory so that the byte with address 0 on the PCI flows to byte 0 in memory — 1 to 1, 2 to 2, and so on. For example, if BE0# and BE1# are active and A/D 2 is a 0, then memory byte lanes 0 and 1 are addressed (cas 0 and cas 1 active on writes).
- Note that the LE devices which interpret data structures in the memory require that their control data be arranged in LE order even in BE mode. For example, SCSI scripts in memory must always be arranged in LE order because that is what the device expects.
- Devices such as video may require the bytes to be swapped unless these devices have byte swap capability.

Section 4
CPU Card

The IBM PowerPC 604 CPU Card (Cheetah0) is a PowerPC 604 CPU card which is designed to provide a CPU card for the 604 SMP reference design motherboard (see Figure 29). One Cheetah0 can be used for a uniprocessor system and two cards can be used for a 2-way SMP system. The card is designed meet the electrical, physical, and thermal requirements of the motherboard with the following performance and features:

- Allows up to 66MHz system bus speed.
- Contains one PowerPC 604 CPU with fansink and basic options.
- Allows up to 132MHz 604 CPU operation with the specified power supplies and cooling.
- Contains power supply support for possible future lower voltage CPUs.

Cheetah0 interfaces to the reference design motherboard through the CPU slot connector, which is a 256-pin card edge connector. The Cheetah0 schematics are located in NO TAG.

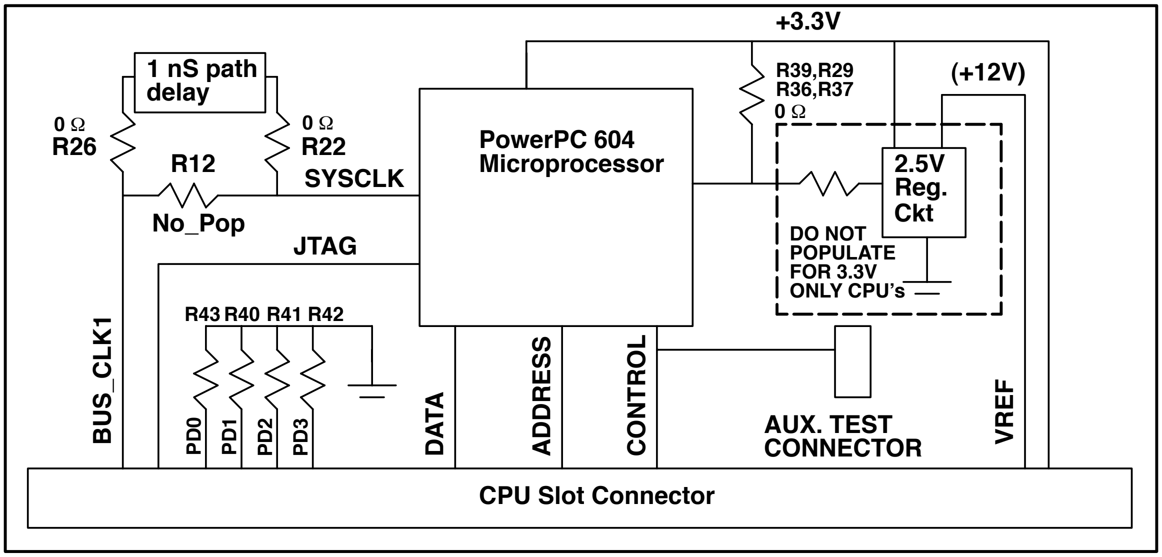


Figure 29. CPU Card Block Diagram

4.1 Major Components

For component part numbers, refer to the Cheetah0 BOM in Section 19.

4.1.1 PowerPC 604 Processor

The CPU card uses the PowerPC 604 processor in a Ball Grid Array package. The 604 processor uses an advanced 3.3v CMOS technology and is fully I/O compatible with TTL devices. The reference design allows the Cheetah0 CPU to run at bus speeds of 60MHz and 66MHz using the current fansink and power supplies.

As shown on the Cheetah0 schematics, most of the 604 I/Os exit Cheetah0 to the reference design motherboard via the CPU slot connector. Exceptions to this are discussed below.

4.1.2 JTAG/RISCWatch Interface

Refer to the 604 User's Manual for information on the 604 RISCWatch/ESP facility. Refer to the Cheetah0 schematics in NO TAG for specifics of the interconnection between the 604 and the RISCWatch connector. The RISCWatch signals exit Cheetah0 via the CPU slot connector.

4.1.3 J2 Auxiliary Test Connector

J2 provides access to some of the signals on the CPU bus interface for debug purposes. IBM makes no recommendation as to the use of this connector. J2 is a 2x7 header on 0.1 inch centers (see Figure 30). Table 30 shows the pinout of J2.

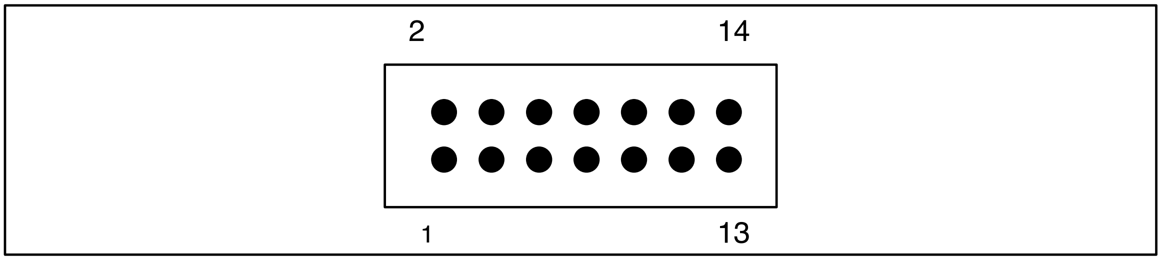


Figure 30. J2 Auxiliary Test Connector

Table 30. J2 Aux Connector Pinout

Pin No.	Signal Name
1	SYS_BR#
2	TC2
3	TC1
4	TC0
5	nc
6	nc
7	RSRV#

Pin No.	Signal Name
8	SRESET_A#
9	nc
10	HRESET_A#
11	INT_A#
12	DPE#
13	APE#
14	GND

4.1.4 Fansink

The 604 processor on Cheetah0 requires a fansink to ensure proper cooling under worst case conditions in the reference system. As shown in Figure 31, the fansink is held to the CPU with a spring clip which is mounted to the Cheetah0 circuit board by a pair of standoffs. A piece of Thermostrate™ thermal tape is used to thermally connect the fansink to the chip package. The tape also allows for vibration and for non-parallelism between the sink and the chip package.

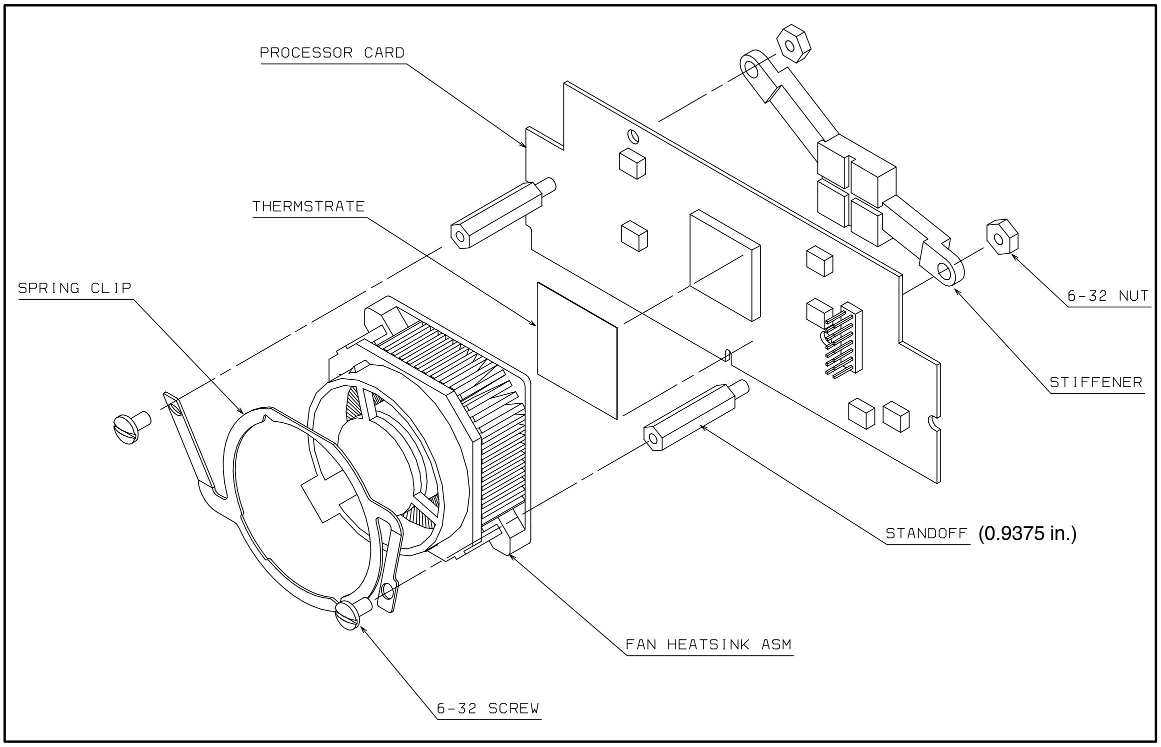


Figure 31. CPU Card Fansink Assembly

The fansink requires +12v, which is provided via a 2 wire cable (see Figure 32) which has a 3 position connector attached. The connector attaches to a 3 pin (0.1 in. centers) header on the motherboard. It is recommended that the motherboard header connect both pin 1 and pin 3 to ground to allow the connector to function properly in either orientation.

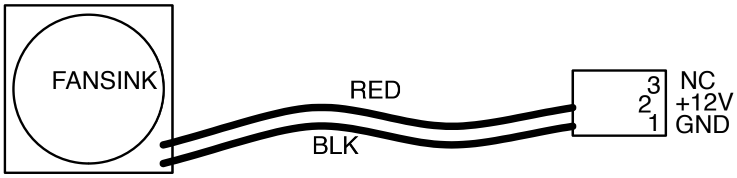


Figure 32. Fansink With Power Cable

The spring clip is manufactured by A.K. Stamping (908) 232 – 7300. Their part number is 83866072.

4.2 Configuration Options

Cheetah0 is equipped with several jumper and pop/no-pop (either populate or not populate the component) options to increase the flexibility of the card.

4.2.1 Bus Clock Skew Circuit

604_SYSCLK, the CPU bus clock supplied by the motherboard, can be configured for minimum clock skew or for a forced 1ns delay. This can sometimes be useful when matching the timing characteristics of a given CPU card to those of the motherboard. Cheetah0 is supplied with the delay activated (0Ω resistors installed for R22 and R26, and R12 not populated). The delay can be taken out of the clock line by removing R22 and R26, and installing a 0Ω resistor in R12.

The delay element is a trace that is about 5 inches long.

4.2.2 2.5v Power Supply

Cheetah0 is designed specifically for the 604 CPU, but provisions have been made for generating a 2.5v power supply from the 3.3v power supply. R29, R36, R37, and R39 are part of this circuit. This circuit is not installed or supported on Cheetah0, and both the circuit and CPU pinout are subject to change without notice. See the hardware specification of the particular CPU for power supply pinout information.

4.2.3 Presence Detect Bits

The presence detect bits PD[0:3] on Cheetah0 tell the motherboard the speed capability of the CPU on the card. The motherboard detects the speed capability of each of the CPU cards in the system. It then sets the operating frequency of the CPU bus clock supplied to each CPU card. The motherboard then sets the value of the PLL configuration bits PLL_CFG[0:3] by driving FREQ_ID[0:3] to the appropriate levels.

PD[0:3] are pulled up by 10kΩ resistors on the motherboard. To assert a logic 0, the CPU card should pull the signal low with a 100 ohm resistor. Cheetah 0 sets PD[0:3] = 0011, indicating that a 132MHz CPU is present and that there is no serial ROM on the CPU card. Cheetah0 uses R40, R41, R42, and R43 to set the value of the PD bits.

4.2.4 DRVMOD Bits

The DRVMOD[0:1] inputs to the 604 control certain of the 604 output driver characteristics. The value of these bits is set by the motherboard using DRVMOD[0:1], which are pulled up on Cheetah0 by a pair of 10kΩ resistors.

4.2.5 Additional Bits

The spare PAL site U2 is not populated or supported.

SYS_DBWO# and APE# are pulled up with a 10k resistor because they are not driven by the motherboard.

LSSD#, L1_TEST_CLK, L2_TEST_CLK, and ARRAY_WR are pulled up with a 10k resistor as recommended by the 604 User's Manual.

L2_INT is pulled down with a 1kΩ resistor as per the 604 User's Manual.

AVDD is connected as shown to increase the noise immunity of the 604 PLL by decoupling the PLL supply.

R46 and R47 are supplied to allow the heatsink to be grounded for EMC purposes.

4.3 Electrical and Thermal Requirements

4.3.1 Absolute Maximum Ratings

Table 31. Voltage Ratings

	Parameter[1]	Min.	Max.	Unit
V_{DD}	3.3v Supply Voltage	–0.3	3.6	v
V_{CC}	5.0v Supply Voltage	–0.3	7.0	v
V_{IN}	Input Voltage	–0.3	5.5	v[2]
T_{JCPU}	Processor Junction Temperature	0	105	°C
T_{STG}	Storage temperature range	–55	150	°C

Notes:

1. Absolute Maximum ratings are stress ratings only, functional operation at the maximums is not guaranteed. Stresses beyond those listed may affect device reliability or cause permanent damage to the devices on the card.
2. PowerPC 604 processor additionally requires that no Input exceed the 3.3v supply voltage by more than 2.5v during power-on reset.

4.3.2 DC Specifications

Table 32. DC Recommended Operating Conditions

	Parameter	Min.	Max.	Unit
V_{IN}	Input High Voltage	2.0	5.5	v[1]
V_{IL}	Input Low Voltage	0.0	0.8	v
CV_{IH}	BUS_CLK Input High Voltage	2.4	5.5	v
CV_{IL}	BUS_CLK Input Low Voltage	0.0	0.4	v
I_{IH}	Input High Current		100	uA
I_{IL}	Input Low Current		100	uA
V_{OH}	Output High Voltage I_{OH} = –18mA	2.4	3.78	v[1]
V_{OL}	Output Low Voltage I_{OL} = 14mA	0.0	0.4	v
$V_{DD5.0}$	5v Power Supply Voltage	4.5	5.5	v
$I_{DD5.0}$	5v Power Supply Current		0	A
$V_{DD3.3}$	V_{DD} Power Supply Voltage	3.0	3.6	v
$I_{DD3.3}$	V_{DD} Power Supply Current		5.6	A
V_{12}	12v Power Supply Voltage	10.8	13.2	v
I_{12}	12v Power Supply Current		.1	A[2]

Notes:

1. All CPU bus receivers on Cheetah0 are 5.0v tolerant.
2. The CPU fansink requires 12v from a connector on the motherboard.

4.3.3 Thermal

Table 33. Thermal Recommended Operating Conditions

	Parameter	Min.	Max.	Unit
T_{JCPU}	Processor Junction Temperature	0	101	°C
	Airflow[1]	150	—	lfpm
T_{AMB}	Ambient Temperature at Leading Edge of Card		34	°C

Notes:

1. This air velocity is required at the pass transistor (if installed) and at the CPU heatsink (or fansink).

4.3.4 AC Timing Requirements

Table 34. AC Timing Recommended Operating Conditions

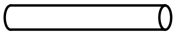
Symbol	Parameter	1nS Skew		No Skew		UNIT
		Min.	Max.	Min.	Max.	
F_{CPU}	CPU max frequency [2]		132		132	MHz
F_{BUS}	BUS_CLK Frequency [2]	50.0	66.6	50.0	66.6	MHz
T_{CYC}	BUS_CLK cycle time at 1.5v	15	20	15	20	nS
Clock	BUS_CLK(n) Duty Cycle	40	60	40	60	%
	BUS_CLK(n) Clock Trace Length	8.65	8.85	3.65	3.85	inches[3]
	BUS_CLK(n) Clock Net Loading		10		10	pF

Notes:

1. All timings are derived from the PowerPC 604 timing specifications for a 66.67MHz bus device with a 2:1 CPU to SYSCLK ratio. Timings are measured at the CPU on Cheetah0 and and are given as relative to the SYSCLK input on the CPU. Where conflicts exist, the *PowerPC 604 Hardware Specification* supersedes this document.
2. CPU and bus frequency are set by the motherboard logic as a function of the capabilities of the CPU cards that are installed in the system. See Section 7.
3. Wire each BUS_CLK(n) point to point between J1 and a single load at the trace length specified. This will match this net to the other clock nets on the CPU bus.
4. **Critical nets** should be wired point to point. These nets are: TS#, TA#, TBST#, TSIZ(0..2), TT(0..4), DPE#, ABB#, AACK#, ARTRY#, DRTRY#, DBB#, DBG#, BG#. BR#, GBL#, SHD#, DP(0..7)#, A(0..31), DL(0..31), DH(0..31).

4.4 Electrical Model of Major Signal Groups

The physical signal paths (traces) of some major signal groups are modeled in Figure 33, where:

-  indicates a trace length that is modeled as a transmission line, where
  Z_o = 70Ω . . . . Characteristic Impedance,
  T_D = 177ps/in . . . . Time Delay, and
  R_S = 159Ω/in . . . . DC Resistance.
  L gives the maximum length of the trace in inches. For critical nets, L must be less than or equal to three inches. The critical nets are: TS#, TA#, TBST#, TSIZ[0:2], TT[0:4], DPE#, ABB#, AACK#, ARTRY#, DRTRY#, DBB#, DBG#, BG#. BR#, GBL#, SHD#, DP[0:7]#, A[0:31], DL[0:31], DH[0:31].

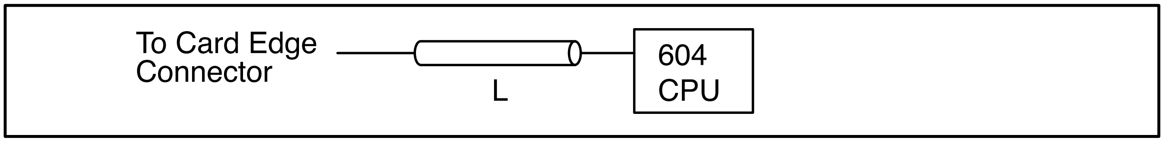


Figure 33. Signal Group Electrical Model

4.5 CPU Card Slot Connector

Cheetah0 connects to the reference design motherboard mainly through the CPU card connector, J1 (see Figure 34). The pinout of J1 is shown in Table 35. The signal descriptions are shown in Section 2, CPU Bus.

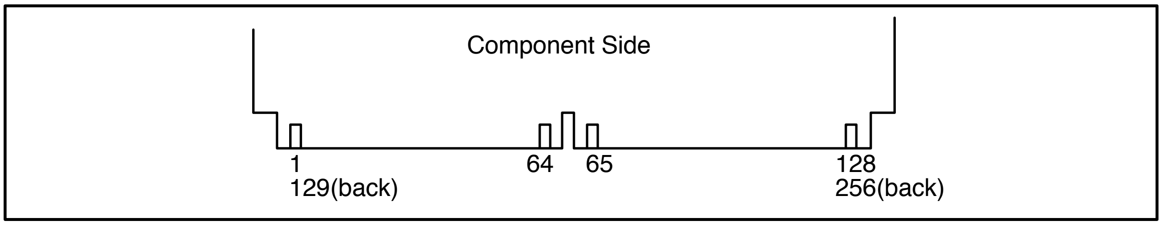


Figure 34. J1 Outline

4.5.1 Pin Definitions

Table 35. CPU Card Pin List

PIN#	SIGNAL NAME	TYPE
1	GND	
2	DRVMOD0	I
3	PD0	O
4	PD2	O
5	GND	
6	SYS_DATA_H[30]	I/O
7	3.6v/3.3v	Unused
8	SYS_DATA_H[28]	I/O
9	SYS_DATA_H[26]	I/O
10	SYS_DATA_H[24]	I/O
11	GND	
12	SYS_DATA_H[22]	I/O
13	SYS_DATA_H[20]	I/O
14	SYS_DATA_H[19]	I/O
15	SYS_DATA_H[17]	I/O
16	SYS_DATA_H[15]	I/O
17	GND	
18	3.6v/3.3v	Unused
19	SYS_DATA_H[12]	I/O
20	SYS_DATA_H[11]	I/O
21	SYS_DATA_H[9]	I/O
22	SYS_DATA_H[7]	I/O
23	GND	
24	SYS_DATA_H[5]	I/O
25	SYS_DATA_H[3]	I/O
26	SYS_DATA_H[2]	I/O
27	SYS_DATA_H[0]	I/O
28	GND	
29	3.3v	
30	3.3v	
31	SYS_DATA_L[28]	I/O
32	SYS_DATA_L[26]	I/O
33	SYS_DATA_L[24]	I/O
34	GND	
35	SYS_DATA_L[22]	I/O
36	3.3v	
37	SYS_DATA_L[20]	I/O
38	SYS_DATA_L[18]	I/O
39	SYS_DATA_L[16]	I/O
40	SYS_DATA_L[14]	I/O
41	GND	
42	3.3v	
43	SYS_DATA_L[12]	I/O
44	SYS_DATA_L[11]	I/O
45	SYS_DATA_L[9]	I/O
46	SYS_DATA_L[7]	I/O
47	GND	
48	3.3v	
49	SYS_DATA_L[4]	I/O
50	SYS_DATA_L[3]	I/O
51	SYS_DATA_L[1]	I/O
52	SYS_DATA_L[0]	I/O
53	SYS_DP5	I/O
54	3.3v	
55	3.3v	
56	SYS_DP2	I/O

PIN#	SIGNAL NAME	TYPE
57	SYS_DP0	I/O
58	GND	
59	BUS_CLK2	Unused
60	GND	
61	SYS_ADDR[29]	I/O
62	SYS_ADDR[28]	I/O
63	SYS_ADDR[26]	I/O
64	SYS_ADDR[25]	I/O
65	SYS_ADDR[23]	I/O
66	SYS_ADDR[21]	I/O
67	GND	
68	DRTRY#	I/O
69	GND	
70	SYS_ADDR[19]	I/O
71	3.3v	
72	SYS_ADDR[17]	I/O
73	GND	
74	SYS_ADDR[13]	I/O
75	SYS_ADDR[12]	I/O
76	SYS_ADDR[11]	I/O
77	SYS_ADDR[9]	I/O
78	GND	
79	SYS_ADDR[7]	I/O
80	SYS_ADDR[5]	I/O
81	SYS_ADDR[3]	I/O
82	3.3v	
83	SYS_ADDR[0]	I/O
84	3.3v	
85	GND	
86	SYS_AACK#	I
87	GND	
88	SYS_DATA_BUS_GR#	I
89	SYS_TA#	I
90	SYS_DBB#	I
91	SYS_TEA#	I
92	+5v	
93	GND	
94	XATS#	I/O
95	SYS_WT#	I/O
96	SYS_TT[1]	I/O
97	SYS_ARTRY#	I/O
98	SYS_SHARED#	I/O
99	GND	
100	+5v	I/O

PIN#	SIGNAL NAME	TYPE
101	SYS_ABB#	I/O
102	+5v	
103	SYS_TT[2]	I/O
104	GND	
105	TCK	I
106	TDI	I
107	SRESET_A#	I
108	L2_AACK_EN#	Unused
109	CONFIG#	Unused
110	GND	
111	TRST#	I
112	+5v	
113	HALT_A/QREQ_A	O
114	L2_WT#	Unused
115	RUN/QAK	I
116	GND	
117	+5v	
118	SYS_ADDRP[0]	I/O
119	GND	
120	SYS_ADDRP[1]	I/O
121	VREF	
122	L2_BR#	Unused
123	L2_CLAIM#	Unused
124	L2_CLR#	Unused
125	PWR_DWN	I
126	FREQ_ID0	I
127	FREQ_ID2	I
128	GND	
129	DRVMOD1	I
130	GND	
131	PD1	O
132	PD3	O
133	SYS_DATA_H[31]	I/O
134	SYS_DATA_H[29]	I/O
135	GND	
136	SYS_DATA_H[27]	I/O
137	SYS_DATA_H[25]	I/O
138	SYS_DATA_H[23]	I/O
139	3.6v/3.3v	Unused
140	SYS_DATA_H[21]	I/O
141	GND	
142	SYS_DATA_H[18]	I/O
143	SYS_DATA_H[16]	I/O
144	SYS_DATA_H[14]	I/O

PIN#	SIGNAL NAME	TYPE
145	SYS_DATA_H[13]	I/O
146	3.6v/3.3v	Unused
147	GND	
148	SYS_DATA_H[10]	I/O
149	SYS_DATA_H[8]	I/O
150	SYS_DATA_H[6]	I/O
151	3.3v	
152	SYS_DATA_H[4]	I/O
153	GND	
154	SYS_DATA_H[1]	I/O
155	SYS_DATA_L[31]	I/O
156	SYS_DATA_L[30]	I/O
157	SYS_DATA_L[29]	I/O
158	GND	
159	SYS_DATA_L[27]	I/O
160	SYS_DATA_L[25]	I/O
161	SYS_DATA_L[23]	I/O
162	3.3v	
163	SYS_DATA_L[21]	I/O
164	GND	
165	SYS_DATA_L[19]	I/O
166	SYS_DATA_L[17]	I/O
167	SYS_DATA_L[15]	I/O
168	SYS_DATA_L[13]	I/O
169	3.3v	
170	3.3v	
171	GND	
172	SYS_DATA_L[10]	I/O
173	SYS_DATA_L[8]	I/O
174	SYS_DATA_L[6]	I/O
175	3.3v	
176	SYS_DATA_L[5]	I/O
177	GND	
178	SYS_DATA_L[2]	I/O
179	SYS_DP7	I/O
180	SYS_DP6	I/O
181	SYS_DP4	I/O
182	3.3v	
183	SYS_DP3	I/O
184	SYS_DP1	I/O
185	SYS_ADDR[31]	I/O
186	SYS_ADDR[30]	I/O
187	GND	
188	BUS_CLK0	Unused

PIN#	SIGNAL NAME	TYPE
189	GND	
190	SYS_ADDR[27]	I/O
191	3.3v	
192	SYS_ADDR[24]	I/O
193	SYS_ADDR[22]	I/O
194	GND	
195	BUS_CLK1	I
196	GND	
197	3.3v	
198	SYS_ADDR[20]	I/O
199	SYS_ADDR[18]	I/O
200	SYS_ADDR[16]	I/O
201	SYS_ADDR[15]	I/O
202	SYS_ADDR[14]	I/O
203	GND	
204	3.3v	
205	SYS_ADDR[10]	I/O
206	SYS_ADDR[8]	I/O
207	SYS_ADDR[6]	I/O
208	SYS_ADDR[4]	I/O
209	GND	
210	SYS_ADDR[2]	I/O
211	SYS_ADDR[1]	I/O
212	GND	
213	3.3v	
214	SYS_ADDR_BUS_GR#	I
215	SYS_GBL#	I/O
216	SYS_TSIZ[0]	I/O
217	3.3v	
218	GND	
219	SYS_TSIZ[2]	I/O
220	SYS_TSIZ[1]	I/O
221	SYS_TS#	I/O
222	SYS_TT[3]	I/O
223	TMS	I
224	GND	
225	+5v	
226	SYS_TT[0]	I/O
227	CHECKSTOP#	I/O
228	SYS_TT[4]	I/O
229	SYS_CI#	I/O
230	GND	
231	SYS_ADDR_BUS_RQ#	O
232	SYS_TBRST#	I/O

PIN#	SIGNAL NAME	TYPE
233	DPE#	I
234	TDO	O
235	HRESET_A#	I
236	Reserved	Unused
237	GND	
238	+5v	
239	SMI#	I
240	INT_A#	I
241	L2_PD0	Unused
242	GND	
243	MCP_A#	I
244	L2_PD1	Unused

PIN#	SIGNAL NAME	TYPE
245	+5v	
246	GND	
247	TBEN	I/O
248	SYS_ADDRP[2]	I/O
249	SYS_ADDRP[3]	I/O
250	L2_FLUSH#	Unused
251	L2_BG#	Unused
252	L2_INH#	Unused
253	L2_DISABLE#	Unused
254	FREQ_ID1	I
255	GND	
256	FREQ_ID3	I

Section 5
DRAM and ROM

This Section discusses the DRAM and ROM subsystems of the reference design. The DRAM and ROM controllers are located inside the 660 Bridge. The DRAM modules are plug into cards (SIMMs). The ROM is a Flash™ device which is socketed on the motherboard.

5.1 DRAM

The reference design system memory is composed of DRAM modules. The 660 Bridge memory controller interfaces the system memory to the rest of the reference design. The memory controller (MC) handles CPU and PCI access to memory, DRAM refreshing, and DRAM error checking.

The actual operation of the MC is covered in detail in the 660 Bridge User's Manual. The purpose of this section is to describe how the MC capabilities are implemented on the reference design.

The reference design supports up to 256M of DRAM, arranged as eight banks of the following:

- Industry standard 70ns, 72-pin, 4-byte SIMM
- One parity bit per byte, which can be used for parity or ECC
- Presence detect bits
- Asynchronous DRAM, page mode, or EDO (hyper-page mode).

Table 36 shows some supported devices. Different banks may use different sized SIMMs. Both of the SIMMs in the same bank must be the same size.

Table 36. Supported DRAM Modules

Size	Organization	IBM Part Number	DRAM Data Sheet
4MB	1M x 36b	IBM11E1360BA	MMDS14DSU-00
8MB	2M x 36b	IBM11E2360BA	MMDS22DSU-00
16MB	4M x 36b	IBM11E4360B	MMDS26DSU-00
32MB	8M x 36b	IBM11E8360B	MMDS27DSU-00

5.1.1 Memory Controller (DRAM)

The 660 Bridge contains a high performance memory controller which is extensively programmable. Refer to the 660 Bridge User's Manual for details.

In general, the MC is configured by the reference firmware for 70ns, page mode, asynchronous, parity DRAM arranged as pairs of 4-byte, 72-pin modules (see Section 13 for more setup information). EDO timings are not used, ECC memory checking is disabled, and parity checking is enabled. These settings can be modified as desired by the firmware.

5.1.2 Organization

The 4-byte (plus 4 parity bit) wide DRAM modules (SIMMs) are arranged in pairs to form 8-byte (plus 8 parity bit) wide memory banks, as shown in Figure 35 and Figure 36. The 2 SIMMs in each bank must be the same size.

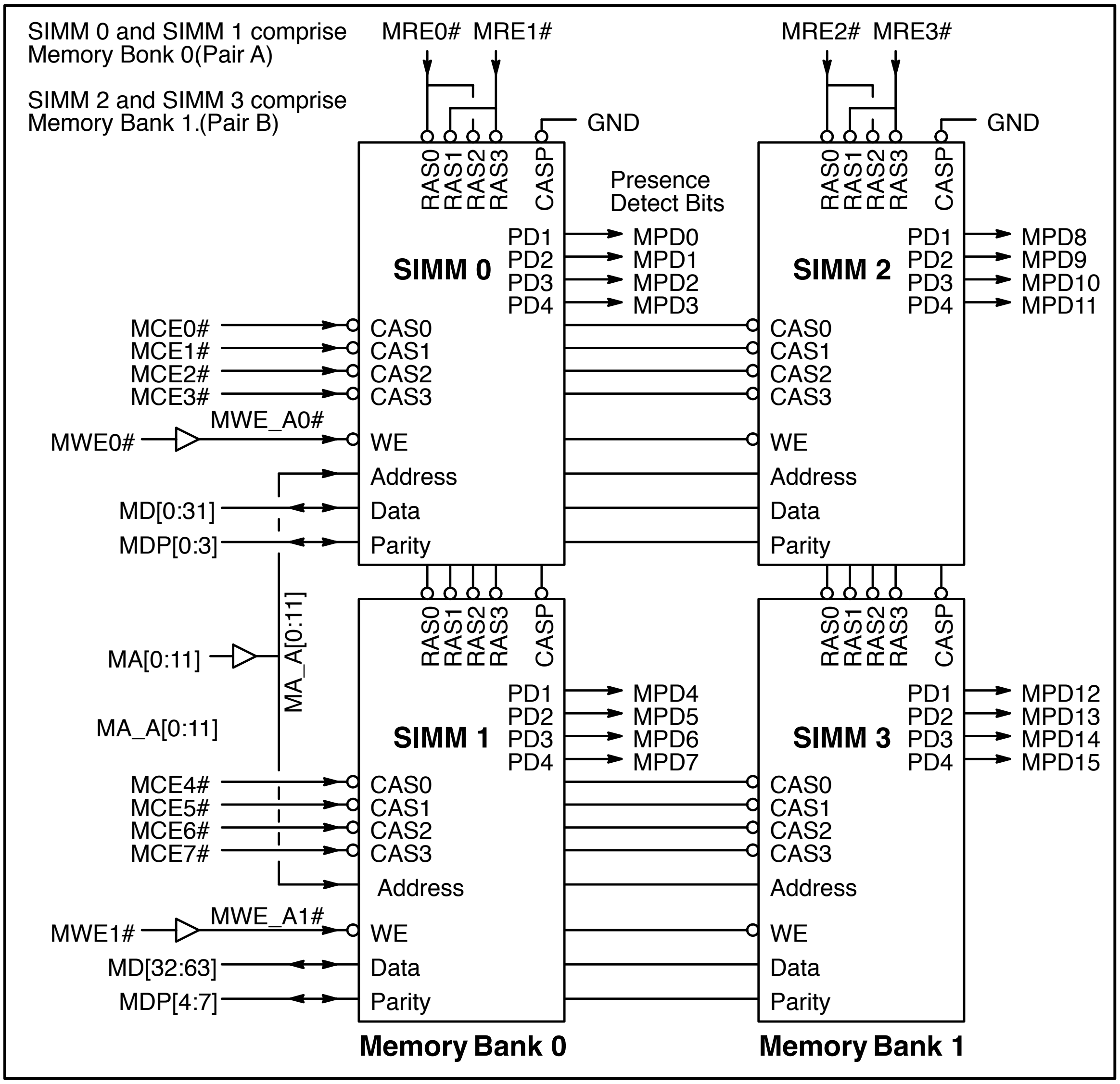


Figure 35. DRAM Banks 0 and 1 Organization

Combining the internal organization of the SIMMs (found in the respective data sheets) with the bank organization (found in Figure 35 and Figure 36) shows that each parity bit is accessed with the associated data byte. The 660 bridge uses this standard DRAM module organization for both parity and ECC modes of operation.

5.1.3 Refresh

The memory controller in the 660 bridge provides a flexible refresh capability for the reference design. See *The 660 Bridge User's Manual* for more information.

The ISA bus bridge provides the ISA_REFRESH# signal to refresh ISA bus memory. Refer to the SIO data book for more information.

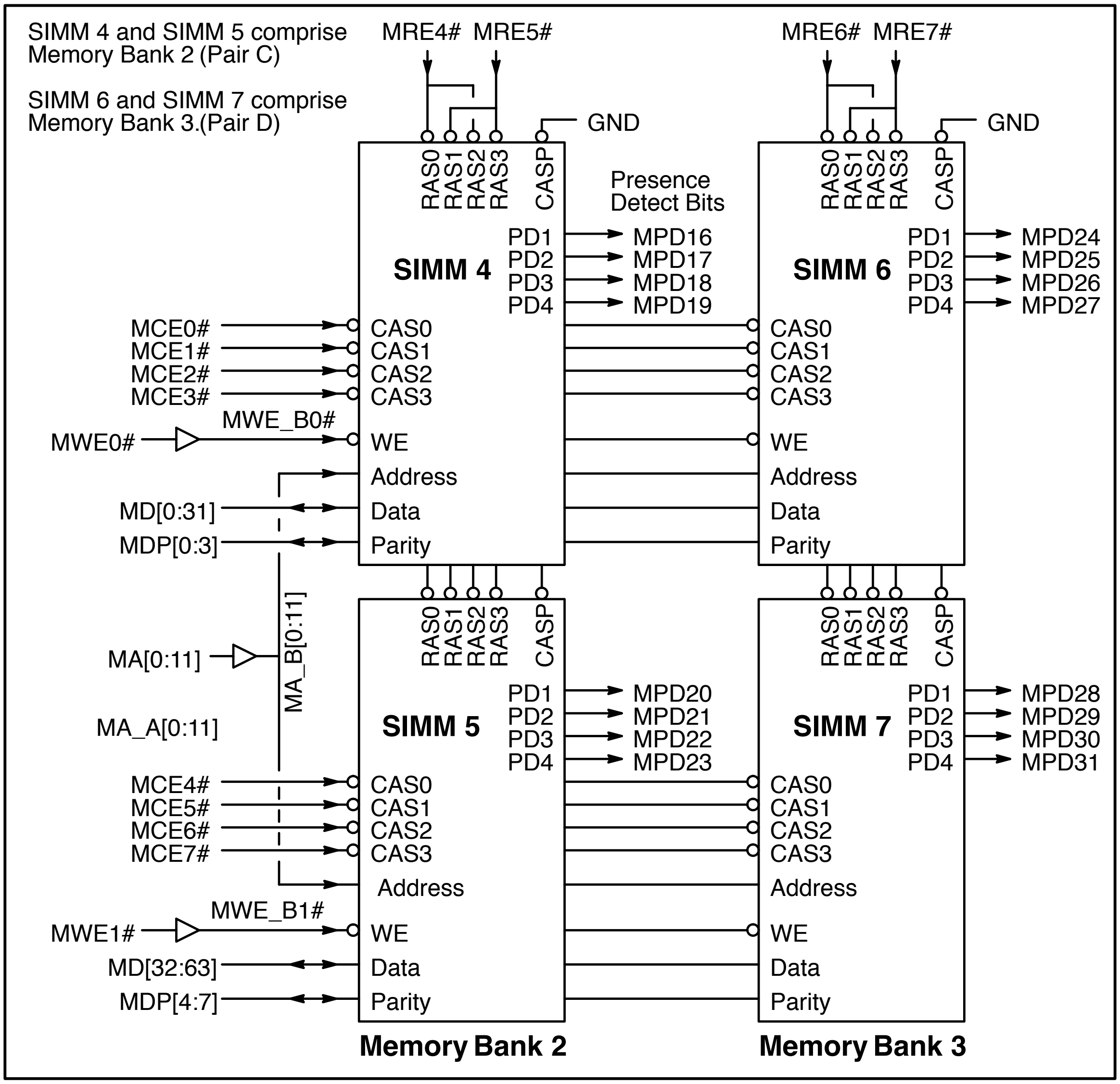


Figure 36. DRAM Banks 2 and 3 Organization

5.1.4 DRAM Presence Detection

Figure 35 and Figure 36 show the arrangement of the presence detect bits, MPD[31:0], which the firmware uses to detect and identify installed DRAM modules. See Section 2.7.1.

5.1.5 DRAM Bank Rules

All the DRAM SIMMs must:

- Be asynchronous, 72-pin, 4-byte SIMMs with presence detect bits.
- Be capable of operating at the timings selected by the memory controller registers.
- Be page mode if page mode operation is selected. If EDO operation is selected, then all SIMMs must be EDO.
- Be capable of the selected type of memory data checking (none or parity). ECC mode uses standard parity DRAM SIMMs.

Additionally, each SIMM in a given pair must be the same size and addressing (row vs. column) type.

5.2 ROM

The reference design uses an AMD AM29F040-120 Flash™ ROM to contain the POST and boot code. It is recommended that vital product data such as the motherboard speed and native I/O complement be programmed into in this device. It is possible to program the Flash before or during the manufacturing process. For more information, see *The 660 Bridge User's Manual* and Section 2.6 of this document. See Section 12 for information on the system firmware, and Section 13 for system setup information.

5.2.1 PCI Bus ROM

The reference design uses the direct-attach or PCI ROM attachment method. The ROM is connected to the PCI_AD[31:0] lines in a manner that makes it invisible to the PCI agents. The reference design uses 75Ω series isolation resistors to decouple the ROM from the PCI bus.

5.2.2 Remote ROM

The reference design has pads on the motherboard for a remote-attach ROM, which is located on the X-bus and accessed through the ISA bridge. The (SIO) ISA bridge currently installed on the reference design does not support this function, so it is not implemented.

5.2.3 ROM Read, Write, and Write Protect

The ROM used on the reference design is an AMD™ Flash™ ROM, 29F040. This device is read like a standard ROM. It can also be written to using a special protocol. The ROM can be placed in write-protected mode by writing to the 660 Bridge BCR. For more information, see *The 660 Bridge User's Manual* and Section 2.6 of this document.

Section 6 Exceptions

The 604 SMP reference design implements a full suite of SMP support features for the handling of interrupts, resets, and error conditions. Section 6.1 describes the interrupt capability of the reference design, Section 6.2 covers resets, and Section 6.4 describes the error handler.

6.1 Interrupts

The reference design interrupt subsystem uses the cascaded 8259 interrupt controllers in the ISA bridge in conjunction with the Multi-Processor Interrupt Controller (MPIC) to handle interrupts in a symmetrical manner (see Figure 37 and the data sheet in Section NO TAG). The software-configured hardware allows routing of interrupts from any source to any CPU, and provides the following capabilities:

- Routing of any PCI interrupt to any combination of from 1 to 4 CPUs.
- Assignment of the interrupt vector and threshold level to any PCI interrupt source.
- Assignment of the interrupt threshold level for each CPU.
- Handling of ISA interrupt by conventional means via cascaded 8259s mapped to a single interrupt input to MPIC. This combined interrupt is assigned a threshold level, a vector, and CPU routing attributes as needed.
- Implementation of CPU to CPU interrupts.
- Software control of soft reset to each CPU.
- Hardware and software control of hard reset to each CPU.
- Use of global timers to interrupt each CPU for SMP time sharing and other applications.

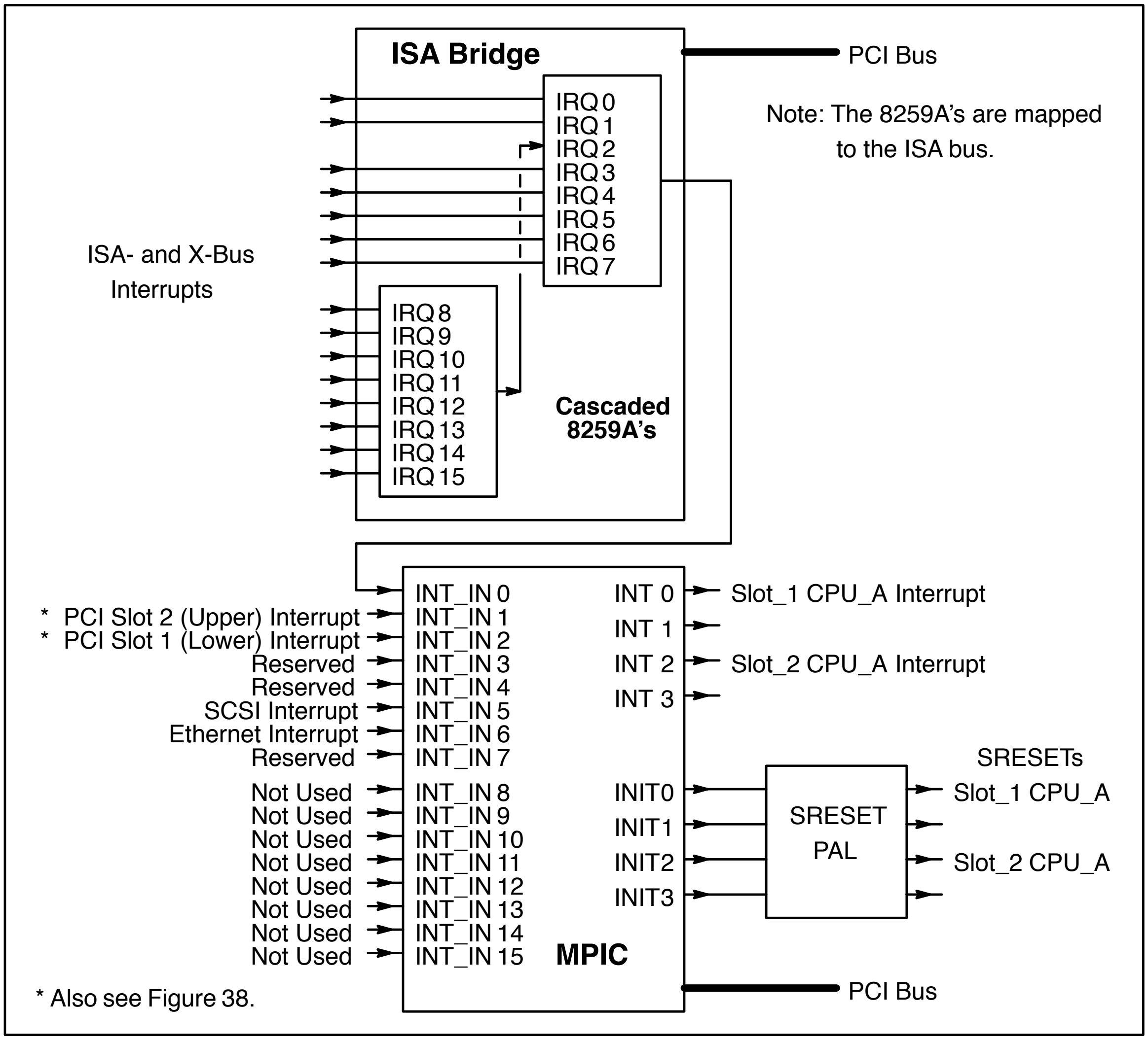


Figure 37. Interrupt Subsystem

6.1.1 PCI interrupt handling

The handling of a PCI interrupt is done using the following sequence:

1. The PCI device asserts an interrupt to MPIC.
2. MPIC routes the interrupt to a CPU, based on the assigned attributes for that interrupt and the CPU.
3. The CPU receives the interrupt vector by reading a memory mapped MPIC register.
4. MPIC responds with the interrupt vector.

6.1.2 ISA interrupt handling

The handling of a ISA interrupt is done using the following sequence:

1. An ISA or X-Bus device asserts an ISA interrupt (to the ISA Bridge).
2. The cascaded 8259A controllers in the ISA bridge route the ISA interrupt to MPIC.

3. MPIC routes the interrupt to a CPU, based on the assigned attributes for that interrupt and the CPU.
4. The CPU receives the interrupt vector by reading a memory mapped MPIC register.
5. MPIC responds with the vector assigned to all ISA interrupts.
6. The CPU then performs a memory mapped read to BFFF FFF0, which causes the 660 Bridge to initiate a PCI interrupt acknowledge cycle.
7. The ISA bridge responds to this PCI cycle with the 8 bit ISA interrupt vector.

6.1.3 CPU to CPU Interrupt Handling

A CPU can cause MPIC to interrupt a CPU by a write to a memory mapped register in MPIC. Handling of the interrupt by the target CPU follows the same sequence as a PCI interrupt.

6.1.4 PCI Interrupt Assignments

At each PCI slot, the INTA#, INTB#, INTC#, and INTD# interrupts are tied together and routed to MPIC as shown in Figure 38. Table 37 shows the MPIC interrupt assignments.

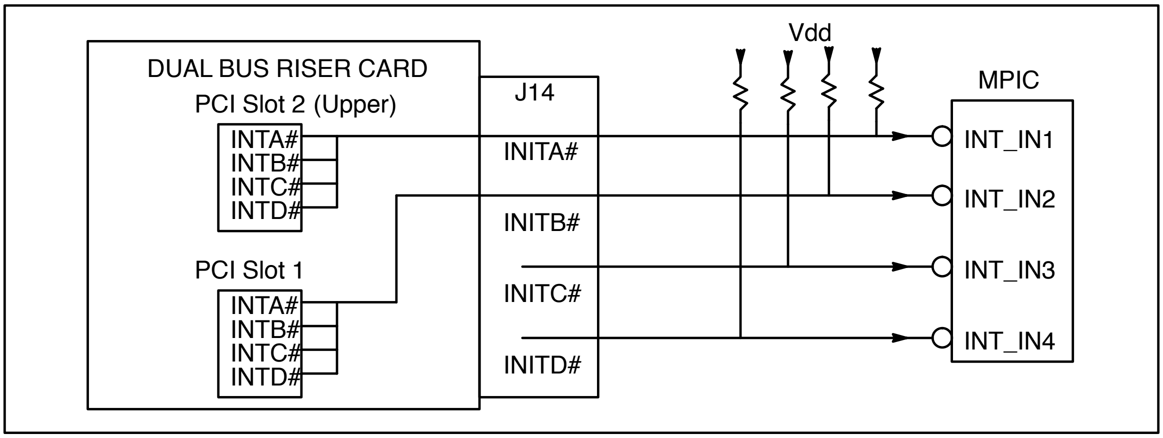


Figure 38. PCI Expansion Slot Interrupt Connections

Table 37. MPIC Interrupts

Number	TYPE	ASSIGNMENT OR CONNECTION
0	Level (Active High)	ISA Interrupts from PCI to ISA Bridge
1	Level (Active Low)	PCI Slot 2 (Upper) INTA,B,C,D
2	Level (Active Low)	PCI Slot 1 (Lower) INTA,B,C,D
3,4		Reserved (Should be masked)
5	Level (Active Low)	SCSI HDD
6	Level (Active Low)	Ethernet
7		Reserved (Should be masked)
8-15		Not Used (Should be masked)

6.1.5 ISA Interrupt Assignments

In accordance with the ISA standard, the reference design ISA interrupts operate in edge sensitive mode. Interrupts are assigned a priority level according to ISA conventions. For correct operation, program the ISA bridge to operate in edge sensitive mode.

Table 38. ISA Interrupt Assignments

	IRQ	PRIORITY	ASSIGNMENT or CONNECTION
Master	0	1	Timer 1 Counter 0 (Internal to the ISA Bridge)
	1	2	Keyboard
	2	3-10	Cascade from Controller 2
	3	11	COM 2, COM 4, ISA Pin B25
	4	12	COM 1, COM 3, ISA Pin B24
	5	13	Parallel LPT 2,3, ISA Pin B21, Audio (note 2)
	6	14	Floppy Disk, ISA Pin B23
	7	15	Parallel LPT 1,2, ISA Pin B21, Audio (note 2)
Slave	8#	3	Time of Day (aka RTC)
	9	4	ISA pin B04, Audio (note 2)
	10	5	ISA pin D03
	11	6	ISA pin D04, Audio (note 2)
	12/M	7	Mouse, ISA pin D05
	13#	8	DMA Scatter/Gather completion (programmable)
	14	9	ISA pin D07, Audio (note 2)
	15	10	ISA pin D06

Notes:

Note 1) IRQ10 and IRQ15 are available for ISA option cards. Also, either IRQ5 or IRQ7 can be used for ISA option cards, depending on which IRQ line the parallel port and the Audio controller are configured to use. IRQ9, IRQ11, and IRQ14 can be used for ISA option cards, depending on which IRQ lines the Audio controller is configured to use. IRQ0, IRQ1, IRQ2, IRQ8 and IRQ13 are not connected to the ISA slots. IRQ3, IRQ4, IRQ6, and IRQ12 are connected to motherboard devices and should not be used by ISA option cards. These IRQs are wired to the ISA option slots so that cards may detect, by sensing low levels, that they are used .

Note 2) IRQ5, IRQ7, IRQ9, IRQ11, IRQ12, and IRQ14 are available to the Audio subsystem. Enabling these interrupts can be done inside the Crystal CS4232 controller. The interrupts are connected to the system as follows:

CS4232 IRQ	System IRQ
A	5
B	7
C	9
D	11
E	12
F	14

6.1.5.1 Scatter/Gather (SG) Interrupts

Scatter/gather (SG) DMA support can use either IRQ13 or end of process (EOP) to indicate to the system that the SG sequence has been completed by the DMA controller. The use of EOP is recommended, since IRQ13 can be used by other ISA devices. The EOP signal from the ISA bridge generates the Terminal Count (ISA_TC) signal on the ISA bus.

IRQ13 can be connected optionally to the SCSI INT output on the motherboard to assist in the development of software that uses drivers which do expect an ISA-based HDD interrupt, and which do not support MPIC interrupts.

6.1.6 SCSI Bus Interrupts

The NCR 53C810 SCSI Controller has several interrupt sources; however, it will normally present only one interrupt at a time. A special case occurs when two interrupts arrive exactly on the same clock. In order to avoid dropping interrupts, the software must check all status registers in the controller and service all outstanding SCSI interrupts each time the CPU is interrupted by the SCSI controller.

6.1.7 MCP# Considerations

MCP# is an interrupt which is used to report system faults to PowerPC CPUs. The motherboard connects the MCP# of the 660 Bridge and the MCP# of both CPU slots together with a pullup resistor. Drivers on this net must be open drain to avoid bus contention with other devices. Refer to the 660 Bridge User's Manual for details on fault conditions that activate MCP#. CPU cards should wire MCP# to all devices which generate or monitor MCP#. There is no method on the motherboard for generating a CPU or slot specific MCP#.

6.1.8 SMI# Considerations

SMI# is a system management interrupt for the PowerPC 604 CPU. It can be used by power management circuits to awaken a NAPping or SLEEPing CPU back to full activity. The motherboard does not support SMI# and has this signal pulled up and connected to SMI# on both CPU slots.

6.2 Resets

The reference design features flexible SMP-capable support for the generation of HRESET#, SRESET#, and TRST# (for the JTAG interface).

6.2.1 HRESET# Logic

The reference design uses the circuit shown in Figure 39 to generate the HRESET# signals for the CPUs. The following are four methods of generating an HRESET# for an individual CPU. Three of the four are not CPU specific and will generate an HRESET# for all of the CPUs. None of these methods will reset any power management controller that may be in the system.

1. POWER_GOOD. When a power supply out-of-regulation condition is signaled by a low on POWER_GOOD from the power supply, the logic generates a hard reset to each CPU and to the motherboard devices.
2. CPU Enable Register. As detailed in Section 10, a CPU-specific hard reset or system reset is generated by using the system EPLD CPU Enable Register (Port 871) and the HRESET PAL.
3. RESET Jumper. When pins 1 and 2 of the RESET jumper J38 are shorted together, the logic generates a hard reset to each CPU and to the motherboard devices.
4. RISCWatch Reset. The ESP connector J18 allows the use of the RISCWatch tool, which is a JTAG-based debugging system. This tool requires the ability to initiate and hold all CPUs in the system in hard reset. When the RISCWatch asserts HRESET_ESP#, the HRESET PAL generates a hard reset to each CPU. The motherboard devices are not reset.

Figure 39 shows certain signals that are physically present on the motherboard, but which are not supported. These signals are subject to change or elimination from the reference design at any time. Each CPU slot is shown with reset support for two CPUs per slot; however, HRESET_CPU1B# and HRESET_CPU2B# are unsupported signals.

6.2.1.1 JTAG Interface Hard Resets

The JTAG interface with the 604 CPU(s) must be held in reset while HRESET# is active. The HRESET PAL supplies the TRST_CPU1# signal to reset the JTAG interface with the CPU(s) in CPU slot 1, and supplies the TRST_CPU2# signal to reset the JTAG interface of the CPU(s) in CPU slot 2.

TRST_CPU1# is asserted whenever HRESET_CPU1A# or HRESET_CPU1B# is asserted, and TRST_CPU2# is asserted whenever either of the HRESET#s for that CPU slot is asserted.

The reference design also asserts both TRST# signals while the RISCWatch interface is asserting OCS_OVRIDE.

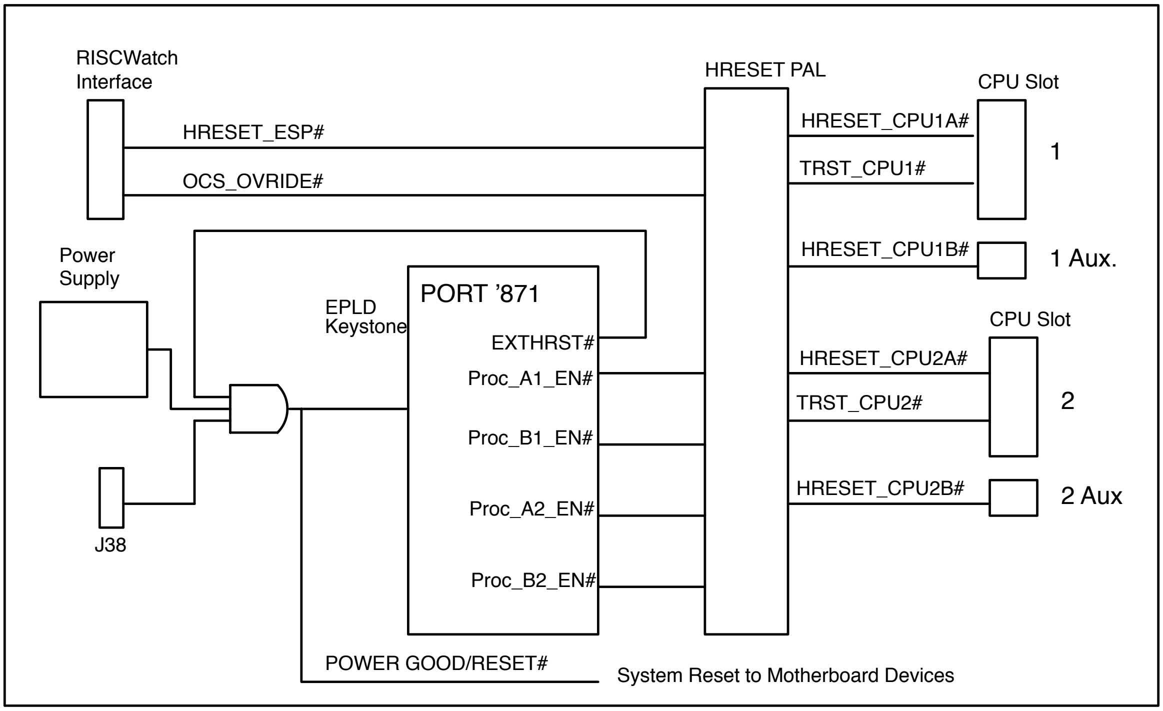


Figure 39. HRESET# Signal Circuit

6.2.1.2 HRESET PAL Equations

```
TITLE    Hard–Reset Logic
PATTERN  hrst2.pds
REVISION 0
COMPANY  IBM
DATE     05/17/95  05:45pm
CHIP   hreset   PAL16L8

;——————————————————— PIN Declarations —————————————— 20plcc PIN

;PIN            NC                         ; INPUT      1
;PIN            NC                         ; INPUT      2
PIN  3          CKST_P2_                   ; INPUT      3
PIN  4          CKST_P1_                   ; INPUT      4
PIN  5          KSRS_P1A_                  ; INPUT      5
PIN  6          KSRS_P1B_                  ; INPUT      6
PIN  7          KSRS_P2A_                  ; INPUT      7
PIN  8          KSRS_P2B_                  ; INPUT      8
PIN  9          OCS_OVRIDE                 ; INPUT      9
;PIN            NC                         ;        gnd 10
PIN  11         HRST_ESP_                  ; INPUT      11
PIN  12         CKST_ESP_    COMBINATORIAL   ; OUTPUT      12
PIN  13         HRST_P1A_    COMBINATORIAL   ; OUTPUT      13
PIN  14         HRST_P1B_    COMBINATORIAL   ; OUTPUT      14
PIN  15         HRST_P2A_    COMBINATORIAL   ; OUTPUT      15
PIN  16         HRST_P2B_    COMBINATORIAL   ; OUTPUT      16
PIN  17         TRST_P1_     COMBINATORIAL   ; OUTPUT      17
PIN  18         TRST_P2_     COMBINATORIAL   ; OUTPUT      18
;PIN            NC                           ; INPUT       19
;PIN            NC                           ;         vcc 20

;———————————— Boolean Equation Segment ——————

/CKST_ESP_  = /CKST_P1_ * /CKST_P2_

/HRST_P1A_  = /HRST_ESP_ + /KSRS_P1A_

/HRST_P1B_  = /HRST_ESP_ + /KSRS_P1B_

/HRST_P2A_  = /HRST_ESP_ + /KSRS_P2A_

/HRST_P2B_  = /HRST_ESP_ + /KSRS_P2B_

/TRST_P1_  = /OCS_OVRIDE + /KSRS_P1A_

/TRST_P2_  = /OCS_OVRIDE + /KSRS_P2A_
```

6.2.2 SRESET Logic

The SRESET# signal provides a method of software resetting the CPUs in the reference design. It can be used by software to restart from a know state and, in SMP systems, to sequence and initialize CPUs. This signal is negative edge sensitive.

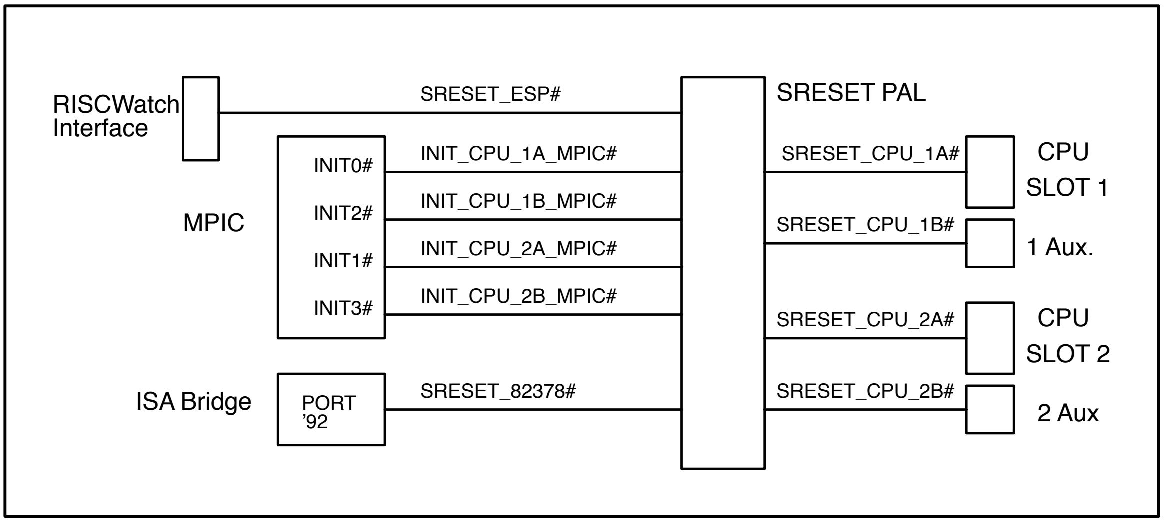


Figure 40. SRESET# Signal Circuit

The reference design uses the circuit shown in Figure 40 to generate the SRESET# signals to the CPUs. There are three methods of generating an SRESET# to an individual CPU. Two of the three are not CPU specific and will generate an SRESET# to all of the CPUs. Note that none of these methods will reset either the motherboard devices or any power management controller which may be in the system.

- RISCWatch Reset. The ESP connector, J18, allows the use of the RISCWatch tool, which is a JTAG-based debugging system. This tool requires the ability to issue SRESET# to all CPUs in the system. When the RISCWatch asserts SRESET_ESP#, the SRESET PAL generates a soft reset to each CPU (by sending each SRESET# low). Note that while SRESET_ESP# is active, soft resets from other sources will be masked and not stored. It is, therefore, important to deactivate SRESET_ESP# once the CPUs have been reset.
- Port 92. When a 1 is written to bit 0 of port 92, the ISA bridge generates a low pulse to the SRESET PAL, which in turn pulses each SRESET# low.
- CPU Init Register. As detailed in the MPIC data sheet (see Section NO TAG), a CPU-specific soft reset can be generated by using the memory-mapped MPIC INIT register to assert INIT_CPU_x_MPIC#. (**Note:** The SRESET# is a negative-edge-triggered signal.) The best way to do this is to first write a 0 (to send INIT_CPU_x_MPIC# high) then a 1 (to send it back low). The first write may or may not effect a change in the state of the SRESET#, but the second write will send the SRESET# low. Sending the INIT_CPU_x_MPIC# back high (to restore it) can be done either by the SRESET# service routine, or as above.

Port 92 and the MPIC registers can be used asynchronously to generate soft resets without masking each other because the SRESET PAL XORs the INIT_CPU_x_MPIC# signals with an inverted Port 92 signal. Also see Section 6.2.3 for the SRESET PAL equations.

Figure 40 shows certain signals that are physically present on the motherboard but are not supported. These signals are subject to change or elimination from the reference design at any time. Each CPU slot is shown with reset support for two CPUs per slot; however, SRESET_CPU1B# and SRESET_CPU2B# are unsupported signals.

6.2.3 SRESET PAL equations

```
TITLE    Soft–Reset Logic for SIO
PATTERN  SRSTP2A.pds
REVISION 3.0
COMPANY  IBM
DATE     09/11/95 11am
CHIP   sreset  PAL16L8


;------------------------ PIN Declarations ---------  20plcc PIN
PIN  1          SRS_ESP_                    ; INPUT       1
PIN  2          SRS_PORT92                  ; INPUT       2
PIN  3          INIT_P1A_MPIC_              ; INPUT       3
PIN  4          INIT_P1B_MPIC_              ; INPUT       4
PIN  5          INIT_P2A_MPIC_              ; INPUT       5
PIN  6          INIT_P2B_MPIC_              ; INPUT       6
PIN  7          HLTQ_P1A                    ; INPUT       7
PIN  8          HLTQ_P1B                    ; INPUT       8
PIN  9          HLTQ_P2A                    ; INPUT       9
;PIN            NC                          ;       gnd   10
PIN  11         HLTQ_P2B                    ; INPUT       11
PIN  12         RUNH_ESP   COMBINATORIAL    ; OUTPUT      12
PIN  13         SRS_P1A_   COMBINATORIAL    ; OUTPUT      13
PIN  14         SRS_P1B_   COMBINATORIAL    ; OUTPUT      14
PIN  15         SRS_P2A_   COMBINATORIAL    ; OUTPUT      15
PIN  16         SRS_P2B_   COMBINATORIAL    ; OUTPUT      16
PIN  17         HALT_P1    COMBINATORIAL    ; OUTPUT      17
PIN  18         HALT_P2    COMBINATORIAL    ; OUTPUT      18
;PIN            NC                          ; INPUT       19
;PIN            NC                          ;       vcc   20
;
;
;
```

```
;----------------------------- Boolean Equation Segment -------
;
; Enable edge sensitive Sreset on MPIC output. SRESET from ISA
; bridge is assumed to be a negative pulse.
;

EQUATIONS

/SRS_P1A_  = /SRS_ESP_ + /(/SRS_PORT92 :+: INIT_P1A_MPIC_)

/SRS_P1B_  = /SRS_ESP_ + /(/SRS_PORT92 :+: INIT_P1B_MPIC_)

/SRS_P2A_  = /SRS_ESP_ + /(/SRS_PORT92 :+: INIT_P2A_MPIC_)

/SRS_P2B_  = /SRS_ESP_ + /(/SRS_PORT92 :+: INIT_P2B_MPIC_)
;
; Halt outputs are active high when both halt inputs are high
;

/HALT_P1  = /HLTQ_P1A + /HLTQ_P1B

/HALT_P2  = /HLTQ_P2A + /HLTQ_P2B

/RUNH_ESP = HLTQ_P1A * HLTQ_P1B * HLTQ_P2A * HLTQ_P2B
```

6.3 SMP Reset Considerations

This section describes SMP reset and system initialization procedures. See Figure 41.

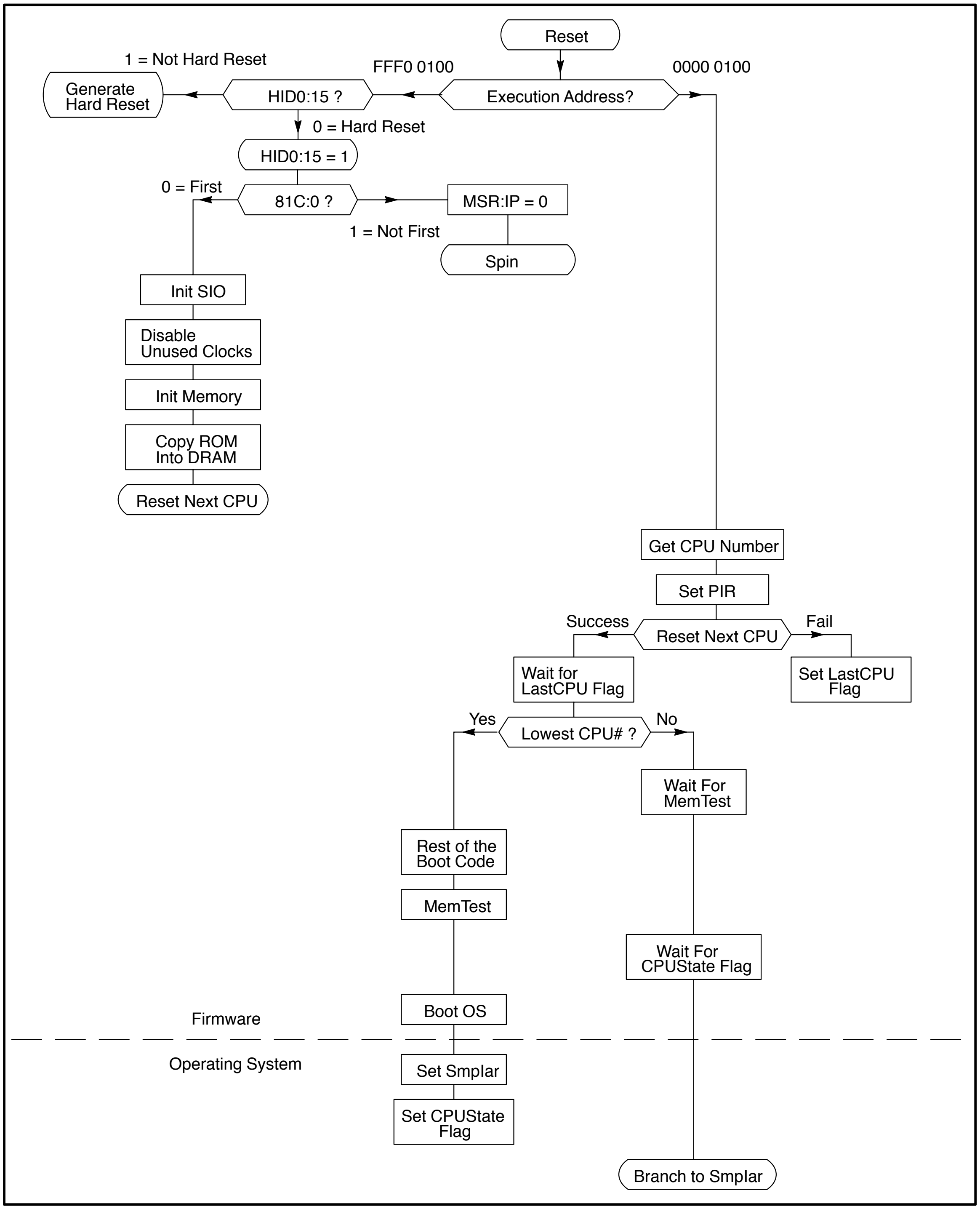


Figure 41. SMP System Reset

6.3.1 Hard Reset

This section describes the system initialization procedure following the assertion of HRESET. Section 6.2.1 describes how HRESET is generated. See Figure 41.

Booting an SMP system is more complex than booting a uniprocessor system, and has the following additional requirements:

- Each processor must have a unique identification number stored in its Processor Identification Register (PIR).
- A single processor must assume control of the boot process. All other processors must be placed in a harmless state until the operating system restarts them.

A three phase hard reset process is used. These phases are summarized below and are described in detail in subsequent sections:

1. After hard reset, one processor is selected as the master for the first phase of booting. All other processors are disabled. The initial master does some minimal system initialization (memory and ISA bridge) then begins resetting each processor slot in sequence until a processor is reset.
2. Upon being reset, each processor does some processor-specific initialization, tries to reset the next available processor, and then waits until all processors have been reset.
3. After all processors have been reset, the lowest numbered processor becomes the system master and proceeds with the remainder of the system initialization sequence. All other processors are placed in a loop, waiting for a memory flag to be set, after which they will branch to an address contained in a predetermined memory location.

6.3.1.1 Hard Reset Phase 1

A processor enters Hard Reset Phase 1 if it is executing instructions out of ROM (address 0xFFF00100) and if bit 15 of the HID0 register is 0. This bit is guaranteed to be a zero after the assertion of the processor's HRESET input.

HID0:15 is set to 1 so that subsequent execution from 0xFFF00100 will be correctly interpreted as a Soft Reset.

Each processor reads the System Control Register (0x8000081C). The first processor to read this register will read a 0 in the FirstRead# bit (D0). All other processors will read a 1.

The processor that reads the 0 becomes the master for the remainder of this phase of the boot process. All other processors set their MSR:IP to 0 so that on a subsequent soft reset they will begin executing out of RAM at address 0x00000100. They then begin looping until they are reset.

The Phase 1 master does some minimal system initialization (SIO, memory, &etc.) and copies the firmware image from ROM address 0xFFF00000 to RAM address 0x00000000.

The Phase 1 master sets its MSR:IP to 0 and uses the MPIC Processor Init Register to issue a Soft Reset to processor 0. Three possibilities now exist:

1. If Processor 0 was the Phase 1 master, then it has just reset itself and enters Hard Reset Phase 2, described below.
2. If Processor 0 was not the Phase 1 master, then the Phase 1 master waits up to 10ms for processor 0 to set a flag indicating that it has been successfully reset. If processor 0 has been reset, then the Phase 1 master spins in a loop until it is reset.
3. If Processor 0 was not reset, then the Phase 1 master increments the CPU sequence register by reading it and discarding the result. This ensures that the number returned by the CPU Sequence register is in sync with the number of the processor currently being reset. The Phase 1 master then tries to reset processor 1, and repeats this process until a processor is successfully reset.

