VMICPCI-7697

Single Board Pentium II® Embedded Module

Processor-Based Peripheral Slot CompactPCI® CPU

Product Manual

12090 South Memorial Parkway
Huntsville, Alabama 35803-3308, USA
(256) 880-0444 ◆ (800) 322-3616 ◆ Fax: (256) 882-0859

500-657697-000 Rev. B

FCC

This card has been tested and met FCC Rules, Part 15, Class B.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

(1) This device may not cause harmful interference, and
(2) This device must accept any interference received, including interference that may cause undesired operation.

CE

This card has been tested and met the following standards:
EN60950
EN55022
EN61000-3-2
EN61000-4-11
EN61000-4-6
EN61000-4-4
EN61000-4-3
EN61000-4-2
VMIC declares this card meets CE Conformity.

UL1950

This card has been tested and met UL1950: 1992 (including Amendments A1:1993, A2:1993, A3:1995, and A4:1997).

Special Considerations:

1) A suitable fire and electrical enclosure shall be provided.
2) The input to the unit is considered to be an isolated SELV source.
3) A maximum operating base plate of 105 degrees C must be observed .

12090 South Memorial Parkway
Huntsville, Alabama 35803-3308, USA
(256) 880-0444 ◆ (800) 322-3616 ◆ Fax: (256) 882-0859

COPYRIGHT AND TRADEMARKS

© Copyright 1999. The information in this document has been carefully checked and is believed to be entirely reliable. While all reasonable efforts to ensure accuracy have been taken in the preparation of this manual, VMIC assumes no responsibility resulting from omissions or errors in this manual, or from the use of information contained herein.

VMIC reserves the right to make any changes, without notice, to this or any of VMIC's products to improve reliability, performance, function, or design.

VMIC does not assume any liability arising out of the application or use of any product or circuit described herein; nor does VMIC convey any license under its patent rights or the rights of others.

For warranty and repair policies, refer to VMIC's Standard Conditions of Sale.

AMXbus, BITMODULE, COSMODULE, DMAbus, Instant OPC wizard logo, IOWorks Access, IOWorks Foundation, IOWorks man figure, IOWorks Manager, IOWorks Server, MAGICWARE, MEGAMODULE, PLC ACCELERATOR (ACCELERATION), Quick Link, RTnet, Soft Logic Link, SRTbus, TESTCAL, "The Next Generation PLC", The PLC Connection, TURBOMODULE, UCLIO, UIOD, UPLC, Visual Soft Logic Control(ler), *VMEaccess*, *VMEmanager*, *VMEmonitor*, VMEnet, VMEnet II, and *VMEprobe* are trademarks. The I/O Experts, The I/O Systems Experts, The Soft Logic Experts, and The Total Solutions Provider are service marks of VMIC.

(I/O man figure)

(Instant OPC wizard logo)

(IOWorks man figure)

The I/O man figure, IOWorks, UIOC, Visual IOWorks, and *WinUIOC* are registered trademarks of VMIC.

ActiveX is a trademark. Microsoft, Microsoft Access, MS-DOS, Visual Basic, Visual C++, Win32, Windows, Windows NT, and XENIX are registered trademarks of Microsoft Corporation.

MMX is a trademark and Celeron is a registered trademark of Intel Corporation.

PICMG and CompactPCI are registered trademarks of PCI Industrial Computer Manufacturers' Group.

Other registered trademarks are the property of their respective owners.

VMIC
All Rights Reserved
This document shall not be duplicated, nor its contents used for any
purpose, unless granted express written permission from VMIC.

12090 South Memorial Parkway
Huntsville, Alabama 35803-3308, USA
(256) 880-0444 ◆ (800) 322-3616 ◆ Fax: (256) 882-0859

Table of Contents

List of Figures

List of Tables

Overview

Introduction

VMIC's VMICPCI-7697 is a complete IBM PC / AT-compatible Pentium II Embedded Module processor-based computer with the additional benefits of Eurocard construction and full compatibility with the CompactPCI Specification Rev. 2.1. The VMICPCI-7697 with advanced CompactPCI interface and SDRAM that is dual-ported to the CompactPCI bus, is ideal for CompactPCI peripheral slot applications.

The single-slot CPU board functions as a standard PC / AT, executing a PC / AT-type power-on self-test, then boots up MS-DOS, Windows 3.11, Windows 95, or Windows NT. The PC / AT mode of the VMICPCI-7697 is discussed in Chapter 3 of this manual.

The VMICPCI-7697 operates as a CompactPCI peripheral slot CPU and interacts with other CompactPCI modules via the on-board embedded bridge. The VMICPCI-7697 is also compatible with the CompactPCI Hot Swap specification PICMG 2.1 R1.0 and supports both Full Hot Swap and High Availability Hot Swap.

The VMICPCI-7697 programmer may quickly and easily control CompactPCI bus functions simply by linking to a library of interrupt and control functions. This library is available with VMIC's VMISFT-9421 IOWorks Access software for Windows NT users.

The VMICPCI-7697 also provides capabilities beyond the features of a typical PC / AT compatible CPU including general-purpose timers, a programmable watchdog timer, a bootable flash disk system, remote LANboot, and nonvolatile, battery-backed SRAM. These features make the unit ideal for embedded applications. These nonstandard PC / AT functions are discussed in Chapter 4 of this manual.

Organization of the Manual

This manual is composed of the following chapters and appendices:

Chapter 1 - VMICPCI-7697 Features and Options describes the features of the base unit followed by descriptions of the associated features of the unit in operation on a CompactPCI bus.

Chapter 2 - Installation and Setup describes unpacking, inspection, hardware jumper settings, connector definitions, installation, system setup, and operation of the VMICPCI-7697.

Chapter 3 - PC/AT Functions describes the unit design in terms of the standard PC memory and I/O maps, along with the standard interrupt architecture.

Chapter 4 - Embedded PC/RTOS Features describes the unit features that are beyond standard PC/AT functions.

Chapter 5 - Maintenance provides information relative to the care and maintenance of the unit.

Appendix A - Connector Pinouts illustrates and defines the connectors included in the unit's I/O ports.

Appendix B - System Driver Software provides details for installing drivers under Windows 3.11, Windows 95 and, Windows NT.

Appendix C - LANWorks BIOS describes the menus and options associated with the LANWorks BIOS.

Appendix D - Award BIOS describes the menus and options associated with the Award (system) BIOS.

Appendix E - Device Configuration: I/O and Interrupt Control provides the user with the information needed to develop custom applications such as the revision of the current BIOS configuration to a user-specific configuration.

Appendix F - Sample 'C' software provides example code to use with the VMICPCI-7697.

References

For the most up-to-date specifications for the VMICPCI-7697, please refer to:

VMIC specification number 800-657697-000

Pentium II® Processor with MMX™ Technology
January 1997, Order Number 243185-001
Intel Corporation
2200 Mission College Blvd.
P.O. Box 58119
Santa Clara, CA 95052-8119
(408) 765-8080

Intel 82440BX AGP set: 82443BX Host Bridge/Controller
Intel Corporation
2200 Mission College Boulevard
P.O. Box 58119
Santa Clara, CA 95052-8119

Intel 21554 Embedded PCI Bridge
Intel Corporation
2200 Mission College Boulevard
P.O. Box 58119
Santa Clara, CA 95052-8119

Intel 21143 10/100 Mb/s Ethernet LAN Controller
Intel Corporation
2200 Mission College Boulevard
P.O. Box 58119
Santa Clara, CA 95052-8119

Intel 82440BX PCIset ISA Bridge
82371EB PCI ISA IDE Xcellerator (PIIX4E)
2200 Mission College Boulevard
P.O. Box 58119
Santa Clara, CA 95052-8119

PCI Local Bus Specification, Rev. 2.1
PCI Special Interest Group
P.O. Box 14070
Portland, OR 97214
(800) 433-5177 (U.S.)
(503) 797-4207 (International)
(503) 234-6762 (FAX)

SMC FDC37C67X Enhanced Super I/O Controller
SMC Component Products Division
300 Kennedy Drive
Hauppauge, NY 11788
(516) 435-6000
(516) 231-6004 (FAX)

ISA & EISA, Theory and Operation

Solari, Edward
Annabooks
15010 Avenue of Science, Suite 101
San Diego, CA 92128 USA
ISBN 0-929392 -15-9

82C54 CHMOS Programmable Internal Timer

Intel Corporation
2200 Mission College Boulevard
P.O. Box 58119
Santa Clara, CA 95052-8119

Flash ChipSet Product Manual

SanDisk Corporation
140 Caspian Court
Sunnyvale, CA 94089-9820

DS 1384 Watchdog Timekeeping Controller

Dallas Semiconductor
4461 South Beltwood Pwky.
Dallas, TX 75244-3292

M-Systems Corporate Headquarters

USA Office
39899 Balentine Dr.
Suite 335
Newark, CA 94560
Tel: 510-413-5950
Fax: 510-413-5980
Email: info@m-sys.com

S3 Trio 3d AGP Video Controller

P.O. Box 58058
Santa Clara, CA 95052-8058
(408) 588-8000

CMC Specification, P1386/Draft 2.0 from:

IEEE Standards Department
Copyrights and Permissions
445 Hoes Lanes, P.O. Box 1331
Piscataway, NJ 08855-1331, USA

PMC Specification, P1386.1/Draft 2.0 from:

IEEE Standards Department
Copyrights and Permissions
445 Hoes Lanes, P.O. Box 1331
Piscataway, NJ 08855-1331, USA

For a detailed description and specification of the CompactPCI bus, please refer to:

CompactPCI Specification PICMG 2.0 R2.1
PCI Industrial Computer Manufactures' Group
301 Edgewater Place
Suite 220
Wakefield, MA 01880
(617) 224-1100
(617) 224-1239 (FAX)
www.picmg.org (Web)

The following is useful information related to remote ethernet booting of the VMICPCI-7697:

Microsoft Windows NT Server Resource Kit
Microsoft Corporation
ISBN: 1-57231-344-7
www.microsoft.com

Safety Summary

The following general safety precautions must be observed during all phases of the operation, service, and repair of this product. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of this product.

VMIC assumes no liability for the customer's failure to comply with these requirements.

Ground the System

To minimize shock hazard, the chassis and system cabinet must be connected to an electrical ground. A three-conductor AC power cable should be used. The power cable must either be plugged into an approved three-contact electrical outlet or used with a three-contact to two-contact adapter with the grounding wire (green) firmly connected to an electrical ground (safety ground) at the power outlet.

Do Not Operate in an Explosive Atmosphere

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

Keep Away from Live Circuits

Operating personnel must not remove product covers. Component replacement and internal adjustments must be made by qualified maintenance personnel. 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.

Do Not Substitute Parts or Modify System

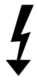
Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification to the product. Return the product to VMIC for service and repair to ensure that safety features are maintained.

Dangerous Procedure Warnings

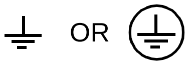
Warnings, such as the example below, precede only potentially dangerous procedures throughout this manual. Instructions contained in the warnings must be followed.

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

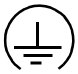
Safety Symbols Used in This Manual



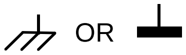
Indicates dangerous voltage (terminals fed from the interior by voltage exceeding 1000 V are so marked).



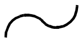
Protective conductor terminal. For protection against electrical shock in case of a fault. Used with field wiring terminals to indicate the terminal which must be connected to ground before operating equipment.



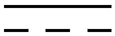
Low-noise or noiseless, clean ground (earth) terminal. Used for a signal common, as well as providing protection against electrical shock in case of a fault. Before operating the equipment, terminal marked with this symbol must be connected to ground in the manner described in the installation (operation) manual.



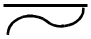
Frame or chassis terminal. A connection to the frame (chassis) of the equipment which normally includes all exposed metal structures.



Alternating current (power line).



Direct current (power line).



Alternating or direct current (power line).

STOP The STOP symbol informs the operator the that a practice or procedure should not be performed. Actions could result in injury or death to personnel, or could result in damage to or destruction of part or all of the system.

WARNING The WARNING sign denotes a hazard. It calls attention to a procedure, a practice, a condition, which, if not correctly performed or adhered to, could result in injury or death to personnel.

CAUTION The CAUTION sign denotes a hazard. It calls attention to an operating procedure, a practice, or a condition, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the system.

NOTE The NOTE sign denotes important information. It calls attention to a procedure, a practice, a condition or the like, which is essential to highlight.

CHAPTER

1

Features and Options

Contents

Introduction

The VMICPCI-7697 performs all the functions of a standard IBM PC / AT motherboard with the following features:

- Single-slot CompactPCI bus 6U Eurocard form factor
- Includes a high-performance Intel Pentium II® Embedded Module processor
- Up to 512 Mbyte of Synchronous DRAM
- AGP video with 4 Mbyte SDRAM
- Real-time clock / calendar
- Front panel reset switch and miniature speaker
- Onboard port for keyboard and mouse
- UltraEIDE hard drive, floppy drive through the CompactPCI J3 connector
- Onboard fast Ethernet controller supporting 10BaseT and 100BaseTX interfaces
- Front panel ‘vital sign’ indicators (power, UltraEIDE hard drive activity, Ready, and Ethernet status)
- Three general-purpose programmable 16-bit timers
- Software-selectable watchdog timer with reset
- Optional Msystems DiskOnChip flash memory (12 to 144 Mbyte available)
- 16 Mbyte of bootable flash on secondary IDE
- 32 Kbyte of battery-backed SRAM
- Two Serial ports, one available by way of the front panel RJ11 connector and both available through the CompactPCI J3 connector
- One enhanced parallel port available on the front panel and through the CompactPCI J3 connector

- Two USB Ports: one available on the front panel, and one through the CompactPCI J3 connector (rear I/O)

The VMICPCI-7697 supports standard PC/AT I/O features such as those listed in Table 1-1. Figure 1-1 on page 23 shows a block diagram of the VMICPCI-7697 emphasizing the I/O features, including the embedded PCI bridge. The serial, parallel, IDE, and floppy signals are also routed through the backplane to VMIC's (optional) VMIACC-0576 CompactPCI Rear Transition Utility Board.

Table 1-1 PC/AT I/O Features

I/O FEATURE	IDENTIFIER	PHYSICAL ACCESS
Two high-performance 16550-compatible serial ports	COM1, COM2	One through the front panel RJ11, both through CompactPCI J3 Connector (See Note)
One Enhanced Parallel Port, Supports ECP/EPP Modes	Parallel	Through front panel and CompactPCI J3 connector (See Note)
AT-Style Keyboard/Mouse Controller	M/K	Front Panel PS/2-Style Connector, Mini-DIN Circular (female)
AGP Video Controller with 4 Mbyte SDRAM	SVGA	Front Panel DB15HD High Density (female)
Real-Time Clock/Calendar with miniature speaker	Date, Time, and Sound	
On-board Fast Ethernet controller supporting 10BaseT and 100BaseTX interfaces	LAN	Front Panel R-J45
Floppy Disk Controller	Drive A	Through CompactPCI J3 (See Note)
Ultra EIDE Hard Drive Controller	Drive C	Through CompactPCI J3 (See Note)
Two USB Ports	USB	One through the front panel, the other through CompactPCI J3
Hardware Reset	RST	Front Panel Push-Button
Power Status, Hard Drive Activity, Ready, and Ethernet Status for each controller	LED Indicators	Front Panel

NOTE: The VMIACC-0576 CompactPCI Rear Transition Utility Board provides USB, RJ-45 serial port connectors, standard (DB25) parallel port connector, IDE, and Floppy connectors, by way of the CompactPCI backplane connector (J3).

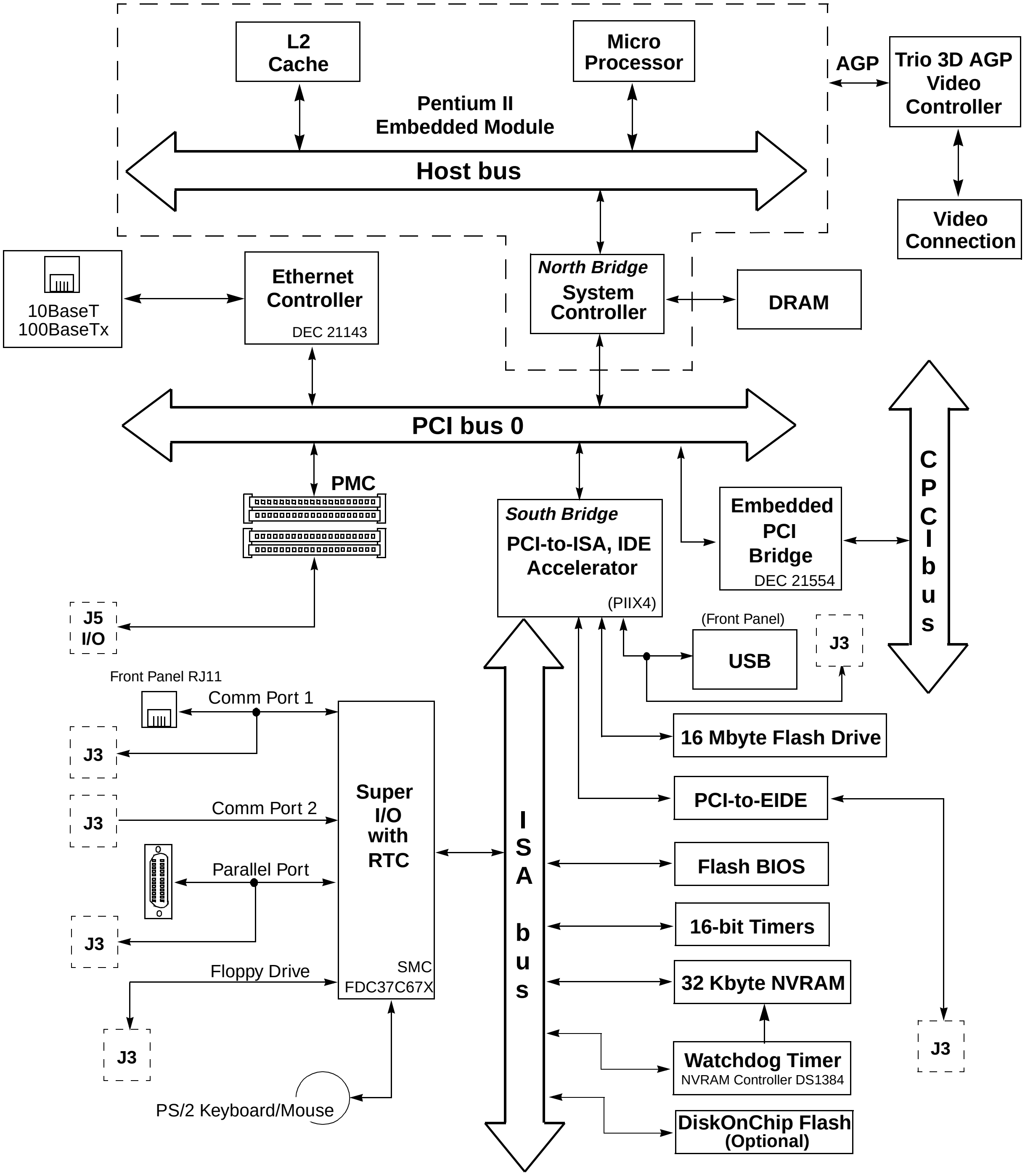


Figure 1-1 VMICPCI-7697 Block Diagram

CompactPCI Bus Bridge

The VMICPCI-7697 incorporates a PCI peripheral device that performs PCI bridging functions for embedded and intelligent I/O applications. The PCI device acts as a gateway to an intelligent subsystem. It allows the local VMICPCI-7697 Pentium II processor to configure and control the on-board local subsystem independent from the CompactPCI bus host processor. The VMICPCI-7697 local PCI subsystem is presented to the CompactPCI bus host as a single CompactPCI device. For detailed information concerning the embedded PCI bus bridge consult the manual titled *'Intel 21554 Embedded PCI-to-PCI Bridge for the VMICPCI-7697 CompactPCI CPU'*.

The VMICPCI-7697 PCI bridge device provides the following features:

PCI Interface

- Full compliance with the PCI Local Bus Specification, Revision 2.1
- 3.3 V operation with 5.0 V tolerant I/O
- Concurrent local (secondary) and CompactPCI (primary) bus operation
- Compliant with CompactPCI Hot Swap specification PICMG 2.1 R1.0 for Full and High Availability Hot Swap

Buffer Architecture

- Queuing of multiple transactions in either direction
- 256 byte of posted write (data and address) buffering in each direction
- 256 byte of read data buffering in each direction
- Four delayed transaction entries in each direction

Configuration Register and Control/Status Registers (CSRs)

- Two sets of standard PCI configuration registers corresponding to the local and CompactPCI interfaces. Each set is accessible from either the local or CompactPCI interface.
- Four 32-bit base address configuration registers mapping the CSRs.

Transaction Forwarding

- Four primary interface base address configuration registers for downstream forwarding with size and prefetchable programming for all four address ranges
- Direct offset address translation for downstream memory and I/O transactions
- Three secondary interface address ranges for upstream forwarding with size and prefetchable programming for all three address ranges
- Inverse decoding above 4 Gbyte address boundary for upstream Digital-to-Analog Controllers (DACs)
- Ability to generate type 0 and type 1 configuration commands on the primary or secondary interface by way of configuration cycles or I/O cycles accessing the CSR

VMICPCI-7697 Hot Swap Support

The VMICPCI-7697 implements hardware to support both Full Hot Swap and High Availability Hot Swap per the CompactPCI Hot Swap Specification Rev 1.0. The hot swap circuits contain an integrated power controller, an I/O signal pre-charge circuit, and a CompactPCI hot swap controller. These hot swap functions allow the VMICPCI-7697 to be inserted into, or extracted from, a CompactPCI system while the system is still powered.

When the VMICPCI-7697 is inserted into a powered system, a physical connection will be made with the chassis in order to properly discharge the voltage potentials to ground. As the VMICPCI-7697 connector is mated with the backplane connector, the long power pins of the backplane will provide early power to the unit. As soon as power is applied; the onboard VMICPCI-7697 power controller will assert the reset outputs to the backend logic, turn off gate voltages to external power FETs, and begin outputting a reference voltage that is used to pre-charge the CompactPCI interface logic I/O pins before they mate with the bus signals. The next pins to mate are the I/O pins and the balance of the power pins. The I/O pins will have been precharged by the onboard power controller circuit. The BD_SEL# pins are the last inputs to be driven to their true state. In Basic or Full Hot Swap systems, these inputs are connected to ground when the short pins are mated. If the system supports High Availability Hot Swap, the BD_SEL# pin can be actively driven by the CompactPCI system controller. Once the board is fully inserted and power is supplied to the rest of the subsystem, the power controller will de-assert the local reset signal. The VMICPCI-7697 initialization process will begin, and once the on-board 21554 hot swap controller has been initialized by a serial ROM load, it will drive the CPCI ENUM# signal to the System Controller or Host. Host software will then clear the CPCI ENUM# status bit, configure the 21554, and load any required subsystem drivers. The board will then be fully initialized.

The card removal process operates in the opposite sequence. As the ejector handle is rotated, the ejector switch will open causing a change of state that will activate the CPCI ENUM# signal. The host will then bring the VIMIPCI-7697 unit into a known software state and turn on the blue front panel LED to signify that it is now safe to proceed with card removal. In High Availability systems, the Host can actively de-assert the BD_SEL# signal, removing power from all but the backend circuitry. For all other forms of hot swap, the action of card removal disconnects the BD_SEL# (short pins) from ground.

CAUTION: In order to Hot Swap the VMICPCI-7697 the user must have a Compact PCI backplane that supports Hot Swap. If the VMICPCI-7697 is inserted into or extracted from a powered backplane that does not support Hot Swap, damage can occur to the unit.

VMICPCI-7697 Product Options

VMIC’s VMICPCI-7697 is built around three fundamental hardware configurations. These involve processor power, RAM memory, and the optional DiskOnChip flash memory size. *These options are subject to change based on emerging technologies and availability of vendor configurations.*

The options and current details available with the VMICPCI-7697 are defined in the device specification sheet available from your VMIC representative.

CHAPTER
2

Installation and Setup

Contents

Introduction

This chapter describes the hardware jumper settings, connector descriptions, installation, system setup, and operation of the VMICPCI-7697.

Unpacking Procedures

Any precautions found in the shipping container should be observed. All items should be carefully unpacked and thoroughly inspected for damage that might have occurred during shipment. The board(s) should be checked for broken components, damaged printed circuit board(s), heat damage, and other visible contamination. All claims arising from shipping damage should be filed with the carrier and a complete report sent to VMIC Customer Service together with a request for advice concerning the disposition of the damaged item(s).

CAUTION: Some of the components assembled on VMIC's products may be sensitive to electrostatic discharge and damage may occur on boards that are subjected to a high energy electrostatic field. When the board is placed on a bench for configuring, etc., it is suggested that conductive material be inserted under the board to provide a conductive shunt. Unused boards should be stored in the same protective boxes in which they were shipped.

