Ultra Plus and Ultra 60*x*
Installation and Use

ULMB60XA/IH3

Notice

While reasonable efforts have been made to assure the accuracy of this document, Motorola, Inc. assumes no liability resulting from any omissions in this document, or from the use of the information obtained therein. Motorola reserves the right to revise this document and to make changes from time to time in the content hereof without obligation of Motorola to notify any person of such revision or changes.

It is possible that this publication may contain reference to, or information about Motorola products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that Motorola intends to announce such Motorola products, programming, or services in your country.

Restricted Rights Legend

If the documentation contained herein is supplied, directly or indirectly, to the U.S. Government, the following notice shall apply unless otherwise agreed to in writing by Motorola, Inc.

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Motorola, Inc.
Computer Group
2900 South Diablo Way
Tempe, Arizona 85282-9602

Preface

This document provides general information, hardware installation and serviceability, operating instructions, and functional description for the Ultra Plus 603e, Ultra Plus 604, Ultra 603, Ultra 603e, and Ultra 604 PowerPC™ series of motherboard platforms. It also includes general information and instructions for using the PPCBug debug firmware.

This document is intended for PC manufacturers, technical original equipment manufacturers (OEMs), and system integrators, who want to design OEM systems, supply additional capability to an existing PC-compatible system, or work in a lab environment for experimental purposes.

A basic knowledge of computers and digital logic is assumed.

To use this document, you may wish to become familiar with the publications listed in Appendix A, *Related Documentation*.

Document Conventions

The following conventions are used in this document:

bold

is used for user input that you type just as it appears. Bold is also used for commands, options and arguments to commands, and names of programs, directories, and files.

italic

is used for names of variables to which you assign values. Italic is also used for comments in screen displays and examples.

`courier`

is used for system output (e.g., screen displays, reports), examples, and system prompts.

RETURN or **<CR>** or **ENTER**

represents the Return or Enter key.

CTRL

represents the Control key. Execute control characters by pressing the **CTRL** key and the letter simultaneously, e.g., **CTRL-d**.

Document Terminology

Throughout this document, a convention has been maintained whereby data and address parameters are preceded by a character which specifies the numeric format, as follows:

$	dollar	specifies a *hexadecimal* number
%	percent	specifies a *binary* number
&	ampersand	specifies a *decimal* number

For example, "12" is the decimal number twelve, and "$12" is the decimal number eighteen. Unless otherwise specified, all address references are in hexadecimal throughout this document.

An asterisk (*) following the signal name for signals which are *level significant* denotes that the signal is *true* or valid when the signal is low.

An asterisk (*) following the signal name for signals which are *edge significan*t denotes that the actions initiated by that signal occur on high to low transition.

In this document, *assertion* and *negation* are used to specify forcing a signal to a particular state. In particular, *assertion* and *assert* refer to a signal that is active or true; *negation* and *negate* indicate a signal that is inactive or false. These terms are used independently of the voltage level (high or low) that they represent.

Data and address sizes are defined as follows:

A *byte* is eight bits, numbered 0 through 7, with bit 0 being the least significant.

A *two-byte* is 16 bits, numbered 0 through 15, with bit 0 being the least significant. For other RISC modules, this is called a *half-word*.

A *four-byte* is 32 bits, numbered 0 through 31, with bit 0 being the least significant. For the other RISC modules, this is called a *word*.

An *eight-byte* is 64 bits, numbered 0 through 63, with bit 0 being the least significant. For the other RISC modules, this is called a *double-word*.

Safety Summary
Safety Depends On You

The following general safety precautions must be observed during all phases of operation, service, and repair of this equipment. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the equipment. Motorola, Inc. assumes no liability for the customer's failure to comply with these requirements.

The safety precautions listed below represent warnings of certain dangers of which Motorola is aware. You, as the user of the product, should follow these warnings and all other safety precautions necessary for the safe operation of the equipment in your operating environment.

Ground the Instrument.

To minimize shock hazard, the equipment chassis and enclosure must be connected to an electrical ground. The equipment is supplied with a three-conductor ac power cable. The power cable must be plugged into an approved three-contact electrical outlet. The power jack and mating plug of the power cable meet International Electrotechnical Commission (IEC) safety standards.

Do Not Operate in an Explosive Atmosphere.

Do not operate the equipment in the presence of flammable gases or fumes. Operation of any electrical equipment in such an environment constitutes a definite safety hazard.

Keep Away From Live Circuits.

Operating personnel must not remove equipment covers. Only Factory Authorized Service Personnel or other qualified maintenance personnel may remove equipment covers for internal subassembly or component replacement or any internal adjustment. Do not replace components with power cable connected. Under certain conditions, dangerous voltages may exist even with the power cable removed. To avoid injuries, always disconnect power and discharge circuits before touching them.

Do Not Service or Adjust Alone.

Do not attempt internal service or adjustment unless another person capable of rendering first aid and resuscitation is present.

Use Caution When Exposing or Handling the CRT.

Breakage of the Cathode-Ray Tube (CRT) causes a high-velocity scattering of glass fragments (implosion). To prevent CRT implosion, avoid rough handling or jarring of the equipment. Handling of the CRT should be done only by qualified maintenance personnel using approved safety mask and gloves.

Do Not Substitute Parts or Modify Equipment.

Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification of the equipment. Contact your local Motorola representative for service and repair to ensure that safety features are maintained.

Dangerous Procedure Warnings.

Warnings, such as the example below, precede potentially dangerous procedures throughout this manual. Instructions contained in the warnings must be followed. You should also employ all other safety precautions which you deem necessary for the operation of the equipment in your operating environment.

WARNING

Dangerous voltages, capable of causing death, are present in this equipment. Use extreme caution when handling, testing, and adjusting.

The computer programs stored in the Read Only Memory of this device contain material copyrighted by Motorola Inc., first published 1991, and may be used only under license such as the License for Computer Programs (Article 14) contained in Motorola's Terms and Conditions of Sale, Rev. 1/79.

All Motorola PWBs (printed wiring boards) are manufactured by UL-recognized manufacturers, with a flammability rating of 94V-0.

WARNING

This equipment generates, uses, and can radiate electro-magnetic energy. It may cause or be susceptible to electro-magnetic interference (EMI) if not installed and used in a cabinet with adequate EMI protection.

CE European Notice: Board products with the CE marking comply with the EMC Directive (89/336/EEC). Compliance with this directive implies conformity to the following European Norms:

EN55022 (CISPR 22) Radio Frequency Interference
EN50082-1 (IEC801-2, IEC801-3, IEEC801-4) Electromagnetic Immunity

The product also fulfills EN60950 (product safety) which is essentially the requirement for the Low Voltage Directive (73/23/EEC).

This board product was tested in a representative system to show compliance with the above mentioned requirements. A proper installation in a CE-marked system will maintain the required EMC/safety performance.

Contents

List of Figures

List of Tables

General Information 1

Introduction

This manual describes the following Ultra family of PowerPC-based low profile form-factor motherboards: Ultra Plus 603e, Ultra Plus 604, Ultra 603, Ultra 603e, and Ultra 604.

Unless otherwise specified, the Ultra Plus 603e and Ultra 604 motherboard platforms are referred to throughout this manual as *Ultra Plus*. The Ultra 603, Ultra 603e, and Ultra 604 motherboard platforms are referred to as the *Ultra 60x*. The Ultra Plus and the Ultra 60*x* are collectively referred to as the Ultra *system board*.

The Ultra system board platform is an all-in-one motherboard implemented as a 9-inch by 13-inch single-plane printed circuit board. The system board allows for either an MPC603, MPC603e, or MPC604 RISC processor for the MPU (factory installed option) and an MPC105 PowerPC-to-PCI bridge as the memory controller between the processor bus (MPU) and the Peripheral Component Interconnect (PCI) bus. An Intel i82378ZB PCI-to-ISA bridge component (referred to as the PIB) allows Industry Standard Architecture (ISA) bus-compatible peripherals to be accessed by the MPC60*x* processor. The system board also provides for the addition of a riser card which supports PCI bus and/or ISA bus cards.

Some of the standard PC (personal computer) interconnections that are included on the system board are two asynchronous serial ports, a parallel port, an IDE controller and connector, a floppy disk controller and connector, a keyboard, and a mouse.

In addition to the standard PC interconnections, the system board also provides a SCSI-2 PCI controller with standard connector, an Ethernet PCI controller with 10base-T RJ-45 connector and AUI DB15 connector, a Graphics PCI controller with SVGA connector, as well as business audio input/output capabilities, all on the primary motherboard.

The system board is designed to support Microsoft's Windows NT and IBM's AIX operating systems.

The system board platform can be obtained as either a motherboard only, a motherboard integrated in a chassis, or a full computer system with motherboard, chassis, and peripherals.

Features

The following table describes the major features of the system board:

Table 1-1. Ultra Plus and Ultra 60*x* Features

Feature	Description	
Microprocessor	Ultra Plus 603e	MPC603e (@100 MHz)
	Ultra Plus 604	MPC604 (@ 100 MHz or 133 MHz) RISC
	Ultra 603	MPC603 (@ 66 MHz)
	Ultra 603e	MPC603e (@100 MHz)
	Ultra 604	MPC604 (@ 100 MHz or 133 MHz) RISC
Bridge/memory controller	MPC105 Eagle PowerPC-to-PCI bridge memory controller	
Memory bus	64-bit + 8-bits parity, 66MHz local memory bus	
Cache	Second-level cache/Processor Direct Slot (PDS)	
Platform	PowerPC Reference Platform (PRP) specification enabled	
DRAM	Four 72-pin SIMM sockets for 8 to 256 MB of DRAM (with or without parity)	
I/O	Ethernet (AUI, 10base-T) interface onboard	
	SCSI-2 interface onboard	Fast (Ultra 60*x*)
		Fast/Wide (Ultra Plus)
	Dual PC16550A asynchronous serial ports	
	IEEE 1284 8-bit bidirectional parallel port	
	Low/high-density floppy disk controller	
	IDE hard-disk drive controller	
	VGA-compatible 8-bit color graphics system and SVGA, with onboard memory-expansion capability	
	Stereo 16-bit 44kHz business-class audio system	
	Audio line I/O, internal CD-ROM line inputs, headphone outputs	
	PS/2 keyboard and mouse interface	
	Power management of all onboard peripherals	

Table 1-1. Ultra Plus and Ultra 60*x* Features (Continued)

Feature	Description	
Expansion slots	PCI and ISA slot for either of two optional riser cards with PCI/ISA slots utilized as follows:	
	Six-slot card	Five slots usable: Three PCI slots and two ISA slots, or two PCI slots and three ISA slots
	Four-slot card	Three slots usable: Two PCI slots and one ISA slot, or one PCI slot and two ISA slots
Indicators and switches	Onboard status LEDs and switches	
	Provision for external connections	
Software compatibility	Windows NT readiness	
	AIX 4.1 readiness	
	Extensive diagnostics included	

Specifications

The specifications for the system board are listed in the following table.

Table 1-2. System Board Specifications

Characteristics	Specifications
Power Requirements (Exclusive of any attached peripheral devices)	
MPC603 processor @ 66 MHz or MPC603e processor @ 100 MHz	Maximum: 31 watts +5 Vdc (± 5%), 3 A (typical), 5 A (maximum) +12 Vdc (± 10%), 250 mA (typical), 500 mA (maximum) -12 Vdc (± 10%), 5 mA (typical), 10 mA maximum) -5 Vdc (± 5%), < 5 mA (maximum)

Table 1-2. System Board Specifications (Continued)

Characteristics		Specifications
MPC604 processor @ 100 MHz or MPC604 processor @ 133 MHz		Maximum: 43.5 watts +5 Vdc (± 5%), 5.5 A (typical), 7.5 A (maximum) +12 Vdc (± 10%), 250 mA (typical), 500 mA (maximum) -12 Vdc (± 10%), 5 mA (typical), 10 mA maximum) -5 Vdc (± 5%), < 5 mA (maximum)
Environmental		
Temperature	Operating	0˚ C to 50 ˚ C at point of entry of forced air cooling
	Non-operating	-40˚ C to 85˚ C
Altitude	Operating	5,000 meters (16,405 feet)
	Non-operating	15,000 meters (49,215 feet)
Relative humidity	Operating	10% to 80% (non-condensing)
	Non-operating	10% to 90% (non-condensing)
Physical Dimensions		
Motherboard	Height	1.375 inches (338.20 mm) (with connectors and slots only)
		3.50 inches (88.90 mm) (with connectors, slots, and add-in riser card and/or cache card)
	Length	13 inches (330.0 mm)
	Width	9 inches (228.6 mm)
	Thickness	0.093 inch (2.36 mm)
Motherboard Rear Panel I/O Connectors		
Ethernet interface	AUI	A 2-row, 15-pin, female DB-15 connector provides the AUI or 10base-2/10base-5 interface.
	10base-T	An RJ-45 connector provides the 10base-T interface.
SCSI-2 interface		A 50-pin header connector provides the Fast SCSI interface.
		A 68-pin header connector provides the Fast/Wide SCSI interface.
Serial port interface		Two 9-pin connectors provide the EIA-232-D compliant serial port connection.

Table 1-2. System Board Specifications (Continued)

Characteristics		Specifications
Parallel port interface		A 25-pin connector provides the standard IEEE 1284 parallel port connection.
Mouse interface		A circular 6-pin mini-DIN connector provides the mouse interface with existing PS/2-type peripheral devices.
Keyboard interface		A circular 6-pin mini-DIN connector provides the keyboard interface with existing PS/2-type peripheral devices.
Graphics interface		A 3-row, 15-pin, female DB-15 connector provides the VGA-compatible graphics interface.
Audio interface	Microphone	A mini-stereo jack connector provides the external sound capture interface.
	Audio output	A mini-stereo jack connector provides the audio transfer to external recording devices, headphones, or amplifier speakers.
	Audio input	A mini-stereo jack connector provides the external sound capture from playback devices interface.

Cooling Requirements

The system board is designed and tested to operate reliably with an incoming air temperature range from 0˚ to 50˚ C (32˚ to 122˚ F) with forced air cooling at a velocity typically achievable by using a 30 CFM (Cubic Feet/Minute) axial fan.

Temperature qualification is performed using a standard PC enclosure, with the system board unit under test simulated in a high power system configuration.

Test software is executed as the system board is subjected to ambient temperature variations. Case temperatures of critical, high power density integrated circuits are monitored to ensure component vendors' specifications are not exceeded.

While the exact amount of airflow required for cooling depends on the ambient air temperature and the type, number, and location of components, and other heat sources; adequate cooling for the motherboard low power components is achieved with an airflow of 100 LFM (Linear Feet/Minute).

Adequate cooling for the critical components (the microprocessors) requires an airflow of 200 LFM for the MPC603/MPC603e processor with no heat sink, and an airflow of 250 LFM for the MPC604 processor with the current heat sink.

Regulatory Compliance

The system board is a board level product and will be used in several markets, including OEM sales for using in industry-standard chassis. For those chassis not manufactured by Motorola, it is the responsibility of the OEM to meet the regulatory guidelines as determined by intended application. All external I/O connectors are shielded to aid in meeting EMI emissions standards.

The system board will be tested in a MCG chassis for EMI evaluation.

FCC Compliance

The system board is intended for use in systems meeting the following EMI/RFI regulations:

US	FCC Class B
Canada	DOC Class B
Europe	VDE Class B, CISPR-B

Equipment Required

The following equipment is required to make a complete system:

2	4 MB SIMMs (or more; at least 16 MB is recommended for NT)
1	Boot Flash device
1	SCSI and/or IDE hard disk and/or floppy disk
1	PC-type power supply or equivalent, with +5 Vdc and +12Vdc, and adequate power rating to support your application

Inter-Operability

The system board will inter-operate with the following boards.

PCI-compatible cards

ISA-compatible cards

Performance

The performance of the system board is based on the MPC60*x* RISC processor, the MPC105 PowerPC-to-PCI bridge/memory controller, the speed of the installed memory devices, and the associated peripheral devices on the PCI (local) bus and the ISA bus.

Caution

If the system board is manufactured with all onboard PCI peripherals installed (graphics, SCSI, and Ethernet), then only **two** PCI slots may be used. The third slot must remain empty or be used with an ISA card. Otherwise, erratic performance on the PCI bus may occur (due to PCI bus overloading).

Preparing and Installing the Ultra System Board

2

Introduction

The Ultra system board is primarily an all-in-one motherboard implemented on a 9-inch by 13-inch single-plane printed circuit board. The system board provides for the installation of several user-changeable components to enhance system features. In addition, the system board provides for the installation of several types of internal support or "daughter" boards/cards, such as a riser card that supports PCI bus and/or ISA bus cards and a secondary cache card that supports additional cache memory.

Before You Begin

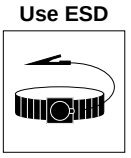
Before making any attempt to install your system board or replace or insert any add-on modules, cards, or components, make sure to turn all equipment power **OFF** and disconnect the power cable from the power source.

Warning

Dangerous voltages, capable of causing extreme injury or death, are present in this equipment. Use extreme caution when handling, testing, or adjusting modules, add-on cards, or components on the system board.

Caution

Inserting or removing modules, add-on cards, or system components while power is applied could result in damage to other system board components.



Motorola strongly recommends that you use an antistatic wrist strap and a conductive foam pad when installing or upgrading the system. Electronic components, such as disk drives, computer boards, and memory modules, can be extremely sensitive to ESD. After removing the component from the system or its protective wrapper, place the component flat on a grounded, static-free surface, and in the case of a board, component-side up. Do not slide the component over any surface.

If an ESD station is not available, you can avoid damage resulting from ESD by wearing an antistatic wrist strap (available at electronics stores) that is attached to an unpainted metal part of the system chassis.

Overview of Start-Up Procedures

The following table lists the things you will need to do before you can use this board, and tells where to find the information you need to perform each step. The order in which you perform these steps may vary depending upon your equipment. Be sure to read this entire chapter and read all Caution and Warning notes before beginning.

Table 2-1. Start-Up Overview

What you need to do ...	Refer to ...	On page ...
Unpack the hardware.	***Unpacking the Hardware***	2-5
Install optional SIMMs, if any.	The documentation furnished with the SIMMs	
	SIMM Memory Modules	2-7
Replace video DRAMs, if you do not intend to use those provided.	***Video DRAM Devices***	2-8
Replace Flash memory devices, if you do not intend to use those provided.	***Flash Memory Devices***	2-9
Install optional riser cards, if any.	The documentation furnished with the riser cards	
	Riser Card	2-12
Install optional cache card, if any.	The documentation furnished with the cache card	
	Cache Card	2-17
Install the Ultra system board in your chassis/enclosure	The documentation furnished with your chassis/enclosure	
	Cooling Considerations	2-20
Connect a console terminal.	The documentation furnished with your console terminal and with your chassis/enclosure	
	After Installing the Ultra System Board	2-19
	Display Device	5-5

Table 2-1. Start-Up Overview (Continued)

What you need to do ...	Refer to ...	On page ...
Connect any other optional devices or equipment you will be using.	The documentation provided with your optional equipment	
	Interconnect Signals	B-1
Power up the system.	The documentation furnished with your enclosure	
	After Installing the Ultra System Board	2-19
	LED Indicators	3-2
	Performing Diagnostic Tests.	5-20
	You may also wish to obtain the ***PPC1Bug Diagnostics Manual***, listed in ***Ordering Related Documentation***.	A-1
Ensure that the RTC is running, and initialize it if necessary.	***After Installing the Ultra System Board***	2-19
	Battery Backed NVRAM Devices	2-11
	PPCBug Debugger Command Set	6-8
	You may also wish to obtain the ***PPCBug Firmware Package User's Manual***, listed in ***Ordering Related Documentation***.	A-1
Change any environmental parameters as needed for your configuration.	***After Installing the Ultra System Board***	2-19
	Modifying Parameters in NVRAM	7-1
	Configuring the PPCBug Parameters	7-3
	You may also wish to obtain the ***PPCBug Firmware Package User's Manual***, listed in ***Ordering Related Documentation***.	A-1
Program the Ultra Plus or Ultra 60*x* as needed for your applications.	***After Installing the Ultra System Board***	2-19
	You may also wish to obtain the ***Ultra 603, Ultra 603e, and Ultra 604 Programmer's Reference Guide***, listed in ***Ordering Related Documentation***.	A-1

Unpacking the Hardware

Note If the shipping carton is damaged upon receipt, request that the carrier's agent be present during the unpacking and inspection of the equipment.

Unpack the equipment from the shipping carton. Refer to the packing list and verify that all items are present. Save the packing material for storing and reshipping of equipment.

Caution Avoid touching areas of integrated circuitry; static discharge can damage circuits.

Field Replaceable Units

The system board contains six types of user- or field-replaceable/installable units. These devices range from multiple-pin IC (integrated circuit) chips to more complex units such as component modules and boards/cards. These units are:

- SIMM memory modules (two or four)
- Graphics expansion video DRAM devices (two, optional)
- Flash memory devices (two, socketed)
- Battery-backed NVRAM battery/crystal housing unit (one)
- PCI/ISA riser card (one, several variations available)
- Cache memory card (one, 256 KB or 512 KB options)

Refer to Figure 2-1 for the locations of the above components on the system board.

Note The cache card and the riser card are basically internal "daughter" boards or cards. More detailed information on each unit and the installation of such is given in the unit specific section under *Optional Internal Board Units* of this chapter.

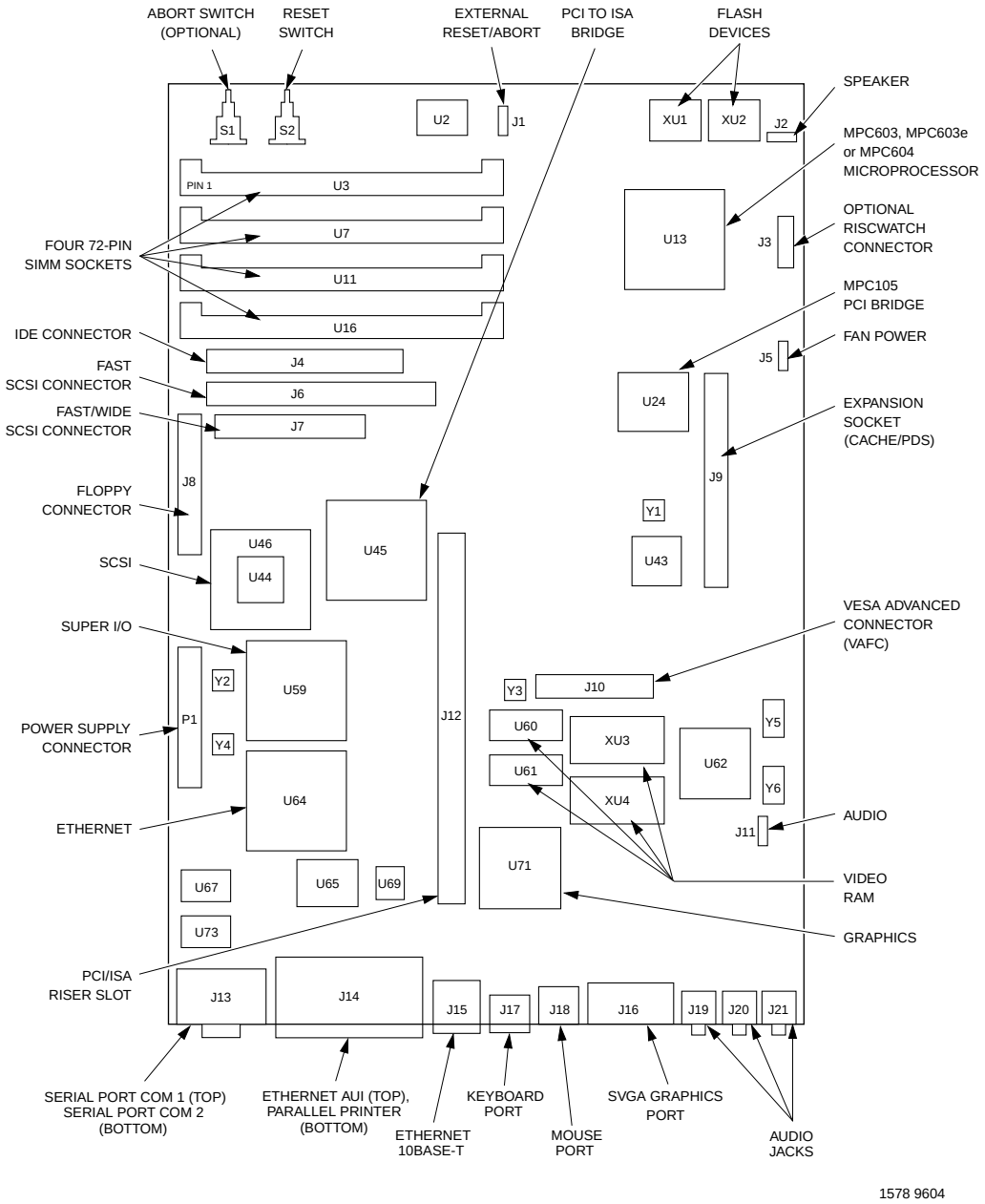


Figure 2-1. User/Field Replaceable Components Location Diagram

SIMM Memory Modules

The system board provides four tin-plated SIMM slots, labeled U3, U7, U11, and U16. The minimum number of SIMM modules required for operation is two. The total number of SIMM modules that the system board can hold is four. For proper operation, SIMM modules must be installed in pairs.

Note The system board uses SIMM (Single In-Line Memory Module) technology for implementing DRAM memory. The system board can only be populated with either 32-bit or 36-bit (with parity) 72-pin SIMM memory modules.

Installing a SIMM

1. Turn the power off and disconnect the power cable from the power source.
2. All SIMMs, regardless of memory size, have a notch on one side of the connector finger section. This side notch denotes connector pin 1 (refer to Figure 2-2). Pin 1 on each SIMM slot is located toward the edge of the motherboard.

   With the motherboard in the horizontal position, align pin 1 on the SIMM with pin 1 on the slot and carefully slide the SIMM module into the slot, noting that the SIMM module will be at a slight angle as it enters and sits in the slot. Be sure the SIMM module is seated properly.

   Firmly press the SIMM memory module backwards until the left and right lock pins on the slot move outwards and then snap back inwards.

   The SIMM is now securely in place.
3. Reconnect the power cable to the power source and turn the power on.

Removing a SIMM

1. Turn the power off and disconnect the power cable from the power source.
2. Simultaneously push the left and right lock pins on the slot outwards. The SIMM module releases and snaps forward. Pull the SIMM module out.
3. Reconnect the power cable to the power source and turn the power on.

For additional information on SIMMs, refer to the *Memory Controller* section in the *Functional Description* chapter.

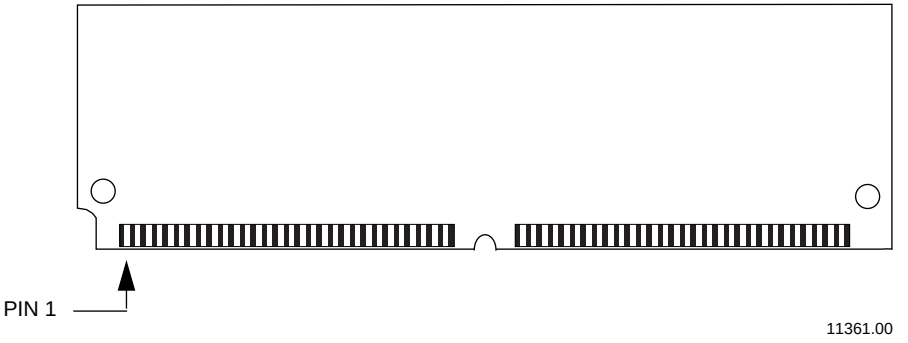


Figure 2-2. SIMM Memory Module Outline

Video DRAM Devices

The system board provides two sockets, labeled XU3 and XU4, for the installation of additional video DRAM devices.

The system board is manufactured with 1 MB of video DRAM and can display graphics at a resolution of 1024 x 768 at 256 colors. The system board can also display graphics at a higher resolution of 1280 x 1024 at 256 colors with the addition of another 1 MB of video DRAM.

Caution

The DRAM component is a 40-pin integrated circuit chip; care should be taken not to damage the device pins.

Installing a Video DRAM Device

1. Turn the power off and disconnect the power cable from the power source.
2. Carefully align pin 1 on the video DRAM chip with pin 1 on the 40-pin socket (refer to Figure 2-1 for location of the socket). Firmly press the video DRAM chip downward. The video DRAM chip should descend into the socket to be securely in place. Repeat the procedure for the other video DRAM chip.
3. Reconnect the power cable to the power source and turn the power on.