6.3.1.2 Hard Reset Phase 2

A processor enters Hard Reset Phase 2 if it is executing instructions out of RAM at address 0x00000100.

Upon entering Hard Reset Phase 2, each processor sets a bit in a processor reset table to indicate that it has been reset.

The processor then reads a Proccessor ID number (PID) from the CPU Sequence Register and stores the result into its Processor Identification Register (PIR).

The processor then tries to reset the next processor, using the same procedure used in Hard Reset Phase 1.

If a subsequent processor cannot be reset, then the current processor sets the LAST_PROC_RESET flag to indicate that the last processor has been reset. If a subsequent processor is reset, then the current processor waits for the LAST_PROC_RESET flag to be set.

6.3.1.3 Hard Reset Phase 3

After all processors have been reset, the processor with the lowest Processor ID number becomes the system master for the remainder of the boot process. All other processors loop, waiting until the MEMORY_TEST_COMPLETE flag is set (which indicates that the system memory test is complete).

Then the system master initializes the rest of the system hardware.

After the system memory has been tested, the system master sets the MEMORY_TEST_COMPLETE flag, which releases all other processors from their polling loop. These other processors then begin polling a field in the residual data structure.

The system master loads the operating system loader from the selected boot device and transfers control to it.

The operating system now has control of the system. When it wants to start up the other processors, it uses two fields in the residual data structure, which is passed by the firmware. Residual.VitalProductData.SmpIar is loaded with the address that the second processor should branch to. The appropriate Residual.Cpus[].CpuState field is set to CPU_GOOD.

The second processor, upon seeing CPU_GOOD in the CpuState field, branches to the address stored in the SmpIar field.

6.3.2 Soft Reset

It is often desirable to have a software-initiated system reset. For example, some operating systems give the user the option of restarting the system after a system shutdown. See Figure 41.

A processor enters Soft Reset phase if it is executing instructions from ROM address 0xFFF00100 and HID0:15 = 1. This can happen either by branching to address 0xFFF00100 or by setting MSR:IP = 1 and issuing a SRESET to the processor. Section 6.2.2 describes how SRESET is generated.

Upon entering Soft Reset phase, a processor writes the value 0x00 to the CPU Enable Register (Port 871). This causes a Hard Reset to be generated, which resets all system hardware and causes all processors to enter Hard Reset Phase 1.

6.4 Error Handling

The reference design offers a full set of error handling features, most of which are accomplished by the 660 Bridge. For more information, see the 660 Bridge User's Manual.

6.4.1 Data Error Checking

The reference design is initially configured to use parity, and this documentation reflects that configuration; however, the 660 Bridge can be programmed to execute an error checking and correction (ECC) algorithm on the memory data. This generates ECC check bits during memory writes and checks-corrects the data during memory reads. ECC can be implemented using normal parity DRAM. See *The 660 Bridge User's Manual* for more information.

For each memory read operation, eight bytes of memory are read, and parity on eight bytes is checked regardless of the transfer size; therefore, all memory must be initialized (at least up to the end of any cache line that can be accessed).

The reference design does not generate or check CPU bus address parity.

6.4.1.1 CPU to Memory Writes

During CPU to memory writes, the CPU drives data parity information onto the CPU data bus. Correct parity is then generated in the 660 and written to DRAM memory along with the data. The L2 SRAM is updated (when required) with the data and the parity information that the CPU drove onto the CPU data bus.

During CPU to memory writes, the 660 Bridge checks the data parity sourced by the CPU, and normally reports any detected parity errors via TEA#.

6.4.1.2 CPU to Memory Reads

When the CPU reads from memory, the data and accompanying parity information can come from either the L2 SRAM or from DRAM memory. If the data is sourced from the L2, the parity information also comes from the L2.

If the data is sourced by memory, the parity information also comes from memory. The L2 SRAM is updated (when required) using the data and parity from memory.

During CPU to memory reads, the 660 Bridge samples the DPE# output of the CPU to determine parity errors, and reports them back to the CPU via MCP#. The particular memory read data beat will be terminated normally with TA#.

6.4.1.3 PCI to Memory Parity Errors

During PCI to memory writes, the 660 Bridge generates the data parity that is written into DRAM memory. The 660 Bridge also checks the parity of the data and asserts PCI_PERR# if it detects a data parity error.

During PCI-to-memory reads, the 660 Bridge checks the parity of the memory data, and then generates the data parity that is driven onto the PCI bus. If there is a parity error in the data/parity returned to the 660 Bridge from the DRAM, the bridge drives PCI_PAR incorrectly to propagate the parity error (and also reports the error to the CPU via MCP#). The data beat with the bad parity is not target aborted because doing so would slow all data beats for one PCI clock (TRDY# is generated before the data is known good); however, if the agent is bursting and there is another transfer in the burst, the next cycle is stopped with target abort protocol.

During PCI to memory reads, the 660 Bridge also samples the PCI_PERR# signal, which other agents can be programmed to activate when they detect a PCI parity error.

6.4.1.4 CPU to PCI Transaction Data Parity Errors

During CPU to PCI writes, the 660 Bridge sources the PCI parity information, and monitors PCI_PERR#, which other agents can be programmed to activate when they detect a PCI parity error.

During CPU to PCI reads, the 660 Bridge checks the data parity and asserts PCI_PERR# if it detects a data parity error.

6.4.2 Illegal CPU cycles

Whenever a CPU transfer which is not supported for memory or for the PCI is detected, the cycle is terminated with a TEA# and the illegal transfer register is set. No memory or PCI cycle is initiated. Read data returned is all 1's. The CPU address is captured in the Error Address Register.

6.4.3 SERR, I/O Channel Check, and NMI Logic

The PCI bus defines a signal called SERR#, which any agent can pulse. This signal is to report error events within the devices, not bus parity errors. The signal is wired to the ISA bus bridge. The ISA bus signal IOCHCK is also wired to the ISA bridge. If either of these lines activate, the ISA bridge asserts NMI to the 660 Bridge, unless the condition is masked by a register within the ISA bridge. The NMI signal causes the 660 Bridge to generate an interrupt to the CPU and to assert MCP# to the CPU. The ISA bridge contains status registers to identify the NMI source. Software may interrogate the ISA bridge and other devices to determine the source of the error.

No address is associated with this type of error; therefore, the contents of the error address register are not defined.

6.4.4 Out of Bounds PCI Memory Accesses

If a PCI bus master runs a cycle to a system memory address above the top of physical memory, no one will respond, and the initiator master aborts the cycle. The initiating bus master must be programmed to notify the system of master aborts as needed. The system logic does not notify the CPU.

6.4.5 No Response on CPU to PCI Cycles — Master Abort

The 660 Bridge master aborts if no agent responds with DEVSEL# within eight clocks after the start of a CPU to PCI cycle. The cycle is ended with a TEA# response to the CPU, all 1's data is returned on reads, the Illegal Transfer Error register is set, and the Error Address register is held.

The 660 Bridge also checks for bus hung conditions. If a CPU to PCI cycle does not terminate within approximately 60 usec after the PCI is owned by the CPU, the cycle is terminated with TEA#. This is true for all CPU to PCI transaction types except configuration transactions. This feature may be disabled via a 660 Bridge control register.

In the case of configuration cycles that do not receive a DEVSEL# (no device present at that address), the PCI cycle is master aborted, and TA# (normal response) is returned. Write data is thrown away and all 1's are returned on read cycles. No error register is set and no address is captured in the error address register.

6.4.6 CPU to PCI Cycles That Are Target Aborted

When any CPU to PCI cycle of any sort receives a Target Abort from its target PCI agent, the CPU cycle is terminated with a TEA#, the Error Address register is held, and the Illegal Transfer Error register is set.

6.4.7 Error Status Registers

Error status registers in the 660 Bridge may be read to determine the types of outstanding errors. Errors are not accumulated while an error is outstanding; however, there will be one TEA# or MCP# for each error that occurs. For example, if an illegal transfer error causes a TEA#, a memory parity error can occur while the CPU is processing the code that handles the TEA#. The second error can occur before the error status registers are read. If so, then the second error status is not registered, but the MCP# from the memory parity error is asserted.

6.4.8 Reporting Error Addresses

One register holds the address for any memory parity error, multi-bit ECC error, or illegal transfer error when either the CPU or a PCI agent reads memory. It is also loaded on CPU cycles that cause the Illegal Transfer Error register to be set.

See the *System Error Address BCR* information in the *Bridge Control Registers* section of *The IBM27-82660 PowerPC to PCI Bridge User's Manual* for more information.

6.4.9 Errant Masters

Either PCI or ISA masters can access certain motherboard and ISA bridge registers. For example, various control registers such as the I/O Map Type register, the BE/LE mode bit, the Memory Control registers, etc. are accessible. Faulty code in the PCI or ISA masters can defeat password security, read the NVRAM, and cause the system to crash without recovery. Take care when writing device drivers to prevent these events.

6.4.10 Special Events Not Reported as Errors

- A PCI to memory cycle at any memory address above that programmed into the top of memory register.

  The 660 bridge ignores this cycle, and the initiator master aborts. No data is written into system memory on writes, and the data returned on reads is indeterminate. The bus master must be programmed to respond to a target abort by alerting the host.

- CPU to PCI configuration cycles to which no device responds with a DEVSEL# signal within 8 clocks (no device at this address).

  The data returned on a read cycle is all 1's, and write data is discarded. It is consistent with the PCI specification and allows software to scan the PCI at all possible configuration addresses.

- A CPU read of the IACK address having a transfer size other than 1, or having other than 4-byte alignment.

  These conditions return indeterminate data. The ISA bridge requires the byte enables, CBE#3:0, to be 1110 in order to place the data on the correct byte lane (0). Accesses other than one byte at the address BFFF FFF0h are undefined.

- A read of the IACK address when no interrupt is pending.

  A DEFAULT 7 vector is returned in this case. This is the same vector that is returned on spurious interrupts.

- Parity error in Flash/ROM.

  Parity is not stored in the Flash ROM. Therefore the memory parity error signal and the DPE signal are ignored during ROM reads. The Flash or ROM should include CRC with software checking to insure integrity.

- Write to Flash with TSIZ other than 4.

  This will cause indeterminate data to be written into the Flash at an indeterminate address.

- Caching ROM space.

  An L1 or CB-L2 cast out will cause indeterminate results.

- Running any cycle to the PCI configuration space with an undefined address.

  Some of these could potentially cause damage. See the warning under the PCI configuration cycle section.

- Accessing any ISA device with the wrong data size for that device.

  Indeterminate results will occur.

Section 7
Clocks

The 604 SMP Reference Design provides a separate CPU clock line for each CPU bus agent or other CPU clock consumer. The CPU bus clocks used on the reference design motherboard are generated by the master PLL, which supplies all of the motherboard CPU clocks. It also supplies three CPU clocks to each CPU slot. When required, additional CPU bus clocks can be generated on a CPU card using another PLL.

The reference design also provides a separate PCI clock for each PCI agent. These clocks are generated by the PCI PLL on the motherboard, which is synchronized to the master PLL. The PLL produces PCI clocks that are nominally not delayed from the CPU clock.

Other clocks in the system are generated by oscillators or crystals which are provided on the motherboard, and are located close to the intended load.

7.1 CPU Clocks

As shown in Figure 42, the master PLL is a Motorola™ MPC970 PLL clock generator. The master PLL is configured to drive all of its outputs at the same frequency (even the PCI_CLK outputs). The master PLL supplies:

• BCLK3, BCLK4, and 2X_PCLK	CPU Slot 1
• BCLK1, BCLK2, and PCLK_EN	CPU Slot 2
• BCLK0 and BCLK_EN	660 Bridge
• PCI_CLK[5:1]	L2 TagRAM and SRAM clocks
• PCI_CLK[0]	Seed clock for PCI clock PLL
• PCI_CLK[6]	Spare

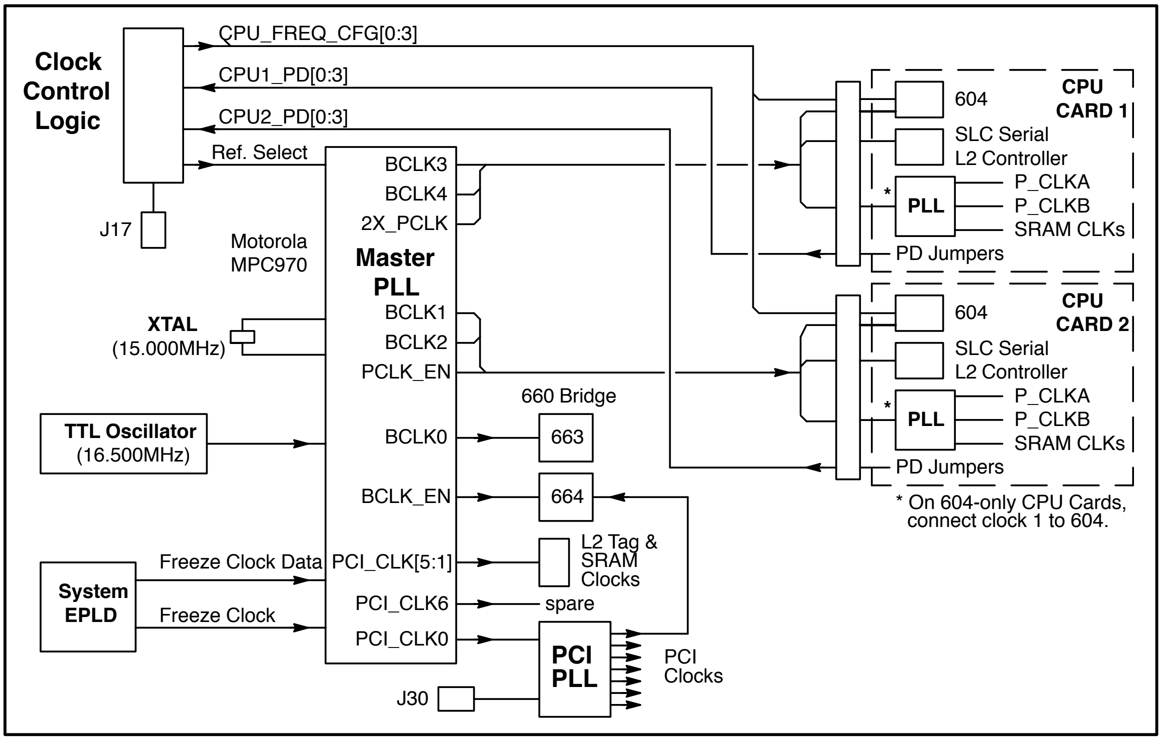


Figure 42. System Clocks

7.1.1 CPU Card Clock Repeater

The reference design motherboard supplies three CPU clocks to each CPU slot. If a CPU card requires more than 2 clocks, a ’zero delay’ clock repeater such as the Motorola™ MPC930 PLL can be used on the CPU card to generate the additional clocks, using one of the supplied CPU clocks as the seed clock.

7.1.2 CPU Clock Physical Design Rules

The reference design board was physically designed with careful attention to the fact that, at PowerPC operating frequencies, the circuit board itself becomes a component that materially affects circuit behavior. The following steps were used to control this:

- Noise generation, reflections, and impedance mismatches were reduced by routing the CPU clock lines on internal planes of the motherboard with (frequently grounded) ground traces running in adjacent wiring channels.
- The number of vias per clock line was minimized.
- Source termination resistors were chosen according to the MPC970 data sheet recommendations.
- EMC capacitors were added to the clock nets.

To minimize clock skew, each of the CPU clock lines was made the same total length. In the case of the clock lines that run to CPU cards, part of the clock length appears on the motherboard, and part of it appears on the CPU card. Each of the clock lines runs only from the master PLL driver to the clock receiver of the CPU clock consumer. Also see Section 18, Physical Design, for more information.

7.2 CPU Clock Control Logic

The reference design automatically sets the CPU bus clock frequency and the configuration bits of the PLLs in the 604 CPU cards. As shown in Figure 43, the clock control logic makes these decisions based on the value of the presence detect bits from each CPU slot and the state of J17. The reference design:

- Supplies the same frequency CPU clocks to each CPU slot.
- Supplies the same PLL configuration information to each CPU slot.
- Selects the fastest allowed bus clock frequency and CPU frequency that is acceptable to both CPUs.

The clock control logic is implemented in the frequency select PAL (FRQSEL) and an ’F157 quad 2:1 Multiplexer.

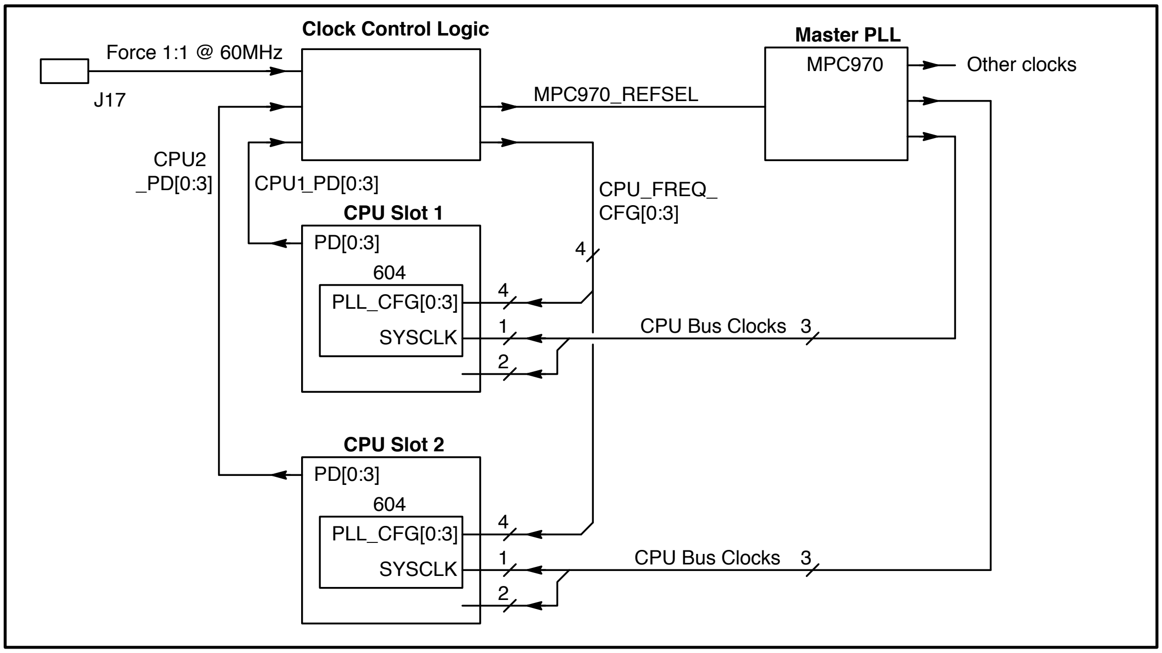


Figure 43. Frequency Selection Block Diagram

7.2.1 Clock Logic Input – J17

Regardless of the values present on the CPU slot PD bits, if J17 is installed, the clock control logic sets the CPU clock speed to 60MHz, and will set the CPU PLL control bits to 0001 (1:1 CPU to bus clock ratio and a 4:1 PLL to CPU clock ratio).

7.2.1.1 Clock Logic Input – CPU Slot PD Bits

As shown in Figure 43, the clock control logic receives the speed capability information from each CPU slot on the presence detect bits for that slot. The encoding of these bits is shown in Table 39.

Table 39. CPU Slot PD Bit Encoding

CPU_PD[0:3]	CPU Type
0000	No CPU on card.
0001	100 MHz CPU
0010	120 MHz CPU
0011	132 MHz CPU
0100	150 MHz CPU
0101	167 MHz CPU
0110	180 Mhz CPU
0111–1110	Reserved
1111	No CPU card present.

7.2.1.2 Clock Logic Output – CPU Bus Clock Frequency Select

The clock control logic controls the frequency of the CPU bus clocks by setting the master PLL (MPC970) REFSEL input to select either 60MHz or 66MHz operation. This instructs the MPC970 to select either the crystal or the oscillator as the source of the seed frequency.

7.2.1.3 Clock Logic Output – 604 PLL Configuration

The clock control logic asserts CPU_FREQ_CFG[0:3] to both CPU cards, which connect these lines to the PLL_CFG[0:3] inputs of the 604 CPU. This information causes the 604 to configure its onboard PLL as described in the 604 User's Manual. This configuration determines the ratio between the 604 internal CPU clock, the 604 internal PLL frequency, and the CPU bus clock.

7.2.1.4 Clock Logic Transfer Function

Table 40 shows the outputs of the clock control logic as a function of the inputs. The headings along the top of the table show the presence detect bits from CPU slot 1, along with their meanings. The headings on the left side of the table show the same information for CPU slot 2.

Except as noted, each of the entries in the body of the table shows the CPU bus clock as either 60 or 66 MHz, and also shows the value that the logic asserts on the CPU_FREQ_CFG[0:3] lines to both CPU cards.

Table 40. CPU Clock Control Logic

Frequency Table Bus Speed (Mhz) and PLL Settings (freq_id 0,1,2,3)			CPU Card Slot 1 PD[0:3]									
			0000	0001	0010	0011	0100	0101	0110	0111	1000– 1110	1111
			No CPU	100 Mhz CPU	120 Mhz CPU	132 Mhz CPU	150 Mhz CPU	167 Mhz CPU	180 Mhz CPU	Rsrvd	Rsrvd	No card pres.
CPU Card Slot 2 PD [0:3]	0000	No CPU	**ERR**	**66** (1100)	**60** (0100)	**66** (0100)	**60** (0110)	**66** (0110)	**60** (1000)	**ERR**	**ERR**	**ERR**
	0001	100 Mhz CPU	**66** (1100)	**66** (1100)	**66** (1100)	**66** (1100)	**66** (1100)	**66** (1100)	**66** (1100)	**66** (1100) **	**60** (0001)	**66** (1100)
	0010	120 Mhz CPU	**60** (0100)	**66** (1100)	**60** (0100)	**60** (0100)	**60** (0100)	**60** (0100)	**60** (0100)	**60** (0100) **	**60** (0001)	**60** (0100)
	0011	132 Mhz CPU	**66** (0100)	**66** (1100)	**60** (0100)	**66** (0100)	**66** (0100)	**66** (0100)	**66** (0100)	**66** (0100) **	**60** (0001)	**66** (0100)
	0100	150 Mhz CPU	**60** (0110)	**66** (1100)	**60** (0100)	**66** (0100)	**60** (0110)	**60** (0110)	**60** (0110)	**60** (0110) **	**60** (0001)	**60** (0110)
	0101	167 Mhz CPU	**66** (0110)	**66** (1100)	**60** (0100)	**66** (0100)	**60** (0110)	**66** (0110)	**66** (0110)	**66** (0110) **	**60** (0001)	**66** (0110)
	0110	180 Mhz CPU	**60** (1000)	**66** (1100)	**60** (0100)	**66** (0100)	**60** (0110)	**66** (0110)	**60** (1000)	**60** (1000) **	**60** (0001)	**60** (1000)
	0111	Rsrvd	**ERR**	**66** (1100) **	**60** (0100) **	**66** (0100) **	**60** (0110) **	**66** (0110) **	**60** (1000) **	**ERR**	**ERR**	**ERR**
	1000– 1110	Rsrvd	**ERR**	**60** (0001)	**60** (0001)	**60** (0001)	**60** (0001)	**60** (0001)	**60** (0001)	**ERR**	**ERR**	**ERR**
	1111	No card pres.	**ERR**	**66** (1100)	**60** (0100)	**66** (0100)	**60** (0110)	**66** (0110)	**60** (1000)	**ERR**	**ERR**	**ERR**

░	Indicates a non-optimized system configuration. One of the two CPU slots is populated with a faster CPU card than is in the other slot. The clock control logic will operate both CPUs at the slower of the two CPU clock speeds. One CPU will not be operating at full rated speed.
▓	Indicates an unsupported configuration on one of the slots. The clock control logic will set the bus clock frequency to 60 Mhz and configure the CPU internal clock for 1:1 mode (PLL @ 4x). Software should disable the slot that is reporting the reserved PD bit value (see Processor Enable Register, Port 0871h).
ERR	Indicates an error condition. The clock control logic sets the CPU bus clock frequency to 60MHz and configure the CPU internal clocks for 1:1 mode.
**	Indicates that one of the CPU cards is using reserved PD[0:3] pattern 0111. The clock control logic sets the bus clock frequency and CPU_FREQ_CFG[0:3] lines as shown. Developers may wish to use this PD code for experimental devices.

7.2.2 Frequency Select PAL Equations

```
TITLE    Zapatos Frequency Selection Logic
PATTERN  Frqsel1.PDS
REVISION 0
COMPANY  ibm
DATE     09/20/95

CHIP   frqsel1  PAL20L8

;------------------------- PIN Declarations ---------- 28plcc PIN
;PIN             NC                               ;             1
PIN  1           PID0_0                           ; INPUT       2
PIN  2           PID0_1                           ; INPUT       3
PIN  3           PID0_2                           ; INPUT       4
PIN  4           PID0_3                           ; INPUT       5
PIN  5           PID1_0                           ; INPUT       6
PIN  6           PID1_1                           ; INPUT       7
;PIN             NC                               ; INPUT       8
PIN  7           PID1_2                           ; INPUT       9
PIN  8           PID1_3                           ; INPUT       10
PIN  9           ID0                              ; INPUT       11
PIN  10          ID1                              ; INPUT       12
PIN  11          ID2                              ; INPUT       13
;PIN  12         GND                              ;             14
;PIN             NC                               ;             15
PIN  13          SPARE0                           ; INPUT       16
PIN  14          ID3                              ; INPUT       17
PIN  15          FRQID0           COMBINATORIAL   ; OUTPUT      18
PIN  16          FRQID1           COMBINATORIAL   ; OUTPUT      19
PIN  17          FRQID2           COMBINATORIAL   ; OUTPUT      20
PIN  18          SEL0             COMBINATORIAL   ; OUTPUT      21
;PIN             NC                               ;             22
PIN  19          SEL0_NA          COMBINATORIAL   ; OUTPUT      23
PIN  20          SEL0_NB          COMBINATORIAL   ; OUTPUT      24
PIN  21          FASTBUS_         COMBINATORIAL   ; OUTPUT      25
PIN  22          FRQID3           COMBINATORIAL   ; OUTPUT      26
PIN  23          FRQOVR_                          ; INPUT       27
;PIN 24          VCC                              ;             28

;------------------------------- Boolean Equation Segment ------

FRQID0.TRST = VCC
FRQID1.TRST = VCC
FRQID2.TRST = VCC
FRQID3.TRST = VCC
FASTBUS_.TRST=VCC
SEL0.TRST = VCC
SEL0_NA.trst = VCC
SEL0_NB.trst = VCC

/FASTBUS_  =  /ID0* /ID2* ID3* FRQOVR_
             +/ID0* /ID1* ID3* FRQOVR_

/FRQID0    =   ID0
            + /ID2* /ID3
            +  ID2*  ID3
            + /ID0*  ID1* /ID2*  ID3
            + /FRQOVR_

/FRQID1    =   ID0
            + /ID0*  ID1*  ID2
            + /ID0* /ID1* /ID2* /ID3
            + /FRQOVR_

/FRQID2    =   ID0
            +  ID2
            + /ID0* /ID1
            + /FRQOVR_

/FRQID3    =  /ID0* ID1* ID2* FRQOVR_
            + /ID0*/ID2* ID3* FRQOVR_
            + /ID0*/ID1* ID2* FRQOVR_
            + /ID0* ID2*/ID3* FRQOVR_
```

```
/SEL0    =                         /PID0_3*/PID1_0*/PID1_1*/PID1_2* PID1_3
         + /PID0_0* PID0_1*                /PID1_0*/PID1_1
         + /PID0_0* PID0_1*/PID0_2*        /PID1_0* PID1_1* PID1_2
         + /PID0_0*/PID0_1*         PID0_3*/PID1_0*/PID1_1*/PID1_2* PID1_3
         + /PID0_0*/PID0_1* PID0_2* PID0_3*/PID1_0*/PID1_1*/PID1_2* PID1_3
         + /SEL0_NA
         + /SEL0_NB

/SEL0_NA = /PID0_0*/PID0_1*/PID0_2*/PID0_3*/PID1_0*/PID1_1*         PID1_3
         + /PID0_0*/PID0_1*/PID0_2*/PID0_3*/PID1_0*         PID1_2*/PID1_3
         + /PID0_0*/PID0_1*/PID0_2*/PID0_3*/PID1_0* PID1_1*/PID1_2
         + /PID0_0* PID0_1* PID0_2* PID0_3*/PID1_0* PID1_1* PID1_2*/PID1_3
         + /PID0_0*/PID0_1* PID0_2* PID0_3*/PID1_0*/PID1_1* PID1_2*/PID1_3
         + /PID0_0* PID0_1* PID0_2*        /PID1_0* PID1_1*/PID1_2
         + /PID0_0* PID0_1*/PID0_2* PID0_3*/PID1_0* PID1_1*/PID1_2*/PID1_3

/SEL0_NB  = PID0_0* PID0_1* PID0_2* PID0_3*/PID1_0*/PID1_1*         PID1_3
          + PID0_0* PID0_1* PID0_2* PID0_3*/PID1_0*         PID1_2*/PID1_3
          + PID0_0* PID0_1* PID0_2* PID0_3*/PID1_0* PID1_1*/PID1_2
```

7.2.3 Clock Freezing

Some of the CPU clocks generated by the master PLL may not be required. For example, if no L2 is installed in the system, then the PCI_CLK[5:1] outputs of the MPC970 will not be required. The ability to stop (freeze) the unused clocks is useful to reduce run-time power consumption and EMI emissions.

The MPC970 can freeze any set of output clocks in the low state, while allowing the other clocks to continue running. The freeze command is given to the MPC970 via a two wire synchronous serial interface that originates in the system EPLD. See the clock freeze register description in Section 10 for more information on activating this feature.

Firmware can read the presence detect bits of the L2 and CPU card slots to determine which slots have devices present. For card slots which are not populated, firmware can disable the BUS_CLKs for those slot.

This function can also be used by power management functions to further reduce power consumption of the motherboard by freezing the clocks of devices which have been placed in a power managed mode.

7.3 PCI Clocks

The PCI clocks are generated by an AMCC™ 4403B PLL zero-delay clock generator, which divides the CPU clock by two and synchronizes the output clocks to the CPU seed clock. Five of the PCI clocks are used on the motherboard, and one PCI_CLK is provided for each of the two PCI expansion slots.

To minimize PCI to PCI clock skew, the PCI clock lengths have been made as equal as was feasible. Physical design of these traces follows the same rules as were used for the CPU clock traces. Also see Section 18, Physical Design, for more information.

7.3.1 PCI Bus speed Selection

The CPU:PCI bus clock ratio can be set to either 2:1 or 1:1.

- To make the CPU:PCI ratio 2:1, install a jumper between pin 2 and pin 3 of J30.
- To make the CPU:PCI ratio 1:1, install a jumper between pin 1 and pin 2 of J30. This mode has not been tested and is not recommended.

7.4 ISA Bus Clock

The ISA bridge derives the ISA SYSCLK from the PCI clock by dividing it by 4 or 3. This is controlled by the ISA Clock Divisor Register within the SIO chip. For PCI bus speeds of 33 and 30 Mhz, a divide by 4 should be selected to give ISA clock frequencies of 8.25 MHz and 7.5 MHz respectively.

7.5 Oscillators, Crystals, and Clocks

Table 41 shows the various sources of frequency generation in the system.

Table 41. Supported DRAM Modules

Clock	Source	Frequency
Reference crystal to MPC970 (60 Mhz CPU bus clock reference)	Y13 or Y8 crystal	15 MHz
Reference clock to MPC970 (66MHz CPU bus clock reference)	M10 oscillator	16.5 MHz
SCSI Clock reference for NCR53C810	Y1 oscillator	40 MHz
Ethernet clock reference for AM79C970	Y19 crystal	20 MHz
ISA OSC clk and 8254 timer reference for the ISA bridge	Y6 oscillator	14.3181 MHz
PC87332 SuperI/O baud rate reference for serial ports	Y9 oscillator	24 MHz
Reference crystal for Business audio subsystem CS4232 controller	Y2 crystal	24.576 MHz
Reference crystal for Business audio subsystem CS4232 controller	Y12 crystal	16.934 MHz
Reference crystal for Business audio subsystem YMF289S	Y11 or Y15 crystal	33.8688 MHz
Reference clock for bus clock shutdown interface. EPLD, MPC970	Y7 oscillator	1.8432 MHz
Reference crystal for power management controller	Y4 crystal	16 MHz
Reference crystal for RTC circuit DS1385	Y3 crystal	32.768 kHz
Reference crystal for Keyboard/Mouse controller	Y5 crystal	12 MHz

Section 8
PCI Bus

The reference design provides two 5.0v or 3.3v, 33MHz, 32 bit, PCI expansion slots via the riser card (see Section 14 for riser card information). The PCI bus implementation on the reference design is provided by the 660 Bridge, and it features:

- *PCI Local Bus Specification* version 2.0 and 2.1 compliant.
- PCI bus frequency up to 33 MHz. PCI bus runs synchronously to the CPU bus at one-half or the same frequency.
- PCI to DRAM access with L1 and L2 cache snooping.
- Support for the PowerPC Reference Platform memory map of CPU address space into PCI transactions:
  - PCI I/O reads and writes.
  - PCI memory reads and writes.
  - PCI configuration reads and writes.
  - PCI interrupt acknowledge reads.
- Supports ISA bus master access to system memory with ISA bridge on the PCI bus.
- Supports contiguous ISA I/O and non-contiguous ISA I/O mapping.
- Supports PCI resource locking of system memory.
- Supports type 0 and type 1 PCI configuration cycles.
- 5v tolerant I/O.

The SCSI controller, the Ethernet controller, the Multi-Processor Interrupt Controller (MPIC), and the ISA bridge also reside on the PCI bus.

Section 8.1 and Section 8.2 discuss the response of the reference design to transactions initiated by a PCI busmaster. See Section 2, CPU Bus, for information on accessing the PCI bus from the CPU.

Succeeding sections discuss PCI bus arbitration, configuration transaction, and other PCI topics.

For information on PCI interrupts and MPIC, see Section 6, Exceptions. Also see the MPIC data sheet.

For information the Ethernet and SCSI controllers, see Section 11, I/O Systems. Also see the controller data sheets.

For more information on the ISA bridge, see Section 9, ISA and X-Bus. Also see the ISA bridge data sheet.

8.1 PCI Transaction Decoding

When a PCI bus master initiates a transaction on the PCI bus, the transaction either misses and is master aborted, or it is claimed by a PCI target. The target can be either the 660 Bridge or another PCI target.

8.1.1 PCI Transaction Decoding By Bus Command

Table 42 shows the responses of the 660 Bridge and other agents to various PCI bus transactions initiated by a PCI bus master other than the 660 Bridge. As shown in Table 42, the 660 Bridge ignores (No response) all PCI bus transactions except PCI memory read and write transactions, which it decodes as possible system memory accesses.

Table 42. Reference Design Responses to PCI_C[3:0] Bus Commands

C 3:0	PCI Bus Command	Can a PCI Bus Master-Initiate this Transaction?	660 Bridge Response to the Transaction	Can Another PCI Target Claim the Transaction?
0000	Interrupt Acknowledge	No. Only the 660 Bridge is allowed to initiate.	None	Yes. The ISA bridge is intended to be the target.
0001	Special Cycle	Yes	None	Yes
0010	I/O Read	Yes	None	Yes
0011	I/O Write	Yes	None	Yes
0100	Reserved	No. Reserved	None	n/a
0101	Reserved	No. Reserved	None	n/a
0110	Memory Read	Yes	System memory read	Yes, if no address conflict.
0111	Memory Write	Yes	System memory write	Yes, if no address conflict.
1000	Reserved	No. Reserved	None	n/a
1001	Reserved	No. Reserved	None	n/a
1010	Configuration Read	No. Only 660 Bridge.	None	Yes
1011	Configuration Write	No. Only 660 Bridge.	None	Yes
1100	Memory Read Multiple	Yes	System memory read	Yes, if no address conflict.
1101	Dual Address Cycle	Yes	None	Yes
1110	Memory Read Line	Yes	System memory read	Yes, if no address conflict.
1111	Memory Write and Invalidate	Yes	System memory write	Yes, if no address conflict.

8.1.2 PCI Memory Transaction Decoding By Address Range

When a PCI bus master transaction is decoded by bus command as a system memory read or write, the 660 Bridge checks the address range. Table 43 shows the mapping of PCI bus master memory accesses to system memory. This is the mapping that the 660 Bridge uses when it decodes the bus command to indicate a system memory access.

Unless the IGN_PCI_AD31# signal is asserted, PCI memory accesses in the 0 to 2G address range are ignored by the 660 Bridge. There is no system memory access, no snoop cycle, and the 660 Bridge does not claim the transaction. When the IGN_PCI_AD31# signal is asserted, the 660 Bridge maps PCI memory accesses from 0 to 2G directly to system memory at 0 to 2G. PCI memory accesses from 2G to 4G are mapped to system memory from 0 to 2G.

Table 43. Mapping of PCI Memory Space, Part 1

PCI Bus Address	Other Conditions	Target Cycle Decoded	Target Address	Notes
0 to 2G	IGN_PCI_AD31# Deasserted	Not Decoded	N/A	No Response.
	IGN_PCI_AD31# Asserted	System Memory *	0 to 2G	Snooped by caches.
2G to 4G		System Memory *	0 to 2G	Snooped by caches.

Note:
*Memory does not occupy this entire address space. Accesses to unoccupied space are not decoded.

PCI memory accesses that are mapped to system memory cause the 660 Bridge to claim the transaction, access system memory, arbitrate for the CPU bus, and broadcast a snoop operation on the CPU bus. A detailed description of the snoop process is presented in the 660 Bridge User's Manual.

Table 44 gives a more detailed breakdown of the reference design response to PCI memory transactions in the 0 to 2G range. Note that the preferred mapping of PCI memory is from 16M to 1G–2M so that it can be accessed by both the CPU and by PCI bus masters.

Table 44. Mapping of PCI Memory Space, Part 2

PCI Bus Address	Target Resource	System Memory Address	Snoop Address
2G to 4G	System memory (1)	0 to 2G	0 to 2G
1G–2M to 2G	Reserved (2)	No system memory access. The 660 Bridge ignores PCI memory transactions in this range.	No snoop.
16M to 1G–2M	PCI Memory		
0 to 16M	PCI/ISA Memory (3)		

Notes:

1) The 660 Bridge maps PCI bus master memory transactions in the 2G to 4G range to system memory, and the CPU is unable to initiate PCI memory transactions to this address range, so do not map devices to this PCI memory address range.

2) The CPU (through the 660 Bridge) can not access the 1G–2M to 2G address range, so do not map PCI devices herein unless the CPU will not access them.

3) Transactions initiated on the PCI bus by the ISA bridge on behalf of an ISA bus master only (IGN_PCI_AD31# asserted for an SIO), are forwarded to system memory and broadcast snooped to the CPU bus from 0 to 16M. If this is not an ISA bus master transaction, then the 660 Bridge ignores it. Note that the 660 Bridge will also forward PCI transactions from 16M to 2G if IGN_PCI_AD31# is asserted during an ISA-bridge-mastered transaction, and that this capability is not normally used.

8.1.3 PCI I/O Transaction Decoding

The 660 Bridge initiates PCI I/O transactions on behalf of the CPU. Other PCI bus masters are also allowed to initiate PCI I/O transactions. Table 45 shows the reference design mapping of PCI I/O transactions. The 660 Bridge ignores PCI I/O transactions.

PCI/ISA I/O is mapped to PCI I/O space from 0 to 64K. The ISA bridge subtractively decodes these transactions (and PCI memory transactions from 0 to 16M). Other devices may actively decode and claim these transactions without contention.

PCI I/O is assigned from 16M to 1G–8M.

Table 45. Mapping of PCI Master I/O Transactions

PCI Bus Address	Target Resource	Other System Activity
1G–8M to 4G	Reserved (1)	The 660 Bridge ignores I/O transactions initiated by PCI bus masters.
16M to 1G–8M	PCI I/O devices	
8M to 16M	Reserved (1)	
64K to 8M	Reserved (2)	
0 to 64K	PCI/ISA I/O	

Notes:

1) The CPU (through the 660 Bridge) can not access this address range, so do not map PCI devices herein unless the CPU will not access them.

2) In contiguous mode, the CPU (through the 660 Bridge) can create PCI I/O addresses in the 64K to 8M range. In non–contiguous mode, the CPU can only access PCI addresses from 0 to 64K.

8.1.4 ISA Master Considerations

Since the reference design implements IGN_PCI_AD31# and uses an Intel SIO, memory transactions produced on the PCI bus by the ISA bridge on behalf of an ISA master are forwarded to system memory at the corresponding address (0 —16M).

If ISA masters are utilized and the SIO is programmed to forward their cycles to the PCI bus, then no other PCI device (e.g., video) is allowed to be mapped at the same addresses because contention would result.

The SIO chip contains registers to control which ranges of ISA addresses are forwarded to the PCI bus.

ISA masters cannot access any PCI memory.

For more information on the handling of ISA bus master operations, see the 660 Bridge User's Manual and the SIO data book.

8.2 PCI Transaction Details

Details of the reference design implementation of various PCI transactions, including sequencing, timing, and interactions with the CPU bus, are found in the 660 Bridge User's Manual.

PCI bus masters are not able to access the boot ROM, the BCRs in the 660 Bridge, or the CPU bus.

8.2.1 Bus Snooping on PCI to Memory Cycles

Each time a PCI (or ISA) bus master accesses memory, (and once again for each time a PCI burst crosses a cache block boundary) the 660 Bridge broadcasts a snoop operation on the CPU bus. If the CPU signals an L1 snoop hit by asserting ARTRY#, the 660 Bridge retries the PCI transaction. The ISA bridge then removes the grant from the PCI agent, who (according to PCI protocol) releases the bus for at least one cycle and then arbitrates again. Meanwhile, the 660 Bridge grants the CPU bus to the CPU, allowing it to do a snoop push. Then the PCI agent again initiates the original transaction.

During the transaction, the 660 Bridge L2 cache is monitoring the memory addresses. The L2 takes no action on L2 misses and read hits. If there is an L2 write hit, the L2 marks that block as invalid, does not update the block in SRAM, and does not affect the PCI transaction. L2 operations have no effect on PCI to memory bursts.

8.2.2 PCI Peer to PCI Peer Transactions

Peer to peer PCI transactions are supported consistently with the memory maps of Table 42, Table 43, Table 44, and Table 45, which, together, show the range of different bus command transactions that are supported. The SIO can not perform PCI memory transactions to a PCI peer because all SIO memory transactions are mapped to system memory.

8.2.3 PCI to System Memory Transactions

PCI to system memory transactions are described in detail in the 660 Bridge User's Manual.

Single and burst transfers are supported. Bursts are supported without special software restrictions. That is, bursts can start at any byte address and end on any byte address and can be of arbitrary length. Also, the arbitration logic insures that the PCI does not monopolize the PCI bus.

As per the PCI specification, the byte enables are allowed to change on each data phase. This has no practical effect on reads, but is supported on writes. The memory addresses linearly increment by 4 on each beat of the PCI burst. All PCI devices must use only linear burst incrementing.

Table 46 shows which CAS# lines are activated when a PCI master writes memory. Note that CAS[0]# refers to byte addresses 0 mod 8, CAS[1]# refers to byte addresses 1 mod 8, etc. For read cycles, eight bytes of memory data are read on each access, but the master receives only the desired 4 bytes. The bytes are read or written to memory independently of BE or LE mode (the endian mode byte swappers are situated between the CPU and the rest of the system, not between the PCI and the rest of the system).

In ECC mode, PCI to memory transactions that result in less than 8-byte writes, cause the memory controller in the 660 Bridge to execute a read-modify-write operation, during which

8 bytes of memory data are read, the appropriate bytes are modified, the ECC byte is modified, and then the resulting 8 bytes are written to memory.

Table 46. Active CAS# Lines – PCI to Memory Writes, BE or LE Mode

PCI_ AD[2]	Byte Enables BE[ ]#				Column Address Selects CAS[ ]#							
	3	2	1	0	0	1	2	3	4	5	6	7
0	1	1	1	1								
0	1	1	1	0	X							
0	1	1	0	1		X						
0	1	1	0	0	X	X						
0	1	0	1	1			X					
0	1	0	1	0	X		X					
0	1	0	0	1		X	X					
0	1	0	0	0	X	X	X					
0	0	1	1	1				X				
0	0	1	1	0	X			X				
0	0	1	0	1		X		X				
0	0	1	0	0	X	X		X				
0	0	0	1	1			X	X				
0	0	0	1	0	X		X	X				
0	0	0	0	1		X	X	X				
0	0	0	0	0	X	X	X	X				
1	1	1	1	1								
1	1	1	1	0					X			
1	1	1	0	1						X		
1	1	1	0	0					X	X		
1	1	0	1	1							X	
1	1	0	1	0					X		X	
1	1	0	0	1						X	X	
1	1	0	0	0					X	X	X	
1	0	1	1	1								X
1	0	1	1	0					X			X
1	0	1	0	1						X		X
1	0	1	0	0					X	X		X
1	0	0	1	1							X	X
1	0	0	1	0					X		X	X
1	0	0	0	1						X	X	X
1	0	0	0	0					X	X	X	X

Note:
X = active. Blank = inactive. Byte enables normally represent contiguous addresses. This table shows all cases.

8.3 Bus Arbitration Logic

The reference design uses the Intel SIO as the PCI bus arbiter. The PCI arbiter sees the 660 Bridge as one of several PCI agents. The order of priority for PCI arbitration is programmable, and is initially set to be:

1. (CPUREQ) – 660 Bridge (the SIO normally parks the bus on the 660 Bridge)
2. (REQ1) – PCI slot 2
3. (SIOREQ) – SIO
4. (REQ0) – SCSI
5. (REQ2) – Ethernet
6. (REQ3) – PCI Slot 1.

For more information on arbitration, see the Registers section and the PCI Arbitration Controller section of the SIO data book. See the reference design schematics for the connection of the various PCI requests and grants.

There can be concurrency of cycles on the ISA bus (caused by DMA or ISA masters) with PCI or CPU transactions as long as the ISA bus operations are not forwarded to the PCI bus. Forwarding of ISA bus operations must wait for the ISA bridge to grant the PCI bus access to its ISA interface.

8.4 PCI Configuration Transactions

The preferred method for generating PCI configuration cycles is via the 660 Bridge indexed Bridge Control Registers (BCR). This configuration method is described in Section 4 of the 660 User's Manual. The IDSEL assignment and their respective PCI_AD lines are shown here in Table 47. The addresses used for configuration are assigned as shown.

Table 47. 660 Bridge Address Mapping of CPU Bus Transactions

Device	ID Sel Line	60X Address*	PCI Address
82378ZB PCI to ISA Bridge	A/D 11	8080 08XXh	0080 08XX
53C810 SCSI Controller	A/D 12	8080 10XXh	0080 10XX
PCI Slot 1 (lower)	A/D 13	8080 20XXh	0080 20XX
PCI Slot 2 (upper)	A/D 14	8080 40XXh	0080 40XX
Multiprocessor Interrupt Controller	A/D 15	8080 80XXh	0080 80XX
79C970 Ethernet Controller	A/D 16	8081 00XXh	0081 00XX

Note:
* This address is independent of contiguous I/O mode.

8.4.1 650 Bridge Compatible Method

If it is not possible to use indexed BCRs to generate PCI configuration cycles, they can be generated by an alternate method known as the 650 Bridge compatible method. CPU accesses to the address range 2G+8M to 2G+16M cause the Bridge to arbitrate for the PCI bus and then to execute a type 0 PCI configuration transaction as described in the PowerPC Reference Platform Specification and implemented by the IBM27-82650 PowerPC to PCI Bridge. This is referred to as the 650 compatible configuration method.

8.5 PCI Bus Loading

The reference design motherboard presents 11 to 13 loads to the PCI bus. Two of these, the FLASH/ROM and IDSEL selects, are resistively coupled as recommended in the PCI

2.0 specification to reduce loading. Of the remaining 11 loads, 2 are devices in PCI slots. The remaining 9 loads consist of connectors and PCI devices on the motherboard and riser card. The motherboard and riser card have been designed to allow the timing requirements for 33MHz operation as defined in PCI rev 2.1 if PCI compliant adaptors are installed in both PCI slots. See the Section 18 for more information.

Section 9
ISA and X-Bus

The reference design provides two standard ISA expansion slots via the riser card (see Section 14 for riser card information). The ISA bus is the tertiary or I/O bus for the reference design. It is controlled and interfaced to the PCI bus via an ISA bridge. The reference design uses a buffered subset of the ISA bus, called the X-bus, to interface to various onboard devices.

The Native (Super) I/O controller and the Business Audio controller reside on the ISA bus. For information on these controllers, see Section 11, I/O Systems, and the controller data sheets.

Sections 9.1 through 9.5 describe the operation of the ISA bus.

Section 9.6 gives a functional description of the X-bus.

Section 9.7 describes the X-bus keyboard/mouse controller and registers.

Section 9.8 describes the X-bus real time clock (RTC) and registers.

Section 9.9 describes the X-bus NVRAM and registers.

Section 9.10 describes the X-bus presence detect registers.

For information on X-bus registers that are internal to the system EPLD, see Section 10, EPLD.

For a complete listing of the ISA/X-bus registers, see Section 13, Registers.

For more information on ISA interrupts and MPIC, see Section 6, Exceptions. Also see the MPIC data sheet.
Also see the Intel™ 82378ZB data book.

9.1 The ISA Bridge

The ISA bridge function is provided by an Intel™ 82378ZB chip (SIO). It provides a PCI to ISA bus bridge, with the following major functions:

- Bridge between PCI and ISA
    - 8/16 bit ISA devices
    - 24 bit addressing on ISA
    - Partially decodes native I/O addresses
    - Unclaimed PCI memory address below 16MB forwarded to the ISA bus
    - Unclaimed PCI I/O address below 64K forwarded to the ISA bus
    - Powers up to an open condition (i.e., cycles may be passed to the ISA bus)
    - Generates ISA clock, with a programmable divide ratio of three or four
    - Allows ISA mastering and has programmable decodes that map ISA memory cycles to the PCI bus
    - 32-bit posted memory write data buffer (no I/O buffering)
- Seven channel ISA DMA controller
    - Function of two 83C37s with 32-bit extensions
    - Supports 8-bit or 16-bit devices on the ISA bus
    - Supports 32-bit addressing for ISA to PCI memory transfers
    - 8-byte bidirectional buffer for DMA data
- Timer block (function of 82C54)
- Interrupt Controller (function of two 8259A's)
- PCI bus arbiter
- Functions as PCI target during programming and ISA target cycles, and as bus master during DMA or ISA master cycles
- Generates ISA_REFRESH# signal to refresh ISA bus DRAM.

9.2 Address Ranges

The ISA bus address ranges which may be separately enabled in the ISA bridge for forwarding to the PCI bus are:

- 0 - 512K
- 512K - 640K
- 640K - 768K
- 768K - 896K (in 8 ranges of 16K each)
- 896K - 960K
- 1M - xM (where x < or = 16) with the hole (see the ISA bridge data book). The hole may be 64K or 8M. Also see Section 2.3 for contiguous and non-contiguous address mapping of this space from the CPU memory space.

If an ISA DMA produces an address in the 0-16M range and this address is enabled in the ISA bridge for forwarding to the PCI, the ISA bridge will initiate a PCI transaction which the

660 bridge will, in all cases (see Section 9.4), forward to system memory. Do not enable other PCI agents to claim PCI memory transactions in this range.

The 660 bridge actively decodes and claims these ISA master originated cycles on the PCI bus. It does not use subtractive decoding. Any other PCI agent that attempts to claim these transactions will create a bus contention. Thus ISA masters can only access other ISA devices or system memory. They may not access PCI devices.

The software must not map any PCI memory at PCI addresses which ISA masters can create (those addresses from 0 to 16M which are programmed for forwarding from ISA to PCI). In that case, an ISA transaction forwarded to the PCI bus would be claimed both by the 660 Bridge and by the PCI memory device. Alternatively stated, ISA masters are not allowed to create accesses to system memory using any address from 0 to 16M that is mapped to a PCI device, such as video. Also see Section 9.4.

9.3 ISA Bus Concurrency

ISA bus cycles which are not enabled for forwarding, including the hole, remain on the ISA bus. That is, DMA or ISA bus master cycles on the ISA bus can run concurrently with PCI or CPU cycles.

9.4 ISA Busmasters and IGN_PCI_AD31

The ISA bridge supports ISA bus masters. System memory accesses from an ISA bus master are designed to be mapped to the 0 to 16M range, and the ISA bridge forwards them to the PCI bus at the same range, which is not compliant to the PowerPC Reference Platform specification. Other PCI to system memory accesses, however, are compliantly mapped to the 2G to 4G range (for system memory address range from 0 to 2G). In some architectures this problem is handled by using the ISA_MASTER# signal, which is active during the ISA bus master operation.

However, the ISA bridge allows ISA masters to run posted writes to system memory without latching in the accompanying ISA_MASTER# signal. In this situation, the ISA_MASTER# signal is no longer synchronized to the ISA bus master operation.

To overcome this challenge, the 660 Bridge does not use the ISA_MASTER# signal. Instead, it detects PCI memory transactions that are initiated by the ISA bridge. The reference design ANDs together GNT0#, GNT1#, GNT2#, and GNT3# (all of the PCI grants except the ISA bridge internal grant) to generate IGN_PCI_AD31, which is active high during the address phase of any PCI transaction that is not initiated by one of the four possible PCI agents (besides the 660 bridge and the ISA bridge). If the PCI transaction was not initiated by the 660 bridge, and if it is a memory transaction, then the 660 bridge knows that it is a system memory transaction initiated on the PCI bus by the SIO. It then assumes that the transaction is on behalf of an ISA bus master, and so forwards it to the correct system memory address in the 0 to 16M range.

As a consequence of this design, the ISA bridge must be programmed to map ISA DMA (that is bound for system memory) to a PCI memory transaction using the 0 to 2G address range, rather than the apparently correct 2G to 4G range. Since the DMA-sourced PCI transaction also causes IGN_PCI_AD31 to be asserted during the address phase of a PCI transaction initiated by the ISA bridge, the 660 bridge will not do the usual inversion of the highest order address bit, but will forward the transaction to system memory in the 0 to 2G range.

Another consequence of the design is that the ISA bridge can not initiate peer to peer PCI memory transactions, because no matter what the PCI address is, it will be claimed by the 660 bridge (if the address is that of a populated memory location) and mapped to system memory, possibly causing various inappropriate results. Also see Section 9.2.

These are the only limitations on the normal operation of the ISA bridge that are caused by the IGN_PCI_AD31 design, and there are no implications for other PCI or ISA agents, which are totally unaffected by the situation.

9.5 DMA

The DMA controller in the ISA bridge consists of two 82C37A DMA controllers with 32-bit addressability extensions and enhanced functionality. The DMA request/grant lines are connected on the reference design as shown in Table 48.

Table 48. DMA Assignments

DMA Channel	Assignment or Connection
0	ISA slots, Audio (note 2), EPP (note 1)
1	ISA slots, Audio (note 2), EPP (note 1)
2	ISA slots, Floppy Disk Controller (SuperI/O)
3	ISA slots, Audio (note 2), EPP (note 1)
4	Cascade in
5	ISA slots, EPP (note 1)
6	ISA slots
7	ISA slots

Notes:

1. The Enhanced Parallel Port (EPP) feature of the National 87332 SuperI/O chip can be mapped to DMA channel 0, 1, 3, or 5 using the control mechanism provided in Port 87C, the System Control Port of the system EPLD.

Port 87Cbits [2:0]

0	x	x	No DMA
1	0	0	Channel 0
1	0	1	Channel 1
1	1	0	Channel 3
1	1	1	Channel 5

2. The Audio DMA channels are selected within the Crystal CS4332 controller. The DMA channels are assigned as follows:

CS4232 DMA Channel	System DMA Channel
A	0
B	1
C	3

9.5.1 Supported DMA Paths

DMA operations can be performed only:

- From ISA I/O mapped devices to ISA memory mapped devices, and
- From ISA I/O mapped devices to system memory (via the PCI bus). In these transfers, the system memory address must be mapped to the PCI address range 0 to 2G (see Section 9.4).

The DMA source device can be located on the ISA or X-bus. If the DMA target is ISA memory mapped, it can also reside on the X-bus.

- ISA DMA to PCI I/O devices is not allowed.
- ISA DMA to PCI memory devices is not allowed.
- ISA DMA from ISA memory mapped devices is not allowed.

9.5.2 DMA Timing

The DMA controller runs compatible cycles for all ISA to ISA DMA transfers. Type A, type B and type F timing is available only for ISA I/O to system memory (via the PCI) DMA transfers.

9.5.3 Scatter-Gather

The reference design permits the use of independent scatter-gather (SG) operations on DMA channels 0-3 and 5-7. This operation chains together a number of DMA transfers to different memory locations so that they appear as one DMA transfer. The SG command, descriptor table, and status registers are relocatable via a configuration register in the ISA bridge. The termination of an operation may be signaled to the software by configuring any (or all) of the SG channels to signal end of process (aka Terminal Count) to the DMA device. It is also possible to use IRQ13 to signal the end of DMA, but this is not recommended. See Section 6.1.5.1.

9.6 X-Bus

The X-bus is a utility bus, an 8 bit buffered subset of the ISA bus that is implemented on the reference board to support motherboard native I/O devices and motherboard registers. The address range of the X-bus is a subset of the ISA bus address range. Figure 44 shows the ISA bridge, a PCI agent which sources the ISA and X-busses. The X-bus data transceiver is controlled by the ISA bridge via XDIR and XDEN#. The ISA bridge also generates the X-bus control signals and partial ISA bus address decode signals, which the system EPLD uses to decode the address strobes for ISA registers.

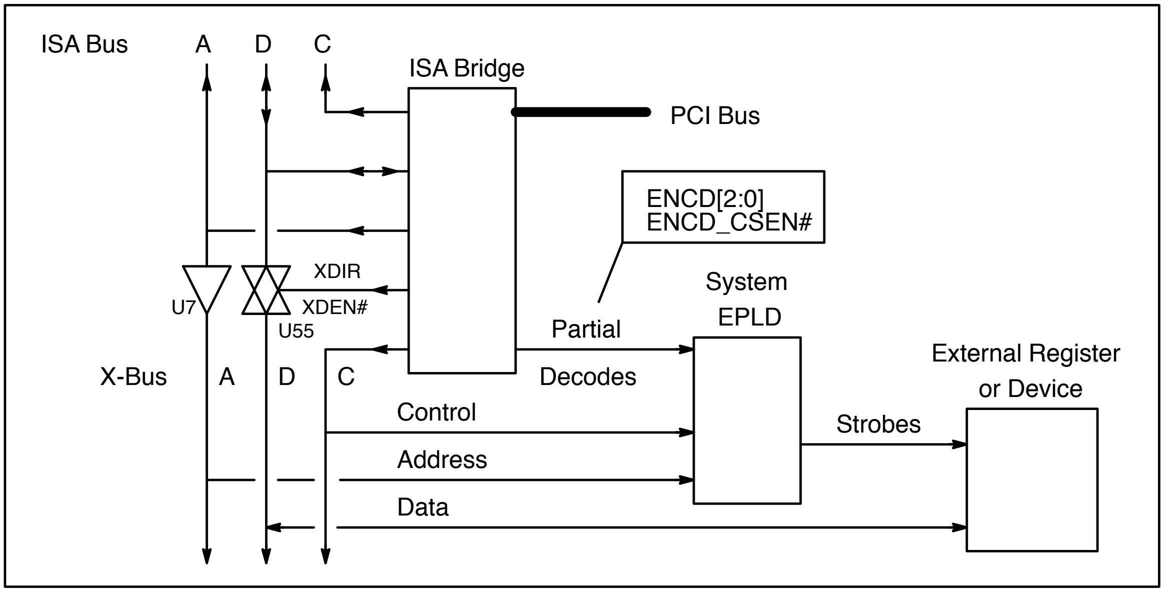


Figure 44. Typical X-Bus Registers

Reference design X-bus registers are located in the following areas:

• Keyboard/mouse controller	See Section 9.7.
• Real time clock	See Section 9.8.
• NVRAM	See Section 9.9.
• System EPLD	See Section 10.
• Motherboard discrete logic	See Section 9.10.

Additional ISA bus registers are located in the Business Audio and Super I/O controllers. Certain ISA bridge registers are also mapped to ISA space.

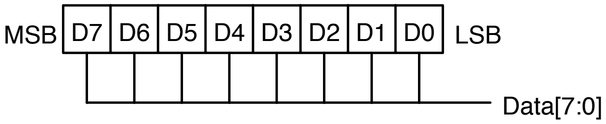
9.7 Keyboard/Mouse Controller

The reference design uses an Intel 8042AH as a keyboard and mouse controller. It resides on the X-bus. The code used is the same version as is used in IBM Personal System/2 machines. This microcode may differ from other 8042-type keyboard controllers. These differences are usually only significant when porting AIX to the system (for more information contact your IBM technical representative). See the controller data sheet for more information. This device contains several registers.

9.7.1 Keyboard/Mouse Control Registers

ISA Port 0060	Read/Write	Reset to n/a
ISA Port 0062	Read/Write	Reset to n/a
ISA Port 0064	Read/Write	Reset to n/a
ISA Port 0066	Read/Write	Reset to n/a

This register set is located inside the keyboard/mouse controller. See the keyboard/mouse controller data sheet for details. The system EPLD asserts KYBD_CS# to access these ports.



9.8 Real Time Clock (RTC)

The reference design uses a Dallas Semiconductor™ DS1385S to provide the real time clock (TOD or RTC) function. This device is PC compatible and resides on the X-bus. It features an additional 4K of NVRAM and a replaceable battery. This device contains several registers. See the data sheet for more information.

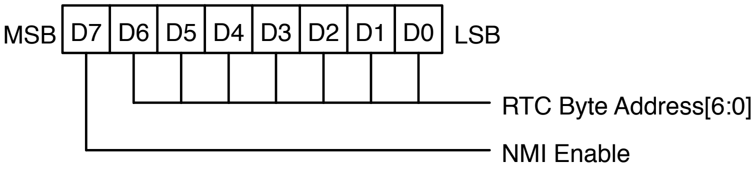
The RTC section of the DS1385S contains 64 8-bit registers. There are 50 bytes of user RAM, 10 bytes of time, calender, and alarm data, and 4 bytes of control and status information. Access to RTC registers can be accomplished by writing to the following registers in the given sequence:

- Write the byte address to port 0070.
- Read/write the data from/to port 0071.

9.8.1 RTC Address and NMI Enable Register

ISA Port	0070	Write Only	Reset to n/a

This register is shared by the RTC and the ISA bridge.



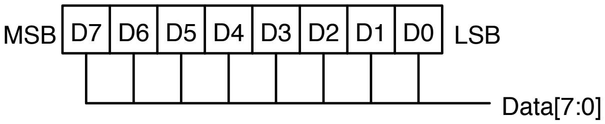
Bit 6:0 . RTC Internal byte address (1 of 64). The SIO ignores these bits during writes and does not drive them during reads. The SIO decodes accesses to this port internally.

Bit 7 . . . NMI Enable. The SIO uses this bit to enable its NMI output. The RTC ignores the bit on writes and does not drive the bit on reads. The system EPLD asserts RTC_ALE to the RTC to access these bits of the port.

9.8.2 RTC Data Register

ISA Port	0071	Read/Write	Reset to n/a

This register is used to access the registers within the RTC. Reads and writes to this port access the internal RTC register pointed to by the RTC Address and NMI Enable register. This register is located inside the RTC. The system EPLD asserts RTC_WR# to write to this register, and RTC_RD# to read the register.



9.9 Non-Volatile RAM (NVRAM)

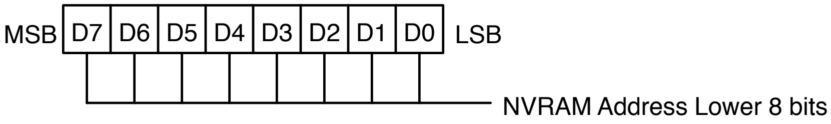
The system NVRAM is locate inside the RTC package. The DS1385S has 4Kx8 of SRAM. The NVRAM is accessed by the following sequence of operations:

- Write the lower 8 bits of the NVRAM internal address to port 0074.
- Write the upper 4 bits of the NVRAM internal address to port 0075.
- Read/write the data from/to port 0077.

9.9.1 NVRAM Address Register Low

ISA Port	0074	Write Only	Reset n/a

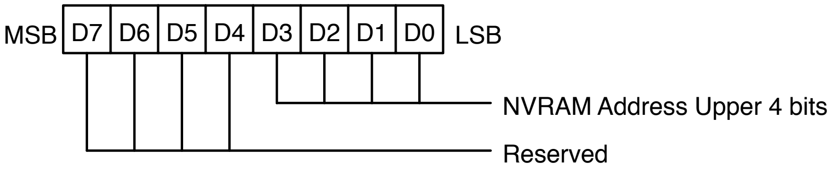
This register contains the lower 8 bits of the 12 bit NVRAM internal address. This register is located inside the RTC. The system EPLD decodes accesses to this register by asserting AS0#.



9.9.2 NVRAM Address Register High

ISA Port	0075	Write Only	Reset n/a

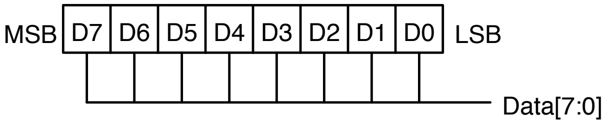
This register contains the lower 8 bits of the 12 bit NVRAM internal address. This register is located inside the RTC. The system EPLD decodes accesses to this register by asserting AS1#.



9.9.3 NVRAM Data Register

ISA Port	0077	Read/Write	Reset to n/a

This register is used to access the registers within the NVRAM. Reads and writes to this port access the internal NVRAM byte pointed to by the NVRAM address registers. This register is located inside the RTC. The system EPLD asserts NVRAM_WR# to write to this register, and NVRAM_RD# to read the register.



9.10 Motherboard Presence Detect Registers

The following registers are located on the motherboard. They are generally implemented as shown in Figure 44 and Figure 45.

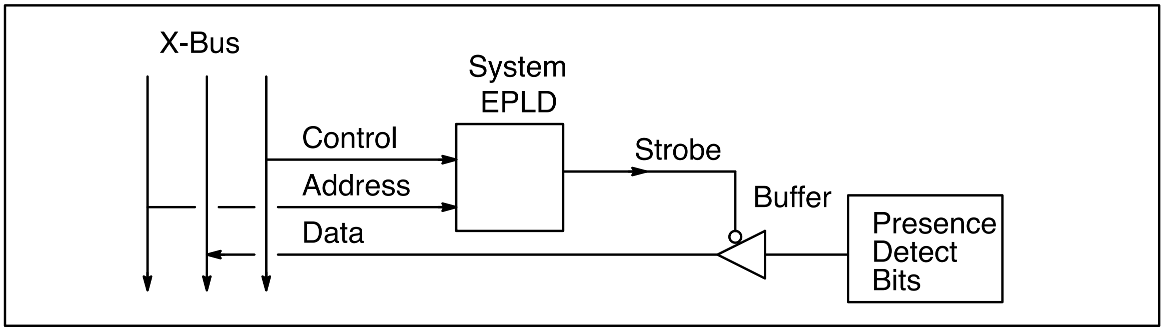


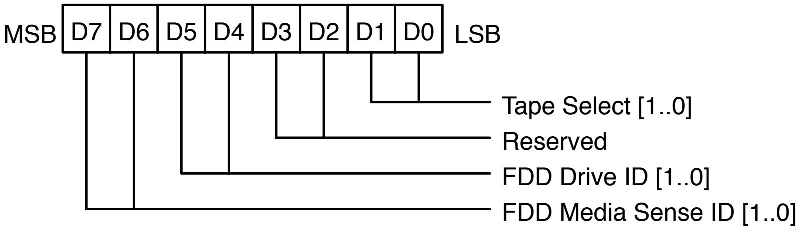
Figure 45. Typical X-Bus Registers

9.10.1 Floppy Media Sense ID

ISA Port	03F3	Read Only	Reset n/a

The reference design uses U27 to buffer the Floppy Media Sense bits from the floppy drive connector onto the X-bus SD[7:4]. The system EPLD asserts RD_3F3# to read this register.

The SuperI/O chip also decodes reads to 03F3, and should be configured to respond only on bits SD[3:0].



Bit 1:0 Tape Select [1:0] — Driven by the FDC in the SuperI/O.

Bit 3:2 Reserved

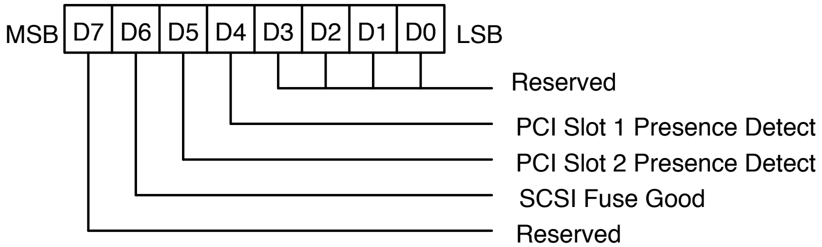
Bit 5:4 Floppy Drive ID [1:0] — driven by U27.

Bit 7:6 Floppy Media Sense [1:0] — driven by U27.

9.10.2 Equipment Presence Register

ISA Port	080C	Read Only	Reset n/a

The reference design uses U23 to buffer the equipment presence detect bits onto the X-bus. The system EPLD asserts EQP_PRSNT_RD# to read this register.



Bit 4 . . . PCI Slot 1 PD bit. 0 means there is a PCI device in slot 1.
Bit 5 . . . PCI Slot 2 PD bit. 0 means there is a PCI device in slot 2.
Bit 6 . . . SCSI Fuse Good. 1 means the fuse is intact.
Bit 7 . . . Reserved.

9.10.3 L2 ID Register

ISA Port	080D	Read Only	Reset n/a

The reference design supports a 182 pin tag/SRAM module in the L2 slot. The L2 ID register (U23) is used to buffer the L2 slot presence detect bits onto the X-bus. The system EPLD asserts CACHE_PD_RD# to read this register. Table 49 shows the encoding of these bits.

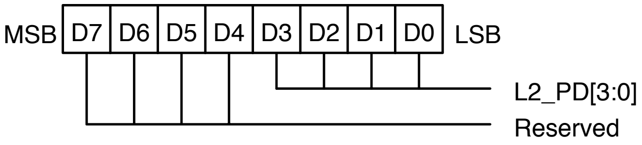


Table 49. L2 Cache SRAM/TagRAM PD Table

PD3	PD2	PD1 *	PD0 *	Description		
0	0	0	0	Burst	256K	No Parity
0	0	0	1	Burst	512K	No Parity
0	0	1	0	Burst	1M	No Parity
0	0	1	1	Reserved		
0	1	0	0	Burst	256K	Parity
0	1	0	1	Burst	512K	Parity
0	1	1	0	Burst	1M	Parity
0	1	1	1	Reserved		
1	0	0	0	Async	256K	No Parity
1	0	0	1	Async	512K	No Parity
1	0	1	0	Async	1M	No Parity
1	0	1	1	Reserved		
1	1	0	0	Async	256K	Parity
1	1	0	1	Async	512K	Parity
1	1	1	0	Async	1M	Parity
1	1	1	1	No L2 card or serial EEPROM		

Note:
* When a serial ROM is installed, PD[1:0] become IDS_DATA and IDS_CLK.

9.10.4 Motherboard ID Register

ISA Port	0852	Read Only	Reset n/a

The reference design uses U29 to buffer the motherboard ID bits onto the X-bus. The system EPLD asserts PLANAR_ID_RD# to read this register.

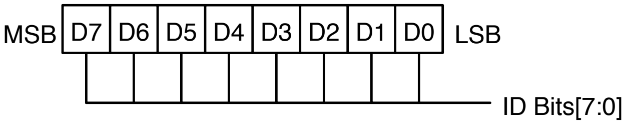


Table 50. Motherboard ID Encoding

Motherboard ID[7:0]	Schematic
5A	PowerPC 604 SMP Reference Design—Preliminary
other	Reserved

9.10.5 CPU ID Registers

ISA Port	0866	Read Only	Reset n/a
ISA Port	0867	Read Only	Reset n/a

The reference design supports two CPU slots. The CPU ID registers are used to read the presence detect bits of these slots. The system EPLD asserts PROC1_PD_RD# to enable U28 to drive the PD bits from CPU slot 1 onto the X-bus. The system EPLD asserts PROC2_PD_RD# to enable U56 to drive the PD bits from CPU slot 2 onto the X-bus.

A shown in Table 51, the CPU_PD bits report information about the CPU, and the CPU_L2_PD bits report information about any L2 which might be on the CPU card. These L2 PD bits are not the same as the L2 PD bits associated with the L2 slot.