Hardware Setup

The VMICPCI-7697 is factory populated with user-specified options as part of the VMICPCI-7697 ordering information. The CPU speed, RAM size, and flash memory size are not user-upgradable. To change CPU speeds, RAM / flash size, contact customer service to receive a Return Material Authorization (RMA).

VMIC Customer Service is available at: 1-800-240-7782.

The VMICPCI-7697 is tested for system operation and shipped with factory-installed header jumpers. The physical location of the jumpers and connectors for the single board CPU are illustrated in Figure 2-1 on page 30. The definitions of the CPU board jumpers and connectors are included in Table 2-1 through Table 2-15.

CAUTION: All jumpers marked User Configurable in the following tables may be changed or modified by the user. All jumpers marked factory configured should not be modified by the user.

Care must be taken when making jumper modifications to ensure against improper settings or connections. Improper settings may result in damage to the unit.

Modifying any other jumper will void the Warranty and may damage the unit. The default jumper condition of the VMICPCI-7697 is expressed in Table 2-1 through Table 2-15 with **bold text** in the table cells.

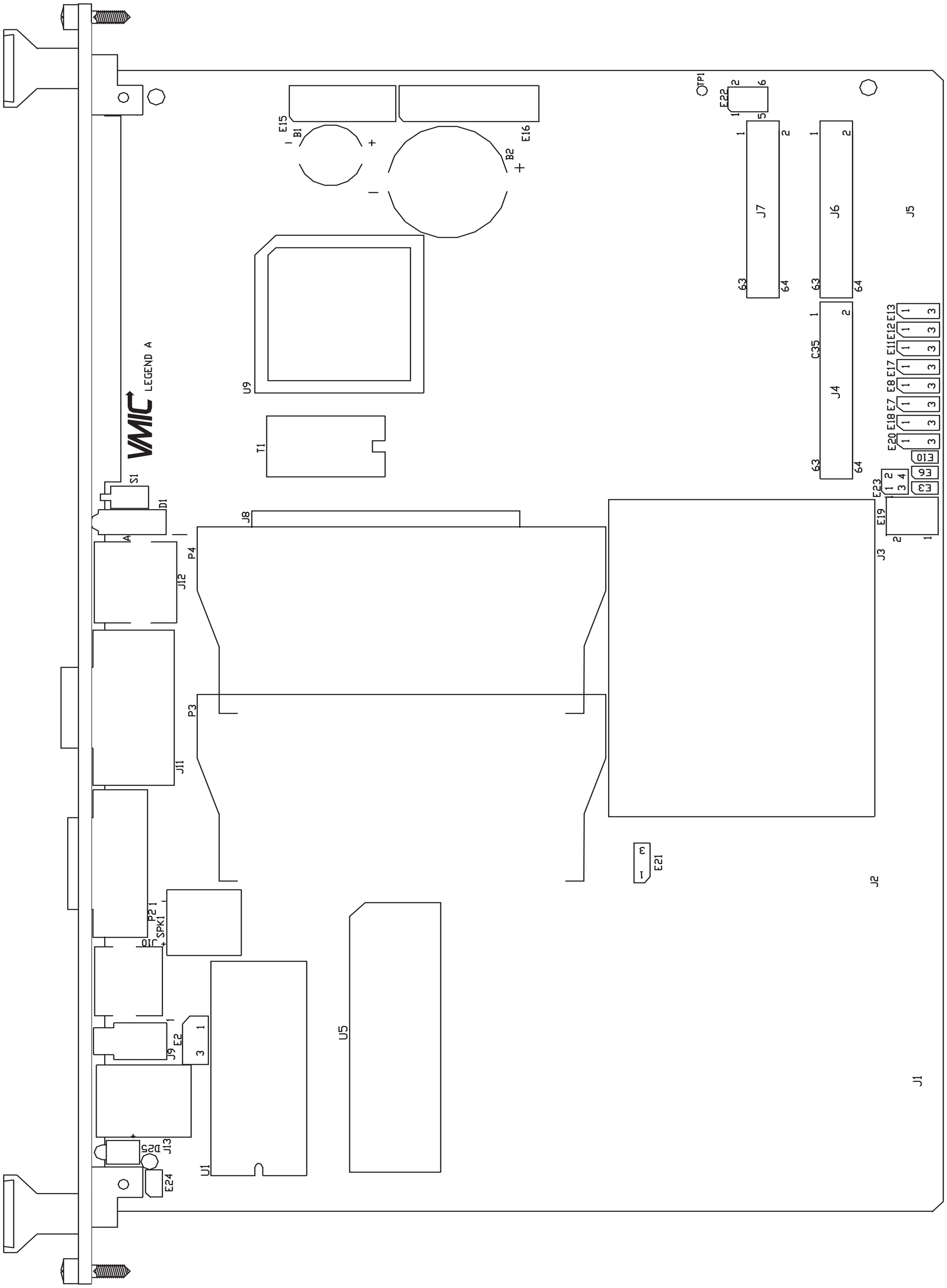


Figure 2-1 VMICPCI-7697 Embedded Module, and Jumper Locations

Table 2-1 CPU Board Connectors

Connector	Function
J10	Mouse/Keyboard Connector
J9, J3	USB Connector
J13	COM1
J3	COM1/COM2
J3, P2	Parallel Port Connector
J12	Ethernet Connector
J11	Video Connector
J3	Floppy Interface Connector
E19	Factory Reserved Do Not Use
E15	ITP Test Header
E16	Port 80 Test Header
J3	EIDE Connector
E2	Fan Connector
J6, J7, J4	PMC Slot
J1, J2, J5	CompactPCI Connectors

Table 2-2 Password Clear (**User Configurable**) - Jumper (E3)

	Jumper Position
Normal	**Out**
Clear NVRAM/CMOS/ Password	In

NOTE: The BIOS has the capability (not currently enabled) of password protecting casual access to the unit's CMOS set-up screens. The Password Clear jumper allows the user to clear the Password in the case of a forgotten password.

To clear the CMOS password:

1. Turn off power to the unit
2. Install a jumper at E3
3. Power up the unit
4. Turn off the power to the unit and remove the jumper from E3

When power is reapplied to the unit, the CMOS password will be cleared.

Table 2-3 Factory Configured - PIIX4E Battery Jumper (E6)

PIIX4E Battery	Jumper Position
Enabled	**In**
Disabled	Out

Table 2-4 Watchdog Timer Jumper (**User Configurable**) - (E7)

Select	Jumper Position
Watchdog Timer Reset	1-2
No Watchdog Timer Reset or SERR#	**Out**
Watchdog Timer SERR#	2-3

Table 2-5 Factory Configured - Clock Speed Select Jumper (E8)

CPU Bus Frequency (MHz)	PCI Bus Frequency (MHz)	E8 Position
66.6	**33.3**	**2-3**
100	33.3	1-2

Table 2-6 Factory Configured - SRAM Battery Jumper (E10)

SRAM Battery	Jumper Position
Enabled	**In**
Disabled	Out

Table 2-7 Programmable Timer Clock Select (**User Configurable**) - Jumper (E11)

Select	Jumper Position
2 MHz	1-2
1 MHz	**2-3**

Table 2-8 Factory Configured - Reserved Jumper (E12)

Reserved	1 - 2	Out
Reserved	**2 - 3**	**In**

Table 2-9 Factory Configured - DiskOnChip Address SEL Jumper (E13)

DiskOnChip	Jumper Position	
D000	**1 - 2**	**In**
E000	2 - 3	In
Disabled	Out	Out

Table 2-10 Factory Configured - Boot Block Lock Jumper (E17)

Select	Jumper Position
Boot Block Programming	1 - 2
No Program	**2 - 3**

Table 2-11 Factory Configured - BootBlock Lock Jumper (E18)

Select	Jumper Position
Reserved	1-2
Reserved	2 -3
Reserved	**Jumper Not Loaded**

Table 2-12 Primary Lockout (**User Configurable**) - Jumper (E20)
(Refer to Section 16.2.1 in Volume II)

Select	Jumper Position
Enabled	1 - 2
Disabled	**2 - 3**

Table 2-13 CPCI Reset Input Enable (**User Configurable**) - Jumper (E21)

Select	Jumper Position
CPCI bus Reset Causes 21554 Chip Reset	**2 - 3**
CPCI bus Reset Does Not Cause 21554 Chip Reset	1 - 2

Table 2-14 PMC I/O Connector and CPCI J5 Connections to Ground (**User Configurable**) Jumper (E22)

Select	Jumper Position
Connect J5 Pins E10, D10, All To GND	1-2 In 3-4 In 5-6 In
Do Not Connect J5 Pins E10, D10, All To GND	**1-2 Out** **3-4 Out** **5-6 Out**

Table 2-15 PMC I/O Connector and CPCI J5 Connection to +5V (**User Configurable**) (E23)

Select	Jumper Position
Connect J5 Pins B10 and C10 To +5V	1-2 In 2-3 In
Do Not Connect J5 Pins B10 and C10 To +5V	**1-2 Out** **2-3 Out**

Installation

The VMICPCI-7697 conforms to the CompactPCI physical specification for a 6U board. The VMICPCI-7697 is a peripheral slot only board. It can be plugged directly into any standard chassis accepting this type of board. The following picture illustrates the symbol use to identify the slots in a standard CompactCPI chassis.

△

This symbol identifies the System Controller slot

○

This symbol identifies peripheral slot

The following steps describe the VMIC recommended method for installation and powerup of the VMICPCI-7697:

1. If a PMC module such as VMIC's VMIPMC-7441 is to be used, connect it to the VMICPCI-7697 prior to board installation. Refer to the Product Manual for the PMC module for configuration and setup.
2. The VMICPCI-7697 must be installed in a designated peripheral slot of the CompactPCI backplane (See symbols above for selection of the correct slot).

NOTE: The VMICPCI-7697 requires forced air cooling. It is advisable to install blank panels over any exposed slots. This will ensure air flow over the VMICPCI-7697 board.

3. Insert the VMICPCI-7697 into a CompactPCI chassis peripheral slot. While ensuring that the board is properly aligned and oriented in the supporting board guides, slide the board smoothly forward against the mating connector. Use the ejector handles to firmly seat the board.
4. All needed peripherals can be accessed from the front panel and the rear I/O VMIACC-0576 Rear Transition utility board. Each connector is clearly labeled on the front panel, and detailed pinouts are in Appendix A.
5. The VMICPCI-7697 features a Flash Disk resident on the board. Refer to Chapter 4 for set up details.
6. If an external drive module is installed, the BIOS Setup program must be run to configure the drive types. See Appendix D to properly configure the system.
7. If a drive module is present, install the operating system according to the manufacturer's instructions.
8. A keyboard and a mouse are required if the user has not previously configured the system.

See Appendix B for instructions on installing VMICPCI-7697 peripheral driver software during operating system installation.

BIOS Setup

The VMICPCI-7697 has an on-board BIOS Setup program that controls many configuration options. These options are saved in a special nonvolatile, battery-backed memory chip and are collectively referred to as the board's 'CMOS Configuration'. The CMOS configuration controls many details concerning the behavior of the hardware from the moment power is applied.

The VMICPCI-7697 is shipped from the factory with hard drive type configuration set to AUTO in the CMOS.

Details of the VMICPCI-7697 BIOS setup program are included in Appendix D.

Front Panel Connectors

The front panel connections, including connector pinouts and orientation, for the VMICPCI-7697 are defined in detail in Appendix A. Rear panel connections are defined in detail in the Installation Guide for the VMIACC-0576.

PMC Site Connector

The VMICPCI-7697 supports IEEE-P1386 Common Mezzanine Card Specification with a 5 V PMC site. Figure 2-2 shows the installation of the PMC card on the VMICPCI-7697.

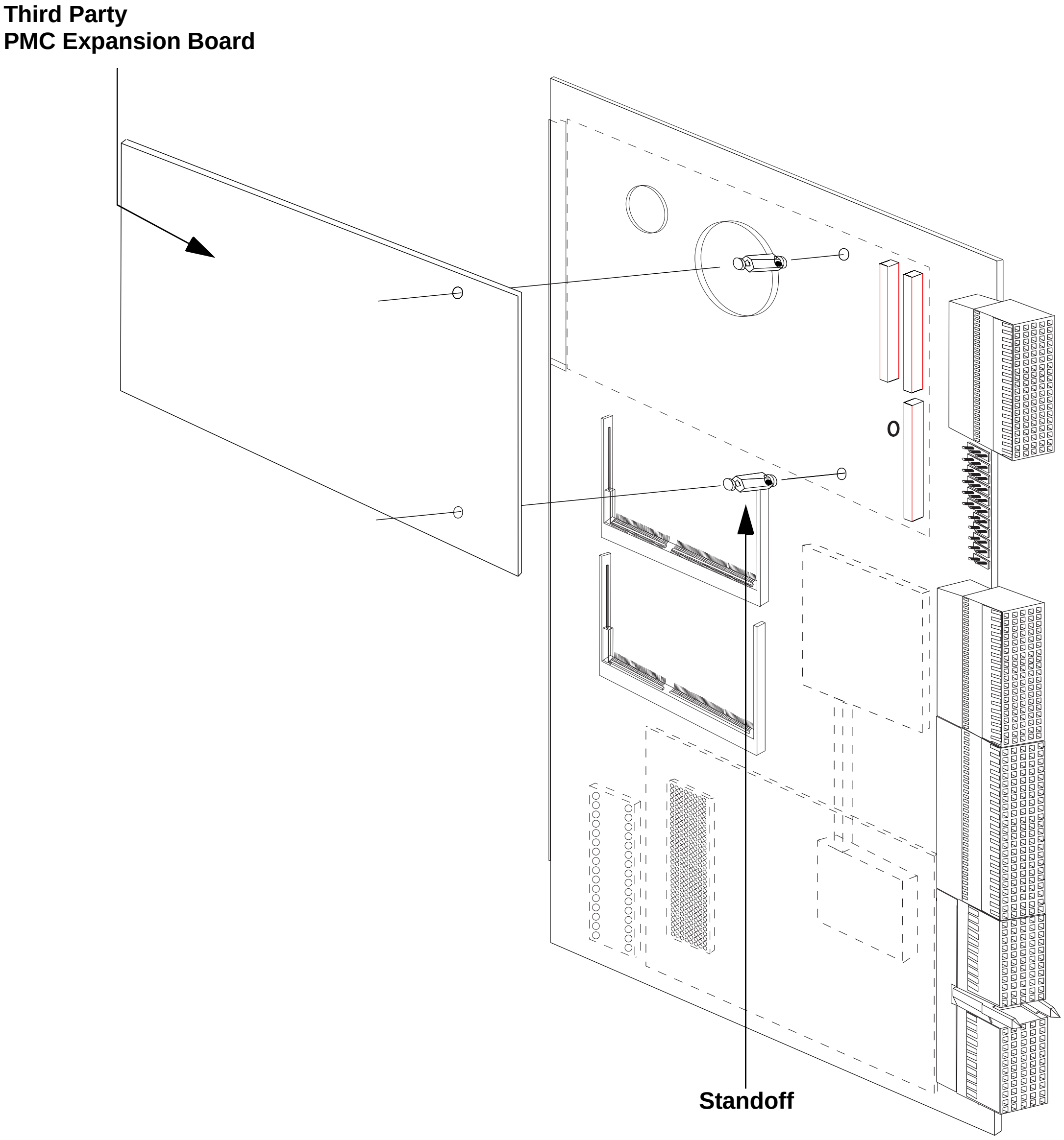


Figure 2-2 Installing a PMC Card on the VMICPCI-7697

LED Definition

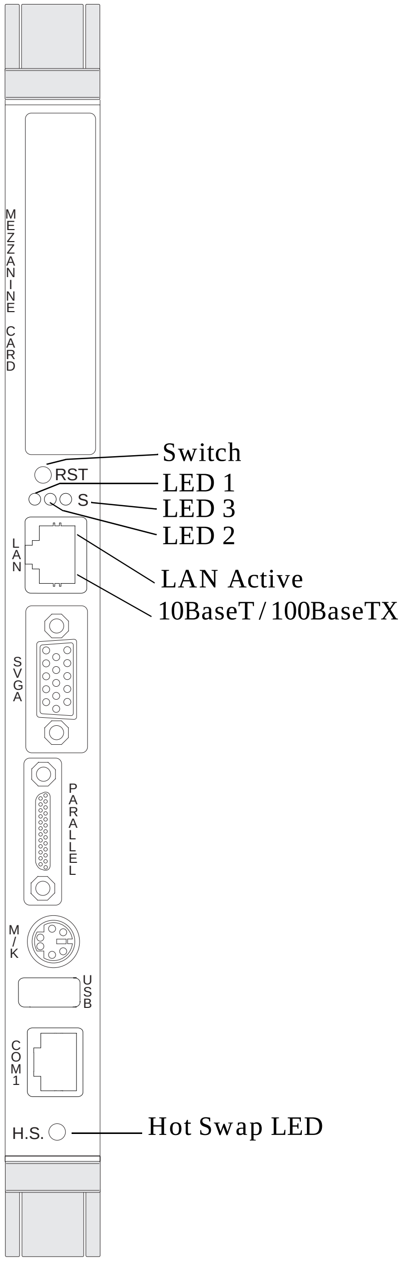


Figure 2-3 Front Panel LED Positions

Switch *Reset* - Allows the system to be reset from the front panel.

LED 1 *Power* - Indicates when power is applied to the board.

LED 2 *Hard Drive Indicator* - Indicates when hard drive activity is occurring.

LED 3 *System* - Indicates BIOS Boot is in progress. When LED is off, module is ready.

LAN Active Indicates the Ethernet is active, (yellow LED).

10 / 100BaseTX Indicates whether 10BaseT or 100BaseTX mode. Yellow LED for 10BaseT, and Green LED for 100BaseTX.

Hot Swap Lights during Hot Swap insertion and extraction.

CHAPTER

3

PC/AT Functions

Contents

Introduction

The VMICPCI-7697 is a complete IBM PC/AT-compatible Pentium II Embedded Module processor-based computer. The design includes a high-speed microprocessor with current technology memory, and optional Msystems DiskOnChip flash. Reference the VMIC product specifications for available component options.

Because the design is PC/AT compatible, it retains standard PC memory and I/O maps along with standard interrupt architecture. Furthermore, the VMICPCI-7697 includes a PCI-compatible Ethernet controller.

The following sections describe in detail the PC/AT functions of the VMICPCI-7697.

CPU Socket

The VMICPCI-7697 CPU socket is factory populated with a high-speed Pentium II processor. The CPU speed and RAM / flash size are user-specified as part of the VMICPCI-7697 ordering information. The CPU speed is not user-upgradable.

To change CPU speeds, RAM size, or flash size contact customer service to receive a Return Material Authorization (RMA).

VMIC Customer Service is available at: 1-800-240-7782.

Physical Memory

The VMICPCI-7697 provides Synchronous DRAM (SODIMM) as on-board system memory. Memory can be accessed as bytes, words, or longwords.

The VMICPCI-7697 accepts up to three 144-pin SODIMM DRAM modules for a maximum memory capacity of 512 Mbyte. The on-board DRAM is dual-ported to the CPCI bus through the PCI-to-PCI bridge. The memory is addressable by the local processor.

NOTE: When using the Configure utility of VMIC's IOWorks Access with Windows NT 4.0 to configure RAM, do not request more than 25 percent of the physical RAM. Exceeding the 25 percent limit may result in a known Windows NT bug causing unpredictable behavior during the Windows NT boot sequence and require the use of an emergency repair disk to restore the computer. The bug is present in Windows NT 4.0 service pack level 3. It is recommended that an emergency repair disk be kept up-to-date and easily accessible.

The VMICPCI-7697 provides 16 Mbyte of flash memory (See Note) accessible through the Secondary IDE port. The VMICPCI-7697 BIOS includes an option to allow the board to boot from the flash memory or from the Ethernet port.

The VMICPCI-7697 is also available with an optional M-systems DiskOnChip flash up to 144 Mbytes. Reference the VMICPCI-7697 Product Specification for available flash memory options.

The VMICPCI-7697 memory includes 32 Kbyte of battery-backed SRAM addressed at $D8000 to $DFFFF. All but the first 14 bytes are accessible as SRAM. Bytes $D8000 through $D800D are reserved for the Watchdog Timer registers. The battery-backed SRAM can be accessed by the CPU at anytime, and is used to store system data that must not be lost during power-off conditions. Battery support for the 32 Kbyte of SRAM is B1 and uses Jumper E10.

NOTE: Memory capacity may be extended as parts become available.

I/O Port Map

The Pentium II processor-based CPU includes special input/output instructions that access I/O peripherals residing in I/O addressing space (separate and distinct from memory addressing space). Locations in I/O address space are referred to as *ports*. When the CPU decodes and executes an I/O instruction, it produces a 16-bit I/O address on lines A00 to A15 and identifies the I/O cycle with the M/I/O control line. Thus, the CPU includes an independent 64 Kbytes I/O address space which is accessible as bytes, words, or longwords.

Standard PC/AT hardware circuitry reserves only 1,024 bytes of I/O addressing space from I/O $000 to $3FF for peripherals. All standard PC I/O peripherals such as serial and parallel ports, hard and floppy drive controllers, video system, real-time clock, system timers, and interrupt controllers are addressed in this region of I/O space. The BIOS initializes and configures all these registers properly; adjusting these I/O ports directly is not normally necessary.

The assigned and user-available I/O addresses are summarized in the I/O Address Map, Table 3-1.

Table 3-1 VMICPCI-7697 I/O Address Map

I/O ADDRESS RANGE	SIZE IN BYTES	HW DEVICE	PC/AT FUNCTION
$000 - $00F	16		DMA Controller 1 (Intel 8237A Compatible)
$010 - $01F	16		Reserved
$020 - $021	2		Master Interrupt Controller (Intel 8259A Compatible)
$022 - $03F	30		Reserved
$040 - $043	4		Programmable Timer (Intel 8254 Compatible)
$044 - $05F	30		Reserved
$060 - $064	5		Keyboard, Speaker, Eqpt. Configuration (Intel 8042 Compatible)
$065 - $06F	11		Reserved
$070 - $071	2		Real-Time Clock,
$072 - $07F	14		Reserved
$080 - $08F	16		DMA Page Registers
$090 - $091	2		Reserved
$092	1		Alt. Gate A20/Fast Reset Register
$093	1		Reserved

Table 3-1 VMICPCI-7697 I/O Address Map (Continued)

I/O ADDRESS RANGE	SIZE IN BYTES	HW DEVICE	PC/AT FUNCTION
$094	1		
$095 - $09F	11		Reserved
$0A0 - $0A1	2		Slave Interrupt Controller (Intel 8259A Compatible)
$0A2 - $0BF	30		Reserved
$0C0 - $0DF	32		DMA Controller 2 (Intel 8237A Compatible)
$0E0 - $16F	142	----	Reserved
$170 - $177	8	PIIX4E	Secondary Hard Disk Controller
$178 - $1EF	120		User I/O
$1F0 - $1F7	8	PIIX4E	Primary Hard Disk Controller
$1F8 - $277	128		User I/O
$278 - $27F	8	I/O Chip*	LPT2 Parallel I/O*
$280 - $2E7	104		Reserved
$2E8 - $2EE	7	UART*	COM4 Serial I/O*
$2EF - $2F7	9		User I/O
$2F8 - $2FE	7	Super-I/O Chip	COM2 Serial I/O (16550 Compatible)
$2FF - $36F	113		Reserved
$370 - $377	8	Super-I/O Chip	Secondary Floppy Disk Controller
$378 - $37F	8	Super-I/O Chip	LPT1 Parallel I/O
$380 - $3E7	108		Reserved
$3E8 - $3EE	7	UART*	COM3 Serial I/O*
$3F0 - $3F7	8	Super-I/O Chip	Primary Floppy Disk Controller
$3F8 - $3FE	7	Super-I/O Chip	COM1 Serial I/O (16550 Compatible)
$3FF - $4FF			Reserved
$500 - $503	4	82C54 Timer	Programmable Internal Timer
$504 - CFF			Reserved
* While these I/O ports are reserved for the listed functions, they are not implemented on the VMICPCI-7697. They are listed here to make the user aware of the standard PC/AT usage of these ports.			

Embedded PCI Bridge

The VMICPCI-7697 uses the Intel Embedded PCI bridge to interface between the primary PCI bus and the CompactPCI (CPCI) bus. The CompactPCI bus appears as a secondary PCI bus, and all devices in the seven peripheral slots of the CPCI chassis are accessed with normal PCI accesses. Complete details of the embedded PCI Bridge are included in Volume II of the VMICPCI-7697 manual.

PC/AT Interrupts

In addition to an I/O port address, an I/O device has a separate hardware interrupt line assignment. Assigned to each interrupt line is a corresponding interrupt vector in the 256-vector interrupt table at $00000 to $003FF in memory. The 16 maskable interrupts and the single Non-Maskable Interrupt (NMI) are listed in Table 3-2 along with their functions. Table 3-3 on page 44 details the vectors in the interrupt vector table. The interrupt number in HEX and decimal are also defined for real and protected mode in Table 3-3 on page 44.