For more information on the video DRAM, refer to the *Graphics Controller* section in the *Functional Description* chapter.

Removing a Video DRAM Device

1. Turn the power off and disconnect the power cable from the power source.
2. Using an extractor tool appropriate to the 40-pin device, carefully extract the video DRAM chip from the socket.
3. Reconnect the power cable to the power source and turn the power on.

Flash Memory Devices

The system board provides two sockets, labeled XU1 and XU2, for the installation of Flash devices containing up to 8 megabits of Flash. If only one device is to be installed, socket XU1 should be used. Refer to Figure 2-1 for the location of the Flash sockets.

Caution

As supplied, the two sockets hold Flash devices that contain bootstrap firmware. If these devices are removed or corrupted, your system will not boot. Replacement Flash devices must perform board initialization and boot capability.

The Flash devices also contain the PPCBug debugger firmware. If removed or corrupted, you will not be able to use the debugger command set.

Caution

The Flash component is a 32-pin integrated circuit chip; care should be taken not to damage the device pins.

Installing a Flash Device

1. Turn the power off and disconnect the power cable from the power source.
2. Carefully align pin 1 on the Flash device with pin 1 on the 32-pin socket. Firmly press the Flash device downward. The Flash device should descend into the socket to be securely installed.
3. Reconnect the power cable to the power source and turn the power on.

Removing a Flash Device

1. Turn the power off and disconnect the power cable from the power source.
2. Using a PLCC extractor tool appropriate to the 32-pin device, carefully extract the Flash device from the socket.
3. Reconnect the power cable to the power source and turn the power on.

Refer to the *Flash Device* section in the *Functional Description* chapter for more details on the Flash memory and compatible Flash devices

Battery Backed NVRAM Devices

At least 8KB of non-volatile RAM memory is provided by the Ultra system board for data retention. Installation of this memory storage is done in the factory because of the surface mounting nature of the devices.

The real-time clock (RTC) portion of the NVRAM is part of a SNAPHAT housing (battery and crystal) which is replaceable. This housing mounts on top of the NVRAM and provides 10 years of clock operation in the absence of power. Currently, this housing is installed in the factory and is designed to be removed by service technicians due to the tooling required to remove the device.

On some boards, the RTC may have been put in PowerSave mode at the factory before shipping. You use PPCBug's **SET** command can to see if the clock is running, and start it if it is not. The clock must be running before you can use the Ultra system board. Refer to the *PPCBug Debugger Command Set* in Chapter 6.

Riser and Cache Cards

For information on the installation of the PCI/ISA riser card and cache card, refer to the sections *Riser Card* and *Cache Card* in the *Optional Internal Board Units* section below.

Optional Internal Board Units

The system board may contain one or more internal “daughter” boards/cards, to implement PCI and ISA slots and to add cache memory.

Riser Card

The optional riser card is inserted into the motherboard’s riser card slot, labeled J12, and accepts the installation of third-party ISA or PCI bus cards. The riser cards currently available through Motorola include those listed in Table 2-1. They are pictured in Figures 2-3 and 2-4.

An IDSEL wired to each PCI slot determines the configuration address used to initialize the board upon power-up. Board interrupts are routed to the interrupt controller on the Ultra system board. Details on connections and interrupt routing for PCI slots are provided in Appendix B.

Table 2-2. Riser Card Types

Motorola Part Number	Total Slots	PCI Slots	ISA Slots	Total Usable Slots	Notes
MT-RISER	6	3	3	5	Of the two “middle” slots, you may use the PCI slot or the ISA slot, but not both.
DT-RISER	4	2	2	3	Of the two “middle” slots, you may use the PCI slot or the ISA slot, but not both.

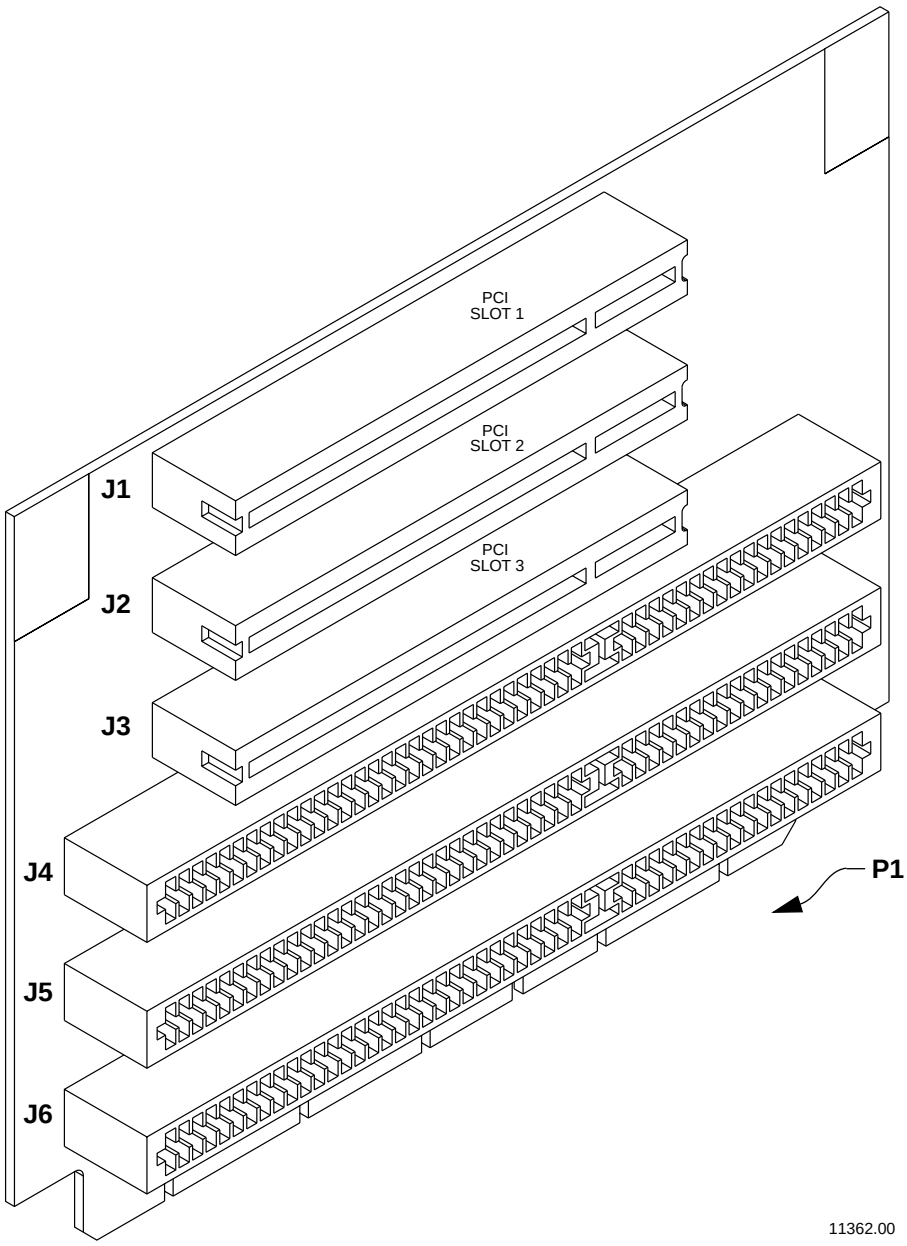


Figure 2-3. PCI/ISA Riser Card, 6-Slot

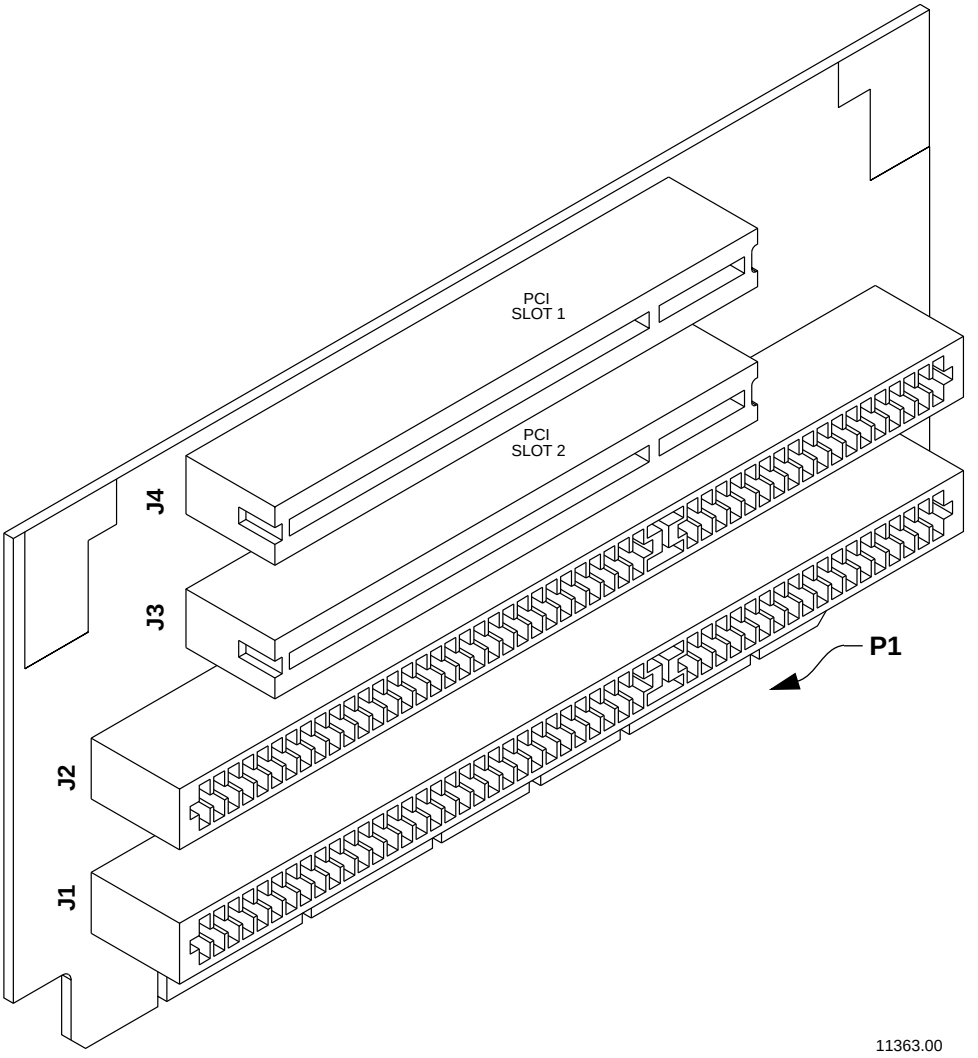


Figure 2-4. PCI/ISA Riser Card, 4-Slot

Although the PCI specification accommodates up to four interrupts per PCI card, it is common practice that only one interrupt (PCIINTA*) is implemented. To accommodate PCI cards that may make use of multiple interrupts, the motherboard provides support for up to three PCI interrupts per card (with software ramifications). Refer to Figure 2-5.

The actual mapping of the PCI interrupts is performed in software, via the I82378ZB PCI-to-ISA bridge chip. A suggested mapping is provided in Appendix B.

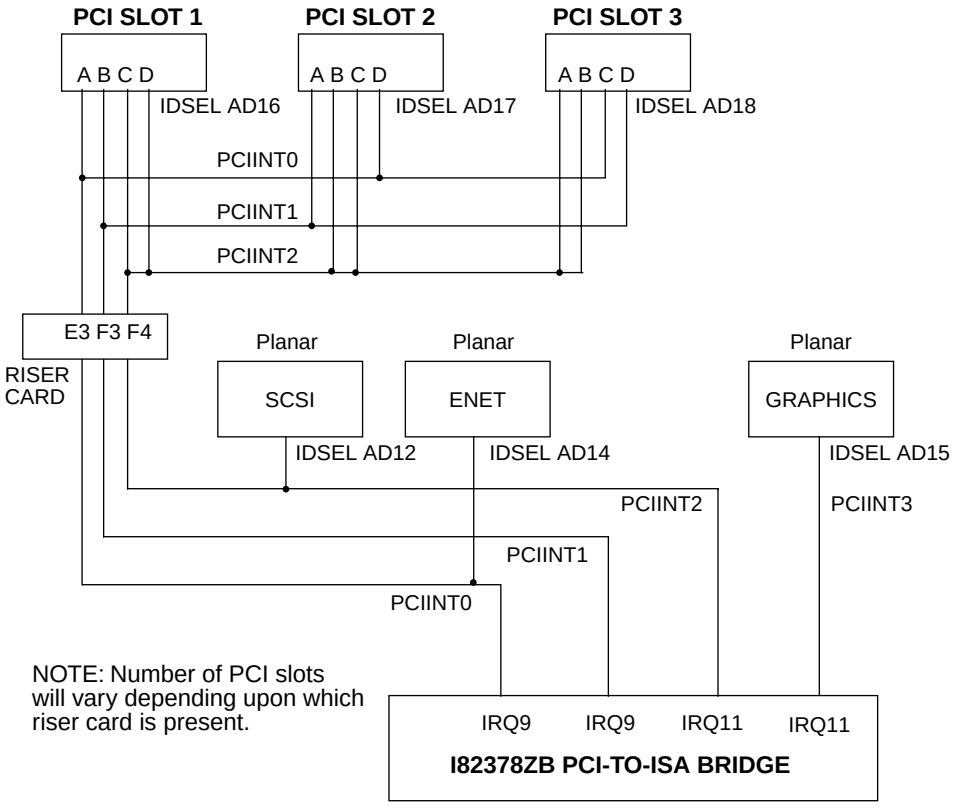


Figure 2-5. PCI Interrupt Routing

Installing and Removing a PCI or ISA Card

Caution

This procedure assumes that the riser card is not yet connected to the motherboard.

6-Slot Card: You may insert up to three PCI cards and two ISA cards, or up to two PCI cards and three ISA cards into the connectors on the 6-slot riser card. The PCI cards are inserted component-side down into slots J1, J2, and/or J3 on the riser card, and the ISA cards are inserted component-side up into slots J6, J5, and/or J4. Note that slots J3 and J4 cannot both be used at the same time.

Note If using all three PCI slots, refer to the CAUTION in Chapter 1 under *Performance.*

1. To install a PCI or ISA card, align pin 1 on the PCI or ISA card with pin 1 of the appropriate PCI or ISA slot on the riser card (refer to Figure 2-3 for location of the PCI and ISA slots on the riser card). Firmly press the PCI or ISA card securely into the connector slot.
2. To remove a PCI or ISA card, hold it on opposite ends with both hands and pull up on it while rocking it from side to side. Continue until the unit works loose.

4-Slot Card: You may insert up to two PCI cards and one ISA card, or one PCI card and up to two ISA cards into the connectors on the 4-slot riser card. The PCI cards are inserted component-side down into slots J4 and/or J3 on the riser card, and the ISA cards are inserted component-side up into slots J1 and/or J2 on the riser card. Note that slots J2 and J3 cannot both be used at the same time.

1. To install a PCI or ISA card, align pin 1 on the PCI or ISA card with pin 1 of the appropriate PCI or ISA slot on the riser card (refer to Figure 2-4 for location of the PCI and ISA slots). Firmly press the PCI or ISA card securely into the connector slot.
2. To remove a PCI or ISA card, hold it on opposite ends with both hands and pull up on it while rocking it from side to side. Continue until the unit works loose.

Installing a Riser Card

1. Turn the power off and disconnect the power cable from the power source.
2. Position the riser card's P1 connector correctly above the socket (refer to Figure 2-1 for location of the riser socket). Firmly press the riser card down into the connector slot. The connector finger section of the riser card should completely descend into the connector slot to be securely installed.
3. Reconnect the power cable to the power source and turn the power on.

Removing a Riser Card

1. Turn the power off and disconnect the power cable from the power source.
2. To remove the riser card, hold it on opposite ends with both hands and pull up on it while rocking it from side to side. Continue until the unit works loose.
3. Reconnect the power cable to the power source and turn the power on.

Cache Card

The system board may be enhanced with additional cache memory by using a cache board in the cache/PDS connector slot, labeled J9. The cache board is pictured in Figure 2-6.

The cache card may be populated (in the factory only) with various sizes of fast static RAM.

Installing a Cache Card

1. Turn the power off and disconnect the power cable from the power source.
2. Align pin 1 of connector J1 on the cache card with pin 1 of the socket (refer to Figure 2-1 for location of the cache/PDS

socket). Firmly press the cache card securely into the connector slot and install the four screws provided.

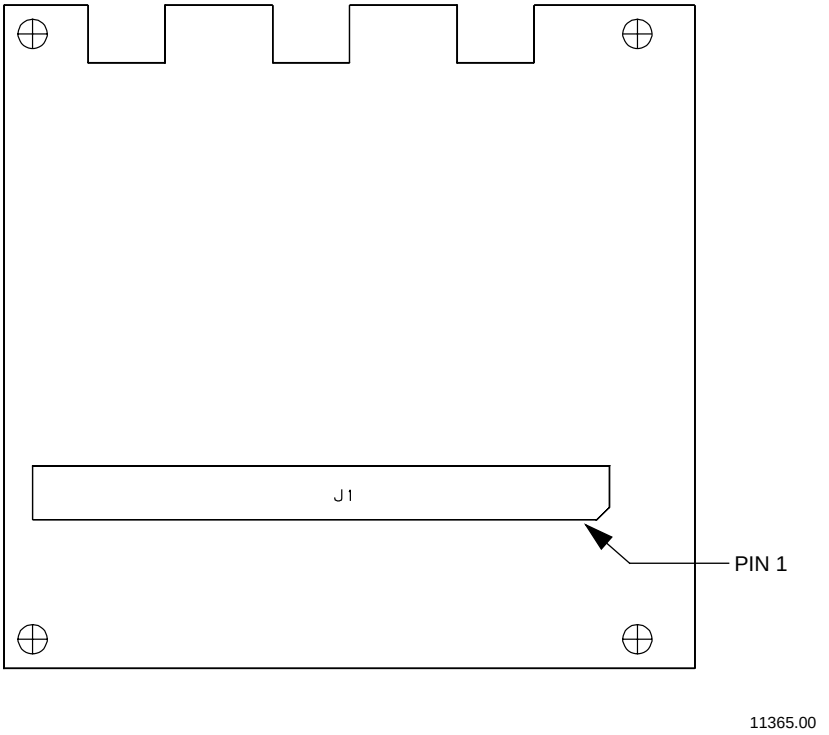


Figure 2-6. Cache Card

Removing a Cache Card

1. Turn the power off and disconnect the power cable from the power source.
2. Remove the four screws and hold the cache card on opposite ends with both hands. Pull up on it while rocking it from side to side. Continue until the unit works loose.
3. Reconnect the power cable to the power source and turn the power on.

After Installing the Ultra System Board

The following paragraphs discuss what you will need to do after you have installed the Ultra system board into a *chassis* or an *enclosure*, as instructed in the documentation furnished with your chassis/enclosure. It assumes that you have installed or connected any optional devices or cards on the Ultra system board, and that you are using appropriate antistatic equipment. It also assumes that the Flash devices containing the PPCBug firmware are installed on the Ultra stem board.

1. Make sure that the power to the system is turned ***off***.

1. If you intend to use PPCBug interactively, connect a video monitor and keyboard to the SVGA port on the Ultra system board, or connect a terminal to be used as the PPCBug system console to the COM1 port. If you are using COM1, set the console terminal up as follows:
   - Eight bits per character
   - One stop bit per character
   - Parity disabled (no parity)
   - Baud rate = 9600 baud (default baud rate of the port at power-up)

   Refer to *Display Device* in Chapter 5.

2. Replace the chassis panels and/or covers and connect cables to peripheral devices as appropriate

3. Reconnect the system to the AC or DC power source, and turn the equipment power on. The Ultra system board's PWR LED will light.

4. Note that a set of confidence tests is run, and the debugger prompt PPC1-Bug> appears.

Note If the debugger prompt does not appear, refer to *Board Failure* and *Performing Diagnostic Tests* in Chapter 5.

5. Use the **SET** command to verify that the RTC clock has been started. If not, set the date and time of day with the **SET** command. Refer to *PPCBug Debugger Command Set* in Chapter 6.
6. Use the **CNFG** and **ENV** commands to verify that the default configuration and environmental parameters are suitable and/or to customize your environment. Refer to *Auto Boot*, *ROMboot*, and *Network Auto Boot* in Chapter 5, and to Chapter 7, *Advanced Debugger Topics*.
7. Program the Ultra system board as desired. Refer to Appendix B, *Technical Data*, and to the *Ultra 603, Ultra 603e, and Ultra 604 Programmer's Reference Guide*.

Cooling Considerations

A 3-pin header, labeled J5, is provided on the Ultra system board for powering a dedicated fan to supply the forced air cooling required.

Caution

Position is important; the air must blow across the MPU.

Refer to the *Cooling Requirements* section in Chapter 1 for temperature qualification information.

Using the Controls and Indicators 3

Introduction

This chapter describes the control switches and indicators located on the Ultra system board. These devices are illustrated in the controls and indicators location diagram, Figure 3-1.

The system board may have one or two push-button switches (one is optional) and four LED (light-emitting diode) indicators. They are located adjacent to the SIMM sockets.

Control Switches

ABORT (S1)

When you press the **ABORT** push-button switch, it generates an interrupt request (IRQ8) to the Super I/O device. This can then be handled by software, usually to abort program execution.

Note The **ABORT** push-button switch may be optional on your system board.

RESET (S2)

When you press the **RESET** push-button switch, it resets all devices on the system board.

LED Indicators

The four LED indicators located on the system board are: **PWR**, **DSK**, **ENET**, and **BEAT**.

PWR (DS1)

The green **PWR** LED is lit when +5V power is applied to the system board.

DSK (DS2)

The green **DSK** LED is lit when the BSY signal of the SCSI device is asserted or the IDE LED signal on the IDE connector is asserted by the IDE device.

ENET (DS3)

The yellow **ENET** LED is connected to the Ethernet device and indicates that Ethernet packets are being received when lit.

BEAT (DS4)

The yellow "heartbeat" **BEAT** LED is connected to the TS signal of the MCP60*x* processor and when lit indicates that the MPU is executing cycles on the MPU bus.

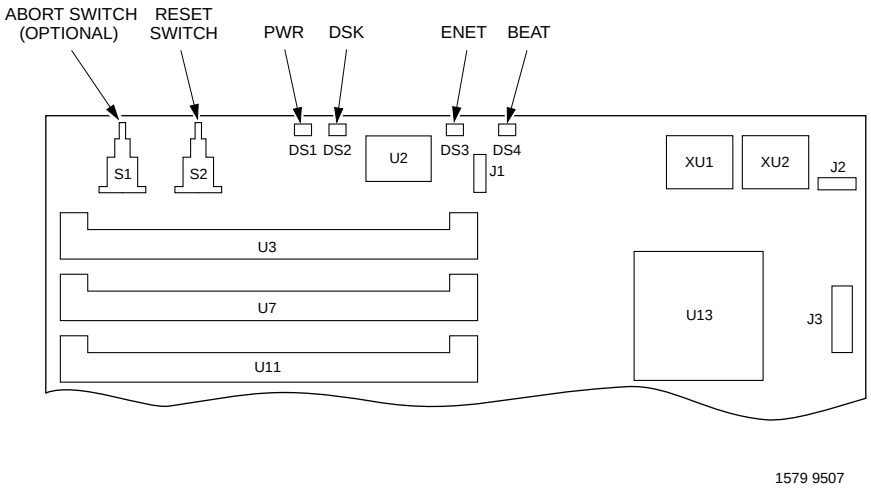


Figure 3-1. Controls and Indicators Location Diagram

Functional Description 4

System Board Description

This chapter provides a block diagram level description of the Ultra system board. The functional description provides an overview of each of the major sections, grouped according to their locations on the processor bus, the PCI (local) bus, and the ISA bus. The block diagram shown in Figure 4-1 illustrates the general architecture of the system board.

Processor Bus

The system board contains the processor, memory, and all the associated peripheral and I/O controllers.

Microprocessor

The system board directly supports the MPC603, MPC603e, and MPC604 processors at speeds up to 100 MHz or more, where those speeds are supported by both the installed MPC60*x* processor and the MPC105 memory controller.

The restrictions are that the CPU bus speed may not exceed 66 MHz and the PCI bus speed may not exceed 33 MHz.

The operating speeds of the system board are set in the factory, and cannot be changed by the end-user. The processor and the MPC105 Eagle chip are also surface-mounted and cannot be upgraded.

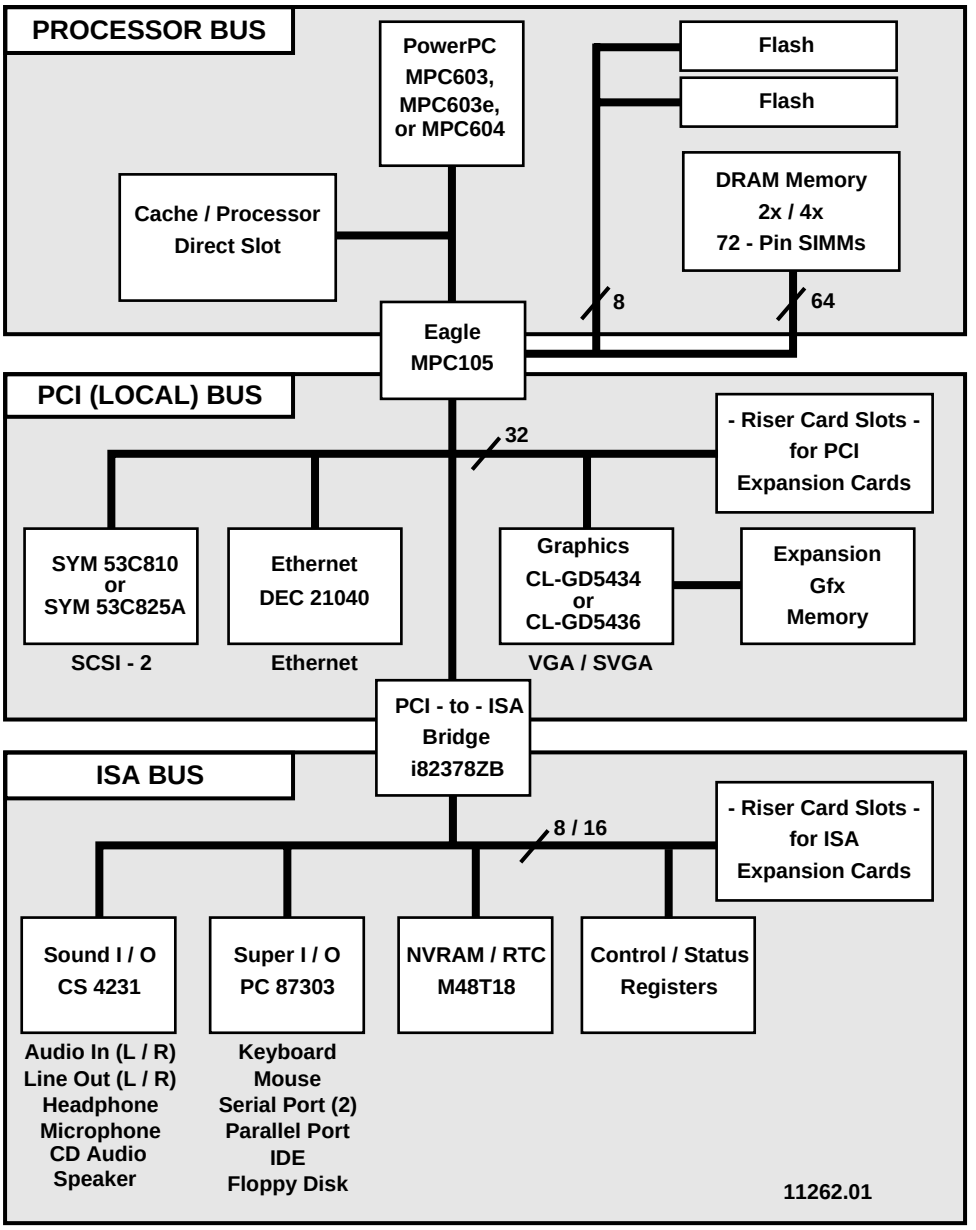


Figure 4-1. System Board Block Diagram

Cache Memory/ PDS Slot

The MPC60*x* bus signals and the MPC105 chip cache-control signals, plus a few additional signals, are connected to a 190-pin connector. This allows the system board to accommodate a secondary cache board.

This slot may also be used to analyze critical signals using a specialized card and test equipment.