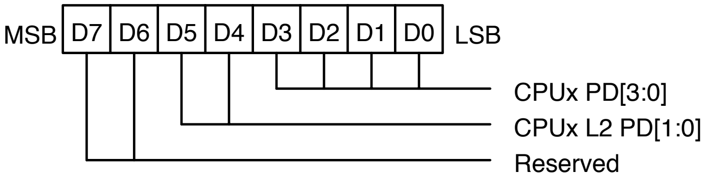


Table 51. CPU_PD Bits

CPU_PD0	CPU_PD1	CPU_PD2	CPU_PD3	CPU_L2_ PD0	CPU_L2_ PD1	Description
0	0	0	0	x	x	No CPU
0	0	0	1	x	x	100 MHz CPU
0	0	1	0	x	x	120 MHz CPU
0	0	1	1	x	x	132 MHz CPU
0	1	0	0	x	x	150 MHz CPU
0	1	0	1	x	x	167 MHz CPU
0	1	1	0	x	x	180 MHz CPU
0111 thru 1110				x	x	Reserved
1	1	1	1	x	x	No CPU card present.
x	x	x	x	0	0	Reserved
x	x	x	x	0	1	TBD
x	x	x	x	1	0	TBD
x	x	x	x	1	1	Serial ROM or no L2.

Note:
For information on the L2 serial ROM option, see Section 2.8.1.

9.10.6 DRAM ID Registers

ISA Port	0880 thru 0883	Read Only	Reset n/a

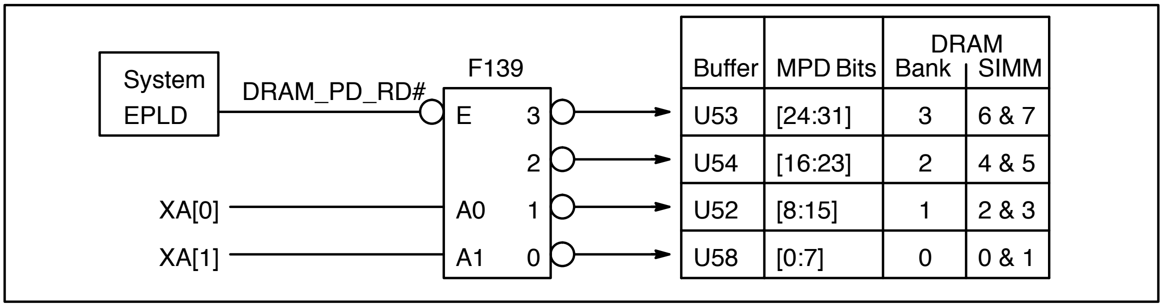
The reference design supports 72 pin, 4-byte wide industry standard parity DRAM modules (SIMMs). The DRAM ID registers are buffers that drive the SIMM presence detect bits onto the X-bus. Figure 46 shows how the registers are accessed. Table 52 shows the bit fields that correspond to each SIMM. Table 53 shows the encoding of the SIMM ID bits.

MSB	D7	D6	D5	D4	D3	D2	D1	D0	LSB

See Table 52.

Table 52. DRAM PD Registers

Bank	Buffer	SIMM	MPD[]	Port	Bits
0	U58	0	3:0	0880	3:0
		1	7:4		7:4
1	U52	2	11:8	0881	3:0
		3	15:12		7:4
2	U54	4	19:16	0882	3:0
		5	23:20		7:4
3	U53	6	27:24	0883	3:0
		7	31:28		7:4



Buffer	MPD Bits	DRAM Bank	DRAM SIMM
U53	[24:31]	3	6 & 7
U54	[16:23]	2	4 & 5
U52	[8:15]	1	2 & 3
U58	[0:7]	0	0 & 1

Figure 46. DRAM PD Registers

Table 53. SIMM Definition

SIMM ID [3:0]	SIMM Type	SIMM Speed
1111	Absent	—
1111	8 M	60 ns
1110	16 M	60 ns
1101	32 M	60 ns
1100	4 M	60 ns
1011	8 M	70 ns
1010	16 M	70 ns
1001	32 M	70 ns
1000	4 M	70 ns

To distinguish between no SIMM present and an 8M, 60ns SIMM, a write-read verify test can be performed.

Section 10
System EPLD

The System EPLD (EPLD) supports registers and logic which are needed by the system and are not contained in any other device. It provides glue logic, internal control registers, and control signals to access external control registers. Most applications will use a customized version of the EPLD to handle their unique requirements.

The EPLD is a programmed Altera™ EPM5130QC100 electrically programmable logic device. For timing and electrical specifications, see that data sheet.

The EPLD supports both internal (contained in the EPLD) and external (contained in other devices) registers.

10.1 External Registers

The EPLD supports a group of external registers, which are latches or other devices that are physically located in a device other than the EPLD. As shown in Figure 47, the EPLD supplies control signals to the external registers, based on a decode of the address and control lines of the X-bus (or ISA bus) and the encoded chip selects from the ISA bridge. The ISA bridge must be programmed to perform this function. In response to the signals from the EPLD, the external register either reads or writes data to the X-bus.

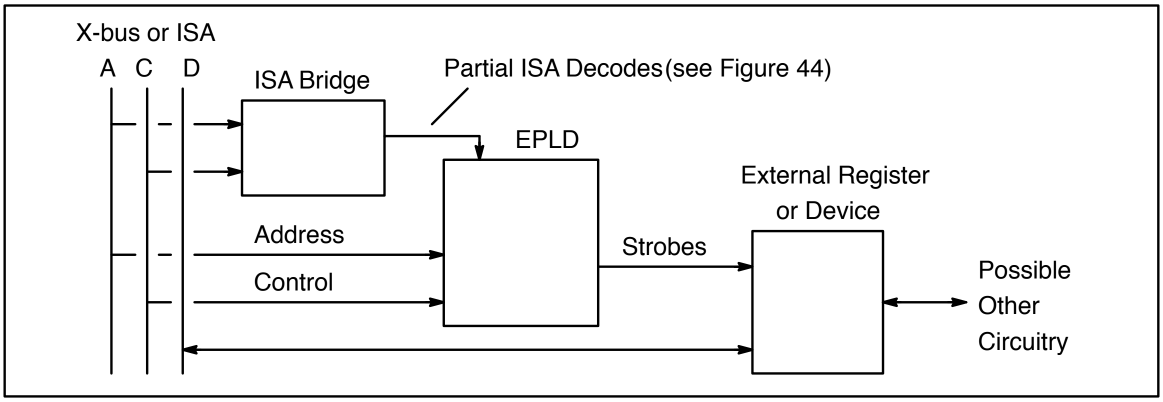


Figure 47. Typical External Register

For the external registers that the EPLD supports, Table 54 shows the external register, the ISA I/O port address, and the supplied control signal(s). Section 9.10, Motherboard Presence Detect Registers, contains descriptions of these registers.

Table 54. External Register Support

ISA Port Address	Register	Read/ Write	Register Location	Signal	Strobe or Function
0060 or 0062 or 0064 or 0066	Keyboard Controller Registers	R/W	Keyboard Controller	KYBD_CS# (asserted for an access to any of these addresses)	Address Decode
0070	RTC Address Latch Enable	W/O	RTC	RTC_ALE	Write strobe
0071	RTC Data	Write	RTC	RTC_WR#	Write strobe
		Read		RTC_RD#	Read strobe
0074	NVRAM Address Low Byte	W/O	NVRAM	RTC_AS0#	Address Decode
0075	NVRAM Address High Byte	W/O	NVRAM	RTC_AS1#	Address Decode
0077	NVRAM Data	Write	NVRAM	NVRAM_WR#	Write Strobe
		Read		NVRAM_RD#	Read Strobe
03F3	Floppy Media Sense	R/O	FDC U27	RD_3F3#	Read Strobe
080C	Equipment Present Reg.	R/O	Board U23	EQP_PRSNT_RD#	Read Strobe
080D	L2 ID Register	R/O	Board U23	CACHE_PD_RD#	Read Strobe
0852	Motherboard ID Register	R/O	Board U29	PLANAR_ID_RD#	Read Strobe
0866	CPU 1 ID Register	R/O	Board U28	PROC1_PD_RD#	Read Strobe
0867	CPU 2 ID Register	R/O	Board U56	PROC2_PD_RD#	Read Strobe
0880	DRAM ID Registers 0	R/O	Board U58	DRAM_PD_RD#	Read Strobe
0881	DRAM ID Registers 1	R/O	Board U52	DRAM_PD_RD#	Read Strobe
0882	DRAM ID Registers 2	R/O	Board U54	DRAM_PD_RD#	Read Strobe
0883	DRAM ID Registers 3	R/O	Board U53	DRAM_PD_RD#	Read Strobe

10.2 Internal Registers

The EPLD contains a group of internal registers, which are accessed via the ISA bus I/O port address shown for each register (see Table 55).

Table 55. Internal Register Support

ISA Port Address	Register	Access	Note
0808	Storage Light Register	Read/Write	
081C	System Control Register	Read/Write	1
082A	Power Management Control Register 1	Read/Write	
082B	Power Management Control Register 2	Read/Write	
0838	IRQ13 Interrupt Request Active Register	Read/Write	
0860	Freeze Clock Register Low	Write	
0862	Freeze Clock Register High	Write	
0868	Serial ROM Control Register	Read/Write	
086B	L2 Control Register	Write	
0870	CPU Sequence Register	Read	
0871	CPU Enable Register	Write	

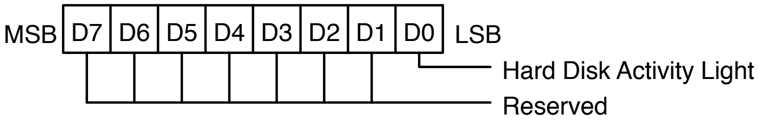
Note:

1. Except bits 2:0 (W/O). This register is shared with the 660 Bridge System Control 81C BCR.

10.2.1 Storage Light Register

ISA Port	0808	Read/Write	Reset to xxxx xxx0

This register controls the HDD_LED# output of the EPLD. This signal normally controls the hard disc drive activity LED.

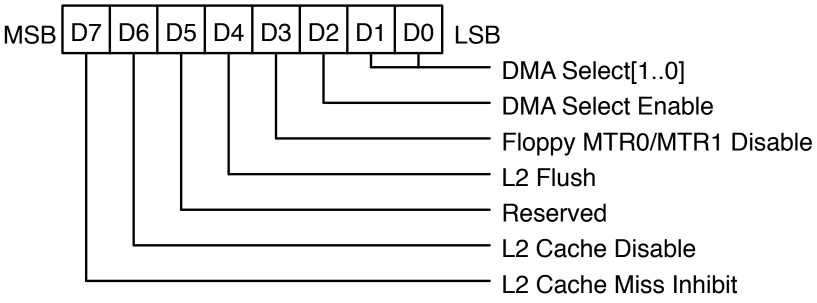


Bit 0 Hard Disk Activity Light:
0 = Turn light off (negate HDD_LED#).
1 = Turn light on (assert HDD_LED#).

10.2.2 System Control Register 081C

ISA Port 081C	Read/Write	Reset to 00x0 0000

Access to this register can be affected by the configuration of the IBM27-82660 Bridge controllers. See the System Control 81C BCR section of the 660 Bridge User's Manual for details.



Bit 1:0 DMA Select[1:0] – These bits select which DMA channel is used by the ECP function. These bits are write only. See Table 56.

Bit 2 DMA Select Enable – This bit enables or disables the ECP DMA function. If this bit is 0, then no DMA channel is allocated to the parallel port. This bit is write only. See Table 56.

Table 56. External Register Support

Bit 2	Bit 1	Bit 0	ECP DMA Channel	ECPA#	ECPB#	ECPC#	ECPD#
0	x	x	None	1	1	1	1
1	0	0	0	0	1	1	1
1	0	1	1	1	0	1	1
1	1	0	3	1	1	0	1
1	1	1	5	1	1	1	0

The ECPx# outputs are asserted continuously, and are used on the motherboard to enable tristate buffers that do the actual routing of the DMA request.

Bit 3 Floppy Controller Motor Select Disable – This bit enables/disables the MTR0# and MTR1# motor select signals driven by the floppy disk controller. Disabling these signals allows the user to read media sense information without spinning up the drive motor.
0 = Enable MTR0# and MTR1# (assert FD_MTR_EN#)
1 = Disable MTR0# and MTR1# (negate FD_MTR_EN#).

Bit 4 L2 Flush – If there is an external lookaside L2 in the system, this bit can be used to flush it.
0 = Drive L2_FLUSH# low.
1 = Drive L2_FLUSH# high.

Bit 5 Reserved.

Bit 6 L2 Cache Disable – This bit will disable an external cache but will not invalidate the contents. While disabled, the L2 will not respond to any cycles.
0 = L2 Cache Disable (assert L2_CACHE_DIS#).
1 = L2 Cache Enable (negate L2_CACHE_DIS#).

Bit 7 L2 Cache Miss Inhibit – This bit is intended to prevent L2 misses from updating an external L2. This allows the external L2 to retain its contents during memory accesses and remain coherent.
0 = Prevent updates (assert L2_CACHE_INH#)
1 = Normal operation (negate L2_CACHE_INH#).

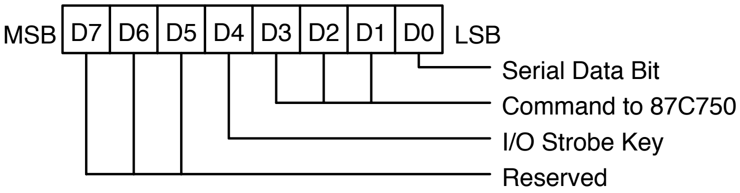
10.2.3 Power Management Control Registers

These registers are part of the power management (PM) system. The PM system is not supported on the reference design. PM information is included for evaluation purposes only.

10.2.4 Power Management Control Register 1

ISA Port	082A	Read/Write	Reset to xxxx xxxx

This register is part of the power management (PM) system..



Bit 0 Data Bit – This bit is the serial data to and from the 83C750 power management controller (PMC).

Bit 3:1 PMC Command [2:0] **–** PMC command

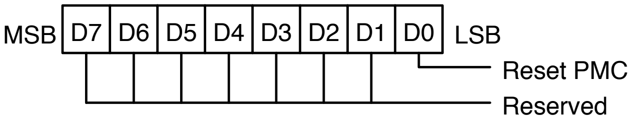
Bit 4 I/O Strobe – This bit strobes the serial data in and out of the 83C750 PMC.

Bit 7:5 Reserved

10.2.5 Power Management Control Register 2

ISA Port	082B	Read/Write	Reset to 0xxx 0xx0

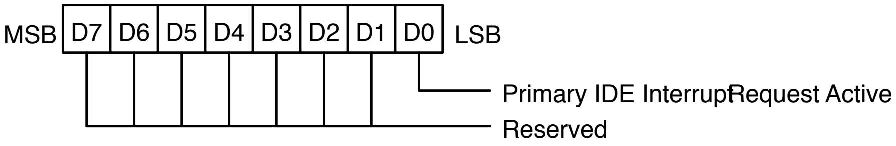
This register is part of the power management system.



Bit 0 Reset Power Management Controller – This bit is used by POST to guarantee the state of the PM Controller on every Power-On Reset.
0 = Writing a 0 to this bit causes the EPLD to negate 83CX_RESET.
1 = While this bit is a 1, the EPLD asserts 83CX_RESET to the PMC. It is the responsibility of the host to hold the reset long enough to meet the specification of the PM Controller. This bit is not reset to ‘0’ by RESET# signal.

10.2.6 IRQ13 Interrupt Request Active Register (Not Supported)

ISA Port 0838	Read/Write	Reset to xxxx xxx0



This register exists to allow power management systems and primary IDE to share one system interrupt. Power management and IDE are both unsupported on the reference design. This information is for evaluation purposes only.

Bit 0 Primary IDE Interrupt Request Active – When read, this bit indicates status of the primary IDE interrupt request. When an external primary IDE device has driven the system interrupt request signal active, this bit is set to 1 and remains set to 1 until software clears the condition by setting the bit to 0 or by the RESET# signal. Writing a 1 to this bit has no effect.

Bit 7:1 Reserved

10.2.7 Freeze Clock Registers

The EPLD contains a unidirectional synchronous serial data link for writing to the MPC970 master PLL (see Section 7). The freeze clock register (FCR) is a 13 bit register that is accessed by the system via the X-bus as two 8-bit registers. The FCR Low register (ISA Port 0860) contains the low 8 bits of the register, and the FCR High register (ISA Port 0862) contains the upper 5 bits.

To use the registers, first write the low register, and then the high register. Writing the high register triggers a write operation. The FCR data in ports 0860 and 0862 is shifted out as a serial data stream on FRZ_DATA. Initially, FRZ_DATA will be set to zero and then it will start shifting13 times. At the end of the shifting, FRZ_DATA will stay at 1.

After triggering the data transfer, wait at least 10 usec. for the transfer to complete before accessing the FCR or retriggering the data transfer.

A 0 in a bit position of the port 0860 or 0862 instructs the MPC970 to freeze the corresponding clock output of the MPC970 and a 1 in that bit position instructs the MPC970 to unfreeze the clock output.

On the reference design, this data transfer is clocked by a 1.843MHz oscillator. For details of the data transfer operation and the meaning of the data, see the MPC970 data sheet.

10.2.7.1 Freeze Clock Register (FCR) Low

ISA Port 0860	Write Only	Reset to FFh

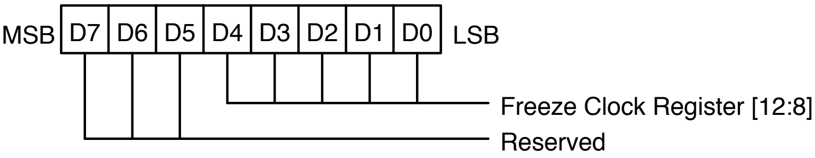
MSB	D7	D6	D5	D4	D3	D2	D1	D0	LSB

Freeze Clock Register[7:0]

Bit	Reference Design Clock Consumer
0	CPU Slot 1_2 (BCLK 2)
1	CPU Slot 2_2 (BCLK 2)
2	664 BCLK
3	CPU Slot 2_0 (BCLK 0)
4	CPU Slot 2_1 (BCLK 1)
5	CPU Slot 1_0 (BCLK 0)
6	CPU Slot 1_1 (BCLK 1)
7	L2 Slot (Tag) (Tag BCLK) See section 10.2.7.2.

10.2.7.2 Freeze Clock Register (FCR) High

ISA Port	0862	Write Only	Reset to xxx1 1111



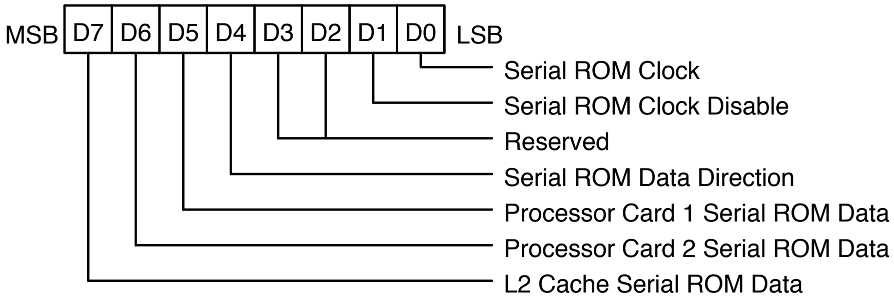
Bit	Reference Design Clock Consumer
0	SRAM_BCLK0
1	SRAM_BCLK1
2	SRAM_BCLK2
3	SRAM_BCLK3
4	Spares (PALs) (MC_BCLK_SPARE)

Bit 7:5 Reserved

10.2.8 Serial ROM Control Register

A serial communications protocol is defined for the L2 and CPU slots that allows an L2 or CPU card to substitute a serial ROM for the PD bits. This allows the card to transmit much more configuration data to to the system. The serial ROM control register is used on the system side to implement the protocol.

ISA Port 0868	Read/Write	Reset to 0000 xx10



Bit 0 Serial ROM Clock – This bit sets the state (non-inverted) of the SER_CLK output. The clock is toggled manually by setting this bit high and low. While the clock is 'running,' maintain the frequency from 0 to 500kHz.

Bit 1 Serial ROM Clock Disable – The serial clock shares a net with the PD bits of the CPU and L2 cards.
0 = Assert SER_CLK_EN# to enable external buffers to drive the serial clock onto the PD0 lines of the CPU and L2 slots.
1 = Negate SER_CLK_EN# to disable the external buffers.

Bit 3:2 Reserved

Bit 4 Serial ROM Data Direction – The serial ROM data bit I/Os share PD lines with the CPU and L2 cards. In input mode, these I/Os do not interfere with the PD bits.
0 = Input – Direction of data flow is in to the EPLD from the serial ROM.
1 = Output – Direction of data flow is out from the EPLD to the serial ROM.

Bit 5 CPU Card 1 Serial ROM Data Bit – This is a bidirectional signal.
Output = This bit sets the state (non-inverted) of the PC1_DATA output.
Input = This bit reports the state (non-inverted) of the PC1_DATA input.

Bit 6 CPU Card 2 Serial ROM Data Bit – This is a bidirectional signal.
Output = This bit sets the state (non-inverted) of the PC2_DATA output.
Input = This bit reports the state (non-inverted) of the PC2_DATA input.

Bit 7 L2 Cache Serial ROM Data Bit – This is a bidirectional signal.
Output = This bit sets the state (non-inverted) of the L2_DATA output.
Input = This bit reports the state (non-inverted) of the L2_DATA input.

10.2.8.1 Serial ROM Communications Protocol

The serial communication protocol used by the L2 and CPU card ID ROM is defined as follows. One start bit followed by an eight bit control byte then a returned eight bits of data. The control byte consists of a two bit command (1,0 for reads), a four bit address A[3:0], and two don't care bits (x,x). The four bit address selects which of the 16 bytes to read.

Start	1	0	A3	A2	A1	A0	X	X	D7	D6	D5	D4	D3	D2	D1	D0

The start bit consists of a falling edge on SDA while SCLK is held high. With the exception of the start bit, all transitions on SDA occur while SCLK is low. The read cycle is driven on the bus as shown in Figure 48.

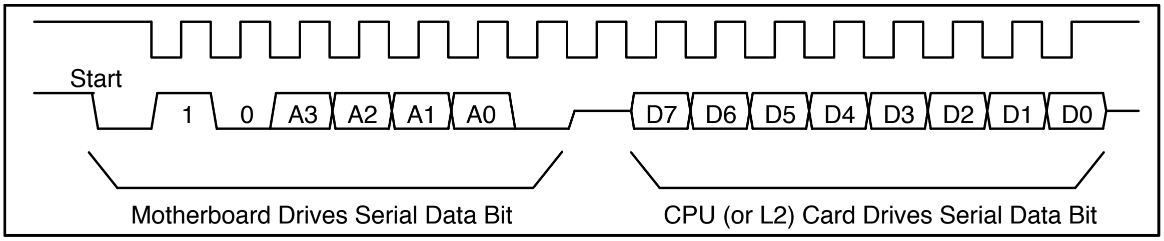

Start
1
0
A3
A2
A1
A0
D7
D6
D5
D4
D3
D2
D1
D0
Motherboard Drives Serial Data Bit
CPU (or L2) Card Drives Serial Data Bit

Figure 48. Read Cycle On the Bus

10.2.8.2 Serial ROM vs. PD bits

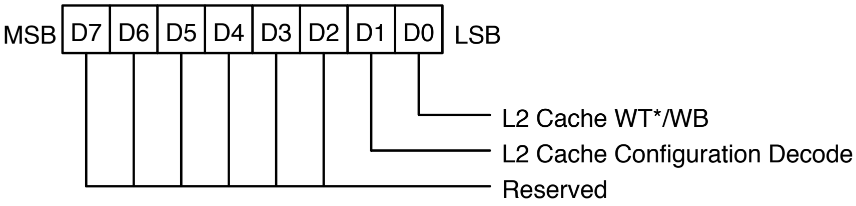
The serial ROM clock shares a net with L2_PD[0] from each CPU slot and SRAM_PD[0] of the L2 slot. The serial ROM clock enable controls external buffers that gate the clock onto these lines. The same enable controls all three buffers (see SER_CLK_EN# on the schematics).

When not reading or testing for the presence of a serial ROM, leave the serial ROM clock disabled, and leave the serial ROM data bit I/Os configured as inputs.

The PD bits on the cards are implemented with pulldown and pullup resistors (e.g., 10k pullup and 100 ohm pulldown) instead of by shorting the lines to ground or V_{CC}. When reading the serial ROM(s), enable the clock and read the ROM quickly, then disable the clock. In this way, the electronics of cards that have implemented are not subject to increased current draw for long enough for significant heating to occur.

10.2.9 L2 Control Register

ISA Port 086B	Write Only	Reset to xxxx xx10



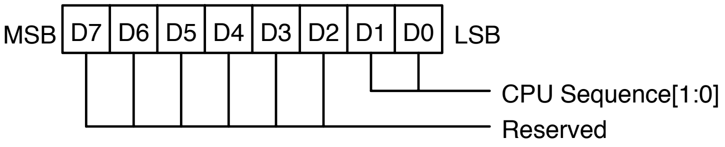
Bit 0 L2 Cache WT/WB
0 = L2 Cache must operate in write through mode (L2_WT# is asserted)
1 = L2 Cache is allowed to operate in write back mode (L2_WT# is negated).

Bit 1 L2 Cache Configuration Decode
0 = L2 Cache Configuration is enabled (L2_CONF_DCD# is asserted)
1 = L2 Cache Configuration is disabled (L2_CONF_DCD# is negated)

Bit 7:2 Reserved

10.2.10 CPU Sequence Register

ISA Port 0870	Read Only	Reset to xxxx xx00



Bit 1:0 CPU Sequence bits[1:0]

Bit 7:2 Reserved

This register provides CPUs with a way to get a unique CPU number in an SMP system. Bits 1:0 form a binary counter.

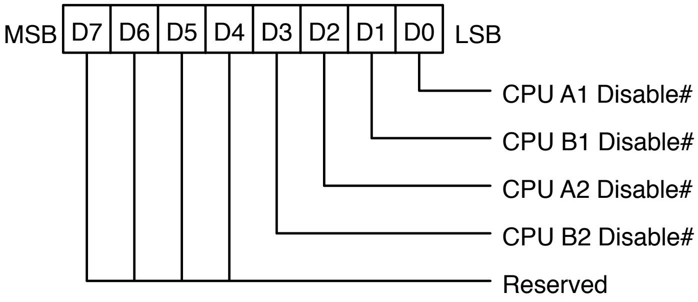
- The first time after reset that the counter is read, it reports 00b. At the end of the read, it increments to 01b.
- The next read returns 01b, and the counter post-increments to 10b.
- The next read returns 10b, and the counter post-increments to 11b.
- The next and all subsequent reads of this register return 11b.

The SEMIFOR_ST output of the EPLD is low leaving reset, and is normally low. It is asserted high while ISA I/O port 0870 is being accessed. SEMIFOR_ST is an active high decode of the register access.

10.2.11 CPU Enable Register

ISA Port	0871	Read/Write	Reset to xxxx 1111

Setting a given CPU disable bit low (active) forces the HRESET PAL to assert HRESET# to the specified CPU. Setting the bit high (inactive) allows the HRESET PAL to negate HRESET# to that CPU. Setting all four bits low allows the HRESET PAL to negate HRESET to all of the CPUs. Setting all four bits low while RESET# is deasserted causes the EXTHRST# output to pulse low.



Bit 0	CPU A1 Disable # 1 = Set PROC_A1_DIS# high 0 = Set PROC_A1_DIS# low. See Note.
Bit 1	CPU B1 Disable # 1 = Set PROC_B1_DIS# high. 0 = Set PROC_B1_DIS# low. See Note.
Bit 2	CPU A2 Disable # 1 = Set PROC_A2_DIS# high. 0 = Set PROC_A2_DIS# low. See Note.
Bit 3	CPU B2 Disable # 1 = Set PROC_B2_DIS# high. 0 = Set PROC_B2_DIS# low. See Note.
Bit 7:4	Reserved

Note: When D[3:0] = 0000 during a write to this port, the EPLD will set all four PROC_x_DIS# outputs high.

10.2.11.1 External Hardware Reset

If D[3:0] = 0000 during a write to this port while the RESET# input is deasserted, then the EPLD will pulse the EXTHRST# output low. EXTHRST# is asserted for 1024 cycles of the PWR_MNG_CLK.

EXTHRST# is normally connected to the HRESET# circuit on the motherboard, such that it acts in parallel to an external reset push button.

10.3 Signal Descriptions

Table 57 shows the active signals of the EPLD. Pins not shown should not be connected.

Table 57. Signal Descriptions

Signal Name	Pin	I/O	Description
External Register Support			
CACHE_PD_RD#	8	O 6mA	**L2 presence detect read strobe.** EPLD asserts this signal to read X-bus port 080D.
DRAM_PD_RD#	90	O 6mA	**DRAM SIMM presence detect read enable 1.** EPLD asserts this signal to read X-bus ports 0880-0883.
EQP_PR_RD#	79	O 6mA	**Equipment present register read.** EPLD asserts this signal to read X-bus port 080C.
KYBD_CS#	52	O 6mA	**Keyboard chip select.** EPLD asserts this signal to read X-bus ports 0060, 0062, 0064, and 0066.
NVRAM_RD#	75	O 6mA	**NVRAM output enable (read strobe).** EPLD asserts this signal to read X-bus port 0077. This normally causes a read of the NVRAM data stored at the location contained in the NVRAM address register.
NVRAM_WR#	34	O 6mA	**NVRAM data write strobe.** EPLD asserts this signal to write to X-bus port 0077. This normally causes the data associated with this write to be written into the NVRAM location contained in the NVRAM address register.
PLANAR_ID_RD#	33	O 6mA	**Planar ID read.** EPLD asserts this signal to read X-bus port 0852.
PROC1_PD_RD#	7	O	**CPU 1 Card Present Detect Read:** EPLD asserts this signal to read X-bus port 0866.
PROC2_PD_RD#	6	O	**CPU 2 Card Present Detect Read:** EPLD asserts this signal to read X-bus port 0867.
RTC_ALE#	57	O 6mA	**Real time clock address latch enable.** EPLD asserts this signal to write X-bus port 0070.
RTC_AS0#	24	O 6mA	**NVRAM address register low byte write strobe.** EPLD asserts this signal to write to X-bus port 0074.
RTC_AS1#	23	O 6mA	**NVRAM address register high byte write strobe.** EPLD asserts this signal to write to X-bus port 0075.
RTC_RD#	32	O 6mA	**Real time clock read strobe.** EPLD asserts this signal to read X-bus port 0071.
RTC_WR#	31	O 6mA	**Real time clock write strobe.** EPLD asserts this signal to write to X-bus port 0071.

Table 57. Signal Descriptions (Continued)

Signal Name	Pin	I/O	Description
ISA and X-bus			
ECPA#	89	O	**Printer DMA request:** This signal enables external logic to route the ECP DMA Request to DMA channel 0.
ECPB#	86	O	**Printer DMA request:** This signal enables external logic to route the ECP DMA Request to DMA channel 1.
ECPC#	85	O	**Printer DMA request:** This signal enables external logic to route the ECP DMA Request to DMA channel 3.
ECPD#	84	O	**Printer DMA request:** This signal enables external logic to route the ECP DMA Request to DMA channel 5.
ECS[2:0]	59, 22, 21	I	**Encoded Chip Select [2:0].** Encoded chip selects for peripheral devices supported by the ISA I/O Bridge. Used by EPLD X-bus I/O port address decoders. From SIO.
ECSEN#	60	I	**Encoded Chip Select Enable.** Asserted to enable the base decoder. Negated to select the option decoder. Used by EPLD X-bus I/O port address decoders. From SIO.
ISA_CLK	26	I	**ISA Clock:** The ISA bus clock.
ROM_EN#	45	O	**ROM Chip Enable:** This signal is the chip select for the system boot ROM when it is located on the ISA bus.
XA[7:0]	61 ,64, 65, 66, 67, 70, 71, 72.	I	**X-address bus [7:0].** Used by EPLD X-bus I/O port address decoders.
XD[7:0]	73,92,95, 96,97,74, 98,99	I/O 24mA	**X-data bus [7:0].** Used by EPLD to transfer data.
XIOR#	9	I	**X-bus I/O Read.** Indicates that the system is reading from an X-bus I/O device. Used by EPLD X-bus I/O port address decoders.
XIOW#	16	I	**X-bus I/O Write.** Indicates that the system is writing to an X-bus I/O device. Used by EPLD X-bus I/O port address decoders.
Interrupts			
IRQ1_IN	14	I	**Keyboard interrupt.** EPLD is designed to intercept the keyboard interrupt between the keyboard and the ISA bus bridge. It is monitored in the suspend state as a wakeup indicator. Either connect IRQ1_SIO and IRQ1_IN as shown in the reference design, or disconnect both signals from the system (routing the keyboard interrupt to the ISA bus bridge).
IRQ1SIO	56	O	**Interrupt request 1.** Latched active when IRQ1_IN is asserted. Negated when KYBD_CS# is asserted.
IRQ12	15	I	**Interrupt request 12 input.** Connect to system IRQ12, the mouse interrupt. It is monitored in the suspend state as a wakeup indicator.

Table 57. Signal Descriptions (Continued)

Signal Name	Pin	I/O	Description
Floppy Disk Interface			
RD_3F3#	25	O	**Media Sense Read:** The EPLD asserts this signal during reads to ISA port 03F3. This signal is used to enable motherboard logic to drive SD[7:4]. The SuperI/O also internally decodes reads of this register and drives information onto SD[3:0]. See the SuperI/O data book.
FD_MTR_EN#	1	O	**Motor Enable:**This active low output pin is used to disable MTR0# and MTR1# signals from the SuperI/O chip, and is controlled by Port 081C, bit 3.
HDD_LED#	83	O 6mA	**Hard disk drive activity light.** EPLD asserts this signal while bit 0 of the storage light register (port 0808) is 1. This signal normally indicates hard disk drive activity.
L2 Cache Interface			
L2_CACHE_DIS#	81	O	**L2 cache disable:** This output can be used to disable an external L2. It is controlled by bit 6 of Port 081C.
L2_CACHE_INH#	76	O	**L2 cache inhibit:** This output can be used to inhibit an external L2. It is controlled by bit 7 of Port x‘081C’.
L2_CONF_DCD#	82	O	**L2 configuration decode:** This active low output is used to enable configuration mode in certain serial L2 caches.
L2_FLUSH#	51	O	**L2 cache flush:** This signal follows the value written to System Control BCR 0 (address 8000 081Ch bit 4): while bit 4 is 0, L2_FLUSH# will be asserted. It can be used to flush an external L2 cache.
L2_WT#	3	O	**L2 write through:** This active low output can be used to force an external L2 into write through only mode.
SMP Interface (see the CPU Sequence Register)			
HALT1	27	I	**Halt 1:** This active high signal indicates that a CPU on CPU card 1 is asserting HALT. Also see RUN.
HALT2	28	I	**Halt 2:** This active high signal indicates that a CPU on CPU card 2 is asserting HALT. Also see RUN.
PROC_A1_DIS#	35	O	**Processor A1 enable:** This active low output is used to force the assertion of HRESET# to CPU A in CPU card 1.
PROC_B1_DIS#	36	O	**Processor B1 enable:** This active low output is used to force the assertion of HRESET# to CPU B in CPU card 1.
PROC_A2_DIS#	39	O	**Processor A2 enable:** This active low output is used to force the assertion of HRESET# to CPU A in CPU card 2.
PROC_B2_DIS#	40	O	**Processor B2 enable:** This active low output is used to force the assertion of HRESET# to CPU B in CPU card 2.
RUN	2	O	**Run:** This active high output is routed to each CPU. RUN enables the CPU to snoop and conduct bus operations. The EPLD negates RUN only while HALT1 and HALT2 are both active. All of the CPUs in the system have to assert HALT for RUN to be negated.
SEMIFOR_ST#	46	O	**Semaphore status:** This normally high output pulses low during accesses to the CPU sequence register.

Table 57. Signal Descriptions (Continued)

Signal Name	Pin	I/O	Description
SMP Interface (see the CPU Sequence Register)			
Serial ROM Interface (see the Serial ROM Control Register)			
L2_DATA	53	I/O	**L2 Cache Card Serial Data:** This is a bidirectional pin used for L2 card serial data.
PC1_DATA	55	I/O	**Processor Card 1 Serial Data:** This is a bidirectional pin used for processor card 1 serial data.
PC2_DATA	54	I/O	**Processor Card 2 Serial Data:** This is a bidirectional pin used for processor card 2 serial data.
SER_CLK	78	O	**Serial Clock:** This is the clock output for serial ROM.
SER_CLKEN	29	O	**Serial Clock Enable:** This active high output enables buffers to drive the serial clock onto PD0 of each card.
System Clock Interface			
FRZ_DATA	49	O	**Freeze data out.** Serial data stream to MPC970 clock chip. See the MPC970 data sheet.
PWR_MNG_CLK	20	I	**Power Management clock.** Used to clock the freeze serial data stream from EPLD to the MPC970 clock chip. See the MPC970 data sheet connected to ISA_CLK.
Power Management (not supported in release 1.0)[1]			
83CX_RESET	91	O 6mA	**PMC reset.** Resets the PMC. EPLD asserts this signal while bit 0 of Port 082B (Power Management Control Register 2) is a 1 (see note 1).
ACTIVITY#	30	O 6mA	**Activity.** EPLD asserts this signal to alert the PMC that system activity (mouse or keyboard) is occurring. No-connect or connect as shown (see note 1).
CMD_STATE#	10	I	**PMC command state.** Indicates that the PMC is in the command state. No-connect or connect as shown (see note 1).
IO_STROBE#	100	O	**I/O strobe.** EPLD asserts this signal to the PMC while 83CX_RESET is low and bit 4 of Port 082A (Power Management Control Register 1) is high. No-connect or connect as shown (see note 1).
PROC_RDY	50	I	**PMC ready.** Indicates that the PMC is in the ready state. No-connect or connect as shown (see note 1).
RWD0	48	I/O 24mA	**PMC serial read/write data bit.** This Is the bi-directional serial data line between the EPLD and the PMC. No-connect or connect as shown (see note 1).
SUSACK#	17	I	**Suspend ACK:** This active low input from the ISA bridge indicates that the system is ready to go into suspend mode. This signal is not used with the Intel SIO – instead it is connected to SUSREQ#.
SUSREQ#	80	O	**Suspend request:** This active low output indicates that suspend has been requested. This signal is not used with the Intel SIO – instead it is connected to SUSACK#.

Table 57. Signal Descriptions (Continued)

Signal Name	Pin	I/O	Description
Power Management (not supported in release 1.0)[1]			
UNFREEZE	47	I	**Unfreeze.** The PMC asserts this signal to EPLD to tell EPLD to unfreeze the clocks. No-connect or connect as shown (see note 1).

Reserved Pins			
HFCS0#	77	O	**Reserved Function.**
HFCS1#	58	O	**Reserved Function.**
IDEIRQP	5	I	**Reserved Function.**
RSVD_BIDI1	42	I /O	**Reserved BIDI 1:** This is reserved BIDI.
RSVD_IN	41	I	**Reserved Input:** This is a reserved input.
Other Signals			
EXTHRST#	4	O(3S)	**External Hard Reset:** This signal is tristate while deasserted. It is driven low for 1024 cycles of the PWR_MGN_CLK to reset the system. See Section 10.2.11.
RESET#	11	I	**System reset.** Used by EPLD to reset internal state machines and internal registers. Also see Section 10.2.11.
GND	12, 13, 37, 38, 62, 63, 87, 88	I	**Ground.**
VCC	18, 19, 43, 44, 68, 69, 93, 94	I	**+5v:**

Note:

1. Power management is not supported on the reference design. The descriptions of the above signals show how to connect those signals if the power management circuits are removed from the board.

10.4 EPLD Design Equations

```
%******************************************************************%
%*********  AHDL  SOURCE CODE FOR KEYSTONE  V1.8   ***************%
%******************************************************************%
          DESIGN IS ″kstnsio″
          BEGIN
              DEVICE ″kstnsio″ IS ″EPM5130WC–1″
              BEGIN
                  A0                             @  72     : INPUT ;
                  A1                             @  71     : INPUT ;
                  A2                             @  70     : INPUT ;
                  A3                             @  67     : INPUT ;
                  A4                             @  66     : INPUT ;
                  A5                             @  65     : INPUT ;
                  A6                             @  64     : INPUT ;
                  A7                             @  61     : INPUT ;
                  /CMDSTATE                      @  10     : INPUT ;
                  /ECSEN                         @  60     : INPUT ;
                  ECS2                           @  59     : INPUT ;
                  ECS1                           @  22     : INPUT ;
                  ECS0                           @  21     : INPUT ;
                  /SUSACK                        @  17     : INPUT ;
                  IDEIRQP                        @   5     : INPUT ;
                  HALT1                          @  27     : INPUT ;
                  HALT2                          @  28     : INPUT ;
                  IRQ1                           @  14     : INPUT ;
                  IRQ12                          @  15     : INPUT ;
                  PWR_MNG_CLK                    @  20     : INPUT ;
                  PROC_RDY                       @  50     : INPUT ;
                  /RESET                         @  11     : INPUT ;
                  /UNFREEZE                      @  47     : INPUT ;
                  /XIOR                          @   9     : INPUT ;
                  /XIOW                          @  16     : INPUT ;
                  RSVD_IN                        @  41     : INPUT ;
                  ISA_CLK                        @  26     : INPUT ;

                  /ACTIVITY                      @  30     : OUTPUT;
                  /SUSREQ                        @  80     : OUTPUT;
                  /CACHE_PD_RD                   @   8     : OUTPUT;
                  /PROC1_PD_RD                   @   7     : OUTPUT;
                  /PROC2_PD_RD                   @   6     : OUTPUT;
                  /DRAM_PD_RD                    @  90     : OUTPUT;
                  /ECPA                          @  89     : OUTPUT;
                  /ECPB                          @  86     : OUTPUT;
                  /ECPC                          @  85     : OUTPUT;
                  /ECPD                          @  84     : OUTPUT;
                  /EQP_PR_RD                     @  79     : OUTPUT;
                  FRZ_DATA                       @  49     : OUTPUT;
                  /HDD_LED                       @  83     : OUTPUT;
                  /HFCS0                         @  77     : OUTPUT;
```

```
            /HFCS1                          @  58      : OUTPUT;
            RUN                             @   2      : OUTPUT;
            /IO_STROBE                      @ 100      : OUTPUT;
            /KYBD_CS                        @  52      : OUTPUT;
            /L2_CACHE_DIS                   @  81      : OUTPUT;
            /L2_CACHE_INH                   @  76      : OUTPUT;
            /L2_FLUSH                       @  51      : OUTPUT;
            /L2_WT                          @   3      : OUTPUT;
            /L2_CONF_DCD                    @  82      : OUTPUT;
            /MOTOR_EN                       @   1      : OUTPUT;
            /NVRAM_RD                       @  75      : OUTPUT;
            /NVRAM_WR                       @  34      : OUTPUT;
            /PLANAR_ID_RD                   @  33      : OUTPUT;
            /RD_3F3                         @  25      : OUTPUT;
            /RTC_AS0                        @  24      : OUTPUT;
            /RTC_AS1                        @  23      : OUTPUT;
            /RTC_RD                         @  32      : OUTPUT;
            /RTC_WR                         @  31      : OUTPUT;
            83CX_RESET                      @  91      : OUTPUT;
            /ROM_EN                         @  45      : OUTPUT;
            /PROC_A1_DIS                    @  35      : OUTPUT;
            /PROC_B1_DIS                    @  36      : OUTPUT;
            /PROC_A2_DIS                    @  39      : OUTPUT;
            /PROC_B2_DIS                    @  40      : OUTPUT;
            /SEMIFOR_ST                     @  46      : OUTPUT;
            SER_CLK                         @  78      : OUTPUT;
            SER_CLKEN                       @  29      : OUTPUT;
            /RTC_ALE                        @  57      : OUTPUT;
            IRQ1SIO                         @  56      : OUTPUT;
           /EXTHRST                         @   4      : OUTPUT;


            PC1_DATA                        @  55      : BIDIR ;
            PC2_DATA                        @  54      : BIDIR ;
            L2_DATA                         @  53      : BIDIR ;
            RWD0                            @  48      : BIDIR ;
            XD0                             @  99      : BIDIR ;
            XD1                             @  98      : BIDIR ;
            XD2                             @  74      : BIDIR ;
            XD3                             @  97      : BIDIR ;
            XD4                             @  96      : BIDIR ;
            XD5                             @  95      : BIDIR ;
            XD6                             @  92      : BIDIR ;
            XD7                             @  73      : BIDIR ;
            RSVD_BIDI1                      @  42      : BIDIR ;
        END;
    END;
```

```
SUBDESIGN KSTNSIO
(
%****************************************************************%
%        DEFINE PRIMARY INPUTS AND OUTPUTS                      *%
%****************************************************************%

%---------------------  Inputs  ---------------------%

        /SUSACK                         :INPUT;
        ECS[2..0]                       :INPUT;
        /ECSEN                          :INPUT;
        A[7..0]                         :INPUT;
        /XIOR                           :INPUT;
        /XIOW                           :INPUT;
        /RESET                          :INPUT;
        PWR_MNG_CLK                     :INPUT;
        IRQ1                            :INPUT;
        IRQ12                           :INPUT;
        IDEIRQP                         :INPUT;
        HALT1                           :INPUT;
        HALT2                           :INPUT;
        PROC_RDY                        :INPUT;
        /CMDSTATE                       :INPUT;
        /UNFREEZE                       :INPUT;
        ISA_CLK                         :INPUT;
        RSVD_IN                        :INPUT;

%---------------  Bidirectional Outputs  ------------%

        XD[7..0]                        :BIDIR;
        RWD0                            :BIDIR;
        PC2_DATA                        :BIDIR;
        PC1_DATA                        :BIDIR;
        L2_DATA                         :BIDIR;
        RSVD_BIDI1                      :BIDIR;

%--------------------  Outputs  ---------------------%

        /EQP_PR_RD                      :OUTPUT;
        /CACHE_PD_RD                    :OUTPUT;
        /PROC1_PD_RD                    :OUTPUT;
        /PROC2_PD_RD                    :OUTPUT;
        /PLANAR_ID_RD                   :OUTPUT;
        /DRAM_PD_RD                     :OUTPUT;
        /ROM_EN                         :OUTPUT;
        /MOTOR_EN                       :OUTPUT;
        /RD_3F3                         :OUTPUT;
        /ECPA                           :OUTPUT;
        /ECPB                           :OUTPUT;
        /ECPC                           :OUTPUT;
        /ECPD                           :OUTPUT;
        /HDD_LED                        :OUTPUT;
        /KYBD_CS                        :OUTPUT;
        /RTC_RD                         :OUTPUT;
        /RTC_WR                         :OUTPUT;
        /RTC_AS0                        :OUTPUT;
        /RTC_AS1                        :OUTPUT;
        /NVRAM_WR                       :OUTPUT;
        /NVRAM_RD                       :OUTPUT;
        /HFCS0                          :OUTPUT;
        /HFCS1                          :OUTPUT;
        RUN                             :OUTPUT;
        FRZ_DATA                        :OUTPUT;
        83CX_RESET                      :OUTPUT;
        /ACTIVITY                       :OUTPUT;
        /IO_STROBE                      :OUTPUT;
        /SUSREQ                         :OUTPUT;
        /L2_CACHE_DIS                   :OUTPUT;
        /L2_CACHE_INH                   :OUTPUT;
        /L2_FLUSH                       :OUTPUT;
        /L2_CONF_DCD                    :OUTPUT;
        /L2_WT                          :OUTPUT;
        SER_CLK                         :OUTPUT;
        SER_CLKEN                       :OUTPUT;
        /SEMIFOR_ST                     :OUTPUT;
        /PROC_A1_DIS                    :OUTPUT;
        /PROC_B1_DIS                    :OUTPUT;
        /PROC_A2_DIS                    :OUTPUT;
        /PROC_B2_DIS                    :OUTPUT;
        /RTC_ALE                        :OUTPUT;
        IRQ1SIO                         :OUTPUT;
        /EXTHRST                        :OUTPUT;
)

VARIABLE

        PSWD1_PRTCTFF          :SRFF;
        PSWD2_PRTCTFF          :SRFF;
        RTC_BLKFF              :SRFF;
        HDD_LEDFF              :SRFF;
        CTL_REG0[7..0]         :SRFF;
        PWR_REG2               :SRFF;
        SHIFT_ENFF             :SRFF;
        START_SHIFTFF          :SRFF;
        IDE_REG[0]             :DFF;
        CNTR[3..0]             :DFF;
        CNTRRST[9..0]          :DFF;
        STARTCNTR1             :DFF;
        /FREEZE                :DFF;
        CLKFF[12..0]           :DFFE;
        SNC_SHIFT_ENFF          :DFFE;
        PROC_A1                 :DFF;
        PROC_B1                 :DFF;
        PROC_A2                 :DFF;
        PROC_B2                 :DFF;
        PROC_SEQ[1..0]          :DFF;
        CMDSTATE_FF             :DFF;
        RSVD_INFF               :DFF;
        SER_REG[5..0]           :DFF;
        L2_CR[1..0]             :DFF;
        SEMIFOR_STFF            :DFF;
        IRQ1SIO_FF              :DFF;

        RWD0_BUFF               :TRI;
        EXTRST                  :TRI;

        DATA0                   :NODE;
        D[7..0]                 :NODE;
        XD_TRI_OE               :NODE;
        EN800_8FF               :NODE;
        RTC_BLK_FLG             :NODE;
        FDC_CS                  :NODE;
        LIGHT_EN                :NODE;
        CTL_REG0_EN             :NODE;
        83CX_CS                 :NODE;
        PWR_REG1_EN             :NODE;
        PWR_REG2_EN             :NODE;
        IDE_REG_EN              :NODE;
        CLKFF_WR                :NODE;
        CLKFF_SELL              :NODE;
        CLKFF_SELH              :NODE;
        GEN_START_BIT           :NODE;
        GEN_STOP_BIT            :NODE;
        STOP_SHIFT              :NODE;
        PROC_SEQ_RD             :NODE;
        PROC_EN                 :NODE;
        PROC_EN_WR              :NODE;
        L2_CR_EN                :NODE;
        L2_CR_WR                :NODE;
        SER_EN                  :NODE;
        SER_REG_WR              :NODE;
        IRQ1SIO_EN0             :NODE;
        IRQ1SIO_EN1             :NODE;
        IRQ1SIO_RST             :NODE;

BEGIN

%       Keyboard Chip Select I/O address range = 0060, 0062, 0064, 0066   %

/KYBD_CS  = !(!ECS[2] & ECS[1] & !ECS[0] & !/ECSEN);

%       RTC RD I/O address range: 0071
        RTC RD will not go active if accessing a range protected by
        PSW1_PRTCTFF or PSW2_PRTCTFF %

/RTC_RD  = !(!RTC_BLKFF.q & !ECS[2] & !ECS[1] & !/ECSEN & !A[2]
           & !A[1] & A[0] & !/XIOR);
```

```
%       RTC WR I/O address range: 0071
        RTC WR will no go active if accessing a range protected by
        PSW1_PRTCTFF or PSW2_PRTCTFF %

/RTC_WR = !(!RTC_BLKFF.q & !ECS[2] & !ECS[1] & !/ECSEN & !A[2]
          & !A[1] & A[0] & !/XIOW);

%       Nvram AS0 I/O address range: 0074 %

/RTC_AS0 = !(!ECS[2] & !ECS[1] & !/ECSEN & A[2] & !A[1] & !A[0]
           & !/XIOW);

%       Nvram AS1 I/O address range: 0075 %

/RTC_AS1 = !(!ECS[2] & !ECS[1] & !/ECSEN & A[2] & !A[1] & A[0]
           & !/XIOW);

%       Nvram Write Enable I/O address : 0077  %

/NVRAM_WR = !(!ECS[2] & !ECS[1] & !/ECSEN & A[2] & A[1] & A[0]
            & !/XIOW);

%       Nvram Write Enable address range:  0077  %

/NVRAM_RD = !(!ECS[2] & !ECS[1] & !/ECSEN & A[2] & A[1] & A[0]
            & !/XIOR);

% RTC_BLK is set if an access to a protected range of the RTC is detected    %
% L4 changes: RTC_BLKFF is clocked by /ECSEN because of the special SIO_CORAL %
% timing of the RTC_ALE signal %

RTC_BLK_FLG     =  PSWD1_PRTCTFF.Q & XD[5] & !XD[4]
                 # PSWD2_PRTCTFF.Q & XD[5] &  XD[4];
RTC_BLKFF.s     = RTC_BLK_FLG & !A[2] & !A[1] & !A[0] & !ECS[2]
                 & !ECS[1] & !/XIOW;
RTC_BLKFF.r     = !RTC_BLK_FLG & !A[2] & !A[1] & !A[0] & !ECS[2]
                 & !ECS[1] & !/XIOW;
RTC_BLKFF.clk   = /ECSEN;
RTC_BLKFF.clrn  = /RESET;

% For SIO, RTCALE I/O address range: 0070 %

/RTC_ALE = !ECS[2] & !ECS[1] & !/ECSEN & !A[2] & !A[1]
           & !A[0] & !/XIOW;

% IDE Hardfile Chip Selects: /HFCS0, /HFCS1 %

/HFCS0 = !(!ECS[2] & !ECS[1] & !ECS[0] & !/ECSEN); % 1F0 – 1F7 %

/HFCS1 = !(!ECS[2] & !ECS[1] &  ECS[0] & !/ECSEN); % 3F6 – 3F7 %

% ROM CHIP ENABLE %

/ROM_EN = !(ECS[2] & !ECS[1] & !ECS[0] & !/ECSEN);

% Floppy Disk Chip Select address range : Primary 3F0 – 3F5, 3F7 %
% Floppy Disk Chip Select address range : Secondary 370–375, 377 %

FDC_CS = ECS[2] & !ECS[1] & !ECS[0] & !/ECSEN ;

% Media sensing enable address range : 03F3 %

/RD_3F3 = !(FDC_CS & A[7] & A[6] & A[5] & A[4] & !A[3]
          & !A[2] & A[1] & A[0] & !/XIOR);
```

% Define EN800_8FF as the code point for the 0800–08FF I/O range %

```
EN800_8FF = !ECS[2] & ECS[1] & !ECS[0] & /ECSEN ;
```

% DRAM_PD_RD I/O address range: 0880 – 883 %

```
/DRAM_PD_RD  = !(EN800_8FF & A[7] & !A[6] & !A[5] & !A[4]
               & !A[3] & !A[2] & !/XIOR);
```

% HDD light I/O address range : 0808 %

```
LIGHT_EN = EN800_8FF & !A[7] & !A[6] & !A[5] & !A[4]
           & A[3] & !A[2] & !A[1] & !A[0];

HDD_LEDFF.s      = LIGHT_EN & XD[0];
HDD_LEDFF.r      = LIGHT_EN & !XD[0];
HDD_LEDFF.clk    =  /XIOW;
HDD_LEDFF.clrn   = /RESET;

/HDD_LED         = !HDD_LEDFF.q ;
```

% Equipment Presence Read (PRSNT_RD) I/O address range: 080C %

```
/EQP_PR_RD       = !(EN800_8FF & !A[7] & !A[6] & !A[5] & !A[4]
                   & A[3] & A[2] & !A[1] & !A[0] & !/XIOR);
```

% L2 CACHE Presence Read (L2_CACHE_RD) I/O address range: 080D %

```
/CACHE_PD_RD     = !(EN800_8FF & !A[7] & !A[6] & !A[5] & !A[4]
                    & A[3] & A[2] & !A[1] & A[0] & !/XIOR);
```

% PC1 Presence Read (PC1_PD_RD) I/O address range: 0866 %

```
/PROC1_PD_RD     = !(EN800_8FF & !A[7] & A[6] & A[5] & !A[4]
                   & !A[3] & A[2] & A[1] & !A[0] & !/XIOR);
```

% PC2 Presence Read (PC2_PD_RD) I/O address range: 0867 %

```
/PROC2_PD_RD     = !(EN800_8FF & !A[7] & A[6] & A[5] & !A[4]
                   & !A[3] & A[2] & A[1] & A[0] & !/XIOR);
```

% L2 Control Register to determine if L2 is WT or WB and set configuration decode bit L2_CR[1..0] I/O address range : 086B %

```
L2_CR_EN  =   EN800_8FF & !A[7] & A[6] & A[5] & !A[4] & A[3] & !A[2]
            & A[1] & A[0];

L2_CR_WR  = L2_CR_EN & !/XIOW;

L2_CR[0].prn  = VCC;
L2_CR[0].clrn = /RESET;
L2_CR[0].clk  = !L2_CR_WR;
L2_CR[0].d    = L2_CR_EN & XD[0];

L2_CR[1].prn  = /RESET;
L2_CR[1].clrn = VCC;
L2_CR[1].clk  = !L2_CR_WR;
L2_CR[1].d    = L2_CR_EN & XD[1];

/L2_WT       = L2_CR[0].q & /RESET;
/L2_CONF_DCD = L2_CR[1].q & /RESET;
```

% Serial ROM Register SER_REG[5..0] I/O address range : 868 %

```
SER_EN  =  EN800_8FF & !A[7] & A[6] & A[5] & !A[4] & A[3] & !A[2]
         & !A[1] & !A[0];

SER_REG_WR  = SER_EN & !/XIOW;

SER_REG[0].prn  = VCC;
SER_REG[0].clrn = /RESET;
SER_REG[0].clk  = !SER_REG_WR;
SER_REG[0].d    =  XD[0];

SER_CLK         = SER_REG[0].q;

SER_REG[1].prn  = /RESET;
SER_REG[1].clrn = VCC;
SER_REG[1].clk  = !SER_REG_WR;
SER_REG[1].d    =  XD[1];

SER_CLKEN       = !SER_REG[1].q;

SER_REG[2].prn  = VCC;
SER_REG[2].clrn = /RESET;
SER_REG[2].clk  = !SER_REG_WR;
SER_REG[2].d    =  XD[4];

SER_REG[3].prn  = VCC;
SER_REG[3].clrn = /RESET;
SER_REG[3].clk  = !SER_REG_WR;
SER_REG[3].d    = XD[5];

SER_REG[4].prn  = VCC;
SER_REG[4].clrn = /RESET;
SER_REG[4].clk  = !SER_REG_WR;
SER_REG[4].d    = XD[6];

SER_REG[5].prn  = VCC;
SER_REG[5].clrn = /RESET;
SER_REG[5].clk  = !SER_REG_WR;
SER_REG[5].d    = XD[7];

PC1_DATA        = TRI(SER_REG[3].q, SER_REG[2].q);
PC2_DATA        = TRI(SER_REG[4].q, SER_REG[2].q);
L2_DATA         = TRI(SER_REG[5].q, SER_REG[2].q);
```

% Processor Sequence register to determine the sequence of the processor. PROC_SEQ[1..0] I/O address range : 0870 %

```
PROC_SEQ_RD =  EN800_8FF & !A[7] & A[6] & A[5] & A[4] & !A[3] & !A[2]
             & !A[1] & !A[0] & !/XIOR;

PROC_SEQ[1..0].prn       = VCC;
PROC_SEQ[1..0].clk       = !PROC_SEQ_RD;
PROC_SEQ[1..0].clrn      = /RESET;

PROC_SEQ[0].d    =  PROC_SEQ_RD & PROC_SEQ[0].q
                  # PROC_SEQ[1].q & PROC_SEQ[0].q
                  # !PROC_SEQ_RD & !PROC_SEQ[0].q;
```

```
PROC_SEQ[1].d   =   PROC_SEQ[1].q
                 # !PROC_SEQ_RD & PROC_SEQ[0].q;
```

% SEMIFOR STATE to detremine processor sequence state %

```
SEMIFOR_STFF.clrn = /RESET;
SEMIFOR_STFF.prn  = VCC;
SEMIFOR_STFF.clk  = PROC_SEQ_RD;
SEMIFOR_STFF.d    = VCC;

/SEMIFOR_ST       = SEMIFOR_STFF.q & PROC_SEQ_RD;
```

% Processor disable register WRITE ONLY PROC_A1, PROC_B1, PROC_A2, & PROC_B2 I/O address range : 0871 %

```
PROC_EN =   EN800_8FF & !A[7] & A[6] & A[5] & A[4] & !A[3] & !A[2]
         & !A[1] & A[0];

PROC_EN_WR    = PROC_EN & !/XIOW;

PROC_A1.prn       = /RESET;
PROC_A1.clk       = !PROC_EN_WR;
PROC_A1.clrn      = VCC;
PROC_A1.d         = ((PROC_EN & XD[0])
                   # (PROC_EN & !XD[3] & !XD[2] & !XD[1] & !XD[0]));

PROC_B1.prn       = /RESET;
PROC_B1.clk       = !PROC_EN_WR;
PROC_B1.clrn      = VCC;
PROC_B1.d         = ((PROC_EN & XD[0])
                   # (PROC_EN & !XD[3] & !XD[2] & !XD[1] & !XD[0]));

 PROC_A2.prn       = /RESET;
 PROC_A2.clk       = !PROC_EN_WR;
 PROC_A2.clrn      = VCC;
 PROC_A2.d         = ((PROC_EN & XD[0])
                    # (PROC_EN & !XD[3] & !XD[2] & !XD[1] & !XD[0]));

 PROC_B2.prn       = /RESET;
 PROC_B2.clk       = !PROC_EN_WR;
 PROC_B2.clrn      = VCC;
 PROC_B2.d         = ((PROC_EN & XD[0])
                    # (PROC_EN & !XD[3] & !XD[2] & !XD[1] & !XD[0]));

/PROC_A1_DIS       =  PROC_A1.q & /RESET;

/PROC_B1_DIS       =  PROC_B1.q & /RESET;

/PROC_A2_DIS       =  PROC_A2.q & /RESET;

/PROC_B2_DIS       =  PROC_B2.q & /RESET;
```

% External Hardware RESET %

```
STARTCNTR1.d    =  /RESET;

STARTCNTR1.clk  = !(PROC_EN_WR & !XD[3] & !XD[2] & !XD[1] & !XD[0] &
/RESET);

STARTCNTR1.clrn = !(CNTRRST[9].q & CNTRRST[8].q
                 & CNTRRST[7].q & CNTRRST[6].q & CNTRRST[5].q
```

```
                & CNTRRST[4].q & CNTRRST[3].q & CNTRRST[2].q
                & CNTRRST[1].q & CNTRRST[0].q);

CNTRRST[].clk   = PWR_MNG_CLK;

CNTRRST[].clrn = !((PROC_EN_WR & !XD[3] & !XD[2] & !XD[1] & !XD[0]
                 & /RESET)
               #  (CNTRRST[9].q & CNTRRST[8].q & CNTRRST[7].q
                 & CNTRRST[6].q & CNTRRST[5].q & CNTRRST[4].q
                 & CNTRRST[3].q & CNTRRST[2].q & CNTRRST[1].q
                 & CNTRRST[0].q));

if STARTCNTR1.q  then                    % Count while STARTCNTR1 %

        CNTRRST[].d = CNTRRST[].q + 1;

   else CNTRRST[].d = CNTRRST[].q;

end if;

EXTRST.oe   =  STARTCNTR1.q;
EXTRST.in   = !STARTCNTR1.q;

/EXTHRST    =  EXTRST.out;

% /CMDSTATE input is temporarily assigned a FF %

CMDSTATE_FF.prn  = VCC;
CMDSTATE_FF.clk  = ISA_CLK;
CMDSTATE_FF.clrn = /RESET;
CMDSTATE_FF.d    = /CMDSTATE;

%  RSVD_IN input is temporarily assigned a FF %

RSVD_INFF.prn    = VCC;
RSVD_INFF.clk    = ISA_CLK;
RSVD_INFF.clrn   = /RESET;
RSVD_INFF.d      = RSVD_IN;

% IRQ1SIO is latched IRQ1 output for SIO %

IRQ1SIO_EN0      = !(/KYBD_CS & /RESET);
IRQ1SIO_EN1      = !(/XIOR & /RESET);

IRQ1SIO_RST      = !(IRQ1SIO_EN0 & IRQ1SIO_EN1);

IRQ1SIO_FF.prn   = VCC;
IRQ1SIO_FF.clk   = IRQ1;
IRQ1SIO_FF.clrn  = IRQ1SIO_RST;
IRQ1SIO_FF.d     = VCC;

IRQ1SIO          =  IRQ1SIO_FF.q;

% RUN  EQUATION  %

RUN   =  !(HALT1 & HALT2);
```

% PSW1_PRTCTFF set one desires to prevent accesses to addresses 20–2F of the RTC space.
PSW1_PRTCTFF I/O address range : 0810 %

```
PSWD1_PRTCTFF.s         = EN800_8FF & !A[7] & !A[6] & !A[5] & A[4]
                          & !A[3] & !A[2] & !A[1] & !A[0];
PSWD1_PRTCTFF.r         = GND;
PSWD1_PRTCTFF.clk       = GLOBAL(/XIOW);
PSWD1_PRTCTFF.clrn      = /RESET;
```

% PSW2_PRTCTFF is set one desires to prevent accesses to addresses 30–3F of the RTC space
PSW1_PRTCTFF I/O address range : 0812 %

```
PSWD2_PRTCTFF.s          =  EN800_8FF & !A[7] & !A[6] & !A[5] & A[4]
                          & !A[3] & !A[2] & A[1] & !A[0];
PSWD2_PRTCTFF.r          = GND;
PSWD2_PRTCTFF.clk        = GLOBAL(/XIOW);
PSWD2_PRTCTFF.clrn       = /RESET;
```

% Control register 0 I/O address range : 081C %

```
CTL_REG0_EN               = EN800_8FF & !A[7] & !A[6] & !A[5] & A[4]
                           & A[3] & A[2] & !A[1] & !A[0];

CTL_REG0[7..0].s = CTL_REG0_EN &  XD[7..0];
CTL_REG0[7..0].r = CTL_REG0_EN & !XD[7..0];
CTL_REG0[].clk   =  /XIOW;
CTL_REG0[].clrn  = /RESET;

/ECPA          = !(CTL_REG0[2] & !CTL_REG0[1] & !CTL_REG0[0]);
/ECPB          = !(CTL_REG0[2] & !CTL_REG0[1] & CTL_REG0[0]);
/ECPC          = !(CTL_REG0[2] & CTL_REG0[1] & !CTL_REG0[0]);
/ECPD          = !(CTL_REG0[2] & CTL_REG0[1] & CTL_REG0[0]);
/MOTOR_EN      = CTL_REG0[3].q;

/L2_FLUSH      = CTL_REG0[4].q;
/L2_CACHE_DIS  = CTL_REG0[6].q & RUN;
/L2_CACHE_INH  = CTL_REG0[7].q;

        %****************************************************%
        % Write Power Control Register 1 I/O address: 082A   %
        %                                                    %
        % (MSB)         Bits 7–5       Reserved              %
        %               Bit  4         I/O Strobe Key  (W/O) %
        %               Bits 3–1       Data/Command          %
        %               B[3..1] to 83C750              (W/O) %
        % (LSB)         Bit  0         83C750 D0       (R/W) %
        %****************************************************%

83CX_CS       = EN800_8FF & !A[7] & !A[6] &  A[5] & !A[4]
                & A[3] &!A[2] & A[1];
```

% Write ZERO to 83C750 %

```
PWR_REG1_EN     = 83CX_CS & !A[0];

RWD0_BUFF.oe    = PWR_REG1_EN & !/XIOW & !XD[0];
RWD0_BUFF.in    = GND;
```

```
RWDO           = RWDO_BUFF.OUT;

% 83C750 I/O Strobe Key %

/IO_STROBE   = !(XD[4] & PWR_REG1_EN & !/XIOW & PROC_RDY
               & !83CX_RESET);

        %**************************************************%
        % Write Power Control Register 2 I/O address: 082B   %
        %                                                    %
        % (LSB)         Bit  0         Reset 83C750 (W/O)    %
        %**************************************************%

PWR_REG2_EN     = 83CX_CS & A[0];

PWR_REG2.s      =  XD[0] & PWR_REG2_EN & /XIOW;
PWR_REG2.r      = !XD[0] & PWR_REG2_EN & /XIOW;
PWR_REG2.clrn   = VCC;
PWR_REG2.clk    = GLOBAL(/XIOW);

83CX_RESET      = PWR_REG2.q;

%  Activity Alert to the 83C750   %

/ACTIVITY       = /FREEZE # !(IRQ1 # IRQ12);

% SUSREQ  signal to CORAL controls when it goes to SUSPEND state %

/SUSREQ   =  VCC;

%  IDE Interrupt Status Read I/O address range: 0838    %

IDE_REG_EN      = !A[7] & !A[6] & A[5] & A[4] & A[3] & !A[2]
                  & !A[1] & !A[0] & EN800_8FF;

%  IDE Primary Interrupt Status bit                    %

IDE_REG[0].d    = VCC;
IDE_REG[0].clk  = IDEIRQP;
IDE_REG[0].clrn = !(!XD[0] & IDE_REG_EN & !/XIOW # !/RESET);
IDE_REG[0].prn  = VCC;

%  Planar ID Read (PLANAR_ID) I/O address range: 0852 %

/PLANAR_ID_RD   = !(EN800_8FF & !A[7] & A[6] & !A[5] & A[4]
                  & !A[3] & !A[2] & A[1] & !A[0] & !/XIOR);


/FREEZE.prn     =  /UNFREEZE & /RESET;
/FREEZE.clrn    =  VCC;
/FREEZE.d       =  GND;
/FREEZE.clk     =  !/SUSACK;

% Freeze Clock Logic – 13 Bit serial shift register              %
% Decode the low order CLKFF[7..0] on addresses 0860   %
% Decode the high order CLKFF[12..8] on address 0862   %

CLKFF_WR     = !A[7] & A[6] & A[5] & !A[4] & !A[3] & !A[2] & !A[0]
               & EN800_8FF & !/XIOW;
CLKFF_SELL   = !A[7] & A[6] & A[5] & !A[4] & !A[3] & !A[2] & !A[1]
               & !A[0] & EN800_8FF;
```

```
CLKFF_SELH    = !A[7] & A[6] & A[5] & !A[4] & !A[3] & !A[2] & A[1]
              & !A[0] & EN800_8FF;

CLKFF[12..0].prn         = /RESET;
CLKFF[12..0].clrn        = VCC;

CLKFF[12..0].ena         = START_SHIFTFF.q # CLKFF_WR;
CLKFF[12..0].clk         = PWR_MNG_CLK;

if  START_SHIFTFF.q  then                  % Shift with wraparound %

        CLKFF[11..0].d    = CLKFF[12..1].q;
        CLKFF[12].d       = CLKFF[0].q;

        else if CLKFF_SELL then            % Write to CLKFF[7..0]  %

                CLKFF[1..0].d    = XD[1..0];
                CLKFF[2].d       = VCC;
                CLKFF[7..3].d    = XD[7..3];
                CLKFF[12..8].d   = CLKFF[12..8].q;

                else if CLKFF_SELH then     % Write to CLKFF[12..8]%

                        CLKFF[7..0].d    = CLKFF[7..0].q;
                        CLKFF[11..8].d   = XD[3..0];
                        CLKFF[12].d      = VCC;

                        else    CLKFF[12..0].d   = CLKFF[12..0].q;

                end if;

        end if;

end if;

SHIFT_ENFF.s    = CLKFF_SELH;           % Start shifting upon   %
                                        % write to 862  %
SHIFT_ENFF.r    = GND;
SHIFT_ENFF.prn  = /RESET;
SHIFT_ENFF.clrn = !GEN_STOP_BIT;        % Clear when it   %
                                                % reaches 15     %
SHIFT_ENFF.clk  = /XIOW;

SNC_SHIFT_ENFF.d        = SHIFT_ENFF.q; % One clock after%
                                                % SHIFT_ENFF    %
SNC_SHIFT_ENFF.ena      = /RESET;
SNC_SHIFT_ENFF.clrn     = /RESET;
SNC_SHIFT_ENFF.clk      = PWR_MNG_CLK;

CNTR[].clk              = PWR_MNG_CLK;
CNTR[].clrn             = /RESET;       % Counter resets to ZERO %

if SNC_SHIFT_ENFF.q  then    % Count while SNC_SHIFT_ENFF%

        CNTR[].d        = CNTR[].q + 1;
        else
                CNTR[].d        = CNTR[].q;

end if;
```

```
GEN_START_BIT   = !CNTR[3].q & !CNTR[2].q & !CNTR[1].q & CNTR[0].q;
GEN_STOP_BIT    =  CNTR[3].q &  CNTR[2].q &  CNTR[1].q & CNTR[0].q;
STOP_SHIFT      =  CNTR[3].q &  CNTR[2].q &  CNTR[1].q & !CNTR[0].q;

START_SHIFTFF.s         = GEN_START_BIT;
START_SHIFTFF.r         = STOP_SHIFT;
START_SHIFTFF.clrn      = /RESET;
START_SHIFTFF.clk       = PWR_MNG_CLK;
```

% FRZ_DATA is a 1 when the counter = 0. It is also a 1 whenever START_SHIFTFF is 1 and CLKFF[0] is a 1 and then when the counter is fifteen to leave it in the HIGH state. %

```
FRZ_DATA     = !CNTR[3].q & !CNTR[2].q & !CNTR[1].q & !CNTR[0].q
             #  CNTR[3].q &  CNTR[2].q &  CNTR[1].q & !CNTR[0].q
             #  CNTR[3].q &  CNTR[2].q &  CNTR[1].q &  CNTR[0].q
             #  START_SHIFTFF.q & CLKFF[0].q;


RSVD_BIDI1 = TRI(DATA0, GND);
DATA0  = GND;


        %****************************************************%
        %                    XD BUS                          %
        %****************************************************%


XD_TRI_OE =  ((LIGHT_EN # CTL_REG0_EN # IDE_REG_EN # PROC_EN
          # SER_EN # CLKFF_SELL # CLKFF_SELH) & !/XIOR)
          # PROC_SEQ_RD;

XD[0]  = TRI(D[0], XD_TRI_OE);

D[0]   = CTL_REG0_EN & CTL_REG0[0].q
       # LIGHT_EN & HDD_LEDFF.q
       # PROC_SEQ[0].q & PROC_SEQ_RD
       # PROC_A1.q & PROC_EN
       # IDE_REG[0].q & IDE_REG_EN
       # SER_REG[0].q & SER_EN;

XD[1]  = TRI(D[1], XD_TRI_OE);
D[1]   = CTL_REG0_EN & CTL_REG0[1].q
       # PROC_SEQ[1].q & PROC_SEQ_RD
       # PROC_B1.q & PROC_EN
       # SER_REG[1].q & SER_EN;

XD[2] = TRI(D[2], XD_TRI_OE);
D[2]  = CTL_REG0_EN & CTL_REG0[2].q # PROC_A2.q & PROC_EN;

XD[3] = TRI(D[3], XD_TRI_OE);
D[3]  = CTL_REG0_EN & CTL_REG0[3].q # PROC_B2.q & PROC_EN;

XD[4] = TRI(D[4], XD_TRI_OE);
D[4]  = CTL_REG0_EN & CTL_REG0[4].q # SER_EN & SER_REG[2].q;

XD[5] = TRI(D[5], XD_TRI_OE);
D[5]  = CTL_REG0_EN & CTL_REG0[5].q # SER_EN & PC1_DATA;
```

```
XD[6] = TRI(D[6], XD_TRI_OE);
D[6]  = CTL_REG0_EN & CTL_REG0[6].q # PROC_EN & RSVD_INFF.q
      # SER_EN & PC2_DATA;

XD[7] = TRI(D[7], XD_TRI_OE);
D[7]  = CTL_REG0_EN & CTL_REG0[7].q # PROC_EN & CMDSTATE_FF.q
      # SER_EN & L2_DATA;

END;
```

10.5 EPLD Pinout

Table 58. EPLD Pinout

PIN#	SIGNAL NAME	TYPE
1	FD_MTR_EN#	O
2	RUN	O
3	L2_WT#	O
4	EXTHRST#	I
5	IDEIRQP(RESERVED)	I
6	PROC2_PD_RD#	O
7	PROC1_PD_RD#	O
8	CACHE_PD_RD#	O
9	XIOR#	I
10	CMD_STATE(RESERVED)#	I
11	RESET#	I
12	GND	
13	GND	
14	IRQ1	I
15	IRQ12	I
16	XIOW#	I
17	SUSACK(RESERVED)#	I
18	VCC	
19	VCC	
20	PWR_MNG_CLK	I
21	ECS[0]	I
22	ECS[1]	I
23	RTC_AS1#	O
24	RTC_AS0#	O
25	RD_3F3#	O
26	ISA_CLK	I
27	HALT1	I
28	HALT2	I
29	SER_CLKEN	O
30	ACTIVITY(RESERVED)#	O
31	RTC_WR#	O
32	RTC_RD#	O
33	PLANAR_ID_RD#	O
34	NVRAM_WR#	O
35	PROC_A1_DIS#	O
36	PROC_B1_DIS#	O
37	GND	
38	GND	
39	PROC_A2_DIS#	O
40	PROC_B2_DIS#	O
41	RSVD_IN	I
42	RSVD_BIDI1	I/O
43	VCC	
44	VCC	
45	ROM_EN#	O
46	SEMIFOR_ST#	O
47	UNFREEZE(RESERVED)#	I
48	RWD0(RESERVED)	I/O(t.s)
49	FRZ_DATA	O
50	PROC_RDY(RESERVED)	I
51	L2_FLUSH#	O
52	KYBD_CS#	
53	L2_DATA	I/O
54	PC2_DATA	I/O
55	PC1_DATA	I/O
56	IRQ1SIO	O
57	RTC_ALE#	O
58	HFCS1(RESERVED)#	O
59	ECS[2]	I
60	ECSEN#	I
61	XA[7]	I
62	GND	
63	GND	
64	XA[6]	I
65	XA[5]	I
66	XA[4]	I
67	XA[3]	I
68	VCC	
69	VCC	
70	XA[2]	I
71	XA[1]	I
72	XA[0]	I

PIN#	SIGNAL NAME	TYPE
73	XD[7]	I/O
74	XD[2]	I/O
75	NVRAM_RD#	O
76	L2_CACHE_INH#	O
77	HFCS0(RESERVED)#	O
78	SER_CLK	O
79	EQP_PR_RD#	O
80	SUSREQ(RESERVED)#	O
81	L2_CACHE_DIS#	O
82	L2_CACHE_DCD#	O
83	HDD_LED#	O
84	ECPD#	O
85	ECPC#	O
86	ECPB#	O

PIN#	SIGNAL NAME	TYPE
87	GND	
88	GND	
89	ECPA#	O
90	DRAM_PD_RD#	O
91	83CX_RESET(RESERVED)	I
92	XD[6]	I/O
93	VCC	
94	VCC	
95	XD[5]	I/O
96	XD[4]	I/O
97	XD[3]	I/O
98	XD[1]	I/O
99	XD[0]	I/O
100	IO_STROBE(RESERVED)#	O

Section 11
I/O Subsystems

11.1 Ethernet Subsystem

The Ethernet implementation in this reference design uses the Advanced Micro Devices (AMD) 79C970A PCnet™-PCI II Ethernet Controller. The 79C970A, transmit and receive termination resistors, filter/transformer module, RJ45 connector, and assorted resistors and capacitors form a 10BASE-T implementation of an Ethernet node. For detailed information on the 79C970A Ethernet controller, see the data sheet, AMD publication number 19436. The reference design documentation describes the basic design guidelines and the unique aspects of the reference design implementation of the device.