The interrupt hardware implementation on the VMICPCI-7697 is standard for computers built around the PC/AT architecture, which evolved from the IBM PC/XT. In the IBM PC/XT computers, only eight interrupt request lines exist, numbered from IRQ0 to IRQ7 at the PIC. The IBM PC/AT computer added eight more IRQx lines, numbered IRQ8 to IRQ15, by cascading a second slave PIC into the original master PIC. IRQ2 at the master PIC was committed as the cascade input from the slave PIC. This architecture is represented in Figure 3-1 on page 48.

To maintain backward compatibility with PC/XT systems, IBM chose to use the new IRQ9 input on the slave PIC to operate as the old IRQ2 interrupt line on the PC/XT Expansion Bus. Thus, in AT systems, the IRQ9 interrupt line connects to the old IRQ2 pin (pin B4) on the AT Expansion Bus (or ISA bus).

Table 3-2 PC/AT Hardware Interrupt Line Assignments

IRQ	AT FUNCTION	COMMENTS
NMI	Parity Errors (Must be enabled in BIOS Setup)	Used by VMICPCI-7697 PCIbus Interface
0	System Timer	Set by BIOS Setup
1	Keyboard	Set by BIOS Setup
2	Duplexed to IRQ9	
3	COM2/COM4	
4	COM1/COM3	
5	Timer/I^{2}C	Assigned to On-Board Timer and I^{2}C Controller
6	Floppy Controller	
7	LPT1	

Table 3-2 PC/AT Hardware Interrupt Line Assignments (Continued)

IRQ	AT FUNCTION	COMMENTS
8	Real-Time Clock	
9	Old IRQ2	SVGA or Network I/O
10	Not Assigned	Determined by BIOS
11	Not Assigned	Determined by BIOS
12	Mouse	
13	Math Coprocessor	
14	AT Hard Drive	
15	Flash Drive	

Table 3-3 PC/AT Interrupt Vector Table

INTERRUPT NO.		IRQ LINE	REAL MODE	PROTECTED MODE
HEX	DEC			
00	0		Divide Error	Same as Real Mode
01	1		Debug Single Step	Same as Real Mode
02	2	NMI	Memory Parity Error, CompactPCI Interrupts	Same as Real Mode (Must be enabled in BIOS Setup)
03	3		Debug Breakpoint	Same as Real Mode
04	4		ALU Overflow	Same as Real Mode
05	5		Print Screen	Array Bounds Check
06	6			Invalid OpCode
07	7			Device Not Available
08	8	IRQ0	Timer Tick	Double Exception Detected
09	9	IRQ1	Keyboard Input	Coprocessor Segment Overrun
0A	10	IRQ2	BIOS Reserved	Invalid Task State Segment
0B	11	IRQ3	COM2 Serial I/O	Segment Not Present
0C	12	IRQ4	COM1 Serial I/O	Stack Segment Overrun
0D	13	IRQ5	Timer/I^{2}C Controller	Same as Real Mode
0E	14	IRQ6	Floppy Disk Controller	Page Fault
0F	15	IRQ7	LPT1 Parallel I/O	Unassigned

Table 3-3 PC/AT Interrupt Vector Table (Continued)

INTERRUPT NO.		IRQ LINE	REAL MODE	PROTECTED MODE
HEX	DEC			
10	16		BIOS Video I/O	Coprocessor Error
11	17		Eqpt Configuration Check	Same as Real Mode
12	18		Memory Size Check	Same as Real Mode
13	19		XT Floppy/Hard Drive	Same as Real Mode
14	20		BIOS Comm I/O	Same as Real Mode
15	21		BIOS Cassette Tape I/O	Same as Real Mode
16	22		BIOS Keyboard I/O	Same as Real Mode
17	23		BIOS Printer I/O	Same as Real Mode
18	24		ROM BASIC Entry Point	Same as Real Mode
19	25		Bootstrap Loader	Same as Real Mode
1A	26	IRQ8	Real-Time Clock	Same as Real Mode
1B	27		Control/Break Handler	Same as Real Mode
1C	28		Timer Control	Same as Real Mode
1D	29		Video Parameter Table Pntr	Same as Real Mode
1E	30		Floppy Parm Table Pntr	Same as Real Mode
1F	31		Video Graphics Table Pntr	Same as Real Mode
20	32		DOS Terminate Program	Same as Real Mode
21	33		DOS Function Entry Point	Same as Real Mode
22	34		DOS Terminate Handler	Same as Real Mode
23	35		DOS Control/Break Handler	Same as Real Mode
24	36		DOS Critical Error Handler	Same as Real Mode
25	37		DOS Absolute Disk Read	Same as Real Mode
26	38		DOS Absolute Disk Write	Same as Real Mode
27	39		DOS Program Terminate, Stay Resident	Same as Real Mode
28	40		DOS Keyboard Idle Loop	Same as Real Mode

Table 3-3 PC/AT Interrupt Vector Table (Continued)

INTERRUPT NO.		IRQ LINE	REAL MODE	PROTECTED MODE
HEX	DEC			
29	41		DOS CON Dev. Raw Output	Same as Real Mode
2A	42		DOS 3.x+ Network Comm	Same as Real Mode
2B	43		DOS Internal Use	Same as Real Mode
2C	44		DOS Internal Use	Same as Real Mode
2D	45		DOS Internal Use	Same as Real Mode
2E	46		DOS Internal Use	Same as Real Mode
2F	47		DOS Print Spooler Driver	Same as Real Mode
30-60	48-96		Reserved by DOS	Same as Real Mode
61-66	97-102		User Available	Same as Real Mode
67-71	103-113		Reserved by DOS	Same as Real Mode
72	114	IRQ10	Not Assigned	
73	115	IRQ11	Not Assigned	
74	116	IRQ12	Mouse	
75	117	IRQ13	Math Coprocessor	
76	118	IRQ14	AT Hard Drive	
77	119	IRQ15	Flash Drive	
78-7F	120-127		Reserved by DOS	Same as Real Mode
80-F0	128-240		Reserved for BASIC	Same as Real Mode
F1-FF	241-255		Reserved by DOS	Same as Real Mode

PCI Interrupts

Interrupts on Peripheral Component Interconnect (PCI) Local Bus are optional and defined as “level sensitive,” asserted low (negative true), using open drain output drivers. The assertion and de-assertion of an interrupt line, INTx#, is asynchronous to CLK. A device asserts its INTx# line when requesting attention from its device driver. Once the INTx# signal is asserted, it remains asserted until the device driver clears the pending request. When the request is cleared, the device de-asserts its INTx# signal.

PCI defines one interrupt line for a single function device and up to four interrupt lines for a multifunction device or connector. For a single function device, only INTA# may be used while the other three interrupt lines have no meaning. Figure 3-1 on page 48 depicts the VMICPCI-7697 interrupt logic pertaining to CompactPCI operations and the PMC site.

Any function on a multifunction device can be connected to any of the INTx# lines. The Interrupt Pin register defines which INTx# line the function uses to request an interrupt. If a device implements a single INTx# line, it is called INTA#; if it implements two lines, they are called INTA# and INTB#; and so forth. For a multifunction device, all functions may use the same INTx# line or each may have its own (up to a maximum of four functions) or any combination thereof. A single function can never generate an interrupt request on more than one INTx# line.

The slave PIC accepts the CompactPCI interrupts through lines that are defined by the BIOS. The BIOS defines which interrupt line to utilize depending on which system requires the use of the line.

The PCI-to-PCI Bridge has the capability of generating a Non-Maskable Interrupt (NMI) via the PCI SERR# line. Table 3-4 describes the register bits that are used by the NMI. The SERR interrupt is routed through logic back to the NMI input line on the CPU. The CPU reads the NMI Status Control register to determine the NMI source (bits set to 1). After the NMI interrupt routine processes the interrupt, software clears the NMI status bits by setting the corresponding enable / disable bit to 1. The NMI Enable and Real-Time Clock register can mask the NMI signal and disable / enable all NMI sources.

Table 3-4 NMI Register Bit Descriptions

Status Control Register (I/O Address $061, Read/Write, Read Only)	
Bit 7	SERR# NMI Source Status (Read Only) - This bit is set to 1 if a system board agent detects a system board error. It then asserts the PCI SERR# line. To reset the interrupt, set bit 2 to 0 and then set it to 1. When writing to port $061, bit 7 must be 0.
Bit 2	PCI SERR# Enable (Read/Write) - 1 = Clear and Disable, 0 = Enable
Enable and Real-Time Clock Address Register (I/O Address $070, Write Only)	
Bit 7	NMI Enable - 1 = Disable, 0 = Enable

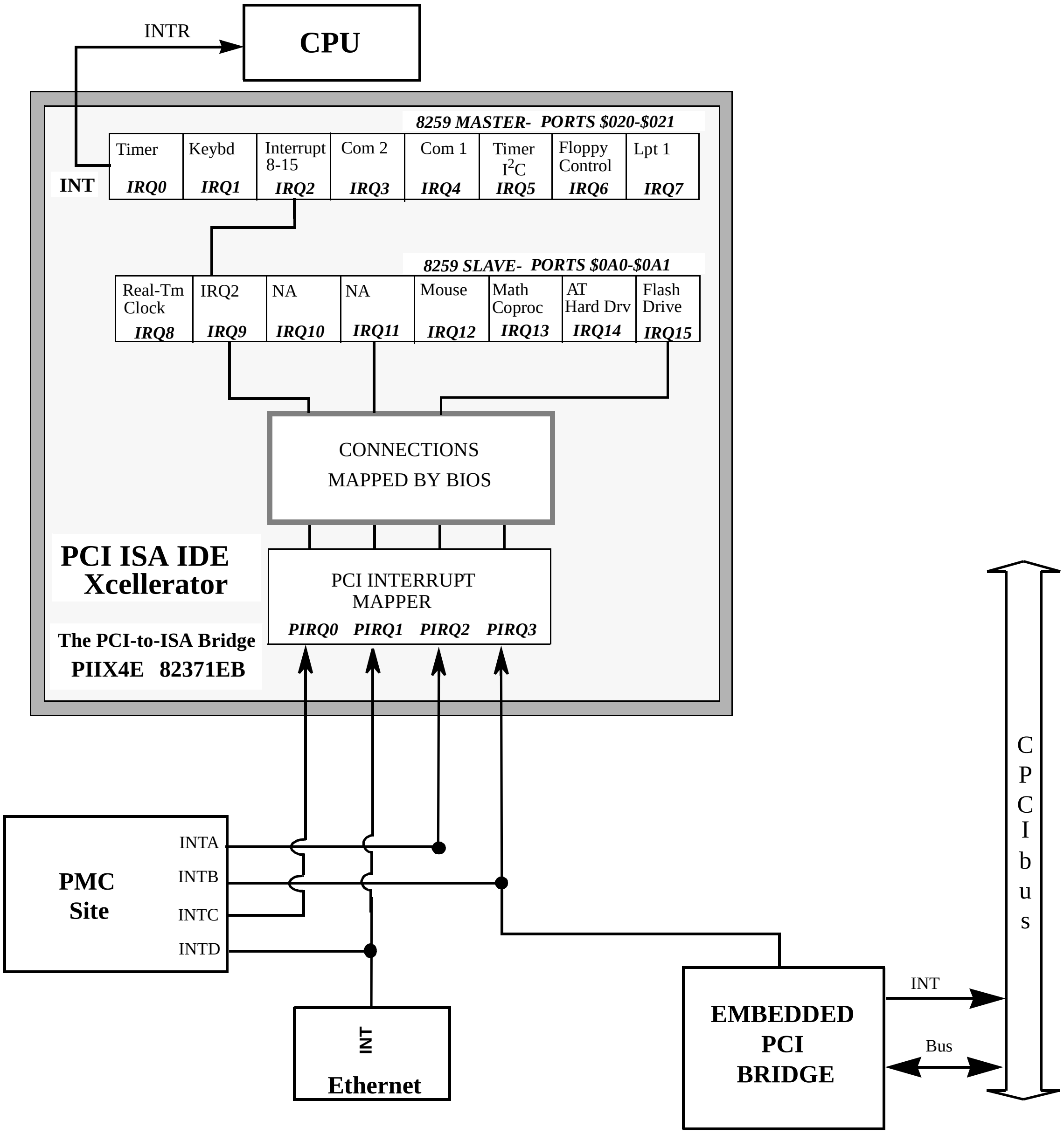


Figure 3-1 Connections for the PC Interrupt Logic Controller

Integrated Peripherals

The VMICPCI-7697 incorporates the SMC Super-I/O chip. The SMC chip provides the VMICPCI-7697 with a standard floppy drive controller, two 16550 UART-compatible serial ports, and one enhanced parallel port with ECP/EPP modes. The Ultra-EIDE hard drive interface is provided by the Intel 82371EB (PIIX4E) PCI ISA IDE Xcelerator chip. All ports are located on the J3 connector, for use with VMIC's VMIACC-0576 CompactPCI Rear Transition Utility Board.

Ethernet Controller

The network capability is provided by the DEC’s 21143 Ethernet Controller. This Ethernet controller is PCI bus based and is software configurable. The VMICPCI-7697 supports 10BaseT and 100BaseTx Ethernet.

10BaseT

A network based on the 10BaseT standard uses unshielded twisted-pair cables, providing an economical solution to networking by allowing the use of existing telephone wiring and connectors. The RJ-45 connector is used with the 10BaseT standard. 10BaseT has a maximum length of 100 meters from the wiring hub to the terminal node.

100BaseTx

The VMICPCI-7697 also supports the 100BaseTx Ethernet. A network based on a 100BaseTx standard uses unshielded twisted-pair cables and a RJ-45 connector. The 100BaseTx has a maximum deployment length of 100 meters.

Remote Ethernet Booting

The VMICPCI-7697 is capable of booting from a server over the Ethernet. Utilizing Lanworks BootWare. BootWare gives you the ability to remotely boot the VMICPCI-7697 using Netware, TCP/IP, or RPL network protocols. The Ethernet must be connected through the LAN front panel (RJ-45) connector to boot remotely. This feature allows users to create systems without the worry of disk drive reliability, or the extra cost of adding Flash drives.

BootWare Features:

- Netware, TCP/IP, RPL Compatible
- Unparalleled boot sector virus protection
- Detailed boot configuration screens
- Comprehensive diagnostics
- Optional disabling of local boots
- Dual-boot option lets users select network or local booting

Video Graphics Adapter

The SVGA port on the VMICPCI-7697 is controlled by a S3 Trio 3D AGP chip with 4 Mbyte video DRAM. The video controller chip is hardware and BIOS compatible with the IBM EGA and SVGA standards supporting both VESA high-resolution and extended video modes. Table 3-5 shows the graphics video modes supported by the Trio 3D video chip.

Table 3-5 Supported Graphics Video Resolutions

Screen Resolution	Maximum Colors	Maximum Refresh Rates (Hz)
640 x 480	16 M	85
800 x 600	16 M	85
1,024 x 768	16 M	85
1,280 x 1,024	64 K	60
1600 x 1200	64 K	60

Not all SVGA monitors support resolutions and refresh rates beyond 640 x 480 at 60 Hz. Do not attempt to drive a monitor to a resolution or refresh rate beyond its capability.

The VMICPCI-7697's processor includes a 64-bit access to video memory with no-wait states. Video I/O registers are accessed using the AGP bus.

CHAPTER

4

Embedded PC/RTOS Features

Contents

Introduction

VMIC's VMICPCI-7697 features additional capabilities beyond those of a typical IBM PC / AT-compatible CPU. The unit provides three software-controlled, general-purpose timers along with a programmable Watchdog timer for synchronizing and controlling multiple events in embedded applications. The VMICPCI-7697 also provides a bootable Flash Disk system and 32 kbyte of nonvolatile, battery-backed SRAM. These features make the unit ideal for embedded applications, particularly applications where standard hard drives and floppy disk drives cannot be used.

Timers

General

The VMICPCI-7697 provides a user-programmable 82C54 internal timer/counter. The 82C54 provides three independent, 16-bit timers each operating at a 1 or 2 MHz clock speed. This is determined by the configuration of jumper E11; reference Chapter 2. These timers are completely available to the user, and are not dedicated to any PC/AT function. These timers may be used to generate system interrupts.

Events can be timed by either polling the timers or generating a system interrupt via circuity external to the 82C54. The external circuity consists of logic which generates the interrupt and a Timer Interrupt Status register which indicates which of the three Timers generated an interrupt.

The 82C54 timers are mapped at I/O address $500. The interrupt used by the Timers is IRQ5. The Timer Interrupt Status register is available via the Power Management I/O address space. The access to this space is explained in the Timer Interrupt Status section.

Timer Interrupt Status

A single interrupt, IRQ5, is used by all three Timers. A Timer Interrupt Status register is provided in order to determine which Timer(s) initiated an interrupt. The Interrupt Status Register is a general-purpose input register located (refer to Figure 4-1), external to the 82C54, at offset $31 from the Power Management Base I/O address. The interrupt status register address can be found by first determining the PCI Configuration Base address for Device ID $7113 and Vendor ID $8086. The Power Management Base I/O address can be found by reading offset $40 from this PCI Configuration address. The Timer Interrupt Status register bits are located at offset $31 from the Power Management Base I/O address, bits 5, 6, and 7.

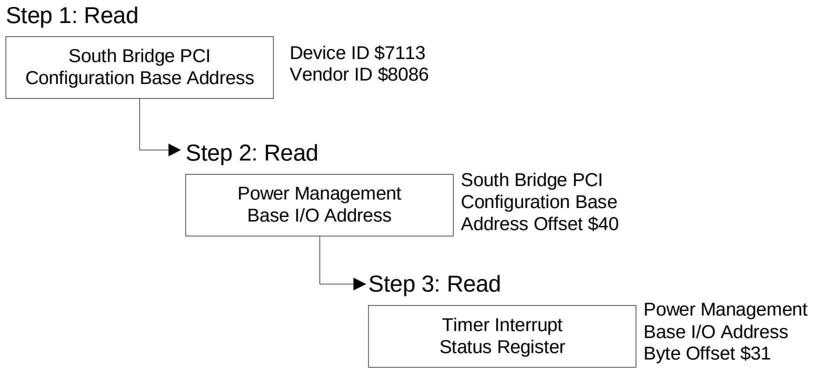

Step 1: Read
South Bridge PCI Configuration Base Address
Device ID $7113
Vendor ID $8086
Step 2: Read
Power Management Base I/O Address
South Bridge PCI Configuration Base Address Offset $40
Step 3: Read
Timer Interrupt Status Register
Power Management Base I/O Address Byte Offset $31

Figure 4-1 Timer Interrupt Status Register Read/Steps

A byte read of offset $31 from the Power Management Base I/O address is used to obtain these bits. Bits 5, 6, and 7 correspond to Timers 2, 1, and 0, respectively

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Timer 0	Timer 1	Timer 2	Unused	Unused	Unused	Unused	Unused

Power Management Base Address
Byte Offset $31

Figure 4-2 Timer Interrupt Status Register

In order to clear the Timer Interrupt Status register, first write zeros (0’s) to the general-purpose output register located at offset $37 of the Power Management Base I/O address bits 3, 4, and 6 (Not bits 3, 4 and 5). Then write ones (1’s) to these same bits to re-enable the Timer Interrupt Status register. Bits 3, 4, and 6 correspond to Timers 2, 1, and 0, respectively.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Unused	Timer 0	Unused	Timer 1	Timer 2	Unused	Unused	Unused

1. Write zeros (0’s) to Power Management Base Address Byte Offset $37 bits 3, 4, and 6
2. Write ones (1’s) to Power Management Base Address Byte Offset $37 bits 3, 4, and 6
   Bits 0, 1, 2, 5, and 7 should remain unaffected

Figure 4-3 Clearing the Timer Interrupt Status Register

Clearing the Interrupt

The Timer Interrupts are cleared using the standard procedure for clearing PC/AT IRQ5. Refer to Appendix F for example code using the 82C54 timers.

Timer Programming

Architecture

The VMICPCI-7697 Timers are mapped in I/O address space starting at $500. See Table 4-1 below. The Timers, consisting of three 16-bit timers and a Control Word register (see Figure 4-4 on page 56) are read from/written to by way of a 8-bit data bus.

Table 4-1 I/O Address of the Control Word Register and Timers

I/O Address	Select
$500	Timer 0
$501	Timer 1
$502	Timer 2
$503	control word Register

Table 4-1 on page 55 shows the I/O address's of the control word Register and Timers.

The control word Register is write only. The Timer status information can be obtained from the Read-Back command (see the Section titled Reading on page 59).

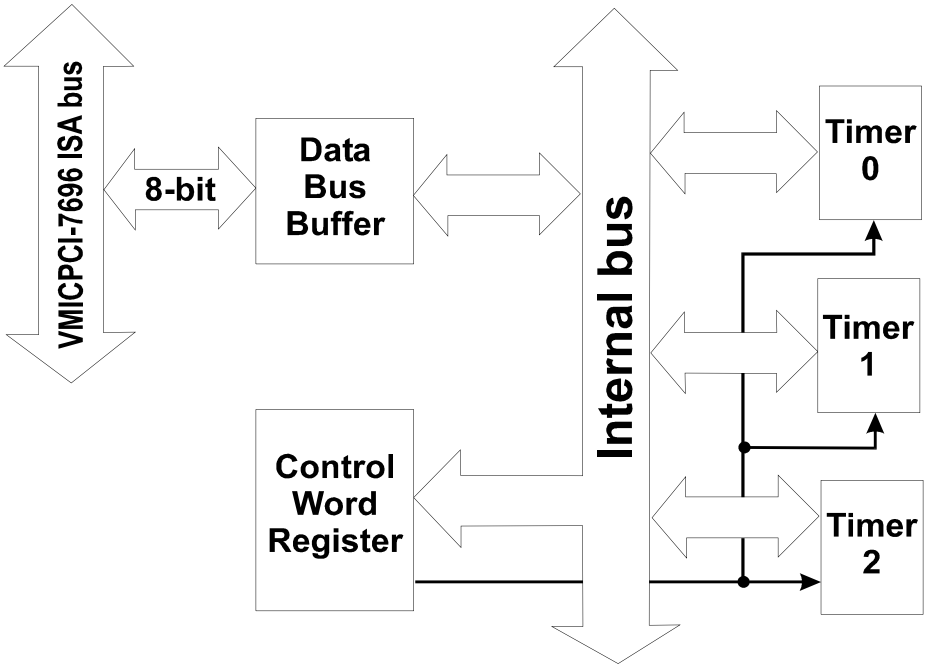


Figure 4-4 82C54 Diagram

The three timers, Timer 0, 1, and 2, are functionally equivalent, therefore only a single Timer will be described. Figure 4-5 on page 57 is a block diagram of a Timer. Each Timer is functionally independent. Although the control word is shown in the Timer block diagram, it is not a part of the Timer, but its contents directly affect how the Timer functions.

The status register, when latched (as shown in Figure 4-5 on page 57), contains the present contents of the Control Word Register and the present state of the output and load count flag. The Status Word is available via the Read-Back command, see the Section titled Reading on page 59. The Timer is labelled TE (Timer Element). It is a 16-bit synchronous presettable down counter.

The blocks labelled OL_M and OL_L are 8-bit Output Latches (OL). The subscripts M and L stand for Most Significant byte and Least Significant byte. These latches usually track the TE but when commanded will latch and hold the present count until the CPU reads the count. When the latched count is read, the OL registers will continue to track the TE. When reading the OL registers, two 8-bit accesses must be performed to retrieve the complete 16-bit value of the Timer as only one latch at a time is enabled. The TE cannot be read, the count is read from the OL registers.

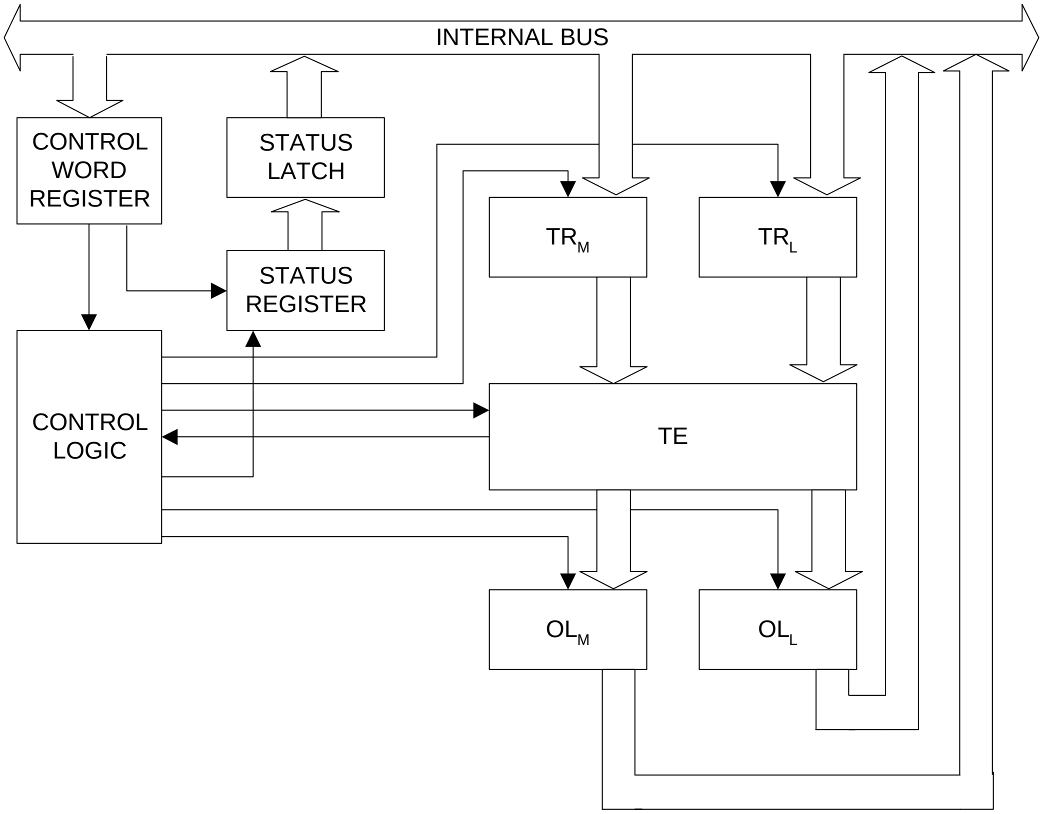


Figure 4-5 Internal Timer Diagram

There are two 8-bit registers labeled TR_M and TR_L (Timer Register). The subscripts M and L stand for Most Significant byte and Least Significant byte. When a new count is written to the Timer the count is loaded into the TR and later transferred to the TE. The Control logic lets one 8-bit TR register be written to at a time. Two 8-bit writes must be performed to load a complete 16-bit count value. Both TR bytes are transferred to the TE at the same time. The TE cannot be directly written to by the user, the count is written to the TR registers then latched to the TE.