Memory Controller

The system board uses the MPC105 PowerPC-to-PCI Bridge/Memory Controller chip to provide memory control, DRAM refresh, and memory decoding for banks of DRAM and/or Flash. The system board contains sockets or board locations to install any of the following memory devices:

- Dynamic RAM (72-pin SIMM DRAM modules)
- Flash device (electrically alterable)
- NVRAM (battery backed)

On the system board, the DRAM interface is connected to four 72-pin SIMM sockets. These may be populated with either 32-bit or 36-bit SIMM memory modules (the latter providing optionally-supported parity checking). This allows a maximum of 256 MB of memory on the system board. Table 4-1 shows how various sizes of memory may be installed in the system board using available SIMMs.

The MPC105 uses a 64-bit memory interface, so SIMMs must always be installed in multiples of two. Different sizes of SIMMs are allowed, as long as the pairs are of the same type, speed, and size. The system board senses the type and speed of installed SIMMs and can configure the MPC105 for the user-installed size and speed of memory automatically, via software.

Table 4-2 provides the MPC105 memory programming assignments for the memory banks on the Ultra system board.

Table 4-1. System Board Memory with Various SIMM Sizes

SIMM Technology	Quantity	System Memory Size
1M x 32/36	2	8MB
	4	16MB
2M x 32/36	2	16MB
	4	32MB
4M x 32/36	2	32MB
	4	64MB
8M x 32/36	2	64MB
	4	128MB
16M x 32/36	2	128MB
	4	256MB

Table 4-2 provides the MPC105 memory programming assignments for the memory banks on the system board.

Table 4-2. MPC105 Memory Programming Assignments

SIMM Bank	Socket	CAS#	RAS#	Data Bits
A	U3	0 - 3	0, 2	DH[0..31]
	U7	4 - 7	0, 2	DL[0..31]
B	U11	0 - 3	1, 3	DH[0..31]
	U16	4 - 7	1, 3	DL[0..31]

Note Both sockets of a bank must be populated.

Flash Devices

The system board provides two 32-pin sockets for Flash devices to provide up to 8 megabits of Flash storage. These devices may be used to store system boot-up software and Flash programming options needed for initial setup (before the Flash has been programmed). These devices may also be used to store HAL code, etc.

Note The MPC105 specification refers to ROM access using a special 32-bit mode (which requires four ROMs or wide parts). This mode is not used on the system board. Instead, the Flash access method is used to access byte-wide devices. The MPC105 handles the interface to these devices transparently so software is not affected.

The system board-compatible Flash device currently supported by Motorola is shown in the following table. These devices are supplied on the board and contain boot firmware. Refer to the cautions in Chapter 2 about modifying or removing these devices.

Table 4-3. Flash Device Compatibility

Type	Size	Organization	Example Part No.	Notes
Flash	4 Mbit	512K x 8	29F040	Only 5V-program parts are compatible with the 32-pin socket.

Under normal operation, the Flash devices are in “read-only” mode, their contents are pre-defined, and they are protected against inadvertent writes due to loss of power conditions. However, for programming purposes, programming voltage is always supplied to the devices and the Flash contents may be modified by executing the proper program command sequence. Refer to the third-party data sheet for further device-specific information and/or to the PFLASH PPCBug command.

Flash device speed is 150 ns. For this speed, software must not program ROMFAL (first access length) and ROMNAL (last access length) in the MPC105 device with values lower than the following minimum values for various processor external clock frequencies (hardware does not support the burst for which NAL is used):

Table 4-4. Minimum ROMFAL and ROMNAL Values

Processor External Bus Speed	ROMFAL Minimum Value	ROMNAL Minimum Value	8-Bit Access Times (Number of Clocks)	64-Bit Single/Burst Access Times (Number of Clocks)
25 MHz	1	1	4	32/32-32-32-32
33 MHz	2	2	5	40/40-40-40-40
40 MHz	3	3	6	48/48-48-48-48
50 Mhz	5	5	8	64/64-64-64-64
66 MHz	7	7	10	80/80-80-80-80

Refer also to the configuration parameters in Chapter 7.

Peripheral Component Interconnect (PCI) Bus

The system board implements a PCI bus operating at speeds up to 33 MHz. This bus is used for high-speed I/O transfers from onboard peripherals as well as devices present on the PCI bus located on the riser card. As the PCI is an un-terminated bus, careful attention is made to the layout of the PCI clocks. In particular, each PCI device receives a separate, skew-controlled clock line from the system clock generator.

The following sections describe the various peripheral devices present on the PCI bus.

PCI-to-ISA Bridge (PIB)

The PCI-to-ISA bridge component is the Intel i82378ZB (the "PIB"), which allows ISA-compatible peripherals to be accessed by the MPC60x processor. The PIB includes an 8-channel DMA controller for ISA devices, a timer, and a PCI/ISA interrupt controller.

All PCI devices use the PIB's ISA interrupt controller to prioritize interrupts into the system processor.

Small Computer Systems Interface (SCSI)

The Ultra Plus system boards implement a SCSI-2 (Fast/Wide) interface using the SYM 53C825A, and the Ultra 60*x* system boards implement a SCSI-2 (Fast) interface using the SYM 53C810. These parts provide a SCSI interface compatible with X2.131-1986 (SCSI-1) and with X3.131-1990 (SCSI-2).

The system board contains active SCSI terminators which may be used when no internal SCSI devices are present yet external ones are desired; or with internal devices only. The terminators may be disabled under software control to eliminate the need for end-users to experiment with termination options.

Ethernet

Ethernet access is provided through the Digital Equipment Corporation DEC21040 Ethernet controller. The system software determines its presence by accessing the Vendor and Device ID registers present in the device (as required for PCI devices).

This device provides a fully-compliant Ethernet and IEEE 802.3 networking interface. The rear connectors of the system board chassis provide direct connection to a 10base-T interface via an RJ-45 connector. Connections to external 10base-2 and 10base-5 transceivers may be made with a DB-15 AUI connector. The following diagram illustrates the Ethernet architecture.

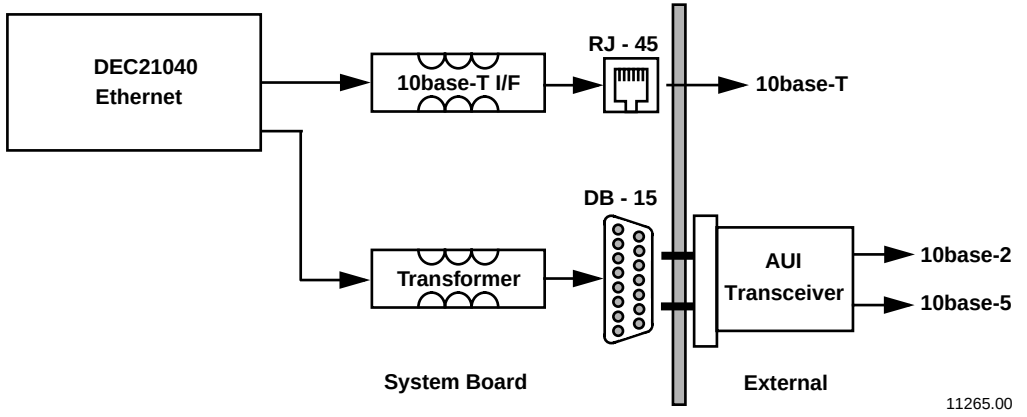


Figure 4-2. Ethernet Block Diagram

Graphics Controller

The system board contains a VGA-compatible Cirrus graphics controller (the CL-GD5434 or CL-GD5436) which is capable of displaying graphics at resolutions of up to 1280 x 1024 at 256 colors, at refresh rates up to 87 Hz. The graphics subsystem may be equipped with either 1 MB or 2 MB of video DRAM; the extra DRAM is only needed for higher resolutions and is not installed for low-cost options. The system board comes equipped with sockets to allow users to install the extra memory when needed.

The GD5434 and GD5436 are programmable and can provide literally hundreds of different resolutions. The following table provides some of the more useful options.

Other features include:

- Internal power-save control
- VESA monitor power-save controls
- Expansion/replacement
- BitBLT engine (for faster rendering)

Notes (1) The graphics hardware does not implement any byte-swapping hardware to resolve differences between "little-endian"operating systems (such as Windows NT) and "big-endian" operating systems (such as AIX).

(2) A VESA Advanced Feature Connector (VAFC) is provided for connection of auxiliary devices.

Table 4-5. System Board Graphics Capability

Width	Height	Colors	Refresh Rates (Hz)	Interlaced?	Additional Memory
640	480	16, 256, 32K	60, 75	No	None
640	480	16M	60	No	None
800	600	16, 256, 32K	60, 75	No	None
800	600	16M	60	No	None
1024	768	16, 256, 64K	60, 70	No	None
1024	768	16M	87	Yes	None
1280	1024	16, 256	60 Hz	No	1 MB
1280	1024	16, 256, 64K	87 Hz	Yes	1 MB

Industry Standard Architecture (ISA) Bus

The system board implements a standard PC-type ISA bus operating at speeds up to 8.33 MHz. This bus is used for inexpensive I/O controllers used on the system board as well as for allowing compatibility with the numerous ISA boards present in the PC market.

The following sections describe the various peripheral devices present on the ISA bus.

Super I/O (SIO) Controller

A large number of general-purpose I/O ports are provided through the National Semiconductor PC87303 "Super I/O" (SIO) controller, including:

- Serial ports (two)
- Parallel port
- IDE (hard disk) interface
- Floppy-disk interface
- Keyboard interface
- Mouse Interface

The SIO chip includes several power-down options, and includes an automatic power-down controller for the floppy disk controller. The following sections detail the capabilities of the SIO chip.

Serial Ports

The SIO chip provides two EIA-232-D compliant DTE serial ports, at baud rates from 50 to 19,200 baud. The rear panel of the system board provides access to these two ports through two 9-pin DB male connectors. The serial ports are software and hardware compatible with the PC16450 and the PC16550A (industry-standard serial controllers), including a 16-level FIFO buffer. These ports may also be used for Musical Instrument Digital Interface (MIDI) communications with the appropriate external buffers.

Parallel Port

The parallel port is compatible with IEEE standard 1284, including the Extended Capability Port (ECP) and Enhanced Parallel Port (EPP), as well as simple Centronics compatibility.

The connector is a 25-pin female DB connector. The printer port is protected against transient signals to allow for conditions where the printer is powered up before the system board.

Intelligent Device Expansion (IDE) Interface

A 40-pin male header is provided for the connection between the SIO chip and Intelligent Device Expansion (IDE) peripherals, such as hard disk or CD-ROM drives. The IDE interface conforms to the X3.221 revision 4.A AT Attachment specification (which covers the majority of IDE devices).

The IDE interface may be used with many types of low-cost disk drives, though only two IDE devices maximum are allowed on any IDE interface.

Floppy Disk Controller (FDC)

A low- and high-density MFM Floppy Disk Controller (FDC) is the standard FDC device on the system board. The controller is compatible with the industry-standard components previously used to implement floppy disk controllers: the DP8473, the 765A, and the N82077. Software designed to operate these devices may be used to control the FDC in the SIO chip without change.

The FDC may be used to support any of the following devices:

- 3.5" 1.44 MB floppy disk drive
- 5.25" 1.2 MB floppy disk drive
- Standard 250 kbps to 2 Mbps tape drive systems

A 34-pin male header provides the connection between the SIO chip and the floppy disk.

Keyboard/Mouse

Keyboard and mouse communication is provided by the PC87303. All electrical connections between the keyboard/mouse and the controller are filtered for RFI, as these devices are often a source of RFI. Additionally, the 5 volt power supplied to the mouse and keyboard are separately current-limited with a self-resetting fuse (also known as a "polyswitch").

The connections to the mouse and keyboard are provided through two PS/2 type 6-pin mini DIN connectors.

Audio Controller

The audio system is provided by the Crystal Semiconductor CS4231, a multimedia audio Coder/Decoder (CODEC). This device is capable of simultaneously recording and playing 16-bit stereo CD-quality audio samples. Both the recording and the playback subsystems feature a 16-level FIFO to allow the interrupt request rate to be minimized.

The CS4231 is compatible with the PowerPC Reference Platform (PRP) specification and with the Windows Sound System. The audio system of the system board has the following features:

- 16-bit stereo audio generation and capture
- MPC-compatible audio mixer
- Internal CD-ROM audio connection
- Full duplex stereo throughout

The general architecture of the audio system on the system board is shown in Figure 4-3.

Using the internal mixing and full duplex record/playback capabilities of the audio system, the following capabilities are possible, some simultaneously.

- Play CD-ROM to internal speaker, external speakers or headphones
- Record CD-ROM samples to disk
- Transfer human speech samples to system memory for recognition
- Play synthesized sounds/noises
- Play simple PC-type “beep” square waves on internal speaker

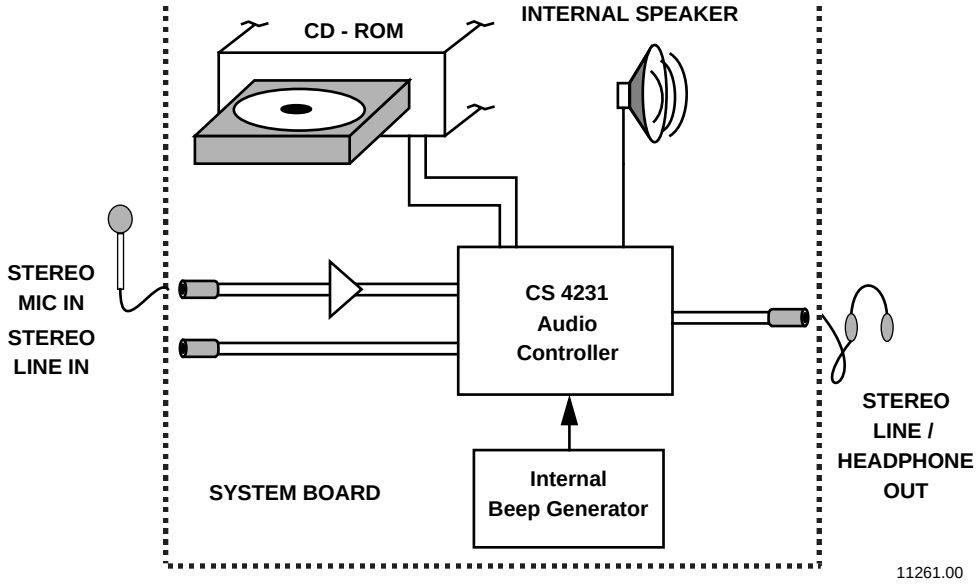


Figure 4-3. System Board Audio System Diagram

The "beep generator" is provided by the PCI-to-ISA bridge controller, and is a simple timer-based square-wave output. This is compatible with the simple sound present on PCs and may be used by system firmware without requiring much software support.

CD-ROM Audio Connection

The audio system contains two connectors which allow audio from CD-ROM drives to be amplified and driven through the audio amplifiers, allowing CD audio to be heard on the external headphone or speaker connectors. The CD-ROM audio connector is configured so that it is compatible with most popular CD-ROM audio connectors. Custom audio cables allow other types of CD-ROM drives to be used as well.

Non-Volatile RAM (NVRAM)

The system board provides 8KB of NVRAM storage for booting information, security passwords, etc. This is provided through the SGS-Thomson M48T18, a device that combines the RAM with a battery and a real-time clock. This memory is located on the 8-bit ISA bus through indirect access registers.

Real-Time Clock (RTC)

The real-time clock is a portion of the non-volatile RAM component. It provides a battery-backed clock which continuously stores and updates the current time, even with the system power removed.

PPCBug -- Getting Started 5

PPCBug Overview

The PPCBug firmware is the layer of software just above the hardware. The firmware provides the proper initialization for the devices on the PowerPC board or system upon power-up or reset.

This chapter describes the basics of PPCBug and the architecture up to the point where the boot is done, and gives details on the different types of boot that are available: ROMboot, Auto Boot, and Network Auto Boot. Auto Boot is the default test configuration in the firmware.

Chapter 6 describes the monitor (interactive command portion of the firmware) in more detail, and gives more details on actually using the PPCBug debugger and the special commands. Chapter 6 also contains a complete list of PPCBug commands.

Chapter 7 contains information about the CNFG and ENV commands, system calls, and other advanced user topics.

PPCBug Basics

The PowerPC debug firmware, PPCBug, is a powerful evaluation and debugging tool for systems built around the Motorola PowerPC microcomputers. Facilities are available for loading and executing user programs under complete operator control for system evaluation.

PPCBug provides a high degree of functionality, user friendliness, portability, and ease of maintenance

It achieves good portability and comprehensibility because it was written entirely in the C programming language, except where necessary to use assembler functions.

PPCBug includes commands for:

- Display and modification of memory
- Breakpoint and tracing capabilities
- A powerful assembler and disassembler useful for patching programs
- A self-test at power-up feature which verifies the integrity of the system

PPCBug consists of three parts:

- A command-driven, user-interactive *software debugger*, described in the *PPCBug Debugging Package User's Manual*. It is hereafter referred to as "the debugger" or "PPCBug".
- A command-driven *diagnostics package* for the Ultra hardware, hereafter referred to as "the diagnostics." The diagnostics package is described in the *PPC1Bug Diagnostics Manual.*
- A *user interface* or *debug/diagnostics monitor* that accepts commands from the system console terminal.

When using PPCBug, you operate out of either the *debugger directory* or the *diagnostic directory.*

- If you are in the debugger directory, the debugger prompt `PPC1-Bug>` is displayed and you have all of the debugger commands at your disposal.
- If you are in the diagnostic directory, the diagnostic prompt `PPC1-Diag>` is displayed and you have all of the diagnostic commands at your disposal as well as all of the debugger commands.

Because PPCBug is command-driven, it performs its various operations in response to user commands entered at the keyboard. When you enter a command, PPCBug executes the command and the prompt reappears. However, if you enter a command that

causes execution of user target code (e.g., **GO**), then control may or may not return to PPCBug, depending on the outcome of the user program.

PPCBug Implementation

PPCBug is written largely in the C programming language, providing benefits of portability and maintainability. Where necessary, assembler has been used in the form of separately compiled program modules containing only assembler code. No mixed language modules are used.

Physically, PPCBug is contained in two Flash devices, providing 1MB of storage. The executable code is checksummed at every power-on or reset firmware entry. The result is checked with a pre-calculated checksum contained in the last 16-bit word of the Flash image.

Comparison with Other MCG Debuggers

PPCBug is similar to previous Motorola Computer Group (MCG) firmware debugging packages such as MVME147Bug, MVME167Bug, and MVME187Bug, with differences due to microprocessor architectures. These differences are primarily reflected in the:

- Instruction mnemonics
- Register displays
- Addressing modes of the assembler/disassembler
- Argument passing to the system calls

Installation and Start-up

Power-Up/Reset Sequence

Figure 5-1 illustrates the basic flow the firmware follows at a power-up/reset sequence.

The default boot routine is Auto Boot. Using the **ENV** command (Chapter 7), you can select either ROMboot or Network Auto Boot as the boot mechanism.

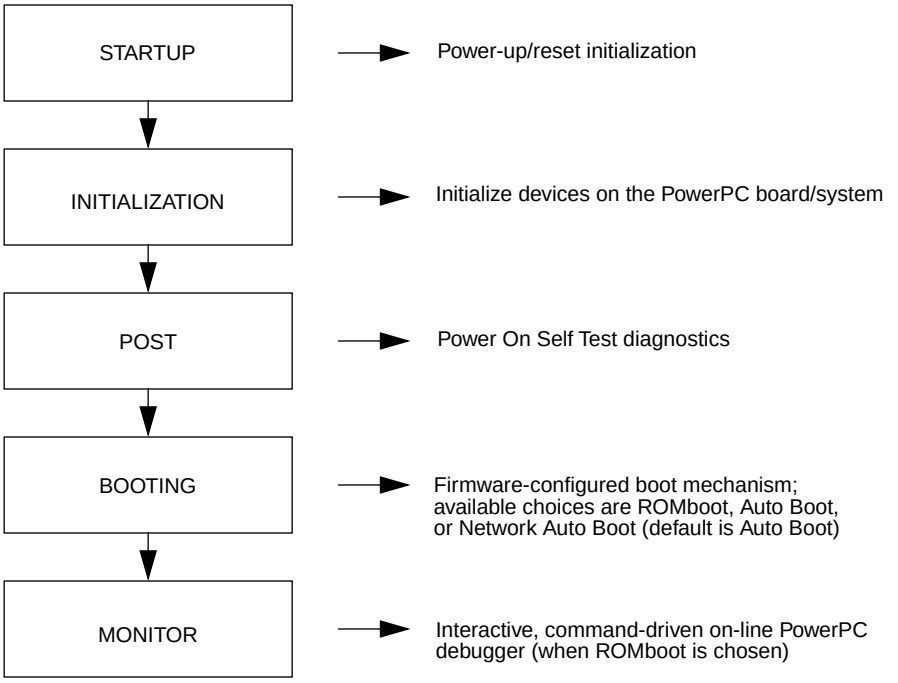


Figure 5-1. PowerPC Debugger Architecture

Debug Monitor

The debug "monitor" (interactive command level) is accessed after a ROMboot is done in the firmware. At the monitor, a number of different commands may be entered to interact with the hardware. Specifically, the commands are routed through character, disk, and network I/O drivers in the firmware (refer to Figure 5-2). This way the actual register settings and commands used by the devices are transparent to a firmware user. The firmware user only needs to be familiar with the basic PPCBug commands.

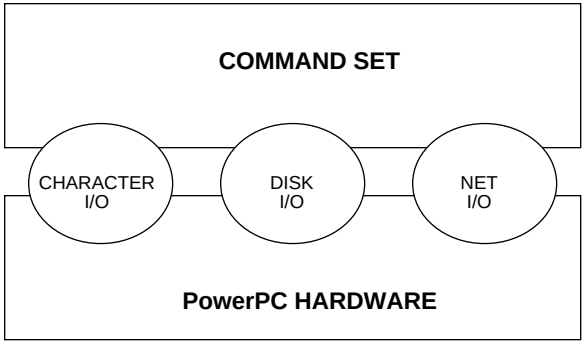


Figure 5-2. Debug Monitor

Display Device

The PPCBug firmware can be displayed either via the conventional mechanism of a console terminal connected to the serial port (COM1) or it can be displayed on a video monitor connected to the SVGA port.

The firmware checks for the presence of a connected keyboard, and for the presence of a connected mouse device. If either (or both) of these devices is connected to the system board and a firmware supported video monitor is found, the firmware is automatically brought up on the connected video monitor.

Table 5-1 illustrates how the display device is determined:

Table 5-1. Display Device Selection

Mouse Connected	Keyboard Connected	Firmware-Supported VGA Device Card Present	Firmware Displays On
Yes	Yes	Yes	VGA monitor
Yes	No	Yes	VGA monitor
No	Yes	Yes	VGA monitor
No	No	Yes	Serial port
No	No	No	Serial port
No	Yes	No	Serial port

Note In the case where the mouse is connected and the keyboard is not connected, the firmware is displayed on the video monitor. However, because a keyboard is required for interactive use on a video monitor, the firmware will display a "Keyboard not connected" message. In order to use the firmware, you must then plug in the keyboard.

If you are using a terminal other than a VGA as your console device, make sure it is configured as follows:

- 8 bits per character
- 1 stop bit per character
- Parity disabled (no parity)
- Baud rate = 9600 baud (default baud rate of the system board ports at power-up)

After power-up, the baud rate of the debug port can be reconfigured by using the Port Format (**PF**) command.

Refer to the *PPCBug Debugging Package User's Manual* for more detailed installation and start-up information.

Note In order for high baud-rate serial communication between PPCBug and the terminal to work properly, the terminal must do some form of handshaking. If the terminal being used does not do hardware handshaking via the CTS line, then it must do XON/XOFF handshaking. If you get garbled messages or missing characters, then you should check the terminal to insure XON/XOFF handshaking is enabled.

Auto Boot

Auto Boot is a PPCBug software routine that provides an independent mechanism for booting an operating system. It is the default boot mechanism. Auto Boot selects the boot device from either a scan list of device types or from a specific Controller Logical Unit Number (CLUN) and Device Logical Unit Number (DLUN).

Refer to Chapter 7 and to the *PPCBug Debugging Package User's Manual* for information on setting the **ENV** command parameters for enabling Auto Boot.

During a power-up/reset condition the firmware examines the validity of its configuration parameters held in NVRAM (i.e., the **ENV** parameters). If a configuration error (for example, corrupted data, checksum error) exists, the firmware initializes the configuration parameters with a set of default values. These values enable:

- System mode
- Execute SelfTest
- Auto Boot feature
- Auto Boot device type scan feature

Following the auto-initialization of the configuration parameters, the firmware resets the system to allow a start-up with the new default configuration parameters.

This auto-configuration and auto-boot process, by default, permits a console-less system boot. A console device does not need to be present.

The Auto Boot process attempts to boot from the following devices in the order listed:

1. FDISK Floppy Diskette
2. CDROM Compact-Disk ROM
3. TAPE Sequential-Access (i.e., streaming tape)
4. HDISK Direct-Access (i.e., hard disk)

You can use the **ENV** command to change this list if desired. The following list shows the **ENV** prompts associated with the Auto Boot feature:

```
Auto Boot Enable [Y/N]           = Y?
Auto Boot at power-up only [Y/N] = N?
Auto Boot Scan Enable      [Y/N] = Y?
Auto Boot Scan Device Type List  = FDISK/CDROM/TAPE/HDISK/?
Auto Boot Controller LUN         = 00?
Auto Boot Device LUN             = 00?
Auto Boot Partition Number       = 00?
Auto Boot Abort Delay            = 7?
Auto Boot Default String   [NULL for an empty string] = ?
```

The Auto Boot Scan feature may be disabled, thus yielding the previous behavior of the Auto Boot feature. The Scan feature does not use the CLUN/DLUN fields, but uses the device type list to search for a bootable device. This string is case sensitive: it must be in *uppercase*.

At power-up, Auto Boot is enabled. The following message is displayed on the system console:

```
Auto Boot about to begin...Press <ESC> to Bypass, <SPC> to Continue
```

Following this message, a delay allows you to abort the Auto Boot process and gain control. You can use these to abort Auto Boot:

- Escape
- Break
- Reset
- Abort

Break, Reset, and Abort are described later in this chapter.

If you do not abort Auto Boot, the actual I/O transfer begins. The program pointed to within the boot-record of the media specified loads into RAM, and control passes to it.

ROMboot

ROMboot is a PPCBug mechanism for booting an operating system from a user-defined routine stored in ROM. ROMboot executes at power-up or optionally at reset, if it is configured and enabled in parameters set with the **ENV** command (refer to Chapter 7). It may also be executed with the **RB** (ROMboot) command. **NORB** disables ROMboot.

For ROMboot to work, four requirements must be met:

- Power must have just been applied (or at reset, if configured to do so with the **ENV** command).
- Your ROMboot routine must be stored within the PowerPC board Flash memory map (or elsewhere in onboard memory, if configured to do so with the **ENV** command).
- The ASCII string "BOOT" must be located within the specified memory range.
- Your ROMboot routine must pass a checksum test, which ensures that this routine was really intended to receive control at power-up.

When the module is ready it can be loaded into RAM. The checksum can be generated, installed, and verified using the **CS** command.

The format of the beginning of the routine is as follows:

Offset	Length	Contents	Description
$00	4 bytes	BOOT	ASCII string indicating possible routine; the checksum must be valid
$04	4 bytes	Entry Address	Word offset from “BOOT”
$08	4 bytes	Routine Length	Word; includes length from “BOOT” to and including a two-byte checksum
$0C	Length of name	Routine name	ASCII string containing routine name

If you want to make use of ROMboot, you do not have to fill a complete EPROM. Any partial amount is acceptable, as long as:

- The identifier string “BOOT” starts on a word (EPROM and Direct spaces) or 8KB (local RAM space) boundary.
- The ROMboot routine size (in bytes) is evenly divisible by 2.
- The length parameter (offset $8) reflects where the checksum is, and the checksum is correct.

ROMboot searches predefined areas of the memory map for possible routines and checks for the “BOOT” indicator. Two events are of interest for any location being tested:

- The map is searched for the ASCII string “BOOT”.
- If the ASCII string “BOOT” is found, it is still undetermined whether the routine is meant to gain control at power-up or reset. To verify that this is the case, the bytes starting from “BOOT” through the end of the routine, excluding the two byte checksum, are run through the debugger checksum algorithm. If the result of the checksum is equal to the final two bytes of the ROMboot routine (the checksum), it is established that the routine was meant to be used for ROMboot.