A well written application note entitled ”Pcnet Family Board Design and Layout Recommendations” is also available from AMD. Some of the design practices mentioned in this section are described more fully in the application note.

An 8-pin socketed serial EEPROM is used to automatically configure the controller at power up, and a green LED is used to display the link status. The reference design does not support writing to the serial EEPROM.

The Ethernet controller offers some additional capabilities that are not implemented in the reference design. Among these are 10BASE-2 and other protocol implementations which require the use of the Attachment Unit Interface (AUI). The Ethernet controller contains a General Purpose Serial Interface (GPSI) which is not used. The expansion ROM interface is also not used.

The combination of analog and digital circuitry in the Ethernet interface poses significant challenges for the circuit, system, and physical designer. Low noise design techniques are required to ensure proper function of the interface.

Refer to Section 13 for register listings and setup information.

11.1.1 Ethernet Physical and Electrical Design Guidelines

The Ethernet implementation in the reference design follows the guidelines recommended by AMD for signal routing and controller placement:

- Signal traces are kept short by correctly orienting the Ethernet controller relative to the board edge where the filter/transformer module and RJ45 connector are located.
- No power or ground plane beneath the filter/transformer module or RJ45 connector.
- No routing of signals under the analog corner of the controller where the PLL is located. This area is located at the corner of the device with pins 99 and 100.

- Put no power or ground plane under the RXD ± signal traces, and route them together as a pair. Route the TXD ±, TXP ±, and all other signals well away from the RXD ± signals.
- Keep Ethernet circuitry together and close to the perimeter of the board, near the media connector.
- Keep other noisy components away from Ethernet circuitry.

11.1.1.1 Ethernet Power and Ground Guidelines

Noise-free power and ground connections are critical for good performance of the Ethernet implementation. Using a circuit board with four or more layers with dedicated power and ground planes can make designing a reliable Ethernet interface easier.

Adequate decoupling is necessary for all designs. The reference design uses a bulk capacitor for low-frequency decoupling (33uF) and 9 high-frequency decoupling capacitors (0.1uF) which are arranged around the perimeter of the device. The V_{CC} and ground pins are connected directly to the high-frequency capacitors, rather than through the planes, so that inductance between the power and ground pins is minimized.

A 0.1uF capacitor is placed between the AVSS1 and AVDD3 pins, and a low-pass filter is placed between pins AVSS2 and AVDD2. In addition to the 10uF capacitor used in this filter, the 79C970A application note suggests a 0.1uF capacitor be placed in parallel with the 10uF.

11.1.1.2 Ethernet 10BASE-T Layout Guidelines

Since common mode noise is the primary source of radiated energy from this interface it is suggested that a common-mode choke be used. This reference design uses a filter/transformer with an integrated common-mode choke for this purpose.

11.1.1.3 Ethernet Oscillator Guidelines

Please refer to the 79C970A specification and the application note described at the start of the Ethernet section to be sure that your crystal oscillator will meet the controller requirements.

11.1.2 Ethernet Sleep Mode

The 79C970A Ethernet controller has a sleep-mode input. When this input is asserted the device will enter a power-saving mode. This reference design connects the SLEEP pin to a socket for control by a power management controller. The power-management controller and necessary programming is not provided with this reference design.

An output from the Ethernet controller (LNKST*) is also wired to the power management controller socket. This signal is intended to allow a power management controller the ability to wake up the system when there is Ethernet activity.

11.2 SCSI Subsystem

This reference design uses the Symbios Logic 53C810 PCI-SCSI I/O Processor, an external oscillator, active terminator, internal and external SCSI connectors, and various decoupling capacitors to implement a high speed synchronous or asynchronous SCSI bus which interfaces to the system through the PCI bus.

The 53C810 supports the following features:

- Eight-bit SCSI-2 interface
- Variable block size and scatter/gather data transfers
- 32-bit word data bursts with variable burst lengths
- Full 32-bit PCI bus master
- 64-byte FIFO buffer.

In the past, the Symbios Logic part was marketed under the NCR brand name. See the manufacturer's data sheet for more information.

Refer to Section 13 for register listings and setup information.

11.2.1 SCSI Physical and Electrical Design Guidelines

Special consideration for component location, bus routing, decoupling, impedance matching and termination are necessary for a reliable SCSI implementation.

11.2.1.1 SCSI Component Location

The PCI and SCSI buses each have requirements which must be satisfied to ensure proper function of each bus. In the reference design, the SCSI controller is located close to the PCI bus to minimize trace length, and it is oriented so that critical path lengths for PCI and SCSI buses are minimized, and so that signal crossovers are avoided.

11.2.1.2 SCSI Bus Routing

The wiring guidelines section of this document describe the general rules to use in routing the SCSI bus. The goal is to provide wiring channels which are low noise and not subject to crosstalk with other signals.

11.2.1.3 SCSI Decoupling

The specification for the 53C810 indicates that bypass capacitor values between .01 and .1uF should provide adequate noise isolation. This reference design used a single bulk decoupling capacitor of value 33uF and about (10) .01uF capacitors located on the back side of the system board, around the perimeter of the device.

11.2.1.4 SCSI Impedance Matching

The SCSI-3 SPI specification requires characteristic SCSI bus impedance to be 84 ±12 ohms. The characteristic impedance of multilayer PC cards can vary, but the internal planes will often be found to be on the lower end of this range. Internal traces for this reference design have an approximate impedance of 75 ohms while external traces have an impedance of 85-95 ohms. This means that any combination of wiring on internal and external layers was acceptable. Traces that run on outer planes have higher impedances than those that run on inner planes.

11.2.1.5 SCSI EMC Considerations

The reference design includes a set of inductors at the internal SCSI connector, which are intended to limit EMI radiated by the internal (flat) SCSI cable. The external SCSI connector was not so equipped because the shield of the external SCSI cable is considered adequate to limit the radiated EMI to acceptable limits.

11.2.2 SCSI Termination

Since the SCSI bus is a true transmission line it must be terminated at each end to prevent signal reflections. The reference design relies on two different styles of termination for the ends of the bus. See Figure 49 for a high level description of how the bus is routed and terminated.

Figure 49 shows a block diagram of the reference design SCSI bus implementation. The design requires at least one SCSI device to be installed on the SCSI bus via J32, the internal SCSI connector. The internal SCSI bus must be terminated off-board, at the SCSI device at the end of the cable. There must be exactly one terminator on the internal SCSI cable. The internal SCSI connector provides fused external terminator power on pin 26.

When a SCSI cable with a compliant SCSI device is plugged into the external SCSI connector (J19), pin 36 of the connector is shorted to ground. This causes the EXT_SCSI_GND24 signal to go low. This causes the active (to 2.9v) terminator located by the external SCSI connector to be disabled. This mechanism enables the onboard terminator while no devices are connected to the external SCSI connector, and it disables the onboard terminator while an external SCSI device is attached.

If an external SCSI device is attached, it is necessary to terminate the last device on the cable in accordance with normal SCSI practice. However, if no SCSI devices are attached to the external SCSI connector, it is not necessary to install a terminator in the connector.

It is still necessary to use a terminator on the last external drive to terminate the SCSI bus on the external side. The external SCSI connector provides fused external terminator power on pin 38.

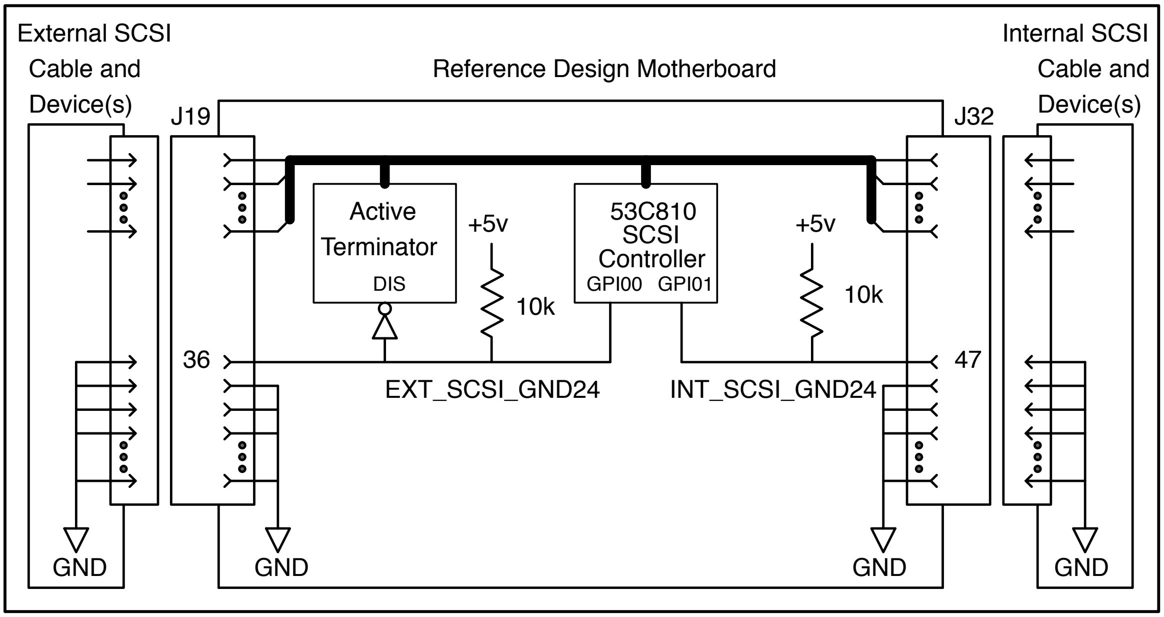


Figure 49. General SCSI Bus Wiring

11.2.3 Cable/Device Presence Detect

Figure 49 also shows the EXT_SCSI_GND24 INT_SCSI_GND24 signals running to the SCSI controller. The reference design implements the general purpose I/O capability of the SCSI controller to detect the presence or absence of cables at the internal and external connectors. Cable presence may be read from the general purpose register of the SCSI device (GPREG 07/87) as follows:

- Bit 0 is external SCSI: 0 = cable connected, 1 = no connection.
- Bit 1 is internal SCSI: 0 = cable connected, 1 = no connection.

Software may use this information to modify SCSI operations.

Warning: The general purpose control register (GPCNTL 47/C7) bits 0:1 (little endian) must remain in their default settings (1's) so that GPIO bits 0:1 are configured as inputs. Changing this setting can result in damage to the hardware.

11.2.4 SCSI Interrupts

There are various conditions that will cause the SCSI controller to assert INT# to the interrupt controller. By the time the interrupt service routine services the interrupt, it is possible that the SCSI controller will have internally stacked up interrupt requests from several sources. Additionally, it is possible for multiple SCSI and DMA interrupt sources to be simultaneously recognized by the SCSI controller. In order to avoid dropping interrupts, the software must check all status registers in the controller and service all outstanding SCSI interrupts each time the processor is interrupted by the SCSI device.

This reference design routes the signal IRQ# from the SCSI controller to several locations including the MPIC, the ISA bridge, and the power management controller socket. This is done to allow the greatest degree of flexibility in handling the interrupt under various operating systems. It is expected that only the interrupt wired to the MPIC will be used, as this is the most efficient way to service SCSI interrupts. The interrupt which is routed to MPIC is level sensitive, and can be treated like any other level sensitive interrupt except as noted in the above paragraph.

11.3 Native I/O Subsystems

The core of the Native I/O subsystem is the National PC87332VLJ SuperI/O™ multi-purpose chip, the keyboard mouse controller and the Dallas Semiconductor™ DS1385S RTC/NVRAM chip. The later two subsystems are described in the X–BUS section. The SuperI/O handles the following functions for the reference design:

- Floppy disk controller. Software compatible with DP8473, 765A, and NS82077.
- Two UARTs. Implemented as two RS–232 serial ports. Software compatible with PC16550A, and PC16450.
- Bidirectional Parallel port with ECP support.

The reference design sets the default configuration of the superIO chip at power–up using configuration strapping resistors connected to the configuration inputs CFG[4..0] =b' 01100'. This enables the Floppy Disk Controller as Primary, UART1 as COM1, UART2 as COM2,and the parallel port as LPT2. The IDE interface is disabled.

Refer to the schematics and to the SuperI/O data sheet for more information. Refer to Section 13 for Register list, configuration, and Setup information.

11.4 Business Audio Subsystem

Business Audio is implemented with the Crystal Semiconductor CS4232 Multimedia Audio System Controller and Codec Module (see Figure 50).

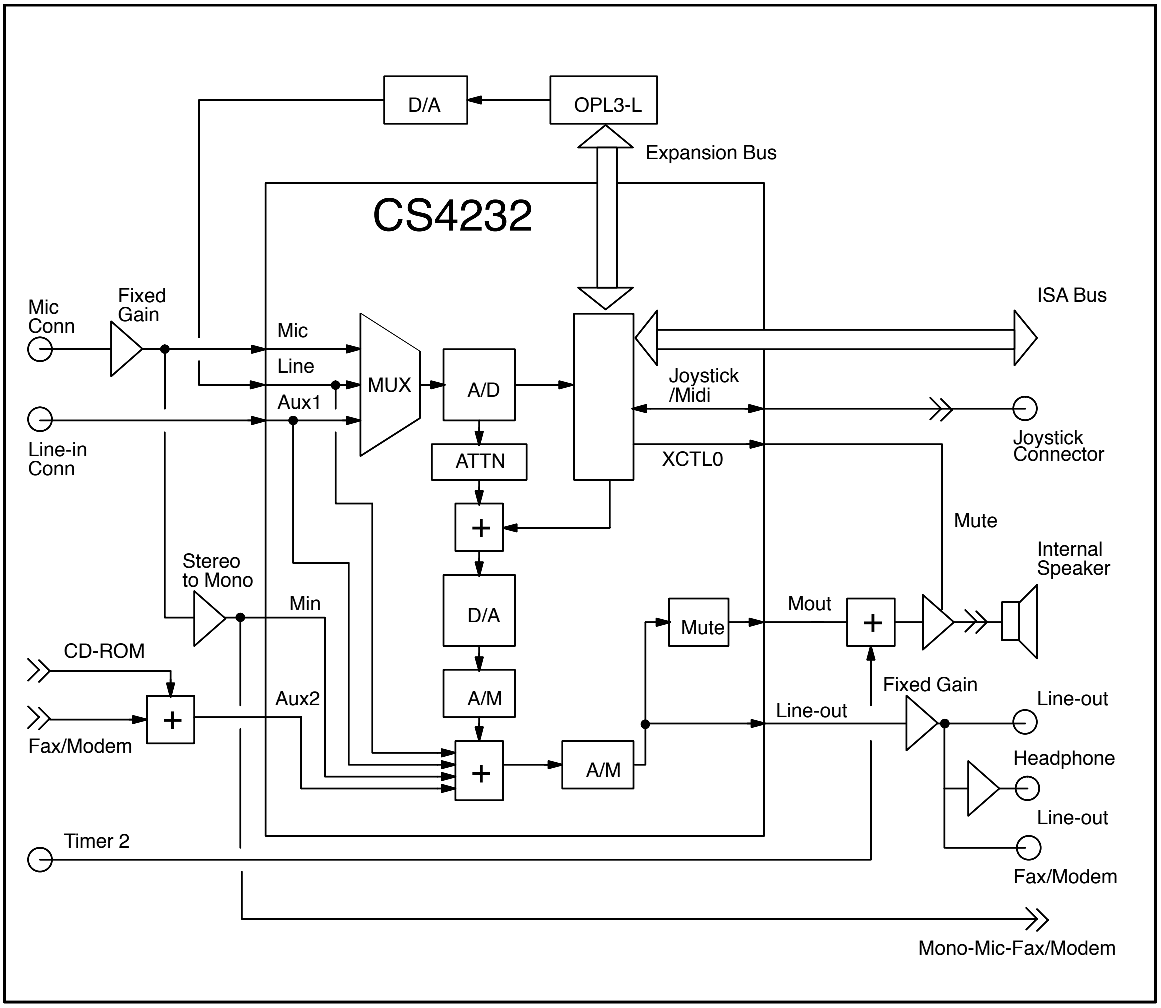


Figure 50. Audio Flow Diagram

Conventional (Timer 2) PC speaker functions are also provided (both the Timer 2 signal from the SIO, and the audio chip, drive the speaker).

The system provides for stereo capture and playback. It can play MIDI files, but it is not a full-functioned MIDI system. It has separate DMA channels for record and playback. The system is processor driven and does not include a DSP. Compression and decompression are supported in the hardware. The audio output is to a single speaker mounted in the cabinet. Also supported are four rear-mounted 3.5 mm jacks for:

- Stereo earphones
- Stereo microphone input
- Stereo line in
- Stereo line out.

There are also motherboard connectors for direct playback from the CD-ROM and for an internal fax-modem card.

Refer to Section 13 for configuration, setup and initialization information.

11.4.1 Audio Performance

Table 59. Business Audio Subsystem Performance

Sampling Rate	48 KHz per Channel Maximum
Channel Bandwidth (Line Out and Headphone)	20 Hz to 20 KHz (+1/–3dB)
Dynamic Range	16-bit Resolution
Line In Signal to Noise + Distortion Ratio	80 dB Min @ 1KHz (Measured at Line Out)
Microphone Signal to Noise + Distortion Ratio	60 dB Min @ 1KHz (Measured at Line Out)

11.4.2 Audio Connector Specifications

Table 60. Business Audio Connector Specifications

Line In	
Input Impedance	4 Kohms (use 2 Kohms source impedance or lower)
Signal Level	2 v rms Nominal

Microphone Input	
Input Impedance	3.8 Kohms (use 50 to 1000 ohm microphone)
Phantom Power	Stereo 0.5ma for Nominal Operation at 2 v DC
Signal Level	10 mv rms Nominal, 50 mv rms Maximum

CD ROM Input (Internal)	
Input Impedance	3 Kohms
Signal Level	1 v rms Nominal

Line Out	
Output Impedance	1 Kohm (use 10K ohm Load Impedance or Higher)
Signal Level	2 v rms Nominal
Headphone Output	
Output Impedance	12 ohms
Signal Level	1.5 v rms Nominal

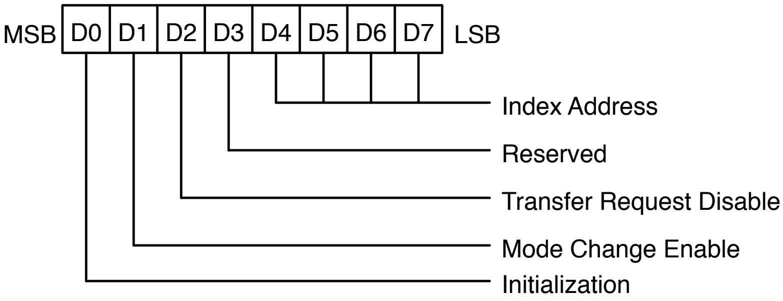
System Speaker Output	
Output Impedance	8 ohms Typical
Power Output	1 Watt Maximum at 8 ohms

Fax/Modem Connection (Internal)	
Input Impedance	4 Kohms (use 2 Kohms source impedance or lower)
Signal Level (Input)	2 v rms Nominal
Output Impedance	12 ohms
Signal Level (Output)	1.5 v rms Nominal
Input Impedance (Microphone)	3.8 Kohms (use 50 to 1000 ohm microphone)
Signal Level (Microphone Input)	10 mv rms Nominal, 50 mv rms Maximum

11.4.3 Audio Control Registers

11.4.3.1 Audio Index Register

ISA Port	0534	Read/Write	Reset to 40h



Bit 0 Initialization. This bit is set when the module is in a state which cannot respond to parallel bus cycles. This bit is read only.

Bit 1 Mode Change Enable. This bit must be set whenever the current functional mode of the module is changed. Specifically, the Clock and Data Format and Interface Configuration registers cannot be changed unless this bit is set. MCE should be cleared at the completion of the desired register changes.

Bit 2 Transfer Request Disable. This bit, when set, causes DMA transfers to cease when the Interrupt Status (INT) bit of the Status Register is set.

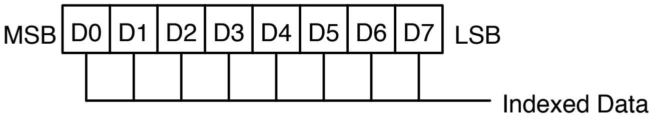
Bit 3 Reserved

Bits 7:4 Index Address. These bits define the address of the register accessed by the Indexed Data Register.

Immediately after reset and once the module has left the INIT state, the initial value of this register will be 40h. During initialization, this register cannot be written and is always read 80h.

11.4.3.2 Audio Indexed Data Register

ISA Port	0535	Read Only	Reset n/a

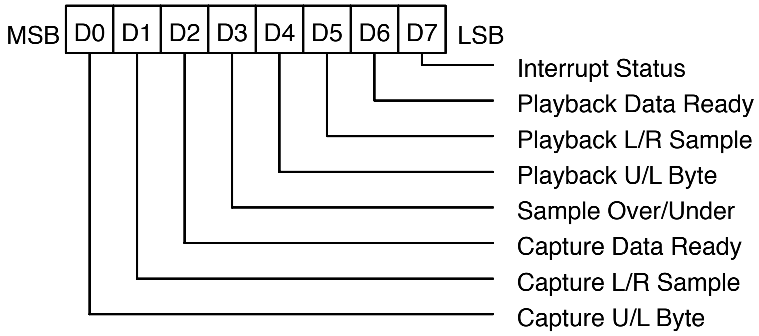


Bits 7:0 Indexed Register Data. These bits contain the contents of the register referenced by the Indexed Data Register.

During initialization, this register cannot be written and is always read as 80h.

11.4.4 Audio Status Register

ISA Port 0536	Read Only	Reset to 40h



Bit 0 Capture Upper/Lower Byte. This bit indicates whether the PIO capture data ready is for the upper or lower byte of the channel. This bit is read only.
0 = Lower byte ready.
1 = Upper byte ready or any 8 bit mode.

Bit 1 Capture Left/Right Sample. This bit indicates whether the PIO capture data waiting is for the right channel or left channel. This bit is read only.
0 = Right channel.
1 = Left channel or mono.

Bit 2 Capture Data Ready. The PIO Capture Data Register contains data ready for reading by the host. This bit should only be used when direct programmed I/O data transfers are desired. This bit is read only.
0 = Data is stale. Do not reread.
1 = Data is fresh. Ready for next host data read.

Bit 3 Sample Over/Underrun. This bit indicates that the most recent sample was not serviced in time and therefore either a Capture Overrun or Playback Underrun has occurred. If both capture and playback are enabled, the source which set this bit can be determined by reading the Playback Underrun and Capture Overrun bits in the Test and Initialization Register.

Bit 4 Playback Upper/Lower Byte. This bit indicates whether the PIO playback data needed is for the upper or lower byte of the channel. This bit is read only.
0 = Lower byte needed.
1 = Upper byte needed or any 8 bit mode.

Bit 5 Playback Left/Right Sample. This bit indicates whether the PIO playback data needed is for the right channel DAC or left channel DAC. This bit is read only.
0 = Right channel needed.
1 = Left channel or mono.

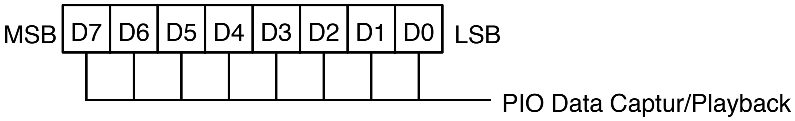
Bit 6 Playback Data Register Ready. The PIO Playback Data Register is ready for more data. This bit should only be used when direct programmed I/O data transfers are desired. This bit is read only.
0 = DAC data is still valid. Do not overwrite.
1 = DAC data is stale. Ready for next host data write value.

Bit 7 Interrupt Status. This sticky bit (the only one) indicates the status of the interrupt logic. This bit is cleared by any host write of any value to this register. The Interrupt Enable bit of the Pin Control Register determines whether the state of this bit is reflected on the Interrupt Pin of the AD1848. The only interrupt condition supported by the AD1848 is generated by the underflow of the DMA Current Count Register.
0 = Interrupt pin inactive.
1 = Interrupt pin active.

11.4.5 Audio PIO Data Register

ISA Port	0537	Read/Write	Reset n/a

The PIO Data Registers are two registers mapped to the same address. Writes send data to the PIO Playback Data Register. Reads will receive data from the PIO Capture Data Register.



Bits 7:0 PIO Capture Data Register. This is the control register where capture data is read during programmed I/O data transfers.

The reading of this register will increment the state machine so that the following read will be from the next appropriate byte in the sample. The exact byte which is next to be read can be determined by reading the Status Register. Once all relevant bytes have been read, the state machine will stay pointed to the last byte of the sample until a new sample is received. Once this has occurred, the state machine and status register will point to the first byte of the sample. Until a new sample is received, reads from this register will return the most significant byte of the sample.

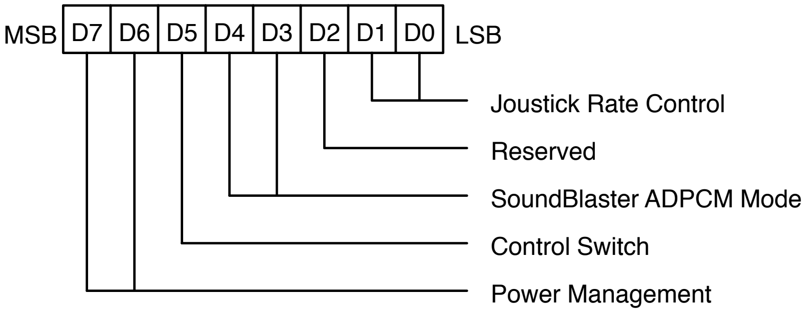
Bits 7:0 PIO Playback Data Register. This is the control register where playback data is written during programmed I/O data transfers.

Writing data to this register will increment the playback byte tracking state machine so that the following write will be to the correct byte of the sample. Once all bytes of a sample have been written, subsequent byte writes to this port are ignored.

11.4.6 The CS4232 Logical Device

The CS4232 logical device includes two registers for controlling functions that did not fit into the other logical devices. These features include game port rate control and power management. All port addresses are relative to the base address set in the configuration.

11.4.7 Control Register 0



Bits 1:0 Joystick Rate Control. Selects the operating speed of the joystick.
00 = Slowest speed.
01 = Medium speed.
10 = Medium fast speed.
11 = Fastest speed.

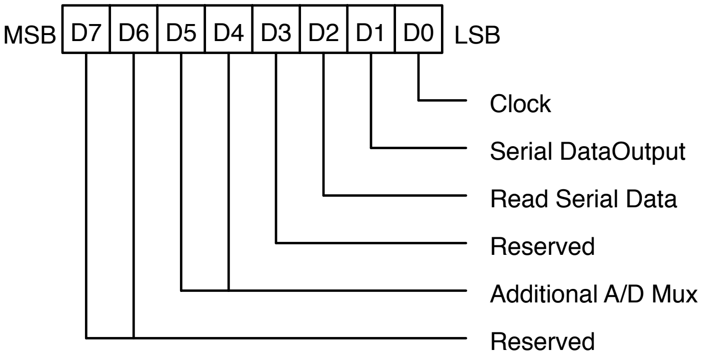
Bit 2 RESERVED.

Bit 4:3 SoundBlaster ADPCM Mode. Selects the SoundBlaster ADPCM playback.
00 = Silence is output (i.e., mute) during SoundBlaster ADPCM playback.
01 = Silent is played for SoundBlaster 3:1 and 4:1 ADPCM. The cs4231 ADPCM is used for 2:1.
10 = An interrupt is generated when SB ADPCM is detected. (see CS4232 configuration for how to assign interrupt level.)
11 = RESERVED.

Bit 5 Control host interrupt generation when a context switch occurs.
0 = No interrupt on context switch.
1 = CS4232 interrupt generated on context switch.

Bit 7:6 Power Management control for CS4232.
00 = All functions active
01 = Codec digital logic powered down. Some register values destroyed
10 = Sound Blaster and MPU–401 functions powered down. All Plug and Play configuration is retained.
11 = Full chip power down. Same as power on reset.

11.4.8 Control Register 1



Bit 0 This bit is used to generate the clock for the Plug and Play EEPROM.

Bit 1 This bit is used to output serial data to the Plug and Play EEPROM.

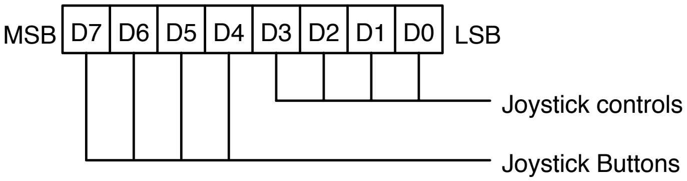
Bit 2 This bit is used to read serial data from the Plug and Play EEPROM and to enable the CLK and DOUT onto the CS4232 pins.
0 = EEPROM interface disabled.
1 = EEPROM interface enabled.

Bit 3 RESERVED.

Bit 5:4 These two bits are used to control an additional A/D mux and enable for an analog loopback path.
00 = EEPROM interface enabled.
01 = Codec Input mux is mixed into output mixer. A/D input is from the input mix.
10 = Codec Input mux is mixed into output mixer. A/D input is from line out puts.
11 = RESERVED.

Bit 7:6 RESERVED.

11.4.9 Joystick Port



Bit 0 Joystick A Control X.

Bit 1 Joystick A Control Y.

Bit 2 Joystick B Control X.

Bit 3 Joystick B Control Y.

Bit 4 Joystick A Button 1.

Bit 5 Joystick A Button 2.

Bit 6 Joystick B Button 1.

Bit 7 Joystick B Button 2.

The joystick consist of one register. Write to it to set the one–shots. Then read at intervals to see how long they take to reset. The buttons can be read at any time. The address of the joystick port is set during the configuration of the CS4232 (see CS4232 configuration).

11.4.10 SoundBlaster Registers

The base address of the SoundBlaster registers is set up in the Plug and Play setup. All addresses found below need to be added to the Base Address setup in the configuration.

Table 61. SoundBlaster Addresses

Address	Description
00 – 03	**Left/Right FM registers.** These registers are inside the OPL3. See the OPL3 spec for their function.
04	**Mixer Index register**
05	**Mixer Data register**
06	**Reset (write only).** setting D[0] (little endian) to 1 and then to 0 resets the Sound Blaster logic
0A	**Read Data Port (read only).** When D[7] (little endian) is set = 1, then valid data is available in this register. the data may be the result of a Command that was previously written to the Command/Write register or audio data.
0C	**Command/Write Data (write only).** The command/write data register is used to send the Sound Blaster commands to the CS4232.
0C	**Write Buffer Status (read only).** The write buffer status register bit D[7] (little endian) indicates when the CS4232 is ready to accept another command to the command/Write Data register. D[7]= 1 indicates ready. D[7]=0 indicates not ready.

Mixer Registers (Little Endian)

Register	D7	D6	D5	D4	D3	D2	D1	D0
00h	Data Reset							
04h	Reserved							
04h	Volume Left				Volume Right			
06h	Reserved							
08h	Reserved							
0Ah	X	X	X	X	X	X	Mic Mix	
0Ch	X	X	Input Filter			Input Select		X
0Eh	X	X	DNF1	X	X	X	VSTC	X
20h	Reserved							
22H	Master Volume Left				Master Volume Right			
24h	Reserved							
26h	FM Volume Left				FM Volume Right			
28h	CD Volume Right				CD Volume Right			
2Ah	Reserved							
2Ch	Reserved							
2Eh	Line Volume Right				Line Volume Right			

Input Filter (Little Endian)

Filter	D5	D4	D3
Low Filter	0	x	0
High Filter	0	x	1
No Filter	1	x	x

Input Select

Input	B2	B1
Microphone	x	0
CD Audio	0	1
Line-In	1	1

11.4.11 MPU–401 MIDI

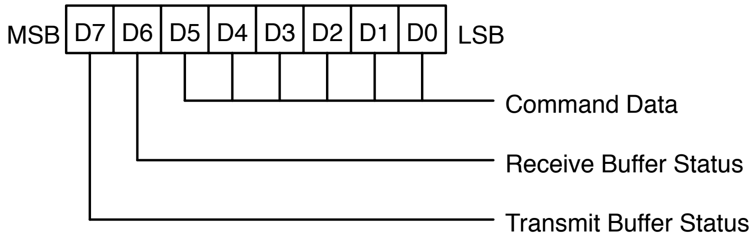
The MPU-401 is an intelligent MIDI interface. There is also a non-intelligent mode where it basically acts as a UART.

The MPU-401 interface consist of two registers that are mapped into the PC I/O address space. The address is determined by the configuration (see CS4232 configuration). The standard address for MIDI Transmit/Receive is port 330h. Port 331h is usually used for the Command/Status port.

Address + Base	Description
0 (Read)	Receive Port
0 (Write)	Transmit Port
1 (Read)	Status Port
1 (write)	Command Port

An interrupt can be assigned to tell when data has been received at the Receive port (see CS4232 configuration).

11.4.11.1 MIDI Status Register



Bits 5:0 These bits are the data of the last command written
0 = EEPROM interface disabled.
1 = EEPROM interface enabled.

Bit 6 CXS—Transmit buffer status flag.
0 = not full.
1 = full, wait.

Bit 7 RXS—Receive buffer status flag.
0 = not empty.
1 = empty, wait. .

When an interrupt is generated by the MPU–401, it is cleared in the CS4232 by a read of the MIDI Receive Port. This is the same port as the Base address for this logical device set in the configuration (see CS4232 configuration).

Section 12
System Firmware

12.1 Introduction

The firmware on the PowerPC 604 SMP reference design handles three major functions:

- Test the system in preparation for execution
- Load and execute an executable image from a bootable device
- Allow user configuration of the system.

Section 12.2 briefly discusses the power on system test function.

Section 12.3 details a structure for boot records which can be loaded by the system firmware.

Section 12.4 describes the system configuration utility.

To obtain a copy of the commented source code of the firmware on diskette, contact your IBM representative. This material is available free of charge with a signed license agreement.

12.2 Power On System Test

The Power On System Test (POST) code tests those subsystems of the reference board which are required for configuration and boot to ensure minimum operability. Tests also assure validity of the firmware image and of the stored system configuration.

12.3 Boot Record Format

The firmware will attempt to boot an executable image from devices specified by the user. See section 12.4 for details on specifying boot devices and order.

The *PowerPC Reference Platform Specification* details a structure for boot records which can be loaded by the system firmware. This specification is described in the following sections.

12.3.1 Boot Record

The format of the boot record is an extension of the PC environment. The boot record is composed of a PC compatibility block and a partition table. To support media interchange, the PC compatibility block may contain an x86-type program. The entries in the partition table identify the PowerPC Reference Platform boot partition and its location in the media.

The layout of the boot record must be designed as shown in Figure 51. The first 446 bytes of the boot record contain a PC compatibility block, the next four 16-byte entries make up a partition table totalling 64 bytes, and the last 2 bytes contain a signature.

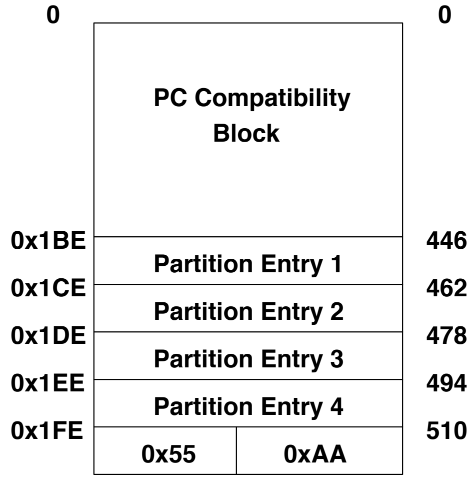


Figure 51. Boot Record

12.3.1.1 PC Partition Table Entry

To support media interchange with the PC, the PowerPC Reference Platform defines the format of the partition table entry based on that for the PC. This section describes the format of the PC partition table entry, which is shown in Figure 52.

Partition Begin	0	Boot Ind	Head	Sector	Cyl
Partition End	4	Sys Ind	Head	Sector	Cyl
Beginning Sector	8	Low Word (LE)		High Word (LE)	
Number of Sectors	12	Low Word (LE)		High Word (LE)	

LE = Little-Endian

Figure 52. Partition Table Entry

- Partition Begin — The beginning address of the partition in head, sector, cylinder notation.
- Partition End — The end address of the partition in cylinder, head, sector notation.

- Beginning Sector — The number of sectors preceding the partition on the disk. That is, the zero-based relative block address of the first sector of the partition.
- Number of Sectors — The number of sectors allocated to the partition.

The subfields of a partition table entry are defined as follows:

- Boot Ind — Boot Indicator. This byte indicates whether the partition is active. If the byte contains 0x00, then the partition is not active and is not considered bootable. If the byte contains 0x80, then the partition is considered active.
- Head — An eight-bit value, zero-based.
- Sector — A six-bit value, one-based. The low-order six bits are the sector value. The high-order two bits are the high-order bits of the 10-bit cylinder value.
- Cyl — Cylinder. The low-order eight-bit component of the 10-bit cylinder value (zero-based). The high-order two bits of the cylinder value are found in the sector field.
- Sys Ind — System Indicator. This byte defines the type of the partition. There are numerous partition types defined. For example, the following list shows several:

0x00	Available partition
0x01	DOS, 12-bit FAT
0x04	DOS, 16-bit FAT
0x05	DOS extended partition
0x41	PowerPC Reference Platform partition.

12.3.1.2 Extended DOS Partition

The extended DOS partition is used to allow more than four partitions in a device. The boot record in the extended DOS partition has a partition table with two entries but does not contain the code section. The first entry describes the location, size and type of the partition. The second entry points to the next partition in the chained list of partitions. The last partition in the list is indicated with a system indicator value of zero in the second entry of its partition table.

Because of the DOS format limitations for a device partition, a partition which starts at a location beyond the first 1 gigabyte is located by using an enhanced format shown in Figure 53.

Partition Begin	**0**	**Boot Ind**	**–1**	**–1**	**–1**
Partition End	**4**	**Sys Ind**	**–1**	**–1**	**–1**
Beginning Sector	**8**	**32–bit start RBA (zero–based) (LE)**			
Number of Sectors	**12**	**32–bit RBA count (one–based) (LE)**			

LE = Little-Endian

–1 = All ones in the field.
RBA = Relative Block Address in units of 512 bytes.

Figure 53. Partition Table Entry Format for an Extended Partition

12.3.1.3 PowerPC Reference Platform Partition Table Entry

The Power PC Reference Platform partition table entry (see Figure 54) is identified by the 0x41 value in the system indicator field. All other fields are ignored by the firmware except for the Beginning Sector and Number of Sectors fields. The CV (Compatible Value—not shown) fields must contain PC-compatible values (i.e. acceptable to DOS) to avoid confusing PC software. The CV fields, however, are ignored by the firmware.

Partition Begin	**0**	**Boot Ind**	**Head**	**Sector**	**Cyl**
Partition End	**4**	**Sys Ind**	**Head**	**Sector**	**Cyl**
Beginning Sector	**8**	**32–bit start RBA (zero–based) (LE)**			
Number of Sectors	**12**	**32–bit RBA count (one–based) (LE)**			

RBA = Relative Block Address in units of 512 bytes.
LE = Little-Endian

Figure 54. Partition Table Entry for PowerPC Reference Platform

The 32-bit start RBA is zero-based. The 32-bit count RBA is one-based and indicates the number of 512-byte blocks. The count is always specified in 512-byte blocks even if the physical sectoring of the target devices is not in 512-byte sectors.

12.3.2 Loading the Load Image

This section describes the layout of the PowerPC 0x41 type partition and the process of loading the load image.

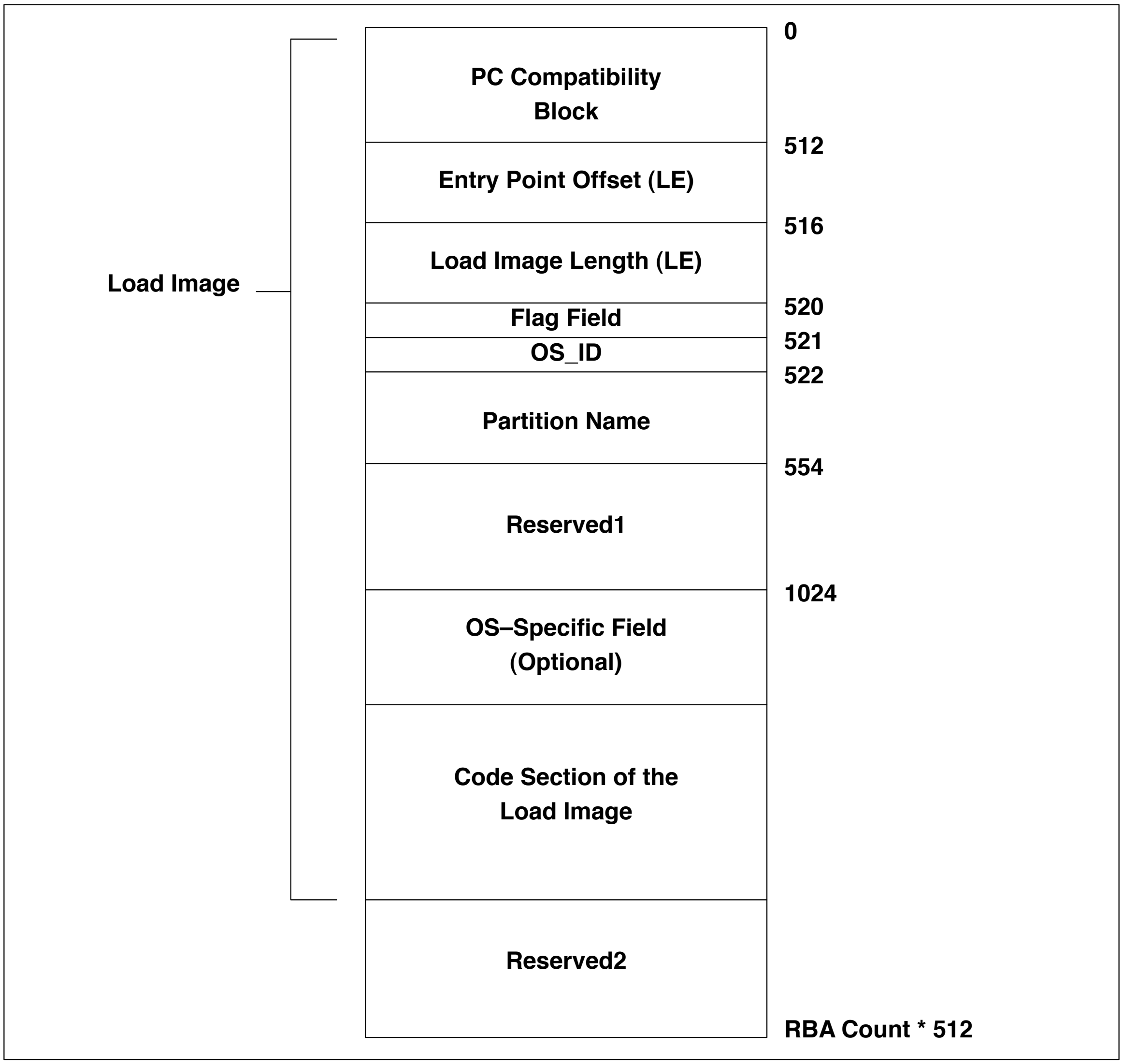


Figure 55. PowerPC Reference Platform Partition

The layout for the 0x41 type partition is shown in Figure 55. The PC compatibility block in the boot partition may contain an x86-type program. When executed on an x86 machine, this program displays a message indicating that this partition is not applicable to the current system environment.

The second relative block in the boot partition contains the entry point offset, load image length, flag field, operating system ID field, ASCII partition name field, and the reserved1 area. The 32-bit entry point offset (little-endian) is the offset (into the image) of the entry point of the PowerPC Reference Platform boot program. The entry point offset is used to allocate the Reserved1 space. The reserved1 area from offset 554 to Entry Point 1 is reserved for implementation specific data and future expansion.

The 32-bit load image length (little-endian) is the length in bytes of the load image. The load image length specifies the size of the data physically copied into the system RAM by the firmware.

The flag field is 8 bits wide. The MSb in the field is allocated for the Open Firmware flag. If this bit is set to 1, the loader requires Open Firmware services to continue loading the operating system.

The second MSb is the endian mode bit. If the mode bit is 0, the code in the section is in big-endian mode. Otherwise, the codes is in little-endian mode. The implication of the endian mode bit is different depending on the Open Firmware flag. If the Open Firmware flag is set to 1, the mode bit indicates the endian mode of the code section pointed to by the load image offset, and the firmware has to establish the hardware endian mode according to this bit. Otherwise, this bit is just an informative field for firmware.

The OS_ID field and partition name field are used to identify the operating system located in the partition. The OS_ID field has the numeric identification value of the operating system located in the partition. The 32 byes of partition name field must have the ASCII notation of the partition name. The name and OS_ID can be used to provide to a user the identification of the boot partition during the manual boot process.

Once the boot partition is identified by the PowerPC Reference Platform boot partition table entry, the firmware:

- Reads into memory the second 512-byte block of the boot partition
- Determines the load image length for reading in the boot image up to but not including the reserved2 space
- Allocates a buffer in system RAM for the load image transfer (no fixed location)
- Transfers the load image into system RAM from the boot device (the reserved2 space is not loaded).

The load image must be fully relocatable, as it may be placed anywhere in memory by the system firmware. Once loaded, the load image may relocate itself anywhere within system RAM.

12.4 System Configuration

This section describes the utilities in the system firmware which allow the system to be customized. These utilities allow viewing of the system configuration and changing of I/O device configurations, console selection, boot devices, and the date and time. These functions are described in the following sections.

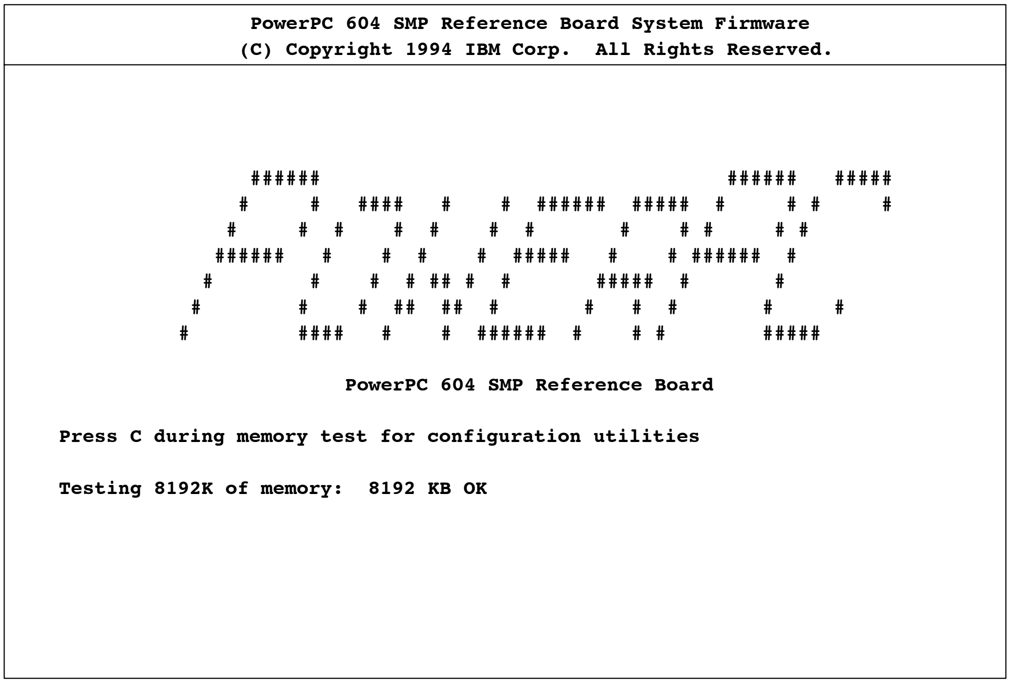
12.4.1 System Console

The system console can be either a screen-oriented video display or a line-oriented serial terminal. The example screens in this section show the S3 video/keyboard interface. When using a serial terminal, the configuration utilities will prompt for numeric input for each prompt instead of using the arrow keys. All choices and options are the same as for the screen-oriented menus.

The configuration of the reference board as shipped is set for S3 video/keyboard console. In the case where either the video adapter or the keyboard fails the power-on test, the system console will default to serial port 1. The baud rate for the serial console is specified in the configuration menus. The value as shipped is 9600 baud.

12.4.2 System Initialization

The logo screen, shown in Figure 56, is displayed at power-on. The logo screen is active while the system initializes and tests memory and performs a scan of the SCSI bus to determine what SCSI devices are installed.



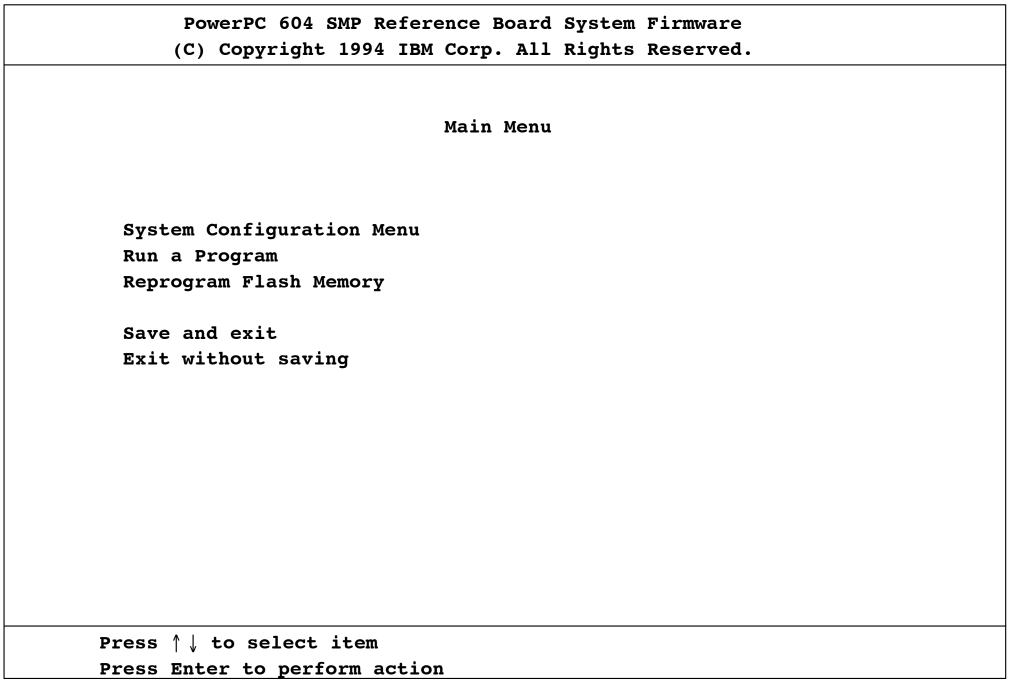
```
                    PowerPC 604 SMP Reference Board System Firmware
                     (C) Copyright 1994 IBM Corp.  All Rights Reserved.


                ######                                  ######   ######
               #     #   ####   #    #  ######  #####  #     # #     #
              #     #  #    #  #    #  #       #    # #     # #
             ######   #    #  #    #  #####   #    # ######  #
            #        #    #  # ## #  #       #####  #       #
           #        #    #  ##  ##  #       #   #  #       #     #
          #         ####   #    #  ######  #    # #        #####

                         PowerPC 604 SMP Reference Board

Press C during memory test for configuration utilities

Testing 8192K of memory:  8192 KB OK
```

Figure 56. System Initialization Screen

While the logo screen is displayed, pressing the 'C' key on the console will enter the system configuration utility. The configuration menu will also be entered if there is no bootable device present, or if the configuration stored in the system non-volatile RAM is not initialized or is corrupt.

12.4.3 Main Menu

Figure 57 shows the Main Menu for the system configuration utility. Selections on the menu are highlighted by using the up and down arrow keys on the keyboard, and are chosen with the Enter key. Each choice is detailed in the following sections.

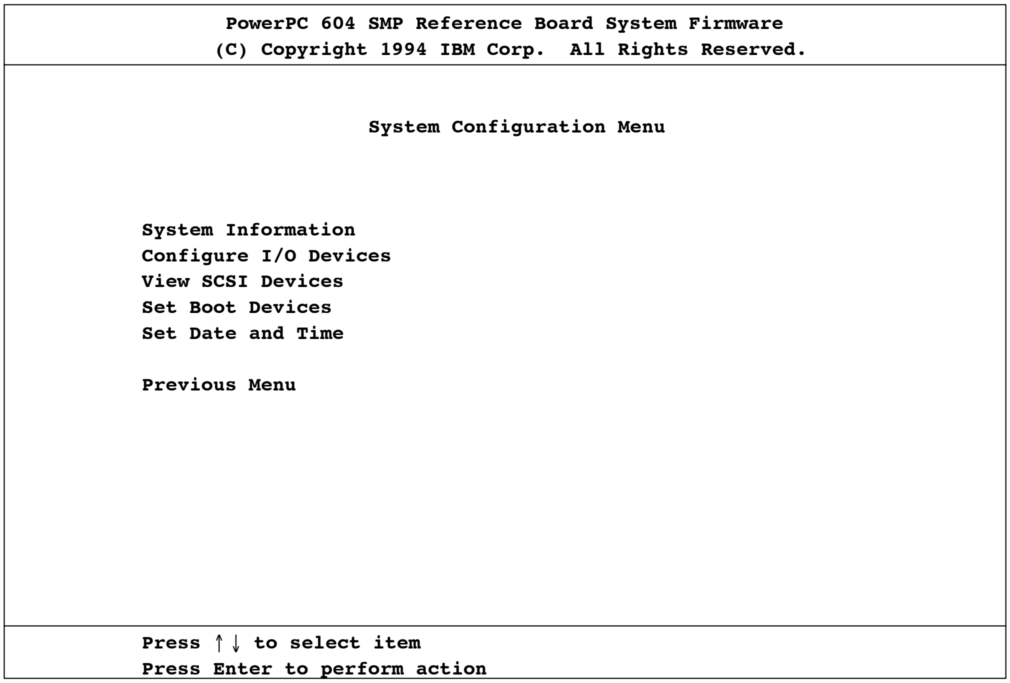


```
          PowerPC 604 SMP Reference Board System Firmware
         (C) Copyright 1994 IBM Corp. All Rights Reserved.


                              Main Menu



       System Configuration Menu
       Run a Program
       Reprogram Flash Memory

       Save and exit
       Exit without saving


     Press ↑ ↓ to select item
     Press Enter to perform action
```

Figure 57. Configuration Utility Main Menu

12.4.3.1 System Configuration Menu

Figure 58 shows the System Configuration Menu, which has choices to display and change the default state of the reference board on boot. Each menu item is discussed in the following sections.

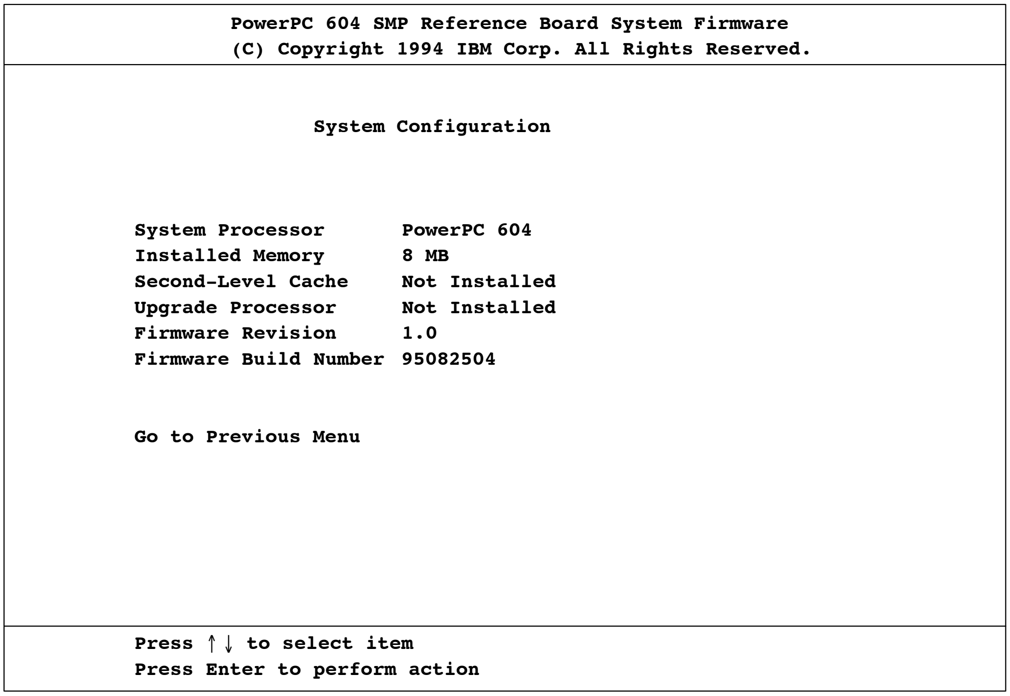


```
                 PowerPC 604 SMP Reference Board System Firmware
                (C) Copyright 1994 IBM Corp.  All Rights Reserved.

                            System Configuration Menu

     System Information
     Configure I/O Devices
     View SCSI Devices
     Set Boot Devices
     Set Date and Time

     Previous Menu

     Press ↑↓ to select item
     Press Enter to perform action
```

Figure 58. System Configuration Menu

System Information

The System Configuration option shows the hardware configuration of the system at power-up—including processor, installed options, and firmware revision level. A sample screen is shown in Figure 59.

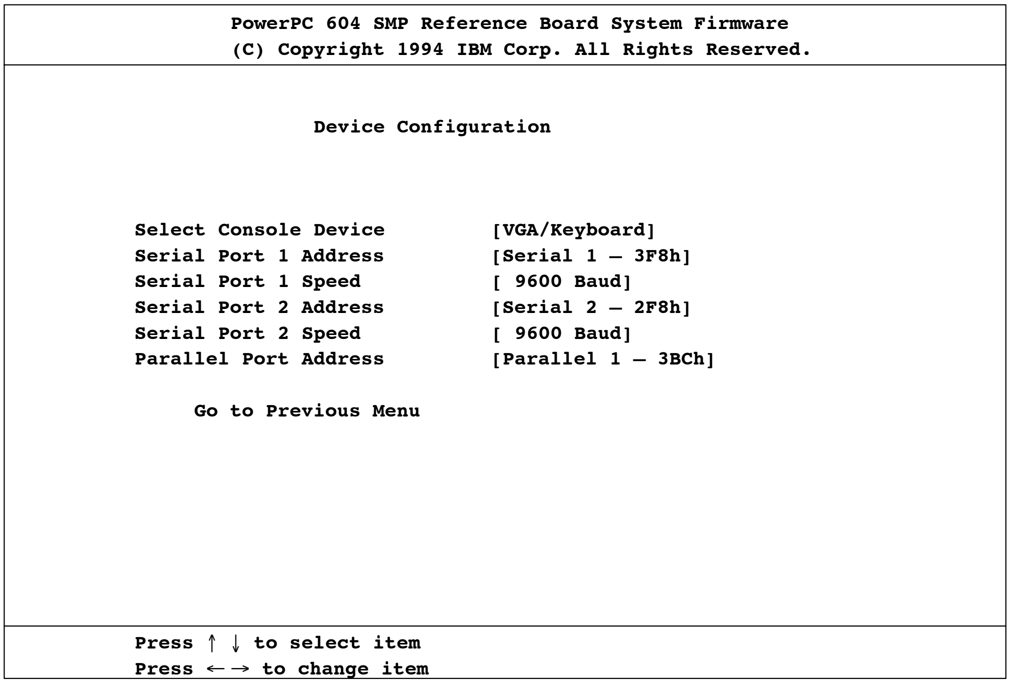


```
                PowerPC 604 SMP Reference Board System Firmware
                (C) Copyright 1994 IBM Corp. All Rights Reserved.

                       System Configuration

        System Processor       PowerPC 604
        Installed Memory       8 MB
        Second-Level Cache     Not Installed
        Upgrade Processor      Not Installed
        Firmware Revision      1.0
        Firmware Build Number  95082504

        Go to Previous Menu

        Press ↑↓ to select item
        Press Enter to perform action
```

Figure 59. System Information Screen

Configure I/O Devices

The configure I/O devices option allows the customization of system I/O ports and the system console. The menu is shown in Figure 60. Options are highlighted by using the up and down arrow keys on the keyboard and are changed with the left and right arrow keys. Options on the menu are discussed below.



```
                 PowerPC 604 SMP Reference Board System Firmware
                 (C) Copyright 1994 IBM Corp. All Rights Reserved.

                        Device Configuration

          Select Console Device          [VGA/Keyboard]
          Serial Port 1 Address          [Serial 1 – 3F8h]
          Serial Port 1 Speed            [ 9600 Baud]
          Serial Port 2 Address          [Serial 2 – 2F8h]
          Serial Port 2 Speed            [ 9600 Baud]
          Parallel Port Address          [Parallel 1 – 3BCh]

               Go to Previous Menu

          Press ↑ ↓ to select item
          Press ← → to change item
```

Figure 60. Device Configuration Screen

Any changes made in I/O device configuration are saved when the Save and Exit option on the main menu is selected. Exiting the system configuration utility in any other manner will cause device configuration changes to be lost.

Select Console Device

The console selection box allows the selection of an option for the system console.

- Serial Port 1 or 2 — Console input and output will be transmitted and received through a serial port on an adapter card. Console input and output will be transmitted and received at the baud rate selected with Serial Port Speed.
- S3 Video/Keyboard — Console output will be displayed on a video monitor connected to an S3 PCI video adapter; console input will be received from a keyboard connected to the keyboard connector on the reference board.

Serial Port 1 or 2 Address

Each of the two serial ports on the Reference Board can be accessed at any one of ten different base addresses. The serial port address selection box allows the user to specify which base address to use for each port. The selection boxes enforce the restriction that base addresses for the serial ports must not conflict. Changing the base address for a serial port does not affect its use as the system console. This option may also be used to disable either or both serial ports.

Serial Port 1 or 2 Speed

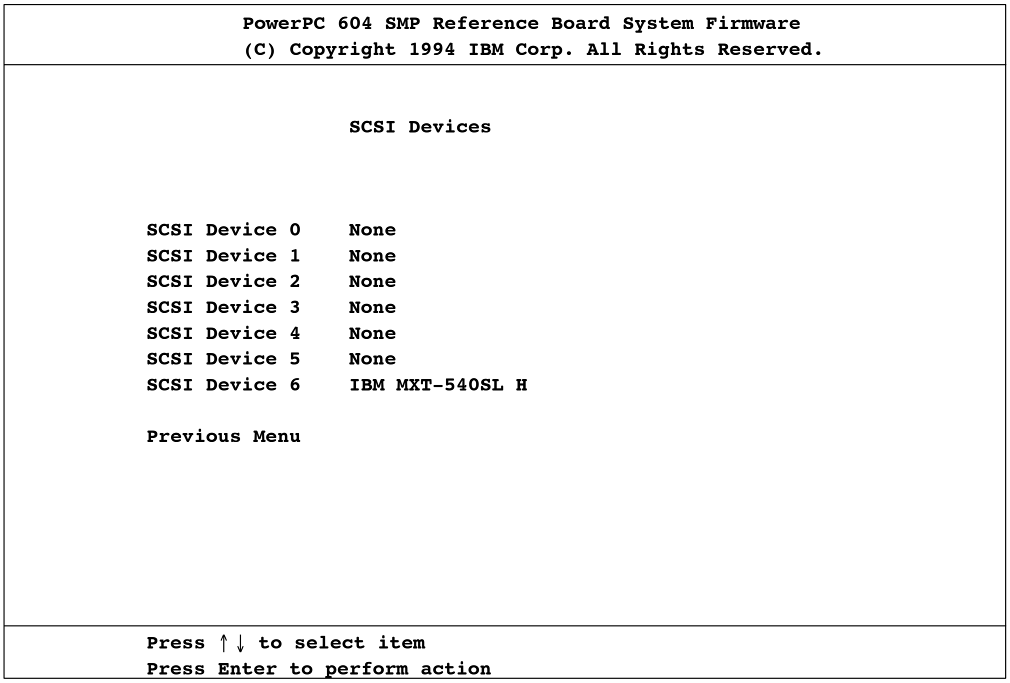
The serial port speed selection box sets the speed of each serial port. Baud rates for the two serial ports are independent. If a serial port is used as the system console, set this value to match the baud rate of the terminal.

Parallel Port Address

The parallel port address selection box sets the base address of the Reference Board's parallel port to any of three predefined addresses. This option may also be used to disable the parallel port.

View SCSI Devices

The SCSI Devices screen shows the devices found on the SCSI bus during power-on initialization. The string shown is the SCSI device's response to the SCSI inquiry command. According to the SCSI specification, this data comprises the manufacturer's ID, device model number, and device revision level. A sample screen is shown in Figure 61.