Writing

The Timers are programmed by first writing a control word and then the initial count. The format of the control word is shown in Tables 4-2 through 4-6. All control words are written into the control word Register while the initial counts are written into the individual Timer registers. The format of the initial count is determined by the control word.

Table 4-2 Control Word Format

D7	D6	D5	D4	D3	D2	D1	D0
ST1	ST0	RW1	RW0	M2	M1	M0	BCD

Table 4-3 ST - Select Timer

ST1*	ST0*	Description
0	0	Select Timer 0
0	1	Select Timer 1
1	0	Select Timer 2
1	1	Read-Back Command (See Reading section on page 1-59)
* The ST bits specify which Timer (0, 1, or 2) the control word refers to and whether this is a Read-Back command		

Table 4-4 RW - Read/Write

RW1*	RW0*	Description
0	0	Timer Latch Command (see Reading section)
0	1	Read / Write least significant byte only
1	0	Read / Write most significant byte only
1	1	Read / Write least significant byte first, then most significant
* The RW bits specify whether this is a Timer Latch command or the byte ordering of the Read / Write transaction.		

Table 4-5 M - Mode

M2*	M1*	M0*	Description
0	0	0	Mode 0
0	0	1	Mode 1
X	***1***	***0***	***Mode 2***
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5
* Only Mode 2 is described in this manual.			

Table 4-6 BCD

BCD*	Description
0	Binary Timer 16-bits
1	Binary Coded Decimal (BCD) Timer (4 Decades)
* The BCD bit specifies whether the Timer count value is in Binary or BCD.	

When programming the 82C54, only two rules need to be followed.

1. For each Timer, the control word must be written first.
2. The initial count must follow the format specified in the control word (least significant byte only, most significant byte only, or least significant byte then most significant byte). As long as these rules are adhered to, any programming sequence is acceptable.

Reading

There are two methods for reading the timers: the Timer Latch Command and the Read-Back Command.

Timer Latch Command

The Timer Latch Command allows the reading of a Timer 'on the fly' without affecting the timing in process.

Like a control word, the Timer Latch Command is written to the Control Word Register (I/O Address $503, see Table 4-1 on page 55). The Select Timer bits (ST1, ST0, see Table 4-3 on page 58) select one of the three timers while the Read/Write bits (RW1,RW0, see Table 4-4 on page 58) select the Timer Latch Command, RW1 = 0 and RW0 = 0. The selected Timer's count is latched into the 0L registers at the time of the Timer Latch Command. The count is held in the 0L latches until it is read. Multiple Timer Latch Commands can be used to latch more than one Timer. Again, each Timer's count is held latched until it is read.

Read-Back Command

The Read-Back Command allows the user to view the Timer count, the Timer Mode, the current state of the OUT pin, and the Load Flag of the selected Timer. Like a control word, the Read-Back Command is written into the Control Word Register and has the format shown in Tables 4-7 and 4-8 below. The Command applies to the Timer(s) selected by setting the corresponding bits Cnt2, Cnt1, Cnt0 = 1.

Table 4-7 Read-Back Command Format

D7	D6	D5	D4	D3	D2	D1	D0
1	1	$\overline{\text{Count}}$	$\overline{\text{Status}}$	Cnt2	Cnt1	Cnt0	0

Table 4-8 Read-Back Command Description

Bit	Description
D5: $\overline{\text{Count}}$	Latch count of selected Timer(s)
D4: $\overline{\text{Status}}$	Latch status of selected Timer(s)
D3: Cnt2	Select Timer 2
D2: Cnt1	Select Timer 1
D1: Cnt0	Select Timer 0
D0	Reserved, must be 0

The Read-Back Command can be used to latch several Timer counts by setting the $\overline{\text{Count}}$ bit = 0 and selecting the Timers. This is the same as using multiple Timer Latch Commands. Again, each Timer's latched count will be held until it is read.

The Read-Back command can also be used to latch the timer status by setting the $\overline{\text{Status}}$ bit = 0 and selecting the Timers. Status of a Timer is accessed by a read from that Timer (see Table 4-1 on page 55). If more than one Timer Status Read-Back command is issued without reading the status, all but the first is ignored.

The format of the Timer Status byte is shown in Tables 4-9 and 4-10 on following page.

Table 4-9 Status Byte

D7	D6	D5	D4	D3	D2	D1	D0
OUT	LOAD	RW1	RW0	M2	M1	M0	BCD

Table 4-10 Status Byte Description

Bit	Description
D7: OUT	Current state of Timers OUT pin
D6: LOAD	Count loaded into Timer
D5-D0	Timer Programmed Mode

Bit D7 contains the state of the Timers OUT pin. This allows viewing of the Timer's OUT pin via software.

Bit D6 indicates that the count written to the Timer is actually loaded into the Timer register. The exact time of the loading depends on the Mode the Timer is in and is defined in the Mode Definitions section. The count cannot be read from the Timer until it has been loaded. If a count is read before this time, the value read will not be the new count just written. Refer to Table 4-11.

Bits D5 through D0 contain the Timer's programmed mode exactly, bit for bit, like the Timer control words bits D5 through D0. See Table 4-2 on page 58.

Table 4-11 LOAD Bit Operation

Action	Causes
1. Write to the control word Register [1]	LOAD bit = 1
2. Write count to Timer [2]	LOAD bit = 1
3. New count loaded into Timer	LOAD bit = 0
[1] Only the Timer specified in the control word will have its LOAD bit set to 1. LOAD bits of other Timers are not affected. *[2] If the Timer is programmed for two byte counts (least significant, then most significant), the LOAD bit will go to 1 when the second byte is written.*	

Both the count and status of the specified Timer(s) can be latched at the same time by setting both the $\overline{\text{Count}}$ bit (D5) and $\overline{\text{Status}}$ bit (D4) to zero (0) in the Read-Back command. If this technique is used, the first read operation of the Timer will return the status while the next one or two reads (depending on whether the Timer is programmed for one or two bytes) will return the count. Succeeding reads will return unlatched counts.

Mode Definitions

The VMICPCI-7697 utilizes an 82C54 Timer / Counter for its Timers. 82C54 Timer / Counters can be programmed to function in six different modes (numbered Mode 0 through Mode 5). The VMICPCI-7697 Timers are hardware configured to operate using Mode 2. Only Mode 2 is defined.

Mode 2 functions as a divide by N counter. Once a control word and an initial count are written to the Timer, the initial count is loaded on the next Clock cycle. When the count decrements to 1, an interrupt is generated. The Timer then reloads the initial count and the process repeats. This Mode is periodic. For an initial count of N, the sequence repeats every N CLK cycles. An initial count of 1 is illegal.

Writing a new count while the Timer is counting does not affect the current sequence. The new count will be loaded at the end of the current sequence.

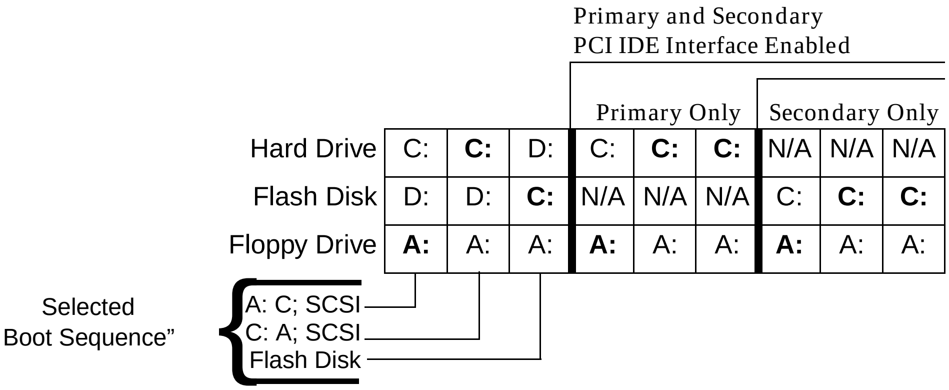
Flash Disk

The VMICPCI-7697 features an on-board 16 Mbyte flash mass storage system. This Flash Disk appears to the user as an intelligent ATA (IDE) disk drive with the same functionality and capabilities as a “rotating media” IDE hard drive. The VMICPCI-7697 BIOS includes an option to allow the board to boot from the Flash Disk.

Configuration

The Flash Disk resides on the VMICPCI-7697 as the secondary IDE bus master device (the secondary IDE bus slave device is not assignable). The default setting in the AWARD BIOS ‘STANDARD CMOS SETUP’ screen is the ‘AUTO’ setting. In the AWARD BIOS ‘PERIPHERAL SETUP’ screen, the secondary PCI IDE interface must be enabled for the Flash Disk to be functional. Refer to Appendix D, Award-Basic Input/Output System (BIOS) for additional details.

Figure 4-6 maps the configuration possibilities for a typical system consisting of the VMICPCI-7697 with a resident Flash Disk, a hard drive attached to the Primary IDE interface, and a floppy drive attached to the floppy interface.



				Primary and Secondary PCI IDE Interface Enabled					
				Primary Only			Secondary Only		
Hard Drive	C:	**C:**	D:	C:	**C:**	**C:**	N/A	N/A	N/A
Flash Disk	D:	D:	**C:**	N/A	N/A	N/A	C:	**C:**	**C:**
Floppy Drive	**A:**	A:	A:	**A:**	A:	A:	**A:**	A:	A:

Figure 4-6 Typical System Configuration

The Primary and Secondary PCI IDE Interfaces are controlled (enabled or disabled) in the Integrated Peripheral Setup screen of the AWARD BIOS. The First Boot Device is selected in the BIOS Features Setup screen.

Figure 4-6 identifies the drive letter assigned to each physical device, and indicates in bold lettering the device booted from in each configuration, using devices that were bootable. Bootable being a device on which an operating system has been installed, or formatted as a system disk using MS-DOS.

NOTE: If, during the configuration efforts, a drive device is identified as a USER setting in the Standard CMOS Setup screen (as occurs after using the Auto-Detect Hard Drives function of the BIOS), then disabled (the device's IDE interface disabled) in the Peripheral Setup screen, the following error may occur:

```
Primary (or Secondary) Master HDD Error
Run Setup
Press <ESC> to Resume
```

Press the Escape key as directed to resume the boot procedure. The settings should be changed to prevent the reoccurrence of this error condition

Functionality

The Flash Disk performs identically to a standard IDE hard drive. Reads and writes to the device are performed using the same methods, utilizing DOS command line entries or the file managers resident in the chosen operating system.

Advanced Configuration

The previous discussion is based on using the IDE disk devices formatted as one large partition per device. Some applications may require the use of multiple partitions. The following discussion of these partitions includes special procedures that must be followed to create multiple partitions on the VMICPCI-7697 IDE disk devices (including the resident Flash Disk).

Partitions may be either a primary or an extended partition. An extended partition may be subdivided farther into logical partitions. Each device may have up to four main partitions; one of which may be an extended partition. However, if multiple primary partitions are created, only one partition may be active at a time. Data in the non-active partitions are not accessible.

Following the creation of the partitioning scheme, the partitions can be formatted to contain the desired file system.

As discussed earlier, a typical system consists of the VMICPCI-7697 with its resident Flash Disk configured as the Secondary IDE device, a hard drive attached to the Primary IDE interface, and a floppy drive attached to the floppy interface.

Using this configuration, it may be desirable to have a logical device on either IDE device, configured as a bootable device, allowing the selection of the first boot device by way of the Advanced CMOS Setup screen. Using this capability, a user could have a system configured with multiple operating systems that would be selectable by assigning the IDE logical device as the boot device.

The DOS utility FDISK is commonly used to configure the partition structure on a hard drive. Comments on the following page pertain to partitioning efforts using FDISK.

CAUTION: Deleting a partition will erase all the data previously stored in that partition.

The Flash Disk will be configured as a single partition device as delivered from the factory. The following sample sequence illustrates a proven method for creating two 8 Mbyte partitions, with one as an active primary partition. Take note of the instructions to exit FDISK. This has been shown to be an important step in a successful partitioning effort.

1. Power up the VMICPCI-7697 and enter the CMOS set-up.
2. Set IDE HDD Master to “Not Installed”.
3. Set Flash Disk Master to “AUTO”.
4. Set boot device to floppy.
5. Boot DOS from the floppy, and verify that the System Configuration Screen shows only the Flash Disk.
6. Run FDISK.
7. Delete all current partitions (any data currently stored in the partitions will be lost).
8. Exit FDISK (this will cause a reboot), then run FDISK again.
9. Create a 8 Mbyte primary partition.
10. Create a 8 Mbyte extended partition.
11. Set-up a logical device for the 8 Mbyte extended partition.

NOTE: If only one partition is required, it will be a 16 Mbyte partition.

12. Set the Primary partition as an active partition.
13. Exit FDISK.

If an operating system has been installed on the Flash Disk that modifies the Master Boot Record (MBR), the following steps are required to rewrite the MBR for DOS.

14. Run FDISK/MBR
15. Run FORMAT C: (use the extension /s option if you want the Flash Disk as a bootable DOS device.)
16. Format D: (this is only required if two partitions were created).
17. Reset the CPU and enter the CMOS set-up.
18. Set Primary Master to “AUTO”.
19. Set boot device to desired boot source.

Drive letter assignments for a simple system were illustrated in Figure 4-6 on page 63. Understanding the order the operating system assigns drive letters is necessary for these multiple partition configurations. The operating system assigns drive letter C: to the active primary partition on the first hard disk (the boot device). Drive D: is assigned to the first recognized primary partition on the next hard disk. The operating system will continue to assign drive letters to the primary partitions in an alternating fashion between the two drives. The next logical partitions will be assigned drive letters starting on the first hard drive lettering each logical device sequentially until all are assigned a drive letter, then doing the same sequential lettering of each logical partition on the second hard disk.

NOTE: Drive letter changes caused by adding an additional drive or changing the initial partitioning scheme may cause difficulties with an operating system installed prior to the changes. Plan your configuration prior to installing the operating system to minimize difficulties.

Flash Disk (Optional)

The VMICPCI-7697 is available with an optional single DiskOnChip which is plugged into a standard 32-pin EEPROM socket. The DiskOnChip is mapped into a 32 Kbyte window in the BIOS expansion address space of the PC, located between address 0xD0000 to 0xD1FFF (default).

The DiskOnChip contains a built-in copy of the M-Systems industry-standard TrueFFS software, which makes the DiskOnChip operate as a standard disk drive. The DiskOnChip can contain the operating system in it to allow systems to boot without a hard disk. The DiskOnChip can also be configured as the boot device in systems with a hard disk.

Installing the DiskOnChip

1. To install the DiskOnChip as drive C on a system without a hard disk, set the CMOS setup of the Primary Master and Primary Slave to Not Installed (indicating that no physical magnetic disk is installed), and reboot the computer. The DiskOnChip will install as drive C. The DiskOnChip needs to be formatted with the System files in order for it to be a bootable drive. See “Configuring the DiskOnChip as the Boot Device” below.
2. To install the DiskOnChip as a logical drive on a system with a hard disk, reboot the system and the DiskOnChip will install as the last drive.
3. To install the DiskOnChip as drive C on a system with a hard disk, see “Configuring the DiskOnChip as the First Drive”.

NOTE: The active (bootable) partition will always be drive C.

Configuring the DiskOnChip as the Boot Device

In order to configure the DiskOnChip as the boot device, the operating system files must be copied into the chip. Copying the operating system files into the DiskOnChip should be done in the same manner as any other hard disk. The following is a example of a typical initialization process:

1. Set the DiskOnChip as a regular drive in your system (not a boot drive).
2. Insert a bootable floppy diskette in drive A and boot the system.
3. At the DOS prompt, type SYS C: to transfer the DOS system files to the DiskOnChip (assuming the DiskOnChip is installed as drive C).
4. Copy any files needed into the DiskOnChip.
5. Remove the floppy diskette and reboot the system. The system will boot from the DiskOnChip and will allow you to run and access any files that have been copied into the DiskOnChip.

Configuring the DiskOnChip as the First Drive

The DiskOnChip can be configured as the last drive (default), or as the first drive in the system. When configured as the last drive, the DiskOnChip is installed as the last drive if there is another drive installed, and as drive C if no other drive is installed. When configured as the first drive, the DiskOnChip is always installed as drive C. The DiskOnChip is shipped from the factory configured to install as the last drive. It is possible to configure the DiskOnChip as the first drive using the utilities available on M-systems Web site (www.m-sys.com) To configure the DiskOnChip to install as the first drive, follow the steps below:

1. Boot the system and make sure the DiskOnChip is installed correctly as drive D.
2. Insert disk containing M-systems utilities (downloaded from Web site) into the floppy drive, type **A:** and press **Return**.
3. At the DOS prompt type: **DUPDATE D:/FIRST/S:DOC2000.EXB**.
4. After rebooting the system, the DiskOnChip will appear as drive C:.

Using DiskOnChip With Different Operating Systems

The DiskOnChip can operate with the DOS operating system without using driver software. However, in order to operate with other operating systems such as Windows NT 4.0, a DiskOnChip driver must be installed. Details on installing the Msystems DiskOnChip driver for Windows NT 4.0 are shown in Appendix B. For details on installing drivers for other operating systems, please contact the Msystems web site at www.m-sys.com, and access the technical support section.

Watchdog Timer

The VMICPCI-7697 utilizes a Dallas DS1384 Watchdog Timekeeping Controller as its Watchdog Timer. The device provides a Time of Day feature, a Watchdog Alarm and an Non-Volatile SRAM controller. The Time of Day feature found within the DS1384 device is explained in this section. The actual Time of Day registers used by the VMICPCI-7697 are located at the standard PC/AT I/O address. The Time of Day feature in the DS1384 Watchdog Timer is available for use at the user's discretion.

The Non-Volatile SRAM is explained in the battery-backed SRAM section of this manual.

The Watchdog Timer provides a Watchdog alarm window and interval timing between 0.01 and 99.99 seconds. If enabled, and if jumper E7 pins 1 and 2 are loaded, the Watchdog alarm will reset the CPU to a known state if not accessed during the alarm window. If pins 2 and 3 are jumpered, the watchdog alarm can be programmed to generate a Non-Maskable Interrupt (NMI) to the CPU.

Figure 4-7 shows a generalized block diagram of how the Watchdog Timer is used in the VMICPCI-7697. The Watchdog Timer registers are memory-mapped in the bottom fourteen locations of battery-backed SRAM addresses $D8000 through $D800D. Table 4-12 on page 70 shows the address, content and the range of each Watchdog Register.

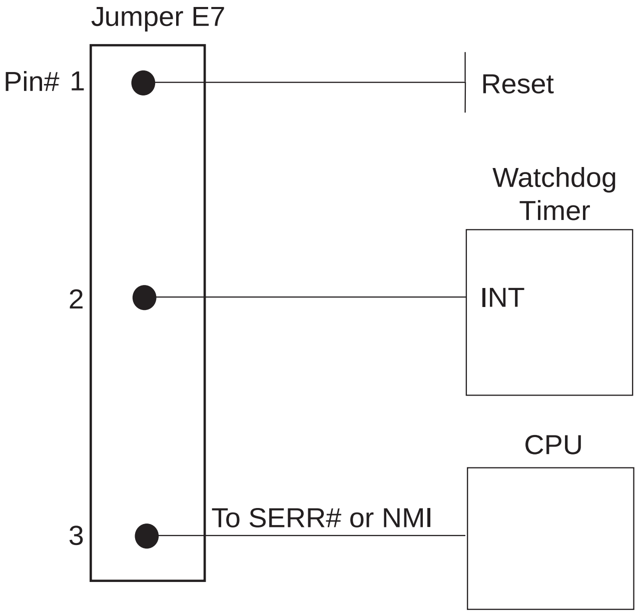


Figure 4-7 Watchdog Alarm Block

Table 4-12 Watchdog Registers

Register	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Range
0	$D8000	0.1 Seconds (BCD)				0.01 Seconds (BCD)				00 - 99
1	$D8001	10 Seconds (BCD)				Seconds (BCD)				00 - 59
2	$D8002	10 Minutes (BCD)				Minutes (BCD)				00 - 59
3	$D8003	M	10 Minute Alarm (BCD)			Minute Alarm (BCD)				00 - 59
4	$D8004	0	12/24	AM/PM *	10 Hr	Hours (BCD)				
5	$D8005	M	12/24	AM/PM *	10 Hr	Hour Alarm (BCD)				
6	$D8006	0	0	0	0	Days (BCD)				01 - 07
7	$D8007	M	0	0	0	Day Alarm (BCD)				01 - 07
8	$D8008	0	0	10 Date (BCD)		Date (BCD)				01 - 31
9	$D8009	Eosc	1**	0	10 Mo	Months (BCD)				01 - 12
A	$D800A	10 Years (BCD)				Years (BCD)				00 - 99
B	$D800B	Te	Ipsw	Ibh/lo	Pu/lvl	Wam	Tdm	Waf	Tdf	
C	$D800C	0.1 Seconds (BCD)				0.01 Seconds (BCD)				00 - 99
D	$D800D	10 Seconds (BCD)				Seconds (BCD)				00 - 99

* In the 12 hour mode bit 5 determines AM (0) or PM (1). In the 24 hour mode bit 5 combines with Bit 4 to represent the 10 hour value.
** Bit 6 of Register 9 must be set to a 1. If set to a 0, an unused square wave will be generated in the circuit.

Registers 0 through A are Clock, Calendar, Time of Day Registers.
Register B is the Command Register.
Registers C and D are Watchdog Alarm Registers.

The Watchdog Timer contains 14 registers which are 8-bits wide. These registers contain all of the Time of Day, Alarm, Watchdog, Control, and Data information. The Clock Calendar, Alarm, and Watchdog Registers have both external (user accessible) and internal memory locations containing copies of the data. The external memory locations are independent of the internal functions except that they are updated periodically by the transfer of the incremented internal values. Registers 0, 1, 2, 4, 6, 8, 9, and A contain Time of Day and Data information in Binary Coded Decimal (BCD). Registers 3, 5, and 7 contain the Time of Day Alarm information in BCD. The Command Register (Register B) contains data in binary. The Watchdog Alarm Registers are Registers C and D and information stored in these registers is in BCD.

Time of Day Registers

Registers 0, 1, 2, 4, 6, 8, 9, and A contain Time of Day data in BCD.

Register 0 contains two Time of Day values. Bits 3 - 0 contain the 0.01 Seconds value with a range of 0 to 9 in BCD, Bits 7 - 4 contain the 0.1 Seconds value with a range of 0 to 9 in BCD. This Register has a total range of 0.00 to 0.99 Seconds.

Register 1 contains two Time of Day values. Bits 3 - 0 contain the 1 Seconds value with a range of 0 to 9 in BCD, Bits 7 - 4 contain the 10 Seconds value with a range of 0 to 5 in BCD. This Register has a total range of 0.0 to 59.0 Seconds. Bit 7 of this register will always be zero, regardless of what value is written to it.

Register 2 contains two Time of Day values. Bits 3 - 0 contain the 1 Minute value with a range of 0 to 9 in BCD, Bits 7 - 4 contain the 10 Minutes value with a range of 0 to 6 in BCD. This Register has a total range of 0 to 59 Minutes. Bit 7 of this register will always be zero regardless of what value is written to it.

Register 4 contains the Hours value of the Time of Day. The Hours can be represented in either 12 or 24 hour format depending on the state of Bit 6. When Bit 6 is set to a one (1,) the format is 12 hour. When Bit 6 is set to a zero (0), the format is 24 hour. For the 12 hour format, Bits 3 - 0 contain the 1 Hour value with a range of 0 to 9 in BCD, Bit 4 contains the 10 Hour value with a range of 0 to 1. In the 12 hour format, Bit 5 is used as the AM / PM bit. When AM, Bit 5 is a zero (0) and when PM, bit 5 is a one (1). The total range of this register in the 12 hour format is 01 AM to 12 AM and 01 PM to 12 PM.