Under control of the **ENV** command, the sequence of searches is as follows:

1. Search direct address for “BOOT”. The “direct address” is a variable that may be set using the **ENV** command and is provided expressly to point to an installed ROMboot routine, and thus eliminate the need to search further.
2. Search complete ROM map.
3. Search local RAM, at all 8KB boundaries starting at the beginning of local RAM.

Sample ROMboot Routine

The ROMboot routine example below performs the following:

- Outputs a **<CR><LF>** sequence to the default output port.
- Displays the date and time from the current cursor position.
- Outputs two more **<CR><LF>** sequences to the default output port.
- Returns control to PPCBug.

Perform the following steps to prepare the ROMboot routine (includes checksum calculation):

1. Assemble and link the code, leaving $00 in the even and odd locations destined to contain the checksum.
2. Load the routine into RAM (with S-records via the **LO** command, or from magnetic media using **IOP**).
3. Display entire ROMboot routine (checksum bytes are at $00010038 and $00010039).

```
PPC1-Bug>md 10000 :10
00010000  424F4F54 00000010 0000003A 54455354  BOOT.......:TEST
00010010  39400026 44000002 39400052 44000002  9@.&D...9@.RD...
00010020  39400026 44000002 39400026 44000002  9@.&D...9@.&D...
00010030  39400063 44000002 0000FFFF FFFFFFFF  9@.cD...........
```

4. Disassemble executable instructions.

```
PPC1-Bug>md 10010:5;di
00010010 39400026  SYSCALL     .PCRLF
00010018 39400052  SYSCALL     .RTC_DSP
00010020 39400026  SYSCALL     .PCRLF
00010028 39400026  SYSCALL     .PCRLF
00010028 39400063  SYSCALL     .RETURN
```

5. Perform checksum on locations $10000 through $10037 (refer to the **CS** command in Chapter 3).


```
PPC1-Bug>cs 10000:38/2;h
Effective address: 00010000
Effective count  : &56
Checksum: ACFA
```

6. Insert checksum into bytes $10038, $10039.

```
PPC1-Bug>m 10038;h
00010038 0000? acfa.
```

7. Display the entire ROMboot routine with checksums.

```
PPC1-Bug>md 10000 :10
00010000  424F4F54 00000010 0000003A 54455354  BOOT.......:TEST
00010010  39400026 44000002 39400026 44000002  9@.&D...9@.RD...
00010020  39400026 44000002 39400026 44000002  9@.&D...9@.&D...
00010030  39400063 44000002 ACFAFFFF FFFFFFFF  9@.cD...........
```

8. Verify the functionality of the user ROMboot routine with the **RB** command.

```
PPC1-Bug>rb;v
ROMboot about to Begin... Press <ESC> to Bypass, <SPC> to Continue
Direct Adr: FFC00000 FFFFFFFC: Searching for ROMboot Module at: 00010000
Executing ROMboot Module "TEST" at 00010000

MON MAR 27 10:39:08.00 1995

PPC1-Bug>
```

The ROMboot routine is now ready for use.

For complete details on how to use ROMboot, refer to the *PPCBug Debugging Package User's Manual*.

Network Auto Boot

Network Auto Boot is a PPCBug software routine that provides a mechanism for booting an operating system using a network (local Ethernet interface) as the boot device. This routine selects the boot device based on the Controller Logical Unit Number (CLUN) and Device Logical Unit Numbers (DLUN) which have been set with the **ENV** command.

At power-up, Network Auto Boot is enabled, if it is configured and enabled in parameters set with the **ENV** command, and the following message is displayed upon the system console:

```
Network Boot in progress... To abort hit <BREAK>
```

Following this message there is approximately a five-second delay before the actual I/O is begun. The program pointed to within the volume ID of the media specified is loaded into RAM and control is passed to it.

During the delay, you can gain control without Network Auto Boot by pressing either the BREAK key or the software abort or reset switches.

Network Auto Boot is controlled by parameters set with the **NIOT** and **ENV** commands. These parameters allow the selection of specific boot devices, systems, and files, and allow programming of the Boot delay. Refer to the **ENV** parameters listed in Chapter 7 for information about **NIOT** and **ENV**, and to Table 6-1 for command syntax. The *PPCBug Debugging Package User's Manual* contains full information about **NIOT** and **ENV**.

Restarting the System

You can initialize the system to a known state in three different ways:

- Break
- Reset
- Abort

Each has certain characteristics which make it more appropriate than the others in given situations.

Break

A "break" is generated by pressing and releasing the BREAK key on the current-console keyboard. Break does not generate an interrupt. The only time break is recognized is when characters are sent or received by the console port. Break performs the following:

- Removes any breakpoints in the user code.
- Keeps the breakpoint table intact.
- Takes a snapshot of the machine state if the function was entered using SYSCALL.
- Allows access to the snapshot for diagnostic purposes.

Many times it may be desirable to terminate a debugger command prior to its completion; for example, the display of a large block of memory. Break allows you to terminate the command immediately.

Reset

A system reset is initiated by pressing and releasing the system board's reset switch, or alternatively, by bridging the appropriate pins on the Reset/Abort Header via a jumper or a remote switch. Refer to Figure 3-1 and Table B-2.

Cold and warm reset modes are available. By default, PPCBug is in cold mode (refer to the **RESET** command description in the *PPCBug Debugging Package User's Manual*). During cold reset, a total system initialization takes place, as if the system board had just been powered up:

- All static variables are restored to their default states.
- The breakpoint table and offset registers are cleared.

- The target registers are invalidated. Input and output character queues are cleared.
- Onboard devices are reset.
- The first two serial ports are reconfigured to their default state.

During warm reset, the PPCBug variables and tables are preserved, as well as the target state registers and breakpoints. Note that revision 1.1 of the PPCBug does not support the warm reset feature.

Reset must be used if the processor ever halts, or if the PPCBug environment is ever lost (vector table is destroyed, stack corrupted, etc.).

Abort

Abort is invoked by pressing and releasing the abort switch, if present on the system board, or alternatively, by bridging the appropriate pins on the Reset/Abort Header via a jumper or a remote switch. Refer to Figure 3-1 and Table B-2.

Whenever abort is invoked when executing a user program (running target code), a "snapshot" of the processor state is captured and stored in the target registers. When working in the debugger, abort captures and stores only the following:

- Instruction pointer
- Status register
- Format and vector information

For this reason, abort is most appropriate when terminating a user program that is being debugged. Abort should be used to regain control if the program gets caught in a loop, etc. The target IP and register contents help to pinpoint the malfunction.

Pressing and releasing the abort switch causes the following:

- An interrupt is sent to the microprocessor.

- The target registers, reflecting the machine state at the time the abort switch was pressed, are displayed on the screen.
- Any breakpoints installed in the user code are removed.
- Breakpoint table remains intact.
- Control is returned to the debugger.

MPU Clock Speed Calculation

The MPU clock speed is calculated and checked against a user definable parameter housed in NVRAM (refer to the **CNFG** command in Chapter 7 of this manual, and to the *PPCBug Debugging Package User's Manual*). If the check fails, a warning message displays. The calculated clock speed is also checked against known clock speeds and tolerances.

Memory Requirements

The debugger requires a total of 512 KB of read/write memory. The debugger allocates this memory starting from the top of memory. For example, on a system which contains 64 MB ($04000000) of read/write memory (i.e., DRAM), the debugger's memory page is located at $03F80000 to $03FFFFFF.

Data and Address Sizes

Data and address sizes are defined as follows:

A *byte* is eight bits, numbered 0 through 7, with bit 0 being the least significant.

A *half-word* is 16 bits, numbered 0 through 15, with bit 0 being the least significant.

A *word* is 32 bits, numbered 0 through 31, with bit 0 being the least significant.

Byte Ordering

The MPU on the system board is programmed to big-endian byte ordering. Any attempt to use small-endian byte ordering immediately renders the PPCBug debugger unusable.

MPU, Hardware, and Firmware Initialization

The debugger performs the MPU, hardware, and firmware initialization process. This process occurs each time the system is reset or powered up. The steps below are a high-level outline; not all of the detailed steps are listed.

1. Sets MPU.MSR to known value.
2. Invalidates the MPU's data/instruction caches.
3. Clears all segment registers of the MPU.
4. Clears all block address translation registers of the MPU.
5. Initializes the MPU-bus-to-PCI-bus bridge device.
6. Initializes the PCI-bus-to-ISA-bus bridge device.
7. Calculates the external bus clock speed of the MPU.
8. Delays for 750 milli-seconds.
9. Determines the CPU base board type.
10. Sizes the local read/write memory (i.e., DRAM).
11. Initializes the read/write memory controller. Sets base address of memory to $00000000.
12. Retrieves the speed of read/write memory from NVRAM.

13. Initializes the read/write memory controller with the speed of read/write memory.
14. Retrieves the speed of read only memory (i.e., FLASH) from NVRAM.
15. Initializes the read only memory controller with the speed of read only memory.
16. Enables the MPU's instruction cache.
17. Copies the MPU's exception vector table from $FFF00000 to $00000000.
18. Initializes the SIO (PC87303) resources' base addresses.
19. Verifies MPU type.
20. Enables the super-scalar feature of the MPU (MPC604 only).
21. Verifies the external bus clock speed of the MPU.
22. Initializes the keyboard controller (PC87303).
23. Determines the debugger's console/host ports, and initializes the appropriate devices (PC16550A/GD5434/GD5436).
24. Displays the debugger's copyright message.
25. Displays any hardware initialization errors that may have occurred.
26. Checksums the debugger object, and displays a warning message if the checksum failed to verify.
27. Displays the amount of local read/write memory found.
28. Verifies the configuration data that is resident in NVRAM, and displays a warning message if the verification failed.
29. Calculates and displays the MPU clock speed, verifies that the MPU clock speed matches the configuration data, and displays a warning message if the verification fails.

30. Displays the BUS clock speed, verifies that the BUS clock speed matches the configuration data, and displays a warning message if the verification fails.
31. Displays any keyboard controller initialization errors that may have occurred.
32. Probes PCI bus for supported network devices.
33. Probes PCI bus for supported mass storage devices.
34. Initializes the memory/IO addresses for the supported PCI bus devices.
35. Executes Self-Test, if so configured.
36. Extinguishes the board fail LED, if Self-Test passed, and outputs any warning messages.
37. Executes ROMboot, if so configured.
38. Executes Auto Boot, if so configured.
39. Executes Network Auto Boot, if so configured.
40. Executes the debugger monitor (i.e., issues the `PPC1-Bug>` prompt).

Board Failure

The following conditions result in a board failure:

- Board initialization error/failure
- Debugger object checksum error
- Environment data (NVRAM **ENV** parameters) failure (i.e., checksum)
- Configuration data (NVRAM **CNFG** parameters) failure (i.e., checksum)

- Calculated MPU clock speed does not match the associative **CNFG** parameter
- Calculated BUS clock speed does not match the associative **CNFG** parameter
- Selftest error/failure

Performing Diagnostic Tests

The PPCBug hardware diagnostics are intended for testing and troubleshooting the system board.

In order to use the diagnostics, you must switch to the diagnostic directory. You may switch between directories by using the **SD** (Switch Directories) command. You may view a list of the commands in the directory that you are currently in by using the **HE** (Help) command.

If you are in the debugger directory, the debugger prompt `PPC1-Bug>` displays, and all of the debugger commands are available. Diagnostics commands cannot be entered at the `PPC1-Bug>` prompt.

If you are in the diagnostic directory, the diagnostic prompt `PPC1-Diag>` displays, and all of the debugger and diagnostic commands are available.

The diagnostic test groups are listed in the following table. Refer to the *PPC1Bug Diagnostics Manual* for complete descriptions of the diagnostic routines available in each test group and instructions on how to invoke them.

Table 5-2. Diagnostic Test Groups

Test Group	Description
CS4231	Audio Codec Tests
DEC21040	DEC21040 Ethernet Controller Tests
I82378	i82378 PCI/ISA Bridge Tests
KBD87303	PC87303 Keyboard/Mouse Tests
L2CACHE	Level 2 Cache Tests
NCR	SYM (formerly NCR) 53C825/53C810 SCSI-2 I/O Processor Tests
PAR87303	PC87303/87323 Parallel Port Test
PC16550	PC16550 UART Tests
PCIBUS	Generic PCI/PMC Slot Tests
RAM	Local RAM Tests
RTC	M48T18 Real-Time Clock Tests
VGA543X	Video Diagnostics Tests

Notes 1. You may enter command names in either uppercase or lowercase.

2. Some diagnostics depend on restart defaults that are set up only in a particular restart mode. Refer to the documentation on a particular diagnostic for the correct mode.

Using the Debugger 6

Entering Commands

PPCBug is command-driven and performs its various operations in response to commands that you enter at the keyboard. When the debugger prompt `PPC1-Bug>` appears on the screen, then the debugger is ready to accept commands.

What you enter is stored in an internal buffer. Execution begins only after you press the Return key, allowing you to correct entry errors, if necessary, using the control characters described in the *PPCBug General Information* chapter.

After the debugger executes the command, the prompt reappears. However, if the command causes execution of user target code (for example **GO**) then control may or may not return to the debugger, depending on what the user program does. For example, if a breakpoint has been specified, then control returns to the debugger when the breakpoint is encountered during execution of the user program. Alternately, the user program could return to the debugger by means of the System Call Handler routine .RETURN (described in the *PPCBug Debugging Package User's Manual*). For more about this, refer to the **GD**, **GO**, and **GT** command descriptions in the *PPCBug Debugging Package User's Manual*.

In general, a debugger command is made up of the following parts:

- The command name (e.g., **MD** or **md**). Note that either uppercase or lowercase is allowed.
- At least one intervening space before the first argument.
- Any required arguments, as specified by command.
- One or more options. Precede an option or a string of options with a semi-colon (**;**). If no option is entered, the command's default option conditions are used.

Command Syntax

The command syntax is shown below:

boldface strings	A boldface string is a literal such as a command name, program name, or option, and is to be typed just as it appears.
italic strings	An italic string is an argument.
|	A vertical bar separating two or more items indicates that a choice is to be made; only one of the items separated by this symbol should be selected.
[ ]	Square brackets enclose an item that is optional. The item may appear zero or one time.
{ }	Braces enclose an optional symbol that may occur zero or more times.

Command Arguments

The following arguments are common to many of the commands. Additional arguments are defined in the descriptions of commands found in the *PPCBug Debugging Package User's Manual*. You will also see the arguments for each command when you type **HE** to display the help menu.

EXP	Expression (see *EXP* on page 6-3).
ADDR	Address (see *ADDR* on page 6-5).
COUNT	Count; the syntax is the same as for *EXP* (see *EXP* on page 6-3).
RANGE	A range of memory addresses specified with a pair of arguments, either *ADDR ADDR* or *ADDR : COUNT*
TEXT	An ASCII string of up to 255 characters, delimited at each end by the single quote mark (').
PORT	Port Number (see *PORT* on page 6-6).

Use either a space or a comma to separate arguments. You may select the default value for an argument by inserting a pair of commas in place of the argument.

EXP

The EXP (expression) argument can be one or more numeric values separated by the arithmetic operators:

+	plus
-	minus
*	multiply by
/	divide by
&	logical AND
<<	shift left
>>	shift right

Numeric values may be expressed in either hexadecimal, decimal, octal, or binary by immediately preceding them with the proper base identifier.

Data Type	Base	Identifier	Examples
Integer	Hexadecimal	$	$FFFFFFFF
Integer	Decimal	&	&1974, &10-&4
Integer	Octal	@	@456
Integer	Binary	%	%1000110

If no base identifier is specified, then the numeric value is assumed to be hexadecimal.

A numeric value may also be expressed as a string literal of up to four characters. The string literal must begin and end with the single quote mark ('). The numeric value is interpreted as the concatenation of the ASCII values of the characters. This value is right-justified, as any other numeric value would be.

String Literal	Numeric Value (Hexadecimal)
'A'	41
'ABC'	414243
'TEST'	54455354

Evaluation of an expression is always from left to right unless parentheses are used to group part of the expression. There is no operator precedence. Sub-expressions within parentheses are evaluated first. Nested parenthetical sub-expressions are evaluated from the inside out.

Valid expression examples:

Expression	Result (Hexadecimal)	Notes
FF0011	FF0011	
45+99	DE	
&45+&99	90	
@35+@67+@10	5C	
%10011110+%1001	A7	
88<<4	880	shift left
AA&F0	A0	logical AND

The total value of the expression must be between 0 and $FFFFFFFF.

ADDR

The syntax for the *ADDR* argument is similar to the syntax accepted by the PowerPC one-line assembler. All control addressing modes are allowed. An "address + offset register" mode is also provided.

ADDR Formats

The *ADDR* format is:

HexadecimalNumber {**[^S]** | **[^s]** | **[^U]** | **[^u]**} | **R***n*

Enter *ADDR* as a hexadecimal number (e.g., 20000 for address $00020000). The address, or starting address of a range, can be qualified by a suffix, either **^S** or **^s** for supervisor address space, or **^U** or **^u** for user address space. The default, when the suffix is not specified, is supervisor.

Once a qualifier has been entered, it remains valid for all addresses entered for that command sequence, until either the PPCBug is reentered or another qualifier is provided.

In the alternate register number (**R***n*) form, the debugger uses the address contained in MPU Register **R***n*, where *n* is 0 through 31 (i.e., 0, 1, . . . 31).

In commands with the address range specified as *ADDR ADDR*, and with size option **H** or **W** chosen, data at the second (ending) address is acted on only if the second address is a proper boundary for a half-word or word, respectively. Otherwise, the range is truncated so that the last byte acted upon is at an address that is a proper boundary.

Offset Registers

Eight pseudo-registers (Z0-Z7) called offset registers are used to simplify the debugging of relocatable and position-independent modules. The listing files in these types of programs usually start at an address (normally 0) that is not the one at which they are loaded, so it is harder to correlate addresses in the listing with addresses in the loaded program. The offset registers solve this problem by taking into account this difference and forcing the display of

addresses in a relative address+offset format. Offset registers have adjustable ranges and may even have overlapping ranges. The range for each offset register is set by two addresses: base and top. Specifying the base and top addresses for an offset register sets its range. In the event that an address falls in two or more offset registers' ranges, the one that yields the least offset is chosen.

Note Relative addresses are limited to 1MB (5 digits), regardless of the range of the closest offset register.

PORT

The PORT argument is the logical number of the port to be used to input or output. Valid port numbers which may be used for these commands are as follows:

0 or **00** Terminal port 0 (“console port”) is used for interactive user input and output (the default). This port number usually refers to the serial port labelled either COM1 or SER1 on the system board. However, this port number may also be used for the graphics adapter device.

1 or **01** Terminal port 10 (“host port”) is the default for downloading, uploading, concurrent mode, and transparent modes. This port is labeled either COM2 or SER2 on the system board.

Command Options

Many commands have one or more options, defined in the command descriptions found in the *PPCBug Debugging Package User’s Manual*. You will also see the options for each command when you type **HE** to display the help menu.

Precede an option or a string of options with a semi-colon (**;**). If no option is entered, the command’s default option conditions are used.

Terminal Input and Output Control Characters

You may use the following control codes for limited editing while entering commands at the PPC1-Bug> prompt:

CTRL-X (cancel line)	Move the cursor to the beginning of the line. If the terminal port is configured with the hardcopy or TTY option (refer to the **PF** command in the *PPCBug Debugging Package User's Manual*), then a **<CR><LF>** sequence is issued along with another prompt.
CTRL-H (backspace)	Moved the cursor back one position. The character at the new cursor position is erased. If the hardcopy option is selected, a "/" character is typed along with the deleted character.
DEL (delete or rubout) key	Performs the same function as **CTRL-H**.
CTRL-D (redisplay)	Redisplay the entire command line as entered so far is on the following line.
CTRL-A (repeat)	Repeat the previous line. This happens only at the command line. The last line entered is redisplayed but not executed. The cursor is positioned at the end of the line. You may enter the line as is or you can add more characters to it. You can edit the line by backspacing and typing over old characters.

The XON and XOFF characters in effect for the terminal port may be entered to control the output from any PPCBug command, if the XON/XOFF protocol is enabled (default). The characters listed are initialized by PPCBug, but you may change them with the **PF** command:

CTRL-S (wait)	Halt console output (XON).
CTRL-Q (resume)	Resume console output (XOFF).

PPCBug Debugger Command Set

The PPCBug debugger commands are summarized in the following table. The command syntax is shown using the symbols explained earlier in this chapter. All command details are explained in the *PPCBug Debugging Package User's Manual.*

Table 6-1. Debugger Commands

Command Mnemonic	Command Title	Command Line Syntax
AS	One Line Assembler	**AS** *ADDR*
BC	Block of Memory Compare	**BC** *RANGE ADDR* [**;B** | **H** | **W**]
BF	Block of Memory Fill	**BF** *RANGE data* [*increment*] [**;B** | **H** | **W**]
BI	Block of Memory Initialize	**BI** *RANGE* [**;B** | **H** | **W**]
BM	Block of Memory Move	**BM** *RANGE ADDR* [**;B** | **H** | **W**]
BR	Breakpoint Insert	**BR** [*ADDR*[**:***COUNT*]]
NOBR	Breakpoint Delete	**NOBR** [*ADDR*]
BS	Block of Memory Search	**BS** *RANGE TEXT* [**;B** | **H** | **W**] or **BS** *RANGE data* [*mask*] [**;B** | **H** | **W** [**,N**] [**,V**]]
BV	Block of Memory Verify	**BV** *RANGE data* [*increment*] [**;B** | **H** | **W**]
CM	Concurrent Mode	**CM** [[*PORT*] [*ID-STRING*] [*BAUD*] [*PHONE-NUMBER*]] | [**;A**] | [**;H**]
NOCM	No Concurrent Mode	**NOCM**
CNFG	Configure Board Information Block	**CNFG** [**;**[**I**] [**M**]]
CS	Checksum	**CS** *RANGE* [**;B** | **H** | **W**]

Table 6-1. Debugger Commands (Continued)

Command Mnemonic	Command Title	Command Line Syntax
DC	Data Conversion	**DC** *EXP* | *ADDR* **[;[B] [O] [A]]**
DMA	Block of Memory Move	**DMA** R*ANGE ADDR VDIR AM BLK* **[;B | H | W]**
DS	One Line Disassembler	**DS** *ADDR* [*:COUNT* | *ADDR*]
DU	Dump S-Records	**DU** [*PORT*] *RANGE* [*TEXT*] [*ADDR*] [*OFFSET*] **[;B | H | W]**
ECHO	Echo String	**ECHO** [*PORT*] {*hexadecimal number*} {'*string*'}
ENV	Set Environment	**ENV [;[D]]**
GD	Go Direct (Ignore Breakpoints)	**GD** [*ADDR*]
GN	Go to Next Instruction	**GN**
GO	Go Execute User Program	**GO** [*ADDR*]
GT	Go to Temporary Breakpoint	**GT** *ADDR*
HE	Help	**HE** [*COMMAND*]
IOC	I/O Control for Disk	**IOC**
IOI	I/O Inquiry	**IOI [;[C | L]]**
IOP	I/O Physical (Direct Disk Access)	**IOP**
IOT	I/O Teach for Configuring Disk Controller	**IOT [;[A | F | H | T]]**

Table 6-1. Debugger Commands (Continued)

Command Mnemonic	Command Title	Command Line Syntax
LO	Load S-Records from Host	**LO** [*PORT*] [*ADDR*] [**;[X]** [**C**] [**T**]] [**=***text*]
MA	Macro Define/Display	**MA** [*NAME* | **;L**]
NOMA	Macro Delete	**NOMA** [*NAME*]
MAE	Macro Edit	**MAE** *NAME LINE #* [*STRING*]
MAL	Enable Macro Listing	**MAL**
NOMAL	Disable Macro Listing	**NOMAL**
MAR	Load Macros	**MAR** [*controllerLUN*] [[*deviceLUN*] [*block#*]]
MAW	Save Macros	**MAW** [*controllerLUN*] [[*deviceLUN*] [*block#*]]
MD, MDS	Memory Display	**MD** *ADDR*[**:***COUNT* | *ADDR*] [**;** [**B** | **H** | **W** | **S** | **D** | **DI**]] **MDS** *ADDR*[**:***COUNT* | *ADDR*] [**;** [**B** | **H** | **W** | **S** | **D** | **DI**]]
MENU	System Menu	**MENU**
MM	Memory Modify	**MM** *ADDR* [**;**[[**B** | **H** | **W** | **S** | **D**] [**A**] [**N**]] | [**DI**]]
MMD	Memory Map Diagnostic	**MMD** *RANGE INCREMENT* [**;B** | **H** | **W**]
MS	Memory Set	**MS** *ADDR* {*Hexadecimal number*} {'*string*'}
MW	Memory Write	**MW** *ADDR DATA* [**;B** | **H** | **W**]
NAB	Automatic Network Boot	**NAB**
NBH	Network Boot Operating System, Halt	**NBH** [*ControllerLUN*] [*DeviceLUN*] [*ClientIPAddress*] [*ServerIPAddress*] [*String*]

Table 6-1. Debugger Commands (Continued)

Command Mnemonic	Command Title	Command Line Syntax
NBO	Network Boot Operating System	**NBO** [*ControllerLUN*] [*DeviceLUN*] [*ClientIPAddress*] [*ServerIPAddress*] [*String*]
NIOC	Network I/O Control	**NIOC**
NIOP	Network I/O Physical	**NIOP**
NIOT	Network I/O Teach (Configuration)	**NIOT** [**;**[**H**] | [**A**]]
NPING	Network Ping	**NPING** *ControllerLUN DeviceLUN SourceIP DestinationIP* [*NPackets*]
OF	Offset Registers Display/Modify	**OF** [**Z***n*[**;A**]]
PA	Printer Attach	**PA** [*PORT*]
NOPA	Printer Detach	**NOPA** [*PORT*]
PBOOT	Bootstrap Operating System	**PBOOT ; A** | **V** | **C** **PBOOT** *CLUN DLUN PARTITION String* [**;**[**H** | **B**]]
PF	Port Format	**PF** [*PORT*]
NOPF	Port Detach	**NOPF** [*PORT*]
PFLASH	Program FLASH Memory	**PFLASH** *SSADDR SEADDR DSADDR* [*IEADDR*] [**;**[**A** | **R**] [**X**]] **PFLASH** *SSADDR***:***COUNT DSADDR* [*IEADDR*] [**;**[**B** | **W** | **L**] [**A** | **R**] [**X**]]
PS	Put RTC into Power Save Mode	**PS**
RB	ROMboot Enable	**RB**[**;V**]
NORB	ROMboot Disable	**NORB**

Table 6-1. Debugger Commands (Continued)

Command Mnemonic	Command Title	Command Line Syntax
RD	Register Display	**RD** [{[+ | - | =] [*DNAME*] [/]}{[+ | - | =] [*REG1*[-*REG2*]] [/]}] [**;E**]
REMOTE	Remote	**REMOTE**
RESET	Cold / Warm Reset	**RESET**
RL	Read Loop	**RL** *ADDR*[**;B** | **H** | **W**]
RM	Register Modify	**RM** [*REG*]
RS	Register Set	**RS** *REG* [*EXP* | *ADDR*]
SD	Switch Directories	**SD**
SET	Set Time and Date	**SET** *mmddyyhhmm*
SYM	Symbol Table Attach	**SYM** [*ADDR*]
NOSYM	Symbol Table Detach	**NOSYM**
SYMS	Symbol Table Display / Search	**SYMS** [*symbol-name*] | [**;S**]
T	Trace	**T** [*COUNT*]
TA	Terminal Attach	**TA** [*PORT*]
TIME	Display Time and Date	**TIME** [**;L**]
TM	Transparent Mode	**TM** [*PORT*] [*ESCAPE*]
TT	Trace to Temporary Breakpoint	**TT** *ADDR*
VE	Verify S-Records Against Memory	**VE** [*PORT*] [*ADDR*] [**;**[**X**] [**C**]] [**=***text*]

Table 6-1. Debugger Commands (Continued)

Command Mnemonic	Command Title	Command Line Syntax
VER	Revision/Version Display	**VER [;E]**
WL	Write Loop	**WL** *ADDR*:*DATA*[**;B** | **H** | **W**]

Advanced Debugger Topics 7

Modifying Parameters in NVRAM

You can use the factory-installed debug monitor, PPCBug, to modify certain parameters contained in the Ultra system board's Non-Volatile RAM (NVRAM), also known as Battery Backed-up RAM (BBRAM).

- The Board Information Block in NVRAM contains various elements concerning operating parameters of the hardware. Use the PPCBug command **CNFG** to change those parameters.
- Use the PPCBug command **ENV** to change configurable PPCBug parameters in NVRAM.