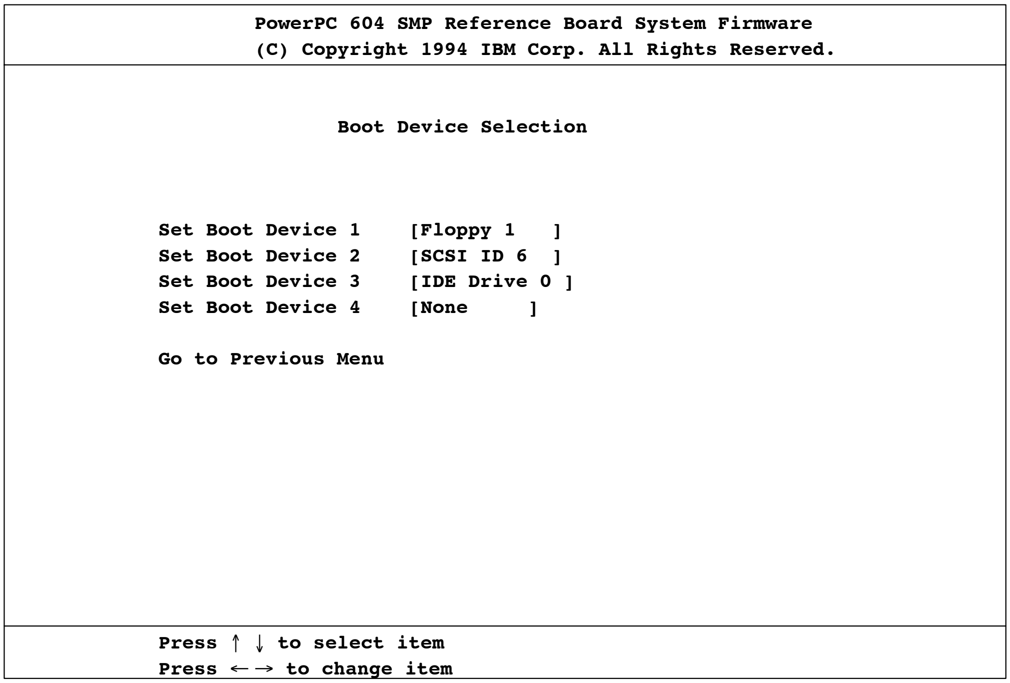


```
              PowerPC 604 SMP Reference Board System Firmware
              (C) Copyright 1994 IBM Corp. All Rights Reserved.

                               SCSI Devices

            SCSI Device 0    None
            SCSI Device 1    None
            SCSI Device 2    None
            SCSI Device 3    None
            SCSI Device 4    None
            SCSI Device 5    None
            SCSI Device 6    IBM MXT-540SL H

            Previous Menu

            Press ↑↓ to select item
            Press Enter to perform action
```

Figure 61. SCSI Devices Screen

Set Boot Devices

The Boot Device Selection menu allows the user to select which devices are queried for boot images and in what order they are selected for boot. Allowable selections are one of the two floppy disk drives, any of six SCSI drive ID numbers, either of two IDE disk drives, or no device selected. The default configuration is shown in Figure 62. In this configuration, the system will attempt to find a boot image on the first floppy disk drive. If this fails, the system will attempt to boot from the SCSI device programmed to SCSI ID 6. If this fails, the system will attempt to boot fro IDE drive zero (master).



```
                PowerPC 604 SMP Reference Board System Firmware
                (C) Copyright 1994 IBM Corp. All Rights Reserved.

                       Boot Device Selection



        Set Boot Device 1    [Floppy 1   ]
        Set Boot Device 2    [SCSI ID 6  ]
        Set Boot Device 3    [IDE Drive 0 ]
        Set Boot Device 4    [None      ]

        Go to Previous Menu









        Press ↑ ↓ to select item
        Press ← → to change item
```

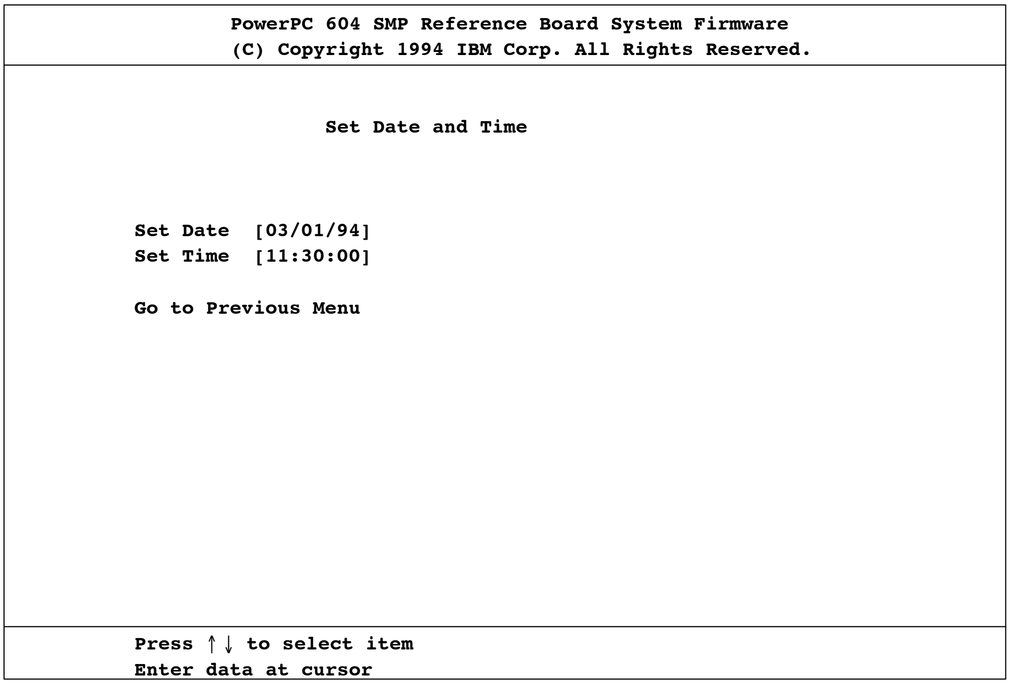
Figure 62. Boot Devices Screen

If the system fails to find a valid boot image (as discussed in section 12.3) on any of the selected boot devices, or if no boot device is selected, the user will be prompted to enter the configuration menu to select a valid boot device.

Any changes made in boot device selection is saved when the Save and Exit option on the main menu is selected. Exiting the system configuration utility in any other manner will cause boot device changes to be lost.

Set Date and Time

The Set Date and Time screen allows the date and time stored in the battery-backed real time clock to be updated. The screen is shown in Figure 63. To change the time, the left and right arrow keys are used to select the digit to modify, and the digit is then typed over with the number keys. The date or time will be updated when Enter or either the up or down arrow is pressed. Changing the date or time is immediate, and is not affected by either the Save and Exit or Exit Without Saving options on the main menu.



```
              PowerPC 604 SMP Reference Board System Firmware
              (C) Copyright 1994 IBM Corp. All Rights Reserved.

                      Set Date and Time

        Set Date  [03/01/94]
        Set Time  [11:30:00]

        Go to Previous Menu

        Press ↑↓ to select item
        Enter data at cursor
```

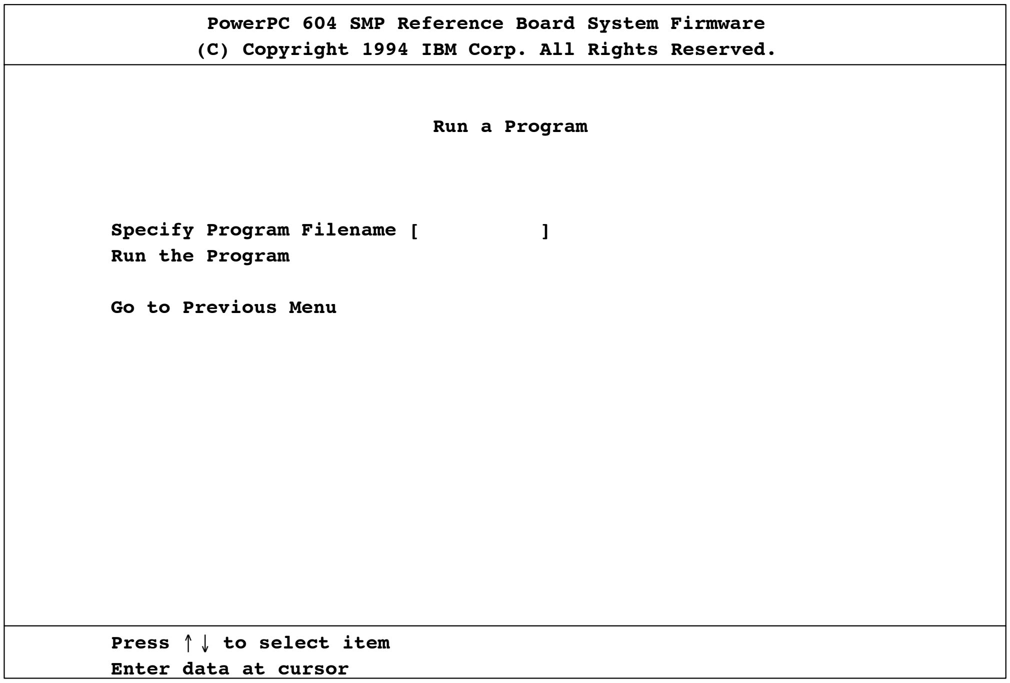
Figure 63. Set Date and Time Screen

12.4.3.2 Run a Program

The Run a Program option on the main menu loads and executes a program from a FAT (DOS) disk or from a CD-ROM in ISO-9660 format. The program is loaded at location 0x00400000 (4 M), and control is passed with a branch to the first address.

All boot devices specified in the Boot Devices Menu will be searched in order for FAT and CD-ROM file systems, and the first matching file on a boot device will be loaded.

The Run a Program screen is shown in Figure 64. To run a program, enter the file name in the Specify Program Filename field and select the Run the Program option.



```
                PowerPC 604 SMP Reference Board System Firmware
               (C) Copyright 1994 IBM Corp. All Rights Reserved.

                                 Run a Program


         Specify Program Filename [          ]
         Run the Program

         Go to Previous Menu












         Press ↑↓ to select item
         Enter data at cursor
```

Figure 64. Run a Program Screen

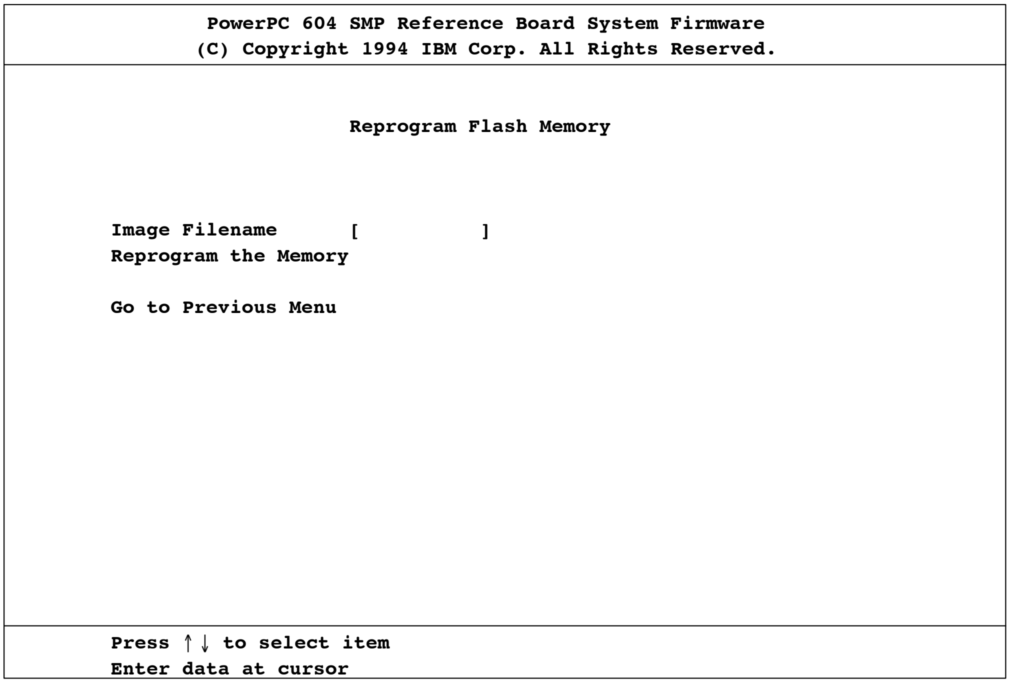
12.4.3.3 Reprogram Flash Memory

The PowerPC 604 SMP reference board stores its system firmware in a reprogrammable flash memory on the system board. The Reprogram Flash Memory option on the main menu allows the reprogramming of the flash device with a DOS-formatted diskette. This allows future revisions of the system firmware to be provided on diskette without the need for removal of the device from the board.

If done improperly, reprogramming the flash memory can cause the system to become unusable until external means are available to reprogram the device. Use this option with care.

All boot devices specified in the Boot Devices Menu will be searched in order for FAT and CD-ROM file systems, and the first matching file on a boot device will be loaded.

The Reprogram Flash Memory screen is shown in Figure 65. To reprogram the flash, enter the file name in the Image Filename field and select the Reprogram the Memory option.



```
                PowerPC 604 SMP Reference Board System Firmware
               (C) Copyright 1994 IBM Corp. All Rights Reserved.

                           Reprogram Flash Memory


         Image Filename      [          ]
         Reprogram the Memory

         Go to Previous Menu

         Press ↑↓ to select item
         Enter data at cursor
```

Figure 65. Reprogram the Flash Memory Screen

12.4.3.4 Exit Options

The two exit options at the bottom of the main menu leave the system configuration utility. The two options are:

- Save and Exit — Saves any changes made in the Configure I/O Devices and Set Boot Devices screens, and restarts the system.
- Exit without Saving — Proceeds with the boot process as if the configuration utility had not been entered. Any changes made in Configure I/O Devices or Set Boot Devices are lost.

12.4.4 Default Configuration Values

When the PowerPC 604 SMP reference board is shipped from the factory, it has the following default configuration:

• Console Device	S3 Video/Keyboard
• Serial Port 1	Address Serial 1—3F8h, 9600 Baud
• Serial Port 2	Address Serial 2—2F8h, 9600 Baud
• Parallel Port	Parallel 1—3BCh
• Boot Devices	Device 1 - Floppy 1 Device 2 - SCSI ID 6 Device 3 - IDE Drive 0

These default values also take effect whenever the system configuration in system nonvolatile RAM becomes corrupted.

Section 13
Registers and System Setup

13.1 CPU Initialization

The 604 CPU exits the reset state with the L1 cache disabled and bus error checking disabled.

All memory pages 2G to 4G must be marked as non-cacheable.

The segment register T bit (bit 0) defaults to 0, which is the normal storage access mode. It must be left in this state for the hardware to function. Direct store (PIO) segments are not supported.

Set the bit that controls ARTRY# negation, HID0[7], to 0 to enable the precharge of ARTRY#. It may be necessary set HID0[7] to 1 to disable the precharge of ARTRY# for reference design configurations having a CPU bus agent (such as an added L2) that drives the ARTRY# line. Software must set this bit before allowing any CPU bus traffic to which the CPU agent might respond. Note that PCI to memory transactions cause the 660 bridge to broadcast snoop operations on the CPU bus.

HID0 bit 0, Master Checkstop Enable, defaults to 1, which is the enabled state. Leave it in this state so that checkstops can occur.

Reference design errors are reported through the 660 bridge by way of the TEA# and MCP# pins. Because of this, the bus error checking in the CPU must be disabled by setting HID0 bits 2 and 3 to zero (in the 604 CPU, enable L1 cache parity checking by setting HID0 bit 1 to one).

13.1.1 Fast L2/Data Streaming Mode (No-DRTRY#)

The reference design default configuration is for DRTRY# mode. For use with CPU cards that use no-DRTRY# (fast L2/data streaming) mode, the reference motherboard can be reconfigured. Remove R232 to isolate the DBB# on each CPU card from the other CPU card. The installed CPU cards must each provide a pullup on DBB.

In no-DRTRY# mode, each CPU card must also provide correct generation of DRTRY# for the CPU on the card. For the PowerPC 604 CPU card, wire DRTRY# of the CPU to #HRESET of the connector (rather than to connector DRTRY#). This will provide a low level on the CPU DRTRY# at reset and a high level under normal operation, which will place the 604 CPU in fast L2/data streaming mode.

13.2 660 Bridge Initialization

Before DRAM memory operations can begin, the software must:

1 Read the SIMM presence detect and SIMM type registers.

2 Set up and check the memory-related registers in the 660 bridge (see the 660 Bridge User's Manual).

3 Program the timer in the ISA bridge register which controls ISA refresh timing. In SIO compatible bridges, it should be programmed to operate in Mode 2 with an interval of approximately 15 usec.

4 Make sure 200 usec has elapsed since starting the refresh timer so that sufficient refresh cycles have occurred to properly start the memory. This will be hidden if approximately 120 Flash accesses occur after the timer is started and before the memory initialization starts.

5 Initialize all of memory so that all parity bits are properly set. (The CPU may cache unnecessary data; hence, all of memory must be initialized.) The 660 bridge does not require reconfiguration when port 4Dh in the ISA bridge is utilized to reset the native I/O and the ISA slots.

13.2.1 CPU to PCI Configuration Transactions

The reference design allows the CPU to generate type 0 and type 1 PCI configuration cycles. The CPU initiates a transfer to the appropriate address, and the 660 Bridge decodes the cycle and generates a request to the PCI arbiter in the SIO. When the PCI bus is acquired, the 660 Bridge enables its PCI_AD drivers and drives the address onto the PCI_AD lines for one PCI clock before it asserts PCI_FRAME#. Predriving the PCI_AD lines for one clock before asserting PCI_FRAME#, allows the IDSELs to be resistively connected to the PCI_AD[31:0] bus at the system level. Table 62 shows the reference design IDSEL map.

The transfer size must match the capabilities of the target PCI device for configuration cycles. The reference design supports 1-, 2-, 3-, and 4-byte transfers that do not cross a 4-byte boundary.

Address unmunging and data byte swapping follows the same rules as for system memory with respect to BE and LE modes of operation. Address unmunging has no effect on the CPU address lines which correspond to the IDSEL inputs of the PCI devices.

Table 62. IDSEL Assignments

Device	ID Sel Line	60X Address*	PCI Address
82378ZB PCI to ISA Bridge	A/D 11	8080 08XXh	080 08XX
53C810 SCSI Controller	A/D 12	8080 10XXh	080 10XX
PCI Slot 1 (lower)	A/D 13	8080 20XXh	080 20XX
PCI Slot 2 (upper)	A/D 14	8080 40XXh	080 40XX
Multiprocessor Interrupt Controller	A/D 15	8080 80XXh	080 80XX
79C970 Ethernet Controller	A/D 16	8081 00XXh	081 00XX

Note: *This address is independent of contiguous I/O mode.

13.2.1.1 Preferred Method of Generating PCI Configuration Transactions

The preferred method for generating PCI configuration cycles is via the 660 Bridge indexed Bridge Control Registers (BCRs). This configuration method is described in Section 4 of the 660 User's Manual. The IDSEL assignment, address assignment, and the PCI_AD lines are shown here in Table 63.

13.2.1.2 650 Bridge Compatible Method

If it is not possible to use indexed BCRs to generate PCI configuration cycles, they can be generated by an alternate method known as the 650 bridge compatible method. CPU accesses to the address range 2G+8M to 2G+16M cause the bridge to arbitrate for the PCI bus and then to execute a type 0 PCI configuration transaction as described in the PowerPC Reference Platform Specification and implemented by the IBM27-82650 PowerPC to PCI Bridge. This is referred to as the 650 compatible configuration method. This method of accessing PCI configuration space does not allow access to the PCI configuration registers in the bridge chip. It should not be used unless required to maintain 650 compatibility.

When using the 650 bridge compatible configuration method, use only the specified address. Using other addresses can cause bus contention because multiple PCI slots could be selected. For example, using any CPU address in the range 8080 0000 to 80FF FFFF with both AD11 = 1 and AD12 = 1, causes selection of both the SIO and any device in slot 1, possibly resulting in damage.

13.2.2 PCI Configuration Scan

The reference design enables the software to implement a scan to determine the complement of PCI devices present. This is because the system returns all ones rather than an error when no PCI device responds to initialization cycles. The software may read each possible PCI device ID to determine devices present.

Software must use only the addresses specified. Using any addresses that causes more than one IDSEL to be asserted (high) can cause bus contention, because multiple PCI agents will be selected.

13.2.2.1 Multi-Function Adaptors

The 660 Bridge supports multi-function adapters. It passes the address of the load or store instruction that causes PCI configuration cycle unmodified (except that the three low-order bits are unmunged in little endian mode and the two low-order address bits are set to zero in either endian mode). Addresses, therefore, may be selected with non-zero CPU address bits (21:23)—corresponding to PCI bits (10:8)—to configure multi-function adaptors. For example, to configure device 3 in slot 1, use address 80C0 03XXh. To configure device 7 in slot 2, use address 8084 07XXh.

13.2.2.2 PCI to PCI Bridges

The 660 bridge supports both Type 0 and Type 1 configuration cycles.

13.2.3 660 Bridge Indexed BCR Summary

Table 63 contains a summary listing of the indexed BCRs. Access to these registers is described in the 660 Bridge User's Manual.

Table 63. 660 Bridge Indexed BCR Listing

Bridge Control Register	Index	R/W	Bytes	Set To[1]	Note
PCI Vendor ID	Index 00-01	R	2	—	
PCI Device ID	Index 02-03	R	2	—	
PCI Command	Index 04-05	R/W	2	—	
PCI Device Status	Index 06-07	R/W	2	—	
Revision ID	Index 08	R	1	—	
PCI Standard Programming Interface	Index 09	R	1	—	
PCI Subclass Code	Index 0A	R	1	—	
PCI Class Code	Index 0B	R	1	—	
PCI Cache Line Size	Index 0C	R	1	—	
PCI Latency Timer	Index 0D	R	1	—	
PCI Header Type	Index 0E	R	1	—	
PCI Built-in Self-Test (BIST) Control	Index 0F	R	1	—	
PCI Interrupt Line	Index 3C	R	1	—	
PCI Interrupt Pin	Index 3D	R	1	—	
PCI MIN_GNT	Index 3E	R	1	—	
PCI MAX_LAT	Index 3F	R	1	—	
PCI Bus Number	Index 40	R	1	—	
PCI Subordinate Bus Number	Index 41	R	1	—	
PCI Disconnect Counter	Index 42	R/W	1	—	
PCI Special Cycle Address BCR	Index 44-45	R	2	—	
Memory Bank 0 Starting Address	Index 80	R/W	1	Memory	2
Memory Bank 1 Starting Address	Index 81	R/W	1	Memory	2
Memory Bank 2 Starting Address	Index 82	R/W	1	Memory	2
Memory Bank 3 Starting Address	Index 83	R/W	1	Memory	2
Memory Bank 4 Starting Address	Index 84	R/W	1	Memory	2
Memory Bank 5 Starting Address	Index 85	R/W	1	Memory	2
Memory Bank 6 Starting Address	Index 86	R/W	1	Memory	2
Memory Bank 7 Starting Address	Index 87	R/W	1	Memory	2
Memory Bank 0 Ext Starting Address	Index 88	R/W	1	Memory	2
Memory Bank 1 Ext Starting Address	Index 89	R/W	1	Memory	2
Memory Bank 2 Ext Starting Address	Index 8A	R/W	1	Memory	2
Memory Bank 3 Ext Starting Address	Index 8B	R/W	1	Memory	2
Memory Bank 4 Ext Starting Address	Index 8C	R/W	1	Memory	2
Memory Bank 5 Ext Starting Address	Index 8D	R/W	1	Memory	2
Memory Bank 6 Ext Starting Address	Index 8E	R/W	1	Memory	2
Memory Bank 7 Ext Starting Address	Index 8F	R/W	1	Memory	2
Memory Bank 0 Ending Address	Index 90	R/W	1	Memory	2
Memory Bank 1 Ending Address	Index 91	R/W	1	Memory	2
Memory Bank 2 Ending Address	Index 92	R/W	1	Memory	2
Memory Bank 3 Ending Address	Index 93	R/W	1	Memory	2

Table 63. 660 Bridge Indexed BCR Listing(Continued)

Bridge Control Register	Index	R/W	Bytes	Set To[1]	Note
Memory Bank 4 Ending Address	Index 94	R/W	1	Memory	2
Memory Bank 5 Ending Address	Index 95	R/W	1	Memory	2
Memory Bank 6 Ending Address	Index 96	R/W	1	Memory	2
Memory Bank 7 Ending Address	Index 97	R/W	1	Memory	2
Memory Bank 0 Ext Ending Address	Index 98	R/W	1	Memory	2
Memory Bank 1 Ext Ending Address	Index 99	R/W	1	Memory	2
Memory Bank 2 Ext Ending Address	Index 9A	R/W	1	Memory	2
Memory Bank 3 Ext Ending Address	Index 9B	R/W	1	Memory	2
Memory Bank 4 Ext Ending Address	Index 9C	R/W	1	Memory	2
Memory Bank 5 Ext Ending Address	Index 9D	R/W	1	Memory	2
Memory Bank 6 Ext Ending Address	Index 9E	R/W	1	Memory	2
Memory Bank 7 Ext Ending Address	Index 9F	R/W	1	Memory	2
Memory Bank Enable	Index A0	R/W	1	Memory	2
Memory Timing 1	Index A1	R/W	1	—	2
Memory Timing 2	Index A2	R/W	1	—	2
Memory Bank 0 & 1 Addressing Mode	Index A4	R/W	1	Mode 2	2
Memory Bank 2 & 3 Addressing Mode	Index A5	R/W	1	—	
Memory Bank 4 & 5 Addressing Mode	Index A6	R/W	1	—	
Memory Bank 6 & 7 Addressing Mode	Index A7	R/W	1	—	
Cache Status	Index B1	R/W	1	—	
Refresh Cycle Definition	Index B4	R	1	—	
Refresh Timer B5 (Not used—see Indexed BCR D0)	Index B5	R	1	—	
RAS Watchdog Timer	Index B6	R/W	1	—	
PCI Bus Timer (Not used)	Index B7	R	1	—	
Single-Bit Error Counter	Index B8	R/W	1	—	
Single-Bit Error Trigger Level	Index B9	R/W	1	—	
Bridge Options 1	Index BA	R/W	1	—	
Bridge Options 2	Index BB	R/W	1	—	
Error Enable 1	Index C0	R/W	1	—	
Error Status 1	Index C1	R/W	1	—	
Error Simulation 1	Index C2	R/W	1	—	
CPU Bus Error Status	Index C3	R	1	—	
Error Enable 2	Index C4	R/W	1	—	
Error Status 2	Index C5	R/W	1	—	
Error Simulation 2	Index C6	R/W	1	—	
PCI Bus Error Status	Index C7	R/W	1	—	
CPU/PCI Error Address	Index C8-CB	R/W	4	—	
Single-Bit ECC Error Address	Index CC-CF	R/W	4	—	
Refresh Timer Divisor	Index D0-D1	R/W	2	—	
Suspend Refresh Timer	Index D2-D3	R/W	2	—	

Table 63. 660 Bridge Indexed BCR Listing(Continued)

Bridge Control Register	Index	R/W	Bytes	Set To[1]	Note
Bridge Chip Set Options 3	Index D4	R/W	1	—	

Notes:

1 In this column, a long dash — means that the initialization firmware does not write to this register. The register is either not used or not written to, or the value of it depends on changing circumstances. If the word Memory appears, please refer to the System Memory section of the 660 User's Manual.

2 The initialization firmware sets these registers depending on the information reported by the DRAM presence detect registers.

13.3 ISA Bridge (SIO) Initialization

The Reference Design uses an Intel 82378ZB SIO as the ISA bridge. The following information applies to SIO compatible ISA bridges.

The SIO chip should be configured prior to any other PCI bus agent. The SIO PCI arbiter is automatically enabled upon power-on reset. During power-on reset, the SIO drives the A/D(31:0), C/BE#(3:0), and par signals on the PCI bus.

The system I/O EPLD uses the decode circuits in the SIO that produce the signals ECSADDR[2:0] and UBUSCOE# to decode the motherboard register addresses. For this reason, utility bus A and B decode registers must be initialized as shown in Table 64.

The ISA clock divisor must be set as indicated prior to running any CPU to PCI transactions. If the configuration information is stored in Flash, this should pose no problem.

The SIO must be programmed so that interval timer 1 operates in mode 2 with a period of approximately 15 microseconds. This timer controls the ISA refresh interval. It must be programmed at least 200 microseconds before any access to ISA DRAM is attempted.

PCI memory write cycles destined for ISA can use a 32-bit posted write buffer in the SIO. Bit 2 of the PCI control register controls the enabling of the posted write buffer. The default (power-on reset) state for the posted write buffer is disabled. It is required that the posted write buffer be enabled.

Note that PCI burst transactions are not supported by the SIO. For burst transactions, the SIO will always target abort after the first data phase. The system will not allow the CPU to burst to the SIO (or any other PCI agent). No PCI master should be programmed to attempt burst transactions to the SIO.

The SIO defaults (after power-on reset) to the slow sampling point (bits 4:3 of the PCI Control Register) for its subtractive decode. Of the three choices for the sampling point: slow (5 PCI cycles), typical (4 PCI cycles) and fast (3 PCI cycles), one should be chosen that is one clock after the slowest I/O device on the PCI bus. If the PCI agents are all memory mapped above 16M Byte and all I/O mapped above 64K, then the fast sampling point for the subtractive decode can be chosen. This insures that no other PCI agent except the SIO will claim these addresses. Configure PCI agents in this manner to improve performance.

The SIO automatically inserts a 4 ISA clock cycle delay between PCI originated back-to-back 8 and 16 bit I/O cycles to the ISA bus. In addition, the ISA Controller Recovery Timer Register (configuration register, address offset=4Ch) enables a number of additional ISA clock cycles of delay to be inserted between these types of back-to-back I/O cycles. The ISA Controller Recovery Timer Register defaults (after power-on reset) to 2 additional ISA clock cycles of delay, making the total delay equal to 6 ISA clock cycles, for both the 8 and 16 bit I/O recovery times. Since none of the native I/O devices on the reference design require such long recovery times, the additional cycles specified by the ISA Controller Timer Register can be disabled. If an ISA card requiring a long recovery time is supported, the driver should insure that the recovery time is met.

Disable scatter/gather mode and GAT mode.

Do not attempt to access DMA channel 4 address and byte count registers.

Always enable the ISA master and DMA buffers. In order to isolate slow ISA Bus I/O devices from the PCI bus, the DMA controller uses the DMA/ISA master Line Buffer. This buffer can operate in single transaction or in 8-byte mode. Bits 0-LE/7-BE and 1-LE/6-BE of the PCI Control register configure the line buffer for DMA and ISA masters separately. It is required that the 8-byte mode be enabled for both (Bits = 1,1).

The registers in Table 64 must be set in order for the reference design I/O hardware to operate properly. Vendors use LE bit nomenclature, and nomenclature within CPU registers is BE.

Table 64. Summary of SIO Register Setup (Configuration Address = 8080 08xx)

Register	Addr	Bit	Set To	Reset Value	DESCRIPTION
PCI Control Register	40h	2-LE 5-BE	1	0	Enable PCI Memory Posted Write Buffer.
PCI Control Register	40h	1-LE 6-BE	1	0	Enable ISA Master Line buffer.
PCI Control Register	40h	0-LE 7-BE	1	0	Enable DMA Line Buffer.
PCI Arbiter Control (Config/PCI)	41h	0-LE 7-BE	0	0	Disable Guaranteed Access Time (GAT) Mode. Note: GAT does not work in SIO.
ISA Clock Divisor (Config/PCI)	4Dh	5-LE 2-BE	0	0	Disable Coprocessor Error Support.
ISA Clock Divisor (Config/PCI)	4Dh	4-LE 3-BE	1	0	Enable IRQ12/M Mouse Support.
ISA Clock Divisor (Config/PCI)	4Dh	3-LE 4-BE	*	0	* This bit should be set to 1 before changing or loading the PCI ISA Clock Divisor value. Setting this bit to 1 asserts the RSTDVR signal, which resets the System I/O EPLD and any devices on the ISA bus slots. All of these devices require reconfiguration after this bit has been asserted. Software must guarantee that RSTDVR be asserted for a minimum of 1 ms after the clock divisor value is set.
ISA Clock Divisor (Config/PCI)	4Dh	2:0-LE 5:7-BE	*	0	* Set this field to 000b (divisor = 4). If PCI clock is slower than 33 MHz, then this field would be 001b (divisor=3).
Utility Bus Chip Select A (Config/PCI)	4Eh	4-LE 4-BE	1	0	Disable generation of ECSADDR(2:0) and UBUSOE# for the IDE and Floppy decode.
Utility Bus Chip Select A (Config/PCI)	4Eh	1-LE 6-BE	1	1	Enable keyboard addresses (60h, 62h, 64h, 66h).
Utility Bus Chip Select A (Config/PCI)	4Eh	0-LE 7-BE	1	1	Enable TOD Addresses (70h, 71h).
Utility Bus Chip Select B (Config/PCI)	4Fh	7-LE 0-BE	1	0	Enable access to the motherboard registers in the 0800-08FF address range.
Utility Bus Chip Select B (Config/PCI)	4Fh	6-LE 1-BE	1	1	Enable Port 92h access.
Utility Bus Chip Select B (Config/PCI)	4Fh	5:4-LE 2:3-BE	11	00	Disable generation of default address for Parallel Port.

Table 64. Summary of SIO Register Setup (Configuration Address = 8080 08xx) (Continued)

Register	Addr	Bit	Set To	Reset Value	DESCRIPTION
Utility Bus Chip Select B (Config/PCI)	4Fh	3:2-LE 4:5-BE	11	00	Disable generation of default address for Serial Port B.
Utility Bus Chip Select B (Config/PCI)	4Fh	1:0-LE 6:7-BE	11	00	Disable generation of default address for Serial Port A.
Interrupt Controller 1 - ICW1 (I/O /PCI)	20h	3-LE 4-BE	0	x	Set Interrupt Controller 1 to edge triggered mode.
Interrupt Controller 1 - ICW1 (I/O /PCI)	20h	1-LE 6-BE	0	x	Set Interrupt Controller 1 to cascade mode.
Interrupt Controller 2 - ICW1 (I/O /PCI)	A0h	3-LE 4-BE	0	x	Set Interrupt Controller 2 to edge triggered mode.
Interrupt Controller 2 - ICW1 (I/O /PCI)	A0h	1-LE 6-BE	0	x	Set Interrupt Controller 2 to cascade mode.
NMI Status and Control (I/O /PCI)	61h	3-LE 4-BE	0	0	IOCHK# NMI enabled.
NMI Status and Control (I/O /PCI)	61h	2-LE 5-BE	0	0	PCI SERR# NMI enabled.
NMI Enable and TOD Address (I/O /PCI)	70h	7-LE 0-BE	0	1	NMI interrupt enabled.
DMA Command (I/O /PCI)	08h, D0h	7-LE 0-BE	0	0	DACK# Assert Level set to low.
DMA Command (I/O /PCI)	08h, D0h	6-LE 1-BE	0	0	DREQ Sense Level set to high.

13.3.1 ISA Bridge PCI Configuration Registers

Table 65. Summary of SIO PCI Configuration Registers

Address	Description	Type	Reset Value	Set To *
8080 0800	Vendor Identification	R	86h	
8080 0801	Vendor Identification	R	80	
8080 0802	Device Identification	R	84	
8080 0803	Device Identification	R	04	
8080 0804	Command	R/W	07	0F
8080 0805	Command	R/W	00	00
8080 0806	Device Status	R/W	00	
8080 0807	Device Status	R/W	02	
8080 0808	Revision Identification	R/W	00	
8080 0840	PCI Control	R/W	20	21
8080 0841	PCI Arbiter Control	R/W	00	00
8080 0842	PCI Arbiter Priority Control	R/W	04	04
8080 0843	PCI Arbiter Priority Control Extension	R/W	00	00
8080 0844	MEMCS# Control	R/W	00	00
8080 0845	MEMCS# Bottom of Hole	R/W	10	10
8080 0846	MEMCS# Top of Hole	R/W	0F	0F
8080 0847	MEMCS# Top of Memory	R/W	00	00
8080 0848	ISA Address Decoder Control	R/W	01	F1
8080 0849	ISA Address Decoder ROM Block	R/W	00	00
8080 084A	ISA Address Bottom of Hole	R/W	10	10
8080 084B	ISA Address Top of Hole	R/W	0F	0F
8080 084C	ISA Controller Recovery Timer	R/W	56	56
8080 084D	ISA Clock Divisor	R/W	40	10
8080 084E	Utility Bus Chip Select A	R/W	07	07
8080 084F	Utility Bus Chip Select B	R/W	4F	FF
8080 0854	MEMCS# Attribute Register #1	R/W	00	
8080 0855	MEMCS# Attribute Register #2	R/W	00	
8080 0856	MEMCS# Attribute Register #3	R/W	00	
8080 0857	Scatter/Gather Relocation Base	R/W	04	
8080 0860	PIRQ Route Control 0	R/W	80	0F
8080 0861	PIRQ Route Control 1	R/W	80	0F
8080 0862	PIRQ Route Control 2	R/W	80	80
8080 0863	PIRQ Route Control 3 (unused)	R/W	80	80
8080 0880	BIOS Timer Base Address	R/W	78	
8080 0881	BIOS Timer Base Address	R/W	00	

Note: * If the entry in this column is blank, then the boot firmware does not write to this register.

13.4 MPIC Initialization

The register set in MPIC can be divided into two groups based on access method:

- PCI configuration registers Access via PCI configuration transactions
- PCI I/O space registers Access via PCI I/O transactions

All information in this Section is presented in little endian nomenclature. See the MPIC data sheet for more information.

13.4.1 MPIC PCI Configuration Registers

Table 66 shows the PCI configuration registers in MPIC. See the *PCI Local Bus Specification* and the MPIC data sheet for programming details. The reference design uses PCI_AD[15] (device number 5) to activate the Ethernet controller IDSEL line during PCI configuration transactions.

Table 66. MPIC PCI Configuration Registers

31 24	23 16	15 8	7 0	Offset
Device ID (0046h)		Vendor ID (1014h)		00h
Status (0200h)		Command (0000h)		04h
Reserved	Reserved	Reserved	Reserved	08h
Reserved	Reserved	Reserved	Reserved	0Ch
Base Address Zero (I/O Space) ()				10h
Base Address One (Memory Space) ()				14h
Reserved				18h
				...
Reserved				FCh

13.4.2 MPIC PCI I/O Registers

MPIC decodes a total of 256K of I/O space (offsets x 0000 0000 – x 0003 FFFF).

13.4.2.1 MPIC Global Registers (PCI I/O)

Table 67. MPIC Global Registers (PCI I/O)

PCI I/O Offset	Name	Description	R/W	Set To	Notes
0000 1000		Feature Reporting Register 0			
0000 1020		Global Configuration Register 0		Bit 29, set=1	
0000 1080		Vendor ID Register			
0000 1090		Processor Init Register			
0000 10A0		IPI 0 Vector/Priority Register		0x800F0020	
0000 10B0		IPI 1 Vector/Priority Register		0x800F0021	
0000 10C0		IP12 Vector/Parity Register (Unused)		0x800F0022	
0000 10D0		IP13 Vector/Parity Register (Unused)		0x800F0023	
0000 10F0		Timer Frequency Reporting Register		—	
0000 1100		Global Timer 0 Current Count		—	
0000 1110		Global Timer 0 Base Count		—	
0000 1120		Global Timer 0 Vector/Priority Register		0x80000030	

Table 67. MPIC Global Registers (PCI I/O) (Continued)

0000 1130		Global Timer 0 Destination Register		—	
0000 1140		Global Timer 1 Current Count		—	
0000 1150		Global Timer 1 Base Count		—	
0000 1160		Global Timer 1 Vector/Priority Register		0x80000031	
0000 1170		Global Timer 1 Destination Register		—	
0000 1180		Global Timer 2 Current Count		—	
0000 1190		Global Timer 2 Base Count		—	
0000 11A0		Global Timer 2 Vector/Priority Register		0x80000032	
0000 11B0		Global Timer 2 Destination Register		—	
0000 11C0		Global Timer 3 Current Count		—	
0000 11D0		Global Timer 3 Base Count		—	
0000 11E0		Global Timer 3 Vector/Priority Register		0x80000033	
0000 11F0		Global Timer 3 Destination Register		—	

13.4.2.2 MPIC Interrupt Source Configuration Registers (PCI I/O)

Table 68. MPIC Interrupt Source Configuration Registers (PCI I/O)

Address Offset	Name	Description	R/W	Set To	Notes
0001 0000		Interrupt Source 0 Vector/Priority		0x00CE0000	
0001 0010		Interrupt Source 0 Destination		—	
0001 0020		Interrupt Source 1 Vector/Priority		0x804E0011	
0001 0030		Interrupt Source 1 Destination		—	
0001 0040		Interrupt Source 2 Vector/Priority		0x804E0012	
0001 0050		Interrupt Source 2 Destination		—	
0001 0060		Interrupt Source 3 Vector/Priority		0x804E0013	
0001 0070		Interrupt Source 3 Destination		—	
0001 0080		Interrupt Source 4 Vector/Priority		0x804E0014	
0001 0090		Interrupt Source 4 Destination		—	
0001 00A0		Interrupt Source 5 Vector/Priority		0x804E0015	
0001 00B0		Interrupt Source 5 Destination		—	
0001 00C0		Interrupt Source 6 Vector/Priority		0x804E0016	
0001 00D0		Interrupt Source 6 Destination		—	
0001 00E0		Interrupt Source 7 Vector/Priority		0x804E0017	
0001 00F0		Interrupt Source 7 Destination		—	
0001 0100		Interrupt Source 8 Vector/Priority		0x804E0018	
0001 0110		Interrupt Source 8 Destination		—	
0001 0120		Interrupt Source 9 Vector/Priority		0x804E0019	
0001 0130		Interrupt Source 9 Destination		—	
0001 0140		Interrupt Source 10 Vector/Priority		0x804E001A	
0001 0150		Interrupt Source 10 Destination		—	
0001 0160		Interrupt Source 11 Vector/Priority		0x804E001B	
0001 0170		Interrupt Source 11 Destination		—	
0001 0180		Interrupt Source 12 Vector/Priority		0x804E001C	
0001 0190		Interrupt Source 12 Destination		—	
0001 01A0		Interrupt Source 13 Vector/Priority		0x804E001D	
0001 01B0		Interrupt Source 13 Destination		—	
0001 01C0		Interrupt Source 14 Vector/Priority		0x804E001E	
0001 01D0		Interrupt Source 14 Destination		—	
0001 01E0		Interrupt Source 15 Vector/Priority		0x804E001F	
0001 01F0		Interrupt Source 15 Destination		—	

13.4.2.3 MPIC Per Processor Registers (PCI I/O)

Table 69. MPIC Per Processor Registers (PCI I/O)

Address Offset	Name	Description	R/W	Set To	Notes
0002 0040		Processor 0 IPI 0 Dispatch		—	
0002 0050		Processor 0 IPI 1 Dispatch		—	
0002 0080		Processor 0 Current Task Priority		0	
0002 00A0		Processor 0 Interrupt Acknowledge		—	
0002 00B0		Processor 0 End of Interrupt		—	
0002 1040		Processor 1 IPI 0 Dispatch		—	
0002 1050		Processor 1 IPI 1 Dispatch		—	
0002 1080		Processor 1 Current Task Priority		0	
0002 10A0		Processor 1 Interrupt Acknowledge		—	
0002 10B0		Processor 1 End of Interrupt		—	

13.5 Ethernet Initialization

The register set in the Ethernet controller can be divided into three groups based on access method:

• PCI configuration registers	Access via PCI configuration transactions
• PCI I/O space registers	Access via PCI I/O transactions
• CSR and BCR registers	Access using pairs of PCI I/O transactions

All information in this Section is presented in little endian nomenclature. See the 79C970A data sheet for more information.

13.5.1 Ethernet PCI Configuration Registers

Table 70 shows the PCI configuration registers in the Ethernet controller. See the *PCI Local Bus Specification* and the 79C970A data sheet for programming details. The reference design uses PCI_AD[16] (device number 6) to activate the Ethernet controller IDSEL line during PCI configuration transactions.

Table 70. Ethernet PCI Configuration Registers

31 24	23 16	15 8	7 0	Offset
Device ID (2000h)		Vendor ID (1022h)		00h
Status		Command		04h
Base-Class (02h)	Sub-Class (00h)	Programming IF (00h)	Revision ID (00h)	08h
Reserved	Header Type (00h)	Latency Timer (00h)	Reserved	0Ch
Base Address Zero (I/O Space) (8080 4000h)				10h
Base Address One (Memory Space)				14h
Reserved				18h
Reserved				1Ch
Reserved				20h
Reserved				24h
Reserved				28h
Expansion ROM Base Address				2Ch
Reserved				30h
Reserved				34h
Reserved				38h
MAX_LAT	MIN_GNT	Interrupt Pin	Interrupt Line	3Ch
Reserved				40h
Reserved				
Reserved				FCh

13.5.2 Ethernet PCI I/O Registers

The reference design implementation of the Ethernet controller uses 32 bytes of PCI I/O space for access to various internal registers. This address space can be mapped to either PCI I/O or PCI memory space by the PCI configuration registers, and would typically be configured soon after system power up.

There are two modes of accessing these registers. For backwards compatibility with earlier AMD Ethernet controllers, the default mode after power up is Word I/O (WIO), in which the PCI I/O registers are mapped as 2-byte entities (see Table 71). The 79C970A can also be configured for Double Word I/O (DWIO), in which the PCI I/O registers are mapped as 4-byte entities (see Table 72). See the 79C970A data sheet for more information.

To change to DWIO, write a 4-byte value to offset 10h. Once the part is in DWIO mode, a hard reset is needed to exit. If the EEPROM programs to DWIO mode, then WIO mode can only be entered by reprogramming the EEPROM and doing a hard reset.

When in WIO mode, APROM addresses may be accessed as bytes on either even or odd addresses. Attempting 4-byte writes to the PCI I/O registers (except for offset 10h) while in WIO mode may cause unpredictable programming of the part. Four byte reads from the PCI I/O registers while in WIO mode are illegal and will produce undefined results. Two byte accesses to non-word boundaries are illegal and may produce unexpected results. Only the APROM locations may be accessed as a non-word.

When in DWIO mode all accesses must be 4-byte and 4-byte aligned. This includes the APROM locations. RDP, RAP, and BDP contain only two bytes of valid data; the other two bytes are reserved for future use (CSR88 is an exception to this rule). The reserved bytes should be written as zero and ignored when read.

The Vendor Specified Word is not implemented. This address is reserved for customer use and will not be used by future AMD ethernet controllers.

Some of this is setup information which may be stored in an external serial EEPROM.

Table 71. Ethernet I/O Map In Word I/O Mode (DWIO = 0)

Offset	Number of Bytes	Register
00h-0Fh	16	APROM
10h	2	RDP
12h	2	RAP (shared by RDP and BDP)
14h	2	Reset Register
16h	2	BDP
18h-1Ph	8	Reserved

Table 72. Ethernet I/O Map In DWord I/O Mode (DWIO = 1)

Offset	Number of Bytes	Register
00h-0Fh	16	APROM
10h	4	RDP
14h	4	RAP (shared by RDP and BDP)
18h	4	Reset Register
1Ch	4	BDP

13.5.3 Ethernet CSR and BSR Registers

The Ethernet controller uses a third set of registers that are reached by pairs of PCI I/O register accesses (indexed addressing). The Control and Status Registers (CSR) are accessed by first writing the offset address into the RAP register, and then writing to or reading from the RDP register to effect the data transfer. The CSR registers (see Table 73) are used to configure the Ethernet MAC engine and to obtain status information.

The Bus Control Registers (BCR) are accessed by first writing the offset address into the RAP register, and then writing to or reading from the BDP register to effect the data transfer. The BCR registers (see Table 74) are used to configure the bus interface unit and the LEDs.

Most of the BCR and CSR registers are only programmed once, during the setup of the Ethernet controller, but there are no restrictions on the number of times that they may be accessed. If the default power up values are acceptable these registers need not be accessed at all.

Table 73. Ethernet CSR Registers

Register	Description
CSR1	Initialization Block Address[15:0]
CSR2	Initialization Block Address[31:16]
CSR3	Interrupt Masks and Deferral Control
CSR4	Test and Features Control
CSR5	Extended Control and Interrupt
CSR8	Logical Address Filter[15:0]
CSR9	Logical Address Filter[31:16]
CSR10	Logical Address Filter[47:32]
CSR11	Logical Address Filter[63:48]
CSR12	Physical Address[15:0]
CSR13	Physical Address[31:16]
CSR14	Physical Address[47:32]
CSR15	Mode
CSR24	Base Address of Receive Descriptor Ring Lower
CSR25	Base Address of Receive Descriptor Ring Upper
CSR30	Base Address of Transmit Descriptor Ring Lower
CSR31	Base Address of Transmit Descriptor Ring Upper
CSR47	Polling Interval
CSR76	Receive Descriptor Ring Length
CSR78	Transmit Descriptor Ring Length
CSR82	Bus Activity Timer
CSR100	Memory Error Timeout
CSR122	Receiver Packet Alignment Control

Table 74. Ethernet BCR Registers

Register	Description
BCR2	Miscellaneous Configuration
BCR4	Link Status LED
BCR5	LED1 Status
BCR6	LED2 Status
BCR7	LED3 Status
BCR9	Full–Duplex Control
BCR18	Bus and Burst Control
BCR20	Software Style

13.5.4 Ethernet EEPROM Interface

The Ethernet controller contains a built-in Microwire EEPROM interface which allows the device to be programmed from the EEPROM at power up. Some of the BCR and other configuration data is stored in the EEPROM.

While the Ethernet controller supports reading and writing to an external serial EEPROM, the reference design only implements reading the serial EEPROM (the EEPROM write enable control is not implemented). The EEPROM is socketed to allow for external reprogramming.

The serial EEPROM used in the reference design is a 64 x 16 bit device which resides in an 8 pin dip socket. Table 75 shows the EEPROM contents and their suggested programming values. Different implementations may require a change to some or all of the suggested values. The automatic EEPROM read operation will access 36 bytes.

Table 75. Ethernet EEPROM Content

Byte Address	Description	Initial Value
00h	First byte of the ISO 8802–3 (IEEE/ANSI 802.3) station physical address for this node, where first byte refers to first byte to appear on the 802.3 medium	08
01h	Second byte of the ISO 8802–3 station physical address for this node	00
02h	Third byte of the node address	5A
03h	Fourth byte of the node address	FC
04h	Fifth byte of the node address	02
05h	Sixth byte of the node address	0F
06h	Reserved location: must be 00h	00
07h	Reserved location: must be 00h	00
08h	Reserved location: must be 00h	00
09h	Hardware ID: must be 11h if compatibility to AMD drivers is desired	10
0Ah	User programmable space	00
0Bh	User programmable space	00
0Ch	LSByte of 2–byte checksum, which is the sum of bytes 00h–0Bh and bytes 0Eh and 0Fh	2E
0Dh	MSByte of 2–byte checksum, which is the sum of bytes 00h–0Bh and bytes 0Eh and 0Fh	02

Table 75. Ethernet EEPROM Content (Continued)

Byte Address	Description	Initial Value
0Eh	Must be ASCII W (57h) if compatibility to AMD driver software is desired	57
0Fh	Must be ASCII W (57h) if compatibility to AMD driver software is desired	57
10h	BCR4[7:0] (Link Status LED)	00
11h	BCR4[15:8] (Link Status LED)	00
12h	BCR5[7:0] (LED1 Status)	00
13h	BCR5[15:8] (LED1 Status)	00
14h	BCR18[7:0] (Burst and Bus Control)	81
15h	BCR18[15:8] (Burst and Bus Control)	21
16h	BCR2[7:0] (Miscellaneous Configuration)	06
17h	BCR2[15:8] (Miscellaneous Configuration)	00
18h	BCR6[7:0] (LED2 Status)	00
19h	BCR6[15:8] (LED2 Status)	00
1Ah	BCR7[7:0] (LED3 Status)	00
1Bh	BCR7[15:8] (LED3 Status)	00
1Ch	BCR9[7:0] (Full–Duplex Control)	00
1Dh	BCR9[15:8] (Full–Duplex Control)	00
1Eh	Reserved location must be 00h	00
1Fh	Checksum adjust byte for the first 36 bytes of the EEPROM contents, checksum of the first 36 bytes of the EEPROM should total to FFh	F4
20h	BCR22[7:0] (PCI Latency)	06
21h	BCR22[15:8] (PCI Latency)	FF
22h	Reserved location must be 00h	00
23h	Reserved location must be 00h	00

13.6 SCSI Initialization

The register set in the SCSI controller can be divided into two groups based on access method:

• PCI configuration registers	Access via PCI configuration transactions
• PCI I/O space registers	Access via PCI I/O transactions

All information in this section is presented in little endian nomenclature. See the SCSI controller data sheet for more information.

13.6.1 SCSI PCI Configuration Registers

Table 76 shows the PCI configuration registers in the SCSI controller. See the *PCI Local Bus Specification* and the SCSI controller data sheet for programming details.

This reference design uses PCI_AD[12] (device number 2) to activate the SCSI controller IDSEL during PCI configuration transactions.

Table 76. SCSI PCI Configuration Registers

31 24	23 16	15 8	7 0	Offset
Device ID (0001h)		Vendor ID (1000h)		00h
Status		Command		04h
Base-Class (00h)	Sub-Class (00h)	Programming IF (00h)	Revision ID (01h)	08h
Not Supported	Header Type (00h)	Latency Timer	Not Supported	0Ch
Base Address Zero (I/O Space)				10h
Base Address One (Memory Space)				14h
Reserved				18h
Reserved				1Ch
Reserved				20h
Reserved				24h
Reserved				28h
Expansion ROM Base Address				2Ch
Reserved				30h
Reserved				34h
Reserved				38h
MAX_LAT	MIN_GNT	Interrupt Pin	Interrupt Line	3Ch
Reserved				40h
Reserved				
Reserved				FCh

13.6.2 SCSI PCI I/O Registers

Table 77 shows registers in the SCSI controller that can be mapped to either PCI I/O space or PCI memory space by the PCI configuration registers. If the PCI I/O Base Address Register is used, then the offset addresses shown in Table 77 are offsets from that PCI I/O space base address. If the PCI Memory Base Address Register is used, then the offset addresses shown in Table 77 are offsets from that PCI I/O space base address.

Table 77. SCSI PCI I/O Registers

Offset	Name	R/W	Set To	Note
00	SCSI Control 0		0xC0	
01	SCSI Control 1		0x00	
02	SCSI Control 2		0x00	
03	SCSI Control 3		0x33	
04	SCSI Chip ID		0x07	
05	SCSI Transfer		—	
06	SCSI Destination OD		—	
07	General Purpose Bits		0x0	
08	SCSI First Byte Received		—	
09	SCSI Output Control Latch		—	
0A	SCSI Selector ID		—	
0B	SCSI Bus Control Lines		—	
0C	DMA Status		—	
0D	SCSI Status 0		—	
0E	SCSI Status 1		—	
0F	SCSI Status 2		—	
10-13	Data Structure Address		—	
14	Interrupt Status		—	1
18	Chip Test 0		0x0	
19	Chip Test 1		—	
1A	Chip Test 2		—	
1B	Chip Test 3		—	
1C-1F	Temporary Stack		—	
20	DMA FIFO		—	
21	Chip Test 4		—	
22	Chip Test 5		—	
23	Chip Test 6		—	
24-26	DMA Byte Counter		—	
27	DMA Command		—	
28-2B	DMA Next Address for Data		—	
2C-2F	DMA SCRIPTS Pointer		—	
30-33	DMA SCRIPTS Pointer Save		—	
34-37	General Purpose Scratch Pad A		—	
38	DMA Mode		0xC0	
39	DMA Interrupt Enable		0x7D	
3A	DMA Watchdog Timer		—	
3B	DMA Control		0x09	
3C-3F	Sum Output of Internal Adder		—	
40	SCSI Interrupt Enable 0		0x8E	
41	SCSI Interrupt Enable 1		0x04	
42	SCSI Interrupt Status 0		—	
43	SCSI Interrupt Status 1		—	
44	SCSI Longitudinal Parity		—	

Table 77. SCSI PCI I/O Registers (Continued)

Offset	Name	R/W	Set To	Note
45	Reserved		—	
46	Memory Access Control		—	
47	General Purpose Control		—	
48	SCSI Timer 0		0xB0	
49	SCSI Timer 1		—	
4A	Response ID		—	
4B	Reserved		—	
4C	SCSI Test 0		—	
4D	SCSI Test 1		—	
4E	SCSI Test 2		—	
4F	SCSI Test 3		0x80	
50	SCSI Input Data Latch		—	
51-53	Reserved		—	
54	SCSI Output Data Latch		—	
55-57	Reserved		—	
58	SCSI Bus Data Lines		—	
59-5B	Reserved		—	
5C-5F	General Purpose Scratch Pad B		—	
60-7F	Reserved		—	
80-FF	(Repeat 00-7F)		—	

Notes:

1The only accessible register when SCRIPTS is executing is the interrupt status register.

13.7 SuperI/O Initialization

SuperI/O initialization and register access can be divided into three distinct phases:

- Pin strap configuration
- Configuration register configuration
- Controller register accesses

All of the SuperI/O registers are accessed by ISA I/O cycles. Refer to the schematics and to the SuperI/O data sheet for more information.

13.7.1 SuperI/O Pin Strap Configuration

The VLD[1:0], CFG[4:0], and BADDR[1:0] signal groups allow various features of the SuperI/O to be strap-programmed during reset.

- VLD[1:0] affect the state of the FDC Tape Drive register (3F3h).
- CFG[4:0] affect the configuration of the Super I/O.
- BADDR[1:0] sets the base addresses of the Index Register and the Data Register.

13.7.2 SuperI/O Configuration Register Configuration

The configuration registers control the basic operation of the SuperI/O. Three of these registers (FER, FAR, and PTR) are set to default values and values specified by the CFG[4:0] strap inputs during reset. These and the other configuration registers can also be loaded with software-supplied values, using the Index register and the Data register, which are located at the ISA bus I/O addresses specified by the BADDR[1:0] (see Table 78) strap inputs during reset.

Table 78. SuperI/O BADDR Encoding

BADDR1	BADDR0	Index Register Address	Data Register Address
0	0	398h	399h
0	1	26Eh	26Fh
1	0	15Ch	15Dh
1	1	2Eh	2Fh

The configuration registers are indirectly (two accesses required) accessed from the ISA bus. To access a given configuration register, first write the index (see Table 79) of the register into the Index register. To read (or write) the register pointed to by the Index register, read (or write) the Data register.

Table 79. SuperI/O Configuration Registers (ISA Bus Indirect Access)

Index	Name	Description
00	FER	Function Enable Register
01	FAR	Function Address Register
02	PTR	Power & Test Register
03	FCR	Function Control Register
04	PCR	Printer Control Register
06	PMC	Power Management Control Register
07	TUP	Tape, UARTs, and Parallel Port Configuration Register
08	SID	SuperI/O Identification Register

13.7.3 SuperI/O Controller Register Access

There are operating registers associated with each of the controllers in the SuperI/O. These registers are located at ISA bus I/O addresses specified by the Function Address Register. For more information, see the SuperI/O data sheet.

13.7.3.1 SuperI/O FDC Registers

Table 80. SuperI/O FDC Registers (ISA I/O)

Index	Name	R/W	Description
0	SRA	R	Status Register A
1	SRB	R	Status Register B
2	DOR	R/W	Digital Output Register
3	TDR	R/W	Tape Drive Register
4	MSR	R	Main Status Register
4	DSR	W	Data Rate Select Register
5	FIFO	R/W	Data Register (FIFO)
6		X	None (Bus Tristate)
7	DIR	R	Digital Input Register
7	CCR	W	Configuration Control Register

13.7.3.2 SuperI/O UART Registers

Table 81. SuperI/O UART Registers (One Set Per UART) (ISA I/O)

DLAB	Index	Name	R/W	Description
0	0		R	Receiver Buffer
0	0		W	Transmitter Holding
0	1	IER	R/W	Interrupt Enable Register
0	2	IIR	R	Interrupt Identification Register
0	2	FCR	W	FIFO Control Register
x	3	LCR	R/W	Line Control Register
x	4	MCR	R/W	Modem Control Register
x	5	LSR	R/W	Line Status Register
x	6	MSR	R	Modem Status Register
x	7	SCR	R/W	Scratchpad Register
1	0			Divisor Latch (Least Significant Byte)
1	1			Divisor Latch (Most Significant Byte)

13.7.3.3 SuperI/O Parallel Port Registers

Table 82. SuperI/O Parallel Port Registers (ISA I/O)

Index	Name	R/W	Description
0		R/W	Data
1		R	Status
2		R/W	Control
3			None (Bus Tristate)

13.7.3.4 SuperI/O IDE Registers

Table 83. SuperI/O IDE Registers (ISA I/O)

Address	Read Function	Write Function
1F0	Data	Data
1F1	Error	Features
1F2	Sector Count	Sector Count
1F3	Sector Number	Sector Number
1F4	Cylinder Low	Cylinder Low
1F5	Cylinder High	Cylinder High
1F6	Drive/Head	Drive/Head
1F7	Status	Command
3F6	Alternate Status	Device Control
3F7	Drive Address (except D7)	Not Used (3S)

13.8 Business Audio Initialization

The CS4232 supports the Plug and Play interface.

Refer to the CS4232 manual for chip setup information. The motherboard imposes a set-up requirement that the XCTL0 bit of Pin Control register I10 be set to 1 following power-on in order to enable the internal speaker. XCTL0 defaults to 0b in order to mute the speaker during power-on. The mono microphone input should be muted prior to demuting the speaker. The mono microphone input should be used only when the mono output of the chip is internally muted; otherwise, undesirable feedbacks may occur. XCTL0 mutes both the timer 2 output and the mono output of the chip when it is zero. It should, therefore, be set to 1 for normal operation.

13.8.1 CS4232 Initialization

To use the CS4232, it must first be initialized. The CS4232 ignores all bus activity, except for the initialization, until it is done.

The initialization key should be written to ISA IO address 0x279. This is system address 0x80000279. The key consist of the following 32 bytes:

0x96, 0x35, 0x9A, 0xCD, 0xE6, 0xF3, 0x79, 0xBC,
0x5E, 0xAF, 0x57, 0x2B, 0x15, 0x8A, 0xC5, 0xE2,
0xF1, 0xF8, 0x7C, 0x3E, 0x9F, 0x4F, 0x27, 0x13,
0x09, 0x84, 0x42, 0xA1, 0xD0, 0x68, 0x34, 0x1A .

Before writing the key, 0 should be written to port 0x279 twice.

After writing each byte to port 0x279, wait for 250 micro-seconds before writing the next. After the key is sent, the chip enters its Config State. After the chip is configured, it must be placed back into the Wait-for-Key state to be used. This is done by sending the WAIT_FOR_KEY command (0x79) to port 0x279.

13.8.1.1 Config State

In the config state, each logical device in the chip may be configured. Base address, DMA channels, and interrupt levels for each logical device may be configured independently. Also, each logical device may be activated or deactivated.

Table 84 and Table 85 show how the numbers used to assign channels map to the system channels. These are the only settings which may be used for assigning DMA and interrupt channels. Notice that the only item that does not have a one-to-one correspondence is interrupt level 14.

Table 84. Settings for DMA Channels

To Use DMA Channel	Program CS4232 Setting For:
0	0
1	1
3	3

Table 85. Settings for Interrupt Channels

To Use Interrupt Level	Program CS4232 Setting For (Interrupt Descriptor):
5	5
7	7
9	9
11	11
12	12
14	15

To configure the part, a series of commands are sent to port 0x279. Each command is a byte followed by one or more data bytes. The number of data bytes depends upon the command, but it is fixed for a particular command. To configure a logical device, first select it with the DEVICE command. Then use the remaining commands to configure the device. Table 86 lists the valid commands.

Table 86. Device Configuration Commands

Command	Description	Code
DEVICE	Select logical device to configure	0x15
ADDRESS0	Select base address number 0	0x47
ADDRESS1	Select base address number 1	0x48
ADDRESS2	Select base address number 2	0x42
INT0	Select interrupt 0 level	0x22
INT1	Select interrupt 1 level	0x27
DMA0	Select dma 0 channel	0x2A
DMA1	Select dma 1 channel	0x25
ACTIVATE	Activate logical device	0x33
WAIT_FOR_KEY	Wait for key	0x79

13.8.1.2 Device

The device command is followed by one byte. This byte selects a logical device for configuration. After selecting a logical device the other commands may be used to configure it. The logical devices for the CS4232 and their identifiers are found in Table 87.

Table 87. CS4232 Logical Devices

Device	Logical Device Number
Windows Sound System/ OPL3/Sound Blaster Pro	0
Game Port	1
CS4232 Control	2
MPU-401	3
CD-ROM	4

The resources required by each logical device vary. Table 88 lists the maximum of each type of resource is associated with each logical device.

Table 88. Logical Device Resources

Logical Device	Base Addresses	Interrupt Levels	DMA Channels
0	3	1	2
1	1	0	0
2	1	1	0
3	1	1	0
4	1	1	1

13.8.1.3 ADDRESS0, ADDRESS1, ADDRESS2

These commands are followed by two bytes. The first byte is the high byte of the address, the second is the low byte of the address. Only logical device 0 uses other than ADDRESS0. The descriptions for logical device 0 are found in Table 89.

Table 89. Base Address Commands

Base Address Command	Logical Device 0 Description
ADDRESS0	Windows Sound System
ADDRESS1	OPL3
ADDRESS2	Sound Blaster Pro

The Windows Sound System is the four registers that were available in the CS4231.

13.8.1.4 INT0, INT1

These commands are followed by one byte that gives the interrupt descriptor for the item being configured. The command used determines the descriptor level of the interrupt being configured. INT0 programs dtermine descriptor level 0. INT1 programs determine descriptor level 1. There are no documented uses of INT1. Each logical device has a maximum of 1 interrupt level.

13.8.1.5 DMA0, DMA1

These commands are followed by one byte. This byte gives the DMA channel. Only logical device 0 uses more than one channel. DMA commands are shown in Table 90.

Table 90. DMA Channels

DMA command	DMA description for logical device 0
DMA0	Playback
DMA1	Capture

13.8.1.6 ACTIVATE

The activate command is followed by one byte which tells whether to turn on or off the current logical device. On/off commands are shown in Table 91.

Table 91. Activate Commands

Action	Value
Turn on	1
Turn off	0

13.8.1.7 WAIT_FOR_KEY

This command signals the end of CS4232 configuration. The chips will look for its key at address 0x279, allowing it to be reconfigured.

13.9 Reference Design Combined Register Listing

13.9.1 ISA Registers

13.9.1.1 Direct Access ISA Registers

Table 92 contains a summary listing of the registers that are physically located in the reference design motherboard. In general, these registers are accessed using single CPU transfers. There is an additional set of registers (see Table 92) located in the 660 bridge, which are accessed using pairs of CPU transfers.