When Register 4 is in 24 hour format (Bit 6 is set to a zero (0)), Bits 3 - 0 contain the 1 Hour value with a range of 0 to 9 in BCD, Bit 5 combines with 4 to represent the 10 Hour value. The 10 Hour range is from 0 to 2. The total range of register 4 in the 24 hour format is 00 to 23 hours. Bit 7 of register 4 will always be zero, regardless of what value is written to it and regardless of format (12 or 24 hour).

Register 6 contains the Days value of the Time of Day. Bits 2 - 0 contain the Days value with a range of 1 to 7 in BCD.

Register 8 contains two Time of Day values. Bits 3 - 0 contain the Date value with a range of 0 to 9 in BCD, Bits 5 - 4 contain the 10 Date value with a range of 0 to 3. This Register has a total range of 01 to 31. Bits 7 - 6 of this register will always be zero regardless of what value is written to it.

Register 9 contains two Time of Day values. Bits 3 - 0 contain the Months value with a range of 0 to 9 in BCD while Bits 4 contain the 10 Date value with a range of 0 to 1. This Register has a total range of 01 to 12. Bit 5 will always be zero regardless of what value is written to it. Bit 6 is unused but must be set to a 1. Bit 7, $\overline{\mathrm{Eosc}}$, is the clock oscillator enable bit. When this bit is set to a zero (0) the oscillator is internally enabled. When set to a one (1), the oscillator is internally disabled. The oscillator via this bit is usually turned on once during system initialization but can be toggled on and off at the users discretion.

There are two techniques for reading the Time of Day from the Watchdog Timer. The first is to halt the External Time of Day Registers from tracking the Internal Time of Day Registers by setting the Te bit (Bit 7 of the Command Register) to a logic zero (0) then reading the contents of the Time of Day Registers. Using this technique eliminates the chance of the Time of Day changing while the read is taking place. At the end of the read, the Te bit is set to a logic one (1) allowing the external Time of Day Registers to resume tracking the internal Time of Day Registers. No time is lost as the internal Time of Day Registers continue to keep time while the external Time of Day Registers are halted. This is the recommended method.

The second technique for reading the Time of Day from the Watchdog Timer is to read the external Time of Day Registers without halting the tracking of the Internal Registers. This is not recommended as the registers may be updated while the reading is taking place, resulting in erroneous data being read.

Time of Day Alarm Registers

Registers 3, 5, and 7 are the Time of Day Alarm Registers and are formatted similar to register 2, 4, and 6 respectively. Bit 7 of registers 3, 5, and 7 is a mask bit. The mask bits, when active (logic one (1)), disable the use of the particular Time of Day Alarm Register in the determination of the Time of Day Alarm (see Table 4-13). When all the mask bits are low (0), an alarm will occur when registers 2, 4, and 6 match the values found in registers 3, 5, and 7. When register 7's mask bit is set to a logic one (1), register 6 will be disregarded in the determination of the Time of Day alarm and an alarm will occur everyday. When registers 7 and 5's mask bit is set to a logic one (1), register 6 and 4 will be disregarded in the determination of the Time of Day alarm and an alarm will occur every hour. When registers 7, 5 and 3's mask bit is set to a logic one (1), register 6, 4, and 2 will be disregarded in the determination of the Time of Day alarm and an alarm will occur every minute (when register 1's seconds step from 59 to 00).

Table 4-13 Time of Day Alarm Registers

Register			Comment
Minutes	Hours	Days	
1	1	1	Alarm once per minute
0	1	1	Alarm when minutes match
0	0	1	Alarm when hours and minutes match
0	0	0	Alarm when hours, minutes, and days match

The Time of Day Alarm registers are read and written to in the same format as the Time of Day registers. The Time of Day alarm flag and interrupt are cleared when the alarm registers are read or written.

Watchdog Alarm Registers

Register C contains two Watchdog alarm values. Bits 3 - 0 contain the 0.01 Seconds value with a range of 0 to 9 in BCD, Bits 7 - 4 contain the 0.1 Seconds value with a range of 0 to 9 in BCD. This Register has a total range of 0.00 to 0.99 Seconds.

Register D contains two Watchdog Alarm values. Bits 3 - 0 contain the 1 Second value with a range of 0 to 9 in BCD, Bits 7 - 4 contain the 10 Seconds value with a range of 0 to 9 in BCD. This Register has a total range of 00.0 to 99.0 Seconds.

The Watchdog Alarm Registers can be read or written in any order. When a new value is entered or the Watchdog registers are read, the Watchdog Timer will start counting down from the entered value. When zero is reached the Watchdog Interrupt Output will go active. If jumper E7 is loaded, the CPU will reset to a known state. Refer to Figure 4-7 on page 69. The Watchdog Timer count is reinitialized back to the entered value, the Watchdog flag bit is cleared, and the Watchdog interrupt output is cleared every time either of the registers are accessed. Periodic accesses to the Watchdog Timer will prevent the Watchdog Alarm from occurring. If access does not occur, the alarm will be repetitive. The Watchdog Alarm Register always reads the entered value. The actual countdown value is internal and not accessible to the user. Writing zero’s to Registers C and D will disable the Watchdog Alarm feature.

Command Register

Register B is the Command Register. Within this register are mask bits, control bits, and flag bits. The following paragraphs describe each bit:

Te - Bit 7 Transfer Enable - This bit enables and disables the tracking of data between the internal and external registers. When set to a logic zero (0), tracking is disabled and the data in the external register is frozen. When set to a logic one (1), tracking is enabled. This bit must be set to a logic one (1) to allow the external register to be updated.

Ipsw - Bit 6 Interrupt Switch - This bit toggles the Interrupt Output between the Time of Day Alarm and the Watchdog Alarm. When set to a logic zero (0), the Interrupt Output is from the Watchdog Alarm. When set to a logic one (1), the Interrupt Output is from the Time of Day Alarm.

Ibh/lo - Bit 5 Reserved - This bit should be set to a logic low (0).

Pu/lvl - Bit 4 Interrupt Pulse Mode or Level Mode - This bit determines whether the Interrupt Output will output as a pulse or a level. When set to a logic zero (0), Interrupt Output will be a level. When set to a logic one (1), Interrupt Output will be a pulse. In pulse mode the Interrupt Output will sink current for a minimum of 3 ms. This bit should be set to a logic one (1).

Wam - Bit 3 Watchdog Alarm Mask - Enables / Disables the Watchdog Alarm to Interrupt Output when Ipsw (Bit 6, Interrupt Switch) is set to logic zero (0). When set to a logic zero (0), Watchdog Alarm Interrupt Output will be enabled. When set to a logic one (1), Watchdog Alarm Interrupt Output will be disabled.

Tdm - Bit 2 Time of Day Alarm Mask - Enables/Disables the Time of Day Alarm to Interrupt Output when Ipsw (see Bit 6, Interrupt Switch) is set to logic one (1). When set to a logic zero (0), Time of Day Alarm Interrupt Output will be enabled. When set to a logic one (1), Time of Day Alarm Interrupt Output will be disabled.

Waf - Bit 1 Watchdog Alarm Flag - This is a read-only bit set to a logic one (1) when a Watchdog Alarm Interrupt occurs. This bit is reset when any of the Watchdog Alarm registers are accessed. When the Interrupt Output is set to Pulse Mode (see Bit 4, Interrupt Pulse Mode or Level Mode), the flag will be set to a logic one (1) only when the Interrupt Output is active.

Tdf - Bit 0 Time of Day Alarm Flag - This is a read-only bit set to a logic one (1) when a Time of Day Alarm Interrupt occurs. This bit is reset when any of the Time of Day Alarm registers are accessed. When the Interrupt Output is set to Pulse Mode (see Bit 4, Interrupt Pulse Mode or Level Mode), the flag will be set to a logic one (1) only when the Interrupt Output is active.

Battery Backed SRAM

The VMICPCI-7697 includes 32 Kbyte of battery-backed SRAM addressed at $D8000 to $DFFFF (the lower 14 bytes, $D8000 to $D800D are dedicated to Watchdog Timer registers and are unavailable for general use. See the Watchdog Timer section). The battery-backed SRAM can be accessed by the CPU at anytime, and can be used to store system data that must not be lost during power-off conditions.

NOTE: If LanWorks boots from the Ethernet BIOS, and Msystem’s DiskOnChip BIOS is enabled, then the battery backed SRAM will automatically be disabled.

CHAPTER 5

Maintenance

If a VMIC product malfunctions, please verify the following:

1. Software resident on the product
2. System configuration
3. Electrical connections
4. Jumper or configuration options
5. Boards are fully inserted into their proper connector location
6. Connector pins are clean and free from contamination
7. No components or adjacent boards were disturbed when inserting or removing the board from the chassis
8. Quality of cables and I/O connections

If products must be returned, contact VMIC for a Return Material Authorization (RMA) Number. **This RMA Number must be obtained prior to any return**.

VMIC Customer Service is available at: 1-800-240-7782.
Or E-mail us at customer.service@vmic.com

Maintenance Prints

User level repairs are not recommended. The drawings and diagrams in this manual are for reference purposes only.

Appendix

A

Connector Pinouts

Contents

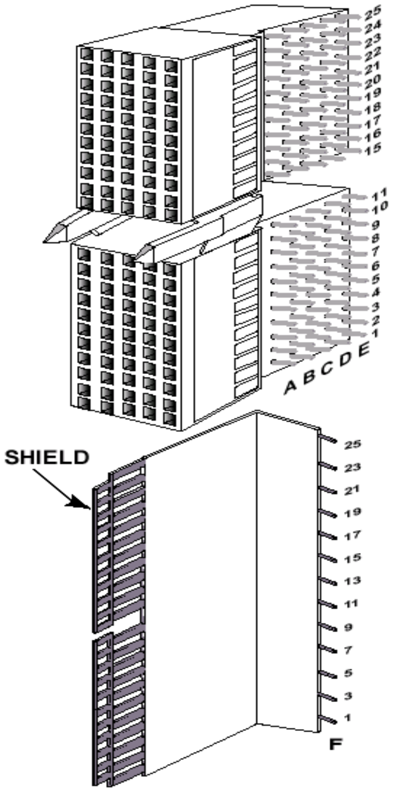
Introduction

The VMICPCI-7697 PC/AT-Compatible CompactPCI Controller has several connectors for its I/O ports. Wherever possible, the VMICPCI-7697 uses connectors and pinouts typical for any desktop PC. This ensures maximum compatibility with a varity of systems.

Connector diagrams in this appendix are generally shown in a natural orientation with the controller board mounted in a CPCI bus chassis.

J1 Connector Pinout

The VMICPCI-7697 utilitizes a high-density 110-pin, low inductance, and controlled impedance connector. This connector meets the IEC-1076 international standard for CompactPCI connectors. An additional external metal shield is required. The large number of ground pins ensures adequate shielding and grounding for low ground bounce and reliable operation in noisy environments. The key prevents mis-alignment of the board when installing in the chassis. Figure A-1 below depicts the J1 connector and the connector pinout.

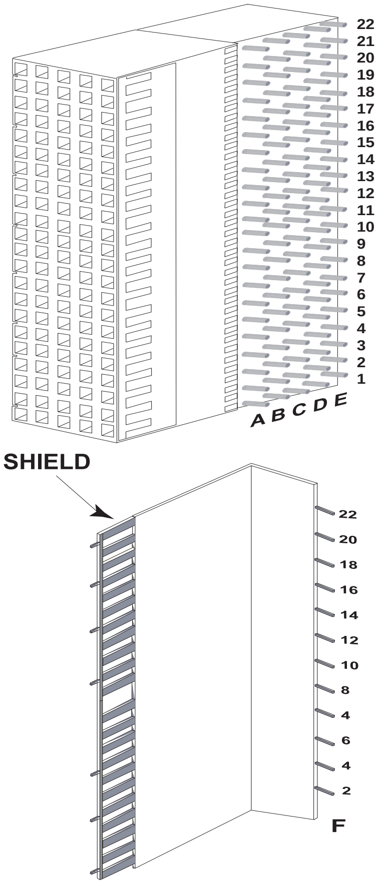


Pin No.	Row A	Row B	Row C	Row D	Row E	Row F
25	+5 V	N/C	C_ENUM#	CPCI_3.3	+5 V	GND
24	C_AD[1]	+5 V	CPCI_VIO	C_AD[0]	N/C	N/C
23	CPCI_3.3	C_AD[4]	C_AD[3]	+5 V	C_AD[2]	GND
22	C_AD[7]	GND	CPCI_3.3	C_AD[6]	C_AD[5]	N/C
21	CPCI_3.3	C_AD[9]	C_AD[8]	GND	C_C/BE0#	GND
20	C_AD[12]	GND	CPCI_VIO	C_AD[11]	C_AD[10]	N/C
19	CPCI_3.3	C_AD[15]	C_AD[14]	GND	C_AD[13]	GND
18	C_SERR#	GND	CPCI_3.3	C_PAR	C_C/BE1#	N/C
17	CPCI_3.3	N/C	N/C	GND	C_PERR#	GND
16	C_DEVSEL#	GND	CPCI_VIO	C_STOP#	C_LOCK#	N/C
15	CPCI_3.3	C_FRAME	C_IRDY#	C_BD_SEL#	C_TRDY#	GND
12 through 14 are lost to the keying area						
11	C_AD[18]	C_AD[17]	C_AD[16]	GND	C_C/BE2#	GND
10	C_AD[21]	GND	CPCI_3.3	C_AD[20]	C_AD[19]	N/C
9	C_CBE3#	C_IDSEL	C_AD[23]	GND	C_AD[22]	GND
8	C_AD[26]	GND	CPCI_VIO	C_AD[25]	C_AD[24]	N/C
7	C_AD[30]	C_AD[29]	C_AD[28]	GND	C_AD[27]	GND
6	C_REQ#	GND	CPCI_3.3	C_CLK	C_AD[31]	N/C
5	N/C	N/C	C_RST#	GND	C_GNT#	GND
4	N/C	C_HEALTHY#	CPCI_VIO	N/C	N/C	N/C
3	C_INTA#	C_INTB#	C_INTC#	+5 V	C_INTD#	GND
2	C_TCK	+5 V	C_TMS	N/C	C_TDI	N/C
1	+5 V	-12 V	C_TRST#	+12 V	+5 V	GND

Figure A-1 J1 Connector and Pinout

J2 Connector and Pinout

The VMICPCI-7697 J2 connector is a 2mm “Hard Metric” CompactPCI connector, with 5 rows, of 22 pins each. J2 is required for system slot CPUs. An additional external metal shield, labeled row F, is also used. This connector’s controlled impedance minimizes unwanted signal reflections. Figure A-2 illustrates the J2 connector and the connector pinout.



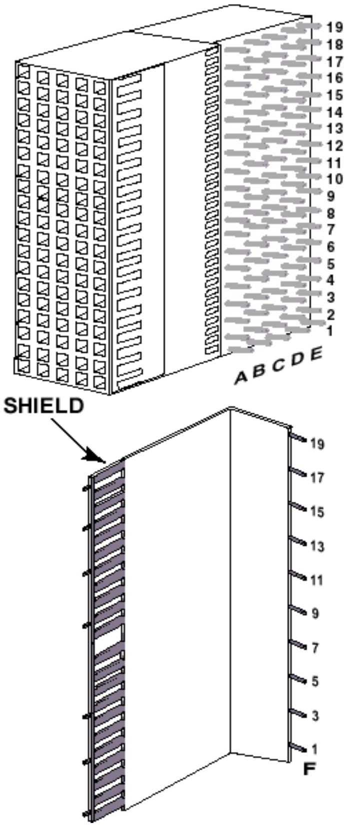
Pin No.	Row A	Row B	Row C	Row D	Row E	Row F
22	N/C	N/C	N/C	N/C	N/C	GND
21	N/C	GND	N/C	N/C	N/C	N/C
20	N/C	GND	N/C	GND	N/C	GND
19	GND	GND	N/C	N/C	N/C	N/C
18	N/C	N/C	N/C	GND	N/C	GND
17	N/C	GND	N/C	N/C	N/C	N/C
16	N/C	N/C	N/C	GND	N/C	GND
15	N/C	GND	N/C	N/C	N/C	N/C
14	N/C	N/C	N/C	GND	N/C	GND
13	N/C	GND	CPCI_VIO	N/C	N/C	N/C
12	N/C	N/C	N/C	GND	N/C	GND
11	N/C	GND	CPCI_VIO	N/C	N/C	N/C
10	N/C	N/C	N/C	GND	N/C	GND
9	N/C	GND	CPCI_VIO	N/C	N/C	N/C
8	N/C	N/C	N/C	GND	N/C	GND
7	N/C	GND	CPCI_VIO	N/C	N/C	N/C
6	N/C	N/C	N/C	GND	N/C	GND
5	N/C	GND	CPCI_VIO	N/C	N/C	N/C
4	CPCI_VIO	N/C	N/C	GND	N/C	GND
3	N/C	GND	N/C	N/C	N/C	N/C
2	N/C	N/C	GND	N/C	N/C	GND
1	N/C	GND	N/C	N/C	N/C	N/C

* CPCI_VIO - The VMICPCI-7697 is a universal VIO design.
* The VMICPCI-7697 supports a 32-bit CPCI bus.
* J2 Row C pins 15 and 16 are not supported.

Figure A-2 J2 Connector and Pinout

J3 Connector Pinout

The J3 connector is a 5 row, 19 pins each, 2mm “Hard Metric” CompactPCI connector. An additional external metal shield is also used, labeled row F. Figure A-3 illustrates the J3 connector and the connector pinout. This connector is also used to route the serial, floppy, and hard drive signals to the backplane I/O.



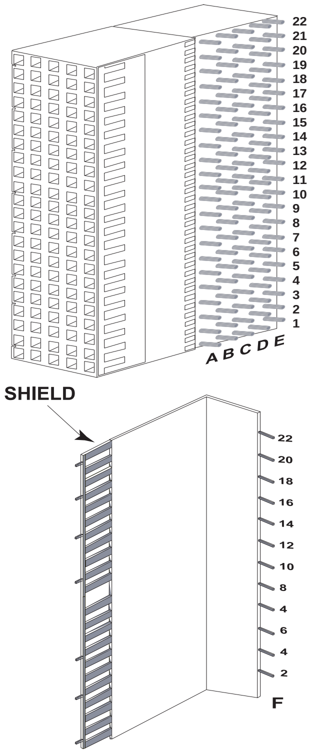
Pin No.	Row A	Row B	Row C	Row D	Row E	Row F
19	GND	ACK#	BUSY	PE	SLCT	GND
18	PD4	PD5	PD6	PD7	GND	N/C
17	PD1	INIT#	PD2	SLIN#	PD3	GND
16	N/C	STB#	AFD#	PD0	ERR#	N/C
15	USBP1+RTC	USBP1-RTC	USB0C1#	DATA	CLK	GND
14	GND	SP0_R1#B	SP0_DTR#B	SP0_CTS#B	SP0_TXB	N/C
13	SP0_RTS	SP0_RXB	SP0_DSR#B	SP0_DCD#B	N/C	GND
12	SP1_RTS	BOOT_LED	+5 V	N/C	+12 V	N/C
11	GND	SP1_R1#B	SP1_DTR#B	SP1_CTS#B	SP1_TXB	GND
10	DAP0	SP1_RXB	SP1_DSR#B	SP1_DCD#	DAP2PR	N/C
9	DAP1	REDWC#	IDEIORDY	IDESELA	IDEIOW0#	GND
8	DIR#	INDEX#	IDECS01#	DOP[8]	DOP[6]	N/C
7	SIDE1#	IDEIOR0#	HD_ACTA#	DOP[9]	DOP[7]	GND
6	RDATA#	TRK0#	IDECS03#	DOP[10]	DOP[2]	N/C
5	DRATE0#	WDATA#	IDEIRQ14	DOP[11]	DOP[3]	GND
4	DRVSB#	STEP#	IDE0ACK0#	DOP[12]	DOP[4]	N/C
3	DSKCHG#	MOTEB#	+5 V	DOP[13]	DOP[5]	GND
2	WPT#	DRVSA#	IDEDRQ0	DOP[14]	DOP[0]	N/C
1	WGATE#	MOTEA#	IDERST#	DOP[15]	DOP[1]	GND

NOTE: Backplane designs should route P3 signals straight through to rear J3. The VMIACC-0576 board can then be utilized.

Figure A-3 J3 Connector and Pinout

J5 Connector and Pinout

The VMICPCI-7697 J5 connector is a 2mm “Hard Metric” CompactPCI connector, with 5 rows of 22 pins each. An additional external metal shield is also used, labeled row F. This connector’s controlled impedance minimizes unwanted signal reflections. Figure A-4 illustrates the J5 connector and the connector pinout.



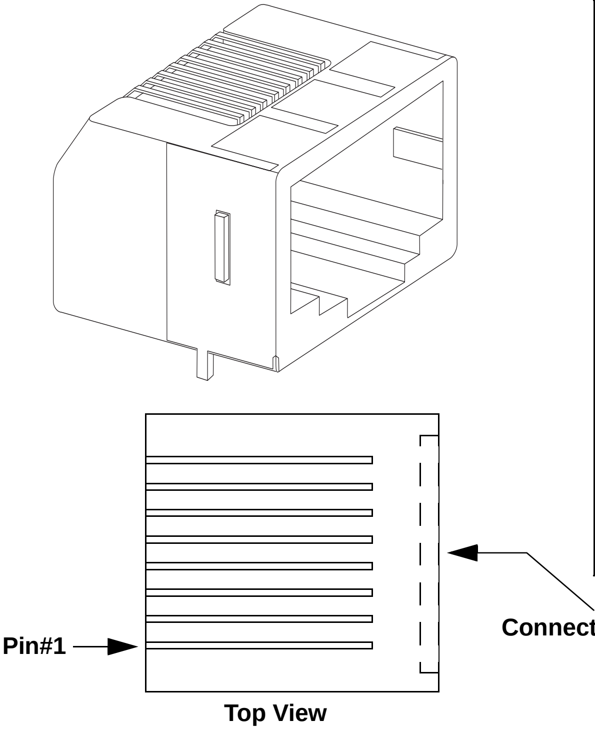
Pin No.	Row A	Row B	Row C	Row D	Row E	Row F
22	PMC_A22_5	PMC_B22_4	PMC_C22_3	PMC_D22_2	PMC_E22_1	GND
21	PMC_A21_10	PMC_B21_9	PMC_C21_8	PMC_D21_7	PMC_E21_6	N/C
20	PMC_A20_15	PMC_B20_14	PMC_C20_13	PMC_D20_12	PMC_E20_11	GND
19	PMC_A19_20	PMC_B19_19	PMC_C19_18	PMC_D19_17	PMC_E19_16	N/C
18	PMC_A18_25	PMC_B18_24	PMC_C18_23	PMC_D18_22	PMC_E18_21	GND
17	PMC_A17_30	PMC_B17_29	PMC_C17_28	PMC_D17_27	PMC_E17_26	N/C
16	PMC_A16_35	PMC_B16_34	PMC_C16_33	PMC_D16_32	PMC_E16_31	GND
15	PMC_A15_40	PMC_B15_39	PMC_C15_38	PMC_D15_37	PMC_E15_36	N/C
14	PMC_A14_45	PMC_B14_44	PMC_C14_43	PMC_D14_42	PMC_E14_41	GND
13	PMC_A13_50	PMC_B13_49	PMC_C13_48	PMC_D13_47	PMC_E13_46	N/C
12	PMC_A12_55	PMC_B12_54	PMC_C12_53	PMC_D12_52	PMC_E12_51	GND
11	PMC_A11_60*	PMC_B11_59	PMC_C11_58	PMC_D11_57	PMC_E11_56	N/C
10	VIO	PMC_B10_64**	PMC_C10_63**	PMC_D10_62*	PMC_E10_61*	GND
9	N/C	N/C	N/C	N/C	N/C	N/C
8	N/C	N/C	N/C	N/C	N/C	GND
7	N/C	N/C	N/C	N/C	N/C	N/C
6	N/C	N/C	N/C	N/C	N/C	GND
5	N/C	N/C	N/C	N/C	N/C	N/C
4	N/C	N/C	N/C	N/C	N/C	GND
3	N/C	N/C	N/C	N/C	N/C	N/C
2	N/C	N/C	N/C	N/C	N/C	GND
1	N/C	N/C	N/C	N/C	N/C	N/C

Notes: * signal can be tied to GND through Jumper E22.
** Signal can be tied to +5 V through Jumper E23.

Figure A-4 J5 Connector and Pinout

Ethernet Connector Pinout (J12)

The pinout diagram for the Ethernet 10BaseT and 100BaseTx connector is shown in Figure A-5.