The **CNFG** and **ENV** commands are both described in the *PPCBug Debugging Package User's Manual*. Refer to that manual for general information about their use and capabilities.

The following paragraphs present additional information about **CNFG** and **ENV** that is specific to the Ultra Plus and Ultra 60*x*, along with the parameters that can be configured with the **ENV** command.

CNFG - Configure Board Information Block

Use this command to display and configure the Board Information Block, which is resident within the NVRAM. The board information block contains various elements detailing specific

operational parameters of the PowerPC board. The board structure for the PowerPC board is as shown in the following typical example:

```
Board (PWA) Serial Number    = “1673590       ”
Board Identifier             = “Ultra 60x     ”
Artwork (PWA) Identifier     = “01-w3107F01A  ”
MPU Clock Speed              = “066           ”
Bus Clock Speed              = “033           ”
Ethernet Address             = 08003E20C983
Local SCSI Identifier        = “07”
System Serial Number         = “1463725       ”
System Identifier            = “Motorola Ultra 60x”
```


The parameters that are quoted are left-justified character (ASCII) strings padded with space characters, and the quotes (“) are displayed to indicate the size of the string. Parameters that are not quoted are considered data strings, and data strings are right-justified. The data strings are padded with zeroes if the length is not met.

The Board Information Block is factory-configured before shipment. There is no need to modify block parameters unless the NVRAM is corrupted.

Refer to the *Ultra 603/Ultra 603e/Ultra 604 Programmer’s Reference Guide* for the actual location and other information about the Board Information Block.

Refer to the *PPCBug Debugging Package User’s Manual* for a description of **CNFG** and examples.

ENV - Set Environment

Use the **ENV** command to view and/or configure interactively all PPCBug operational parameters that are kept in Non-Volatile RAM (NVRAM).

Refer to the *PPCBug Debugging Package User's Manual* for a description of the use of **ENV**. Additional information on control and status registers that affect these parameters is contained in the *Ultra 603/Ultra 603e/Ultra 604 Programmer's Reference Guide*.

Listed and described below are the parameters that you can configure using **ENV**. The default values shown were those in effect when this publication went to print.

Configuring the PPCBug Parameters

The parameters that can be configured using **ENV** are:


```
Bug or System environment [B/S] = S?
```

B Bug is the mode where no system type of support is displayed. However, system-related items are still available.

S System is the standard mode of operation, and is the default mode if NVRAM should fail. System mode is defined in the *PPCBug Debugging Package User's Manual*. (Default)

```
Field Service Menu Enable [Y/N] = Y?
```

Y Display the field service menu. (Default)

N Do not display the field service menu.

```
Probe System for Supported I/O Controllers [Y/N] = Y?
```

Y Accesses will be made to the appropriate system buses to determine the presence of supported controllers. (Default)

N Accesses will not be made to the bus to determine the presence of supported controllers.

```
Local SCSI Bus Reset on Debugger Setup [Y/N] = N?
```

Y Local SCSI bus is reset on debugger setup.

N Local SCSI bus is not reset on debugger setup. (Default)

```
Local SCSI Bus Negotiations Type [A/S/N] = A?
```

A Asynchronous SCSI bus negotiation. (Default)

S Synchronous SCSI bus negotiation.

N None.

```
Local SCSI Data Bus Width [W/N] = N?
```

W Wide SCSI (16-bit bus). (Default)

N Narrow SCSI (8-bit bus).


```
Auto Boot Enable [Y/N]   = Y?
```

Y The Auto Boot function is enabled. (Default)

N The Auto Boot function is disabled.

```
Auto Boot at power-up only [Y/N] = N?
```

Y Auto Boot is attempted at power-up reset only.

N Auto Boot is attempted at any reset. (Default)

```
Auto Boot Scan Enable [Y/N]   = Y?
```

Y If Auto Boot is enabled, the Auto Boot process attempts to boot from devices specified in the scan list (e.g., FDISK/CDROM/TAPE/HDISK). (Default)

N If Auto Boot is enabled, the Auto Boot process uses the Controller LUN and Device LUN to boot.

```
Auto Boot Scan Device Type List = FDISK/CDROM/TAPE/HDISK?
```

This is the listing of boot devices displayed if the Auto Boot Scan option is enabled.

```
Auto Boot Controller LUN   = 00?
```

Refer to the *PPCBug Debugging Package User's Manual* for a listing of disk/tape controller modules currently supported by PPCBug. (Default = $0)

```
Auto Boot Device LUN      = 00?
```

Refer to the *PPCBug Debugging Package User's Manual* for a listing of disk/tape devices currently supported by PPCBug. (Default = $0)

```
Auto Boot Partition Number = 00?
```

Which disk "partition" is to be booted, as specified in the PowerPC Reference Platform (PRP) specification. If set to zero, the firmware will search the partitions in order (1, 2, 3, 4) until it finds the first "bootable" partition. That is then the partition that will be booted. Other acceptable values are 1, 2, 3, or 4. In these four cases, the partition specified will be booted without searching.


```
Auto Boot Abort Delay    = 7?
```

This is the time in seconds that the Auto Boot sequence will delay before starting the boot. The purpose for the delay is to allow you the option of stopping the boot by use of the **BREAK** key. The time value is from 0-255 seconds. (Default = 7)

```
Auto Boot Default String [NULL for an empty string] = ?
```

You may specify a string (filename) which is passed on to the code being booted. The maximum length of this string is 16 characters. (Default = null string)

```
ROM Boot Enable [Y/N] = N?
```

Y The ROMboot function is enabled.

N The ROMboot function is disabled. (Default)

```
ROM Boot at power-up only [Y/N] = Y?
```

Y ROMboot is attempted at power-up only. (Default)

N ROMboot is attempted at any reset.

```
ROM Boot Abort Delay    = 00?
```

This is the time in seconds that the ROMboot sequence will delay before starting the boot. The purpose for the delay is to allow you the option of stopping the boot by use of the **BREAK** key. The time value is from 0-255 seconds. (Default = 0 seconds)

```
ROM Boot Direct Starting Address = FFF00000?
```

This is the first location tested when PPCBug searches for a ROMboot module. (Default = $FFF00000)

```
ROM Boot Direct Ending Address   = FFFFFFFC?
```

This is the last location tested when PPCBug searches for a ROMboot module. (Default = $FFFFFFFC)

```
Network Auto Boot Enable [Y/N]   = N?
```

Y The Network Auto Boot function is enabled.

N The Network Auto Boot function is disabled. (Default)

```
Network Auto Boot at power-up only [Y/N] = Y?
```

Y Network Auto Boot is attempted at power-up reset only. (Default)

N Network Auto Boot is attempted at any reset.

```
Network Auto Boot Controller LUN = 00?
```

Refer to the *PPCBug Debugging Package User's Manual* for a listing of disk/tape controller modules currently supported by PPCBug. (Default = $0)

```
Network Auto Boot Device LUN     = 00?
```

Refer to the *PPCBug Debugging Package User's Manual* for a listing of disk/tape controller modules currently supported by PPCBug. (Default = $0)

```
Network Auto Boot Abort Delay    = 5?
```

This is the time in seconds that the Network Auto Boot sequence will delay before starting the boot. The purpose for the delay is to allow you the option of stopping the boot by use of the **BREAK** key. The time value is from 0-255 seconds. (Default = 5 seconds)

```
Network Auto Boot Configuration Parameters Offset (NVRAM) = FFFFFFFF?
```

This is the address where the network interface configuration parameters are to be saved/retained in NVRAM; these parameters are the necessary parameters to perform an unattended network boot. A typical offset might be $10000000, but this value is application-specific. (Default = $FFFFFFFF)

Caution

If you use the **NIOT** debugger command, these parameters must be saved/retained in NVRAM, somewhere in the address range $00000000 through $00000FFF. The **NIOT** parameters do not exceed 128 bytes in size. The location of these parameters is determined by setting this **ENV** offset. If you have used the exact same space for your own program information or commands, they will be overwritten and lost.

You can relocate the network interface configuration parameters in this space by using the **ENV** command to change the Network Auto Boot Configuration Parameters Offset (NVRAM) from its default of $FFFFFFFF to the value you need so as to be clear of your data within NVRAM.

```
Memory Size Enable [Y/N]         = Y?
```

Y Memory will be sized for Self Test diagnostics. (Default)

N Memory will not be sized for Self Test diagnostics.

```
Memory Size Starting Address     = 00000000?
```

The default Starting Address is $00000000.

```
Memory Size Ending Address       = 80000000?
```

The default Ending Address is the calculated size of local memory. If the memory start is changed from $00000000, this value will also need to be adjusted.

```
DRAM Speed in NANO Seconds = 60?
```

The default setting for this parameter will vary depending on the speed of the DRAM memory parts installed on the board. The default is set to the slowest speed found on the available banks of DRAM memory.


```
ROM First Access Length (0 - 31) = 2?
```

This is the value programmed into the MPC105 “ROMFAL” field (Memory Control Configuration Register 8: bits 23-27) to indicate the number of clock cycles used in accessing the ROM. The lowest allowable ROMFAL setting is $00; the highest allowable is $1F. The value to enter depends on processor speed; refer to your *Processor/Memory Mezzanine Module User’s Manual* for appropriate values. The default value varies according to the system’s bus clock speed.

```
ROM Next Access Length (0 - 15) = 2?
```

This is the value programmed into the MPC105 “ROMNAL” field (Memory Control Configuration Register 8: bits 28-31) to represent wait states in access time for nibble (or burst) mode ROM accesses. The lowest allowable ROMNAL setting is $0; the highest allowable is $F. The value to enter depends on processor speed; refer to your *Processor/Memory Mezzanine Module User’s Manual* for appropriate values. The default value varies according to the system’s bus clock speed.

Entering and Debugging Programs

There are various ways to enter a user program into system memory for execution. One way is to create the program using the **MM** (Memory Modify) command with the **DI** (assembler/disassembler) option, entering the program one source line at a time. After each source line is entered, it is assembled and the object code is loaded to memory. Refer to the *PPCBug Debugging Package User's Manual* for complete details of the PPCBug Assembler/Disassembler.

Another way is to download an object file from a host system. The program must be in S-record format (refer to the *PPCBug Debugging Package User's Manual*) and may have been assembled or compiled on the host system. Alternately, the program may have been previously created using the **MM** command as outlined above, and stored to the host using the **DU** (Dump) command. A communication link must exist between the host system and system board port 1. (Refer to Chapter 1 for hardware configuration information.) The file is downloaded from the host to system board memory by the **LO** (Load) command.

Once the object code has been loaded into memory, you can set breakpoints if desired and run the code or trace through it.

Calling System Routines from User Programs

Access to various PPCBug routines is provided via the System Call Handler. This gives a convenient way of doing character input/output and many other useful operations so that you do not have to write these routines into the target code. Refer to the *PPCBug Debugging Package User's Manual* for details on the routines available and how to invoke them from within a user program.

The System Call Handler is accessible through the **sc** (system call) instruction, with exception vector $00C00 (System Call Exception).

Preserving the Operating Environment

This section explains how to avoid contaminating the operating environment of the debugger. PPCBug uses certain portions of the system board onboard resources and also off-board system memory to contain temporary variables, exception vectors, etc. If you disturb resources upon which PPCBug depends, then the debugger may not function reliably or may not function at all.

If your application enables translation through the Memory Management Unit (MMU), and utilizes resources of the debugger (e.g., system calls), your application must create the necessary translation tables for the debugger to have access to its various resources. The debugger honors the enabling of the MMU; it does not alter or disable translation.

Memory Requirements

The debugger requires a total of 512K bytes of read/write memory. The debugger will allocate this memory from the top of memory. For example, on a system which contains 64 megabytes ($04000000) of read/write memory (i.e., DRAM), the debugger's memory page will be located at $03F80000 to $03FFFFFF.

This memory space is used by the debugger for program stack, I/O buffers, variables, and register files. If a user program is loaded (e.g., booted, S-Records) into memory, and if this program is utilizing the debugger's programmatic interface (i.e., system calls), the program must not modify this allocated memory.

Whenever the host hardware is reset, the following is done:

- Target IP is initialized to $00004000 (i.e., just above the memory space of the exception vector table).
- Target pseudo stack pointer is initialized to the starting location of the debugger's read/write memory space.
- Target IP will be set to the appropriate address if a program load operation (e.g., the **PBOOT** command) is initiated.

Note that user programs should handle the stack area properly in that it should not write starting at the initialized location. Some compilers and assemblers may write to the stack prior to decrementing the stack.

This read/write memory space that is allocated for the debugger, by the debugger, may increase in future releases of the debugger. To properly compensate for the increased read/write memory requirements, user programs may utilize the target IP as indicator for the top (plus 1) of usable memory.

Exception Vectors Used by PPCBug

The following exception vectors are reserved for use by the debugger:

00100	System Reset	Used for the abort switch soft reset feature.
00700	Program	Used for instruction breakpoints.
00C00	System Call	Used for the System Call Handler.
02000	Run Mode	Used for instruction tracing.

These vectors may be taken over under a user's application. However, prior to returning control to the debugger these vectors must be restored for proper operation of the affected features.

MPU Registers

Certain MPU registers must be preserved for their specific uses.

MPU Register SPR275

MPU register SPR275 is reserved for usage by the debugger. If SPR275 is to be used by the user program, it must be restored prior to utilizing debugger resources (system calls) and or returning control to the debugger.

MPU Registers SPR272-SPR274

These MPU registers are utilized by debugger as scratch registers.

Context Switching

Context switching is the switching from the debugger state to the user (target) state, or vice-versa. This switching occurs upon the invocation of either the **GD**, **GN**, **GO**, **GT**, **T**, or **TT** commands, or the return from user state to the debugger state.

User State to Debugger State

When the context switch transitions from the user state to the debugger state, the following MPU registers are captured:

MPC603/MPC603e-based Boards:

R0-R31	General Purpose Registers
FR0-FR31	Floating Point Unit Data Registers
SR0-SR15	Segment Registers
SPR*n*	Special Purpose Registers (*n* is 1, 8, 9, 18, 19, 22, 25, 26, 27, 268, 269, 275, 282, 287, 528 - 543, 976 - 981, 1008, 1010)
IP	Instruction Pointer (copy of SPR26)
MSR	Machine State Register (copy of SPR27)
CR	Condition Register
FPSCR	Floating Point Status/Control Register

MPC604-based Boards:

R0-R31	General Purpose Registers
FR0-FR31	Floating Point Unit Data Registers
SR0-SR15	Segment Registers
SPR*n*	Special Purpose Registers (*n* is 1, 8, 9, 18, 19, 22, 25, 26, 27, 268, 269, 275, 282, 287, 528 - 543, 1008, 1010, 1013, 1023)
IP	Instruction Pointer (copy of SPR26)
MSR	Machine State Register (copy of SPR27)

CR	Condition Register
FPSCR	Floating Point Status/Control Register

Debugger State to User State

When the context switch transitions from the debugger state to the user state, the following MPU registers are restored:

MPC603/MPC603e-based Boards:

R0-R31	General Purpose Registers
FR0-FR31	Floating Point Unit Data Registers
SPR*n*	Special Purpose Registers (*n* is 1, 8, 9, 275, 1010)
IP	Instruction Pointer (copied to SPR26)
MSR	Machine State Register (copied to SPR2)
CR	Condition Register
FPSCR	Floating Point Status/Control Register

MPC604-based Boards:

0-R31	General Purpose Registers
FR0-FR31	Floating Point Unit Data Registers
SPR*n*	Special Purpose Registers (*n* is 1, 8, 9, 275, 1010, 1013, 1023)
IP	Instruction Pointer (copied to SPR26)
MSR	Machine State Register (copied to SPR27)
CR	Condition Register
FPSCR	Floating Point Status/Control Register

Note that on a restoration context switch, registers whose perspectives feature MMU characteristics and operating modes of the MPU are not restored. The debugger honors the user's MMU configuration. If the user's program wishes to utilize the programmatic interface (i.e., system calls) of the debugger, it must maintain the address translation of 1 to 1, and the I/O resources utilized by the debugger must be data cache inhibited.

Floating Point Support

The **MD** and **MM** commands allow display and modification of floating point data in memory. Use either the **MD** command or the **MM** command to assemble or disassemble floating point instructions.

Valid data types that can be used when modifying a floating point data register or a floating point memory location:

Integer Data Types	
Byte	12
Half-Word	1234
Word	12345678

Floating Point Data Types	
Single Precision Real	1_FF_7FFFFF
Double Precision Real	1_7FF_FFFFFFFFFFFF
Scientific Notation (decimal)	-3.12345678901234501_E+123

When entering data in single or double precision format, observe the following rules:

- The sign field is the first field and is a binary field.
- The exponent field is the second field and is a hexadecimal field.
- The mantissa field is the last field and is a hexadecimal field.
- The sign field, the exponent field, and at least the first digit of the mantissa field must be present (any unspecified digits in the mantissa field are set to zero).
- Each field must be separated from adjacent fields by an underscore.
- All the digit positions in the sign and exponent fields must be present.

Single Precision Real

The single precision real format would appear in memory as:

1-bit sign field	(1 binary digit)
8-bit biased exponent field	(2 hex digits. Bias = $7F)
23-bit fraction field	(6 hex digits)

A single precision number takes 4 bytes in memory.

Double Precision Real

The double precision real format would appear in memory as:

1-bit sign field	(1 binary digit)
11-bit biased exponent field	(3 hex digits. Bias = $3FF)
52-bit fraction field	(13 hex digits)

A double precision number takes 8 bytes in memory.

Note The single and double precision formats have an implied integer bit (always 1).

Scientific Notation

The scientific notation format provides a convenient way to enter and display a floating point decimal number. Internally, the number is assembled into a packed decimal number and then converted into a number of the specified data type.

Entering data in this format requires the following fields:

- An optional sign bit (+ or -)
- One decimal digit followed by a decimal point
- Up to 17 decimal digits (at least one must be entered)
- An optional Exponent field that consists of:

- An optional underscore
- The Exponent field identifier, letter E
- An optional Exponent sign (+, -)
- From 1 to 3 decimal digits

For more information about the floating point unit, refer to the *PowerPC 603 RISC Microprocessor User's Manual*.

Disk I/O Support

PPCBug can initiate disk input and output by communicating with intelligent disk controllers over the PCI bus. Disk support facilities built into PPCBug consist of:

- Command-level disk operations
- Disk I/O system calls (only via one of the system call instructions - refer to the *PPCBug Debugging Package User's Manual*) for use by user programs
- Defined data structures for disk parameters

Parameters such as the:

- Address where the module is mapped
- Type of device
- Number of devices attached to the controller module

are kept in tables by PPCBug. Default values for these parameters are assigned at power-up and cold-start reset, but may be altered as described in the *Default PPCBug Controller and Device Parameters* section of this chapter.

Blocks Versus Sectors

The logical block defines the unit of information for disk devices. A disk is viewed by PPCBug as a storage area divided into logical blocks. By default, the logical block size is set to 256 bytes for every block device in the system. The block size can be changed on a per device basis with the **IOT** command.

The sector defines the unit of information for the media itself, as viewed by the controller. The sector size varies for different controllers, and the value for a specific device can be displayed and changed with the **IOT** command.

When a disk transfer is requested:

- The start and size of the transfer is specified in blocks
- PPCBug translates this into an equivalent sector specification
- Sector specification is passed on to the controller to initiate the transfer

If the conversion from blocks to sectors yields a fractional sector count, an error is returned and no data is transferred.

Device Probe Function

A device probe with entry into the device descriptor table is done whenever a specified device is accessed. This happens when these debugger commands or system calls are used:

Commands	System Calls
IOC	.DSKRD
IOP	.DSKWR
IOT	.DSKCFIG
MAR	.DSKFMT
MAW	.DSKCTRL
PBOOT	

The device probe mechanism utilizes the SCSI commands **Inquiry** and **Mode Sense**. If the specified controller is non-SCSI, the probe simply returns a status of **device present and unknown**. The device probe makes an entry into the device descriptor table with the pertinent data. After an entry has been made, the next time a probe is done it simply returns with "device present" status (pointer to the device descriptor).

Disk I/O via PPCBug Commands

These following PPCBug commands are provided for disk I/O. Detailed instructions for their use are found in the *PPCBug Debugging Package User's Manual*. When a command is issued to a particular controller LUN and device LUN, these LUNs are remembered by PPCBug so that the next disk command defaults to use the same controller and device.

IOI (Input/Output Inquiry)

The **IOI** command is used to probe the system for all possible CLUN/DLUN combinations and display inquiry data for devices which support it. The device descriptor table has space for a maximum of 16 device descriptors. With the **IOI** command, you can view the table or clear it if necessary.

IOP (Physical I/O to Disk)

The **IOP** command allows you to:

- Read blocks of data.
- Write blocks of data.
- Format the specified device in a certain way.

IOP creates a command packet from the arguments you specify, and then invokes the proper system call function to carry out the operation.

IOT (I/O Configure)

The **IOT** command allows you to:

- Change any configurable parameters.
- Change device attributes.
- See the controllers available in the system.

IOC (I/O Control)

The **IOC** command allows you to send command packets as defined by the particular controller directly. **IOC** can also be used to examine the resultant device packet after using the **IOP** command.

PBOOT (Bootstrap Operating System)

The **PBOOT** command reads an operating system or control program from the specified device into memory, and then transfers control to it.

With the **H** option, **PBOOT** reads an operating system or control program from a specified device into memory, and then returns control to PPCBug. It is used as a debugging tool.

Disk I/O via PPCBug System Calls

All operations that actually access the disk are done directly or indirectly by PPCBug system calls. (The command-level disk operations provide a convenient way of using these system calls without writing and executing a program).

The following system calls allow user programs to perform disk I/O operations:

.DSKRD	Disk read. Use this system call to read blocks from a disk into memory.

.DSKWR	Disk write. Use this system call to write blocks from memory onto a disk.
.DSKCFIG	Disk configure. Use this system call to change the configuration of the specified device.
.DSKFMT	Disk format. Use this system call to send a format command to the specified device.
.DSKCTRL	Disk control. Use this system call to implement any special device control functions that cannot be accommodated easily using other disk functions.

Refer to the *PPCBug Debugging Package User's Manual* for information on using these and other system calls.

To perform a disk operation, PPCBug must eventually present a particular disk controller module with a controller command packet which has been especially prepared for that type of controller module. (This is accomplished in the respective controller driver module.) A command packet for one type of controller module usually does not have the same format as a command packet for a different type of module. The system call facilities that perform disk I/O operations do the following:

1. Accept a generalized (controller-independent) packet format as an argument.
2. Translate it into a controller-specific packet.
3. Send it to the specified device.

Refer to the system call descriptions in the *PPCBug Debugging Package User's Manual* for details on the format and construction of these standardized "user" packets.

The packets which a controller module expects to be given vary from controller to controller. The disk driver module for the particular board module must take the standardized packet given to a trap function and create a new packet which is specifically tailored for the disk drive controller that receives it. Refer to documentation on the particular controller module for the format of its packets, and for using the **IOC** command.

Default PPCBug Controller and Device Parameters

PPCBug initializes the parameter tables for a default configuration of controllers. If the system needs to be configured differently than this default configuration (for example, to use a drive that is different from the default), then you must change these tables.

Use the **IOT** command to reconfigure the parameter table manually for any controller and/or device that is different from the default. This is a temporary change and is overwritten if a cold-start reset occurs.

PPCBug supports the following disk and tape controller devices. The controller address as listed in the following table is the base address for that controller. The controller can be addressed by the CLUN during these commands or system calls:

Commands	System Calls
PBOOT	.DSKRD
IOP	.DSKWR

If the system board SCSI port is used, the system board has CLUN0.

Table 7-1. Disk and Tape Controllers Supported

CLUN	Controller	Controller Address	PowerPC Board
0	SYM 53C810/825A	$80801000	SCSI controller
1	PC8477	$800003F0	Floppy disk controller

Disk I/O Error Codes

PPCBug returns an error code if an attempted disk operation is unsuccessful. Refer to the *PPCBug Debugging Package User's Manual* for an explanation of disk I/O error codes.

Network I/O Support

The Network Boot Firmware provides the capability to boot the CPU through the ROM debugger using a network (local Ethernet interface) as the boot device.

The booting process is executed in two distinct phases:

- The first phase allows the diskless remote node to discover its network identify and the name of the file to be booted.
- The second phase has the diskless remote node reading the boot file across the network into its memory.

Figure 5-1 depicts the various modules (capabilities) and the dependencies of these modules that support the overall network boot function. They are described in the following paragraphs.

Physical Layer Manager Ethernet Driver

This driver manages/surrounds the Ethernet controller chip or module. Management is in the scope of the reception of packets, the transmission of packets, receive buffer flushing, and interface initialization.

This module ensures that the packaging and unpackaging of Ethernet packets is done correctly in the Boot PROM.

UDP/IP Modules

The Internet Protocol (IP) is designed for use in interconnected systems of packet-switched computer communication networks. The Internet Protocol provides for transmitting of blocks of data called datagrams (hence User Datagram Protocol, or UDP) from sources to destinations, where sources and destinations are hosts identified by fixed length addresses.

The UDP and IP protocols are necessary for the TFTP and BOOTP protocols, TFTP and BOOTP require a UDP/IP connection.

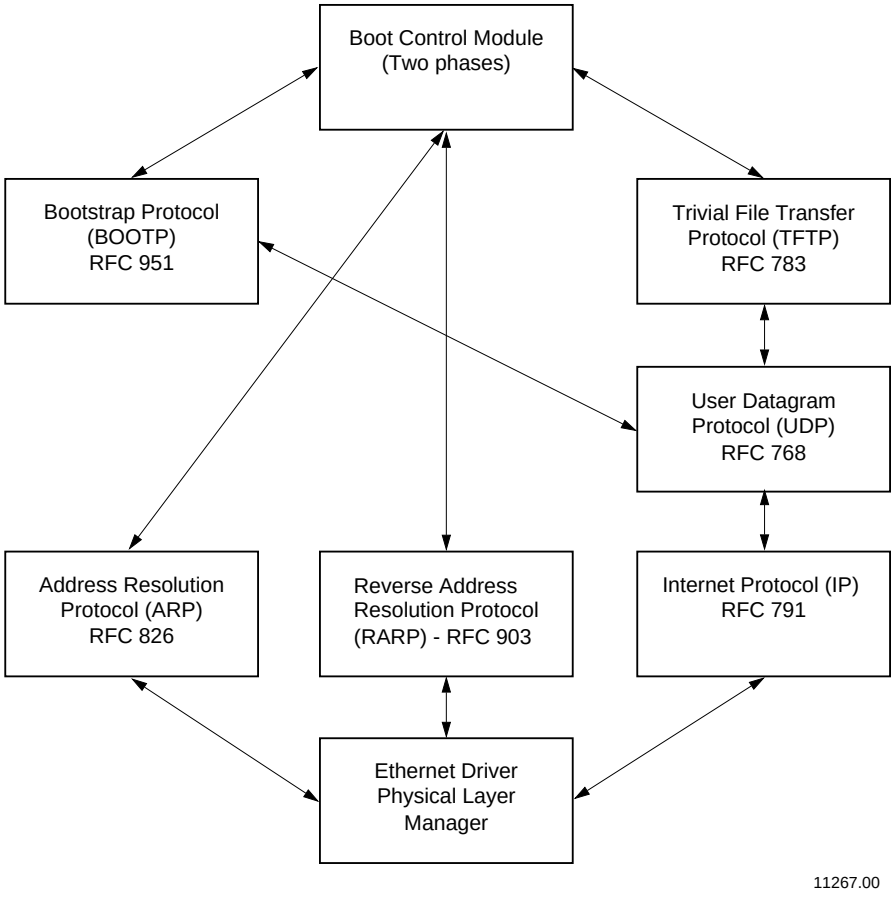


Figure 7-1. Network Boot Modules

RARP/ARP Modules

The Reverse Address Resolution Protocol (RARP) basically consists of an identity-less node broadcasting a “whoami” packet onto the Ethernet, and waiting for an answer. The RARP server fills an Ethernet reply packet up with the target's Internet Address and sends it.

The Address Resolution Protocol (ARP) basically provides a method of converting protocol addresses (e.g., IP addresses) to local area network addresses (e.g., Ethernet addresses). The RARP protocol module supports systems which do not support the BOOTP protocol (refer to the following section, *BOOTP Module*).

BOOTP Module

The Bootstrap Protocol (BOOTP) basically allows a diskless client machine to discover its own IP address, the address of a server host, and the name of a file to be loaded into memory and executed.