Table 92. Combined Direct Access ISA Bus I/O Register Listing

ISA Port	Contiguous Mode Addr[1]	Non-Contig Mode Addr[1]	Description	R/W	Set To[2]	Loc[3]	Note
0000	8000 0000	8000 0000	DMA1 CH0 Base and Current Addr	R/W	—	SIO	
0001	8000 0001	8000 0001	DMA1 CH0 Base and Current Cnt	R/W	—	SIO	
0002	8000 0002	8000 0002	DMA1 CH1 Base and Current Addr	R/W	—	SIO	
0003	8000 0003	8000 0003	DMA1 CH0 Base and Current Cnt	R/W	—	SIO	
0004	8000 0004	8000 0004	DMA1 CH2 Base and Current Addr	R/W	—	SIO	
0005	8000 0005	8000 0005	DMA1 CH2 Base and Current Cnt	R/W	—	SIO	
0006	8000 0006	8000 0006	DMA1 CH3 Base and Current Addr	R/W	—	SIO	
0007	8000 0007	8000 0007	DMA1 CH3 Base and Current Cnt	R/W	—	SIO	
0008	8000 0008	8000 0008	DMA1 Status(R) Command(W)	R/W	—	SIO	
0009	8000 0009	8000 0009	DMA1 Soft Request	W	—	SIO	
000A	8000 000A	8000 000A	DMA1 Write Single Mask Bit	W	—	SIO	
000B	8000 000B	8000 000B	DMA1 Write Mode	W	—	SIO	
000C	8000 000C	8000 000C	DMA1 Clear Byte Pointer	W	—	SIO	
000D	8000 000D	8000 000D	DMA1 Master Clear	W	—	SIO	
000E	8000 000E	8000 000E	DMA1 Clear Mask	W	—	SIO	
000F	8000 000F	8000 000F	DMA1 R/W All Mask Register Bits	R/W	—	SIO	
0020	8000 0020	8000 1000	INT1 Control	R/W	—	SIO	
0021	8000 0021	8000 1001	INT1 Mask	R/W	—	SIO	
0040	8000 0040	8000 2000	Timer Counter 1 - Counter 0 Cnt	R/W	—	SIO	
0041	8000 0041	8000 2001	Timer Counter 1 - Counter 1 Cnt	R/W	—	SIO	
0042	8000 0042	8000 2002	Timer Counter 1 - Counter 2 Cnt	R/W	—	SIO	

Table 92. Combined Direct Access ISA Bus I/O Register Listing (Continued)

ISA Port	Contiguous Mode Addr[1]	Non-Contig Mode Addr[1]	Description	R/W	Set To[2]	Loc[3]	Note
0043	8000 0043	8000 2003	Timer Counter 1 Command Mode	W	—	SIO	
0060	8000 0060	8000 3000	Reset X-Bus (Mouse) IRQ12 and Keyboard	R	—	SIO	
0061	8000 0061	8000 3001	NMI Status and Control	R/W	—	SIO	
0062	8000 0062	8000 3002	Reserved for Keyboard/Mouse	R/W	—	KBD	4
0064	8000 0064	8000 3004	Keyboard/Mouse	R	—	KBD	4
0066	8000 0066	8000 3006	Reserved for Keyboard/Mouse	R/W	—	KBD	4
0070	8000 0070	8000 3010	RTC Addr and NMI Enable	W	—	SIO	
0071	8000 0071	8000 3011	RTC Read/Write	R/W	—	RTC	4
0074	8000 0074	8000 3014	NV RAM Addr Strobe 0	W	—	NVR	4
0075	8000 0075	8000 3015	NV RAM Addr Strobe 1	W	—	NVR	4
0077	8000 0077	8000 3017	NV RAM Data Port	R/W	—	NVR	4
0078	8000 0078	8000 3018	BIOS Timer	R/W	—	SIO	
0079	8000 0079	8000 3019	BIOS Timer	R/W	—	SIO	
007A	8000 007A	8000 301A	BIOS Timer	R/W	—	SIO	
007B	8000 007B	8000 301B	BIOS Timer	R/W	—	SIO	
0080	8000 0080	8000 4000	DMA Page Register Reserved	R/W	—	SIO	
0081	8000 0081	8000 4001	DMA Channel 2 Page Register	R/W	—	SIO	
0082	8000 0082	8000 4002	DMA Channel 3 Page Register	R/W	—	SIO	
0083	8000 0083	8000 4003	DMA Channel 1 Page Register	R/W	—	SIO	
0084	8000 0084	8000 4004	DMA Page Register Reserved	R/W	—	SIO	
0085	8000 0085	8000 4005	DMA Page Register Reserved	R/W	—	SIO	
0086	8000 0086	8000 4006	DMA Page Register Reserved	R/W	—	SIO	
0087	8000 0087	8000 4007	DMA Channel 0 Page Register	R/W	—	SIO	
0088	8000 0088	8000 4008	DMA Page Register Reserved	R/W	—	SIO	
0089	8000 0089	8000 4009	DMA Channel 6 Page Register	R/W	—	SIO	
008A	8000 008A	8000 400A	DMA Channel 7 Page Register	R/W	—	SIO	
008B	8000 008B	8000 400B	DMA Channel 5 Page Register	R/W	—	SIO	
008C	8000 008C	8000 400C	DMA Page Register Reserved	R/W	—	SIO	
008D	8000 008D	8000 400D	DMA Page Register Reserved	R/W	—	SIO	
008E	8000 008E	8000 400E	DMA Page Register Reserved	R/W	—	SIO	
008F	8000 008F	8000 400F	DMA Low Page Register Refresh	R/W	—	SIO	
0090	8000 0090	8000 4010	DMA Page Register Reserved	R/W	—	SIO	
0092	8000 0092	8000 4012	Special Port 92 Register	R/W	—	660	
0094	8000 0094	8000 4014	DMA Page Register Reserved	R/W	—	SIO	5
0095	8000 0095	8000 4015	DMA Page Register Reserved	R/W	—	SIO	
0096	8000 0096	8000 4016	DMA Page Register Reserved	R/W	—	SIO	
0098	8000 0098	8000 4018	DMA Page Register Reserved	R/W	—	SIO	
009C	8000 009C	8000 401C	DMA Page Register Reserved	R/W	—	SIO	
009D	8000 009D	8000 401D	DMA Page Register Reserved	R/W	—	SIO	
009E	8000 009E	8000 401E	DMA Page Register Reserved	R/W	—	SIO	
009F	8000 009F	8000 401F	DMA Low Page Register Refresh	R/W	—	SIO	

Table 92. Combined Direct Access ISA Bus I/O Register Listing (Continued)

ISA Port	Contiguous Mode Addr[1]	Non-Contig Mode Addr[1]	Description	R/W	Set To[2]	Loc[3]	Note
00A0	8000 00A0	8000 5000	INT2 Control Register	R/W	—	SIO	
00A1	8000 00A1	8000 5001	INT2 Mask Register	R/W	—	SIO	
00C0	8000 00C0	8000 6000	DMA2 CH0 Base and Current Addr	R/W	—	SIO	
00C2	8000 00C2	8000 6002	DMA2 CH0 Base and Current Cnt	R/W	—	SIO	
00C4	8000 00C4	8000 6004	DMA2 CH1 Base and Current Addr	R/W	—	SIO	
00C6	8000 00C6	8000 6006	DMA2 CH1 Base and Current Cnt	R/W	—	SIO	
00C8	8000 00C8	8000 6008	DMA2 CH2 Base and Current Addr	R/W	—	SIO	
00CA	8000 00CA	8000 600A	DMA2 CH2 Base and Current Cnt	R/W	—	SIO	
00CC	8000 00CC	8000 600C	DMA2 CH3 Base and Current Addr	R/W	—	SIO	
00CE	8000 00CE	8000 600E	DMA2 CH3 Base and Current Cnt	R/W	—	SIO	
00D0	8000 000D	8000 6010	DMA2 Status(R) Command(W)	R/W	—	SIO	
00D2	8000 00D2	8000 6012	DMA2 Soft Request	W	—	SIO	
00D4	8000 00D4	8000 6014	DMA2 Write Single Mask Bit	W	—	SIO	
00D6	8000 00D6	8000 6016	DMA2 Write Mode	W	—	SIO	
00D8	8000 00D8	8000 6018	DMA2 Clear Byte Pointer	W	—	SIO	
00DA	8000 00DA	8000 601A	DMA2 Master Clear	W	—	SIO	
00DC	8000 00DC	8000 601C	DMA2 Clear Mask	W	—	SIO	
00DE	8000 00DE	8000 601E	DMA2 R/W All Mask Register Bits	R/W	—	SIO	
00F0	8000 00F0	8000 7010	Coprocessor Error Reg - Reserved	R/W	—	SIO	
01F0-01F7	8000 01F0-7	8000 F010-7	IDE Register 0	R/W	xx	332	
0200	8000 0200		Game Port			AUD	
0220-0223	8000 0220-3		Sound Blaster L/R FM Registers	R/W		AUD	
0224	8000 0224		Sound Blaster Mixer Address Register	R/W		AUD	
0225	8000 0225		Sound Blaster Mixer Data Register	R/W		AUD	
0226	8000 0226		Sound Blaster Reset Register	W/O		AUD	
022A	8000 022A		Sound Blaster Read Data Port	W/O		AUD	
022C	8000 022C		Sound Blaster Command/Status	W/O		AUD	
0278	8000 0278	8001 3018	LPT 3	R/W	xx	332	
0279	8000 0279	8001 3019	LPT 3	R/W	xx	332	
027A	8000 027A	8001 301A	LPT 3	R/W	xx	332	
027B	8000 027B	8001 301B	LPT 3	R/W	xx	332	
027C	8000 027C	8001 301C	LPT 3	R/W	xx	332	
027D	8000 027D	8001 301D	LPT 3	R/W	xx	332	
027E	8000 027E	8001 301E	LPT 3	R/W	xx	332	
027F	8000 027F	8001 301F	LPT 3	R/W	xx	332	
02E0	8000 02E0	8001 7000	COM 4 Tertiary	R/W	xx	332	6
02E1	8000 02E1	8001 7001	COM 4 Tertiary	R/W	xx	332	6

Table 92. Combined Direct Access ISA Bus I/O Register Listing (Continued)

ISA Port	Contiguous Mode Addr[1]	Non-Contig Mode Addr[1]	Description	R/W	Set To[2]	Loc[3]	Note
02E2	8000 02E2	8001 7002	COM 4 Tertiary	R/W	xx	332	6
02E3	8000 02E3	8001 7003	COM 4 Tertiary	R/W	xx	332	6
02E4	8000 02E4	8001 7004	COM 4 Tertiary	R/W	xx	332	6
02E5	8000 02E5	8001 7005	COM 4 Tertiary	R/W	xx	332	6
02E6	8000 02E6	8001 7006	COM 4 Tertiary	R/W	xx	332	6
02E7	8000 02E7	8001 7007	COM 4 Tertiary	R/W	xx	332	6
02E8	8000 02E8	8001 7008	COM 3 or COM 4	R/W	xx	332	6
02E9	8000 02E9	8001 7009	COM 3 or COM 4	R/W	xx	332	6
02EA	8000 02EA	8001 700A	COM 3 or COM 4	R/W	xx	332	6
02EB	8000 02EB	8001 700B	COM 3 or COM 4	R/W	xx	332	6
02EC	8000 02EC	8001 700C	COM 3 or COM 4	R/W	xx	332	6
02ED	8000 02ED	8001 700D	COM 3 or COM 4	R/W	xx	332	6
02EE	8000 02EE	8001 700E	COM 3 or COM 4	R/W	xx	332	6
02EF	8000 02EF	8001 700F	COM 3 or COM 4	R/W	xx	332	6
02F8	8000 02F8	8001 7018	COM 2	R/W	xx	332	
02F9	8000 02F9	8001 7019	COM 2	R/W	xx	332	
02FA	8000 02FA	8001 701A	COM 2	R/W	xx	332	
02FB	8000 02FB	8001 701B	COM 2	R/W	xx	332	
02FC	8000 02FC	8001 701C	COM 2	R/W	xx	332	
02FD	8000 02FD	8001 701D	COM 2	R/W	xx	332	
02FE	8000 02FE	8001 701E	COM 2	R/W	xx	332	
02FF	8000 02FF	8001 701F	COM 2	R/W	xx	332	
0330	8000 0330		MIDI Transmit/Receive Register			AUD	
0331	8000 0331		MIDI Status/Command Register			AUD	
0338	8000 0338	8001 9018	COM 3 Primary	R/W	xx	332	7
0339	8000 0339	8001 9019	COM 3 Primary	R/W	xx	332	7
033A	8000 033A	8001 901A	COM 3 Primary	R/W	xx	332	7
033B	8000 033B	8001 901B	COM 3 Primary	R/W	xx	332	7
033C	8000 033C	8001 901C	COM 3 Primary	R/W	xx	332	7
033D	8000 033D	8001 901D	COM 3 Primary	R/W	xx	332	7
033E	8000 033E	8001 901E	COM 3 Primary	R/W	xx	332	7
033F	8000 033F	8001 901F	COM 3 Primary	R/W	xx	332	7
0370- 0377			Secondary Floppy (Reserved)			332	
0378	8000 0378	8001 B018	LPT 2	R/W	xx	332	6
0379	8000 0379	8001 B019	LPT 2	R/W	xx	332	6
037A	8000 037A	8001 B01A	LPT 2	R/W	xx	332	6

Table 92. Combined Direct Access ISA Bus I/O Register Listing (Continued)

ISA Port	Contiguous Mode Addr[1]	Non-Contig Mode Addr[1]	Description	R/W	Set To[2]	Loc[3]	Note
037B	8000 037B	8001 B01B	LPT 2	R/W	xx	332	6
037C	8000 037C	8001 B01C	LPT 2	R/W	xx	332	6
037D	8000 037D	8001 B01D	LPT 2	R/W	xx	332	6
037E	8000 037E	8001 B01E	LPT 2	R/W	xx	332	6
037F	8000 037F	8001 B01F	LPT 2	R/W	xx	332	6
0388							
0389							
038A							
038B							
0398	8000 0398	8001 C018	Super I/O Index Address	R/W	xx	332	
0399	8000 0399	8001 C019	Super I/O Data Address	R/W	xx	332	
03BC	8000 03BC	8001 D01C	LPT 1	R/W	xx	332	
03BD	8000 03BD	8001 D01D	LPT 1	R/W	xx	332	
03BE	8000 03BE	8001 D01E	LPT 1	R/W	xx	332	
03E8	8000 03E8	8001 F008	COM 3	R/W	xx	332	6
03E9	8000 03E9	8001 F009	COM 3	R/W	xx	332	6
03EA	8000 03EA	8001 F00A	COM 3	R/W	xx	332	6
03EB	8000 03EB	8001 F00B	COM 3	R/W	xx	332	6
03EC	8000 03EC	8001 F0OC	COM 3	R/W	xx	332	6
03ED	8000 03ED	8001 F00D	COM 3	R/W	xx	332	6
03EE	8000 03EE	8001 F00E	COM 3	R/W	xx	332	6
03EF	8000 03EF	8001 F00F	COM 3	R/W	xx	332	6
03F0	8000 03F0	8001 F010	Primary Floppy Digital Output	W	xx	332	8
03F1	8000 03F1	8001 F011	Primary Floppy Digital Output	W	xx	332	8
03F2	8000 03F2	8001 F012	Primary Floppy Digital Output	W	xx	332	8, 9
03F3	8000 03F3	8001 F013	Primary Floppy Digital Output (Also, Media Sense)	W	xx	332	8
03F4	8000 03F4	8001 F014	Primary Floppy Digital Output	W	xx	332	8
03F5	8000 03F5	8001 F015	Primary Floppy Digital Output	W	xx	332	8
03F6	8000 03F6	8001 F016	IDE Register 1 (Reserved)	W	xx	332	8
03F7	8000 03F7	8001 F017	IDE Register 1 (Reserved)	W	xx	332	8
03F8	8000 03F8	8001 F018	COM 1	R/W	xx	332	
03FA	8000 03FA	8001 F01A	COM 1	R/W	xx	332	
03FB	8000 03FB	8001 F01B	COM 1	R/W	xx	332	
03FC	8000 03FC	8001 F01C	COM 1	R/W	xx	332	
03FD	8000 03FD	8001 F01D	COM 1	R/W	xx	332	
03FE	8000 03FE	8001 F01E	COM 1	R/W	xx	332	

Table 92. Combined Direct Access ISA Bus I/O Register Listing (Continued)

ISA Port	Contiguous Mode Addr[1]	Non-Contig Mode Addr[1]	Description	R/W	Set To[2]	Loc[3]	Note
03FF	8000 03FF	8001 F01F	COM 1	R/W	xx	332	
040B	8000 040B	8002 0006	DMA1 Extended Mode	W	—	SIO	
0410	8000 0410	8002 0010	CH0 Scatter/Gather Command	W	—	SIO	
0411	8000 0411	8002 0011	CH1 Scatter/Gather Command	W	—	SIO	
0412	8000 0412	8002 0012	CH2 Scatter/Gather Command	W	—	SIO	
0413	8000 0413	8002 0013	CH3 Scatter/Gather Command	W	—	SIO	
0415	8000 0415	8002 0015	CH5 Scatter/Gather Command	W	—	SIO	
0416	8000 0416	8002 0016	CH6 Scatter/Gather Command	W	—	SIO	
0417	8000 0417	8002 0017	CH7 Scatter/Gather Command	W	—	SIO	
0418	8000 0418	8002 0018	CH0 Scatter/Gather Status	R	—	SIO	
0419	8000 0419	8002 0019	CH1 Scatter/Gather Status	R	—	SIO	
041A	8000 041A	8002 001A	CH2 Scatter/Gather Status	R	—	SIO	
041B	8000 041B	8002 001B	CH3 Scatter/Gather Status	R	—	SIO	
041D	8000 041D	8002 001D	CH5 Scatter/Gather Status	R	—	SIO	
041E	8000 041E	8002 001E	CH6 Scatter/Gather Status	R	—	SIO	
041F	8000 041F	8002 001F	CH7 Scatter/Gather Status	R	—	SIO	
0420	8000 0420	8002 1000	CH0 Scatter/Gather Pointer	R/W	—	SIO	
0421	8000 0421	8002 1001	CH0 Scatter/Gather Pointer	R/W	—	SIO	
0422	8000 0422	8002 1002	CH0 Scatter/Gather Pointer	R/W	—	SIO	
0423	8000 0423	8002 1003	CH0 Scatter/Gather Pointer	R/W	—	SIO	
0424	8000 0424	8002 1004	CH1 Scatter/Gather Pointer	R/W	—	SIO	
0425	8000 0425	8002 1005	CH1 Scatter/Gather Pointer	R/W	—	SIO	
0426	8000 0426	8002 1006	CH1 Scatter/Gather Pointer	R/W	—	SIO	
0427	8000 0427	8002 1007	CH1 Scatter/Gather Pointer	R/W	—	SIO	
0428	8000 0428	8002 1008	CH2 Scatter/Gather Pointer	R/W	—	SIO	
0429	8000 0429	8002 1009	CH2 Scatter/Gather Pointer	R/W	—	SIO	
042A	8000 042A	8002 100A	CH2 Scatter/Gather Pointer	R/W	—	SIO	
042B	8000 042B	8002 100B	CH2 Scatter/Gather Pointer	R/W	—	SIO	
042C	8000 042C	8002 100C	CH3 Scatter/Gather Pointer	R/W	—	SIO	
042D	8000 042D	8002 100D	CH3 Scatter/Gather Pointer	R/W	—	SIO	
042E	8000 042E	8002 100E	CH3 Scatter/Gather Pointer	R/W	—	SIO	
042F	8000 042F	8002 100F	CH3 Scatter/Gather Pointer	R/W	—	SIO	
0434	8000 0434	8002 1014	CH5 Scatter/Gather Pointer	R/W	—	SIO	
0435	8000 0435	8002 1015	CH5 Scatter/Gather Pointer	R/W	—	SIO	
0436	8000 0436	8002 1016	CH5 Scatter/Gather Pointer	R/W	—	SIO	
0437	8000 0437	8002 1017	CH5 Scatter/Gather Pointer	R/W	—	SIO	
0438	8000 0438	8002 1018	CH6 Scatter/Gather Pointer	R/W	—	SIO	
0439	8000 0439	8002 1019	CH6 Scatter/Gather Pointer	R/W	—	SIO	
043A	8000 043A	8002 101A	CH6 Scatter/Gather Pointer	R/W	—	SIO	
043B	8000 043B	8002 101B	CH6 Scatter/Gather Pointer	R/W	—	SIO	
043C	8000 043C	8002 101C	CH7 Scatter/Gather Pointer	R/W	—	SIO	
043D	8000 043D	8002 101D	CH7 Scatter/Gather Pointer	R/W	—	SIO	

Table 92. Combined Direct Access ISA Bus I/O Register Listing (Continued)

ISA Port	Contiguous Mode Addr[1]	Non-Contig Mode Addr[1]	Description	R/W	Set To[2]	Loc[3]	Note
043E	8000 043E	8002 101E	CH7 Scatter/Gather Pointer	R/W	—	SIO	
043F	8000 043F	8002 101F	CH7 Scatter/Gather Pointer	R/W	—	SIO	
0481	8000 0481	8002 4001	DMA CH2 High Page	R/W	—	SIO	
0482	8000 0482	8002 4002	DMA CH3 High Page	R/W	—	SIO	
0483	8000 0483	8002 4003	DMA CH1 High Page	R/W	—	SIO	
0487	8000 0487	8002 4007	DMA CH0 High Page	R/W	—	SIO	
0489	8000 0489	8002 4009	DMA CH6 High Page	R/W	—	SIO	
048A	8000 048A	8002 400A	DMA CH7 High Page	R/W	—	SIO	
048B	8000 048B	8002 400B	DMA CH5 High Page	R/W	—	SIO	
04D0	8000 04D0	8002 6010	Interrupt Control 1	R/W	—	SIO	
04D1	8000 04D1	8002 6011	Interrupt Control 2	R/W	—	SIO	
04D6	8000 04D6	8002 6016	DMA2 Extended Mode	W	—	SIO	
0534	8000 0534	8002 9014	Audio chip index address register	R/W		AUD	
0535	8000 0535	8002 9015	Audio chip indexed data register	R/W		AUD	
0536	8000 0536	8002 9016	Audio chip status register	R		AUD	
0537	8000 0537	8002 9017	Audio chip PIO data register	W/O		AUD	
0538	8000 0538	8002 9017	Audio chip control register 0			AUD	
0808	8000 0808	8004 0008	HDD Light	R/W	—	EPLD	
080C	8000 080C	8004 000C	Equipment Present	R	—	logic	4
080D	8000 080D	8004 000D	L2 Cache Status Reg	R	—	logic	4
0814	8000 0814	8004 0014	L2 Flush	W	—	660	
081C	8000 081C	8004 001C	System Control 81C	R/W	10	660	
0821	8000 0821	8004 1001	Memory Controller Misc	R/W	—	660	
082A	8000 082A	8004 100A	Power Mgmt Control Reg1	R/W	—	EPLD	11
082B	8000 082B	8004 100B	Power Mgmt Control Reg2	R/W	—	EPLD	11
0838			Interrupt Request 13 Active	W/R			
0840	8000 0840	8004 2000	Memory Parity Error Status	R	—	660	
0842	8000 0842	8004 2002	L2 Error Status	R	—	660	
0843	8000 0843	8004 2003	L2 Parity Read & Clear	R	—	660	
0844	8000 0844	8004 2004	Unsupported Transfer Type Error	R	—	660	
0850	8000 0850	8004 2010	I/O Map Type	W	—	660	
0852	8000 0852	8004 2012	Board ID	R	—	logic	4
0860	8000 0860	8004 3000	Freeze Clock Reg Low	R/W	—	EPLD	11
0862	8000 0862	8004 3002	Freeze Clock Reg High	R/W	—	EPLD	11
0866	8000 0862		Processor 1 PD Register	R		logic	
0867	8000 0867		Processor 2 PD Register	R		logic	
0868	8000 0868		I2C Control Register	W/R		logic	
086B	8000 086B		L2 Control Register	W/R		logic	
0870	8000 0870		Processor Sequence Register	R		logic	
0871	8000 0871		Processor Enable Register	W		logic	

Table 92. Combined Direct Access ISA Bus I/O Register Listing (Continued)

ISA Port	Contiguous Mode Addr[1]	Non-Contig Mode Addr[1]	Description	R/W	Set To[2]	Loc[3]	Note
0880	8000 0880	8004 4000	SIMM Presence Detect Slot 1/2	R	—	logic	4
0881	8000 0881	8004 4001	SIMM Presence Detect Slot 3/4	R	—	logic	4
	8000 0CF8		PCI/BCR Configuration Address	R/W	—	660	
	8000 0CFC		PCI/BCR Configuration Data	R/W	—	660	
	8080 08xx		PCI Type 0 Configuration Addr	R/W	—	660	
	8080 0800		Vendor Identification	R	—	SIO	
	8080 0801		Vendor Identification	R	—	SIO	
	8080 0802		Device Identification	R	—	SIO	
	8080 0803		Device Identification	R	—	SIO	
	8080 0804		Command	R/W	0F	SIO	
	8080 0805		Command	R/W	00	SIO	
	8080 0806		Device Status	R/W	—	SIO	
	8080 0807		Device Status	R/W	—	SIO	
	8080 0808		Revision Identification	R/W	—	SIO	
	8080 0840		PCI Control	R/W	21	SIO	
	8080 0841		PCI Arbiter Control	R/W	00	SIO	
	8080 0842		PCI Arbiter Priority Control	R/W	04	SIO	
	8080 0843		PCI Arbiter Priority Control Extension	R/W	00	SIO	
	8080 0844		MEMCS# Control	R/W	00	SIO	
	8080 0845		MEMCS# Bottom of Hole	R/W	10	SIO	
	8080 0846		MEMCS# Top of Hole	R/W	0F	SIO	
	8080 0847		MEMCS# Top of Memory	R/W	00	SIO	
	8080 0848		ISA Address Decoder Control	R/W	F1	SIO	
	8080 0849		ISA Address Decoder ROM Block	R/W	00	SIO	
	8080 084A		ISA Address Bottom of Hole	R/W	10	SIO	
	8080 084B		ISA Address Top of Hole	R/W	0F	SIO	
	8080 084C		ISA Controller Recovery Timer	R/W	56	SIO	
	8080 084D		ISA Clock Divisor	R/W	10	SIO	
	8080 084E		Utility Bus Chip Select A	R/W	07	SIO	
	8080 084F		Utility Bus Chip Select B	R/W	FF	SIO	
	8080 0854		MEMCS# Attribute Register #1	R/W	—	SIO	
	8080 0855		MEMCS# Attribute Register #2	R/W	—	SIO	
	8080 0856		MEMCS# Attribute Register #3	R/W	—	SIO	
	8080 0857		Scatter/Gather Relocation Base	R/W	—	SIO	
	8080 0860		PIRQ Route Control 0	R/W	0F	SIO	
	8080 0861		PIRQ Route Control 1	R/W	0F	SIO	
	8080 0862		PIRQ Route Control 2	R/W	80	SIO	
	8080 0863		PIRQ Route Control 3 (Unused)	R/W	80	SIO	
	8080 0880		BIOS Timer Base Address	R/W	—	SIO	
	8080 0881		BIOS Timer Base Address	R/W	—	SIO	
	8080 10xx		PCI Type 0 Configuration Addr	R/W	—	660	
	8080 20xx		PCI Type 0 Configuration Addr	R/W	—	660	

Table 92. Combined Direct Access ISA Bus I/O Register Listing (Continued)

ISA Port	Contiguous Mode Addr[1]	Non-Contig Mode Addr[1]	Description	R/W	Set To[2]	Loc[3]	Note
	8080 40xx		PCI Type 0 Configuration Addr	R/W	—	660	
	8080 80xx		PCI Type 0 Configuration Addr	R/W	—	660	11
	8081 00xx		PCI Type 0 Configuration Addr	R/W	—	660	11
	8082 00xx		PCI Type 0 Configuration Addr	R/W	—	660	11
	8084 00xx		PCI Type 0 Configuration Addr	R/W	—	660	11
	8088 00xx		PCI Type 0 Configuration Addr	R/W	—	660	11
	8090 00xx		PCI Type 0 Configuration Addr	R/W	—	660	11
	80A0 00xx		PCI Type 0 Configuration Addr	R/W	—	660	11
	80C0 00xx		PCI Type 0 Configuration Addr	R/W	—	660	11
	BFFF EFF0	BFFF EFF0	System Error Addr	R	—	660	
	BFFF FFF0	BFFF FFF0	Interrupt Vector	R	—	660	
	FFFF FFF0	FFFF FFF0	Flash Write Addr/Data	W	—	660	
	FFFF FFF1	FFFF FFF1	Flash Lock Out	W	—	660	

Notes:

1 The first 5 hex digits in the contiguous and non-contiguous mode columns represent the memory page number for which the protection attributes may be set in contiguous I/O mode. That is, devices having the same first five digits in this column will have the same attributes in the memory page table.

2 In the Set To column, a long dash — means that the initialization firmware does not write to this register. The register is either not used, not written to, or the value of it depends on changing circumstances.
If the word Memory appears, please refer to the System Memory section of the 660 User's Manual.

3 KBD = Keyboard/Mouse Controller
RTC = Real Time Clock, also known as the TOD (Time Of Day clock)
NVR = Non-Volatile RAM, in the same package as the RTC
660 = The 660 Bridge.

4 The control signals for these ports are partially decoded by the SIO. The System I/O EPLD completes the decodes, and issues control signals to the registers, which are usually X-bus buffers.

5 Port 94 may be used by a certain video controller (e.g., Weitek™ 9100). The SIO chip positively decodes this port; therefore, bus contention may arise when both devices claim the PCI cycle to this port address. Bus contention results in invalid data and possible harm to the hardware.

6 For best I/O protection in non-contiguous I/O mode, do not use these ports. These mix addresses with other devices on the same page.

7 Use of this address range is recommended.

8 For best I/O protection in non-contiguous I/O mode, do not use these ports. These mix addresses with other devices on the same page; however, media sense is not supported at the secondary address range.

9 Ports 60, 92, 3F2, and 0F are partially contained in the CORAL chip. They are subtractively decoded in the CORAL chip.

10 Set register 81C to C0h if an L2 is installed, else leave at reset value.

11 Not used.

Section 14
Riser

The riser is built with back to back connectors to support three ISA cards and two PCI cards. The connection to the motherboard is via a 2x120 Micro Channel^{TM}-style card edge connector. This riser has the general shape outlined below:

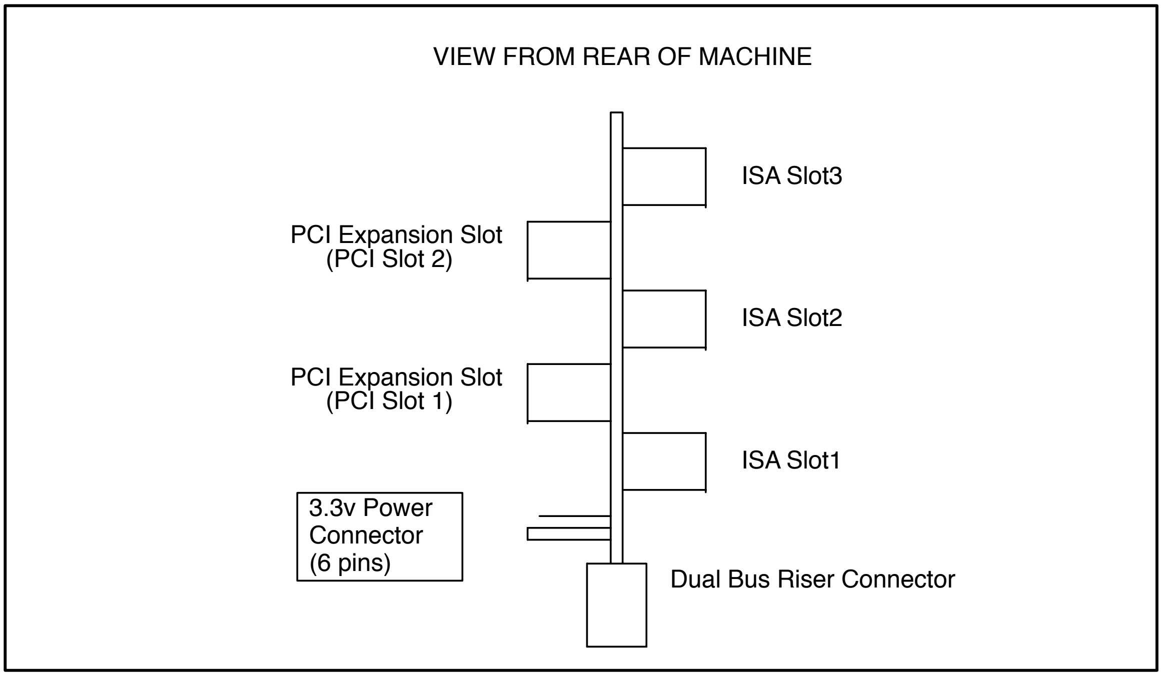


Figure 66. Dual Bus Riser Connector

14.1 Dual Bus Riser Connector Features

The riser connector at the motherboard is defined to support the following (although all of the pins defined are not used in the 604 SMP Reference Platform or on this riser):

- All standard ISA bus signals
- All 32-bit PCI bus signals
- Standard voltage requirements of the ISA and PCI slots
- Adapter present bits for up to 3 PCI slots
- ID select pins for up to 3 PCI slots
- Reset pins for up to 3 PCI slots

- Bus request/grant pairs for up to 3 PCI slots
- Separate clock lines for 2 PCI slots.

14.2 Special Comments about Riser and Motherboard Wiring

1. The IDSEL inputs going to the PCI slots are wired on the motherboard to buffered PCI A/D lines:

Buffered A/D 14	Connects to IDSEL SLOT 2 (upper)
Buffered A/D 13	Connects to IDSEL SLOT 1 (lower)

2. PCI Slot 2 interrupts A, B, C, and D are wired together on the riser and connected at connector pin INTA.
3. PCI Slot 1 interrupts A, B, C, and D are wired together on the riser and connected at connector pin INTB.
4. Separate clock lines are wired to each PCI slot. Equal clock line lengths are maintained system-wide for all PCI clock lines including those on the riser. Riser clock length allocations are 3.7 inches and 2.8 inches for slot 2 and slot 1 respectively.
5. At each PCI slot, both of the PCI presence bits are wired in parallel on the riser, and a single signal line from each slot is connected to a pin on the riser connector which is connected to the I/O interface EPLD. This component generates the corresponding presence bit in the equipment register (PCI card power levels are not distinguished).
6. The JTAG signals, TDI and TDO, are wired daisy chain fashion, and TDI is connected to TDO on the motherboard. The motherboard itself does not support JTAG. The other JTAG signals are wired socket to socket and to the motherboard connector on the riser.
7. Four other unused pins are connected to both PCI sockets and to the motherboard connector — SDONE, SBO, REQ64, and ACK64; however, the 64-bit PCI is not supported; nor are PCI caches. These signals are not connected on the motherboard.
8. The following ISA signals are buffered on the riser with a '125 buffer:
   - AEN
   - TC
   - BALE
   - ISA_RESET
9. The following signals are buffered on the riser with a '243 bidirectional buffer (They are pointed towards the riser until ISA_MASTER# activates):
   - IOW#
   - IOR#
   - SBHE#
   - ISA_REFRESH#

14.3 3.3v, 6-Pin Connector for PCI Slots

The riser card includes a 3.3 volt 6-pin connector which supplies power at the 3.3v pins of each PCI slot. The specifications of the 3.3 volt connector are:

- 3.3v ± 5%
- 5 amps maximum available output (current may be shared between slots)

14.4 Current Capacity of Connectors on the Riser

Table 93 summarizes the current capacity of the various connectors:

Table 93. Connector Capacity

Voltage	Motherboard Connector @ 1 amp/pin (1)	PCI Connector @ 1 amp/ pin	ISA Connector @ 3 amps/ pin
+5v	33 amps	13 amps	9 amps
–5v	1 amp	n/a	3 amps
+12v	3 amps	1 amp	3 amps
–12v	2 amps	1 amp	3 amps
+3.3v	n/a	12 amps(2)	n/a
Common GND	36 amps(3)	22 amps	12 amps

Note:
These are connector capacities, and the amounts specified here may not be available from the motherboard. See Section 2 for the allocation of current for this connector.

14.5 Riser Connector

```
Top view
                                 ..
           B120               B79 B78              B34 B33        B1
          ┌──────────────────┬───────────────────┬────────────┐
          │ x...............x│ x................x│ x.........x│
          │ x...............x│ x................x│ x.........x│
          └──────────────────┴───────────────────┴────────────┘
           A120               A79 A78              A34 A33        A1
```

Figure 67. Riser Connector

Table 94. Riser Connector Pin Assignments

Pin	Function	Pin	Function
A1	ISA IO CHCK#	B1	GROUND
A2	–5 VOLTS	B2	–12 VOLTS
A3	SD(7)	B3	ISA RESET
A4	SD(6)	B4	–12 VOLTS
A5	SD(5)	B5	ISA IRQ9
A6	SD(4)	B6	+12 VOLTS
A7	SD(3)	B7	ISA DRQ(2)
A8	SD(2)	B8	+12 VOLTS
A9	SD(1)	B9	ISA ZERO WS#
A10	SD(0)	B10	+12 VOLTS
A11	GROUND	B11	+5 VOLTS
A12	ISA IO CHRDY	B12	GROUND
A13	ISA AEN	B13	ISA SMEMW#
A14	ISA SA(19)	B14	ISA SMEMR#
A15	ISA SA(18)	B15	ISA IOW#

Table 94. Riser Connector Pin Assignments (Continued)

Pin	Function	Pin	Function
A16	ISA SA(17)	B16	ISA IOR#
A17	ISA SA(16)	B17	ISA DACK(3)#
A18	ISA SA(15)	B18	ISA DRQ(3)
A19	ISA SA(14)	B19	ISA DACK(1)#
A20	ISA SA(13)	B20	ISA DRQ(1)
A21	ISA SA(12)	B21	ISA REFRESH#
A22	GROUND	B22	GROUND
A23	PCI TRST# (res)	B23	+5 VOLTS
A24	+5 VOLTS	B24	ISA TCK (res)
A25	PCI TMS (res)	B25	GROUND
A26	PCI TDI (res)	B26	PCI TDO (res)
A27	+5 VOLTS	B27	GROUND
A28	PCI_IRQ9#	B28	+5 VOLTS
A29	RESERVED	B29	PCI_IRQ11#
A30	+5 VOLTS	B30	RESERVED
A31	RESERVED	B31	PCI PRESENT 1#
A32	+5 VOLTS	B32	GROUND
A33	PCI_GNT0#(res)	B33	PCI PRESENT 2#
A34	PCI_REQ0#(res)	B34	GROUND
A35	+5 VOLTS	B35	CLOCK SLOT 1
A36	PCI RESET 1#	B36	GROUND
A37	+5 VOLTS	B37	CLOCK SLOT 2
A38	PCI RESET 2#	B38	GROUND
A39	PCI GNT 1#	B39	GROUND
A40	GROUND	B40	PCI REQ 1#
A41	PCI GNT 2#	B41	PCI REQ 2#
A42	RESERVED	B42	+5 VOLTS
A43	+5 VOLTS	B43	GROUND
A44	PCI IDSEL 2	B44	RESERVED
A45	GROUND	B45	+5 VOLTS
A46	PCI A/D(30)	B46	PCI A/D(31)
A47	+5 VOLTS	B47	PCI A/D(29)
A48	PCI A/D(28)	B48	GROUND
A49	PCI A/D(26)	B49	PCI A/D(27)
A50	GROUND	B50	PCI A/D(25)
A51	PCI A/D(24)	B51	+5 VOLTS
A52	PCI IDSEL 1	B52	C/BE#(3)
A53	+5 VOLTS	B53	PCI A/D(23)
A54	PCI A/D(22)	B54	GROUND
A55	PCI A/D(20)	B55	PCI A/D(21)
A56	GROUND	B56	PCI A/D(19)
A57	PCI A/D(18)	B57	+5 VOLTS
A58	PCI A/D(16)	B58	PCI A/D(17)

Table 94. Riser Connector Pin Assignments (Continued)

Pin	Function	Pin	Function
A59	+5 VOLTS	B59	C/BE#(2)
A60	PCI FRAME#	B60	GROUND
A61	GROUND	B61	PCI IRDY#
A62	PCI TRDY#	B62	+5 VOLTS
A63	GROUND	B63	PCI DEVSEL#
A64	PCI STOP#	B64	GROUND
A65	+5 VOLTS	B65	PCI LOCK#
A66	PCI SDONE (res)	B66	PCI PERR#
A67	PCI SBO# (res)	B67	+5 VOLTS
A68	GROUND	B68	PCI SERR#
A69	PCI PAR	B69	+5 VOLTS
A70	PCI A/D(15)	B70	C/BE#(1)
A71	+5 VOLTS	B71	PCI A/D(14)
A72	PCI A/D(13)	B72	GROUND
A73	PCI A/D(11)	B73	PCI A/D(12)
A74	GROUND	B74	PCI A/D(10)
A75	PCI A/D(9)	B75	GROUND
A76	C/BE#(0)	B76	PCI A/D(8)
A77	+5 VOLTS	B77	PCI A/D(7)
A78	PCI A/D(6)	B78	+5 VOLTS
A79	PCI A/D(4)	B79	PCI A/D(5)
A80	GROUND	B80	PCI A/D(3)
A81	PCI A/D(2)	B81	GROUND
A82	PCI A/D(0)	B82	PCI A/D(1)
A83	+5 VOLTS	B83	+5 VOLTS
A84	PCI REQ64#	B84	PCI ACK64#
A85	+5 VOLTS	B85	+5 VOLTS
A86	+5 VOLTS	B86	+5 VOLTS
A87	GROUND	B87	GROUND
A88	ISA SA(11)	B88	GROUND
A89	+5 VOLTS	B89	ISA SYSCLK
A90	ISA SA(10)	B90	GROUND
A91	ISA SA(9)	B91	ISA_IRQ7
A92	ISA SA(8)	B92	ISA_IRQ6
A93	ISA SA(7)	B93	ISA_IRQ5
A94	ISA SA(6)	B94	ISA_IRQ4
A95	ISA SA(5)	B95	ISA_IRQ3
A96	ISA SA(4)	B96	ISA DACK(2)#
A97	ISA SA(3)	B97	ISA TC
A98	ISA SA(2)	B98	ISA BALE
A99	ISA SA(1)	B99	RESERVED
A100	ISA SA(0)	B100	GROUND
A101	GROUND	B101	ISA OSCILLATOR

Table 94. Riser Connector Pin Assignments (Continued)

Pin	Function	Pin	Function
A102	RESERVED	B102	+5 VOLTS
A103	ISA SBHE#	B103	ISA MEMCS16#
A104	LA(23)	B104	ISA IOCS16#
A105	LA(22)	B105	ISA_IRQ10
A106	LA(21)	B106	ISA_IRQ11
A107	LA(20)	B107	ISA_IRQ12
A108	LA(19)	B108	ISA_IRQ15
A109	LA(18)	B109	ISA_IRQ14
A110	LA(17)	B110	ISA DACK(0)#
A111	ISA MEMR#	B111	ISA DRQ(0)
A112	ISA MEMW#	B112	ISA DACK(5)#
A113	SD(8)	B113	ISA DRQ(5)
A114	SD(9)	B114	ISA DACK(6)#
A115	SD(10)	B115	ISA DRQ(6)
A116	SD(11)	B116	ISA DACK(7)#
A117	SD(12)	B117	ISA DRQ(7)
A118	SD(13)	B118	+5 VOLTS
A119	SD(14)	B119	ISA MASTER#
A120	SD(15)	B120	GROUND

14.6 PCI Connector (On Riser)

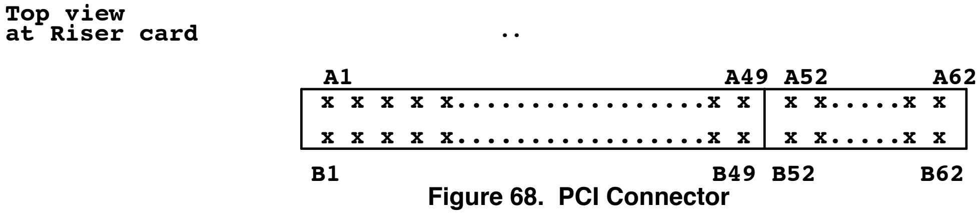


Figure 68. PCI Connector

Table 95. PCI Connector Pin Assignments

Pin	Function	Pin	Function
A1	TRST#	B1	–12 VOLTS#
A2	+12 VOLTS	B2	TCK
A3	TMS	B3	GROUND
A4	TDI	B4	TDO
A5	+5 VOLTS	B5	+5 VOLTS
A6	INTA#	B6	+5 VOLTS
A7	INTC#	B7	INTB#
A8	+5 VOLTS	B8	INTD#
A9	RESERVED	B9	PRESENT 1# *
A10	+5 VOLTS	B10	RESERVED
A11	RESERVED	B11	PRESENT 2# *

Table 95. PCI Connector Pin Assignments (Continued)

Pin	Function	Pin	Function
A12	GROUND	B12	GROUND
A13	GROUND	B13	GROUND
A14	RESERVED	B14	RESERVED
A15	RESET#	B15	GROUND
A16	+5 VOLTS	B16	CLK
A17	GNT#	B17	GROUND
A18	GROUND	B18	REQ#
A19	RESERVED	B19	+5 VOLTS
A20	A/D(30)	B20	A/D(31)
A21	+3.3V	B21	A/D(29)
A22	A/D(28)	B22	GROUND
A23	A/D(26)	B23	A/D(27)
A24	GROUND	B24	A/D(25)
A25	A/D(24)	B25	+3.3V
A26	IDSEL	B26	C/BE#(3)
A27	+3.3V	B27	A/D(23)
A28	A/D(22)	B28	GROUND
A29	A/D(20)	B29	A/D(21)
A30	GROUND	B30	A/D(19)
A31	A/D(18)	B31	+3.3V
A32	A/D(16)	B32	A/D(17)
A33	+3.3V	B33	C/BE#(2)
A34	FRAME#	B34	GROUND
A35	GROUND	B35	IRDY#
A36	TRDY#	B36	+3.3V
A37	GROUND	B37	DEVSEL#
A38	STOP#	B38	GROUND
A39	+3.3V	B39	LOCK#
A40	SDONE	B40	PERR#
A41	SBO#	B41	+3.3V
A42	GROUND	B42	SERR#
A43	PAR	B43	+3.3V
A44	A/D(15)	B44	C/BE#(1)
A45	+3.3V	B45	A/D(14)
A46	A/D(13)	B46	GROUND
A47	A/D(11)	B47	A/D(12)
A48	GROUND	B48	A/D(10)
A49	A/D(9)	B49	GROUND
A50	<Key>	B50	<Key>
A51	<Key>	B51	<Key>
A52	C/BE#(0)	B52	A/D(8)
A53	+3.3V	B53	A/D(7)
A54	A/D(6)	B54	+3.3V

Table 95. PCI Connector Pin Assignments (Continued)

Pin	Function	Pin	Function
A55	A/D(4)	B55	A/D(5)
A56	GROUND	B56	A/D(3)
A57	A/D(2)	B57	GROUND
A58	A/D(0)	B58	A/D(1)
A59	+5 VOLTS	B59	+5 VOLTS
A60	REQ64#	B60	ACK64#
A61	+5 VOLTS	B61	+5 VOLTS
A62	+5 VOLTS	B62	+5 VOLTS

Notes:

* The two card type bits, B9 and B11, are connected on the riser.

** Int A-D, A6, A7, B7,and B8 are connected together on the riser.

14.7 3.3v, 6-Pin Connector (On Riser)

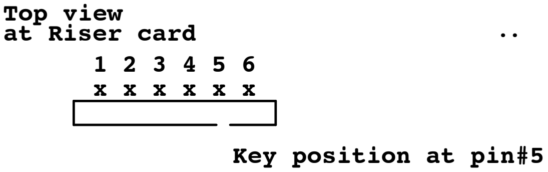


Figure 69. 3.3v, 6-Pin Connector

Table 96. 3.3v, 6-Pin Connector Pin Assignments

Pin	Function	Pin	Function
1-3	3.3 Volts	4-6	GROUND

14.8 ISA Connector (On Riser)

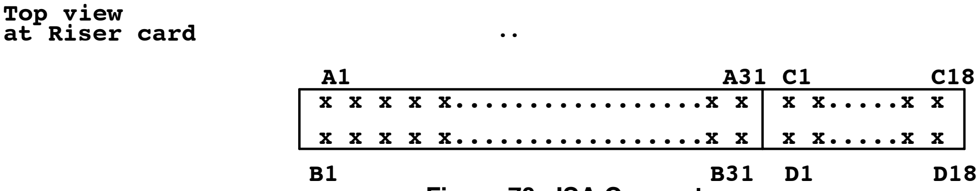


Figure 70. ISA Connector

Table 97. ISA Connector Pin Assignments

Pin	Function	Pin	Function
A1	IO CHCK#	B1	GROUND
A2	SD(7)	B2	RESET_DRV
A3	SD(6)	B3	+5 VOLTS
A4	SD(5)	B4	IRQ9
A5	SD(4)	B5	–5 VOLTS
A6	SD(3)	B6	DRQ2
A7	SD(2)	B7	–12 VOLTS

Table 97. ISA Connector Pin Assignments (Continued)

Pin	Function	Pin	Function
A8	SD(1)	B8	ZERO WS#
A9	SD(0)	B9	+12 VOLTS
A10	IO CHRDY	B10	GROUND
A11	AEN	B11	SMEMW#
A12	SA(19)	B12	SMEMR#
A13	SA(18)	B13	IOW#
A14	SA(17)	B14	IOR#
A15	SA(16)	B15	DACK3#
A16	SA(15)	B16	DRQ3
A17	SA(14)	B17	DACK1#
A18	SA(13)	B18	DRQ1
A19	SA(12)	B19	REFRESH
A20	SA(11)	B20	CLK
A21	SA(10)	B21	IRQ7
A22	SA(9)	B22	IRQ6
A23	SA(8)	B23	IRQ5
A24	SA(7)	B24	IRQ4
A25	SA(6)	B25	IRQ3
A26	SA(5)	B26	DACK2#
A27	SA(4)	B27	T/C
A28	SA(3)	B28	BALE
A29	SA(2)	B29	+5 VOLTS
A30	SA(1)	B30	Oscillator
A31	SA(0)	B31	GROUND
C1	SBHE#	D1	MEMCS16#
C2	LA(23)	D2	IO CS16#
C3	LA(22)	D3	IRQ10
C4	LA(21)	D4	IRQ11
C5	LA(20)	D5	IRQ12
C6	LA(19)	D6	IRQ15
C7	LA(18)	D7	IRQ14
C8	LA(17)	D8	DACK0#
C9	MEMR#	D9	DRQ0
C10	MEMW#	D10	DACK5#
C11	SD(8)	D11	DRQ5
C12	SD(9)	D12	DACK6#
C13	SD(10)	D13	DRQ6
C14	SD(11)	D14	DACK7#
C15	SD(12)	D15	DRQ7
C16	SD(13)	D16	+5 VOLTS
C17	SD(14)	D17	MASTER#
C18	SD(15)	D18	GROUND

Section 15
Power

15.1 External Power Supply Requirements

This section sets out the power supply requirements for the motherboard. They are achievable with low-cost PC power supplies.

Table 98. External Power Supply Requirements

Output	Tolerance		Max Ripple	Note
+5v	+5%	–4%	50 mv P–P	The combined 3.3v and 5.0v power cannot exceed 100W on standard 200W supplies. If more power or current is required, larger supplies can be used.
+12v	+5%	–5%	120 mv P–P	
–12v	+10%	–9%	120 mv P–P	
–5v	+10%	–10%	120 mv P–P	Not used on the motherboard. Routed from the power supply connector to the ISA slots on the PCI/ISA riser card.
+3.3v	+5%	–4%	50 mv P–P	See +5v supply note

15.1.1 Power Supply Current Requirements

Table 99 shows the approximate current capacities of the motherboard and of a typical standard 200W PC power supply.

Table 99. Approximate Current Capacities and Requirements (Amps)

Row	Element	3.3v (2)	+5v (2)	+12v	–12v	–5v (1)
1	Standard 200W power supply total capacity	20.0	20.0	8.0	0.5	0.5
2	Motherboard power connectors capacity	20.0	20.0	5.0	5.0	0.5
3	Maximum current that the 200W power supply can deliver to the motherboard connectors.	20.0	12.6 (4)	3.0 (4)	0.5 (5)	0.5
4	Current requirements of motherboard with two 8M SIMMs, 512K synchronous L2 cache, riser, keyboard, and mouse, running typical code. 66MHz CPU bus. 33MHz PCI bus.	2.8	4	0.1	0.02	0
5	Budgeted Current Requirement for CPU Slot 1	7.5	2.5	0.01	0	0
6	Budgeted Current Requirement for CPU Slot 2	7.5	2.5	0.01	0	0
7	Cheetah0 Current Requirements (66MHz bus & 132MHz 604).	5.6	0	0	0	0
8	Total available current for add-in memory, PCI/ISA cards, and peripherals with 200W supply. Row8 = Row1 – Row4 – Row5 – Row6	2.8 (3)	11	7.8	0.48	0.5
9	Total available current for add-in memory, PCI/ISA cards, and peripherals with 200W supply. Row9 = Row1 – Row4 – 2*Row7	6.0 (3)	11	7.8	0.48	0.5
10	Maximum available current for PCI/ISA cards thru motherboard. Row10 = Row3 – Row4 – Row5 – Row6	2.8 (3)	3.6	2.8	0.48	0.5
11	Maximum available current for PCI/ISA cards thru motherboard. Row11 = Row3 – Row4 – 2*Row7	6.0 (3)	3.6	2.8	0.48	0.5

Notes:

1. The –5v is not used on the motherboard, but it is routed from the power supply connector to the ISA slots.
2. The combined 3.3v and 5.0v power cannot exceed 100W on standard 200W supplies. If more power is required larger supplies can be used.
3. The 3.3v current for the PCI slots on the riser card is provided via a 6 pin connector on the motherboard (J50) and is connected to riser card connector J7 via a 6 conductor cable capable of carrying 9 amps.
4. Limited by the number and current capacity of the power supply connectors that can be plugged into the motherboard.
5. Limited by the capacity of the power supply.

15.1.2 Power Supply Power Requirements

Table 100 shows the approximate power capacities of the motherboard and of a typical standard 200W PC power supply.

Table 100. Approximate Power Capacities and Requirements (Watts)

Row	Element	+5v/3.3v	12v	–12v	–5v	Total
1	Standard 200W power supply total capacity	100	96	6.0	2.5	200
2	Base motherboard with two 8M SIMMs, 512K synchronous L2 cache, riser, keyboard, and mouse, running typical code. 66MHz CPU bus & 33MHz PCI bus.	31	1.2	0.24	0	32.5
3	Budgeted Power Requirements for CPU Slot 1	25	0.12	0	0	25
4	Budgeted Power Requirements for CPU Slot 2	25	0.12	0	0	25
5	Cheetah0 Power Requirements (66MHz bus & 132MHz 604)	15	0	0	0	15
6	Total available power for add-in memory, PCI/ISA cards, and peripherals with 200W supply. Row6 = Row1 – Row2 – Row3 – Row4	19	94.8	5.75	2.5	117
7	Total available power for add-in memory, PCI/ISA cards, and peripherals with 200W supply. Row7 = Row1 – Row2 – 2*Row5	39	96	5.75	2.5	137

15.1.3 Additional Power Supply Requirements

1. Overshoot on any voltage must be less than 10% of nominal and must decay to within the regulation band within 50 msec.
2. In any failure situation, the power supply must shut down before the +5v output reaches 6.5v, to give the motherboard a reasonable chance of surviving; however, damage may occur at any voltage above 5.5v.
3. Power_Good signal requirements:
   - The signal must be at a TTL down level when power is applied until >100 msec to 500 msec after the 5v supply has reached its minimum regulation level, and at TTL high level thereafter, as long as outputs are within regulation.
   - At turn–off, the Power Good signal must drop to a TTL low level before any output drops below its regulation limits.
   - The driver must be capable of driving 400 microamps or sinking 5 milliamps. The rise time/fall time must be less than 1 usec, 10—90%
4. The +5v rise time (10%-90%) shall be 3 msec to 100 msec with a maximum slope of 0.75 volts/msec for voltages above 1.5 volts for all loadings.
5. All supply voltages shall track within 50 msec of each other measured at the 50% point.

15.2 Onboard 3.6v Regulator

There is a 3.6 volt regulator on the reference design to support CPU cards with 3.6v devices. The input for this device is supplied by the +5v regulator.

Table 101. Specifications for 3.6v Regulator on the Motherboard

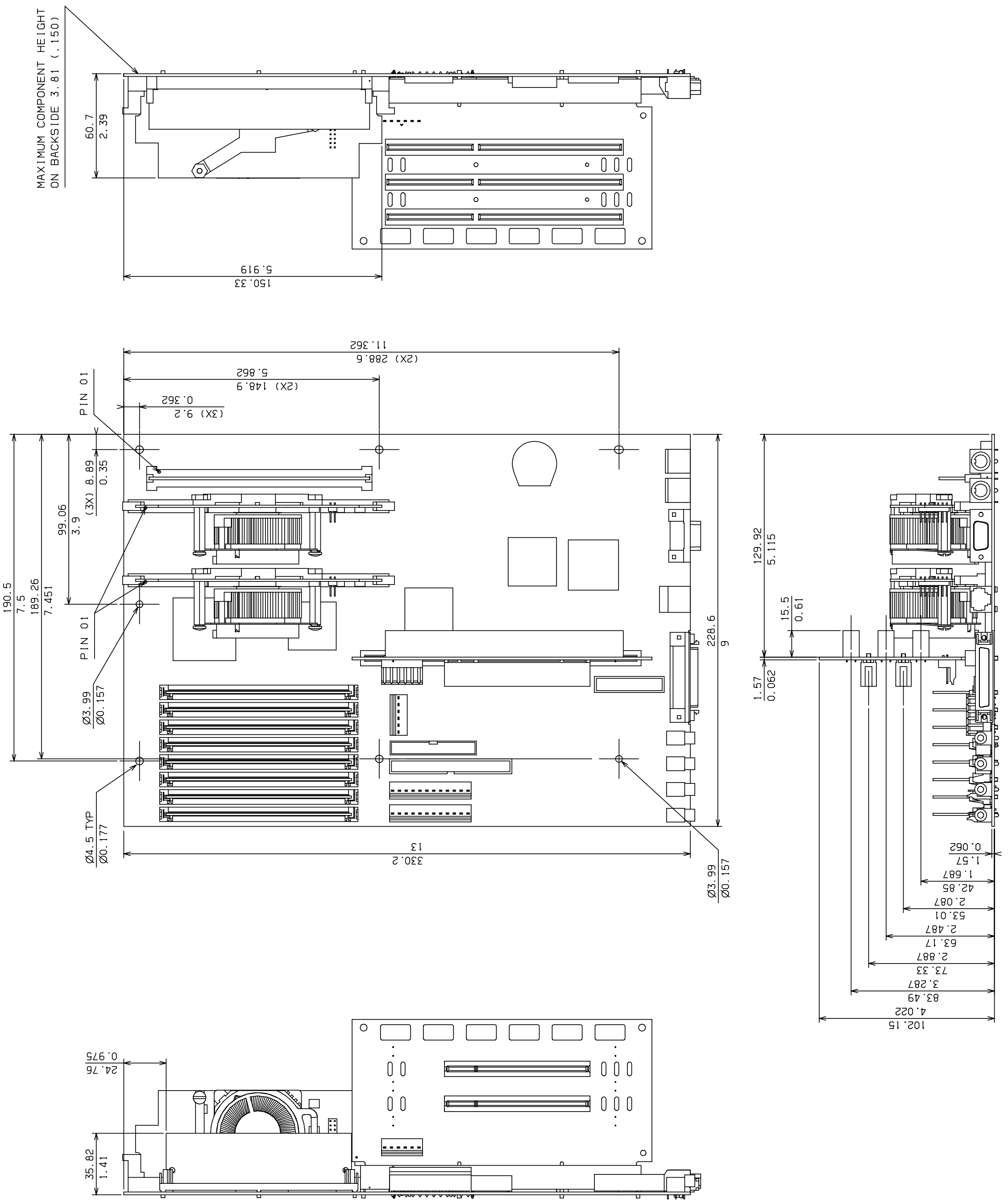
Specification	Value
Output Voltage	3.6v ± 3%
Output Current	0.01 A to 2 A
Input voltage	4.75v to 5.25v
Pass element maximum case temperature	110 °C
Tracking	In regulation <1ms after +5 reaches 4.75v.
Overcurrent	No current limit feature

Section 16
Mechanical

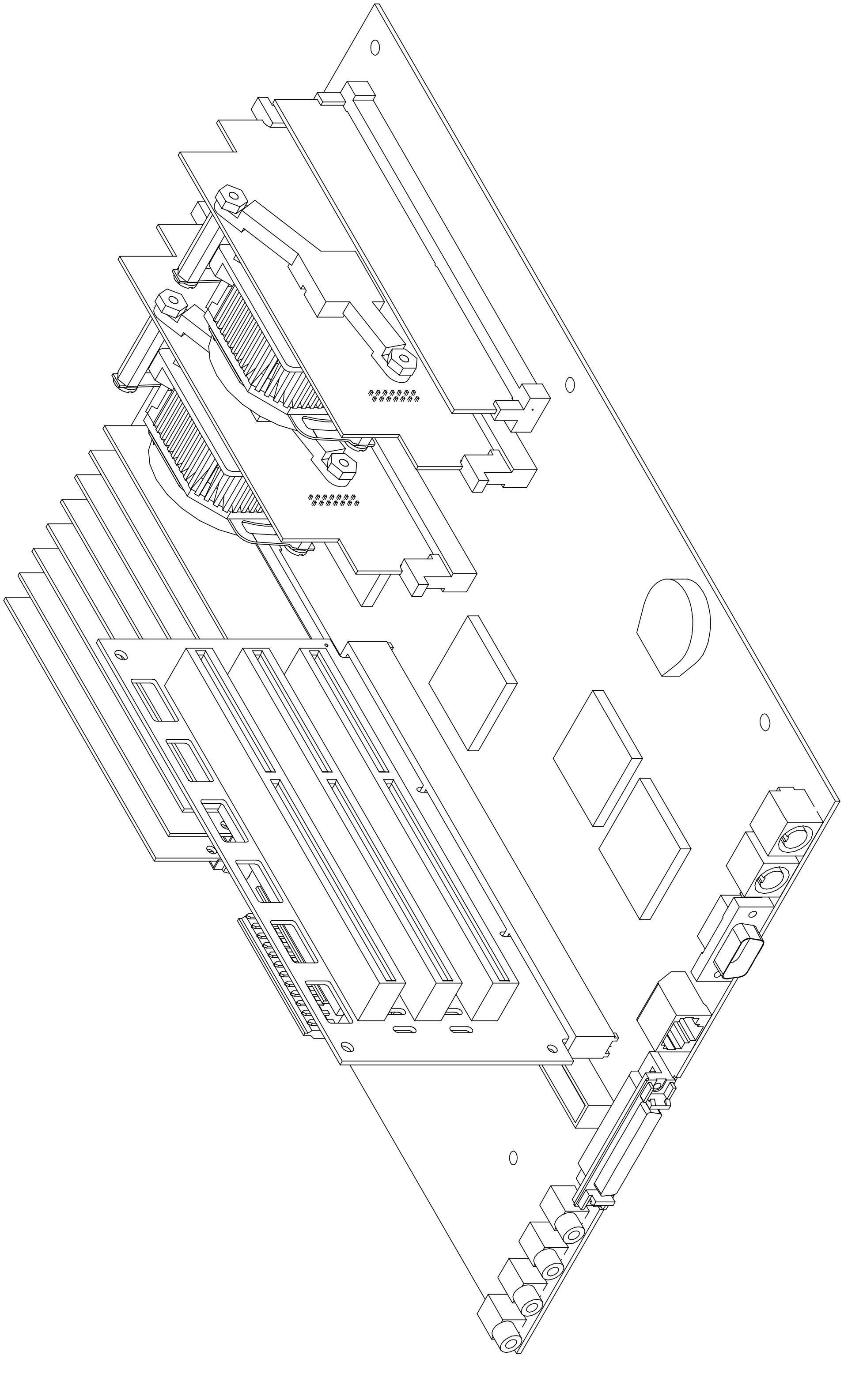
This section presents the various mechanical drawings associated with the reference design.

16.1 System Layout Drawings

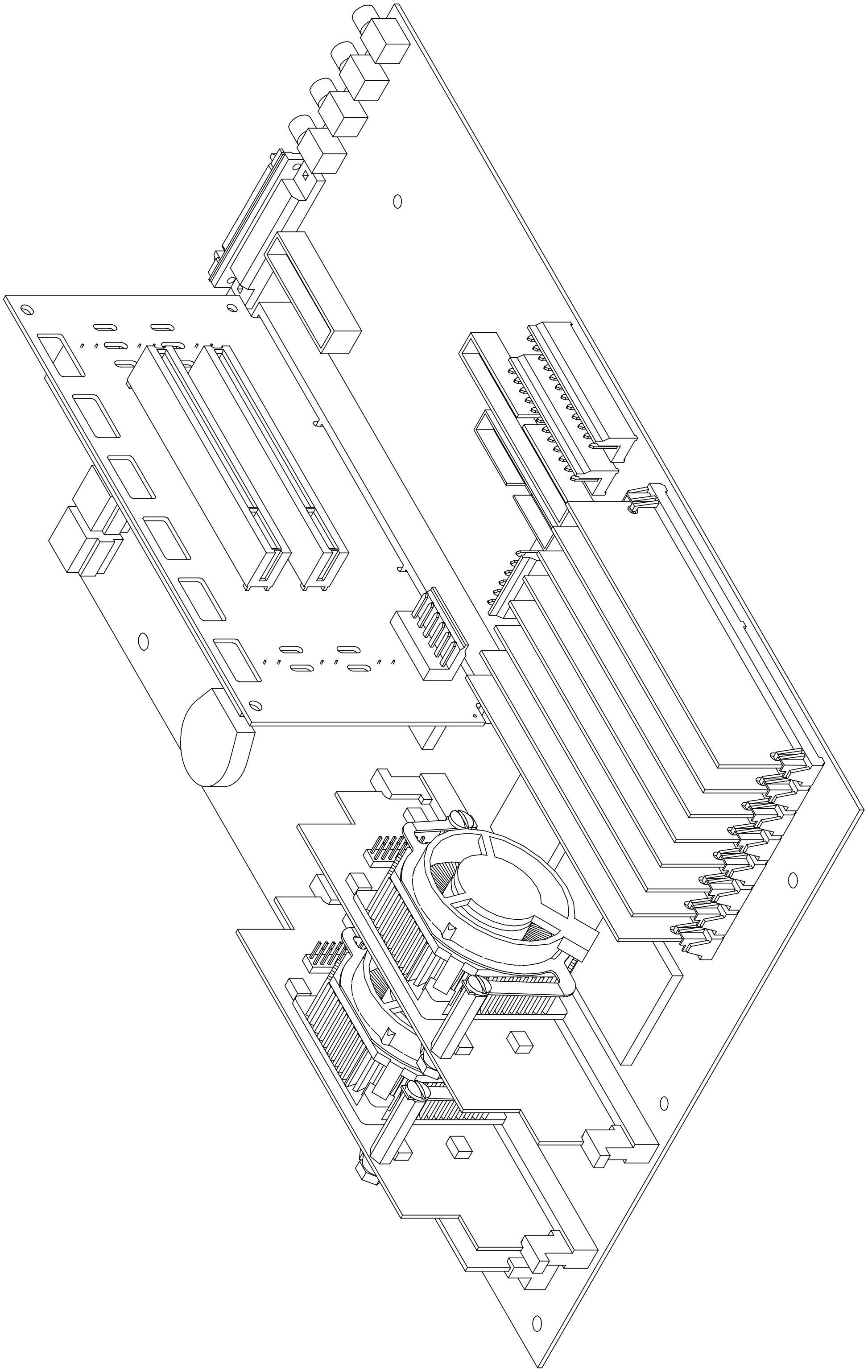
16.1.1 System Layout



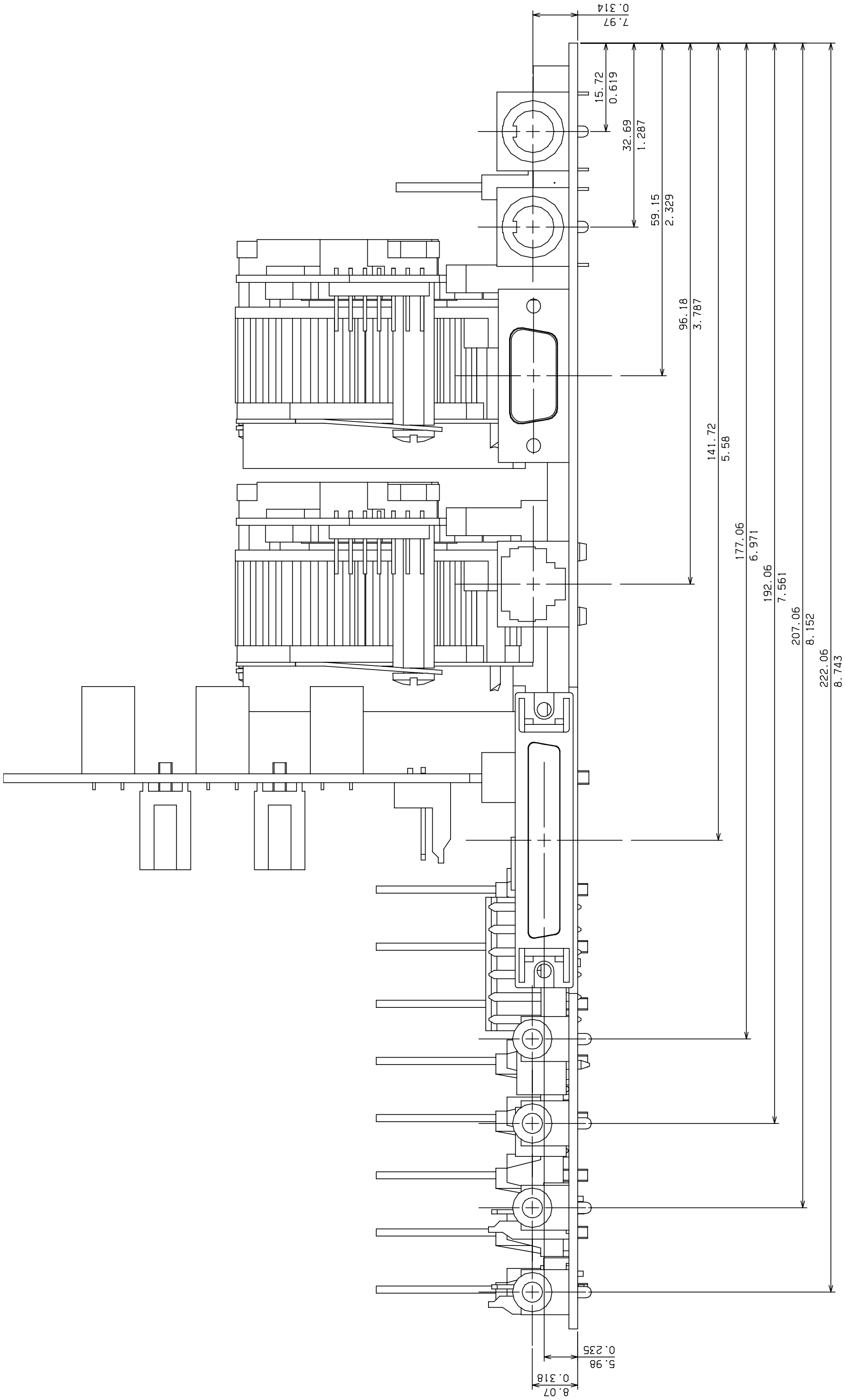
16.1.2 System 3D Sheet 1



16.1.3 System 3D Sheet 2

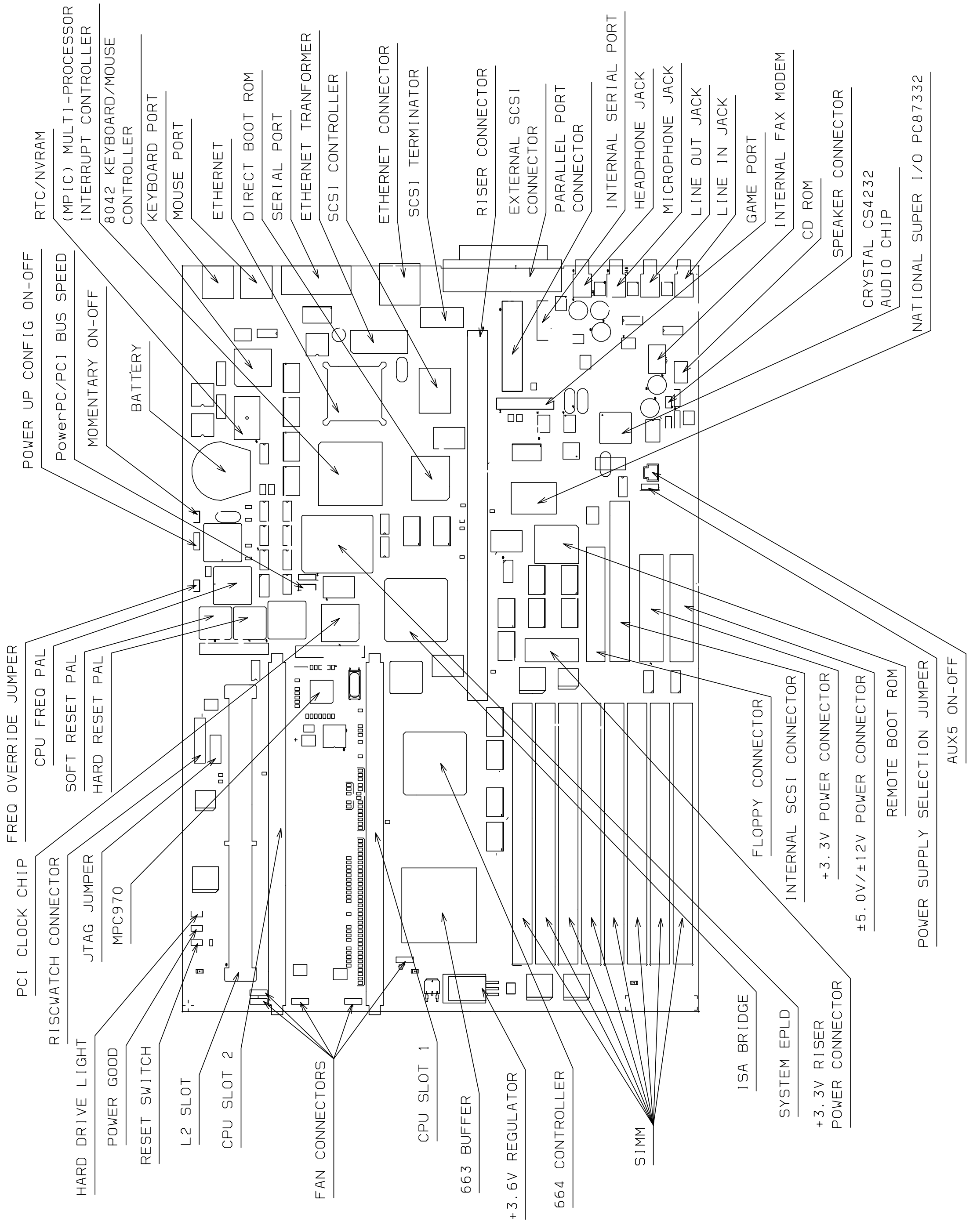


16.1.4 System Layout – Rear Panel Connector Locations



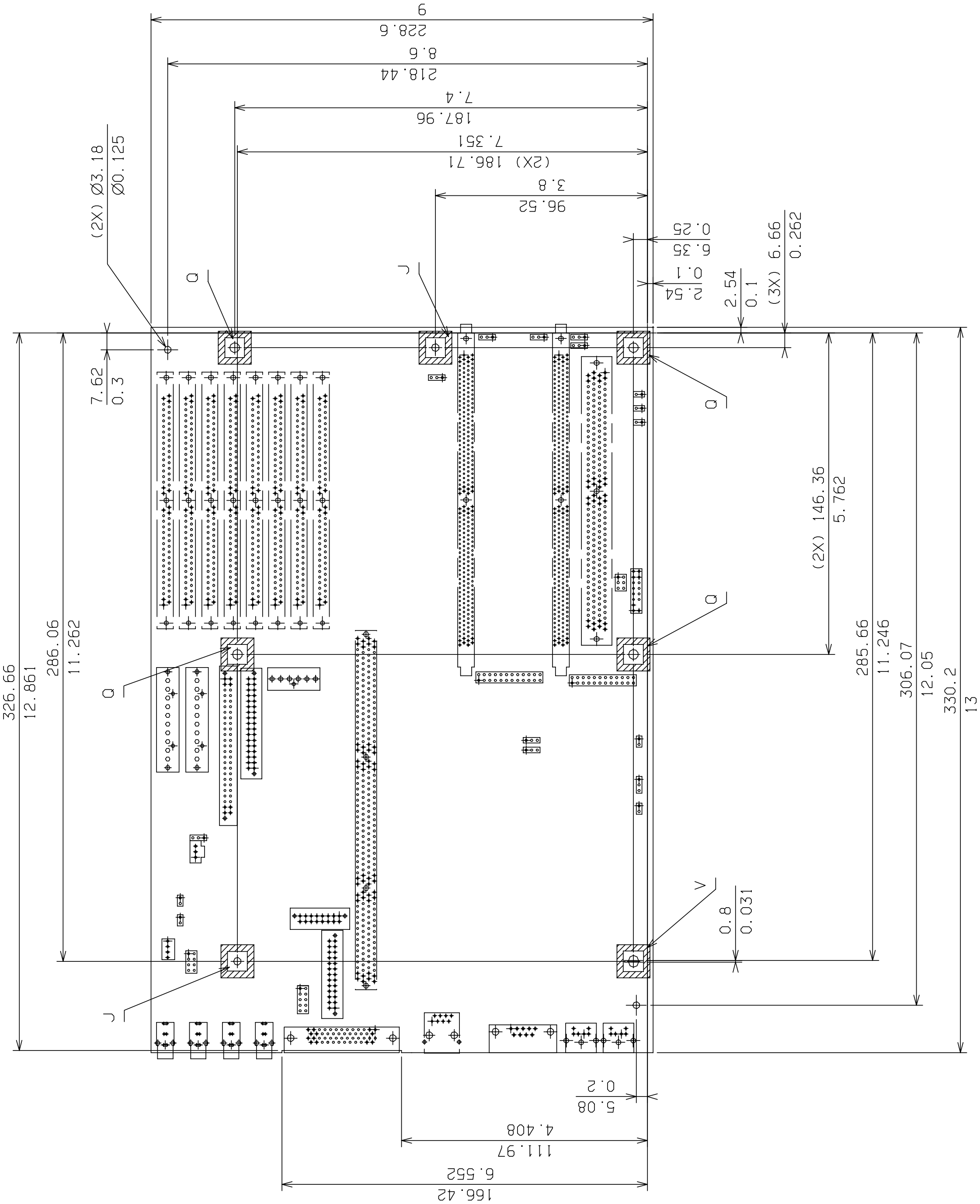
16.2 Motherboard

16.2.1 Motherboard Component Locator

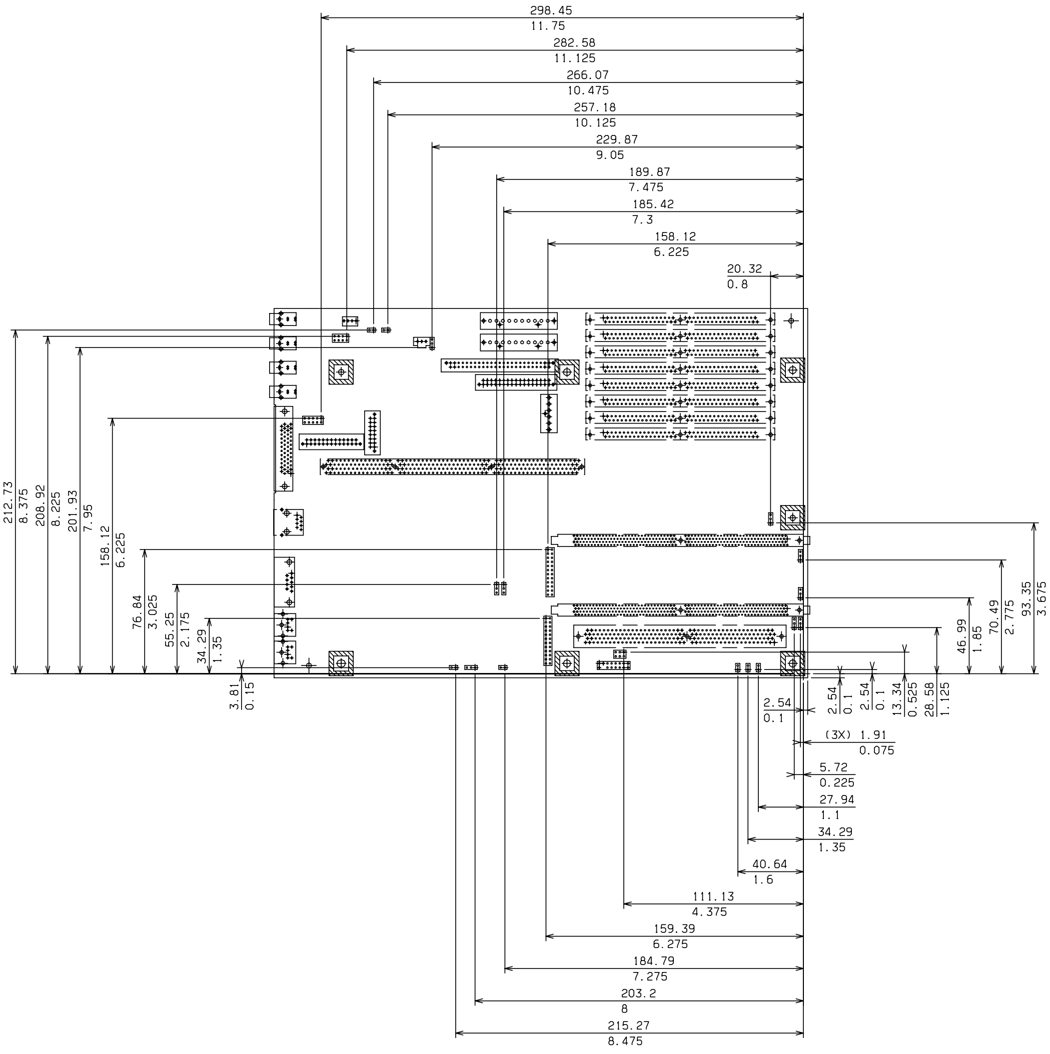


16.2.2 Motherboard Mounting Holes

See Section 16.2.5 and Section 16.2.6 for details.