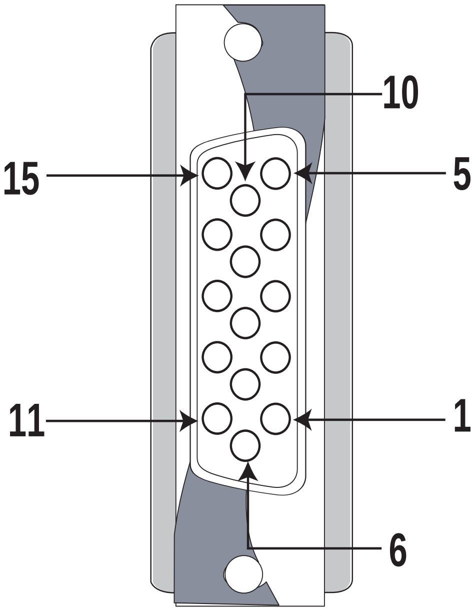


ETHERNET CONNECTOR (10BaseT, 100BaseTx)		
PIN	Signal Name	
1	TD+	Transmit Data
2	TD-	Transmit Data
3	RD+	Receive Data
4	TX_CT_OUT	Transmit Center Tap Out
5	TX_CT_OUT	Transmit Center Tap Out
6	RD-	Receive Data
7	RX_CT_OUT	Receive Center Tap Out
8	RX_CT_OUT	Receive Center Tap Out

Figure A-5 Ethernet Connector and Pinout

Video Connector Pinout (J11)

The video port uses a standard high-density DB15 SVGA connector. Figure A-6 illustrates the connector and pinout.

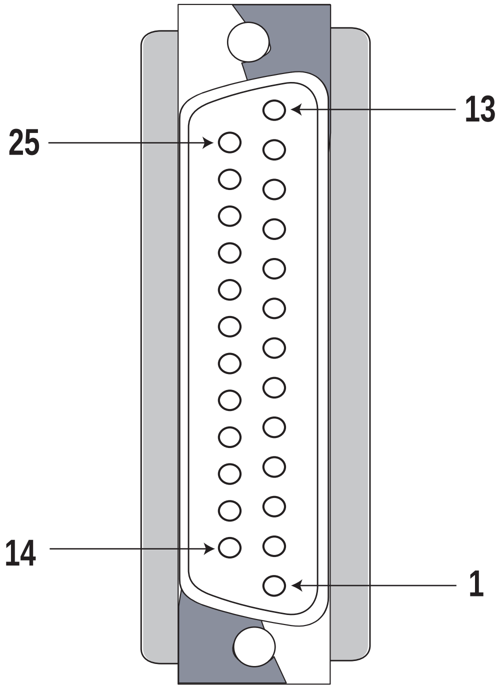


Video Connector		
Pin	**Direction**	**Function**
1	Out	Red
2	Out	Green
3	Out	Blue
4		Reserved
5		Ground
6		Ground
7		Ground
8		Ground
9		5 V
10		Ground
11		Reserved
12		I2 Data
13	Out	Horizontal SYNC
14	Out	Vertical SYNC
15		I2 Clock
Shield		Chassis Ground

Figure A-6 Video Connector and Pinout

Parallel Port Connector Pinout (P2)

The parallel port shown in Figure A-7 uses a Microminiature DB25 female connector typical of any PC / AT system. An adapter to connect standard DB25 parallel port products to the Microminiature connector is available. Please refer to the “VMICPCI-7697 Product Specification” for ordering information.

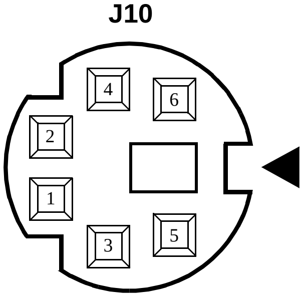


Parallel Port Connector		
Pin	**Direction**	**Function**
1	In / Out	Data Strobe
2	In / Out	Bidirectional Data D0
3	In / Out	Bidirectional Data D1
4	In / Out	Bidirectional Data D2
5	In / Out	Bidirectional Data D3
6	In / Out	Bidirectional Data D4
7	In / Out	Bidirectional Data D5
8	In / Out	Bidirectional Data D6
9	In / Out	Bidirectional Data D7
10	In	Acknowledge
11	In	Device Busy
12	In	Out of Paper
13	In	Device Selected
14	Out	Auto Feed
15	In	Error
16	Out	Initialize Device
17	In	Device Ready for Input
18		Signal Ground
19		Signal Ground
20		Signal Ground
21		Signal Ground
22		Signal Ground
23		Signal Ground
24		Signal Ground
25		Signal Ground
Shield		Chassis Ground

Figure A-7 P2 Connector and Pinout

Keyboard and Mouse Connectors and Pinout (J10)

The keyboard and mouse connectors are standard 6-pin female mini-DIN PS/2 connectors as shown in Figure A-8 and Table A-1.



Keyboard/Mouse Connector*		
Pin	**Dir**	**Function**
1	In/Out	Mouse Data
2	In/Out	Keyboard Data
3		Ground
4		+5 V
5	Out	Mouse Clock
6	Out	Keyboard Clock
Shield		Chassis Ground
An adapter cable is included with the VMICPCI-7697 to separate the keyboard and mouse connector.		

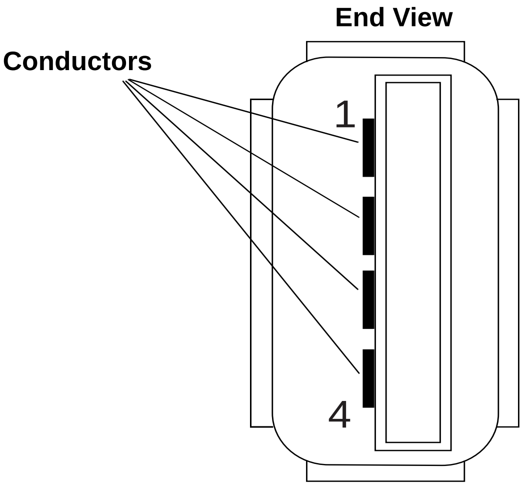
Figure A-8 Keyboard/Mouse Connector and Pinout

Table A-1 Keyboard/Mouse Y Splitter Cable

Keyboard			Mouse		
Pin	**Dir**	**Function**	**Pin**	**Dir**	**Function**
1	In/Out	Keyboard Data	1	In/Out	Mouse Data
2		Unused	2		Unused
3		Ground	3		Ground
4		+5 V	4		+5 V
5	Out	Keyboard Clock	5	Out	Mouse Clock
6		Unused	6		Unused
Shield		Chassis Ground	Shield		Chassis Ground

USB Connector (J9)

The Universal Serial Bus (USB) port uses an industry standard four position shielded connector. Figure A-9 shows the pinout of the USB connector.



Pin	Signal	Function
1	USBV	USB Power
2	USB-	USB Data -
3	USB+	USB Data +
4	USBG	USB Ground

Figure A-9 USB Connector Pinout

PMC Connector Pinout

PMC J7 Connector and Pinout

The PCI Mezzanine Card (PMC) carries the same signals as the PCI standard; however, the PMC standard uses a completely different form factor. Table A-2 through A-4 are the pinouts for the PMC connectors (J6, J7, and J4).

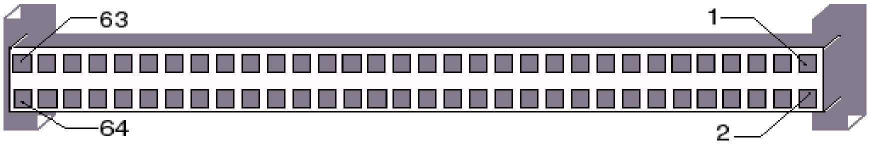


Figure A-10 PMC J7 Connector

Table A-2 PMC J7 Connector Pinout

PMC Connector (J7)				PMC Connector (J7)			
Left Side		Right Side		Left Side		Right Side	
Pin	Name	Pin	Name	Pin	Name	Pin	Name
1	GND	2	-12	33	FRAME#	34	GND
3	GND	4	INTC#	35	GND	36	IRDY#
5	INTD#	6	INTA#	37	DEVSEL#	38	+5 V
7	BMODE1A	8	+5 V	39	GND	40	LOCK#
9	INTB#	10	NC	41	SDONE#	42	NC
11	GND	12	NC	43	PAR	44	GND
13	CLK	14	GND	45	+5 V	46	AD[15]
15	GND	16	GNT#	47	AD[12]	48	AD[11]
17	REQ#	18	+5 V	49	AD[9]	50	+5 V
19	+5 V	20	AD[31]	51	GND	52	C/BE0#
21	AD[28]	22	AD[27]	53	AD[6]	54	AD[5]
23	AD[25]	24	GND	55	AD[4]	56	GND
25	GND	26	C/BE3#	57	+5 V	58	AD[3]
27	AD[22]	28	AD[21]	59	AD[2]	60	AD[1]
29	AD[19]	30	+5 V	61	AD[0]	62	+5 V
31	+5 V	32	AD[17]	63	GND	64	REQ64#

PMC J6 Connector and Pinout

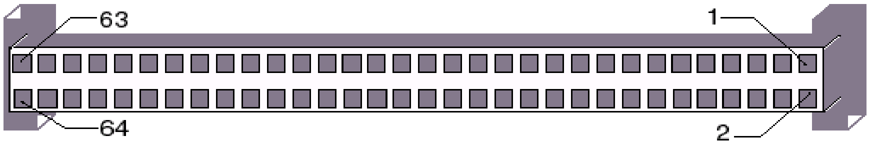


Figure A-11 PMC J6 Connector

Table A-3 PMC J6 Connector Pinout

PMC Connector (J6)				PMC Connector (J6)			
Left Side		Right Side		Left Side		Right Side	
Pin	Name	Pin	Name	Pin	Name	Pin	Name
1	+12 V	2	+5 V	33	GND	34	NC
3	GND	4	NC	35	TRDY	36	+3.3 V
5	+5 V	6	GND	37	GND	38	STOP#
7	GND	8	NC	39	PERR#	40	GND
9	NC	10	NC	41	+3.3 V	42	SERR#
11	PRSNT2	12	+3.3 V	43	C/BE1#	44	GND
13	RST#	14	GND	45	AD[14]	46	AD[13]
15	+3.3 V	16	GND	47	GND	48	AD[10]
17	NC	18	GND	49	AD[8]	50	+3.3 V
19	AD[30]	20	AD[29]	51	AD[7]	52	NC
21	GND	22	AD[26]	53	+3.3 V	54	NC
23	AD[24]	24	+3.3 V	55	NC	56	GND
25	IDSEL	26	AD[23]	57	NC	58	NC
27	+3.3 V	28	AD[20]	59	GND	60	NC
29	AD[18]	30	GND	61	ACK64#	62	+3.3 V
31	AD[16]	32	C/BE2#	63	GND	64	NC

PMC J4 Connector and Pinout

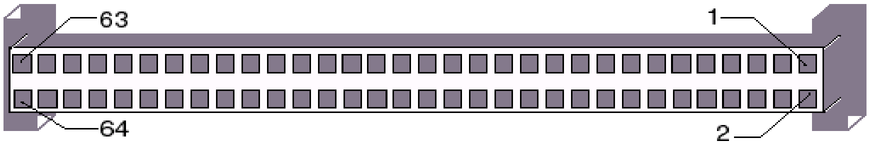


Figure A-12 PMC J4 Connector

Table A-4 PMC J4 Connector Pinout

PMC Connector (J4)				PMC Connector (J4)			
Left Side		Right Side		Left Side		Right Side	
Pin	**Name**	**Pin**	**Name**	**Pin**	**Name**	**Pin**	**Name**
1	J5_PMC_E22_1	2	J5_PMC_D22_2	33	J5_PMC_C16_33	34	J5_PMC_B16_34
3	J5_PMC-C22_3	4	J5_PMC_B22_4	35	J5_PMC_A16_35	36	J5_PMC_E15_36
5	J5_PMC_A22_5	6	J5_PMC_E21_6	37	J5_PMC_D15_37	38	J5_PMC_C15_38
7	J5_PMC_D21_7	8	J5_PMC_C21_8	39	J5_PMC_B15_39	40	J5_PMC_A15_40
9	J5_PMC_B21_9	10	J5_PMC_A21_10	41	J5_PMC_E14_41	42	J5_PMC_D14_42
11	J5_PMC_E20_11	12	J5_PMC_D20_12	43	J5_PMC_C14_43	44	J5_PMC_B14_44
13	J5_PMC_C20_13	14	J5_PMC_B20_14	45	J5_PMC_A14_45	46	J5_PMC_E13_46
15	J5_PMC_A20_15	16	J5_PMC_E19_16	47	J5_PMC_D13_47	48	J5_PMC_C13_48
17	J5_PMC_D19_17	18	J5_PMC_C19_18	49	J5_PMC_B13_49	50	J5_PMC_A13_50
19	J5_PMC_B19_19	20	J5_PMC_A19_20	51	J5_PMC_E12_51	52	J5_PMC_D12_52
21	J5_PMC_E18_21	22	J5_PMC_D18_22	53	J5_PMC_C12_53	54	J5_PMC_B12_54
23	J5_PMC_C18_23	24	J5_PMC_B18_24	55	J5_PMC_A12_55	56	J5_PMC_E11_56
25	J5_PMC_A18_25	26	J5_PMC_E17_26	57	J5_PMC_D11_57	58	J5_PMC_C11_58
27	J5_PMC_D17_27	28	J5_PMC_C17_28	59	J5_PMC_B11_59	60	J5_PMC_A11_60*
29	J5_PMC_B17_29	30	J5_PMC_A17_30	61	J5_PMC_E10_61*	62	J5_PMC_D10_62*
31	J5_PMC_E16_31	32	J5_PMC_D16_32	63	J5_PMC_C10_63**	64	J5_PMC_B10_64**

Notes: * Signal is tied to GND through zero ohm resistor.
** Signal is tied to +5 V through zero ohm resistor.

Appendix

B

System Driver Software

Contents

Introduction

The VMICPCI-7697 provides high-performance video and Local Area Network (LAN) access by means of on-board PCI and AGP-based adapters and associated software drivers. The APG-based video adapter used on the VMICPCI-7697 is the S3 Trio 3d. High-performance LAN operation including 10BaseT and 100BaseTx, is provided by the DEC 21143 Ethernet Controller chip.

To optimize performance of each of these PCI-based subsystems, the VMICPCI-7697 is provided with software drivers compatible with DOS, Windows for Workgroups Version 3.11, Windows 95, and Windows NT operating systems. The following paragraphs provide instructions for loading and installing the adapter software.

Driver Software Installation

In order to properly use the Video and LAN adapters of the VMICPCI-7697, the user must install the driver software located on the distribution diskettes provided with the unit. Detailed instructions for installation of the drivers during installation of Windows for Workgroups Version 3.11, Windows 95, or Windows NT 4.0 operating systems are described in the following sections.

Windows for Workgroups (Version 3.11)

1. Format the IDE hard drive and install MS-DOS.
2. Begin installation of Windows for Workgroups 3.11, following the instructions provided by Microsoft for Express Setup. When you reach the 'NETWORK SETUP' screen, the statement 'No Network Installed' will be displayed.

NOTE: If you do not require LAN operation, click 'CONTINUE' and skip steps 3 through 12.

3. The following steps will load the Digital Semiconductor Ethernet drivers which are used to configure the Intel 21143 adapter. From the main 'NETWORK SETUP' screen, click on the 'NETWORKS' button.
4. The 'Networks' screen should be displayed. Choose 'Install Microsoft Windows Network' option.
5. Then click 'OK'.
6. The 'NETWORK SETUP' screen appears again with the option for 'SHARING'. Click on 'SHARING' and choose the options that fit your system requirements by placing an 'X' in the boxes shown. Then click 'OK'.
7. Again at the 'NETWORK SETUP' screen, click 'CONTINUE'.
8. Under 'ADD NETWORK ADAPTER' click on 'UNLISTED or UPDATED NETWORK ADAPTER'. Then click 'OK'.
9. Insert the VMICPCI-7697 distribution disk labeled 320-500040-004 into drive A: and type: **`A:\WFW311\`**. Then click 'OK'.
10. Under 'UNLISTED' or 'UPDATED NETWORK ADAPTER', choose 'Intel 21143-based 10/100 mpbs Ethernet Controller (NDIS31)'. Then click 'OK'.
11. Under 'MICROSOFT WINDOWS NETWORK NAMES', enter the network names you want for computer name, group name, etc. Click 'OK'.
12. Windows for Workgroups should now continue with regular installation. During the remaining installation steps, use the full path name, A:\WFW311\, whenever prompted for the Intel 21143 driver diskette.
13. After Windows for Workgroups installation is complete, you may choose to install the S3 video drivers. If you do not require the S3 video drivers for operation, please skip steps 14 through 21.
14. Insert the VMICPCI-7697 distribution diskette labeled 320-500040-001 into drive A:
15. From the Program Manager Screen double-click the mouse on the 'Windows Setup' icon.
16. Under the Windows setup screen select Options. Click on 'Change System

Setup'.

17. Click on the display line. Select 'Other Display (Requires Disk from OEM)'.
18. In the Windows Setup window type **A:** then select 'OK'.
19. Select the correct driver from the list and click 'OK'.
20. When the 'Installation Complete' screen is displayed, click 'OK'.
21. The network must be setup properly. It is necessary to setup the proper connection type for your system.

Proceed with the following steps to set your network connection type.

NOTE: If you are not running a network, please skip steps 22 through 26.

22. From the 'PROGRAM MANAGER' screen, double-click on the 'NETWORK' icon.
23. Then double-click on 'NETWORK SETUP' icon.
24. Under 'NETWORK SETUP', double-click on 'Intel 21143-based 10/100 mpbs Ethernet Controller (ND153)'.
25. Under 'ADVANCED NETWORK-ADAPTER SETTINGS', choose 'CONNECTION TYPE' and choose the 'CONNECTION TYPE' value of '*AUTO SENSE TYPE*'. Click on 'SET' and then 'OK'.
26. Under 'NETWORK SETUP', click on 'OK'. If the network connection type has changed, click 'OK' at the next screen for the information message about 'SYSTEM.INI' and 'PROTOCOL.INI' and click on 'RESTART COMPUTER' for the new network settings to take effect.

The unit should now be configured for operation in the WINDOWS FOR WORKGROUPS 3.11 environment.

Windows 95

1. Format the hard drive with MS-DOS.
2. Begin installation of Windows 95, following the instructions provided by the Windows 95 manual.
3. When you reach the ‘WINDOWS 95 SETUP WIZARD SCREEN’, choose ‘TYPICAL’ under ‘SETUP OPTIONS’ and then click on ‘NEXT’.
4. When you reach the ‘ANALYZING YOUR COMPUTER’ screen, place an ‘X’ in the box for ‘NETWORK ADAPTER’, then click on ‘NEXT’.
5. Under the ‘WINDOWS COMPONENTS SCREEN’, select ‘INSTALL THE MOST COMMON COMPONENTS’ and then click on ‘NEXT’.
6. Continue with the installation until Windows 95 is completely installed and has rebooted.

NOTE: If you do not require LAN operation, skip steps 7 through 25.

7. From the main Windows 95 screen, click on ‘START’.
8. Click on ‘SETTINGS’ and then ‘CONTROL PANEL’.
9. Double-click on the ‘SYSTEM’ icon and select the ‘DEVICE MANAGER’ tab.
10. Double-click on ‘OTHER DEVICES’ and then double-click on ‘PCI ETHERNET CONTROLLER’.
11. Select the ‘DRIVER’ tab and then select ‘CHANGE DRIVER’.
12. In the ‘SELECT HARDWARE TYPE’ screen, choose ‘NETWORK ADAPTERS’ and click on ‘OK’.
13. Insert the Diskette labeled 320-500040-004 into drive A:
14. In the ‘SELECT DEVICE’ window, click on ‘HAVE DISK’ and type **`A:\WIN95\INF`** and then click ‘OK’.
15. Under ‘SELECT DEVICES’ choose ‘Intel 21143-BASED 10/100 MPBS ETHERNET CONTROLLER’, then click ‘OK’.
16. Under ‘PCI ETHERNET CONTROL PROPERTIES’ select ‘OK’. The system will prompt you for computer and workgroup names. Type in the names you wish to use in the spaces shown and then choose ‘CLOSE’.
17. An ‘INSERT DISK’ window will be displayed, click on ‘OK’.
18. Under the ‘COPYING FILES’ window type **`A:\WIN95\`**. Click on ‘OK’.
19. An ‘INSERT DISK’ window will be displayed, click on ‘OK’.
20. Under the ‘COPYING FILES’ window type the directory where Windows 95 is located. For example, if you are loading from a CD-ROM located at D:\, the desired response is D:\WIN95.

21. At the ‘SYSTEM PROPERTIES’ window, click ‘OK’.
22. From ‘CONTROL PANEL’, double-click on the ‘NETWORK’ icon.
23. Under ‘NETWORK’, click on ‘FILE AND PRINT SHARING’ and choose the appropriate items for your system, click ‘OK’.
24. At the ‘NETWORK’ window, click ‘OK’. When prompted, insert the diskettes needed to complete the network installation, if required.
25. When the system prompts you to restart your computer, click on ‘YES’ for the network settings to take effect.
26. From the main Windows 95 screen, click on ‘START’.
27. Next, click on ‘SETTINGS’ and the ‘CONTROL PANEL’.
28. Double-click on the ‘DISPLAY’ icon and select the ‘SETTINGS’ tab.
29. A ‘DISPLAY’ window appears prompting for the use of Hardware Installation Wizard, click on ‘CANCEL’.
30. Click on ‘CHANGE DISLAY TYPE’, then click on ‘CHANGE’ in the ‘ADAPTER TYPE’ field.
31. Select ‘S3’ as the manufacturer, then click ‘HAVE DISK’.
32. Insert the diskette labeled 320-500040-002 into drive A:. Type **A:\95** as the files source (if not already displayed) and click on ‘OK’.
33. Under ‘SELECT DEVICE’, choose ‘S3 Trio 3d’, then click ‘OK’.
34. When the CHOOSE DISPLAY TYPE’ screen returns, click on ‘CLOSE’.
35. When the ‘Display Properties” screen returns, click on ‘CLOSE’.
36. Restart the computer if prompted to allow the new settings to take effect.

The unit should now be properly configured for operation in Windows 95.

Windows 95 INF Update Utility for Intel Chipsets

This update allows the operating system to correctly identify the Intel chipset components and properly configure the system.

1. Windows 95 must be fully installed and running on the system prior to running this software.
2. Close any running applications. You may experience some difficulties if you don’t close all applications.
3. Insert the Diskette labeled 320-500040-003 into drive A: and run A:\INF_UP\SETUP.EXE.
4. Click ‘NEXT’ on the ‘WELCOME SCREEN’ to read the license agreement.
5. Click ‘Yes’ if you agree to continue. If you click ‘NO’, the program will terminate.
6. Click ‘NEXT’ in the ‘INSTALLER SCREEN”.
7. Click ‘OK’ to restart.
8. Follow the screen instructions and use the default settings to complete the setup after Windows 95 is re-started.

Bus Master IDE Driver for Windows 95

1. Ensure the system is operating correctly.
2. Close any running applications.
3. The driver files are stored in an integrated application setup program. This program is a Windows 95 program that allows the driver files to be installed or uninstalled. Insert the Diskette labeled 320-500040-003 into drive A: and run ‘A:\SETUP.EXE’.
4. Select 'NEXT' on ‘WELCOME SCREEN’ to continue.
5. View the 'Software Use and Distribution License Agreement' and choose 'YES' if you agree to continue. If you choose 'NO' the program will terminate.
6. Next, select 'INSTALL' to install the Intel BM-IDE Driver.

NOTE: If the driver is currently installed on the system, SETUP will ask you whether or not you want to continue. Follow the prompts on the screen to install the driver if desired.

7. Select 'OK' to restart the system when prompted to do so.
8. Follow the screen instructions and use default settings to complete the setup when Windows 95 is re-started.

Upon re-start, Windows 95 will display that it has found Intel PCI Bus Master IDE controller hardware and is installing hardware for it.

9. Select 'YES' when prompted to re-start Windows 95.

After installation, the following driver and related files are stored as listed.

- <Windows 95 directory>\SYSTEM\IOSUBSYS\IDEATAPI.MPD
- <Windows 95 directory>\SYSTEM\IOSUBSYS\PIIXVSD.VXD
- <Windows 95 directory>\INF\IDEATAPI.INF

NOTE: This driver is not to be used with Windows 98. The setup program must be rerun to uninstall the driver if Windows 98 is to be used.

Windows NT (Version 4.0)

Windows NT 4.0 includes drivers for the on-board LAN and video adapters. The following steps are required to configure the LAN for operation.