TFTP Module

The Trivial File Transfer Protocol (TFTP) is a simple protocol to transfer files. It is implemented on top of the Internet User Datagram Protocol (UDP or Datagram) so it may be used to move files between machines on different networks implementing UDP. The only thing it can do is read and write files from/to a remote server.

Network Boot Control Module

The "control" capability of the Network Boot Control Module is needed to tie together all the necessary modules (capabilities) and to sequence the booting process. The booting sequence consists of two phases:

Phase	Label	Capability Used
1	Address determination and bootfile selection	RARP/BOOTP
2	File transfer	TFTP

Network I/O Error Codes

PPCBug returns an error code if an attempted network operation is unsuccessful. Refer to the *PPCBug Debugging Package User's Manual* for an explanation of network I/O error codes.

Ordering Related Documentation

A

Motorola Computer Group Documents

The publications listed below are on related products, and some may be referenced in this document. Manuals not shipped with this product may be purchased by contacting your local Motorola sales office

Please note that exact titles and part numbers of the documents are subject to change without notice..

Table A-1. Motorola Computer Group Documents

Document Title	Publication Number
PPCBug Firmware Package User's Manual (Parts 1 and 2)	PPCBUGA1/UM PPCBUGA2/UM
PPC1Bug Diagnostics Manual	PPC1DIAA/UM
Ultra Plus and Ultra 60*x* Installation and Use (this manual)	ULMB60XA/IH
Ultra 603/Ultra 603e/Ultra 604 Programmer's Reference Guide	ULMB60XA/PG

Notes Although not shown in the above list, each Motorola Computer Group manual publication number is suffixed with characters that represent the revision level of the document, such as "/*xx*2" (the second revision of a manual); a supplement bears the same number as the manual but has a suffix such as "/*xx*2A1" (the first supplement to the second revision of the manual).

The above documents can also be purchased as a set under part number **LK-UB60X**.

Manufacturers' Documents

For additional information, refer to the following table for manufacturers' data sheets or user's manuals. As an additional help, a source for the listed document is also provided. Please note that in many cases, the information is preliminary and the revision levels of the documents is subject to change without notice.

To further assist your development effort, Motorola has collected some of the non-Motorola documents in this list from the suppliers. This bundle can be ordered as part number **68-PCIKIT**.

Table A-2. Manufacturers' Documents

Document Title and Source	Publication Number
PowerPC 603™ RISC Microprocessor Technical Summary Motorola Literature and Printing Distribution Services P.O. Box 20924 Phoenix, Arizona 85036-0924 Telephone: (602) 994-6561 FAX: (602) 994-6430	MPC603/D
PowerPC 603™ RISC Microprocessor User's Manual Motorola Literature and Printing Distribution Services P.O. Box 20924 Phoenix, Arizona 85036-0924 Telephone: (602) 994-6561 FAX: (602) 994-6430 OR IBM Microelectronics Mail Stop A25/862-1 PowerPC Marketing 1000 River Street Essex Junction, Vermont 05452-4299 Telephone: 1-800-PowerPC Telephone: 1-800-769-3772 FAX: 1-800-POWERfax FAX: 1-800-769-3732	MPC603UM/AD MPR603UMU-01

Table A-2. Manufacturers' Documents (Continued)

Document Title and Source	Publication Number
PowerPC 604™ RISC Microprocessor User's Manual Motorola Literature and Printing Distribution Services P.O. Box 20924 Phoenix, Arizona 85036-0924 Telephone: (602) 994-6561 FAX: (602) 994-6430 OR IBM Microelectronics Mail Stop A25/862-1 PowerPC Marketing 1000 River Street Essex Junction, Vermont 05452-4299 Telephone: 1-800-PowerPC Telephone: 1-800-769-3772 FAX: 1-800-POWERfax FAX: 1-800-769-3732	MPC604UM/AD MPR604UMU-01
MPC105 PCI Bridge/Memory Controller User's Manual Motorola Literature and Printing Distribution Services P.O. Box 20924 Phoenix, Arizona 85036-0924 Telephone: (602) 994-6561 FAX: (602) 994-6430	MPC105UM/AD
PowerPC™ Microprocessor Family: The Programming Environments Motorola Literature and Printing Distribution Services P.O. Box 20924 Phoenix, Arizona 85036-0924 Telephone: (602) 994-6561 FAX: (602) 994-6430 OR IBM Microelectronics Mail Stop A25/862-1 PowerPC Marketing 1000 River Street Essex Junction, Vermont 05452-4299 Telephone: 1-800-PowerPC Telephone: 1-800-769-3772 FAX: 1-800-POWERfax FAX: 1-800-769-3732	MPCFPE/AD MPRPPCFPE-01

Table A-2. Manufacturers' Documents (Continued)

Document Title and Source	Publication Number
Alpine™ VGA Family - CL-GD543X/'4X Technical Reference Manual Fourth Edition Cirrus Logic, Inc. (or nearest Sales Office) 3100 West Warren Avenue Fremont, California 94538-6423 Telephone: (510) 623-8300 FAX: (510) 252-6020	385439-004
DECchip 21040 Ethernet LAN Controller for PCI Hardware Reference Manual Digital Equipment Corporation Maynard, Massachusetts DECchip Information Line Telephone (United States and Canada): 1-800-332-2717 TTY (United States only): 1-800-332-2515 Telephone (outside North America): +1-508-568-6868	EC-N0752-72
PC87303VUL (Super I/O™ Sidewinder Lite) Floppy Disk Controller, Keyboard Controller, Real-Time Clock, Dual UARTs, IEEE 1284 Parallel Port, and IDE Interface National Semiconductor Corporation Customer Support Center (or nearest Sales Office) 2900 Semiconductor Drive P.O. Box 58090 Santa Clara, California 95052-8090 Telephone: 1-800-272-9959	PC87303VUL
PC87323VF (Super I/O™ Sidewinder) Floppy Disk Controller, Keyboard Controller, Real-Time Clock, Dual UARTs, IEEE 1284 Parallel Port, and IDE Interface National Semiconductor Corporation Customer Support Center (or nearest Sales Office) 2900 Semiconductor Drive P.O. Box 58090 Santa Clara, California 95052-8090 Telephone: 1-800-272-9959	PC87323VF
M48T18 CMOS 8K x 8 TIMEKEEPER™ SRAM Data Sheet SGS-Thomson Microelectronics Group Marketing Headquarters (or nearest Sales Office) 1000 East Bell Road Phoenix, Arizona 85022 Telephone: (602) 867-6100	M48T18

Table A-2. Manufacturers' Documents (Continued)

Document Title and Source	Publication Number
DS1643 Nonvolatile Timekeeping RAM Data Manual Dallas Semiconductor 4401 South Beltwood Parkway Dallas, Texas 75244-3292	DS1643/DS1643LPM
82378 System I/O (SIO) PCI-to-ISA Bridge Controller Intel Corporation Literature Sales P.O. Box 7641 Mt. Prospect, Illinois 60056-7641 Telephone: 1-800-548-4725	290473-003
SYM 53C810 (was NCR 53C810) PCI-SCSI I/O Processor Data Manual Symbios Logic Inc 1731 Technology Drive, suite 600 San Jose, CA 95110 Telephone: (408) 441-1080 Hotline: 1-800-334-5454	T46923I
SYM 53C825 (was NCR 53C825) PCI-SCSI I/O Processor with Local ROM Interface Data Manual Symbios Logic Inc 1731 Technology Drive, suite 600 San Jose, CA 95110 Telephone: (408) 441-1080 Hotline: 1-800-334-5454	T40931I
SYM 53C8XX (was NCR 53C8XX) Family PCI-SCSI I/O Processors Programming Guide Symbios Logic Inc 1731 Technology Drive, suite 600 San Jose, CA 95110 Telephone: (408) 441-1080 Hotline: 1-800-334-5454	J10931I
SCC (Serial Communications Controller) User's Manual (for Z85230 and other Zilog parts) Zilog, Inc. 210 East Hacienda Ave., mail stop C1-0 Campbell, California 95008-6600 Telephone: (408) 370-8016 FAX: (408) 370-8056	DC-8293-02

Table A-2. Manufacturers' Documents (Continued)

Document Title and Source	Publication Number
Z8536 CIO Counter/Timer and Parallel I/O Unit Product Specification and User's Manual (in Z8000® Family of Products Data Book) Zilog, Inc. 210 East Hacienda Ave., mail stop C1-0 Campbell, California 95008-6600 Telephone: (408) 370-8016 FAX: (408) 370-8056	DC-8319-00
CS4231 Parallel Interface, Multimedia Audio Codec Data Sheet Crystal Semiconductor Corporation 4210 South Industrial Drive P.O. Box 17847 Austin, Texas 78744-7847 Telephone: 1-800-888-5016 Telephone: (512) 445-7222 FAX: (512) 445-7581	DS111PP4
CSB4231/4248 Evaluation Board Data Sheet Crystal Semiconductor Corporation 4210 South Industrial Drive P.O. Box 17847 Austin, Texas 78744-7847 Telephone: 1-800-888-5016 Telephone: (512) 445-7222 FAX: (512) 445-7581	DS111DB4

Related Specifications

For additional information, refer to the following table for related specifications. Sources for the listed documents are also provided. Please note that in many cases, the information is preliminary and the revision levels of the documents is subject to change without notice.

Table A-3. Related Specifications

Document Title and Source	Publication Number
ANSI Small Computer System Interface-2 (SCSI-2), Draft Document Global Engineering Documents 15 Inverness Way East Englewood, CO 80112-5704 Telephone: 1-800-854-7179 Telephone: (303) 792-2181	ANSI X3.131.1990
ANSI Std X3T9.2, 1994 AT Attachment Interface for Disk Drives Global Engineering Documents 15 Inverness Way East Englewood, CO 80112-5704 Telephone: 1-800-854-7179 Telephone: (303) 792-2181	ANSI X3.221
Bidirectional Parallel Port Interface Specification Institute of Electrical and Electronics Engineers, Inc. Publication and Sales Department 345 East 47th Street New York, New York 10017-21633 Telephone: 1-800-678-4333	IEEE Standard 1284
IEEE - Common Mezzanine Card Specification (CMC) Institute of Electrical and Electronics Engineers, Inc. Publication and Sales Department 345 East 47th Street New York, New York 10017-21633 Telephone: 1-800-678-4333	P1386 Draft 1.3

Table A-3. Related Specifications (Continued)

Document Title and Source	Publication Number
IEEE - PCI Mezzanine Card Specification (PMC) Institute of Electrical and Electronics Engineers, Inc. Publication and Sales Department 345 East 47th Street New York, New York 10017-21633 Telephone: 1-800-678-4333	P1386.1 Draft 2.0
IEEE Standard for Local Area Networks: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications Institute of Electrical and Electronics Engineers, Inc. Publication and Sales Department 345 East 47th Street New York, New York 10017-21633 Telephone: 1-800-678-4333	IEEE 802.3
Information Technology - Local and Metropolitan Networks - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications Global Engineering Documents 15 Inverness Way East Englewood, CO 80112-5704 Telephone: 1-800-854-7179 Telephone: (303) 792-2181 *(This document can also be obtained through the national standards body of member countries.*	ISO/IEC 8802-3
Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange (EIA-232-D) Electronic Industries Association Engineering Department 2001 Eye Street, N.W. Washington, D.C. 20006	ANSI/EIA-232-D Standard

Table A-3. Related Specifications (Continued)

Document Title and Source	Publication Number
Peripheral Component Interconnect (PCI) Local Bus Specification, Revision 2.0 PCI Special Interest Group P.O. Box 14070 Portland, Oregon 97214-4070 Marketing/Help Line Telephone: (503) 696-6111 Document/Specification Ordering Telephone: 1-800-433-5177 Telephone: (503) 797-4207 FAX: (503) 234-6762	PCI Local Bus Specification
PowerPC Reference Platform Specification, Third Edition, Version 1.0, Volumes I and II International Business Machines Corporation Power Personal Systems Architecture 11400 Burnet Road Austin, Texas 78758-3493 Document/Specification Ordering Telephone: 1-800-POWERPC Telephone: 1-800-769-3772 Telephone: (708) 296-9332	MPR-PPC-RPU-02

Technical Data B

Introduction

This appendix lists interconnection signals for components on the Ultra Plus and Ultra 60*x* system board. Refer to Figure 2-1, *User/Field Replaceable Components Location Diagram*, for the location of these items. Also in this appendix are I/O memory maps and interrupt assignments.

Interconnect Signals

Power Connector, P1

Table B-1. Power Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	POWEROK	Power supply operating
2	+5V DC	+5 volts DC
3	+12V DC	+12 volts DC
4	-12V DC	-12 volts DC
5	GND	Ground
6	GND	Ground
7	GND	Ground
8	GND	Ground

Table B-1. Power Connector Pin Assignments (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
9	-5V DC	-5 volts DC
10	+5V DC	+5 volts DC
11	+5V DC	+5 volts DC
12	+5V DC	+5 volts DC

External Reset/Abort Switch Connector, J1

Table B-2. External Reset Switch Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	GND	Ground
2	RESET*	Reset
3	GND	Ground
4	ABORT*	Abort (IRQ 8)

Internal Speaker Connector, J2

Table B-3. Internal Speaker Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	Speaker +	Speaker +
2	Speaker -	Speaker -
3	Speaker +	Speaker +
4	Speaker -	Speaker -

RISCWatch Connector, J3

Table B-4. RISCWatch Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	TDO	Test data out
2	NC	Not connected
3	TDI	Test data in
4	TRST*	Test reset
5	NC	Not connected
6	+3.3V DC	+3.3 volts DC (pulled up)
7	TCK	Test clock
8	NC	Not connected
9	TMS	Test mode select
10	NC	Not connected
11	SRESET*	Soft reset
12	NC	Not connected
13	HRESET*	Hard reset
14	KEY	Connector key (no pin)
15	CKSTPO*	Check stop
16	GND	Ground

IDE Disk Drive Connector, J4

Table B-5. IDE Disk Drive Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	RST-	Reset
2	GND	Ground
3	DD7	IDE data bus (bit 7)
4	DD8	IDE data bus (bit 8)
5	DD6	IDE data bus (bit 6)
6	DD9	IDE data bus (bit 9)
7	DD5	IDE data bus (bit 5)
8	DD10	IDE data bus (bit 10)
9	DD4	IDE data bus (bit 4)
10	DD11	IDE data bus (bit 11)
11	DD3	IDE data bus (bit 3)
12	DD12	IDE data bus (bit 12)
13	DD2	IDE data bus (bit 2)
14	DD13	IDE data bus (bit 13)
15	DD1	IDE data bus (bit 1)
16	DD14	IDE data bus (bit 14)
17	DD0	IDE data bus (bit 0)
18	DD15	IDE data bus (bit 15)
19	GND	Ground
20	NC	Not connected
21	DMARQ	DMA request 5
22	GND	Ground
23	DIOW-	Write
24	GND	Ground
25	DIOR-	Read
26	GND	Ground
27	NC	Not connected
28	ALE	Address latch enable

Table B-5. IDE Disk Drive Connector Pin Assignments (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
29	DMACK-	DMA acknowledge 5
30	GND	Ground
31	INTRQ	Interrupt request IRQ 14
32	IOCS16-	IO chip select 16
33	DA1	Address bus (bit 1)
34	NC	Not connected
35	DA0	Address bus (bit 0)
36	DA2	Address bus (bit 2)
37	CS0-	Hard disk chip select (bit 0)
38	CS1-	Hard disk chip select (bit 1)
39	DASP-	Device active LED/slave present
40	GND	Ground

Fan Power Connector, J5

Table B-6. Fan Power Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	GND	Ground
2	+12V DC Fused	+12 volts DC fused
3	GND	Ground

Fast SCSI Connector, J6

Table B-7. 8-Bit SCSI Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	GND	Ground
2	SCDB0-	SCSI data bus (bit 0)
3	GND	Ground
4	SCDB1-	SCSI data bus (bit 1)
5	GND	Ground
6	SCDB2-	SCSI data bus (bit 2)
7	GND	Ground
8	SCDB3-	SCSI data bus (bit 3)
9	GND	Ground
10	SCDB4-	SCSI data bus (bit 4)
11	GND	Ground
12	SCDB5-	SCSI data bus (bit 5)
13	GND	Ground
14	SCDB6-	SCSI data bus (bit 6)
15	GND	Ground
16	SCDB7-	SCSI data bus (bit 7)
17	GND	Ground
18	SCDP-	SCSI data parity
19	GND	Ground
20	GND	Ground
21	GND	Ground
22	GND	Ground
23	GND	Ground
24	GND	Ground
25	NC	Not connected
26	TERMPWR	Terminator power; +5 V power for external termination
27	GND	Ground
28	GND	Ground

Table B-7. 8-Bit SCSI Connector Pin Assignments (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
29	GND	Ground
30	GND	Ground
31	GND	Ground
32	ATN-	Attention
33	GND	Ground
34	GND	Ground
35	GND	Ground
36	BSY-	Busy
37	GND	Ground
38	ACK-	Acknowledge
39	GND	Ground
40	RESET-	Reset
41	GND	Ground
42	MSG-	Message
43	GND	Ground
44	SEL-	Select
45	GND	Ground
46	CD-	Command/data
47	GND	Ground
48	REQ-	Request
49	GND	Ground
50	I/O-	Input/output

B

Fast/Wide SCSI Connector, J7

Table B-8. 16-Bit SCSI Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	GND	Ground
2	GND	Ground
3	GND	Ground
4	GND	Ground
5	GND	Ground
6	GND	Ground
7	GND	Ground
8	GND	Ground
9	GND	Ground
10	GND	Ground
11	GND	Ground
12	GND	Ground
13	GND	Ground
14	GND	Ground
15	GND	Ground
16	GND	Ground
17	TERMPWR	Terminal power: +5 V power for external termination
18	TERMPWR	Terminal power: +5 V power for external termination
19		Reserved
20	GND	Ground
21	GND	Ground
22	GND	Ground

Pin Number	Signal Mnemonic	Signal Name and Description
35	-DB(12)	SCSI data bus (bit 12)
36	-DB(13)	SCSI data bus (bit 13)
37	-DB(14)	SCSI data bus (bit 14)
38	-DB(15)	SCSI data bus (bit 15)
39	-DB(P1)	SCSI data bus (P1)
40	-DB(0)	SCSI data bus (bit 0)
41	-DB(1)	SCSI data bus (bit 1)
42	-DB(2)	SCSI data bus (bit 2)
43	-DB(3)	SCSI data bus (bit 3)
44	-DB(4)	SCSI data bus (bit 4)
45	-DB(5)	SCSI data bus (bit 5)
46	-DB(6)	SCSI data bus (bit 6)
47	-DB(7)	SCSI data bus (bit 7)
48	-DB(P)	SCSI data bus (P)
49	GND	Ground
50	GND	Ground
51	TERMPWR	Terminal power; +5 V power for external termination
52	TERMPWR	Terminal power: +5 V power for external termination
53		Reserved
54	GND	Ground
55	-ATN	Attention
56	GND	Ground

Table B-8. 16-Bit SCSI Connector Pin Assignments (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
23	GND	Ground
24	GND	Ground
25	GND	Ground
26	GND	Ground
27	GND	Ground
28	GND	Ground
29	GND	Ground
30	GND	Ground
31	GND	Ground
32	GND	Ground
33	GND	Ground
34	GND	Ground

Pin Number	Signal Mnemonic	Signal Name and Description
57	-BSY	Busy
58	-ACK	Acknowledge
59	-RST	Reset
60	-MSG	Message
61	-SEL	Select
62	-C/D	Command/data
63	-REQ	Request
64	-I/O	Input/output
65	-DB(8)	SCSI data bus (bit 8)
66	-DB(9)	SCSI data bus (bit 9)
67	-DB(10)	SCSI data bus (bit 10)
68	-DB(11)	SCSI data bus (bit 11)

Floppy Diskette Drive Connector, J8

Table B-9. Floppy Disk Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	GND	Ground
2	DENSEL	Density select
3	GND	Ground
4	NC	Not connected
5	GND	Ground
6	DRATE0	Data rate 0
7	NC	Not connected
8	INDEX*	Index pulse sense
9	GND	Ground

Table B-9. Floppy Disk Connector Pin Assignments (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
10	MTR0*	Motor 0 select
11	GND	Ground
12	DR1*	Drive 1 select
13	NC	Not connected
14	DR0*	Drive 0 select
15	GND	Ground
16	MTR1*	Motor 1 select
17	MSEN1	Media sense 1
18	DIR*	Step direction control
19	GND	Ground
20	STEP*	Step pulse
21	GND	Ground
22	WDATA*	Write data output
23	GND	Ground
24	WGATE*	Write gate control
25	GND	Ground
26	TRK0*	Track 0 sense
27	MSEN0	Media sense 0
28	WP*	Write protect sense
29	GND	Ground
30	RDATA*	Read data input
31	GND	Ground
32	HDSEL*	Hard select control
33	GND	Ground
34	DSKCHG*	Disk change detect input

Cache/Processor Direct Slot (PDS) Expansion Socket, J9

Table B-10. Cache/PDS Card Connector Pin Assignments, Pins 1 through 96

Pin Number	Signal Mnemonic	Signal Name and Description
1	A0	Address bus (bit 0) -- Most significant bit
2	A1	Address bus (bit 1)
3	A2	Address bus (bit 2)
4	A3	Address bus (bit 3)
5	A4	Address bus (bit 4)
6	A5	Address bus (bit 5)
7	A6	Address bus (bit 6)
8	A7	Address bus (bit 7)
9	A8	Address bus (bit 8)
10	A9	Address bus (bit 9)
11	A10	Address bus (bit 10)
12	A11	Address bus (bit 11)
13	A12	Address bus (bit 12)
14	A13	Address bus (bit 13)
15	A14	Address bus (bit 14)
16	A15	Address bus (bit 15)
17	A16	Address bus (bit 16)
18	A17	Address bus (bit 17)
19	A18	Address bus (bit 18)
20	A19	Address bus (bit 19)
21	A20	Address bus (bit 20)
22	A21	Address bus (bit 21)
23	A22	Address bus (bit 22)
24	A23	Address bus (bit 23)
25	A24	Address bus (bit 24)
26	A25	Address bus (bit 25)
27	A26	Address bus (bit 26)

Table B-10. Cache/PDS Card Connector Pin Assignments, Pins 1 through 96 (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
28	A27	Address bus (bit 27)
29	A28	Address bus (bit 28)
30	A29	Address bus (bit 29)
31	A30	Address bus (bit 30)
32	A31	Address bus (bit 31) -- Least significant bit
33	AP0	Address bus parity 0
34	AP1	Address bus parity 1
35	AP2	Address bus parity 2
36	AP3	Address bus parity 3
35	APE*	Address bus error
38	RSRV*	Reservation
39	DH0	Data bus high 0 -- Most significant bit
40	DH1	Data bus high 1
41	DH2	Data bus high 2
42	DH3	Data bus high 3
43	DH4	Data bus high 4
44	DH5	Data bus high 5
45	DH6	Data bus high 6
46	DH7	Data bus high 7
47	DH8	Data bus high 8
48	DH9	Data bus high 9
49	DH10	Data bus high 10
50	DH11	Data bus high 11
51	DH12	Data bus high 12
52	DH13	Data bus high 13
53	DH14	Data bus high 14
54	DH15	Data bus high 15
55	DH16	Data bus high 16
56	DH17	Data bus high 17

Table B-10. Cache/PDS Card Connector Pin Assignments, Pins 1 through 96 (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
57	DH18	Data bus high 18
58	DH19	Data bus high 19
59	DH20	Data bus high 20
60	DH21	Data bus high 21
61	DH22	Data bus high 22
62	DH23	Data bus high 23
63	DH24	Data bus high 24
64	DH25	Data bus high 25
65	DH26	Data bus high 26
66	DH27	Data bus high 27
67	DH28	Data bus high 28
68	DH29	Data bus high 29
69	DH30	Data bus high 30
70	DH31	Data bus high 31 -- Least significant bit
71	DL0	Data bus low 0 -- Most significant bit
72	DL1	Data bus low 1
73	DL2	Data bus low 2
74	DL3	Data bus low 3
75	DL4	Data bus low 4
76	DL5	Data bus low 5
77	DL6	Data bus low 6
78	DL7	Data bus low 7
79	DL8	Data bus low 8
80	DL9	Data bus low 9
81	DL10	Data bus low 10
82	DL11	Data bus low 11
83	DL12	Data bus low 12
84	DL13	Data bus low 13
85	DL14	Data bus low 14

Table B-10. Cache/PDS Card Connector Pin Assignments, Pins 1 through 96 (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
86	DL15	Data bus low 15
87	DL16	Data bus low 16
88	DL17	Data bus low 17
89	DL18	Data bus low 18
90	DL19	Data bus low 19
91	DL20	Data bus low 20
92	DL21	Data bus low 21
93	DL22	Data bus low 22
94	DL23	Data bus low 23
95	DL24	Data bus low 24
96	DL25	Data bus low 25

Table B-11. Cache/PDS Card Connector Pin Assignments, Pins 97 through 190

Pin Number	Signal Mnemonic	Signal Name and Description
97	DL26	Data bus low
98	DL27	Data bus low
99	DL28	Data bus low
100	DL29	Data bus low
101	DL30	Data bus low
102	DL31	Data bus low -- Least significant bit
103	DP0	Data bus parity 0
104	DP1	Data bus parity 1
105	DP2	Data bus parity 2
106	DP3	Data bus parity 3
107	DP4	Data bus parity 4

Table B-11. Cache/PDS Card Connector Pin Assignments, Pins 97 through 190 (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
108	DP5	Data bus parity 5
109	DP6	Data bus parity 6
110	DP7	Data bus parity 7
111	PDSCLK4	66 MHz clock 4
112	PDSCLK6	66 MHz clock 6
113	DPE*	Data parity error
114	DBDIS*	Data bus disable
115	TT0	Transfer type 0
116	TSIZ0	Transfer size 0
117	TT1	Transfer type 1
118	TSIZ1	Transfer size 1
119	TT2	Transfer type 2
120	TSIZ2	Transfer size 2
121	TT3	Transfer type 3
122	TC0	Transfer code 0
123	TT4	Transfer type 4
124	TC1	Transfer code 1
125	CI*	Cache inhibit
126	TC2	Transfer code 2
127	WT*	Write-through
128	CSE0	Cache set entry 0
129	GBL*	Global
130	CSE1	Cache set entry 1
131	SHD*	Shared
132	DBWO*	Data bus write only
133	AACK*	Address acknowledge
134	TS*	Transfer start
135	ARTRY*	Address retry
136	XATS*	Extended address transfer start

Table B-11. Cache/PDS Card Connector Pin Assignments, Pins 97 through 190 (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
137	DRTRY*	Data retry
138	TBST	Transfer burst
139	TA*	Transfer acknowledge
140	PDSCLK5	66 MHz clock 5
141	TEA*	Transfer error acknowledge
142	DSPARE5	Reserved
143	DSPARE2	Reserved
144	CPUDBG*	CPU data bus grant
145	DSPARE4	Reserved
146	DBB*	Data bus busy
147	DSPARE6	Reserved
148	ABB*	Address bus busy
149	TCLKO	Test clock out
150	CPUBG*	CPU bus grant
151	CACHESIZ0	Cache size 0
152	CPUBR*	CPU bus request
153	L2ADSC*	L2 cache address strobe control
154	CPUINT*	CPU interrupt
155	L2BAA*	L2 cache burst address advance
156	MCP*	Machine check interrupt
157	CACHEBR*	Cache bus request
158	SMI*	System management interrupt
159	CACHEBG*	Cache bus grant
160	CKSTPI*	Checkstop in
161	L2DOE*	L2 cache data RAM output enable
162	CKSTPO*	Checkstop out
163	L2DWE*	L2 cache data RAM write enable
164	HALTED	Processor halted (604 only)
165	L2HIT*	L2 cache hit

Table B-11. Cache/PDS Card Connector Pin Assignments, Pins 97 through 190 (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
166	TLBISYNC*	TLBI synchronize (603 only)
167	L2TALE	L2 cache tag address latch
168	TBEN	Timebase enable
169	L2TALOE*	L2 cache tag address latch output enable
170	SUSPEND*	Suspend
171	L2TOE*	L2 cache tag output enable
172	DRVMOD0	Drive mode 0 (604 only)
173	L2TWE*	L2 cache tag write enable
174	DRVMOD1	Drive mode 1 (604 only)
175	L2TV	L2 cache tag valid
176	NAPRUN	Power management in Nap Mode (604 only)
177	CACHESIZ1	Cache size 1
178	QREQ*	Quiescent request
179	SRESET*	Soft reset
180	QACK*	Quiescent acknowledge
181	HRESET*	Hard reset
182	TDO	Test data out
183	GND	Ground
184	TDI	Test data in
185	PDSCLK1	66 MHz clock 1
186	TCK	Test clock
187	PDSCLK2	66 MHz clock 2
188	TMS	Test mode select
189	PDSCLK3	66 MHz clock 3
190	TRST*	Test reset