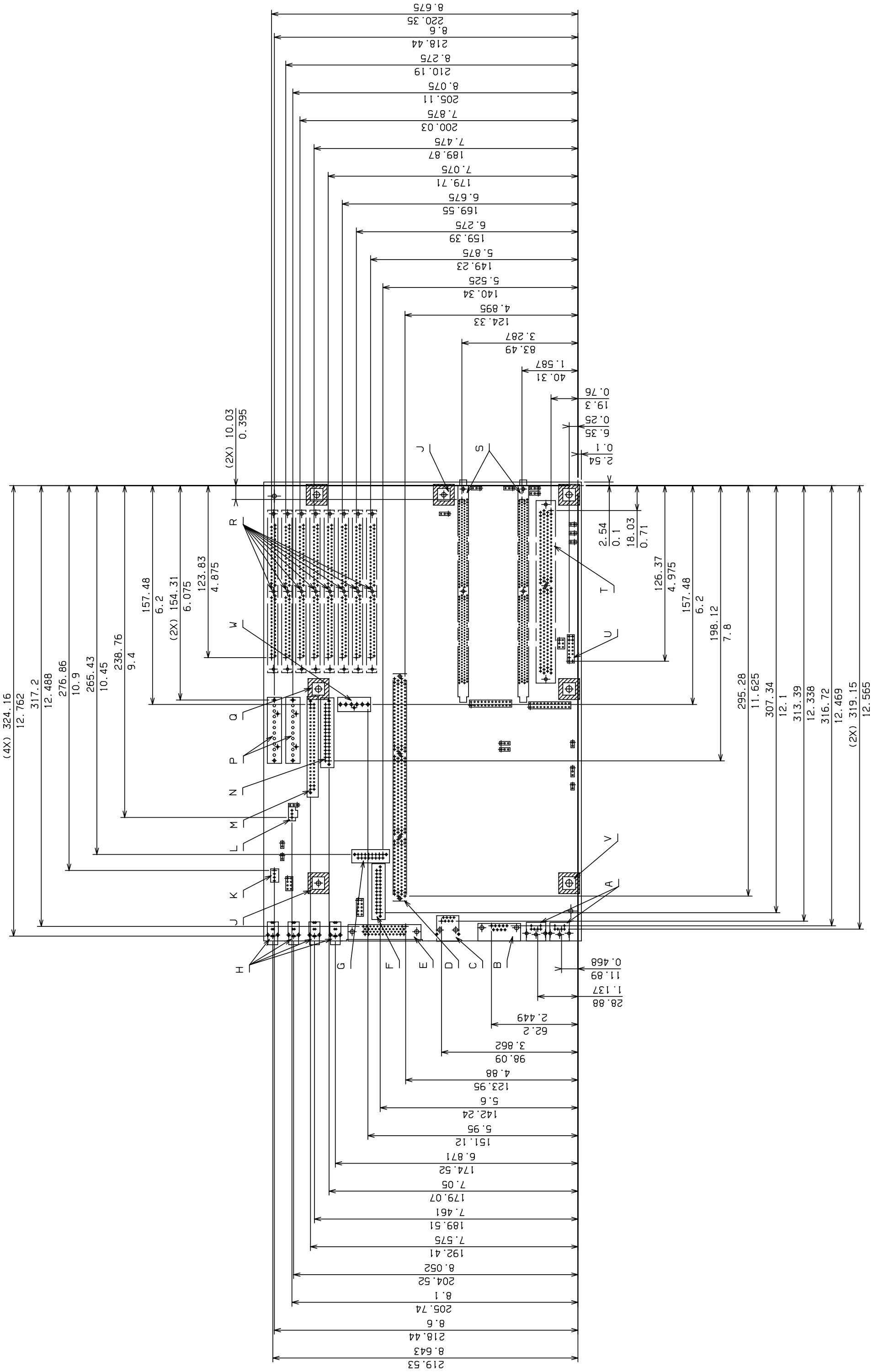


16.2.3 Motherboard Header and Jumper Locations

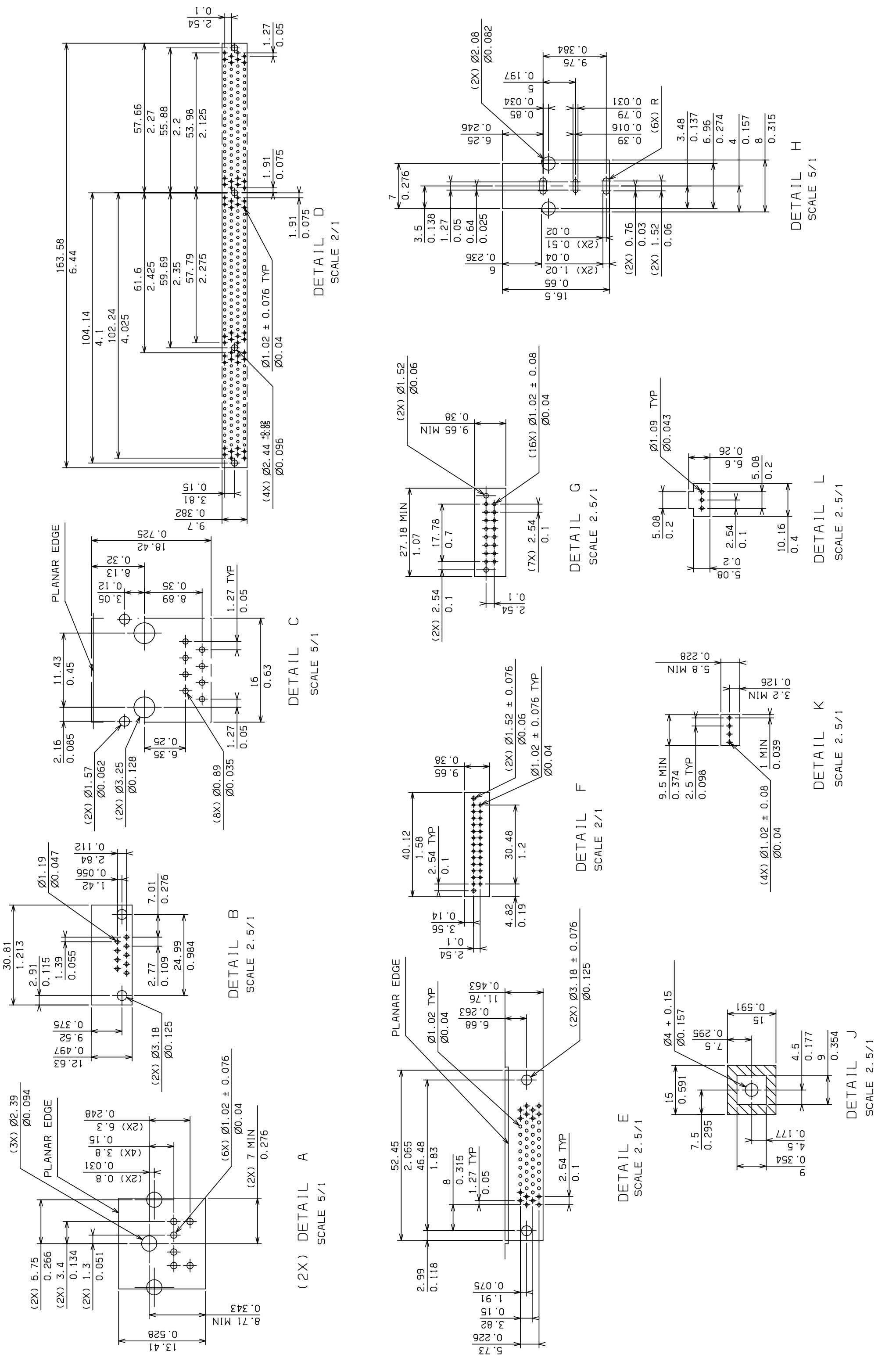


16.2.4 Motherboard Connector Locations

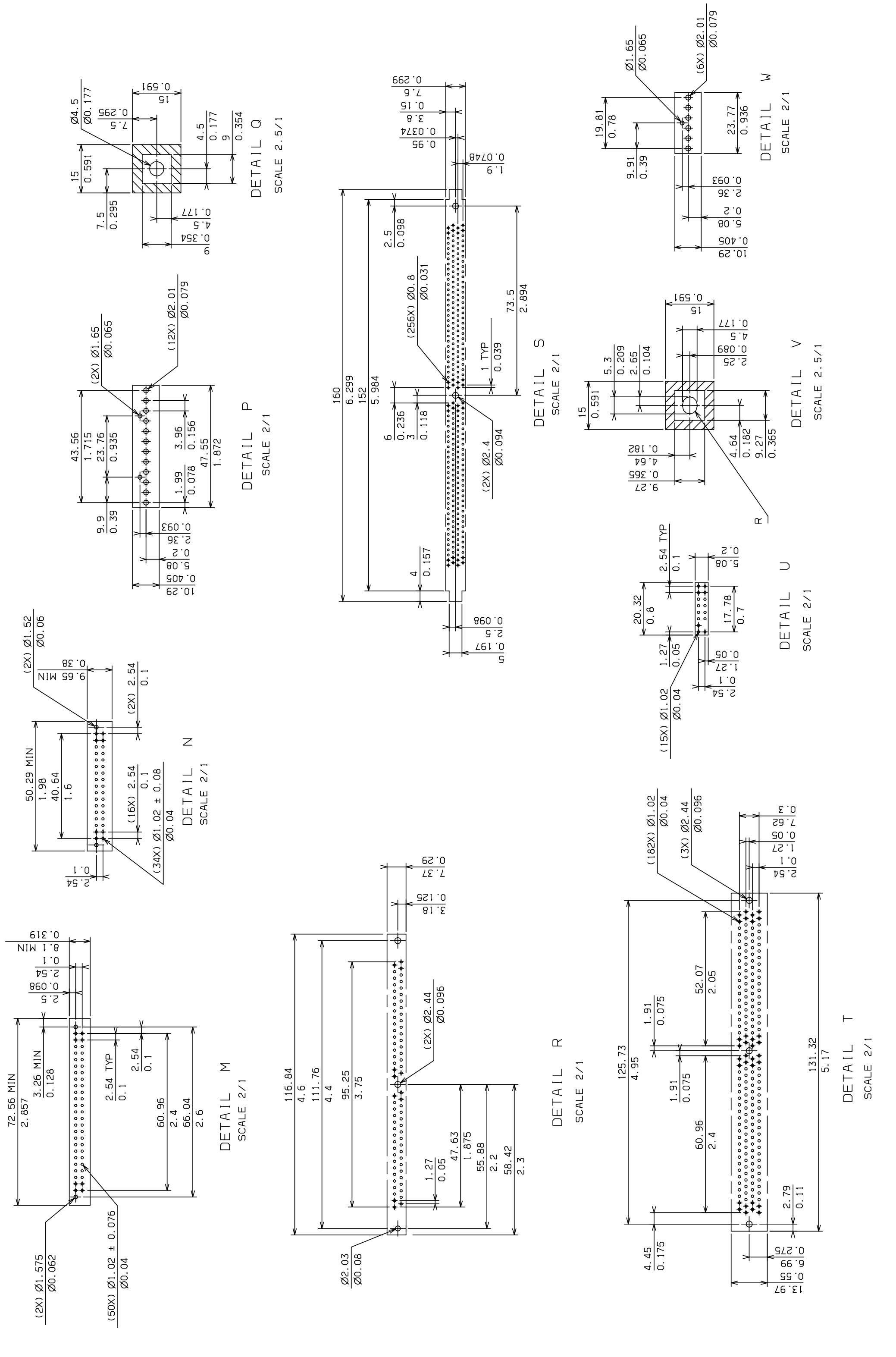
See Section 16.2.5 and Section 16.2.6 for details.



16.2.5 Motherboard Connector Details – Sheet 1 of 2

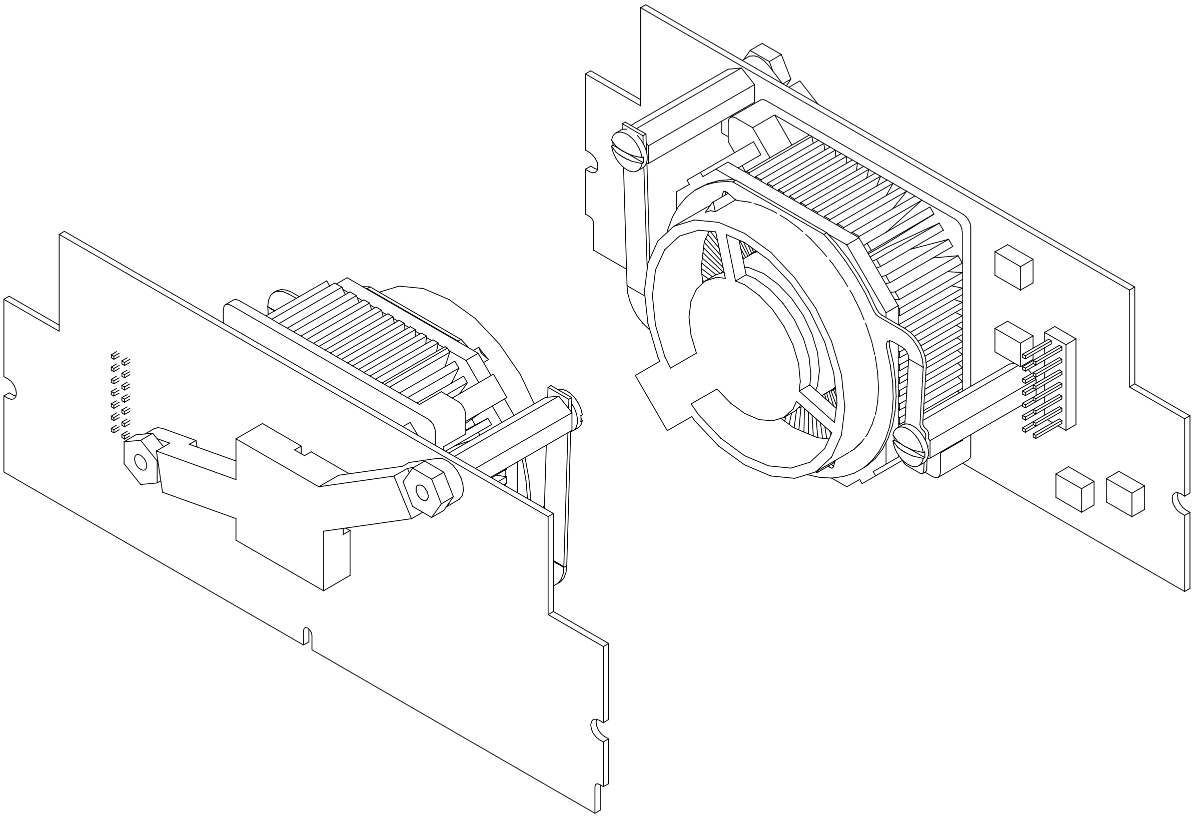


16.2.6 Motherboard Connector Details – Sheet 2 of 2

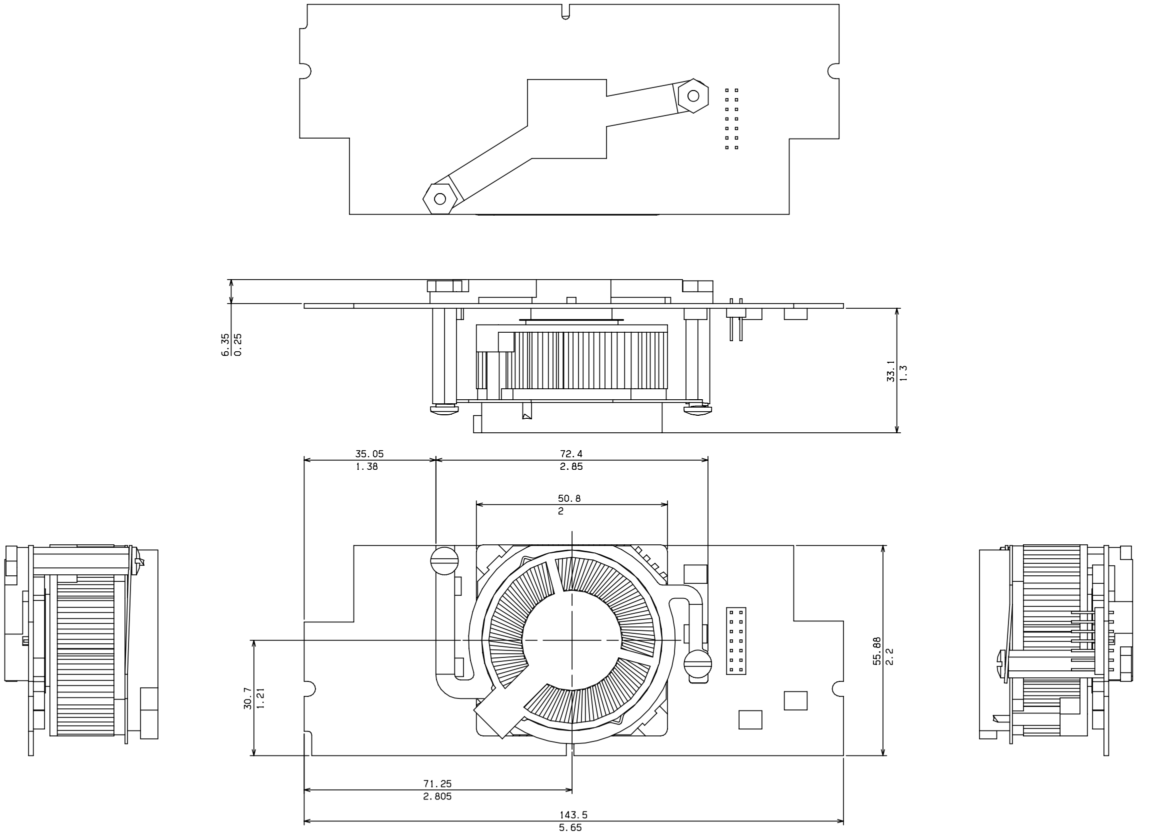


16.3 CPU Card

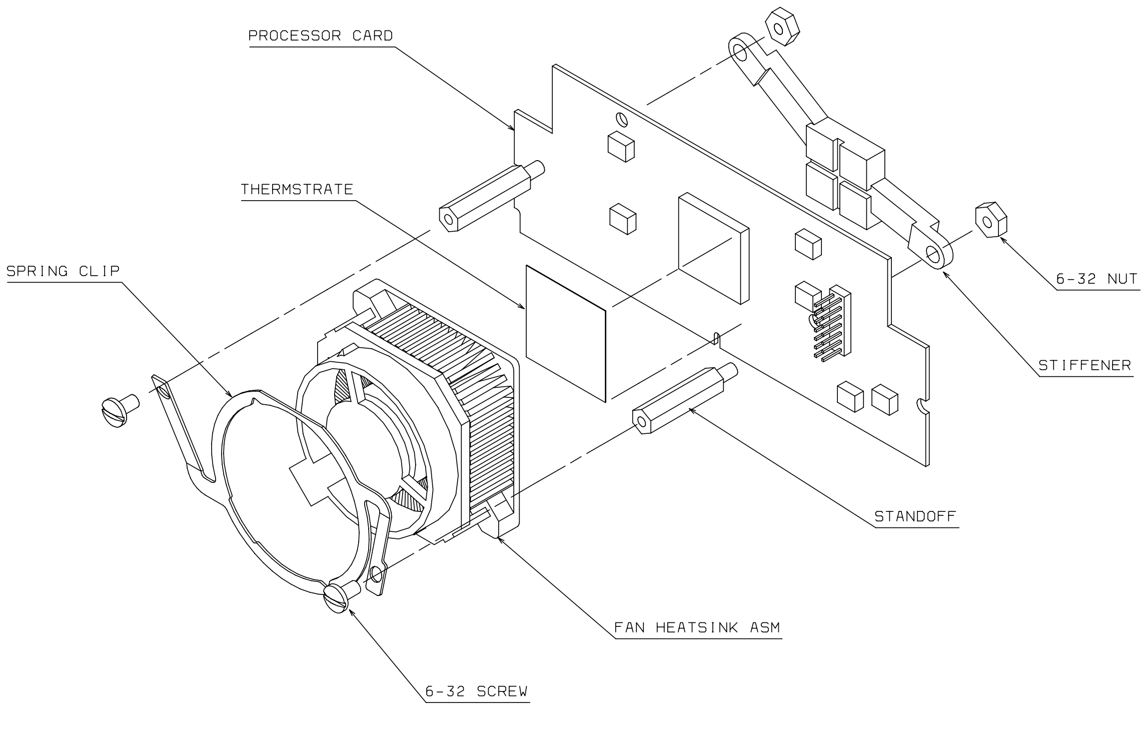
16.3.1 CPU Card 3D Views



16.3.2 CPU Card Layout



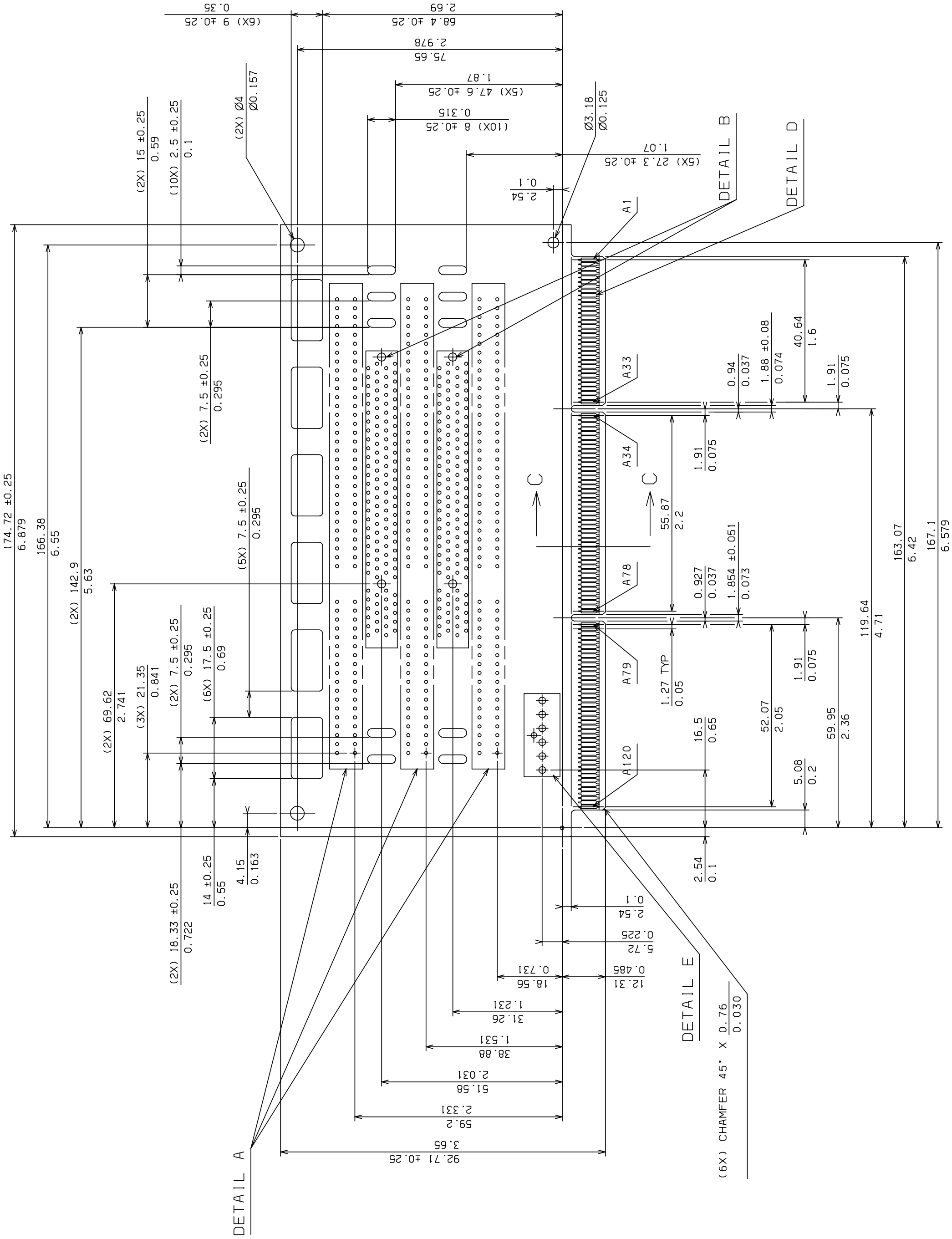
16.3.3 CPU Card Fansink Assembly



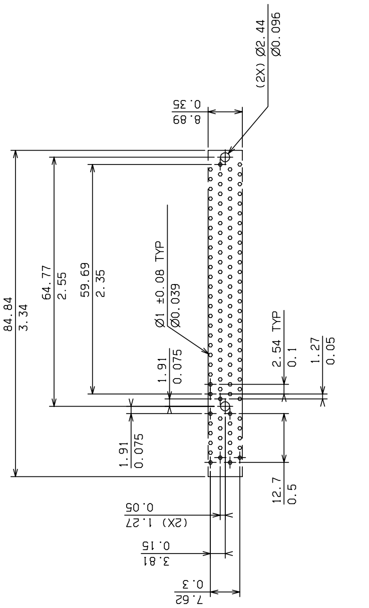
16.4 Riser

16.4.1 Riser Mechanical

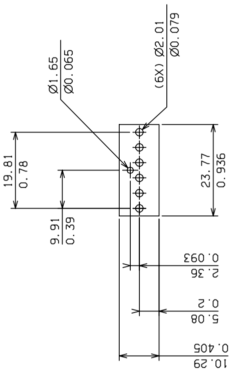
This is a PCI side view. The PCI connectors and the power connector are mounted on this side. The ISA connectors mount to the reverse side. See Section 16.4.2 for details.



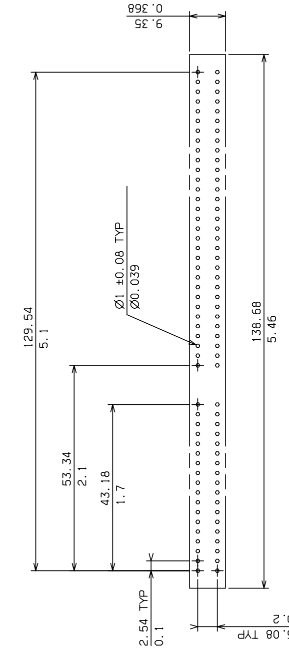
16.4.2 Riser Details



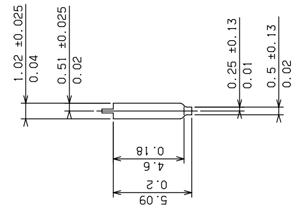
(2X) DETAIL B (PCI)
SCALE: 2.5/1



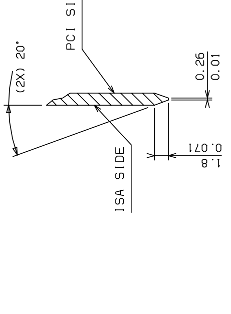
DETAIL E
SCALE 2.5/1



(3X) DETAIL A (ISA)
SCALE: 2.5/1



DETAIL D
SCALE: 10/1



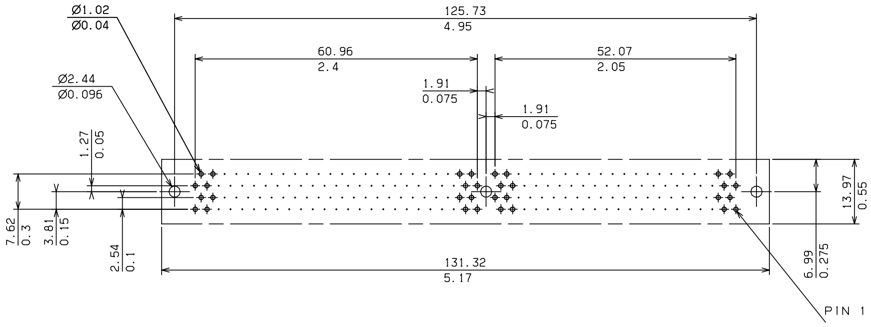
SECTION C-C
SCALE: 5/1

.

16.5 L2 Card

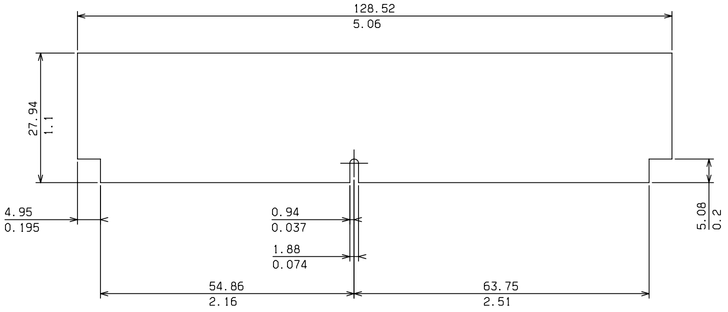
16.5.1 Motherboard L2 Card Connector for 5v Tolerant Systems

This is the footprint of the L2 tag/SRAM card connector (mounted on the motherboard). It is suggested for installation on motherboards that can tolerate L2 cards having 5v outputs. The reference design is supplied with this connector.



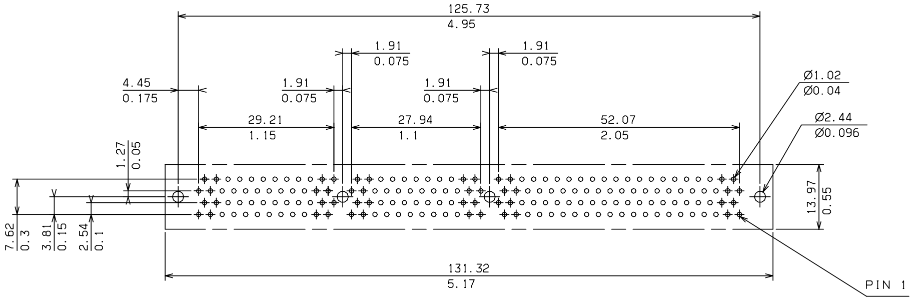
16.5.2 Outline Drawing of 5v Output L2 Card

This is an outline drawing of an L2 tag/SRAM card that has 5v outputs. This card will only mount in the 5v tolerant motherboard connector. It will not mount in the 3.3v only motherboard connector.



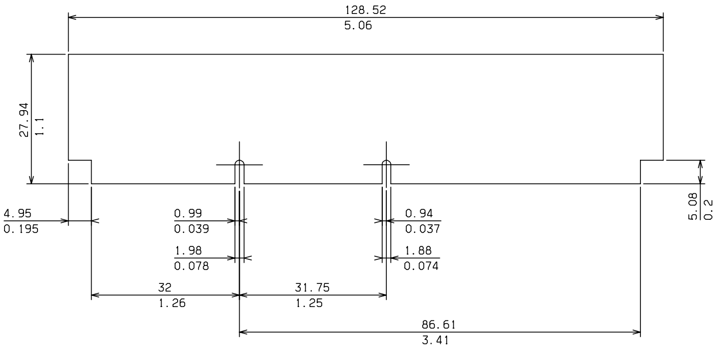
16.5.3 Motherboard L2 Card Connector for 3.3v Only Systems

This is the footprint of the L2 tag/SRAM card connector (mounted on the motherboard) It is suggested for installation on motherboards that can **not** tolerate L2 cards that have 5v outputs. The reference design does not use this connector.



16.5.4 Outline Drawing of 3.3v Output L2 Card

This is an outline drawing of an L2 tag/SRAM card that uses 3.3v outputs. This card can mount in either motherboard connector.



Section 17 Connectors

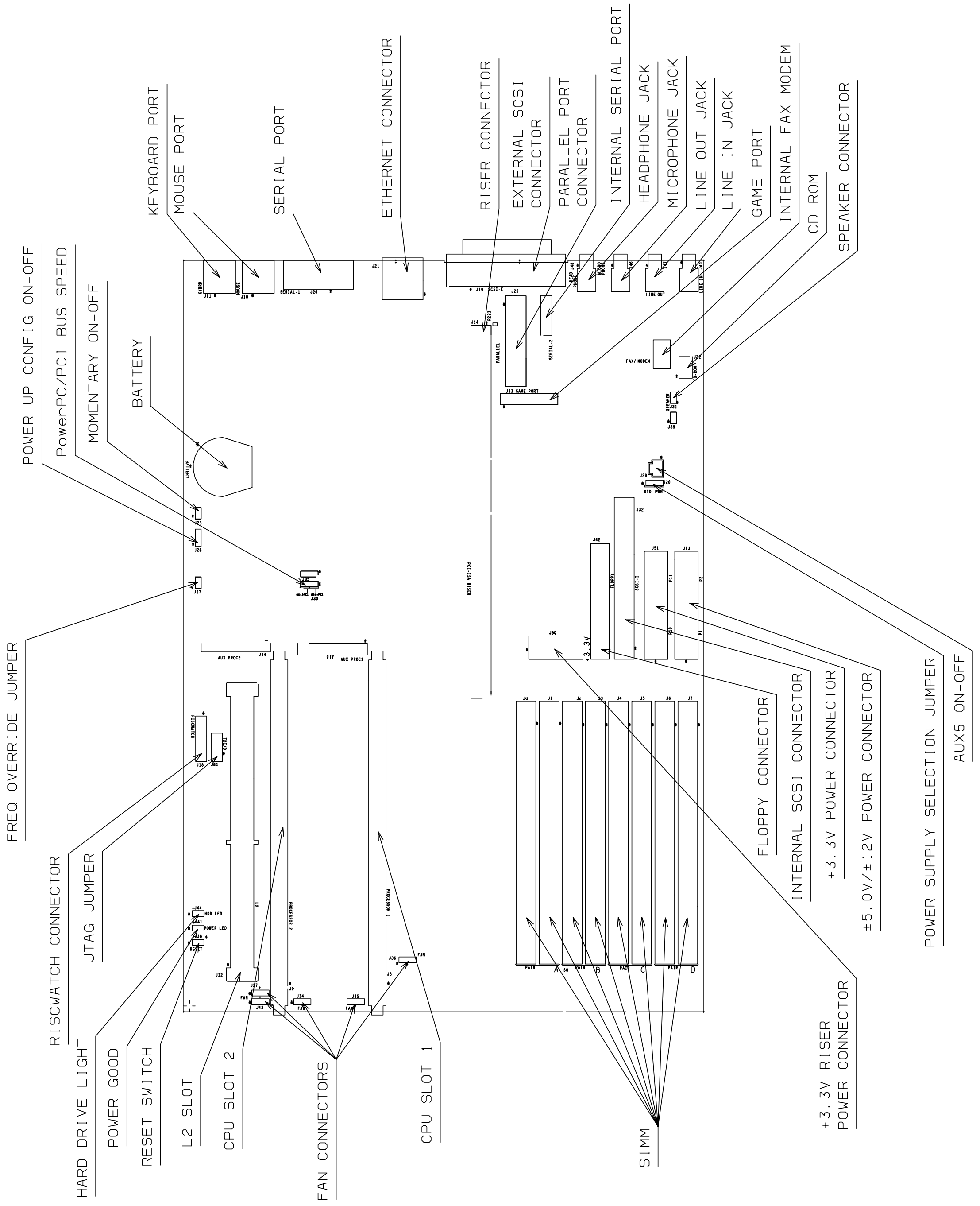
This section contains information on the connectors used by the reference design.

The ISA and PCI bus connectors are mounted to the riser board. See Section 14.

The RISCWatch connector and configuration jumpers are described in Section 2.9.

The physical dimensions of the connectors are shown in Section 16.

17.1 Connector Locations

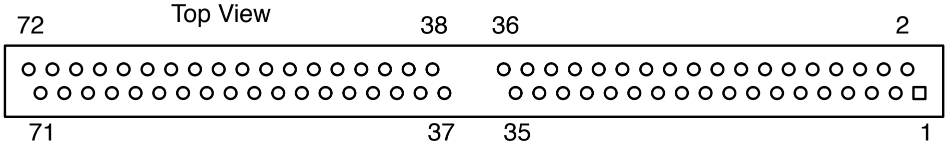


17.2 Connectors

17.2.1 Battery Connector BATTERY

The battery type is CR2032 3 Volt. Insert the battery with + side up.

17.2.2 DRAM SIMM Connectors J0, J1, J2, J3, J4, J5, J6, J7

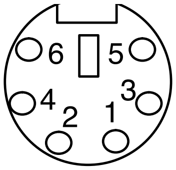


SIMM Pin	Motherboard Pin	Signal	SIMM Pin	Motherboard Pin	Signal
1	1	GROUND	37	37	DQ17
2	2	DQ0	38	38	DQ35
3	3	DQ18	39	39	GROUND
4	4	DQ1	40	40	CAS0#
5	5	DQ19	41	41	CAS2#
6	6	DQ2	42	42	CAS3#
7	7	DQ20	43	43	CAS1#
8	8	DQ3	44	44	RAS0#
9	9	DQ21	45	45	RAS1#
10	10	+5 V	46	46	BS1
11	11	CASP	47	47	WE#
12	12	A0	48	48	RESD1
13	13	A1	49	49	DQ9
14	14	A2	50	50	DQ27
15	15	A3	51	51	DQ10
16	16	A4	52	52	DQ28
17	17	A5	53	53	DQ11
18	18	A6	54	54	DQ29
19	19	A10	55	55	DQ12
20	20	DQ4	56	56	DQ30
21	21	DQ22	57	57	DQ13
22	22	DQ5	58	58	DQ31
23	23	DQ23	59	59	+5 V
24	24	DQ6	60	60	DQ32
25	25	DQ24	61	61	DQ14
26	26	DQ7	62	62	DQ33
27	27	DQ25	63	63	DQ15
28	28	A7	64	64	DQ34
29	29	BS0/A11	65	65	DQ16
30	30	+5V	66	66	BS2
31	31	A8	67	67	PD1
32	32	A9	68	68	PD2
33	33	RAS3#	69	69	PD3
34	34	RAS2#	70	70	PD4
35	35	DQ26	71	71	BS3
36	36	DQ8	72	72	GROUND

17.2.3 Mouse Connector J10

The mouse connector uses a 6-pin miniature DIN connector.

View from rear of box

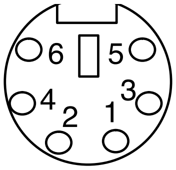


Pin	I/O	SIGNAL NAME
1	I/O	Data
2	NA	Reserved
3	NA	Ground
4	NA	+ 5v
5	I/O	Clock
6	NA	Reserved

17.2.4 Keyboard Connector J11

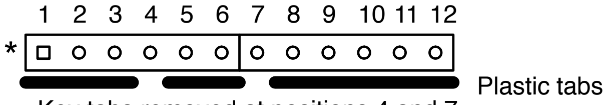
The keyboard connector uses a 6-pin miniature DIN connector.

View from rear of box



Pin	I/O	SIGNAL NAME
1	I/O	Data
2	NA	Reserved
3	NA	Ground
4	NA	+ 5v
5	I/O	Clock
6	NA	Reserved

17.2.5 Power Connector J13

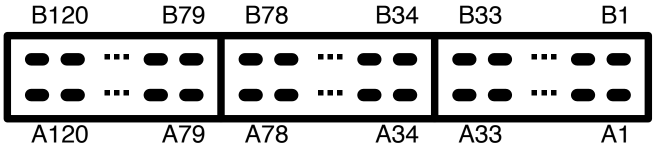


Key tabs removed at positions 4 and 7.

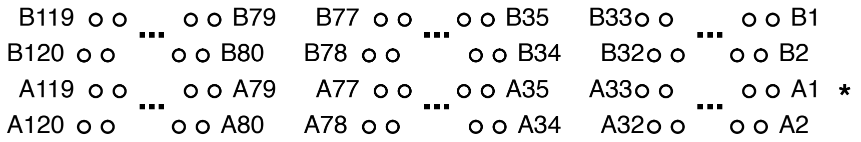
Pin	Signal Name
1	POWER GOOD
2	+5v
3	+12v
4	–12v
5	Ground
6	Ground
7	Ground
8	Ground
9	–5v
10	+5v
11	+5v
12	+5v

17.2.6 Riser Connector J14

This is the motherboard edge card connector that the riser card plugs into.



Edge card pinout



Motherboard hole pattern

Table 102. J14 Riser Connector Pin Assignments

Pin	Function	Pin	Function
A1	ISA IO CHCK#	B1	GROUND
A2	–5 VOLTS	B2	–12 VOLTS
A3	SD(7)	B3	ISA RESET
A4	SD(6)	B4	–12 VOLTS
A5	SD(5)	B5	ISA IRQ9
A6	SD(4)	B6	+12 VOLTS
A7	SD(3)	B7	ISA DRQ(2)
A8	SD(2)	B8	+12 VOLTS
A9	SD(1)	B9	ISA ZERO WS#
A10	SD(0)	B10	+12 VOLTS
A11	GROUND	B11	+5 VOLTS
A12	ISA IO CHRDY	B12	GROUND
A13	ISA AEN	B13	ISA SMEMW#
A14	ISA SA(19)	B14	ISA SMEMR#
A15	ISA SA(18)	B15	ISA IOW#
A16	ISA SA(17)	B16	ISA IOR#
A17	ISA SA(16)	B17	ISA DACK(3)#
A18	ISA SA(15)	B18	ISA DRQ(3)
A19	ISA SA(14)	B19	ISA DACK(1)#
A20	ISA SA(13)	B20	ISA DRQ(1)
A21	ISA SA(12)	B21	ISA REFRESH#
A22	GROUND	B22	GROUND
A23	PCI TRST# (res)	B23	+5 VOLTS
A24	+5 VOLTS	B24	ISA TCK (res)
A25	PCI TMS (res)	B25	GROUND
A26	PCI TDI (res)	B26	PCI TDO (res)
A27	+5 VOLTS	B27	GROUND
A28	PCI_IRQ9#	B28	+5 VOLTS

Table 102. J14 Riser Connector Pin Assignments (Continued)

Pin	Function	Pin	Function
A29	RESERVED	B29	PCI_IRQ11#
A30	+5 VOLTS	B30	RESERVED
A31	RESERVED	B31	PCI PRESENT 1#
A32	+5 VOLTS	B32	GROUND
A33	PCI_GNT0#(res)	B33	PCI PRESENT 2#
A34	PCI_REQ0#(res)	B34	GROUND
A35	+5 VOLTS	B35	CLOCK SLOT 1
A36	PCI RESET 1#	B36	GROUND
A37	+5 VOLTS	B37	CLOCK SLOT 2
A38	PCI RESET 2#	B38	GROUND
A39	PCI GNT 1#	B39	GROUND
A40	GROUND	B40	PCI REQ 1#
A41	PCI GNT 2#	B41	PCI REQ 2#
A42	RESERVED	B42	+5 VOLTS
A43	+5 VOLTS	B43	GROUND
A44	PCI IDSEL 2	B44	RESERVED
A45	GROUND	B45	+5 VOLTS
A46	PCI A/D(30)	B46	PCI A/D(31)
A47	+5 VOLTS	B47	PCI A/D(29)
A48	PCI A/D(28)	B48	GROUND
A49	PCI A/D(26)	B49	PCI A/D(27)
A50	GROUND	B50	PCI A/D(25)
A51	PCI A/D(24)	B51	+5 VOLTS
A52	PCI IDSEL 1	B52	C/BE#(3)
A53	+5 VOLTS	B53	PCI A/D(23)
A54	PCI A/D(22)	B54	GROUND
A55	PCI A/D(20)	B55	PCI A/D(21)
A56	GROUND	B56	PCI A/D(19)
A57	PCI A/D(18)	B57	+5 VOLTS
A58	PCI A/D(16)	B58	PCI A/D(17)
A59	+5 VOLTS	B59	C/BE#(2)
A60	PCI FRAME#	B60	GROUND
A61	GROUND	B61	PCI IRDY#
A62	PCI TRDY#	B62	+5 VOLTS
A63	GROUND	B63	PCI DEVSEL#
A64	PCI STOP#	B64	GROUND
A65	+5 VOLTS	B65	PCI LOCK#
A66	PCI SDONE (res)	B66	PCI PERR#
A67	PCI SBO# (res)	B67	+5 VOLTS
A68	GROUND	B68	PCI SERR#
A69	PCI PAR	B69	+5 VOLTS
A70	PCI A/D(15)	B70	C/BE#(1)
A71	+5 VOLTS	B71	PCI A/D(14)

Table 102. J14 Riser Connector Pin Assignments (Continued)

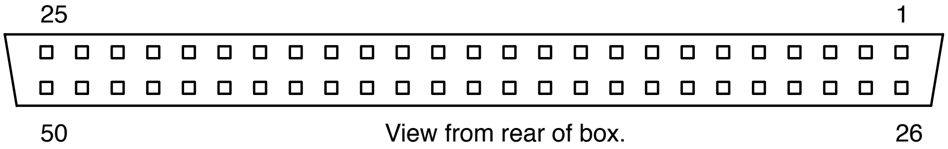
Pin	Function	Pin	Function
A72	PCI A/D(13)	B72	GROUND
A73	PCI A/D(11)	B73	PCI A/D(12)
A74	GROUND	B74	PCI A/D(10)
A75	PCI A/D(9)	B75	GROUND
A76	C/BE#(0)	B76	PCI A/D(8)
A77	+5 VOLTS	B77	PCI A/D(7)
A78	PCI A/D(6)	B78	+5 VOLTS
A79	PCI A/D(4)	B79	PCI A/D(5)
A80	GROUND	B80	PCI A/D(3)
A81	PCI A/D(2)	B81	GROUND
A82	PCI A/D(0)	B82	PCI A/D(1)
A83	+5 VOLTS	B83	+5 VOLTS
A84	PCI REQ64#	B84	PCI ACK64#
A85	+5 VOLTS	B85	+5 VOLTS
A86	+5 VOLTS	B86	+5 VOLTS
A87	GROUND	B87	GROUND
A88	ISA SA(11)	B88	GROUND
A89	+5 VOLTS	B89	ISA SYSCLK
A90	ISA SA(10)	B90	GROUND
A91	ISA SA(9)	B91	ISA_IRQ7
A92	ISA SA(8)	B92	ISA_IRQ6
A93	ISA SA(7)	B93	ISA_IRQ5
A94	ISA SA(6)	B94	ISA_IRQ4
A95	ISA SA(5)	B95	ISA_IRQ3
A96	ISA SA(4)	B96	ISA DACK(2)#
A97	ISA SA(3)	B97	ISA TC
A98	ISA SA(2)	B98	ISA BALE
A99	ISA SA(1)	B99	RESERVED
A100	ISA SA(0)	B100	GROUND
A101	GROUND	B101	ISA OSCILLATOR
A102	RESERVED	B102	+5 VOLTS
A103	ISA SBHE#	B103	ISA MEMCS16#
A104	LA(23)	B104	ISA IOCS16#
A105	LA(22)	B105	ISA_IRQ10
A106	LA(21)	B106	ISA_IRQ11
A107	LA(20)	B107	ISA_IRQ12
A108	LA(19)	B108	ISA_IRQ15
A109	LA(18)	B109	ISA_IRQ14
A110	LA(17)	B110	ISA DACK(0)#
A111	ISA MEMR#	B111	ISA DRQ(0)
A112	ISA MEMW#	B112	ISA DACK(5)#
A113	SD(8)	B113	ISA DRQ(5)
A114	SD(9)	B114	ISA DACK(6)#

Table 102. J14 Riser Connector Pin Assignments (Continued)

Pin	Function	Pin	Function
A115	SD(10)	B115	ISA DRQ(6)
A116	SD(11)	B116	ISA DACK(7)#
A117	SD(12)	B117	ISA DRQ(7)
A118	SD(13)	B118	+5 VOLTS
A119	SD(14)	B119	ISA MASTER#
A120	SD(15)	B120	GROUND

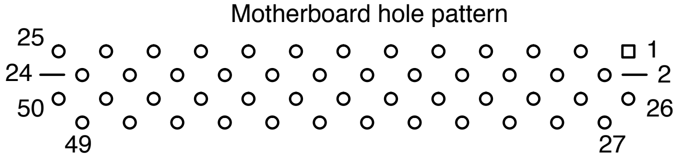
17.2.7 External SCSI Device Connector J19

J19 is a SCSI–2, 50 position socket connector.



View from rear of box.

Motherboard hole pattern

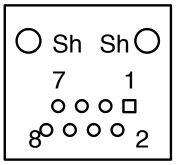


Pin	Function	Pin	Function
1	GROUND	26	DATA 0
2	GROUND	27	DATA 1
3	GROUND	28	DATA 2
4	GROUND	29	DATA 3
5	GROUND	30	DATA 4
6	GROUND	31	DATA 5
7	GROUND	32	DATA 6
8	GROUND	33	DATA 7
9	GROUND	34	DATA P
10	GROUND	35	GROUND
11	GROUND	36	GROUND
12	RESERVED	37	RESERVED
13	NO CONNECTION	38	+5 VOLT
14	RESERVED	39	RESERVED
15	GROUND	40	GROUND
16	GROUND	41	ATTENTION#
17	GROUND	42	GROUND
18	GROUND	43	BUSY#
19	GROUND	44	ACKNOWLEDGE#
20	GROUND	45	RESET#
21	GROUND	46	MESSAGE#
22	GROUND	47	SELECT#
23	GROUND	48	CONTROL/DATA#
24	GROUND	49	REQUEST#
25	GROUND	50	INPUT/OUTPUT#

17.2.8 Ethernet 10BASE-T Connector J21

The motherboard is equipped with an Ethernet Local Area Network communication port.

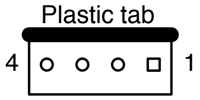
J21 is an 8 pin RJ-45 jack.



Pin	Signal
1	+ TRANSMIT DATA
2	– TRANSMIT DATA
3	+ RECEIVE DATA
4	NO CONNECTION
5	NO CONNECTION
6	– RECEIVE DATA
7	NO CONNECTION
8	NO CONNECTION
Sh	Shield Ground

17.2.9 CD-ROM Connector J22

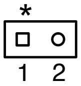
J22 is a 4 position header connector.



Pin	Signal Name
1	LEFT CHANNEL
2	GROUND
3	GROUND
4	RIGHT CHANNEL

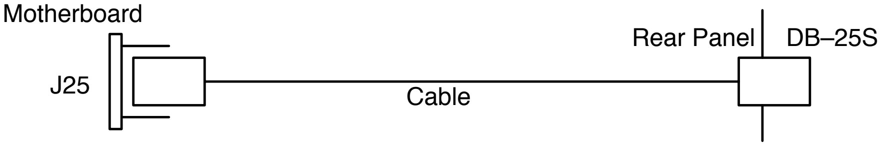
17.2.10 Momentary On–Off Power Switch Connector J23 (Not Used)

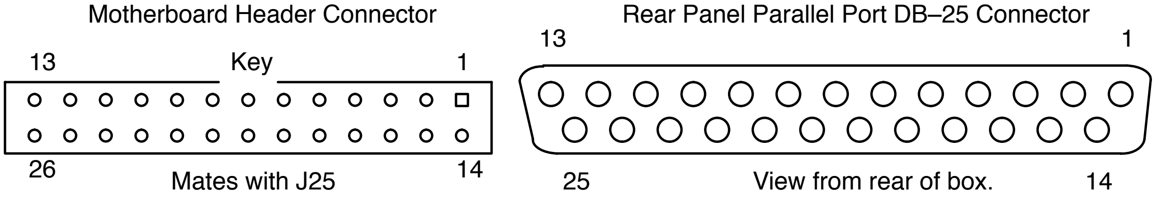
J23 is a 1x2 Berg type connector.



Pin	Signal Name	
1	N.O.	(AUX5v)
2	Common	(Input)

17.2.11 Parallel Connector J25



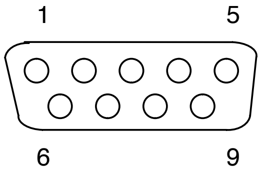

Motherboard Header Connector
13
Key
1
26
Mates with J25
14
Rear Panel Parallel Port DB–25 Connector
13
1
25
View from rear of box.
14

J25 Pin	DB-25S Pin	I/O	Signal Name	J25 Pin	DB-25S Pin	I/O	Signal Name
1	1	O	STROBE#	14	14	O	AUTO FD XT#
2	2	I/O	DATA 0	15	15	I	ERROR#
3	3	I/O	DATA 1	16	16	O	INIT#
4	4	I/O	DATA 2	17	17	O	SLCT IN#
5	5	I/O	DATA 3	18	18	NA	GROUND
6	6	I/O	DATA 4	19	19	NA	GROUND
7	7	I/O	DATA 5	20	20	NA	GROUND
8	8	I/O	DATA 6	21	21	NA	GROUND
9	9	I/O	DATA 7	22	22	NA	GROUND
10	10	I	ACK#	23	23	NA	GROUND
11	11	I	BUSY	24	24	NA	GROUND
12	12	I	PE	25	25	I/O	GROUND
13	13	I	SLCT	26	26	#	No Connection

Note: The external parallel port may be located in an unused ISA slot or on the bulkhead of the enclosure. A 26-pin header and ribbon cable are used to connect the motherboard to the external connector.

17.2.12 Serial Port 1 (External) Connector J26

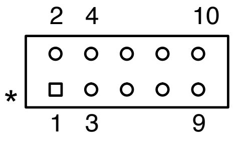
J27 is a DE–9P connector.



Pin	Signal Name
1	DATA CARRIER DETECT
2	RECEIVE DATA
3	TRANSMIT DATA
4	DATA TERMINAL READY
5	SIGNAL GROUND
6	DATA SET READY
7	REQUEST TO SEND
8	CLEAR TO SEND
9	RING INDICATOR

17.2.13 Serial Port 2 (Internal) Connector J27

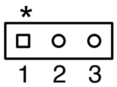
J27 is a 2x5 Berg type header connector.



Pin	Signal Name
1	DATA CARRIER DETECT
2	RECEIVE DATA
3	TRANSMIT DATA
4	DATA TERMINAL READY
5	SIGNAL GROUND
6	DATA SET READY
7	REQUEST TO SEND
8	CLEAR TO SEND
9	RING INDICATOR
10	No Connection

17.2.14 Power Up Configuration Jumper J28

J28 is a 1x3 Berg type connector.

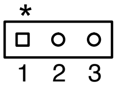


Install 1–2 To configure the power management controller to bring up the system when DC power is applied to the motherboard.

Install 2–3 To configure the power management controller **not** to bring up the system immediately when DC power is applied to the motherboard, but to wait for the reset button to be pressed.

17.2.15 AUX5/ON-OFF Connector J29

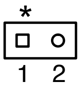
J29 is a 1x3 Berg type connector.



Pin	Signal Name
1	AUX5v
2	On/Off
3	Ground

17.2.16 Speaker Connector J31 and J39

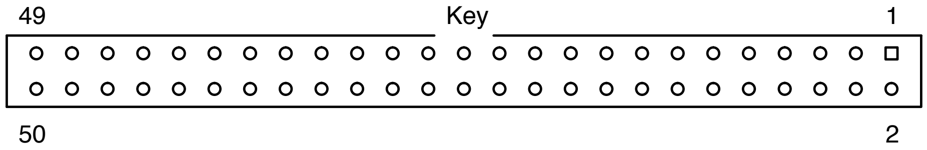
J31 and J39 are identical 1x2 Berg type connectors. J39 pins are connected to J31 pins.



Pin	Signal Name
1	Plus output to speaker
2	Minus output to speaker

17.2.17 Internal SCSI Device Connector J32

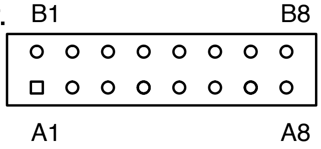
J32 is a 2x25 pin header.



Pin	Function	Pin	Function
1	GROUND	2	DATA 0
3	GROUND	4	DATA 1
5	GROUND	6	DATA 2
7	GROUND	8	DATA 3
9	GROUND	10	DATA 4
11	GROUND	12	DATA 5
13	GROUND	14	DATA 6
15	GROUND	16	DATA 7
17	GROUND	18	DATA P
19	GROUND	20	GROUND
21	GROUND	22	GROUND
23	RESERVED	24	RESERVED
25	NO CONN	26	+5 VOLT
27	RESERVED	28	RESERVED
29	GROUND	30	GROUND
31	PULL UP	32	ATTENTION#
33	GROUND	34	GROUND
35	GROUND	36	BUSY#
37	GROUND	38	ACKNOWLEDGE#
39	GROUND	40	RESET#
41	GROUND	42	MESSAGE#
43	GROUND	44	SELECT#
45	GROUND	46	CONTROL/DATA#
47	GROUND	48	REQUEST#
49	GROUND	50	INPUT/OUTPUT#

17.2.18 Game Port Connector J33

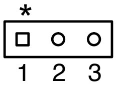
J33 is a 2x8 Berg type connector.



Pin	Signal Name
A 1	+5v
A 2	JAB1 – Joystick A Button 1
A 3	JACX – Joystick A Coordinate X
A 4	Ground
A 5	Ground
A 6	JACY – Joystick A Coordinate Y
A 7	JAB2 – Joystick A Button 2
A 8	+5v
B 1	+5v
B 2	JBB1/FSYNC – Joystick B Button 1 / Frame Sync
B 3	JBCX/SDOUT – Joystick B Coordinate X / Serial Data Out
B 4	MIDIOUT – MIDI Out Transmit Data
B 45	JBCY/SDIN – Joystick B Coordinate Y / Serial Data In
B 6	JBB2/SCLK – Joystick B Button 2 / Serial Clock
B 7	MIDIIN – MIDI In Receive Data
B 8	No Connection

17.2.19 Fansink Connectors J34, J36, J43, J45

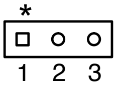
These are 1x3 Berg type connectors.



Pin	Signal Name
1	Ground
2	+12v
3	Ground

17.2.20 Box Fan Connector J37

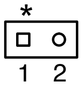
This is a 1x3 Berg type connector.



Pin	Signal Name
1	Ground
2	+12v
3	Ground

17.2.21 Reset Switch Connector J38

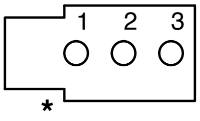
J38 is a 1x2 Berg type connector.



Pin	Signal Name
1	Reset N.O. contact (close to reset)
2	Common contact

17.2.22 Line In Jack J40

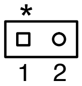
J40 is a stereo 3.5mm (subminiature) 3 conductor phone jack.



Pin	Position	Signal
1	Sleeve	Ground
2	Tip	Left Channel
3	Ring	Right Channel

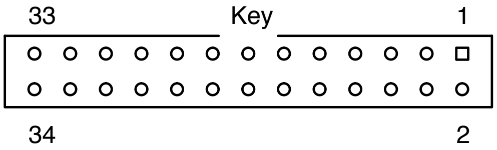
17.2.23 Power Good LED Connector J41

J41 is a 1x2 Berg type connector.



Pin	Signal Name
1	Output to LED anode
2	Output to LED cathode

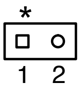
17.2.24 Floppy Diskette Drive (FDD) Connector J42



Pin	Function	Pin	Function
1	GROUND	2	DENSITY SELECT
3	GROUND	4	DENS READ 1 (REG BIT 2)
5	GROUND	6	DATA RATE 0
7	GROUND	8	INDEX#
9	DENS READ 2 (REG BIT 3)	10	MOTOR SELECT 0#
11	GROUND	12	DRIVE SELECT 1#
13	GROUND	14	DRIVE SELECT 0#
15	GROUND	16	MOTOR SELECT 1#
17	DENS READ 3 (REG BIT 0)	18	DIRECTION#
19	GROUND	20	STEP#
21	GROUND	22	WRITE DATA#
23	GROUND	24	WRITE GATE#
25	GROUND	26	TRACK 0#
27	DENS READ 4 (REG BIT 1)	28	WRITE PROTECT#
29	GROUND	30	READ DATA#
31	GROUND	32	HEAD SELECT#
33	GROUND	34	DISK CHANGE#

17.2.25 HDD LED Connector J44

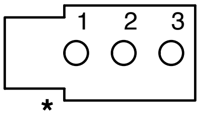
J44 is a 1x2 Berg type connector.



Pin	Signal Name
1	Output to LED anode
2	Output to LED cathode

17.2.26 Microphone Jack J46

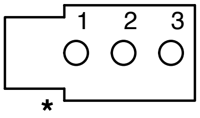
J46 is a stereo 3.5mm (subminiature) 3 conductor phone jack.



Pin	Position	Signal
1	Sleeve	Ground
2	Tip	Left Channel
3	Ring	Right Channel

17.2.27 Line Out Jack J47

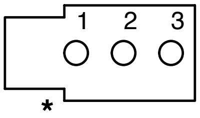
J40 is a stereo 3.5mm (subminiature) 3 conductor phone jack.



Pin	Position	Signal
1	Sleeve	Ground
2	Tip	Left Channel
3	Ring	Right Channel

17.2.28 Headphone Jack J48

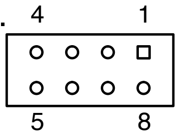
J48 is a stereo 3.5mm (subminiature) 3 conductor phone jack.



Pin	Position	Signal
1	Sleeve	Ground
2	Tip	Left Channel
3	Ring	Right Channel

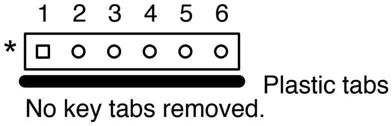
17.2.29 Internal/FaxModem Connector J49

J49 is a 2x4 Berg type connector.



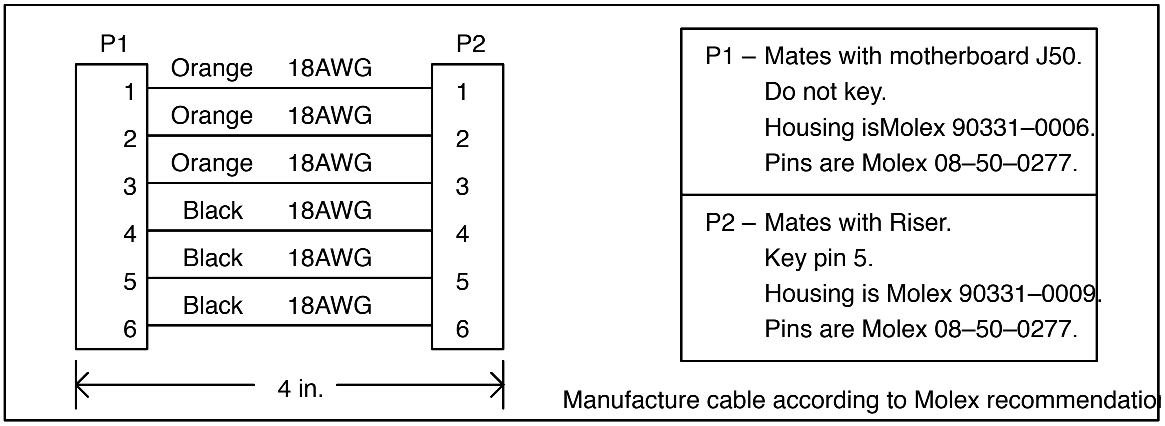
Pin	Signal Name
1	No Connection
2	L+R MICROPHONE OUT (AMPLIFIED SUM OF L AND R MIC)
3	RIGHT CHANNEL OUT
4	LEFT CHANNEL OUT
5	LEFT CHANNEL IN (AUX 2)
6	GROUND
7	GROUND
8	RIGHT CHANNEL IN (AUX 2)

17.2.30 Riser 3.3v Power Connector J50



Pin	Signal Name
1	+3.3v
2	+3.3v
3	+3.3v
4	Ground
5	Ground
6	Ground

17.2.31 Riser 3.3v Power Cable



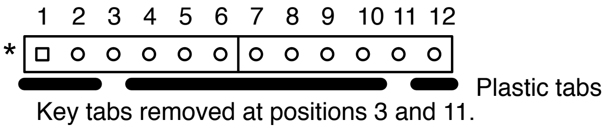
P1 – Mates with motherboard J50.
Do not key.
Housing isMolex 90331–0006.
Pins are Molex 08–50–0277.

P2 – Mates with Riser.
Key pin 5.
Housing is Molex 90331–0009.
Pins are Molex 08–50–0277.

Manufacture cable according to Molex recommendatio

Figure 71. Riser Power Cable

17.2.32 3.3v Power Connector J51



Pin	Signal Name	Pin	Signal Name
1	+3.3v	7	+3.3v
2	+3.3v	8	+3.3v
3	+3.3v	9	+3.3v
4	Ground	10	Ground
5	Ground	11	Ground
6	Ground	12	Ground

Section 18
Physical Design Guidelines

These guidelines are given to aid designers with the physical design phase of their PowerPC reference design board. The guidelines are not intended to replace good physical design practices for the signal types and frequencies discussed, but to supplement standard practice by pointing out sensitive and critical areas. Some discussion of the IBM implementation of the reference board is also included to establish the context for the wiring guidelines.

18.1 Motherboard Construction

The general construction of the PowerPC 604 SMP reference motherboard (motherboard) is shown in Figure 72. It is constructed with two high frequency signal layers in the center, two power planes, and two external general purpose signal layers.

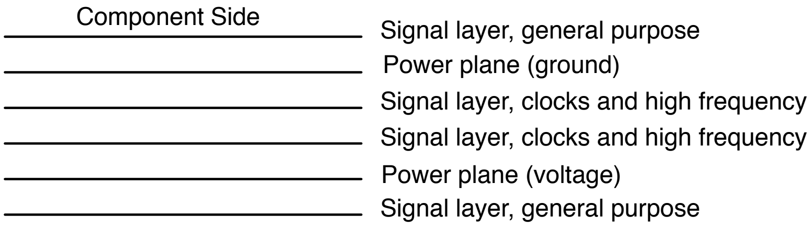

Component Side
Signal layer, general purpose
Power plane (ground)
Signal layer, clocks and high frequency
Signal layer, clocks and high frequency
Power plane (voltage)
Signal layer, general purpose

Figure 72. Motherboard Signal and Power Layers

A top view of a typical wiring channel, as implemented on the motherboard, is shown in Figure 73 (all dimensions are shown in inches). Minimum trace width is 0.004 in. (at 1:1), and minimum space width is 0.006 in. (at 1:1). For fabrication information, see Figure 74.

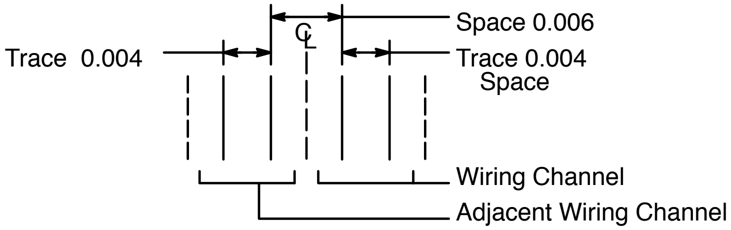
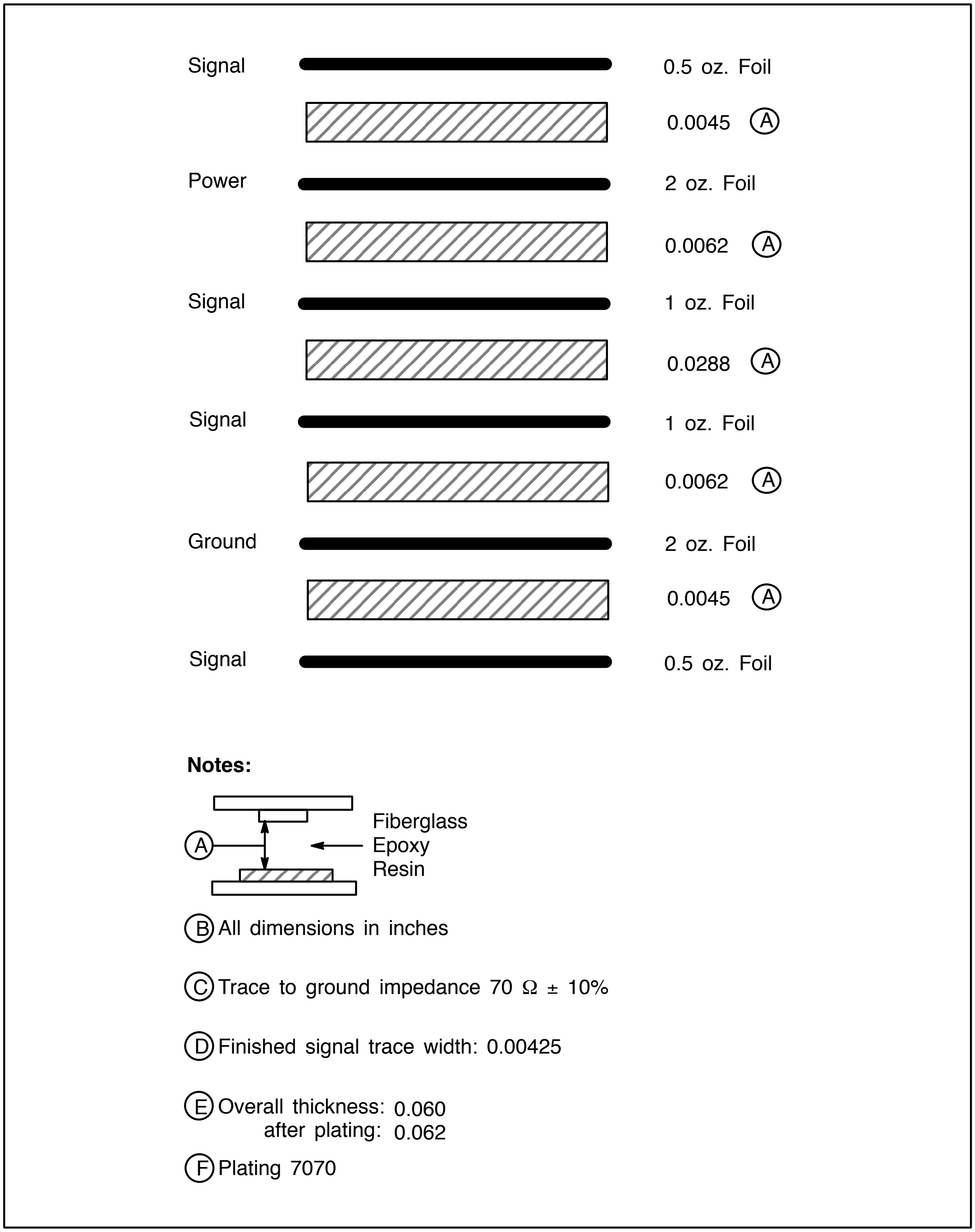


Figure 73. Typical Motherboard Wiring Channel Top View



Notes:

Ⓑ All dimensions in inches

Ⓒ Trace to ground impedance 70 Ω ± 10%

Ⓓ Finished signal trace width: 0.00425

Ⓔ Overall thickness: 0.060
after plating: 0.062

Ⓕ Plating 7070

Figure 74. Motherboard Fabrication

18.2 Riser Construction

The general construction of the PowerPC 604 SMP riser board (riser) is shown in Figure 75. It is constructed with two power planes in the center and two external signal layers.

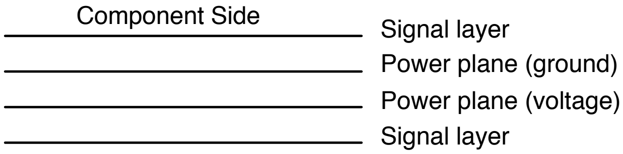


Figure 75. Riser Signal and Power Layers

A top view of a typical wiring channel, as implemented on the riser, is shown in Figure 76 (all dimensions are shown in inches). Minimum trace width is 0.006 in. (at 1:1), and minimum space width is 0.006 in. (at 1:1). For fabrication information, see Figure 77.