1. Follow the normal Windows NT 4.0 installation until you reach the ‘WINDOWS NT WORKSTATION SETUP’ window which states that ‘WINDOWS NT NEEDS TO KNOW HOW THIS COMPUTER SHOULD PARTICIPATE ON A NETWORK’.
2. Click on the radio button next to ‘THIS COMPUTER WILL PARTICIPATE ON A NETWORK’.
3. Click on the box next to ‘WIRED TO THE NETWORK’ and click on ‘NEXT’.
4. In the next screen, click on the ‘SELECT FROM LIST’ button.
5. Click on the ‘HAVE DISK’ button.
6. Insert disk 320-500040-004 into drive A:.
7. Type **`A:\WNT40\NDIS40`** and click ‘OK’.
8. In the ‘SELECT OEM OPTION’, choose ‘Intel 21143-based 10/100 mpbs Ethernet Controller (NDIS3)’, then click ‘OK’.
9. Select the above entry on the displayed list, click on ‘NEXT’.
10. Select the NetBEUI Protocol (only), click on ‘NEXT’.
11. Click on ‘NEXT’ to install selected components.
12. Click ‘CONTINUE’ to allow an Autosense connection type.
13. Step through the remaining screens, providing the data pertinent to your network.
14. Continue through the setup procedure until the ‘DETECTED DISPLAY’ window appears, click ‘OK’ to continue.
15. In the ‘DISPLAY PROPERTIES’ window, click on ‘TEST’.

NOTE: Please note that Windows NT 4.0 does not allow the selection of the S3 drivers during initial setup.

If the display test is successful, click on ‘OK’ to continue. If the display test is not successful, you may have to adjust the display parameter to find a functional setting, for example a lower resolution or lower number of colors.

16. Continue with the procedure to the ‘WINDOWS NT SETUP’ window. Click on ‘RESTART COMPUTER’.

NOTE: Service PACK 3 must be installed.

17. When the computer reboots, double-click on the ‘MY COMPUTER’ window.
18. Double-click on the ‘CONTROL PANEL’ icon in the ‘MY COMPUTER’ window.
19. Double-click on the ‘DISPLAY’ icon in the ‘CONTROL PANEL’.
20. Select the ‘SETTINGS’ tab in the ‘DISPLAY PROPERTIES’ window, then click on the ‘DISPLAY TYPE’ button.
21. In the ‘DISPLAY TYPE’ window, click on ‘CHANGE’.
22. In the ‘CHANGE DISPLAY’ window, click on ‘HAVE DISK’.
23. Insert disk 320-500040-002 into drive A:
24. Type **`A:\NT_4\32bit\`** and click ‘OK’.
25. ‘S3 Trio 3d’ will be displayed in the ‘CHANGE DISPLAY’ window. Click ‘OK’.
26. Proceed as directed, removing the driver disk from the floppy drive, and restart the computer to activate the new settings. When the system reboots, the ‘INVALID DISPLAY SETTINGS’ screen will be displayed. Click ‘OK’.
27. On the ‘DISPLAY PROPERTIES’ screen click on ‘SETTINGS’, then click on ‘TEST’.
28. The ‘TESTING MODE’ screen will be displayed. Click ‘OK’. If the bitmap test image is displayed correctly, click on ‘OK’.

If M-systems DiskOnChip operation under Windows NT 4.0 is not required, skip steps 29 - 33.

29. Insert diskette 320-500040-006 into drive A:.
30. From the Windows NT desktop double-click on ‘MY COMPUTER’.
31. Click on ‘Drive A:’ and click the right mouse button on the file ‘/NT/Trueffs.inf’.
32. From the ‘MENU’ window click ‘INSTALL’.
33. After Windows NT finishes coping the driver, the computer will ask to reboot, reboot the computer. After reboot, the DiskOnChip will appear as a driver in the Windows NT system

The unit should now be configured for operation under Windows NT 4.0.

Appendix

C

LANWorks BIOS

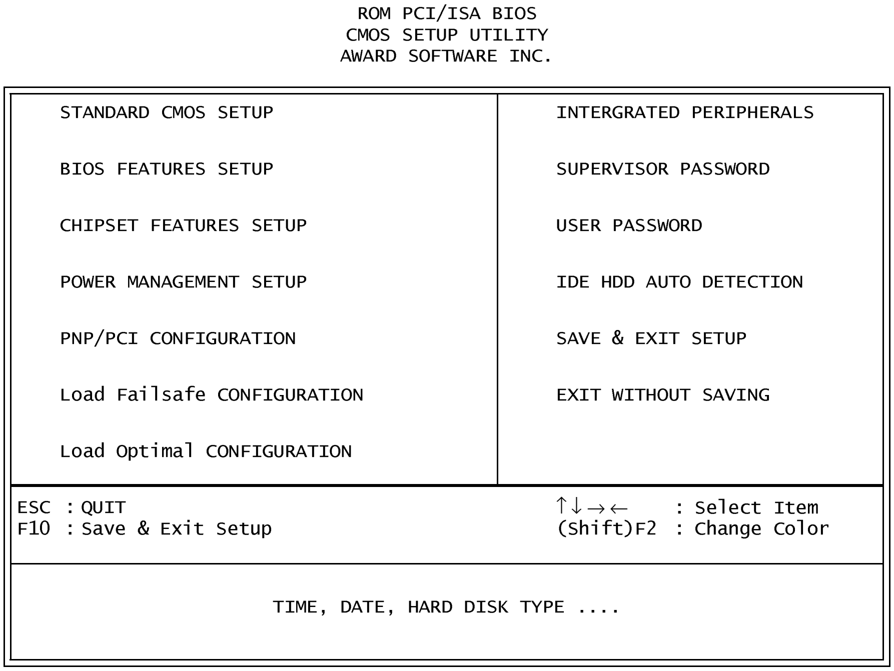
Contents

Introduction

The VMICPCI-7697 includes the LANWorks option which allows the VMICPCI-7697 to be booted from a network. This appendix describes the LANWorks BIOS Setup screen, and the procedures to enable this option.

System BIOS Setup Utility

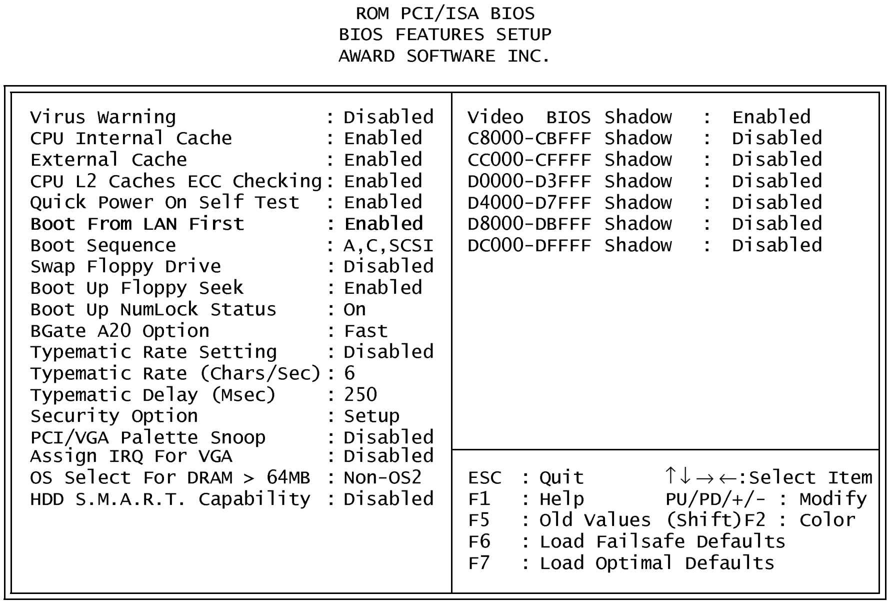
To enable the LANWorks BIOS option reboot the VMICPCI-7697 and when prompted, press the Delete key to access the System BIOS Setup Utility screen shown below.



```
                    ROM PCI/ISA BIOS
                   CMOS SETUP UTILITY
                  AWARD SOFTWARE INC.

    STANDARD CMOS SETUP                 INTERGRATED PERIPHERALS

    BIOS FEATURES SETUP                 SUPERVISOR PASSWORD

    CHIPSET FEATURES SETUP              USER PASSWORD

    POWER MANAGEMENT SETUP              IDE HDD AUTO DETECTION

    PNP/PCI CONFIGURATION               SAVE & EXIT SETUP

    Load Failsafe CONFIGURATION         EXIT WITHOUT SAVING

    Load Optimal CONFIGURATION

ESC  : QUIT                             ↑↓→←        : Select Item
F10  : Save & Exit Setup                (Shift)F2   : Change Color

                  TIME, DATE, HARD DISK TYPE ....
```

Using the arrow keys, highlight *BIOS Features Setup*, and press the Return key. This will display the BIOS Features Setup screen.

BIOS Features Setup



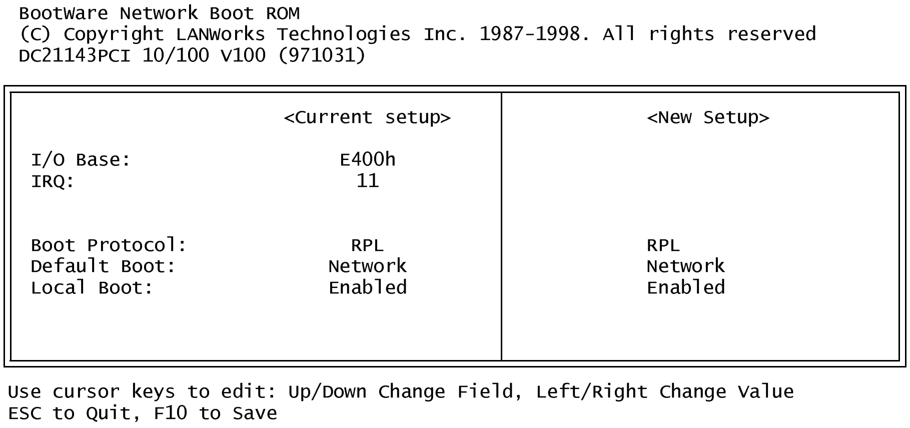
```
                    ROM PCI/ISA BIOS
                   BIOS FEATURES SETUP
                  AWARD SOFTWARE INC.

Virus Warning              : Disabled   Video  BIOS Shadow   :  Enabled
CPU Internal Cache         : Enabled    C8000-CBFFF Shadow   :  Disabled
External Cache             : Enabled    CC000-CFFFF Shadow   :  Disabled
CPU L2 Caches ECC Checking : Enabled    D0000-D3FFF Shadow   :  Disabled
Quick Power On Self Test   : Enabled    D4000-D7FFF Shadow   :  Disabled
Boot From LAN First        : Enabled    D8000-DBFFF Shadow   :  Disabled
Boot Sequence              : A,C,SCSI   DC000-DFFFF Shadow   :  Disabled
Swap Floppy Drive          : Disabled
Boot Up Floppy Seek        : Enabled
Boot Up NumLock Status     : On
BGate A20 Option           : Fast
Typematic Rate Setting     : Disabled
Typematic Rate (Chars/Sec) : 6
Typematic Delay (Msec)     : 250
Security Option            : Setup
PCI/VGA Palette Snoop      : Disabled
Assign IRQ For VGA         : Disabled   ESC  : Quit       ↑↓→←:Select Item
OS Select For DRAM > 64MB  : Non-OS2    F1   : Help       PU/PD/+/- : Modify
HDD S.M.A.R.T. Capability  : Disabled   F5   : Old Values (Shift)F2 : Color
                                        F6   : Load Failsafe Defaults
                                        F7   : Load Optimal Defaults
```

Using the arrow keys, enable *Boot From LAN First*. Exit the BIOS setup, saving changes.

When prompted, press “Control-Alt-B” as the VMICPCI-7697 reboots. This will activate the LANWorks BIOS setup screen.

LANWorks BIOS Setup

Below is the screen in which the various options for booting through LANWorks are set.



```
BootWare Network Boot ROM
(C) Copyright LANWorks Technologies Inc. 1987-1998. All rights reserved
DC21143PCI 10/100 V100 (971031)

+-------------------------------------------+-----------------------------------+
|                   <Current setup>         |           <New Setup>             |
|                                           |                                   |
| I/O Base:                 E400h           |                                   |
| IRQ:                       11             |                                   |
|                                           |                                   |
|                                           |                                   |
| Boot Protocol:             RPL            |           RPL                     |
| Default Boot:            Network          |           Network                 |
| Local Boot:              Enabled          |           Enabled                 |
|                                           |                                   |
+-------------------------------------------+-----------------------------------+

Use cursor keys to edit: Up/Down Change Field, Left/Right Change Value
ESC to Quit, F10 to Save
```

Boot Protocol

The Boot Protocol required is dependent on the network system being utilized, check with your network administrator for the correct protocol. The following options are available in the Boot Protocol field:

- **RPL** **(Default)**
- TCP/IP BootP
- TCP/IP DHCP
- Netware 802.3
- Netware 802.2
- Netware EthII

Default Boot

The following options are available in the Default Boot field:

- **Network (Default) -** Selects Network Boot as the default boot device
- Local **-** Selects a Local Drive as the default boot device

Local Boot

The following options are available in the Local Boot field:

- Disabled **-** Disables Local Boot from a Hard Drive or Floppy
- **Enabled (Default) -** Enables Local Boot from a Hard Drive or Floppy

Appendix

D

Award - BIOS

Contents

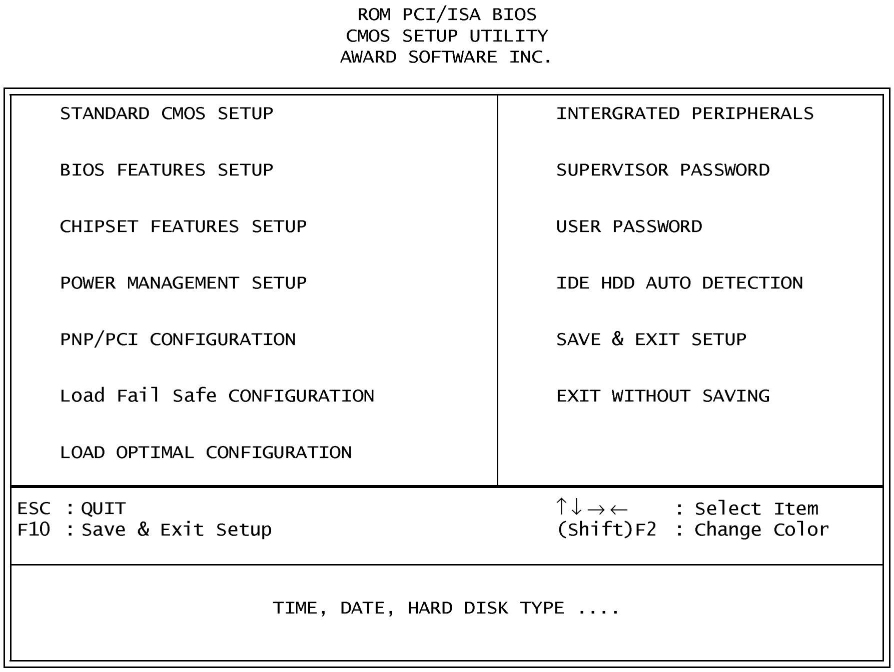
Introduction

The VMICPCI-7697 utilizes the BIOS (Basic Input / Output system) in the same manner as other PC / AT compatible computers. This appendix describes the menus and options associated with the VMICPCI-7697 BIOS.

System BIOS Setup Utility

During system bootup, press the Delete key to access the Award *Elite*BIOS CMOS Setup Utility screen. From this screen, the user can select any section of the Award (system) BIOS for configuration, such as floppy drive configuration or system memory.

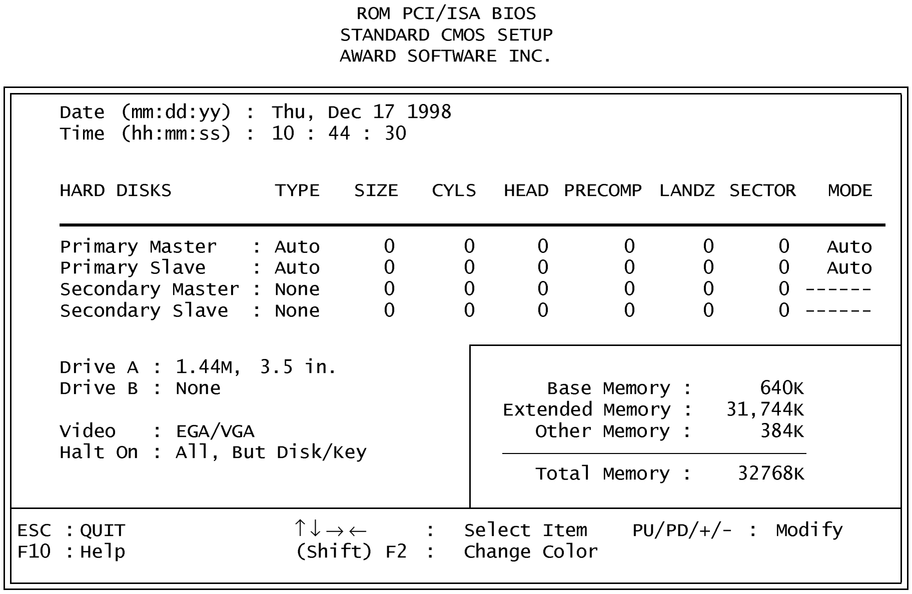
The parameters shown throughout this section are the default values.



```
                              ROM PCI/ISA BIOS
                             CMOS SETUP UTILITY
                            AWARD SOFTWARE INC.

    STANDARD CMOS SETUP                     INTERGRATED PERIPHERALS

    BIOS FEATURES SETUP                     SUPERVISOR PASSWORD

    CHIPSET FEATURES SETUP                  USER PASSWORD

    POWER MANAGEMENT SETUP                  IDE HDD AUTO DETECTION

    PNP/PCI CONFIGURATION                   SAVE & EXIT SETUP

    Load Fail Safe CONFIGURATION            EXIT WITHOUT SAVING

    LOAD OPTIMAL CONFIGURATION

ESC  : QUIT                                 ↑↓→←       : Select Item
F10  : Save & Exit Setup                    (Shift)F2  : Change Color

                      TIME, DATE, HARD DISK TYPE ....
```

Standard CMOS Setup

Selection of the first main menu item, the Standard CMOS Setup, allows the user to set the system clock and calendar, record disk drive parameters, video subsystem type, and select the type of errors that will cause a system halt.



```
                     ROM PCI/ISA BIOS
                    STANDARD CMOS SETUP
                    AWARD SOFTWARE INC.

    Date  (mm:dd:yy) :  Thu, Dec 17 1998
    Time  (hh:mm:ss) :  10 : 44 : 30


    HARD DISKS         TYPE   SIZE   CYLS   HEAD  PRECOMP  LANDZ  SECTOR   MODE
    ___________________________________________________________________________
    Primary Master   : Auto      0      0      0       0      0      0   Auto
    Primary Slave    : Auto      0      0      0       0      0      0   Auto
    Secondary Master : None      0      0      0       0      0      0 ------
    Secondary Slave  : None      0      0      0       0      0      0 ------

    Drive A : 1.44M,  3.5 in.
    Drive B : None                                Base Memory :      640K
                                              Extended Memory :   31,744K
    Video   : EGA/VGA                            Other Memory :      384K
    Halt On : All, But Disk/Key                  ___________________________
                                                 Total Memory :    32768K

 ESC : QUIT             ↑↓→←       :  Select Item    PU/PD/+/-  :  Modify
 F10 : Help             (Shift) F2  :  Change Color
```

Setting The Date

Press the left or right arrow key to move the cursor to the desired field (month, day, year). Press the PgUp or PgDn key to step through the available choices, or type in the information. Note that the day of the week field is for information purposes only, and will be set based on the other date information.

Setting The Time

The time format is based on the 24-hour military-time clock. For example, 1 PM is 13:00:00. Press the left or right arrow key to move the cursor to the desired field (hour, minute, seconds). Press the PgUp or PgDn key to step through the available choices, or type in the information.

Primary Master/Slave

The VMICPCI-7697 has the capability of utilizing one IDE hard disk drive on the Primary Master bus. The default setting is None. The Primary Slave is not used with the VMICPCI-7697.

Secondary Master/Slave

The Secondary Master is the resident flash disk. The default setting is None. The Secondary Slave is not assignable.

Floppy Disk Drive

Floppy Drive A

The VMICPCI-7697 supports one floppy disk drive. The options are:

• None	No diskette drive installed
• 360K, 5.25 in	5-1/4 inch PC-type standard drive; 360 kilobyte capacity
• 1.2M, 5.25 in	5-1/4 inch AT-type high-density drive; 1.2 megabyte capacity
• 720K, 3.5 in	3-1/2 double-sided drive; 720 kilobyte capacity
• 1.44M, 3.5 in	3-1/2 inch double-sided drive; 1.44 megabyte capacity
• 2.88M, 3.5 in	3-1/2 inch double-sided drive; 2.88 megabyte capacity

Use PgUp or Pgdn to select the floppy drive. The default is 1.44M, 3.5 inch.

Floppy Drive B

The VMICPCI-7697 does not support a second floppy drive. The default is None.

Video

The VMICPCI-7697 has an EGA/VGA graphics chip onboard. The BIOS supports a secondary video subsystem, but it is not selected in Setup. The default is EGA/VGA. Use the PgUp or PgDn key to select the video.

Halt On

During the power-on self-test (POST), the computer may be made to halt if the BIOS detects a hardware error. The options are:

- All Errors — If the BIOS detects any non-fatal error, POST stops and prompts you for corrective action
- No Errors — POST does not stop for any errors
- All, But Keyboard — Post does not stop for a keyboard error, but stops for all other errors
- All, But Diskette — POST does not stop for diskette drive errors, but stops for all other errors
- All, But Disk/Key — POST does not stop for a keyboard or disk error, but stops for all other errors (default)

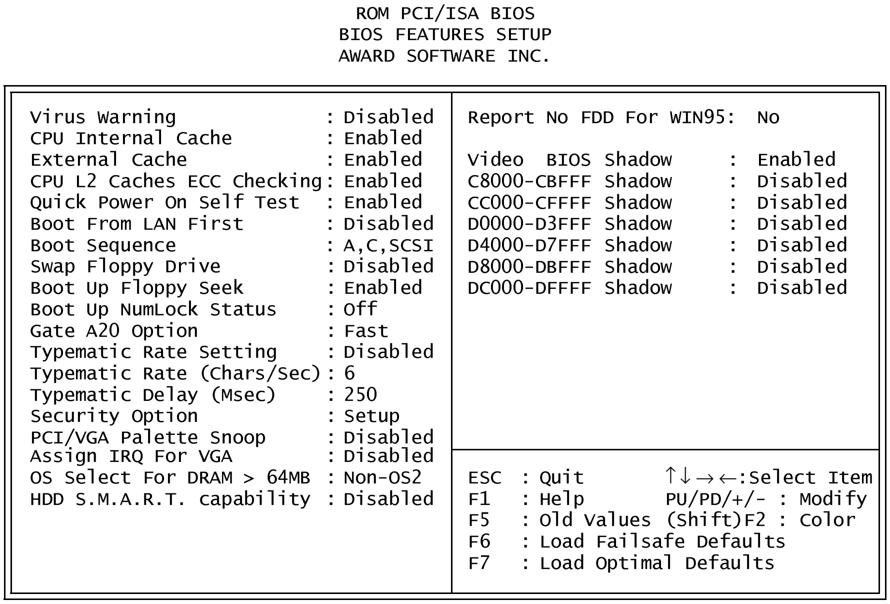
Use the PgUp or PgDn key to select the errors which will halt the system. The default is All, But Disk/Key.

Memory

The Memory field at the lower right of the screen is for informational purposes only and can not be modified by the user. This field displays the total RAM installed in the system, and the amounts allocated to base, extended, and other (high) memory.

BIOS Features Setup

This screen, selected from the CMOS Setup Utility screen, allows the user to configure options that are in addition to the basic BIOS features.



```
                    ROM PCI/ISA BIOS
                  BIOS FEATURES SETUP
                  AWARD SOFTWARE INC.

Virus Warning             : Disabled   Report No FDD For WIN95:  No
CPU Internal Cache        : Enabled
External Cache            : Enabled    Video  BIOS Shadow     :  Enabled
CPU L2 Caches ECC Checking: Enabled    C8000-CBFFF Shadow     :  Disabled
Quick Power On Self Test  : Enabled    CC000-CFFFF Shadow     :  Disabled
Boot From LAN First       : Disabled   D0000-D3FFF Shadow     :  Disabled
Boot Sequence             : A,C,SCSI   D4000-D7FFF Shadow     :  Disabled
Swap Floppy Drive         : Disabled   D8000-DBFFF Shadow     :  Disabled
Boot Up Floppy Seek       : Enabled    DC000-DFFFF Shadow     :  Disabled
Boot Up NumLock Status    : Off
Gate A20 Option           : Fast
Typematic Rate Setting    : Disabled
Typematic Rate (Chars/Sec): 6
Typematic Delay (Msec)    : 250
Security Option           : Setup
PCI/VGA Palette Snoop     : Disabled
Assign IRQ For VGA        : Disabled
OS Select For DRAM > 64MB : Non-OS2    ESC  : Quit       ↑↓→←:Select Item
HDD S.M.A.R.T. capability : Disabled   F1   : Help       PU/PD/+/- : Modify
                                       F5   : Old Values (Shift)F2 : Color
                                       F6   : Load Failsafe Defaults
                                       F7   : Load Optimal Defaults
```

Virus Warning

When enabled, you receive a warning message if a program (specifically, a virus) attempts to write to the boot sector or the partition table of the hard disk drive. You should then run an anti-virus program. Keep in mind that this feature protects only the boot sector, not the entire hard drive. The default is Disabled.