Table B-12. Cache/PDS Card Connector Pin Assignments, Pins S1 through S25

Pin Number	Signal Mnemonic	Signal Name and Description
S1	GND	Ground
S2	GND	Ground
S3	GND	Ground
S4	GND	Ground
S5	GND	Ground
S6	+5V DC	+5 volts DC
S7	+5V DC	+5 volts DC
S8	+5V DC	+5 volts DC
S9	+5V DC	+5 volts DC
S10	+5V DC	+5 volts DC
S11	GND	Ground
S12	GND	Ground
S13	GND	Ground
S14	GND	Ground
S15	GND	Ground
S16	+3.3V DC	+3.3 volts DC
S17	+3.3V DC	+3.3 volts DC
S18	+3.3V DC	+3.3 volts DC
S19	+3.3V DC	+3.3 volts DC
S20	+3.3V DC	+3.3 volts DC
S21	GND	Ground
S22	GND	Ground
S23	GND	Ground
S24	GND	Ground
S25	GND	Ground

VESA Advanced Feature Connector (VAFC), J10

Table B-13. VAFC Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	GND	Ground
2	PIXEL 0	Pixel bus 0
3	GND	Ground
4	PIXEL 1	Pixel bus 1
5	GND	Ground
6	PIXEL 2	Pixel bus 2
7	EVIDEO*	Enable video
8	PIXEL 3	Pixel bus 3
9	ESYNC*	Enable sync and blank
10	PIXEL 4	Pixel bus 4
11	EDOTCLK*	Enable dot clock
12	PIXEL 5	Pixel bus 5
13	VCC/NC	No connect (currently)
14	PIXEL 6	Pixel bus 6
15	GND	Ground
16	PIXEL 7	Pixel bus 7
17	GND	Ground
18	DOTCLK	Dot clock
19	GND	Ground
20	BLANK*	Blank
21	GND	Ground
22	HSYNC	Horizontal sync
23	RMCLK	Memory clock
24	VSYNC	Vertical sync
25	OVRW*	Overlay window
26	GND	Ground

Audio Connector, J11

Table B-14. CD-ROM Audio Input Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	GND	Audio ground
2	LCDIN	Left CD-ROM (audio) input
3	GND	Audio ground
4	RCDIN	Right CD-ROM (audio) input

PCI/ISA Riser Card Connector, J12

Table B-15. Riser Card Connector Pin Assignments, Row A

Pin Number	Signal Mnemonic	Signal Name and Description
A1	IOCHK*	I/O clock
A2	SD7	ISA bus data bus (bit 7)
A3	SD6	ISA bus data bus (bit 6)
A4	SD5	ISA bus data bus (bit 5)
A5	SD4	ISA bus data bus (bit 4)
A6	SD3	ISA bus data bus (bit 3)
A7	SD2	ISA bus data bus (bit 2)
A8	SD1	ISA bus data bus (bit 1)
A9	SD0	ISA bus data bus (bit 0)
A10	IOCHRDY	ISA bus I/O channel ready
A11	AEN	Address enable
A12	SA19	ISA bus system address bus (bit 19)
A13	SA18	ISA bus system address bus (bit 18)

Table B-15. Riser Card Connector Pin Assignments, Row A (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
A14	SA17	ISA bus system address bus (bit 17)
A15	SA16	ISA bus system address bus (bit 16)
A16	SA15	ISA bus system address bus (bit 15)
A17	SA14	ISA bus system address bus (bit 14)
A18	SA13	ISA bus system address bus (bit 13)
A19	SA12	ISA bus system address bus (bit 12)
A20	SA11	ISA bus system address bus (bit 11)
A21	SA10	ISA bus system address bus (bit 10)
A22	SA9	ISA bus system address bus (bit 9)
A23	SA8	ISA bus system address bus (bit 8)
A24	SA7	ISA bus system address bus (bit 7)
A25	SA6	ISA bus system address bus (bit 6)
A26	SA5	ISA bus system address bus (bit 5)
A27	SA4	ISA bus system address bus (bit 4)
A28	SA3	ISA bus system address bus (bit 3)
A29	SA2	ISA bus system address bus (bit 2)
A30	SA1	ISA bus system address bus (bit 1)
A31	SA0	ISA bus system address bus (bit 0)

Table B-16. Riser Card Connector Pin Assignments, Row B

Pin Number	Signal Mnemonic	Signal Name and Description
B1	GND	Ground
B2	ISARST	ISA bus reset
B3	+5V DC	+5 Volts DC
B4	IRQ9	Interrupt request 9
B5	-5V DC	-5 Volts DC
B6	DMARQ2	DMA request 2

Table B-16. Riser Card Connector Pin Assignments, Row B (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
B7	-12V DC	-12 Volts DC
B8	ZEROWS	Zero wait state select
B9	+12V DC	+12 Volts DC
B10	GND	Ground
B11	SYSMWR	System memory write
B12	SYSMRD	System memory read
B13	IOWR*	I/O write
B14	IORD*	I/O read
B15	DMACK3*	DMA acknowledge 3
B16	DMARQ3	DMA request 3
B17	DMACK1*	DMA acknowledge 1
B18	DMARQ1	DMA request 1
B19	REFRESH*	Refresh select
B20	ISACLK	ISA clock (8.33 MHz)
B21	IRQ7	Interrupt request 7
B22	IRQ6	Interrupt request 6
B23	IRQ5	Interrupt request 5
B24	IRQ4	Interrupt request 4
B25	IRQ3	Interrupt request 3
B26	DMACK2*	DMA acknowledge 2
B27	ISATC	Terminal count
B28	ALE	Address latch enable
B29	+5V DC	+5 Volts DC
B30	ISAOSC	ISA oscillator (8.33 MHz)
B31	GND	Ground

Table B-17. Riser Card Connector Pin Assignments, Row C

Pin Number	Signal Mnemonic	Signal Name and Description
C1	SBHE*	System byte high enable
C2	LA23	Latch address bus (bit 23)
C3	LA22	Latch address bus (bit 22)
C4	LA21	Latch address bus (bit 21)
C5	LA20	Latch address bus (bit 20)
C6	LA19	Latch address bus (bit 19)
C7	LA18	Latch address bus (bit 18)
C8	LA17	Latch address bus (bit 17)
C9	MEMRD*	Memory read
C10	MEMWR*	Memory write
C11	SD8	System data bus (bit 8)
C12	SD9	System data bus (bit 9)
C13	SD10	System data bus (bit 10)
C14	SD11	System data bus (bit 11)
C15	SD12	System data bus (bit 12)
C16	SD13	System data bus (bit 13)
C17	SD14	System data bus (bit 14)
C18	SD15	System data bus (bit 15)

Table B-18. Riser Card Connector Pin Assignments, Row D

Pin Number	Signal Mnemonic	Signal Name and Description
D1	MEMCS16*	Memory chip select 16
D2	IOCS16*	I/O chip select 16
D3	IRQ10	Interrupt request 10
D4	IRQ11	Interrupt request 11
D5	IRQ12	Interrupt request 12

Table B-18. Riser Card Connector Pin Assignments, Row D (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
D6	IRQ15	Interrupt request 15
D7	IRQ14	Interrupt request 14
D8	DMACK0*	DMA acknowledge 0
D9	DMARQ0	DMA request 0
D10	DMACK5*	DMA acknowledge 5
D11	DMARQ5	DMA request 5
D12	DMACK6*	DMA acknowledge 6
D13	DMARQ6	DMA request 6
D14	DMACK7*	DMA acknowledge 7
D15	DMARQ7	DMA request 7
D16	+5V DC	+5 Volts DC
D17	ISAMASTER*	ISA master select
D18	GND	Ground

Table B-19. Riser Card Connector Pin Assignments, Row E

Pin Number	Signal Mnemonic	Signal Name and Description
E1	GND	Ground
E2	GND	Ground
E3	PCIINT0*	PCI interrupt 0
E4	PCICLK4B	PCI clock 4B (33 MHz)
E5	+5V DC	+5 volts DC
E6	NC	Not connected
E7	+5V DC	+5 volts DC
E8	PCIRST*	PCI bus reset
E9	RISGNT2*	Riser grant 2
E10	RISREQ2*	Riser request 2
E11	GND	Ground

Table B-19. Riser Card Connector Pin Assignments, Row E (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
E12	PCICLK1	PCI clock 1 (33 MHz)
E13	GND	Ground
E14	AD30	Address/data bus (bit 30)
E15	+3.3V DC	+3.3 volts DC
E16	NC	Not connected
E17	+3.3V DC	+3.3 volts DC
E18	AD28	Address/data bus (bit 28)
E19	AD26	Address/data bus (bit 26)
E20	AD24	Address/data bus (bit 24)
E21	AD22	Address/data bus (bit 22)
E22	AD20	Address/data bus (bit 20)
E23	AD18	Address/data bus (bit 18)
E24	+3.3V DC	+3.3 volts DC
E25	NC	Not connected
E26	+3.3V DC	+3.3 volts DC
E27	AD16	Address/data bus (bit 16)
E28	FRAME*	Frame
E29	CBE2*	Command/byte enable 2
E30	TRDY*	Transfer ready
E31	STOP*	Stop

Table B-20. Riser Card Connector Pin Assignments, Row F

Pin Number	Signal Mnemonic	Signal Name and Description
F1	GND	Ground
F2	GND	Ground
F3	PCIINT1*	PCI interrupt 1
F4	PCIINT2*	PCI interrupt 2

Table B-20. Riser Card Connector Pin Assignments, Row F (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
F5	+5V DC	+5 volts DC
F6	NC	Not connected
F7	+5V DC	+5 volts DC
F8	PCICLK5B	PCI clock 5B
F9	GND	Ground
F10	RISGNT3*	Riser grant 3
F11	GND	Ground
F12	RISREQ3*	Riser request 3
F13	AD31	Address/data bus (bit 31)
F14	AD29	Address/data bus (bit 29)
F15	+3.3V DC	+3.3 volts DC
F16	NC	Not connected
F17	+3.3V DC	+3.3 volts DC
F18	AD27	Address/data bus (bit 27)
F19	AD25	Address/data bus (bit 25)
F20	CBE3*	Command/byte enable 3
F21	AD23	Address/data bus (bit 23)
F22	AD21	Address/data bus (bit 21)
F23	AD19	Address/data bus (bit 19)
F24	+3.3V DC	+3.3 volts DC
F25	NC	Not connected
F26	+3.3V DC	+3.3 volts DC
F27	AD17	Address/data bus (bit 17)
F28	IRDY*	Initiator ready
F29	DEVSEL*	Device select
F30	LOCK*	Lock bus
F31	PERR*	Parity error

Table B-21. Riser Card Connector Pin Assignment, Row G

Pin Number	Signal Mnemonic	Signal Name and Description
G1	RISREQ1*	Riser request 1
G2	RISGNT1*	Riser grant 1
G3	CBE1*	Command/byte enable 1
G4	PAR	Parity I/O
G5	GND	Ground
G6	NC	Not connected
G7	GND	Ground
G8	AD13	Address/data bus (bit 13)
G9	AD11	Address/data bus (bit 11)
G10	AD9	Address/data bus (bit 9)
G11	CBE0*	Command/byte enable 0
G12	AD6	Address/data bus (bit 6)
G13	AD4	Address/data bus (bit 4)
G14	AD2	Address/data bus (bit 2)
G15	NC	Not connected
G16	+5V DC	+5 volts DC
G17	+5V DC	+5 volts DC
G18	GND	Ground
G19	GND	Ground

Table B-22. Riser Card Connector Pin Assignments, Row H

Pin Number	Signal Mnemonic	Signal Name and Description
H1	SERR*	System error
H2	AD15	Address/data bus (bit 15)
H3	AD14	Address/data bus (bit 14)
H4	AD12	Address/data bus (bit 12)
H5	GND	Ground
H6	NC	Not connected
H7	GND	Ground
H8	AD10	Address/data bus (bit 10)
H9	AD8	Address/data bus (bit 8)
H10	AD7	Address/data bus (bit 7)
H11	AD5	Address/data bus (bit 5)
H12	AD3	Address/data bus (bit 3)
H13	AD1	Address/data bus (bit 1)
H14	AD0	Address/data bus (bit 0)
H15	NC	Not connected
H16	+5V DC	+5 volts DC
H17	+5V DC	+5 volts DC
H18	GND	Ground
H19	GND	Ground

Serial Port COM1 Connector, J13 (Top)

Table B-23. COM1 Serial Port Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	DCD*	Data carrier detect (Serial Port 1)
2	SIN	Serial data input (Serial Port 1)
3	SOUT	Serial data output (Serial Port 1)
4	DTR*	Data terminal ready (Serial Port 1)
5	GND	Ground
6	DSR*	Data set ready (Serial Port 1)
7	RTS*	Request to send (Serial Port 1)
8	CTS*	Clear to send (Serial Port 1)
9	RI*	Ring indicator (Serial Port 1)

Serial Port COM2, J13 (Bottom)

Table B-24. COM2 Serial Port Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	DCD*	Data carrier detect (Serial Port 2)
2	SIN	Serial data input (Serial Port 2)
3	SOUT	Serial data output (Serial Port 2)
4	DTR*	Data terminal ready (Serial Port 2)
5	GND	Ground
6	DSR*	Data set ready (Serial Port 2)
7	RTS*	Request to send (Serial Port 2)
8	CTS*	Clear to send (Serial Port 2)
9	RI*	Ring indicator (Serial Port 2)

B

Ethernet AUI Connector, J14 (Top)

Table B-25. Ethernet AUI Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
B1	GND	Ground
B2	C+	Carrier signal
B3	T+	Transmit signal
B4	GND	Ground
B5	R+	Receive signal
B6	GND	Ground
B7	NC	Not connected
B8	GND	Ground
B9	C-	Carrier return
B10	T-	Transmit return
B11	GND	Ground
B12	R-	Receive return
B13	+12V Fused	+12 volts (fused)
B14	GND	Ground
B15	NC	Not connected

Parallel I/O Printer Port, J14 (Bottom)

Table B-26. Parallel I/O Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	STB-	Strobe
2	D0	Data bus (bit 0)
3	D1	Data bus (bit 1)
4	D2	Data bus (bit 2)
5	D3	Data bus (bit 3)
6	D4	Data bus (bit 4)
7	D5	Data bus (bit 5)
8	D6	Data bus (bit 6)
9	D7	Data bus (bit 7)
10	ACK-	Acknowledge
11	BUSY	Busy
12	PE	Paper error
13	SLCT	Select
14	AFD-	Auto feed
15	ERR-	Error
16	INIT-	Initialize
17	SLIN-	Select input
18	GND	Ground
19	GND	Ground
20	GND	Ground
21	GND	Ground
22	GND	Ground
23	GND	Ground
24	GND	Ground
25	GND	Ground

Ethernet 10base-T, J15

Table B-27. Ethernet 10base-T Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	TD+	Transmit data
2	TD-	Transmit data return
3	RD+	Receive data
4	NC	Not connected
5	NC	Not connected
6	RD-	Receive data return
7	NC	Not connected
8	NC	Not connected

SVGA Graphics Port, J16

Table B-28. SVGA Graphics Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	RED	Red video signal
2	GREEN	Green video signal
3	BLUE	Blue video signal
4	PIXEL 2	Pixel bus (bit 2)
5	GND	Ground
6	GND	Ground
7	GND	Ground
8	GND	Ground
9	NC	Not connected
10	GND	Ground
11	PIXEL 0	Pixel bus (bit 0)
12	PIXEL 1	Pixel bus (bit 1)
13	HSYNC	Horizontal sync
14	VS5NC	Vertical sync
15	PIXEL 3	Pixel bus (bit 3)

Keyboard Port, J17

Table B-29. Keyboard Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	KDATA	Keyboard data
2	NC	Not connected
3	GND	Ground
4	+5V DC Fused	+5 volts DC (fused)
5	KCLK	Keyboard clock
6	NC	Not connected

Mouse Port, J18

Table B-30. Mouse Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	MDATA	Mouse data
2	NC	Not connected
3	GND	Ground
4	+5V DC Fused	+5 volts DC (fused)
5	MCLK	Mouse clock
6	NC	Not connected

Line Input Audio Jack, J19

Table B-31. Line Input Connector Pin Assignments

Part	Signal Mnemonic	Signal Name and Description
TIP	Left	Left line input
RING	Right	Right line input
SHELL	Gnd	Audio ground

Microphone Input Audio Jack, J20

Table B-32. Microphone Input Connector Pin Assignments

Part	Signal Mnemonic	Signal Name and Description
TIP	Left	Left microphone line input
RING	Right	Right microphone line input
SHELL	Gnd	Audio ground

Line Output Audio Jack, J21

Table B-33. Line Output Connector Pin Assignments

Part	Signal Mnemonic	Signal Name and Description
TIP	Left	Left line output
RING	Right	Right line output
SHELL	Gnd	Audio ground

SIMM Device Sockets, U3, U7, U11, U16

Table B-34. SIMM Device Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	GND	Ground
2	DQ0	Data I/O (bit 0)
3	DQ18	Data I/O (bit 18)
4	DQ1	Data I/O (bit 1)
5	DQ19	Data I/O (bit 19)
6	DQ2	Data I/O (bit 2)
7	DQ20	Data I/O (bit 20)
8	DQ3	Data I/O (bit 3)
9	DQ21	Data I/O (bit 21)
10	+5V DC	+5 volts DC
11	NC	Not connected
12	A0	Address bus (bit 0)
13	A1	Address bus (bit 1)
14	A2	Address bus (bit 2)
15	A3	Address bus (bit 3)
16	A4	Address bus (bit 4)
17	A5	Address bus (bit 5)
18	A6	Address bus (bit 6)
19	A10	Address bus (bit 10)
20	DQ4	Data I/O (bit 4)
21	DQ22	Data I/O (bit 22)
22	DQ5	Data I/O (bit 5)
23	DQ23	Data I/O (bit 23)
24	DQ6	Data I/O (bit 6)
25	DQ24	Data I/O (bit 24)
26	DQ7	Data I/O (bit 7)
27	DQ25	Data I/O (bit 25)

Table B-34. SIMM Device Pin Assignments (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
28	A7	Address bus (bit 7)
29	NC	Not connected
30	+5V DC	+5 volts DC
31	A8	Address bus (bit 8)
32	A9	Address bus (bit 9)
33	RAS3-	Row address strobe 3
34	RAS2-	Row address strobe 2
35	DQ26	Data I/O (bit 26)
36	DQ8	Data I/O (bit 8)
37	DQ17	Data I/O (bit 17)
38	DQ35	Data I/O (bit 35)
39	GND	Ground
40	CAS0-	Column address strobe 0
41	CAS2-	Column address strobe 2
42	CAS3-	Column address strobe 3
43	CAS1-	Column address strobe 1
44	RAS0-	Row address strobe 0
45	RAS1-	Row address strobe 1
46	NC	Not connected
47	WE-	Write enable
48	NC	Not connected
49	DQ9	Data I/O (bit 9)
50	DQ27	Data I/O (bit 27)
51	DQ10	Data I/O (bit 10)
52	DQ28	Data I/O (bit 28)
53	DQ11	Data I/O (bit 11)
54	DQ29	Data I/O (bit 29)
55	DQ12	Data I/O (bit 12)
56	DQ30	Data I/O (bit 30)
57	DQ13	Data I/O (bit 13)

Table B-34. SIMM Device Pin Assignments (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
58	DQ31	Data I/O (bit 31)
59	+5V DC	+5 volts DC
60	DQ32	Data I/O (bit 32)
61	DQ14	Data I/O (bit 14)
62	DQ33	Data I/O (bit 33)
63	DQ15	Data I/O (bit 15)
64	DQ34	Data I/O (bit 34)
65	DQ16	Data I/O (bit 16)
66	NC	Not connected
67	PD1	Presence detect 1 (SIMM type/speed information)
68	PD2	Presence detect 2 (SIMM type/speed information)
69	PD3	Presence detect 3 (SIMM type/speed information)
70	PD4	Presence detect 4 (SIMM type/speed information)
71	NC	Not connected
72	GND	Ground

Flash Device Sockets, XU1 and XU2

Table B-35. Flash Device Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	A18	Address bus (bit 18)
2	A16	Address bus (bit 16)
3	A15	Address bus (bit 15)
4	A12	Address bus (bit 12)
5	A7	Address bus (bit 7)
6	A6	Address bus (bit 6)
7	A5	Address bus (bit 5)

Table B-35. Flash Device Pin Assignments (Continued)

Pin Number	Signal Mnemonic	Signal Name and Description
8	A4	Address bus (bit 4)
9	A3	Address bus (bit 3)
10	A2	Address bus (bit 2)
11	A1	Address bus (bit 1)
12	A0	Address bus (bit 0)
13	DQ0	Data I/O (bit 0)
14	DQ1	Data I/O (bit 1)
15	DQ2	Data I/O (bit 2)
16	GND	Ground
17	DQ3	Data I/O (bit 3)
18	DQ4	Data I/O (bit 4)
19	DQ5	Data I/O (bit 5)
20	DQ6	Data I/O (bit 6)
21	DQ7	Data I/O (bit 7)
22	CS*	Chip select
23	A10	Address bus (bit 10)
24	OE*	Output enable
25	A11	Address bus (bit 11)
26	A9	Address bus (bit 9)
27	A8	Address bus (bit 8)
28	A13	Address bus (bit 13)
29	A14	Address bus (bit 14)
30	A17	Address bus (bit 17)
31	WE*/PGM*	Write enable/program
32	+5V DC	+5 volts DC

Memory Maps

This section shows the mapping of onboard resources of the system board platform as viewed from the processor bus, the Peripheral Component (PCI) (local) bus, and the Industry Standard Architecture (ISA) bus.

For additional information, refer to the *Ultra 603/Ultra 603e/Ultra 604 Programmer's Reference Guide*.

Processor Memory Space

The system board platform uses the local memory map shown in the following table, by default.

Hardware or software can reconfigure this to use an alternate map.

Table B-36. Processor View of the Memory Map

Processor Address		Size	PCI Address Generated		Definition	Notes
Start	End		Start	End		
0000 0000	7FFF FFFF	2GB			DRAM - Not forwarded to PCI Bus	
8000 0000	807F FFFF	8MB	0000 0000	007F FFFF	ISA/PCI I/O Space	1, 2, 6
8080 0000	80FF FFFF	8MB	0080 0000	00FF FFFF	PCI Configuration Space	3
8100 0000	BF7F FFFF	1GB -24MB	0100 0000	3F7F FFFF	PCI I/O Space	
BF80 0000	BFFF FFEF	8MB -16B	3F80 0000	3FFF FFEF	Reserved - Forwarded to PCI Bus	
BFFF FFF0	BFFF FFFF	16B	3FFF FFF0	3FFF FFFF	PCI IACK/Special Cycles	7
C000 0000	FEFF FFFF	1GB -16MB	0000 0000	3EFF FFFF	PCI Memory Space	
FF00 0000	FF07 FFFF	512KB			Flash Bank 0	4

Table B-36. Processor View of the Memory Map (Continued)

Processor Address		Size	PCI Address Generated		Definition	Notes
Start	End		Start	End		
FF08 0000	FF0F FFFF	512KB			Flash Bank 1	4
FF10 0000	FFEF FFFF	14MB			Reserved	4, 5
FFF0 0000	FFF7 FFFF	512KB			Flash Bank 0 Repeat	4
FFF8 0000	FFFF FFFF	512KB			Flash Bank 1 Repeat	4

Notes

1. PCI configuration accesses to CF8 (configuration address) and CFC (configuration data) are supported by the MPC105 as specified in the *Peripheral Component Interconnect (PCI) Local Bus Specification*.
2. Both contiguous and discontiguous mappings are supported by the system board platform. Refer to the *ISA I/O Space Mapping* section for more details.
3. This space is used for Direct Mapped PCI Configuration Space accesses. Refer to the *Direct Mapped PCI Configuration Space* section for more details.
4. Flash decoding repeats every 1MB for this entire 16MB range.
5. Using this address range for Flash is not recommended since future PowerPC products will redefine this 14MB area.
6. The M48T18 RTC and NVRAM device is mapped in this area. Refer to the *ISA I/O Space Mapping* section for more details.
7. A read of any byte within this 16-byte range (BFFFFFF0 through BFFFFFFF) causes a PCI IACK cycle. The data read is the IACK vector.

Direct Mapped PCI Configuration Space

This map applies to PCI configuration cycles.

Table B-37. PCI Configuration Space Map

IDSEL	Processor Address		PCI Configuration Space Address		Definition
	Start	End	Start	End	
	8080 0000	8080 07FF	0080 0000	0800 07FF	Reserved
AD11	8080 0800	8080 08FF	0080 0800	0080 08FF	PCI/ISA Configuration Registers
	8080 0900	8080 0FFF	0080 0900	0080 0FFF	Reserved
AD12	8080 1000	8080 10FF	0080 1000	0080 10FF	SYM 53C810/53C825A Configuration Registers
	8080 1100	8080 1FFF	0080 1100	0080 1FFF	Reserved
AD13	8080 2000	8080 20FF	0080 2000	0080 20FF	Unused
	8080 2100	8080 3FFF	0080 2100	0080 3FFF	Reserved
AD14	8080 4000	8080 40FF	0080 4000	0080 40FF	DEC21040 Configuration Registers
	8080 4100	8080 7FFF	0080 4100	0080 7FFF	Reserved
AD15	8080 8000	8080 80FF	0080 8000	0080 80FF	CL-GD5434/GD5436 Configuration Registers
	8080 8100	8080 FFFF	0080 8100	0080 FFFF	Reserved
AD16	8081 0000	8081 00FF	0081 0000	0081 00FF	Slot #1 Registers
	8081 0100	8081 FFFF	0081 0100	0081 FFFF	Reserved
AD17	8082 0000	8082 00FF	0082 0000	0082 00FF	Slot #2 Registers
	8082 0100	8083 FFFF	0082 0100	0083 FFFF	Reserved
AD18	8084 0000	8084 00FF	0084 0000	0084 00FF	Slot #3 Registers
	8084 0100	80FF FFFF	0084 0100	00FF FFFF	Reserved

Notes 1. Accessing Reserved space may cause unpredictable results since multiple devices may be selected.

2. This memory map, map “A”, is also referred to as the “IBM” map, and is compatible with the PowerPC Reference Platform (PRP) specification. The alternate map “B”, referred to as the “Apple” map, is not supported.

ISA I/O Space Mapping

The following table details the locations of the peripherals as located on the ISA I/O address range. Whenever possible, addresses used are in compliance with PC-industry standards.