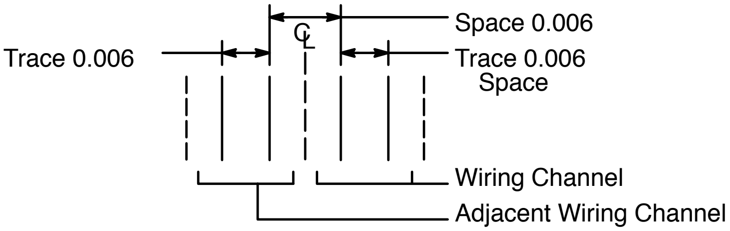
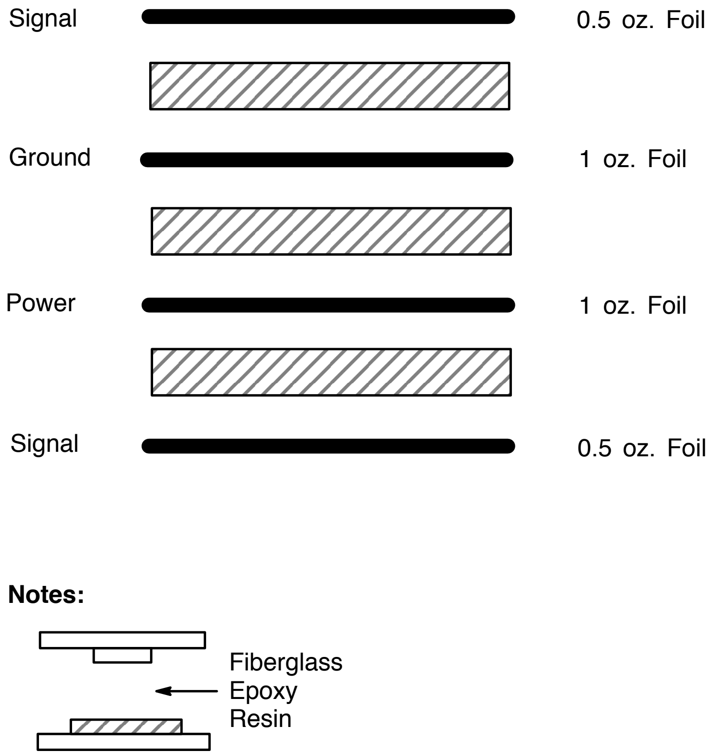


Figure 76. Typical Riser Wiring Channel Top View



Ⓐ All dimensions in inches

Ⓑ Trace to ground impedance 70 Ω ± 10%

Ⓒ Overall thickness: 0.060
after plating: 0.062

Ⓓ Plating 7070

Figure 77. Riser Fabrication

18.3 Cheetah0 Board Construction

The general construction of the PowerPC 604 SMP cheetah0 board (cheetah0) is shown in Figure 78. It is constructed with two voltage planes in the center, two signal layers, two ground planes, and two external signal layers.

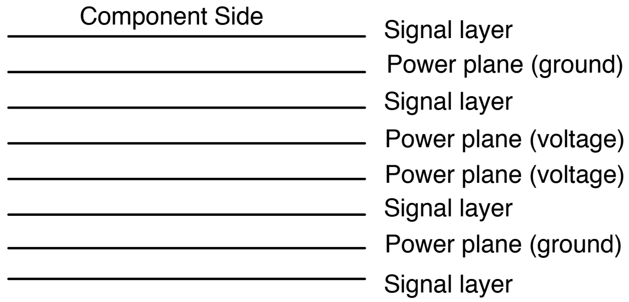


Figure 78. Signal and Power Layers

A top view of a typical wiring channel, as implemented on the cheetah0, is shown in Figure 79 (all dimensions are shown in inches). Minimum trace width is 0.006 in. (at 1:1) and minimum space width is 0.006 in. (at 1:1). For fabrication information, see Figure 80.

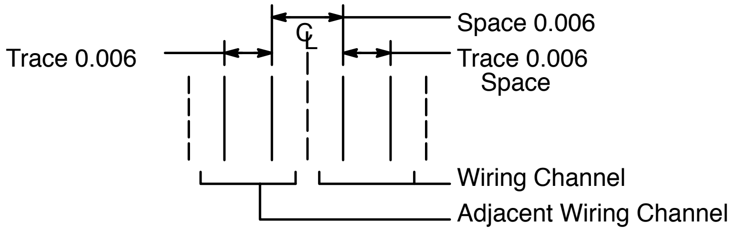
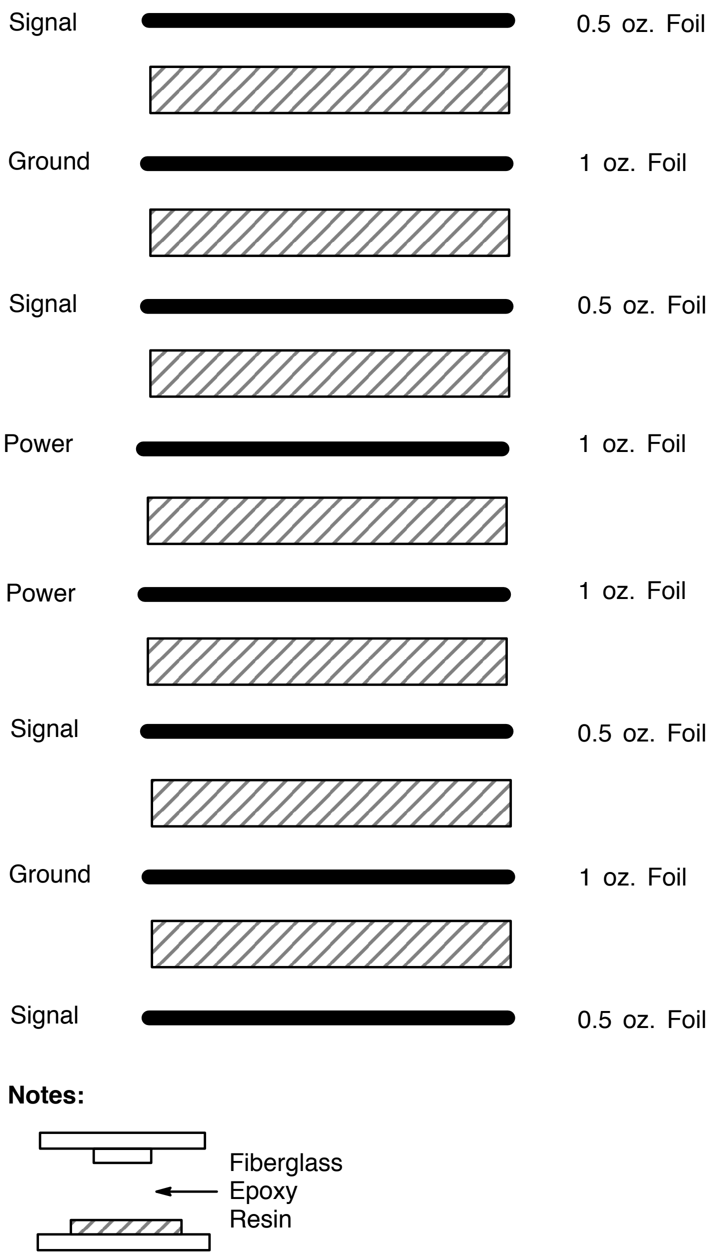


Figure 79. Typical Cheetah0 Wiring Channel Top View



Ⓐ All dimensions in inches

Ⓑ Trace to ground impedance 70 Ω ± 10%

Ⓒ Overall thickness: 0.050 ± 0.0003

Ⓓ Plating 7070

Figure 80. Cheetah0 Board Fabrication

18.4 General Wiring Guidelines

1. No wires are allowed to cross into the AUDIO sub-system unless they are connected to a device in that section.
2. All clock wires must have a minimum number of vias and be routed on internal planes.
3. If a wire is routed near a via that is part of a clock net, there must be at least one vacant wiring channel between the wire and the clock via.
4. No clock wires may be routed within one inch from the edge of the board.
5. A power (ground or voltage) plane split occurs where there is a discontinuity in the plane. As shown in Figure 81, route wires across splits in a power plane in the most perpendicular manner possible. Do not run wires parallel to the split in close proximity to the split.

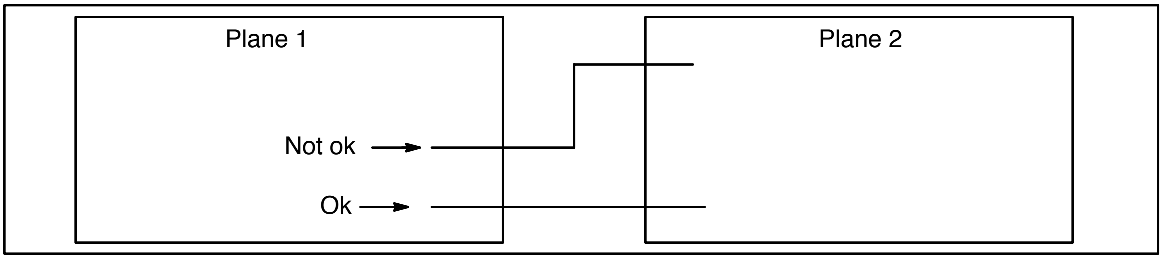


Figure 81. Power Plane Split

6. I/O wires must be routed away from densly wired non-I/O wiring areas.
7. All I/O wires must be routed such that the 22pf shunt capacitor is the last thing on the net before the I/O connector itself.
8. Minimize the number of wires that cross under the MPC970 (U26) and the number of wires that cross under the AMCC (U30) PLL on the top surface. If a wire is routed near a via attached to a clock wire, it must have a vacant wiring channel next to it.
9. Critical nets (604 bus signals) must not be placed on outer plane, if possible.
10. 604 and PCI buses (except ISA and X-buses) must be wired point to point .

Several groups of nets require special attention. They are listed in Table 103 in the order of the importance for meeting their design rules. For example, ensuring that the clock nets are routed according to their design rules is more important than ensuring that the PCI bus nets are routed according to their design rules. The ideal system design will meet all of the suggested design rules, but this information is included to guide the designer in case some tradeoffs have to be evaluated.

Table 103. Order of Importance for Meeting Design Rules

Order of Importance	Group
1	Clock Nets
2	CEC Critical Nets
3	L2 Critical Nets
4	PCI Critical Nets
5	PCI Address/Data Nets
6	Noise Sensitive Nets
7	Other Nets

18.5 Clock Wires

General requirements: See Table 104 for specific wiring length requirements.

1. Clock nets are the most critical wiring on the board. Their wiring requirements must be given priority over the requirements of other groups of signals.
2. Clock nets are to have a minimum number of vias.
3. No clock wires may be routed closer than one inch to the edge of the board.
4. Clock nets with more than two nodes (devices connected to them) are to be daisy-chained. Stubs and star fanouts are not allowed.
5. Clock nets are to be routed as much as possible on internal signal planes.
6. Where series termination resistors are required, place them as close as possible to the clock generator.
7. Route a ground trace as a shield in the adjacent wiring channel on both sides of the clock trace. Connect these shield traces to the ground plane using at least one via in each inch of trace. Completely surround the clock trace with shield traces.
8. Table 104 shows the length and tolerances of the clock nets. All dimensions are shown in inches. The Design Rule column shows the guideline, and the Reference Board column shows the actual length of the net on the reference board.
9. No more than one turning via for a total of three vias per net is allowed.

Table 104. Clock Wires

Net Name	Length
SRAM_BCLK<3:0>	As short as possible. **Define this clock length as L1** (Clock traces on SIMM must be 3.0 in. long).
MC_SRAM_BCLK0	As short as possible
MC_SRAM_BCLK1	As short as possible
MC_SRAM_BCLK2	As short as possible
MC_SRAM_BCLK3	As short as possible
TAG_BCLK	(L1 + 3.0 in.) ± 0.1 in.
MC_TAG_BCLK	As short as possible
PROC1_BCLK<2:0>	(L1 + 0.24 in.) ± 0.1 in. (Clock traces on processor card must be 2.76 in. long)
MC_PROC1_BCLK0	As short as possible
MC_PROC1_BCLK1	As short as possible
MC_PROC1_BCLK2	As short as possible
PROC2_BCLK<2:0>	(L1 + 0.24 in.) ± 0.1 in. (Clock traces on processor card must be 2.76 in. long)
MC_PROC2_BCLK0	As short as possible
MC_PROC2_BCLK1	As short as possible
MC_PROC2_BCLK2	As short as possible
663_BCLK	(L1 + 3.0 in.) ± 0.1 in.
MC_663_BCLK	As short as possible
664_BCLK	(L1 + 3.0 in.) ± 0.1 in.

Table 104. Clock Wires (Continued)

Net Name	Length
MC_664_BCLK	As short as possible
4403_BCLK	(L1 + 3.0 in.) ± 0.1 in.
MC_4403_BCLK	As short as possible
MC_BCLK_SPARE	As short as possible
PAL_BCLK	(L1 + 3.0 in.) ± 0.1 in.
XTAL1_MPC970	As short as possible
XTAL2_MPC970	As short as possible
TTL_CLK	As short as possible
4403_FBCLK	As short as possible
4403_FBCLK_R	As short as possible
X2FOUT0	As short as possible
PLL_FOUT2	As short as possible
SLOT2_PCICLK	As short as possible (If using IBM Riser card 25JP-RISER3-A, clock trace on that riser is 3.73 in. long). **Define this clock length as L2.**
PLL_FOUT3	As short as possible
SLOT1_PCICLK	(L2 + 0.9 in.) ± 0.1 in. (If using IBM Riser card 25JP-RISER3-A, clock trace on that riser is 2.81 in. long)
PLL_FOUT2A	As short as possible
SCSI_PCICLK	(L2 + 6.23 in.) ± 0.1 in. (6.23 in. comes from 2.5 in. of PCI option card clock length per PCI spec plus 3.73 in. of IBM 25JP-RISER3-A clock length of slot2)
PLL_FOUT3A	As short as possible
82376_PCICLK	(L2 + 6.23 in.) ± 0.1 in.
PLL_FOUT1A	As short as possible
ETHNT_PCICLK	(L2 + 6.23 in.) ± 0.1 in.
PLL_FOUT0A	As short as possible
664_PCICLK	(L2 + 6.23 in.) ± 0.1 in.
PLL_FOUT1	As short as possible
MPIC_PCICLK	(L2 + 6.23 in.) ± 0.1 in.
PLL_FOUT0	As short as possible
DPAL_PCICLK	(L2 + 6.23 in.) ± 0.1 in.
FPAL_PCICLK	(L2 + 6.23 in.) ± 0.1 in.
ISA_CLK	As short as possible
14.3181MHZ	As short as possible
OSC_14.3181MHZ	As short as possible
20.00MHz_XTAL1	As short as possible
20.00MHz_XTAL2	As short as possible
24MHZ_SUPERIO	As short as possible
XTAL1I_24.576MHZ	As short as possible

Table 104. Clock Wires (Continued)

Net Name	Length
XTAL1O_24.576MHZ	As short as possible
XTAL1I_16.934MHZ	As short as possible
XTAL1O_16.934MHZ	As short as possible
XTALI_33.8688MHZ	As short as possible
XTALO_33.8688MHZ	As short as possible
1.843MHZ_PMAN_CLK	As short as possible
16_X1	As short as possible
16_X2	As short as possible
XTAL1_32.768KHZ	As short as possible
XTAL2_32.768KHZ	As short as possible
SCSI_40MHZ	As short as possible
OSC_40MHZ	As short as possible
XTAL1_12MHZ	As short as possible
XTAL2_12MHZ	As short as possible
TIMER2_OUT	As short as possible

18.6 Noise Sensitive Wires

18.6.1 To Be Run With Adjacent Grounds

Noise sensitive wires require a ground wire in each adjacent wiring channel. See Table 105 for net names and comments. Connect shield (ground) traces to the ground plane using at least one via in each inch of trace. Completely surround the clock trace with shield traces.

Table 105. Noise Sensitive Wires (Ground Wires Adjacent)

NET NAME	COMMENT
–POWER_GOOD/RESET	
–BUFF_POWER_GOOD/RESET	
–HRESET_CPU1A	
–HRESET_CPU1B	
–HRESET_CPU2A	
–HRESET_CPU2B	
–SRESET_CPU1A	
–SRESET_CPU1B	
–SRESET_CPU2A	
–SRESET_CPU2B	

18.6.2 To Be Run With No Wires in Adjacent Channels

For each of the signals in Table 106, run either no trace or a ground trace in the adjacent channel on each side of the signal. If used, connect shield (ground) traces to the ground plane using at least one via in each inch of trace.

Table 106. Noise Sensitive Wires (No Wires Adjacent)

NET NAME	COMMENT
–ISA_REFRESH	
–PCI_SERR	
–ISA_IOCHCK	
ISA_AEN	
ISA_BALE	
–ISA_IOR	This also has other restrictions
–ISA_IOW	This also has other restrictions
–XIOR	
–XIOW	
ISA_IOCHRDY	
–ISA_MEMCS16	
–ISA_MEMR	
–ISA_MEMW	
–ISA_SBHE	
–ISA_SMEMR	
–ISA_SMEMW	
–ISA_IOCS16	
–ISA_MASTER	
–ISA_0WS	
–SCSI_ATN	This also has other restrictions
–SCSI_BSY	This also has other restrictions
–SCSI_ACK	This also has other restrictions
–SCSI_RESET	This also has other restrictions
–SCSI_MSG	This also has other restrictions
–SCSI_SEL	This also has other restrictions
–SCSI_CD	This also has other restrictions
–SCSI_REQ	This also has other restrictions
–SCSI_IO	This also has other restrictions

18.7 CEC Critical Wires

All CEC critical wires must be wired to be as short as possible (see Table 107).

Table 107. CEC Critical Wires

NET NAME	COMMENT
–DBB	2.5 in. max
DBB_P1TR	As short as possible
–K_DBG	As short as possible
–DBG	5.5 in. max
–K_TA	As short as possible
–TA	5.5 in. max
–K_TEA	As short as possible
–TEA	5.5 in. max
–K_ARTRY	As short as possible
–ARTRY	5.5 in. max
–K_AACK	As short as possible
–AACK	5.5 in. max
–K_TA	As short as possible
–TA	5.5 in. max
–K_TS	As short as possible
–TS	5.5 in. max
–K_XATS	As short as possible
–XATS	5.5 in. max
–K_TBST	As short as possible
–TBST	5.5 in. max
TSIZ<2:0>	5.5 in. max
TT<4:0>	5.5 in. max
A_CPU1<31:0> (1)	See note (1) for route sequence; 2.5 in. max
A_CPU2<31:0> (1)	See note (1) for route sequence; 5.0 in. max
D_CPU1<63:0> (2)	See note (2) for route sequence; 3.6 in. max
D_CPU2<63:0> (2)	See note (2) for route sequence; 5.0 in. max
DP_CPU1<7:0> (3)	See note (3) for route sequence; 3.6 in. max
DP_CPU2<7:0> (3)	See note (3) for route sequence; 5.0 in. max

Notes:

1. Address routing sequence: U3 (82664)—J8 (processor slot 1)—series resistor—J9 (Processor slot 2)—J12 (SRAM+TRAM connector).
2. Data routing sequence: U4 (82663)—J8 (processor slot 1)—series resistor—J9 (Processor slot 2)—J12 (SRAM+TRAM connector).
3. Data parity routing sequence: U4 (82663)—J8 (processor slot 1)—series resistor—J9 (Processor slot 2)—J12 (SRAM+TRAM connector).

18.8 L2 Cache Critical Wires

All L2 control signals from U3 (82664) to J12 (SRAM + TRAM conn.) must be wired as short as possible. The wires shown in Table 108 are particularly critical.

Table 108. L2 Critical Control Wires

NET NAME	COMMENT
TAG_MATCH	
–SRAM_CNT_EN/ADDR1	
–SRAM_ADS/ADDR0	

18.9 PCI Critical Control Wires

General requirements: PCI critical control wires are point to point. No stubs are desired, but if required, stubs must be less than 1.5 inches. All wires must be wired as short as possible. For net names and comments, see Table 109.

Table 109. PCI Critical Control Wires

NET NAME	COMMENT
–PCI_FRAME	J14 (PCI+ISA Dual Riser conn 25JP-RISER3-A) must be the last one on the net.
–PCI_TRDY	J14 must be the last one on the net.
–PCI_IRDY	J14 must be the last one on the net.
–PCI_STOP	J14 must be the last one on the net.
–PCI_DEVSEL	J14 must be the last one on the net.
–PCI_C/BE<3:0>	J14 must be the last one on the net.

18.10PCI Address/Data Wires

For comments and address routing sequence, see Table 110.

Table 110. PCI Address/Data Wires

NET NAME	COMMENT
PCI_AD<0:31>[1]	J14 must be the last one on the net.

Note:

1. PCI Address/Data routing sequence: U4 (82664)—U3 (82663)—U32 (82376)—U10 (MPIC)—U12 (Am79C970)—U12 (NCR53C810)—M14 (Direct ROM)—J14 (Dual Riser 25JP-RISER3-A)

18.11 SCSI I/O Wires

SCSI I/O wires must be as short as possible. At least one wiring channel on either side of these wires must be free of wiring unless it is another I/O wire. These wires must run perpendicular to any non-I/O wires on opposite planes. These wires must be wired on internal planes as much as possible. For net names and comments, see Table 111.

These wires must be wired in the following order: J32 (External SCSI)—U14 (SCSI Active Terminator)—U12 (Symbios 53C810)—J19 (Internal SCSI; unterminated).

Table 111. SCSI I/O Wires

NET NAME	COMMENT
SCSI_D<8:0>	
–SCSI_ATN	
–SCSI_BSY	
–SCSI_ACK	
–SCSI_RESET	
–SCSI_MSG	
–SCSI_SEL	
–SCSI_CD	
–SCSI_REQ	
–SCSI_IO	

18.12 Audio I/O Wires

Audio I/O wires must be as short as possible. All of these wires must remain in the audio section. No wires are allowed to cross over (either completely or partially) the voltage divider around the audio section. For net names and comments, see Table 112.

Table 112. Audio I/O Wires

NET NAME	COMMENT
L_MIC	
R_MIC	
EXT_L_CD	
EXT_R_CD	
L_LINE	
R_LINE	
L_CD	
R_CD	
L_OUT	
R_OUT	
AUD_VREF	
AUD_REFFLT	

18.13 Input/Output Wires

All of these wires must be as short as possible. At least one wiring channel on either side must be free of wiring unless it is another I/O wire. These wires must run perpendicular to any non-I/O wires on opposite planes. These wires must be wired on internal planes as much as possible. For net names and comments, see Table 113.

Table 113. I/O Wires

NET NAME	COMMENT
KYBD_DATA	
KYBD_DATA	
KYBD_CLK	
KYBD_CLK	
MOUSE_DATA	
MOUSE_DATA	
MOUSE_CLK	
MOUSE_CLK	
PARALLEL<16:0>	
–SOUT1	
–DTR1	
–RTS1	
–DCD1	
SIN1	
–DSR1	
–CTS1	
–RI1	
–SOUT2	
–DTR2	
–RTS2	
–DCD2	
SIN2	
–DSR2	
–CTS2	
–RI2	
–EXT_SOUT1	
–EXT_DTR1	
–EXT_RTS1	
–EXT_DCD1	
EXT_SIN1	
–EXT_DSR1	
–EXT_CTS1	

Table 113. I/O Wires (Continued)

NET NAME	COMMENT
–EXT_RI1	
–EXT_SOUT2	
–EXT_DTR2	
–EXT_RTS2	
–EXT_DCD2	
EXT_SIN2	
–EXT_DSR2	
–EXT_CTS2	
–EXT_RI2	

18.14 Ethernet Wires

Power and ground planes must be removed in the area of Twisted-Pair-Interface (C115, C114, T1, L3-4, L6-7, C575-576, C83, C91, & J21). This area must not have any other signal other than TXs & RXs passing by. For net names and comments, see Table 114.

TPI signals (RXs and TXs) must have matched length.

Table 114. TPI Wires

NET NAME	COMMENT
TX+	Path: U2.95 – R10 – T1.1 and U2.94 – R98 – T1.1
TX–	Path: U2.93 – R6 – T1.3 and U2.92 – R13 – T1.3
RX	Path: U2.90 – T1.6 and U2.89 – T1.8
TD+	Path: T1.16 – L7 – C83 – J21.1
TD–	Path: T1.14 – L6 – J21.2
RD+	Path: T1.11 – L4 – C91 – J21.3
RD–	Path: T1.9 – L3 – J21.6

18.15 Memory Wires

General requirements: Memory wires must be wired as short as possible. For net names and comments, see Table 115.

Table 115. Memory Wires

NET NAME	COMMENT
KMA<11:0>	BUS GROUP
MA_A<11:0>	BUS GROUP
MA_B<11:0>	BUS GROUP
MDP<7:0>	
–MRE<7:0>	
–MCE<7:0>	
MD<63:0>	BUS GROUP
–KMWE0	
–MWE_A0	
–MWE_B0	
–KMWE1	
–MWE_A1	
–MWE_B1	

18.16 Battery Wires

Battery connection to RTC (Signals BAT_PWR & VBAT) must be short and have 12-mil trace width.

18.17 Fan Wires

+12 volt traces to fans running at card edge must have 0.125 in. line width and must provide triple vias if required to change planes.

18.18 8mm Tape Contents and Extract Instructions

Part Number = MPRZAPDTU-03

18.18.1 Download Instructions

The enclosed 8mm tape was created using the tar command. First find the address of the 8mm tape drive by executing the following command:

```
isdev -C –c tape.
```

Then change the block size to 1024 by executing the following command:

```
chdev –l rmt0 –a block_size=1024
```

To extract the data, create a directory and ensure that at least 80M of free space is available. Use the cd command to get to the created directory and type the following:

```
tar –xvf/dev/rmt0
```

18.18.2 Cadence Version

- Concept=version 1.7–S1
- Allegro=version 11.0 or later
- packagerXL

18.18.3 Tape Contents

The tape contains the following directories:

PXL

- **Pst.* files,** packager files
- ***View* files,** Allegro feedback files
- **bom.* files,** bill of material files

Flatlab

- Logic models used for this design

(604 SMP Reference Board)

- Schematic data
- Postscript plots *Each page=plot.* total=*.ps*

tscr

- Cross reference sheets.

Section 19
Bills of Materials

19.1 Motherboard Bill of Materials

Table 116. Motherboard Bill of Materials

No	Part Name	Part Number	Reference Designator	Qty	Manufac-turer Name	Manufac-turer Part Number	JEDEC TYPE	Part Description
1	1N5817	87F4917	CR1	1	SANKEN	SFB54	DO_41	DIODE
2	2N7002	31F2311	Q5	1	MOTOROLA	2N7002	SOT23	N-FET
					SILICONIX	2N7002T1		
3	2N7002. DO_NOT_POP	NONE	Q3-Q4	2			SOT23	NFET
4	2X4BERG	72GO322	J49	1	BERG	69192-508H	CONN_ 72G0322	2X4 100MIL VERTICAL HEADER
5	74ABT244	42G3444	U13,U24,U42,U59	4	TI	SN74ABT 244DWR	SO20W	OCTAL BUS DRVR
6	74ALS21	55X8048	U5	1	TI	SN74ALS21 ADR	SO14	DUAL 4-INPUT AND
					NATIONAL	DM74ALS 21AM		
7	7400	17F7699	U61	1	NATIONAL	F00	SO14	QUAD 2-IN HAND
8	74F04	61X9235	U6, U15	2	TI	74FO4	SO14	HEX INVERTER
9	74F06	05H1579	U1	1	NATIONAL	74F06	SO14	HEX INVERTER
10	74F07	98F1773	U11	1	PHILIPS	F07	SO14	HEX BUFFER/DRV (OPEN COLLECTOR)
11	74F08	61X9236	U16	1	NATIONAL	74F08SCX	SO14	QUAD 2-IN AND
					PHILIPS	N74F08DT		
12	74F125	68X2888	U25,U34,U43,U44	4	NATIONAL	74F125	SO14	QUAD 3-ST BUFF
13	74F126	09F1697	U60	1	NATIONAL	74F126SCX	SO14	QUAD BUFFER F126
					PHILIPS	N74F126DT		
14	74F139	17F7853	U50	1	NATIONAL	74F139SCX	SO16	DUAL 1TO4 DEMUX
15	74F157	6300802	U22	1	NATIONAL	74F157SCX	SO16	QUAD 2-IN MUX
16	74F244	6480438	U7,U23,U27-U29, U45,U52-U54,U56, U58	11	PHILIPS	74F244	SO20W	OCT BUFF W 3-ST
17	74F245	55X8091	U55	1	NATIONAL	74F245	SO20W	OCTAL BUS TRANSCEIVER W 3-ST
18	74HCT14	37F9034	U21	1	PHILIPS	HCT14	SO14	HEX SCHWITT TRIG INV
19	8042H	1054195	U20	1	INTEL	8042H	PLCC44	MOUSE_KEYBRD CONTROLLER

Table 116. Motherboard Bill of Materials (Continued)

No	Part Name	Part Number	Reference Designator	Qty	Manufacturer Name	Manufacturer Part Number	JEDEC TYPE	Part Description
20	82378_82376	82G6542	U32	1	INTEL	82378ZB	QFP208_5MM	SIO PCI-ISA BRIDGE ASIC
21	87C750-PLCC28SKT_DO_NOT_POP	NONE	M1	1			PLCC28SKT	28-PIN PLCC SOCKET
22	AM79C970	40H8231	U2	1	AMD	AM79C970 AKC/W V.B2	QFP132_635MM	PCI ETHERNET CONTROLLER
23	CAPACITOR-0.001UF, 20%	98F1292	C62,C66,C76,C87, C94,C96,C109, C136-C138,C148, C149,C228,C230, C311,C550-C552	19	KYOCERA	08055C102 KAT2A	SMC0805	CAPACITOR
24	CAPACITOR-0.01UF, 20%	41F0313	C5,C7,C15,C37-C40,C44-C46,C50, C52,C53,C55-C57, C63,C64,C70-C72, C80-C82,C88-C90, C97-C100,C102, C110-C113,C120, C125,C126,C129, C133,C139,C142, C144,C150,C152, C153,C159,C160, C164,C166,C168, C169,C172,C174, C177,C178,C180, C183,C193-C198, C200,C204-C208, C211,C213,C232, C236,C237,C239, C240,C247,C250-C254,C258,C260-C263,C266,C267, C269,C271-C277, C281,C283,C284, C286-C292,C296, C301-C308,C313-C318,C321-C323, C328-C333,C337, C338,C347-C349, C362,C363,C375, C379,C380,C383, C384,C394,C395, C398,C399,C401, C405,C408,C411, C414,C419,C426, C428,C429,C434, C438,C443,C448, C458,C464,C481, C507,C508,C537-C544,C547,C548, C553-C557,C559-C564,C566-C571, C574,C577,C581-C587,C597-C600, C602-C605,C612-C619,C627-C629	222	KYOCERA	0805X103M 2B05	SMC0805	CAPACITOR
25	CAPACITOR-0.01UF_H,20%	75G8363	C83,C91	2	CERAMITE	5HKSS10R E	PIH2_D470M	CAPACITOR
26	CAPACITOR-0.047UF_H,20%	98F1521	C242	1	KYOCERA	0805Y104Z 1B05	SMC0805	CAPACITOR

Table 116. Motherboard Bill of Materials (Continued)

No	Part Name	Part Number	Reference Designator	Qty	Manufac-turer Name	Manufac-turer Part Number	JEDEC TYPE	Part Description
27	CAPACITOR-0.1UF,20%	41F0316	C3,C4,C6,C9,C12-C14,C16-C29,C58,C92,C107,C114,C115,C121,C140,C146,C156,C163,C167,C175,C191,C201,C212,C229,C231,C235,C238,C243,C246,C249,C257,C268,C280,C300,C334,C350,C354,C368,C374,C376,C385,C391,C400,C410,C415,C425,C427,C433,C480,C509,C526-C534,C565,C573,C704,C709,C712-C714	79	KYOCERA	0805Y104Z1B05	SMC0805	CAPACITOR
28	CAPACITOR-0.330UF,20%	70G3212	C60,C85,C467,C468	4	MURATA	GRM42-6X7R334K016AL	SMC1206	CAPACITOR
29	CAPACITOR-1000PF_NPO, 5%	6212672	C184,C190,C202,C209	4	AVX	08055A102JAT2A	SMC0805	CAPACITOR
30	CAPACITOR-100PF, 5%	57F7290	C11,C33,C41,C78	4	KYOCERA	1206C101K3B05	SMC1206	CAPACITOR
31	CAPACITOR-12PF,20%	98F1293	C241,C248,C255,C256,C264,C265,C278,C279,C293,C294,C309,C310,C319,C320,C335	15	AVX	08055A100JAT2A	SMC0805	CAPACITOR
32	CAPACITOR-12PF, 20%	98F1293	C706,C707	2	AVX	08055A120JAT2A	SMC0805	CAPACITOR
33	CAPACITOR-18PF, 20%	98F1294	C233	1	KYOCERA	0805C180J2B05	SMC0805	CAPACITOR
34	CAPACITOR-1UF, 20%	03G9351	C36,C47,C48,C67,C69,C74,C75,C86,C93,C105,C106,C122,C134,C135,C151,C158,C185,C299,C312,C325,C326,C340,C341,C343,C344,C357,C373,C377,C406,C407,C422,C452,C462,C549	34	MURATA	1206YG105ZAT1A	SMC1206	CAPACITOR
35	CAPACITOR- 2.2NF, 20%	42G3220	C54,C77,C128,C361,C366,C430,C441,C446,C447,C456,C461,C465,C471,C506,C812	15	MURATA	GRM40X7R222J050AD	SMC0805	CAPACITOR
36	CAPACITOR- 220PF, 20%	03G9606	C103,C104,C116-C118,C131,C132,C143,C154,C161,C162,C170,C171,C181,C182,C188,C189,C199	18	KYOCERA	08055A221JAT2A	SMC0805	CAPACITOR
37	CAPACITOR-220UF,20%	87F5126	C165,C176,C342,C378,C451	5	UCC	SXE6R3VB221V6X11FT	PTHRAD2	CAPACITOR

Table 116. Motherboard Bill of Materials (Continued)

No	Part Name	Part Number	Reference Designator	Qty	Manufac-turer Name	Manufac-turer Part Number	JEDEC TYPE	Part Description
38	CAPACITOR-22PF, 20%	41F0306	C8,C10,C119,C155, C336,C352,C353, C364,C365,C381. C382,C396,C397, C412,C413,C423, C424,C431,C432, C439,C440,C445	22	KYOCERA	0805C220J 2B05	SMC0805	CAPACITOR
39	CAPACITOR-330PF, 20%	03G9745	C795,C123	2	AVX	08055A331 JAT2A	SMC0805	CAPACITOR
40	CAPACITOR- 33PF, 20%	98F1297	C59,C65,C297, C298	4	KYOCERA	0805C330J 2B05	SMC0805	CAPACITOR
41	CAPACITOR- 470PF, 5%	41F0311	C32,C34,C715, C716	4	KEMET	C0805C471 J5GAC	SMC0805	CAPACITOR
42	CAPACITOR- 47PF, 20%	98F1299	C73,C157,C187, C210,C358,C359, C444,C450,C811	9	KEMET	C0805C470 J5GAC	SMC0805	CAPACITOR
43	CAPACITOR- 47PF_ 3KV,20%	NONE	C575,C576	2	PHILIPS COMP	D470M33S3 NRAAAL	PIH2_D470M	CAPACITOR
44	CAPACITOR- 5.6NF, 20%	34G3081	C43,C49,C108, C124,C545,C546	6	KEMET	C0805C562 J5RAC	SMC0805	CAPACITOR
45	CAPACITOR- 51PF, 10%	98F1517	C345	1	KYOCERA	0805C510K 2B05	SMC0805	CAPACITOR
46	CAPACITOR- 56PF, 20%	70G3305	C814	1	KEMET	C0805C560 K5GAC	SMC0805	CAPACITOR
47	CAPACITOR- 68PF, 20%	41F0307	C421,C437	2	KYOCERA	08051A680 GAT2A	SMC0805	CAPACITOR
48	CAPACITOR- 75PF, 20%	03G9753	C390,C420	2	AVX	08055A750J AT2A	SMC0805	CAPACITOR
49	CAPACITOR-DO_NOT_POP, 20%	NONE	C1,C51,C61,C79, C84,C101,C127, C147,C192,C203, C234,C245,C282, C285,C324,C339, C346,C351,C355, C356,C367,C369-C371,C386-C389, C402-C404,C416-C418,C435,C436, C442,C449,C580	39			SMC0805	CAPACITOR
50	CAPACITOR-DO_NOT_POP1206, 20%	NONE	C244,C578,C579, C592-C596,C601, C606-C611,C813	16			SMC1206	CAPACITOR
51	CDROM_CONN	6819495	J22	1	MOLEX	22041041	CONN_ 6819495	CDROM CONN
52	CHOKE3	10H6202	L33-L36	4	TDK	ZCYS51R5-M3PAT	CHOKE	EMI/RFI FILTER
53	CONN3-3POWER	—	J51	1	MOLEX	15-48-0217	CONN33 POWER	+3.3v POWER CONN (12-PIN)
54	CONN33POWER6	—	J50	1	MOLEX	15-48-0106	CONN33 POWER6	+3.3v POWER CONN (6-PIN)
55	CONNCPU256	81G4340	J8,J9	2	AMP	94-1793-132	CONN_ AMP256PIN	AMP 256-PIN DUAL-PIN READOUT PROCESSOR CONN
56	CONNPOWER6	55X8085	J13	1	MOLEX BURNDY	15480212 GTC12R4A	CONN_55X 8085	+/–5v & +/–12v POWER-CONN(12-PIN)
57	CONNSCSI50I	06H5792	J32	1	BERG	90333-050	CONN_06H 5792	2X25 INT SCSI CONN

Table 116. Motherboard Bill of Materials (Continued)

No	Part Name	Part Number	Reference Designator	Qty	Manufac-turer Name	Manufac-turer Part Number	JEDEC TYPE	Part Description
58	CONNSCSI50X	81F7827	J19	1	AMP	749830-5 OR 787171-5	CONN_81F 7827	SCSI 50-PIN PIH CONN
59	CONNSER9I	—	J27	1	AMP	104350-5	CONN_64F 4533	2X5 100MIL HDR (NO SHROUD)
60	CONN_BATTERY	19G2441	M6	1	SONY	CR2032Q 834	CONN_19G 2441	BATTERY HOLDER
61	CONN_BERG1X2	6181127	J17,J23,J31,J38, J39,J41,J44	7	AMP MOLEX	10435 1-1 90368-3002	CONN_ 6181127	1X2 100MIL HDR
62	CONN_BERG1X3	1501831	J20,J30,J34-J37, J43,J45	8	AMP	103240	CONN_ 1501831	1X3 100MIL VERT HDR
63	CONN_BERG1X3_ DO_NOT_POP	NONE	J28	1			CONN_ 1501831	1X3 100MIL VERT HDR
64	CONN_BERG1X3	1501831	J20,J30,J34-J37, J43,J45	8	BERG	79282-516	CONN_ 1501831	1X3 100MIL VERT HDR
65	CONN_BERG2X12_ DO_NOT_POP	NONE	J15,J16	2			CONN_99F 0481	2X12 100MIL VERTICAL HEADER (NO SHROUD)
66	CONN_BERG2X3	61G2325	J81	1	MOLEX	10-89-7062	CONN_61G 2325	2X3 100MIL VERTICAL HEADER
67	CONN_BERG2X8	91F9864	J33	1	BERG MOLEX	88873-016 87256-1641	CONN_91F 9864	2X8 100MIL SHROUDED VERTICAL HEADER
68	CONN_ESP	42F6867	J18	1	DUPONT	79282-516	CONN_42F 6867	2X8 HDR
69	CONN_FLOPPY34	06H4491	J42	1	BERG	90333-034	CONN_06H 4491	2X17 FLOPPY CONN
70	CONN_L2	—	J12	1	BURNDY	ELF182 JSC_3Z50	CONN_ELF 2X91	ELF 2-BAY 2X91 CONN
71	CONN_MDIN6	80F6855	J10,J11	2	MOLEX	871-23- 0602	CONN_15F 6890	6-PIN MINI DIN CONN
72	CONN_PARALLEL 25	10G4759	J25	1	BERG MOLEX	88873-026 87256-2611	CONN_10G 4759	2X13 VT SHRD HDR
73	CONN_PCI2X120	19G5789	J14	1	BURNDY	CEE2X120 SV-30Z14	CONN_72G 0353	2X120 DUALRISER CONN
74	CONN_SERIAL	90X8092	J26	1	T&B	MEM09RA 29CS	CONN_90X 8092	RS232C.D.SHELL.CONN
75	CONN_SIMM72	64F5806	J0-J7	8	AMP MOLEX	821997-3 15-82-0762	CONN_64F 5806	1X72 VERT SIMM
76	CRYSTL- DO_NOT_POP	NONE	Y11	1			XTALSMT2	QUARTZ CRYSTAL
77	CRYSTAL-NO- POP_MA505	NONE	Y8	1	EPSON	PKG.MA505	MA505-506	QUARTZ CRYSTAL
78	CS4232	05H1613	U8	1	CRYSTAL_ SEMI	CS4232	QFP100S_5MM	CRYSTAL_ SEMICONDUCTOR AUDIO CHIP
79	DIP4	—	M10	1	AMP	545566-4	DIP4_OSCSKT	DIP-4 SOCKET
80	DO-NOT-POP_16	NONE	S4,S5	2			SO16	
81	DO-NOT-POP_ PLCC20	NONE	S3	1			PLCC20	
82	DO-NOT-POP_ PLCC28	NONE	S6-S9	4			PLCC28	
83	DS1385	70G6764	U19	1	DALLAS	DS1385S	SO28W	RTC+4KX8 SRAM

Table 116. Motherboard Bill of Materials (Continued)

No	Part Name	Part Number	Reference Designator	Qty	Manufac-turer Name	Manufac-turer Part Number	JEDEC TYPE	Part Description
84	ELCAP-10UF,20%	74G1279	C145,C173,C179,C186,C327,C372,C469,C492,C524,C525	10	NEC	NRS106K16R8	SMC1311	CAPACITOR
85	ELCAP-22UF,20%	58F1742	C360	1	KEMET	T496D226M016AS	CAP33UF3SMT	CAPACITOR
86	ELCAP-33UF,20%	57G9281	C30,C31,C214-C227,C453-C455,C457,C459,C460,C463,C466,C470,C472-C479,C482-C491,C493-C505,C510-C523,C535,C536,C558,C620-C626	80	KEMET	T491D336MO16AS	CAP33UF3SMT	CAPACITOR
87	ELCAP-47UF,20%	10G0047	C141,C392,C572	3	KEMET	T496D476M010AS	CAP33UF3SMT	CAPACITOR
88	ELCAP_NOPOP2816	NONE	C2	1			CAP33UF3SMT	CAPACITOR
89	EMC_PAD	NONE	X7,X10	2	N/A		SKID	EMC PADS FOR AUDIO JACKS
90	E_XFORM-FL1066	59G2488	T1	1	VALOR	FL1066	EXFORM_FL1066	TRANSFORMER FILTER
91	FERITE-1000	78G9534	L47-L58	12	TDK	MMZ2012Y102BT	SMC0805	FERRITE
92	FERITE-HI-Z	98F1475	L3,L4,L6,L7,L37	5	TDK	HF70ACB-453215T	SMC1812	FERRITE
93	FERITE-LO-Z	26F4865	L42	1	TDK	HF50ACB-321611T	SMC1206	FERRITE
94	FERITE-MID-Z	26F4864	L1,L2,L5,L8,L11,L12,L14,L16,L27,L38-L41,L43,L45,L46	16	TDK	HF50ACB-322513T	SMC1210	FERRITE
95	HSJ1453	72G0318	J40,J46-J48	4	HOSIDEN	HSJ1453-010	JACK_3	AUDIO JACK
96	IBM82663	94G0178	U4	1	IBM	27-82663	QFP240_5MM	ASIC POWERPC-PCI BRIDGE (BUFFER)
97	IBM82664	20H2842	U3	1	IBM	27-82664	QFP208_5MM	ASIC POWERPC-PCI BRIDGE (CONTROLLER)
98	INDUCTOR-40	88G4874	L9,L10,L13,L15,L17-L26,L28-L32,L44	20	TDK	ACB2012M-040-T	SMC0805	INDUCTOR0R
99	IRFR014	57F7276	Q6	1	IR	IRFR014	D_PAKSMD	NFET-60V-8.4A
100	IRFZ44	03G9500	Q1	1	IR	IRFZ44	TO220AB_WHS	TRANSISTOR 0.50v 0.35A W/AAVID HEAT SINK 576802
101	KEYSTONE-QFP100SKT	—	M13	1	ALTERA	PL-SKT/Q100	100QFPSKT	SYSTEM EPLD SOCKET
102	LED_A-GREEN	42G3090	CR2	1	HP	HLMP1790	LED_100	LED HLMP 5v 7MA
103	LM317	62G9960	U35	1	NATIONAL	LM317T	TO220X	1.5A ADJ. VOLT REG
104	LM78M05	05H1525	U37	1	TI ST	UA7805KC L7805CT	TO220X	VOLT REG
105	LT1431REG	31F2428	U36	2	LINEARTECH	LT1431CS8	SO8	VOLT REGULATOR
106	MMBD914LT1	33F0009	CR3,CR6,CR18,CR22	4	MOTOROLA	MMBD914LT1	SOT23	0.2A 70v SWITCH DIODE
107	MPC970	05H1509	U26	1	MOTOROLA	MC88LV970FA	QFP52_65MM	PLL CLK DRIVER

Table 116. Motherboard Bill of Materials (Continued)

No	Part Name	Part Number	Reference Designator	Qty	Manufacturer Name	Manufacturer Part Number	JEDEC TYPE	Part Description
108	MPIC-ASIC-2A	39H9930	U10	1	TOSHIBA	TC160G 54AF–1414	QFP160_65MM	MP INTERRUPT CONTRLR (V.2A)
109	NCR53C810	70G6680	U12	1	SYMBIOS	SYM53C 810 609-0391635	QFP100R_65MM	SCSI CONTROLLER
110	OSCILLATOR-1.8432MHZ	31F2546	Y7	1	EPSON	SG625P-1.8432MC	OSCSMT4	OSCILLATOR
111	OSCILLATOR-14.3181MHZ	87F5263	Y6	1	EPSON	SG615P-14.31818 MC	OSCSMT4	OSCLR
112	OSCILLATOR-24.0MHZ	87F5265	Y9	1	EPSON	SG615P-24.0000MC	OSCSMT4	OSCILLATOR
113	OSCILLATOR-40.0MHZ	87F5269	Y1	1	EPSON	SG615PTJ-40.0000MC	OSCSMT4	OSCILLATOR
114	PAL16L8_PLCC-PLCC20_SKT	19G5840	M2,M3	2	AMP	822014-3	PLCC20SKT	PAL16L8
115	PAL20L8_PLCC-PLCC28_SKT	99F2946	M8	1	AMP	822039-3	PLCC28SKT	PAL20L8 SOCKET
116	PAL22V10_PLCC-PLCC28_SKT	99F2946	M12	1	AMP	822039-3	PLCC28SKT	PAL22V10 SOCKET
117	PALCE16V8_PLCC-PLCC20_SKT	19G5840	M11	1	AMP	822014-3	PLCC20SKT	PALCE16V8 SOCKET
118	16.5MHZ_OSC		X19	1	EPSON	16.5000	SG_531P	OSCILLATOR
119	87C750 _DO_NOT_POP	NONE	X1	1			PLCC28	UNPROGRAMMED MICROCONTROLLER
120	BATTERY_3V	15F8409	X6	1	SONY	CR2032		BATTERY
121	CPUFREQ_PAL		X8	1	AMD	PAL20L8-10/2JC	PLCC28	UNPROGRAMMED PAL
122	HRESET_PAL	68X6041	X3	1	TI	BPAL16L8-10CFN	PLCC20	UNPROGRAMMED PAL
123	HSINK576802	11F3898	X17	1	AAVID	576802		REG. HEAT SINK
124	KEYSTONE	05H0966	X13	1	ALTERA	EPM5130Q C100-1	PQFP-100	UNPROGRAMMED EPLD
125	SRESET_PAL	68X6041	X2	1	TI	BPAL16L8-10CFN	PLCC20	UNPROGRAMMED PAL
126	ST93CS46B1	88G9857	X16	1	SGS THOMSON	ST93CS46 B1	PDIP-8	UNPROGRAMMED EE-PROM
127	VPD_ROM	82G6496	X9	1	AMD	AM29F040-120JC	PLCC32	UNPROGRAMMED FLASH ROM
128	PC87332	07H5030	U33	1	NATIONAL	PC87332	QFP100R_65MM	SUPERIO CHIP
129	PLCC32ROMSKT	10G7624	M14	1	AMP	821977-1	PLCC32SKT	PLCC32PIN SOCKET W/O POSTS
130	PLCC32ROMSKT_DO_NOT_POP	NONE	M9	1			PLCC32SKT	SOCKET
131	POLYSWITCH-1.1A	34G3113	F1-F3	3	RAYCHEM	SMD100	POLYSWITCH	THERMISTOR 0.180 O 0.1W
132	POWER1X3	65G3724	J29	1	MOLEX FOXCONN	705430037 HF01030	CONN_65G 3724	AUX POWER CONN (3-PIN HEADER)

Table 116. Motherboard Bill of Materials (Continued)

No	Part Name	Part Number	Reference Designator	Qty	Manufac-turer Name	Manufac-turer Part Number	JEDEC TYPE	Part Description
133	RESISTOR-0,5%	98F1665	R8,R26,R33,R41, R46,R47,R55,R167, R169,R174,R250, R274,R280,R284, R295,R296,R299, R305-R307,R309, R322,R327,R328, R330,R345,R346, R351,R361,R364, R368,R377,R396, R402,R404,R436, R445,R448,R477, R479,R487,R509, R526,R538,R540, R544,R564,R704, R726-R728,R736, R738,R741-R758, R806,R807,R811, R818,R826,R828, R829,R853,R855, R859,R860,R884, R891,R912	87	PANASONIC	ERJ8 GVJ0R	SMC0805	RESISTOR
134	RESISTOR-0_ 1206,5%	40G7355	R801-R805,R913	6	PANASONIC	ERJ8GVY 0R00S	SMC1206	RESISTOR
135	RESISTOR-1.21K,5%	40G6980	R175	1	PANASONIC	ERJ-6VNF1 211S	SMC0805	RESISTOR
136	RESISTOR-1.47K,1%	58F1830	R537	1	PANASONIC	ERJ6VF 1471S	SMC0805	RESISTOR
137	RESISTOR- 1.5K,5%	98F1741	R192, R405	2	ROHM	MCR10EZH LJW152	SMC0805	RESISTOR
138	RESISTOR-10,5%	58F1831	R185,R298,R319, R341,R449,R493, R533,R553,R740	9	PANASONIC	ERJ6GVYJ 100S	SMC0805	RESISTOR
139	RESISTOR- 100,5%	41F0328	R4,R5,R9,R14,R44, R70,R100,R110, R136,R155,R234, R267,R288,R300, R323,R338,R343, R348,R359,R376, R411,R416,R458, R521,R523,R555, R560,R584,R759, R823,R896,R900	32	PANASONIC	EFJ-6GVYJ 101S	SMC0805	RESISTOR
140	RESISTOR- 100K,5%	40G7315	R108	1	PANASONIC	ERJ-6GVYJ 104S	SMC0805	RESISTOR

Table 116. Motherboard Bill of Materials (Continued)

No	Part Name	Part Number	Reference Designator	Qty	Manufacturer Name	Manufacturer Part Number	JEDEC TYPE	Part Description
141	RESISTOR-10K,5%	41F0337	R1-R3,R7,R11, R15-R17,R19,R23-R25,R27,R29,R31, R32,R34-R36,R39, R43,R48,R49,R53, R54,R56-R58,R62, R63,R68-R69,R72, R75,R77,R78,R81, R84,R85,R87,R88, R90-R96,R105, R107,R114,R119, R122,R128,R137, R140,R141,R144, R148-R150,R152, R158,R160,R163, R166,R168,R173, R179,R183,R184, R190,R194,R198, R200,R201,R206, R222,R242,R244, R259,R261-R263, R265,R268,R271, R285,R302-R304, R316,R339,R344, R354,R378,R382, R392,R397,R398, R407,R423,R424, R427,R431,R435, R438,R442,R446, R454,R457,R462, R491,R503,R516, R519,R525,R532, R534,R535,R557, R561,R572,R573, R575,R582,R583, R683,R686,R698, R730,R731,R765, R768,R769,R842, R843,R845,R846-R850,R856,R857, R861,R903-R906, R909-R910	153	PANASONIC	ERJ6GVJ 103S	SMC0805	RESISTOR
142	RESISTOR-110,1%	08G5106	R239	1	PANASONIC	ERJ6VNF 1100S	SMC0805	RESISTOR
143	RESISTOR-12,5%	40G7454	R227,R518	2	PANASONIC	ERJ6GVYJ 120S	SMC0805	RESISTOR
144	RESISTOR-120,5%	09G9649	R241,R586	2	PANASONIC	ERJ-6GVYJ 121S	SMC0805	RESISTOR
145	RESISTOR-150,5%	98F1671	R203	1	ROHM	MCR10EZH MJW151	SMC0805	RESISTOR
146	RESISTOR-1K,5%	41F0333	R176,R180,R193, R197,R209,R223, R225,R235,R238, R275,R287,R301, R340,R347,R350, R362,R366,R379, R383,R386,R401, R453,R459,R466, R469,R482,R498, R515,R524,R539, R556,R565,R568, R570,R585,R703, R708,R710,R729, R739,R777-R799, R841,R844	65	PANASONIC	ERJ6GVJ 102S	SMC0805	RESISTOR
147	RESISTOR-2.0K,5%	58F1835	R569,R571	2	PANASONIC	ERJ6GVJ 202S	SMC0805	RESISTOR
148	RESISTOR-2.2,5%	42G3063	R240	1	PANASONIC	ERJ6GVYK 2R2S	SMC0805	RESISTOR

Table 116. Motherboard Bill of Materials (Continued)

No	Part Name	Part Number	Reference Designator	Qty	Manufac-turer Name	Manufac-turer Part Number	JEDEC TYPE	Part Description
149	RESISTOR-2.2K,5%	40G7473	R121,R133,R134, R210,R342,R440	6	PANASONIC	ERJ6GVYJ 222S	SMC0805	RESISTOR
150	RESISTOR-2.7K,5%	09G9748	R290,R310,R331, R335,R336,R355, R356,R372,R373, R390,R391,R408, R432,R447,R463, R476	16	PANASONIC	ERJ6GVYJ 272S	SMC0805	RESISTOR
151	RESISTOR-200,5%	98F1424	R858	1	PANASONIC	ERJ-6GVYJ 201S	SMC0805	RESISTOR
152	RESISTOR-22,5%	98F1736	R99,R101,R102, R104,R106,R112, R117,R125,R216, R255,R258,R260, R264,R266,R273, R277,R283,R291, R292,R311,R320, R332,R352,R367, R369,R387,R444, R474,R718	29	PANASONIC	ERJ6GVYJ 220S	SMC0805	RESISTOR
153	RESISTOR-220,5%	61F2951	R50	1	PANASONIC	ERJ6GVJ 221S	SMC0805	RESISTOR
154	RESISTOR-220K,5%	61F2952	R170	1	PANASONIC	ERJ6GVJ 224S	SMC0805	RESISTOR
155	RESISTOR-22,5%	08G5178	R116,R120,R131, R165,R186,R207, R489,R502	8	PANASONIC	ERJ6GVYJ 223S	SMC0805	RESISTOR
156	RESISTOR-237,1%	61F2953	R549	1	PANASONIC	ERJ6VF 2370S	SMC0805	RESISTOR
157	RESISTOR-249,1%	40G7233	R213	1	PANASONIC	ERJ6VNF 2490S	SMC0805	RESISTOR
158	RESISTOR-24K,5%	62G9907	R420	1	PANASONIC	ERJ6GVYJ 243S	SMC0805	RESISTOR
159	RESISTOR-3.0K,5%	61F2955	R115,R130	2	PANASONIC	ERJ6GVYJ 302S	SMC0805	RESISTOR
160	RESISTOR-3.9K,5%	42G3067	R189,R522,R554	3	PANASONIC	ERJ6GVYJ 392S	SMC0805	RESISTOR
161	RESISTOR-300,5%	98F1674	R233,R279,R294, R314	4	ROHM	MCR10EZH MJW301	SMC0805	RESISTOR
162	RESISTOR-33,5%	41F0327	R30,R38,R45,R52, R61,R73,R82,R89, R123,R138,R139, R146,R147,R156, R157,R162,R172, R182,R196,R205, R215,R252-R254, R256,R257,R325, R337,R357,R374, R380,R393-R395, R410,R414,R415, R417,R425,R434, R437,R452,R455, R467,R468,R470, R480,R481,R496, R497,R510,R512-R514,R527,R529, R530,R542,R567, R588-R671,R684, R687-R694,R702, R711-R717,R764, R767,R800,R886-R887,R889-R890	167	PANASONIC	ERJ6GVYJ 330S	SMC0805	RESISTOR

Table 116. Motherboard Bill of Materials (Continued)

No	Part Name	Part Number	Reference Designator	Qty	Manufac-turer Name	Manufac-turer Part Number	JEDEC TYPE	Part Description
163	RESISTOR-330,5%	08G4750	R460,R507	2	ROHM	MCR10EZH LJW3310	SMC0805	RESISTOR
164	RESISTOR-330_0.25W,5%	40G7369	R381	1	PANASONIC	ERJ8GEYJ 331V	SMC1206	RESISTOR
165	RESISTOR-360K,5%	74G1122	R520	1	ROHM	MCR10EZH MJW364	SMC0805	RESISTOR
166	RESISTOR-39,5%	61F2958	R202,R204,R230, R236	4	PANASONIC	ERJ6GVJ3 90S	SMC0805	RESISTOR
167	RESISTOR-39K,5%	70G6020	R153,R164,R421, R441	4	PANASONIC	ERJ8GVYJ 393S	SMT0805	RESISTOR
168	RESISTOR-4.7K,5%	41F0336	R109,R135,R143, R145,R171,R181, R211,R217,R221, R237,R243,R278, R293,R315,R413, R471,R483-R485, R500,R504,R508, R562,R677,R685, R719-R724	31	PANASONIC	ERJ-6GVJ 472S	SMC0805	RESISTOR
169	RESISTOR-422,1%	40G6945	R13,R98,R142	3	PANASONIC	ERJ6VNF 4220S	SMC0805	RESISTOR
170	RESISTOR-47,5%	03G9708	R232,R317,R898	3	ROHM	MCR10EZH LJW470	SMC0805	RESISTOR
171	RESISTOR-470K,5%	03G9709	R214	1	ROHM	MCR10EZH LJW474	SMC0805	RESISTOR
172	RESISTOR-5.6K,5%	98F1737	R177,R178,R187, R188,R313,R333, R353,R370,R388, R406,R430	11	PANASONIC	ERJ-6GVJ 562S	SMC0805	RESISTOR
173	RESISTOR-510,5%	41F0331	R429,R443	2	PANASONIC	ERJ-6GVYJ 511S	SMC0805	RESISTOR
174	RESISTOR-51K,5%	03G9688	R229	1	ROHM	MCR10EZH LJW513	SMC0805	RESISTOR
175	RESISTOR-54.9,1%	40G6873	R97,R103,R118, R126,R127,R270, R276,R289,R308, R329,R365,R384, R385,R403,R428	15	PANASONIC	ERJ-6ENF 54R9V	SMC0805	RESISTOR
176	RESISTOR-560,5%	09G9749	R422	1	PANASONIC	ERJ-6GVYJ 561S	SMC0805	RESISTOR
177	RESISTOR-6.8K,5%	03G9710	R129,R132,R199, R208,R219,R358, R548,R559	8	PANASONIC	ERJ6GVYJ 682S	SMC0805	RESISTOR
178	RESISTOR-61.9,1%	40G6878	R6,R10	2	PANASONIC	ERJ6VNF 61R9S	SMC0805	RESISTOR
179	RESISTOR-620,5%	61F2960	R349,R472,R488	3	PANASONIC	ERJ6GVJ 621S	SMC0805	RESISTOR
180	RESISTOR-62K,5%	03G9707	R248,R251,R517, R547	4	PANASONIC	ERJ-6VNF 6192S	SMC0805	RESISTOR
181	RESISTOR-75,5%	61F2961	R113,R228,R324, R334,R672,R673, R676,R682,R808-R810,R812-R815, R817,R819-R822, R824,R825,R827, R830,R832-R839, R851,R852	34	PANASONIC	ERJ6GVJ 750S	SMC0805	RESISTOR

Table 116. Motherboard Bill of Materials (Continued)

No	Part Name	Part Number	Reference Designator	Qty	Manufacturer Name	Manufacturer Part Number	JEDEC TYPE	Part Description
182	RESISTOR-8.2K,5%	03G9712	R65,R272,R281, R286,R297,R318, R371,R389,R450, R465,R478,R494, R495,R511,R528, R536,R541,R550, R674,R675,R699	21	ROHM	MCR10EZH LJW822	SMC0805	RESISTOR
183	RESISTOR-8.66K,5%	03G9682	R419	1	PANASONIC	ERJ6VF 8661S	SMC0805	RESISTOR
184	RESISTOR-910,5%	57G8589	R375	1	PANASONIC	ERJ6GVYJ 910S	SMC0805	RESISTOR
185	RESISTOR-DO_NOT_POP,5%	NONE	R12,R18,R20-R22, R28,R37,R40,R42, R51,R59,R60,R64, R66,R67,R71,R74, R76,R79,R80,R83, R86,R111,R124, R151,R154,R159, R161R191,R195, R212,R218,R220, R224,R226,R231, R245,R246,R248, R249,R251,R269, R326,R282,R312, R360,R363,R399, R400,R409,R412, R418,R426,R427, R433,R439,R451, R456,R461,R464, R473,R475,R486, R490,R492,R499, R501,R505,R506, R531,R543,R545, R546,R551,R552, R558,R563,R566, R574,R576-R581, R587,R678-R681 R695-R697,R700, R701,R705-R707, R709,R725R732-R733,R761-R763, R766,R770-R776, R816,R831,R840, R854,R862-R864, R885,R888,R892, R893-R895,R897, R899,R901-R902, R907-R908,R911, R914	134			SMC0805	RESISTOR
186	RESISTOR-NO_POP1812,5%	NONE	R737	1			SMC1812	RESISTOR
187	RJ45S	59G2489	J21	1	AMP	555164-1	CONN_RJ45	RJ45 CONN W/O PANEL STOPS
188	S4403	82G6505	U30	1	AMCC	S4403B	PLCC44	AMCC CLOCK CHIP
189	SER_DIODE	45F8654	CR4,CR5,CR7-CR16	12	ROHM	DAN217	SC59	70v 100MA SWITCH DIODE
190	SFPB54VL	87F4929	CR19-CR20	2	SANKEN/ ALLEGRO	SFPB-54VL	D64	SCHOTTKY RECTIFIER
191	SFPB54VL _DO_NOT_POP	NONE	CR21,CR23	2			D64	SCHOTTKY RECTIFIER
192	SN75C185	82G2416	U46,U47	2	TI	SN75C185	SO20W	LO-POW 3-DRVR X 5-RCVR RS-232-D
193	ST93CS46-SOCKET	6359335	M16	1	AMP	2-640463-3	DIP8_300SKT	8-PIN SERIAL EEPROM SOCKET
194	TDA1308T	05H1618	U40	1	PHILIPS	TDA1308 TD-T	SO8	HEADPHONE DRV CLASS AB

Table 116. Motherboard Bill of Materials (Continued)

No	Part Name	Part Number	Reference Designator	Qty	Manufacturer Name	Manufacturer Part Number	JEDEC TYPE	Part Description
195	TDA7052	70G6352	U39	1	PHILIPS	TDA7052AN	DIP8_300	LOW-VOLT AUDIO PWR AMP
196	TL084	57G8584	U18,U38	2	TI	TL084ACD	SO14	UAD-OPAMP
197	UC5601	70G6712	U14	1	UNITRODE	UC5601DW	SO28	SCSI ACTIVE TERMINATOR
198	XC17000-DO_NOT_POP	NONE	M4,M5	2			DIP8_300SKT	XC17000 SERIAL PROM SOCKET
199	XTAL-12.000MHZ	57G8930	Y5	1	EPSON	12.000MHZ	MA505	QUARTZ CRYSTAL
200	XTAL-15.00MHZ	89G3878	Y13	1	ECLIPTEK	EC150S-15.00	HC49U	QUARTZ CRYSTAL
201	XTAL-16.00MHZ_DO_NOT_POP	NONE	Y4	1			HC49U	QUARTZ CRYSTAL
202	XTAL-20.00MHZ	89G3768	Y10	1	FOX	20.000MHZ	HC49US	QUARTZ CRYSTAL
203	XTAL-24.576MHZ	80G5442	Y2	1	CARDINAL	CLP24576D-176D-16	CLP_XTAL	QUARTZ CRYSTAL
204	XTAL-32.768KHZ	03G9527	Y3	1	EPSON	32.768KHZ	MC405	QUARTZ CRYSTAL
205	XTAL-33.8688MHZ	89G3793	Y15	1	FOX	33.86 88M	HC49U	QUARTZ CRYSTAL
					NDK	33.8688M		
206	XTAL-DO_NOT_POP	NONE	Y12	1			CLP_XTAL	QUARTZ CRYSTAL
207	YAC516E	05H1501	U57	1	YAMAHA	YAC516-E	SSOP24	SOUND
208	YMF289S	05H1502	U51	1	YAMAHA	YMF289-S	QFP48_5MM	FM SYNTHESIZER
209	ZENER_DO_NOT_POP	NONE	CR17	1			SOT23	ZENER DIODE

19.2 CPU Card Bill of Materials

Table 117. Cheetah0 (CPU Card) Bill of Materials

No	Part Name	Part Number	Reference Designator	Qty	Manufac-turer Name	Manufac-turer Part Number	JEDEC TYPE	Part Description
1	604BGANH-SIOC	A123_XX	U1	1	IBM	03H4964-D D1_66MHZ	SBC-256NH	604_DD1_66MHZ IN BGA
2	CAP-.1UF,20%	57G8514	C1,C4,C21-C40	22	TDK	C1608Y5V1 E1042T 000A	SMC0603	CAPACITOR
3	CAP-0.1UF,20%	41F0316	C14-C18	5	KYOCERA	0805Y104Z 1B05	SMC0805	CAPACITOR
4	CAP-DO_NOT_POP, 20%	NONE	C3,C5	2			SMC0805	CAPACITOR
5	CAPPLUS-10UF,20%	70G6616	C2	1	KEMET	T491D106M 020AS	CAP33UF3SMT	CAPACITOR
6	CAPPLUS-220.0UF, 20%	78G9368	C6	1	AVX	TAJE227 K010	CAP33UF3SMT	CAPACITOR
7	CAPPLUS-33UF,20%	57G9281	C7,C8,C10,C12	4	KEMET	T491D336M 016AS	CAP33UF3SMT	CAPACITOR
8	CAPPLUS-DO_NOT_POP, 20%	NONE	C9,C11,C13	3			CAP33UF3SMT	CAPACITOR
9	CONN_TEST14-SIOC		J2	1	BERG		CONN_BERG 2X7	BERG2X7
10	CPUHEATSINK-SIOC		M1	1	SANYO DENKI	REMP5412 H2026	CPUHEATSINK	FANSINK
11	EDGECONN256-SIOC	NONE	J1	1		NONE	DIP256	256-PIN DUAL-PIN READOUT PROCESSOR CARD EDGE
12	IRFZ44A DO_NOT_POP	NONE	Q1	1			TO220	
13	LT1431REG-SIOC DO_NOT_POP	NONE	U3	1			SO8	
14	PAL_28PLCC-DO_NOT_POP	NONE	U2				PLCC28	
15	RESISTOR-0.00,5%	15G14I70	R29,R36,R37,R39	4	PANASONIC	ERJ14Y0R 00H	SMC1210	RESISTOR
16	RESISTOR-00.0,5%	71F7680	R12,R40,R43,R46, R47	5	PANASONIC	ERJ3GSY 0R00V	SMC0603	RESISTOR
17	RESISTOR-1.0K,5%	71F7670	R14	1	PANASONIC	ERJ3GSY J102V	SMC0603	RESISTOR
18	RESISTOR-10.0,5%	71F7664	R38	1	PANASONIC	ERJ3GSY J100V	SMC0603	RESISTOR
19	RESISTOR-10.0K,5%	71F7676	R3,R4,R7,R17,R20, R21,R23,R28,R30, R31,R33,R35	12	PANASONIC	ERJ3GSY J103V	SMC0603	RESISTOR
20	RESISTOR-33.0,5%	71F7666	R25	1	PANASONIC	ERJ3GSYJ 330V	SMC0603	RESISTOR
21	RESISTOR-NO_POP, 5%	NONE	R1,R2,R5,R8,R9, R11,R13,R16,R19, R22,R24,R26,R32, R34,R41,R42,R44, R45	18			SMC0603	RESISTOR
22	RESISTOR-NO_POP 1,5%	NONE	R6,R10,R15,R18, R27	5			SMC0805	RESISTOR
23	VR0HEATSINK-SIOC DO_NOT_POP	NONE	HS1	1			VR0HEATSINK	

19.3 Riser Bill of Materials

Table 118. Riser Bill of Materials

No	Part Name	Part Number	Ref Des	Qty	Manufacturer Name	Manufacturer Part Number	JEDEC_TYPE	Part Description
1	74ALS243-1	37F8957	U6	1	NATIONAL	74ALS243	SOG14_250	QUAD XCVR
2	74ALS35-1	49G3287	U2	1	NATIONAL	74ALS35	SOG14_250	HEX NONINVERTERS
3	74F125-1	68X2888	U7	1	NATIONAL	74F125	SOG14_250	QUAD TRI-STATE BUFFER
4	CAP-.01UF	41F0313	C12,C15,C16,C18, C22,C24,C26,C27, C30,C59,C64,C65, C67,C68,C71,C72, C74,C77,C78,C81, C82	21	KYOCERA	0805X103M2B05	SMC805	CAPACITOR
5	CAP-220UF	78G9368	C21,C29	2	AVX	TAJE227K010	CAP33UF3SMT	CAPACITOR
6	CAP-33UF	10G0056	C1-C9,C17,C19, C20,C23,C25,C58, C96	16	KEMET	T496D336M016AS	SMC2816C	CAPACITOR
7	CAP-DO_NOT_POP	NONE	C63,C66,C70,C73, C79,C80,C83-C87, C90,C91,C93-C95, C97-C103	23			SMC1206	CAPACITOR
8	CAP-DO_NOT_POP1	NONE	C10,C11,C13,C14, C57	5			SMC2816C	CAPACITOR
9	CONN33POWER6-1	—	J7	1	MOLEX	15-48-0410	CONN33POWER6	+3.3v POWER CONN(6-PIN)
10	CONN_2X120PCI-1	TABCONN	J5	1			TAB2X120	2x120 CONNECTOR
11	CONN_PCI2X60-1	72G0316	J4,J6	2	AMP	646255-1	PCI2X605V	2x60 CONNECTOR
12	ISA_BUS-1	6137473	J1-J3	3	AMP	645169-4	ISA2X49	CARD-EDGE CONNECTOR
13	RESISTOR-0OHM5%	98F1665	R2,R8	2	PANASONIC	ERJ8GV0R	SMC805	RESISTOR
14	RESISTOR-100OHM5%	40G7357	R27	1	PANASONIC	ERJ8GEYJ101V	SMC805	RESISTOR
15	RESISTOR-10OHM5%	58F1831	R20-R22,R29,R33, R34,R36,R47,R62-R65	12	PANASONIC	ERJ6GVYJ100S	SMC805	RESISTOR
16	RESISTOR-4.7K5%	41F0336	R4,R5,R10,R11	4	PANASONIC	ERJ6GVYJ472S	SMC805	RESISTOR
17	RESISTOR-8.2K5%	03G9712	R12-R15,R18,R23, R24,R44-R46,R66	11	ROHM	MCR10EZHLJW822	SMC805	RESISTOR
18	RESISTOR-DO_NOT_POP	NONE	R1,R3,R6,R7,R9, R16,R17,R19,R25, R26,R28,R31,R32, R35,R37-R42,R48-R50,R53,R55,R57, R59-R61	29			SMC805	RESISTOR
			TOTAL	136				

Section 20
Errata

This is the errata section of release 3.0 of the reference design, which is associated with pass 3 of the reference board. The previous (preliminary) version was known as release 1.0 of the reference design, which corresponds to pass 2.0 of the reference board.

At this time, there are no known errata to this release of the reference design.

Release 3.0 of the reference design incorporates the workarounds that were required for the errata found on release 1.0 of the reference design. These workarounds are of two types — board level and 660 bridge related.

Board level errata to the release 1.0 reference design required trace cutting, component removals, wire additions, and component additions. All of these modifications have been incorporated into the physical design of the release 3.0 reference design.

Errata related to the 660 bridge were required on the release 1.0 reference design to accomodate the errata and product changes associated with the previous revision of the 660 bridge (663 revision 2.0 and 664 revision 1.1). The final revision of the 660 bridge is 663 revision 2.0 and 664 revision 1.2.

Changes to the release 1.0 reference design to accomodate the 663 have been incorporated into the physical design of the release 3.0 reference design.

Changes made to the release 1.0 reference design to accomodate the 664 have been incorporated into the physical design of release 3.0 of the reference design in such a way as to allow the design to be used with either revision 1.1 or revision 1.2 of the 664. One set of 0Ω resistors is populated to select revision 1.2 of the 664 (no PALs). A different set of resistors is populated (and the workaround PALs installed) to select revision 1.1 of the 664. The schematic and BOM reflect usage of revision 1.2 of the 664.

Revision 3.0 of the reference design contains no workaround PALs for revision 1.2 of the 664. Should any workaround PALs be required for revision 1.2 of the 664, it may be possible to use the sites used by the workaround PALs associated with revision 1.1 of the 664. For more information on the 660 Bridge, contact an IBM technical representative for current documentation.