Table B-38. ISA I/O Space Map

ISA I/O Address	Processor Address		Function	Notes
	Contiguous	Discontiguous		
0000 - 000F	8000 0000 - 8000 000F	8000 0000 - 8000 000F	PIB: DMA1 Registers and Control	2
0020 - 0021	8000 0020 - 8000 0021	8000 1000 - 8000 1001	PIB: Interrupt 1 Control and Mask	2
0040 - 0043	8000 0040 - 8000 0043	8000 2000 - 8000 2003	PIB: Timer Counter 1 Registers	2
0060	8000 0060	8000 3000	PIB: Reset Ubus IRQ12	2
0061	8000 0061	8000 3001	PIB: NMI Status and Control	2
0064	8000 0064	8000 3004	SIO: Keyboard Controller Port	3
0074	8000 0074	8000 3014	NVRAM/RTC Address STB0	
0075	8000 0075	8000 3015	NVRAM/RTC Address STB1	
0077	8000 0077	8000 3017	NVRAM/RTC Data Port	
0080 - 0090	8000 0080 - 8000 0090	8000 4000 - 8000 4010	PIB: DMA Page Registers	2
0092	8000 0092	8000 4012	PIB: Port 92 Register	2
0094 - 009F	8000 0094 - 8000 009F	8000 4014 - 8000 401F	PIB: DMA Page Registers	2
00A0 - 00A1	8000 00A0 - 8000 00A1	8000 5000 - 8000 5001	PIB: Interrupt 2 Control and Mask	2

Table B-38. ISA I/O Space Map (Continued)

ISA I/O Address	Processor Address		Function	Notes
	Contiguous	Discontiguous		
00C0 - 00CF	8000 00C0 - 8000 00CF	8000 6000 - 8000 600F	PIB: DMA2 Address Registers	2
00D0 - 00DF	8000 00D0 - 8000 00DF	8000 7000 - 8000 700F	PIB: DMA2 Control Registers	2
01F0 - 01F7	8000 01F0 - 8000 01F7	8000 F010 - 8000 F017	SIO: IDE Registers & Control	3
02F8 - 02FF	8000 02F8 - 8000 02FF	8001 7018 - 8001 701F	SIO: Serial Port 2 (COM2)	3
0370 - 0377	8000 0370 - 8000 0377	8001 B010 - 8001 B017	SIO: Secondary FDC	3
0398	8000 0398	8001 C018	SIO: Index Register	3
0399	8000 0399	8001 C019	SIO: Data Register	3
03BC - 03BF	8000 03BC - 8000 03BF	8001 D01C - 8001 D01F	SIO: Parallel Port (LPT1)	3
03F0 - 03F7	8000 03F0 - 8000 03F7	8001 F010 - 8001 F017	SIO: Primary FDC/IDE Control	3
03F8 - 03FF	8000 03F8 - 8000 03FF	8001 F018 - 8001 F01F	SIO: Serial Port 1 (COM1)	3
040A	8000 040A	8002 000A	PIB: Scatter/Gather Interrupt Status Register	2
040B	8000 040B	8002 000B	PIB: DMA1 Extended Mode Register	2
0410 - 041F	8000 0410 - 8000 041F	8002 0010 - 8002 001F	PIB: DMA Scatter/Gather Command and Status Registers	2
0420 - 042F	8000 0420 - 8000 042F	8002 1000 - 8002 100F	PIB: DMA CH0-CH3 Scatter/Gather Descriptor Table Pointers	2

Table B-38. ISA I/O Space Map (Continued)

ISA I/O Address	Processor Address		Function	Notes
	Contiguous	Discontiguous		
0430 - 043F	8000 0430 - 8000 043F	8002 1010 - 8002 101F	PIB: DMA CH5-CH7 Scatter/Gather Descriptor Table Pointers	2
0481 - 048B	8000 0481 - 8000 048B	8002 4000 - 8002 400B	PIB: DMA High Page Registers	2
04D0	8000 04D0	8002 6010	PIB: INT1 Edge Level Control	2
04D1	8000 04D1	8002 6011	PIB: INT2 Edge Level Control	2
04D6	8000 04D6	8002 6016	PIB: DMA2 Extended Mode Register	2
0800	8000 0800	8004 0000	CPU Type Register	
0801	8000 0801	8004 0001	Reserved	
0802	8000 0802	8004 0002	System Configuration	
0803	8000 0803	8004 0003	Reserved	
0804	8000 0804	8004 0004	DRAM Size Register 1	
0805	8000 0805	8004 0005	Reserved	
0806	8000 0806	8004 0006	Reserved	
0807	8000 0807	8004 0007	Power Control	
0830	8000 0830	8004 1010	Audio: Index Address Register	
0831	8000 0831	8004 1011	Audio: Index Data Register	
0832	8000 0832	8004 1012	Audio: Status Register	

Table B-38. ISA I/O Space Map (Continued)

ISA I/O Address	Processor Address		Function	Notes
	Contiguous	Discontiguous		
0833	8000 0833	8004 1013	Audio: PIO Data Register	
0C01	8000 0C01	8006 0001	PIB: Test Mode Control Port/ Shadow Register of Port 70	2, 4
0C04	8000 0C04	8006 0004	PIB: Power Control Output Port	2, 4

Notes

1. All ISA I/O locations not specified in this table are user definable.
2. These locations are internally decoded by the PIB.
3. These locations are internally decoded by the SIO.
4. These locations are undefined.

PCI Space Mapping

The system board platform uses the PCI memory map, shown in the following table, by default.

Hardware or software can reconfigure this to use an alternate map.

Table B-39. PCI View of the PCI Memory Map

PCI Address		Size	Processor Bus Address		Definition
Start	End		Start	End	
0000 0000	00FF FFFF	16MB	Not forwarded to MPU bus		PCI/ISA Memory Space
0100 0000	7FFF FFFF	2GB -16MB	Not forwarded to MPU bus		PCI Memory Space
8000 0000	FFFF FFFF	2GB	0000 0000	7FFF FFFF (see Note)	Onboard DRAM (via MPC105)
0000 0000	FFFF FFFF	4GB	Not forwarded to MPU bus		PCI/ISA I/O Space

Note This space does not appear on the processor bus, but instead appears on the memory bus.

Interrupts

The assignments of the PCI and ISA interrupts are given in this section.

PCI Interrupts

The following table describes the interrupts dedicated to PCI peripherals. The interrupt sources are connected as shown in the table.

Table B-40. Riser Card Interrupt Routing

PCI Slot	Riser Card Label		IDSEL	Configuration Address	PCI Board Interrupt Routing			
	6-Slot	4-Slot			PCI INT A	PCI INT B	PCI INT C	PCI INT D
1	J1	J4	AD16	0x8081_0000	PCIINT0	PCIINT1	PCIINT2	PCIINT2
2	J2	J3	AD17	0x8082_0000	PCIINT1	PCIINT2	PCIINT2	PCIINT0
3	J3	N/A	AD18	0x8084_0000	PCIINT2	PCIINT2	PCIINT0	PCIINT1

Note Although the PCI Specification accommodates up to four interrupts per PCI card, it is common practice that only one interrupt (PCIINTA*) is implemented. To accommodate PCI cards that may make use of multiple interrupts, the Ultra system board provides support for up to three PCI interrupts per card (with software ramifications). The interrupt routing is further delineated in the *PCI Interrupt Routing* figure in Chapter 2. The actual mapping of the PCI interrupts is performed in software via the PCI to ISA bridge (PIB). One possible mapping is provided in Table B-40, *ISA Interrupt Assignments*.

ISA Interrupts

The following table describes the interrupts dedicated to ISA peripherals. The interrupt sources are connected as shown in the table.

Table B-41. ISA Interrupt Assignments

Interrupts		Priority	Connection	Hardware Source	Type	Notes
ISA Bus	PCI Bus					
IRQ0	- - -	1	Interval Timer 1, Counter 0	PIB (Internal)	Edge, High	
IRQ1	- - -	2	Keyboard	SIO	Edge, High	
IRQ2	- - -	3-10	Cascade from slaves (IRQ8-IRQ15)	PIB (Internal)	Edge, High	
IRQ3	- - -	11	COM2, ISA Slots	SIO, Riser	Edge, High	
IRQ4	- - -	12	COM1, ISA Slots	SIO, Riser	Edge, High	
IRQ5	- - -	13	ISA Slots	Riser	Edge, High	
IRQ6	- - -	14	Floppy Disk, ISA Slots	SIO, Riser	Edge, High	
IRQ7	- - -	15	Parallel Port, ISA Slots	SIO, Riser	Edge, High	
IRQ8	- - -	3	Real-Time Clock, Abort Switch	RTC, Abort	Edge, Low	2
IRQ9	PCIINT0	4	Ethernet, PCI Slot 1 (A*) PCI Slot 2 (D*) PCI Slot 3 (C*) ISA Slots	DEC, Riser	Level, Low	2, 3
	PCIINT1		PCI Slot 1 (B*) PCI Slot 2 (A*) PCI Slot 3 (D*) ISA Slots	Riser	Level, Low	3
IRQ10	- - -	5	Audio, ISA Slots	Crystal, Riser	Edge, High	2
IRQ11	PCIINT2	6	SCSI, PCI Slot 1 (C*, D*) PCI Slot 2 (B*, C*) PCI Slot 3 (A*, B*) ISA Slots	SYM, Riser	Level, Low	2, 3
	PCIINT3		Graphics, ISA Slots	Cirrus, Riser	Level, Low	2, 3
IRQ12	- - -	7	Mouse, ISA Slots	SIO, Riser	Edge, High	
IRQ13	- - -	8	Unused	- - -	Edge, High	1
IRQ14	- - -	9	IDE, ISA Slots	SIO, Riser	Edge, High	
IRQ15	- - -	10	ISA Slots	Riser	Edge, High	

Notes 1. Unused by the motherboard. Available for other uses.

2. Optional internal equipment.

3. If PCI devices are installed, this interrupt level may not be used or shared with ISA interrupts (Edge/Level sensitivity not compatible).

Note PCI interrupts may be shared with other PCI interrupts, but PCI and ISA interrupts may not be mixed. Note that the assignment of PCI interrupts to ISA interrupts is programmed in the PIB, and could be altered by firmware. More ISA interrupts could be made available by placing multiple PCI interrupts onto the same ISA interrupt.

Direct Memory Access (DMA) Channels

The following DMA channels are used by the indicated ISA peripherals. These channels may be shared with devices inserted in the ISA slots.

Table B-42. DMA Channels

Priority (Note)	DMA Channel	Size	Peripheral
Highest	0	8 bit	-
	1	8 bit	-
	2	8 bit	Floppy Disk
	3	8 bit	Parallel Port
	4	16 bit	(not available)
	5	16 bit	IDE Interface
	6	16 bit	Audio Playback
Lowest	7	16 bit	Audio Capture

Note Default configuration in the PIB; changeable by software.

Glossary

Abbreviations, Acronyms, and Terms to Know

This glossary defines some of the abbreviations, acronyms, and key terms used in this document.

10base-5 See thick Ethernet.

10base-2 See thin Ethernet.

10base-T See twisted-pair Ethernet.

ACIA **A**synchronous **C**ommunications **I**nterface **A**dapter

AIX **A**dvanced **I**nteractive e**X**ecutive (IBM version of UNIX)

architecture The main overall design in which each individual hardware component of the computer system is interrelated. The most common uses of this term are 8-bit, 16-bit, or 32-bit architectural design systems.

ASCII **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange. This is a 7-bit code used to encode alphanumeric information. In the IBM-compatible world, this is expanded to 8-bits to encode a total of 256 alphanumeric and control characters.

ASIC **A**pplication-**S**pecific **I**ntegrated **C**ircuit

AUI **A**ttachment **U**nit **I**nterface

BBRAM **B**attery **B**acked-up **R**andom **A**ccess **M**emory

bi-endian Having big-endian and little-endian byte ordering capability.

big-endian A byte-ordering method in memory where the address *n* of a word corresponds to the most significant byte. In an addressed memory word, the bytes are ordered (left to right) 0, 1, 2, 3, with 0 being the most significant byte.

BIOS **B**asic **I**nput/**O**utput **S**ystem. This is the built-in program that controls the basic functions of communications between the processor and the I/O (peripherals) devices. Also referred to as ROM BIOS.

BitBLT **Bit** Boundary **BL**ock **T**ransfer. A type of graphics drawing routine that moves a rectangle of data from one area of display memory to another. The data specifically need not have any particular alignment.

BLT **BL**ock **T**ransfer

board The term more commonly used to refer to a PCB (printed circuit board). Basically, a flat board made of nonconducting material, such as plastic or fiberglass, on which chips and other electronic components are mounted. Also referred to as a circuit board or card.

bpi **b**its **p**er **i**nch

bps **b**its **p**er **s**econd

bus The pathway used to communicate between the CPU, memory, and various input/output devices, including floppy and hard disk drives. Available in various widths (8-, 16-, and 32-bit), with accompanying increases in speed.

cache A high-speed memory that resides logically between a central processing unit (CPU) and the main memory. This temporary memory holds the data and/or instructions that the CPU is most likely to use over and over again and avoids accessing the slower hard or floppy disk drive.

CAS **C**olumn **A**ddress **S**trobe. The clock signal used in dynamic RAMs to control the input of column addresses.

CD **C**ompact **D**isc. A hard, round, flat portable storage unit that stores information digitally.

CD-ROM **C**ompact **D**isk **R**ead-**O**nly **M**emory

CFM **C**ubic **F**eet per **M**inute

CISC **C**omplex-**I**nstruction-**S**et **C**omputer. A computer whose processor is designed to sequentially run variable-length instructions, many of which require several clock cycles, that perform complex tasks and thereby simplify programming.

CODEC **CO**der/**DEC**oder

Color Difference (CD) The signals of (R-Y) and (B-Y) without the luminance (-Y) signal. The Green signals (G-Y) can be extracted by these two signals.

Composite Video Signal (CVS/CVBS) Signal that carries video picture information for color, brightness and synchronizing signals for both horizontal and vertical scans. Sometimes referred to as "Baseband Video".

cpi **c**haracters **p**er **i**nch

cpl **c**haracters **p**er **l**ine

CPU **C**entral **P**rocessing **U**nit. The master computer unit in a system.

DCE **D**ata **C**ircuit-terminating **E**quipment.

DLL **D**ynamic **L**ink **L**ibrary. A set of functions that are linked to the referencing program at the time it is loaded into memory.

DMA **D**irect **M**emory **A**ccess. A method by which a device may read or write to memory directly without processor intervention. DMA is typically used by block I/O devices.

DOS **D**isk **O**perating **S**ystem

dpi **d**ots **p**er **i**nch

DRAM **D**ynamic **R**andom **A**ccess **M**emory. A memory technology that is characterized by extreme high density, low power, and low cost. It must be more or less continuously refreshed to avoid loss of data.

DTE **D**ata **T**erminal **E**quipment.

ECC **E**rror **C**orrection **C**ode

ECP **E**xtended **C**apability **P**ort

EEPROM — **E**lectrically **E**rasable **P**rogrammable **R**ead-**O**nly **M**emory. A memory storage device that can be written repeatedly with no special erasure fixture. EEPROMs do not lose their contents when they are powered down.

EISA (bus) — **E**xtended **I**ndustry **S**tandard **A**rchitecture (bus) (IBM). An architectural system using a 32-bit bus that allows data to be transferred between peripherals in 32-bit chunks instead of 16-bit or 8-bit that most systems use. With the transfer of larger bits of information, the machine is able to perform much faster than the standard ISA bus system.

EPP — **E**nhanced **P**arallel **P**ort

EPROM — **E**rasable **P**rogrammable **R**ead-**O**nly **M**emory. A memory storage device that can be written once (per erasure cycle) and read many times.

ESCC — **E**nhanced **S**erial **C**ommunication **C**ontroller

ESD — **E**lectro-**S**tatic **D**ischarge/Damage

Ethernet — A local area network standard that uses radio frequency signals carried by coaxial cables.

FDC — **F**loppy **D**isk **C**ontroller

FDDI — **F**iber **D**istributed **D**ata **I**nterface. A network based on the use of optical-fiber cable to transmit data in non-return-to-zero, invert-on-1s (NRZI) format at speeds up to 100 Mbps.

FIFO — **F**irst-**I**n, **F**irst-**O**ut. A memory that can temporarily hold data so that the sending device can send data faster than the receiving device can accept it. The sending and receiving devices typically operate asynchronously.

firmware — The program or specific software instructions that have been more or less permanently burned into an electronic component, such as a ROM (read-only memory) or an EPROM (erasable programmable read-only memory).

frame — One complete television picture frame consists of 525 horizontal lines with the NTSC system. One frame consists of two Fields.

graphics controller On EGA and VGA, a section of circuitry that can provide hardware assist for graphics drawing algorithms by performing logical functions on data written to display memory.

HAL **H**ardware **A**bstraction **L**ayer. The lower level hardware interface module of the Windows NT operating system. It contains platform specific functionality.

hardware A computing system is normally spoken of as having two major components: hardware and software. Hardware is the term used to describe any of the physical embodiments of a computer system, with emphasis on the electronic circuits (the computer) and electromechanical devices (peripherals) that make up the system.

HCT **H**ardware **C**onformance **T**est. A test used to ensure that both hardware and software conform to the Windows NT interface.

I/O **I**nput/**O**utput

IBC PCI/**I**SA **B**ridge **C**ontroller

IDE **I**ntelligent **D**evice **E**xpansion

IEEE **I**nstitute of **E**lectrical and **E**lectronics **E**ngineers

interlaced A graphics system in which the even scanlines are refreshed in one vertical cycle (field), and the odd scanlines are refreshed in another vertical cycle. The advantage is that the video bandwidth is roughly half that required for a non-interlaced system of the same resolution. This results in less costly hardware. It also may make it possible to display a resolution that would otherwise be impossible on given hardware. The disadvantage of an interlaced system is flicker, especially when displaying objects that are only a few scanlines high.

IQ Signals Similar to the color difference signals (R-Y), (B-Y) but using different vector axis for encoding or decoding. Used by some USA TV and IC manufacturers for color decoding.

ISA (bus)	**I**ndustry **S**tandard **A**rchitecture (bus). The de facto standard system bus for IBM-compatible computers until the introduction of VESA and PCI. Used in the reference platform specification. (IBM)
ISASIO	**ISA S**uper **I**nput/**O**utput device
ISDN	**I**ntegrated **S**ervices **D**igital **N**etwork. A standard for digitally transmitting video, audio, and electronic data over public phone networks.
LAN	**L**ocal **A**rea **N**etwork
LED	**L**ight-**E**mitting **D**iode
LFM	**L**inear **F**eet per **M**inute
little-endian	A byte-ordering method in memory where the address *n* of a word corresponds to the least significant byte. In an addressed memory word, the bytes are ordered (left to right) 3, 2, 1, 0, with 3 being the most significant byte.
MBLT	**M**ultiplexed **BL**ock **T**ransfer
MCA (bus)	**M**icro **C**hannel **A**rchitecture
MCG	**M**otorola **C**omputer **G**roup
MFM	**M**odified **F**requency **M**odulation
MIDI	**M**usical **I**nstrument **D**igital **I**nterface. The standard format for recording, storing, and playing digital music.
MPC	**M**ultimedia **P**ersonal **C**omputer
MPC105	The PowerPC-to-PCI bus bridge chip developed by Motorola for the Ultra Plus/Ultra 60*x* system board. It provides the necessary interface between the MPC603/MPC603e/MPC604 processor and the Boot ROM (secondary cache), the DRAM (system memory array), and the PCI bus.
MPC601	Motorola's component designation for the PowerPC 601 microprocessor.
MPC603	Motorola's component designation for the PowerPC 603 microprocessor.
MPC603e	Motorola's component designation for the PowerPC 603e microprocessor.

MPC604 Motorola's component designation for the PowerPC 604 microprocessor.

MPU **M**icro**P**rocessing **U**nit

MTBF **M**ean **T**ime **B**etween **F**ailures. A statistical term relating to reliability as expressed in power on hours (poh). It was originally developed for the military and can be calculated several different ways, yielding substantially different results. The specification is based on a large number of samplings in one place, running continuously, and the rate at which failure occurs. MTBF is not representative of how long a device, or any individual device is likely to last, nor is it a warranty, but rather, of the relative reliability of a family of products.

multisession The ability to record additional information, such as digitized photographs, on a CD-ROM after a prior recording session has ended.

non-interlaced A video system in which every pixel is refreshed during every vertical scan. A non-interlaced system is normally more expensive than an interlaced system of the same resolution, and is usually said to have a more pleasing appearance.

nonvolatile memory A memory in which the data content is maintained whether the power supply is connected or not.

NTSC **N**ational **T**elevision **S**tandards **C**ommittee (USA)

NVRAM **N**on-**V**olatile **R**andom **A**ccess **M**emory

OEM **O**riginal **E**quipment **M**anufacturer

OMPAC **O**ver - **M**olded **P**ad **A**rray **C**arrier

OS **O**perating **S**ystem. The software that manages the computer resources, accesses files, and dispatches programs.

OTP **O**ne-**T**ime **P**rogrammable

palette The range of colors available on the screen, not necessarily simultaneously. For VGA, this is either 16 or 256 simultaneous colors out of 262,144.

parallel port A connector that can exchange data with an I/O device eight bits at a time. This port is more commonly used for the connection of a printer to a system.

PCI (local bus) **P**eripheral **C**omponent **I**nterconnect (local bus) (Intel). A high-performance, 32-bit internal interconnect bus used for data transfer to peripheral controller components, such as those for audio, video, and graphics.

PCMCIA (bus) **P**ersonal **C**omputer **M**emory **C**ard **I**nternational **A**ssociation (bus). A standard external interconnect bus which allows peripherals adhering to the standard to be plugged in and used without further system modification.

PDS **P**rocessor **D**irect **S**lot

physical address A binary address that refers to the actual location of information stored in secondary storage.

PIB **P**CI-to-**I**SA **B**ridge

pixel An acronym for picture element, and is also called a pel. A pixel is the smallest addressable graphic on a display screen. In RGB systems, the color of a pixel is defined by some Red intensity, some Green intensity, and some Blue intensity.

PLL **P**hase-**L**ocked **L**oop

PMC **P**CI **M**ezzanine **C**ard

POWER **P**erformance **O**ptimized **W**ith **E**nhanced **R**ISC architecture (IBM)

PowerPC™ The trademark used to describe the **P**erformance **O**ptimized **W**ith **E**nhanced **R**ISC microprocessor architecture for **P**ersonal **C**omputers developed by the IBM Corporation. PowerPC is superscalar, which means it can handle more than one instruction per clock cycle. Instructions can be sent simultaneously to three types of independent execution units (branch units, fixed-point units, and floating-point units), where they can execute concurrently, but finish out of order. PowerPC is used by Motorola, Inc. under license from IBM.

PowerPC 601™ The first implementation of the PowerPC family of microprocessors. This CPU incorporates a memory

management unit with a 256-entry buffer and a 32KB unified (instruction and data) cache. It provides a 64-bit data bus and a separate 32-bit address bus. PowerPC 601 is used by Motorola, Inc. under license from IBM.

PowerPC 603™ The second implementation of the PowerPC family of microprocessors. This CPU incorporates a memory management unit with a 64-entry buffer and 8KB instruction and data caches. It provides a selectable 32-bit or 64-bit data bus and a separate 32-bit address bus. PowerPC 603 is used by Motorola, Inc. under license from IBM.

PowerPC 603e™ A high-performance extension of the PowerPC 603. Contains 16KB instruction and data caches. PowerPC 603e is used by Motorola, Inc. under license from IBM.

PowerPC 604™ The third implementation of the PowerPC family of microprocessors. This CPU incorporates a memory management unit with a 128-entry buffer and 16KB instruction and data caches. It provides a 64-bit data bus and a separate 32-bit address bus. PowerPC 604 is used by Motorola, Inc. under license from IBM.

PowerPC Reference Platform (PRP)
A specification published by the IBM Power Personal Systems Division which defines the devices, interfaces, and data formats that make up a PRP-compliant system using a PowerPC processor.

PowerStack™ RISC PC (System Board)
A PowerPC-based computer board platform developed by the Motorola Computer Group. It supports Microsoft's Windows NT and IBM's AIX operating systems.

PRP See PowerPC Reference Platform (PRP).

PRP-compliant See PowerPC Reference Platform (PRP).

PRP Spec See PowerPC Reference Platform (PRP).

PROM **P**rogrammable **R**ead-**O**nly **M**emory

PS/2 **P**ersonal **S**ystem/**2** (IBM)

QFP **Q**uad **F**lat **P**ackage

RAM **R**andom-**A**ccess **M**emory. The temporary memory that a computer uses to hold the instructions and data currently being worked with. All data in RAM is lost when the computer is turned off.

RAS **R**ow **A**ddress **S**trobe. A clock signal used in dynamic RAMs to control the input of the row addresses.

Reduced-Instruction-Set Computer (RISC) A computer in which the processor's instruction set is limited to constant-length instructions that can usually be executed in a single clock cycle.

RFI **R**adio **F**requency **I**nterference

RGB The three separate color signals: **R**ed, **G**reen, and **B**lue. Used with color displays, an interface that uses these three color signals as opposed to an interface used with a monochrome display that requires only a single signal. Both digital and analog RGB interfaces exist.

RISC See Reduced Instruction Set Computer (RISC).

ROM **R**ead-**O**nly **M**emory

RTC **R**eal-**T**ime **C**lock

SBC **S**ingle **B**oard **C**omputer

SCSI **S**mall **C**omputer **S**ystems **I**nterface. An industry-standard high-speed interface primarily used for secondary storage. SCSI-1 provides up to 5 Mbps data transfer.

SCSI-2 (Fast/Wide) An improvement over plain SCSI; and includes command queuing. Fast SCSI provides 10 Mbps data transfer on an 8-bit bus. Wide SCSI provides up to 40 Mbps data transfer on a 16- or 32-bit bus.

serial port A connector that can exchange data with an I/O device one bit at a time. It may operate synchronously or asynchronously, and may include start bits, stop bits, and/or parity.

SIM **S**erial **I**nterface **M**odule

SIMM — **S**ingle **I**n-line **M**emory **M**odule. A small circuit board with RAM chips (normally surface mounted) on it designed to fit into a standard slot.

SIO — **S**uper **I/O** controller

SMP — **S**ymmetric **M**ulti**P**rocessing. A computer architecture in which tasks are distributed among two or more local processors.

SMT — **S**urface **M**ount **T**echnology. A method of mounting devices (such as integrated circuits, resistors, capacitors, and others) on a printed circuit board, characterized by not requiring mounting holes. Rather, the devices are soldered to pads on the printed circuit board. Surface-mount devices are typically smaller than the equivalent through-hole devices.

software — A computing system is normally spoken of as having two major components: hardware and software. Software is the term used to describe any single program or group of programs, languages, operating procedures, and documentation of a computer system. Software is the real interface between the user and the computer.

SRAM — **S**tatic **R**andom **A**ccess **M**emory

SSBLT — **S**ource **S**ynchronous **BL**ock **T**ransfer

standard(s) — A set of detailed technical guidelines used as a means of establishing uniformity in an area of hardware or software development.

SVGA — **S**uper **V**ideo **G**raphics **A**rray (IBM). An improved VGA monitor standard that provides at least 256 simultaneous colors and a screen resolution of 800 x 600 pixels.

Teletext — One way broadcast of digital information. The digital information is injected in the broadcast TV signal, VBI, or full field, The transmission medium could be satellite, microwave, cable, etc. The display medium is a regular TV receiver.

thick Ethernet (10base-5)
An Ethernet in which the physical medium is a doubly shielded, 50-ohm coaxial cable capable of carrying data at 10 Mbps for a length of 500 meters (also referred to as thicknet).

thin Ethernet (10base-2)
An Ethernet in which the physical medium is a single-shielded, 50-ohm RG58A/U coaxial cable capable of carrying data at 10 Mbps for a length of 185 meters (also referred to as AUI or thinnet).

twisted-pair Ethernet (10base-T)
An Ethernet in which the physical medium is an unshielded pair of entwined wires capable of carrying data at 10 Mbps for a maximum distance of 185 meters.

UART **U**niversal **A**synchronous **R**eceiver/**T**ransmitter

UV **U**ltra**V**iolet

UVGA **U**ltra **V**ideo **G**raphics **A**rray. An improved VGA monitor standard that provides at least 256 simultaneous colors and a screen resolution of 1024 x 768 pixels.

VAFC **V**ESA **A**dvanced **F**eature **C**onnector.

Vertical Blanking Interval (VBI)
The time it takes the beam to fly back to the top of the screen in order to retrace the opposite field (odd or even). VBI is in the order of 20 TV lines. Teletext information is transmitted over 4 of these lines (lines 14-17).

VESA (bus) **V**ideo **E**lectronics **S**tandards **A**ssociation (or VL bus). An internal interconnect standard for transferring video information to a computer display system.

VGA **V**ideo **G**raphics **A**rray (IBM). The third and most common monitor standard used today. It provides up to 256 simultaneous colors and a screen resolution of 640 x 480 pixels.

virtual address A binary address issued by a CPU that indirectly refers to the location of information in primary memory, such as main memory. When data is copied from disk to main memory, the physical address is changed to the virtual address.

VL bus See **V**ESA **L**ocal bus (VL bus).

VMEchip2 MCG second generation VMEbus interface ASIC (Motorola)

VME2PCI MCG ASIC that interfaces between the PCI bus and the VMEchip2 device.

volatile memory A memory in which the data content is lost when the power supply is disconnected.

VRAM **V**ideo (Dynamic) **R**andom **A**ccess **M**emory. Memory chips with two ports, one used for random accesses and the other capable of serial accesses. Once the serial port has been initialized (with a transfer cycle), it can operate independently of the random port. This frees the random port for CPU accesses. The result of adding the serial port is a significantly reduced amount of interference from screen refresh. VRAMs cost more per bit than DRAMs.

Windows NT™ The trademark representing **Windows N**ew **T**echnology, a computer operating system developed by the Microsoft Corporation.

XGA E**X**tended **G**raphics **A**rray. An improved IBM VGA monitor standard that provides at least 256 simultaneous colors and a screen resolution of 1024 x 768 pixels.

Y Signal Luminance. This determines the brightness of each spot (pixel) on a CRT screen either color or B/W systems, but not the color.

Index

C

D

E

F

G

H

I

J

K

P

R

S

T

INDEX