NOTE: Many disk diagnostic programs that access the boot sector table can trigger the virus warning message. If you plan to run such a program, we recommend that you first disable the virus warning.

CPU Internal Cache

Enabling the cache memory enhances the speed of the processor. When the CPU requests data, the system transfers the requested data from the main DRAM into the cache memory were it is stored until processed by the CPU. The default is Enabled.

External Cache

The external cache (up to 512K) provides additional cache for use by the CPU. This cache functions in the same manner as internal cache. The default is Enabled.

CPU L2 Cache ECC Checking

When you select Enabled, memory checking is enable when the external cache contains ECC SRAMs. The default is Enabled.

Quick Power On Self Test

When enabled, certain checks normally performed during the POST are omitted, decreasing the time required to run the POST. The default is Enabled.

Boot From LAN First

When enabled, this option allows the CPU to boot off of a connected network. The default is Disabled.

Boot Sequence

Determines the order in which the BIOS will seek a bootable drive. The default order is the floppy disk (A:), then the internal hard drive (C:), followed by the SCSI drive.

Swap Floppy Drive

The option is functional only in a system with two floppy drives. The VMICPCI-7697 supports only one floppy drive. Changing this option will have no effect on the system. The default is Disabled.

Boot Up Floppy Seek

When enabled, the BIOS will test the floppy to determine if the floppy drive has 40 or 80 tracks. Only 360K floppy drives have 40 tracks. It is recommended that this option be set to disabled unless a third party 360K floppy drive is being used in the system. The default is Enabled.

Boot Up NumLock Status

Toggle between On or Off to control the state of the NumLock key when the system boots. When toggled On, the numeric keypad generates numbers instead of controlling the cursor operations. The default is Off.

Gate A20 Option

Gate A20 refers to the way the system addresses memory above 1 MB (extended memory). When set to Fast, the system chipset controls Gate A20. When set to Normal, a pin in the keyboard controller controls Gate A20. Setting Gate A20 to Fast improves system speed, particularly with OS/2 and Windows. The default setting is Fast.

Typematic Rate Setting

This option enables or disables the Typematic Rate and Delay settings. When disabled, the values in the Typematic Rate and Delay are ignored. The default is Disabled.

Typematic Rate (Chars/Sec)

If the Typematic Rate Setting (above) is enabled, this feature determines the rate a character is repeated when a key is held down. The options are: 6, 8, 10, 12, 15, 20, 24, or 30 characters per second.

Typematic Delay (Msec)

If the Typematic Rate Setting (above) is enabled, this feature determines the delay before a character starts repeating when a key is held down. The options are: 250, 500, 750, or 1000 milliseconds.

Security Option

If a password has been set, this determines whether the password is required every time the system boots, or only when the Setup is accessed. The default is Setup.

PCI/VGA Palette Snoop

Enabling the video palette snoop allows the PMC add-in card to share a common palette with the on-board graphics controller. The default is Disabled.

Assign IRQ for VGA

Select Enabled only if your VGA card requires an assigned IRQ. Most ordinary cards do not; some high-end cards with video capture function do. Consult the information that comes with your VGA card to determine whether it needs an assigned IRQ. The default is Disabled

OS Select For DRAM>64MB

Select OS2 only if you are running OS/2 operating system with greater than 64 MB of RAM on the system. The default is Non-OS2.

HDD S.M.A.R.T. Capability

SMART is an acronym for Self-Monitoring Analysis and Reporting Technology system. SMART is a hard drive self-diagnostic feature available on some IDE hard drives. The default is Disabled.

Report No FDD For WIN95

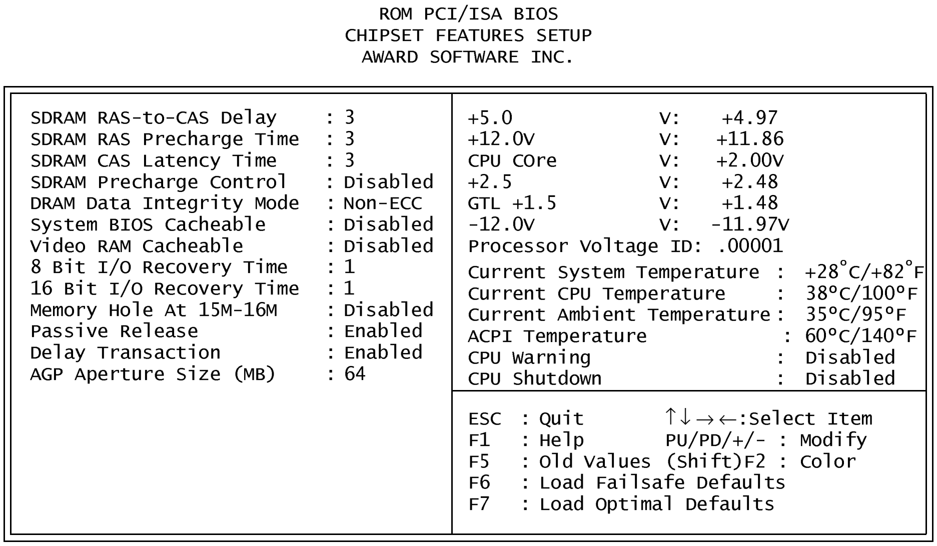
Select *Yes* to release IRQ6 when the system contains no floppy drive, for compatibility with Windows 95 logo certification. In the Integrated Peripherals screen, select *Disabled* for the Onboard FDC Controller field. The default is No.

Chipset Features Setup

This section describes features of the Intel 82430TX PCIset.

Advanced Options

The parameters in this screen are for system designers, service personnel, and technically competent users only. Do not reset these values without a complete understanding of the consequences.



```
                         ROM PCI/ISA BIOS
                      CHIPSET FEATURES SETUP
                        AWARD SOFTWARE INC.

SDRAM RAS-to-CAS Delay     : 3         +5.0            V:    +4.97
SDRAM RAS Precharge Time   : 3         +12.0V          V:   +11.86
SDRAM CAS Latency Time     : 3         CPU COre        V:   +2.00V
SDRAM Precharge Control    : Disabled  +2.5            V:    +2.48
DRAM Data Integrity Mode   : Non-ECC   GTL +1.5        V:    +1.48
System BIOS Cacheable      : Disabled  -12.0V          V:   -11.97V
Video RAM Cacheable        : Disabled  Processor Voltage ID: .00001
8 Bit I/O Recovery Time    : 1         Current System Temperature :  +28°C/+82°F
16 Bit I/O Recovery Time   : 1         Current CPU Temperature    :  38°C/100°F
Memory Hole At 15M-16M     : Disabled  Current Ambient Temperature:  35°C/95°F
Passive Release            : Enabled   ACPI Temperature            : 60°C/140°F
Delay Transaction          : Enabled   CPU Warning                :  Disabled
AGP Aperture Size (MB)     : 64        CPU Shutdown               :  Disabled

                                       ESC  : Quit       ↑↓→←:Select Item
                                       F1   : Help       PU/PD/+/- : Modify
                                       F5   : Old Values (Shift)F2 : Color
                                       F6   : Load Failsafe Defaults
                                       F7   : Load Optimal Defaults
```

SDRAM(CAS Lat/RAS-to-CAS)

You can select a combination of CAS latency and RAS-to-CAS delay in HCLKs of 2 or 3. The values in this field depend on the DRAM installed. Do not change the values in this field unless you change specifications of the installed DRAM or the installed CPU. The default is 3.

SDRAM RAS Precharge Time

The precharge time is the number of cycles it takes for the RAS to accumulate its charge before DRAM refresh. If insufficient time is allowed, refresh may be incomplete and the DRAM may fail to retain data. This field applies only if synchronous DRAM is installed in the system. The default is 3.

SDRAM CAS Latency Time

When synchronous DRAM is installed, you can control the number of CLKs between when the SDRAMs sample a read command and when the controller samples read data from the SDRAMs. Do not reset this field from the default value specified by the system designer. The default is 3.

SDRAM Precharge Control

When Enabled, all CPU cycles to SDRAM result in an All Banks Precharge Command on the SDRAM interface. The default is Disabled.

DRAM Data Integrity Mode

Select Non-ECC or ECC (error-correcting code), according to the type of installed DRAM. The default is Non-ECC

System BIOS Cacheable

Selecting Enabled allows caching of the system BIOS ROM at F0000h-FFFFFh, resulting in better system performance. However, if any program writes to this memory area, a system error may result. The default is Disabled

Video RAM Cacheable

Selecting Enabled allows caching of the video BIOS ROM at C0000h to C7FFFh. If a program writes to this memory area, a system error may occur. The default is Disabled.

8 Bit I/O Recovery Time

The I/O recovery mechanism adds bus clock cycles between PCI-originated I/O cycles to the ISA bus. This delay takes place because the PCI bus is faster than the ISA bus. The default is 1 (one).

16 Bit I/O Recovery Time

The I/O recovery mechanism adds bus clock cycles between PCI-originated I/O cycles to the ISA bus. This delay takes place because the PCI bus is faster than the ISA bus. The default is 1.

These two fields (8 Bit and 16 Bit) let you add recovery time (in bus clock cycles) for 8-bit and 16-bit I/O. The default is 1.

Memory Hole at 15M-16M

This area of system memory may be reserved for ISA adapter ROM. When this area is reserved, it cannot be cached. Refer to the documentation that came with the peripheral that require the use of this area of system memory for memory requirements. The default is Disabled.

Passive Release

When Enabled, CPU to PCI bus accesses are allowed during passive release. Otherwise, the arbiter only accepts another PCI master access to local DRAM. The default is Enabled.

Delayed Transaction

The chipset has an embedded 32-bit posted write buffer to support delay transactions cycles. Select Enabled to support compliance with PCI specification version 2.1. The default is Enabled.

AGP Aperture Size (MB)

Select the size of the Accelerated Graphics Port (AGP) aperture. The aperture is a portion of the PCI memory address range dedicated for graphics memory address space. Host cycles that hit the aperture range are forwarded to the AGP without any translation. See http://www.agpforum.org for APG information. The default is 64.

CPU Warning Temperature

Select the combination of lower and upper limits for the CPU temperature. If the CPU temperature extends beyond either limit, any warning mechanism programmed into your system will be activated. The default is Disabled.

CPU Shutdown

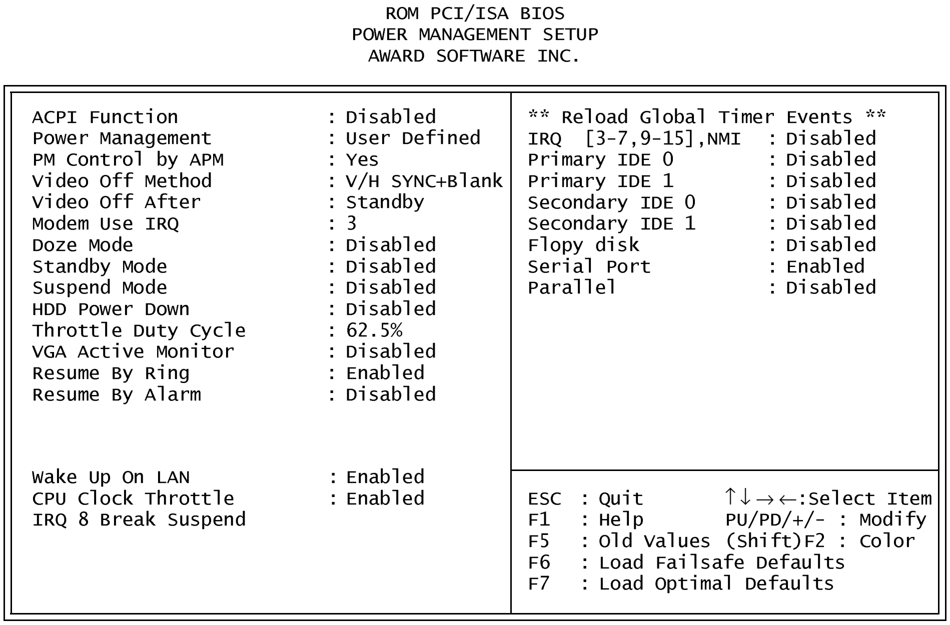
When Enabled, this feature will throttle the CPU speed down when the temperature exceeds the preset limit. The percentage the CPU will be throttled is set in the Power Management screen by the Throttle Duty Cycle. The default is Disabled.

ACPI Temperature

If the ACPI function is enabled in the Power Management setup screen, then the ACPI Temperature will determine the temperature at which the ACPI power reduction functions will become enabled. The ACPI function is DISABLED in the default BIOS setup, and should only be enabled when using an operating system that fully supports ACPI.

Power Management

This section discusses the power management features provided with the installed Award BIOS.



```
                    ROM PCI/ISA BIOS
                 POWER MANAGEMENT SETUP
                  AWARD SOFTWARE INC.

ACPI Function              : Disabled         ** Reload Global Timer Events **
Power Management           : User Defined     IRQ  [3-7,9-15],NMI   : Disabled
PM Control by APM          : Yes              Primary IDE 0         : Disabled
Video Off Method           : V/H SYNC+Blank   Primary IDE 1         : Disabled
Video Off After            : Standby          Secondary IDE 0       : Disabled
Modem Use IRQ              : 3                Secondary IDE 1       : Disabled
Doze Mode                  : Disabled         Flopy disk            : Disabled
Standby Mode               : Disabled         Serial Port           : Enabled
Suspend Mode               : Disabled         Parallel              : Disabled
HDD Power Down             : Disabled
Throttle Duty Cycle        : 62.5%
VGA Active Monitor         : Disabled
Resume By Ring             : Enabled
Resume By Alarm            : Disabled


Wake Up On LAN             : Enabled
CPU Clock Throttle         : Enabled          ESC  : Quit       ↑↓→←:Select Item
IRQ 8 Break Suspend                           F1   : Help       PU/PD/+/- : Modify
                                              F5   : Old Values (Shift)F2 : Color
                                              F6   : Load Failsafe Defaults
                                              F7   : Load Optimal Defaults
```

ACPI Function

ACPI is Advanced Configuration Power Management Interface. Selecting Enabled enables ACPI device node reporting from the BIOS to the operating system. This method is used by Windows NT and Windows 98 to power manage the computer. The default is Disabled.

Power Management

This option disables or sets the power management options. The options are:

• Max Saving	Maximum power savings. This option is only available for SL CPUs. Inactivity period is 1 minute in each mode.
• User Define	Set each mode individually. Select time-out periods in the PM timers section.
• Min Savings	Minimum power savings. Inactivity period is 1 hour in each mode (except the hard drive).
• Disabled	Turns off all power management features.

The default is Disabled.

PM Control by APM

Advanced Power Management (APM) provides better power savings. The default is No.

Video Off Method

Determines the manner in which the monitor is blanked. The options are:

- V/H SYNC+Blank System turns off vertical and horizontal synchronization ports and writes blanks to the video buffer.
- DPMS Support Select this option if your monitor supports the Display Power Management Signaling (DPMS) standard of the Video Electronics Standards Association (VESA). Use the software supplied for your video subsystem to select video power management values.
- Blank Screen System only writes blanks to the video buffer.

The default setting is V/H SYNC+Blank.

Video Off After

As the system moves from lesser to greater power-saving modes, select the mode in which you want the monitor to blank. The default is Standby.

Modem Use IRQ

Name the interrupt request (IRQ) line assigned to the modem (if any) on your system. Activity of the selected IRQ awakens the system. The default setting is N/A.

Doze Mode

After the selected period of system inactivity (1 minute to 1 hour), the CPU clock runs at a slower speed while all other devices still operate at full speed. The default is Disabled.

Standby Mode

After the selected period of system inactivity (1 minute to 1 hour), the fixed disk drive and the video shut off while all other devices still operate at full speed. The default setting is Disabled.

Suspend Mode

After the selected period of system inactivity (1 minute to 1 hour), all devices except the CPU shut off. The default setting is Disabled.

HDD Power Down

After the selected period of drive inactivity (1 to 15 minutes), the hard disk drive powers down while all other devices remain active. The default is Disabled.

Throttle Duty Cycle

When the system enters Doze mode, the CPU clock runs only part of the time. You may select the percent of time that the clock runs. The default is 62.5%.

VGA Active Monitor

When Enabled, any video activity restarts the global timer for Standby mode. The default is Disabled.

Resume By Ring

When Enabled, an input signal on the serial Ring Indicator (RI) line (in other words, an incoming call on the modem) awakens the system from a soft off state. The default is Enabled.

Resume By Alarm

When Enabled, an alarm signal awakens the system from a soft off state. The default is Disabled.

Wake Up On LAN

When Enabled, a request from the LAN awakens the system from a soft off state. The default is Enabled.

CPU Clock Throttle

When Enabled, this allows the power management routines to adjust the clock speed of the processor down to compensate for temperature. The default is Enabled.

IRQ 8 Break Suspend

You can Enable or Disable monitoring of IRQ8 (the Real Time Clock) so it does not awaken the system from Suspend mode. The default is Disabled.

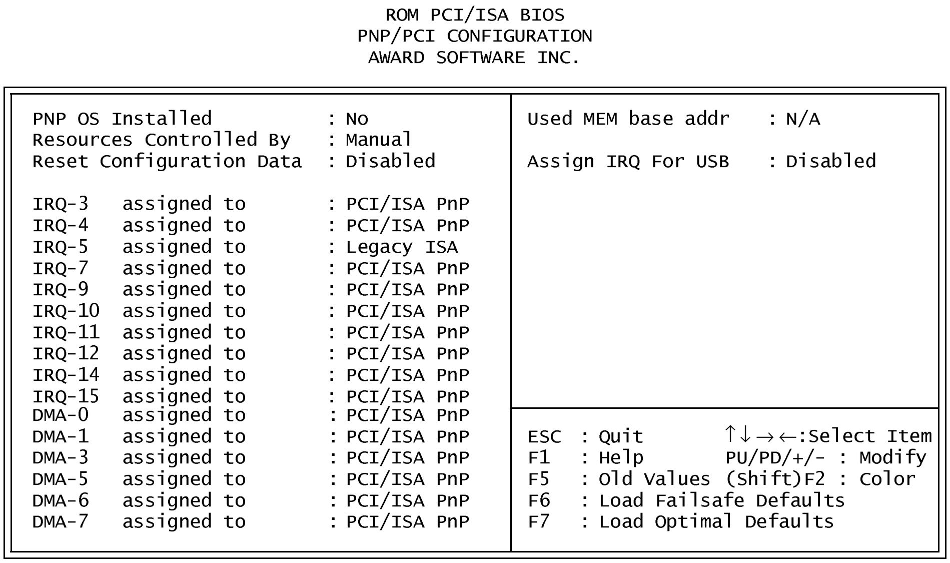
Reload Global Timer Events

When Enabled, an event occurring on each device listed below restarts the global timer for Standby mode.

- IRQ3-7, 9-15, NMI
- Primary IDE 0
- Primary IDE 1
- Secondary IDE 0
- Secondary IDE 1
- Floppy Disk
- Serial Port
- Parallel Port

PnP/PCI Configuration

This section describes the PNP / PCI options available.



```
                    ROM PCI/ISA BIOS
                  PNP/PCI CONFIGURATION
                  AWARD SOFTWARE INC.

 PNP OS Installed          : No             Used MEM base addr    : N/A
 Resources Controlled By   : Manual
 Reset Configuration Data  : Disabled       Assign IRQ For USB    : Disabled

 IRQ-3   assigned to       : PCI/ISA PnP
 IRQ-4   assigned to       : PCI/ISA PnP
 IRQ-5   assigned to       : Legacy ISA
 IRQ-7   assigned to       : PCI/ISA PnP
 IRQ-9   assigned to       : PCI/ISA PnP
 IRQ-10  assigned to       : PCI/ISA PnP
 IRQ-11  assigned to       : PCI/ISA PnP
 IRQ-12  assigned to       : PCI/ISA PnP
 IRQ-14  assigned to       : PCI/ISA PnP
 IRQ-15  assigned to       : PCI/ISA PnP
 DMA-0   assigned to       : PCI/ISA PnP
 DMA-1   assigned to       : PCI/ISA PnP    ESC  : Quit       ↑↓→←:Select Item
 DMA-3   assigned to       : PCI/ISA PnP    F1   : Help       PU/PD/+/- : Modify
 DMA-5   assigned to       : PCI/ISA PnP    F5   : Old Values (Shift)F2 : Color
 DMA-6   assigned to       : PCI/ISA PnP    F6   : Load Failsafe Defaults
 DMA-7   assigned to       : PCI/ISA PnP    F7   : Load Optimal Defaults
```

PNP OS Installed

Select Yes if the system operating environment is Plug-and-Play aware (e.g., Windows 95). The default is No.

Resources Controlled By

The Plug-and-Play BIOS can automatically configure all the boot and Plug-and-Play compatible devices. If Auto is selected, all the interrupt request (IRQ) and DMA assignment and memory base address fields disappear, as the BIOS automatically assigns them. The default is Manual.

Reset Configuration Data

Select Enabled to reset Extended System Configuration Data (ESCD) when exiting Setup. Use this feature if a new add-on has been installed and the system reconfiguration has caused such a serious conflict that the operating system cannot boot. The default is Disabled.

IRQ *n* Assigned to

When resources are controlled manually, assign each system interrupt as one of the following types, depending on the type of device using the interrupt:

- Legacy ISA Devices compliant with the original PC AT bus specification, requiring a specific interrupt (such as IRQ4 for serial port 1).
- PCI/ISA PnP Devices compliant with the Plug-and-Play standard, whether designed for PCI or ISA bus architecture.

DMA *n* Assigned to

When resources are controlled manually, assign each system DMA channel as one of the following types, depending on the type of device using the interrupt:

- Legacy ISA Devices compliant with the original PC AT bus specification, requiring a specific DMA channel
- PCI/ISA PnP Devices compliant with the Plug-and-Play standard, whether designed for PCI or ISA bus architecture.

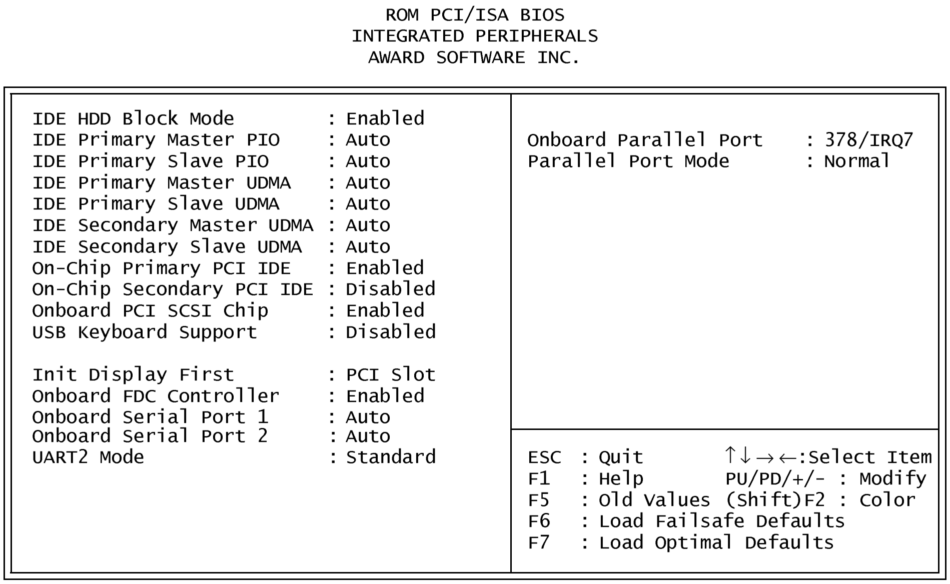
Used Mem Base Addr

Select a base address for the memory area used by any peripheral that requires high memory. The defaults are:

• Used MEM base addr	N/A
• Assign IRQ For USB	Disabled

Integrated Peripherals

This section describes the setup for integrated peripherals in the system.



```
                        ROM PCI/ISA BIOS
                     INTEGRATED PERIPHERALS
                      AWARD SOFTWARE INC.

 IDE HDD Block Mode        : Enabled
 IDE Primary Master PIO    : Auto       Onboard Parallel Port    : 378/IRQ7
 IDE Primary Slave PIO     : Auto       Parallel Port Mode       : Normal
 IDE Primary Master UDMA   : Auto
 IDE Primary Slave UDMA    : Auto
 IDE Secondary Master UDMA : Auto
 IDE Secondary Slave UDMA  : Auto
 On-Chip Primary PCI IDE   : Enabled
 On-Chip Secondary PCI IDE : Disabled
 Onboard PCI SCSI Chip     : Enabled
 USB Keyboard Support      : Disabled

 Init Display First        : PCI Slot
 Onboard FDC Controller    : Enabled
 Onboard Serial Port 1     : Auto
 Onboard Serial Port 2     : Auto
 UART2 Mode                : Standard   ESC  : Quit       ↑↓→←:Select Item
                                        F1   : Help       PU/PD/+/- : Modify
                                        F5   : Old Values (Shift)F2 : Color
                                        F6   : Load Failsafe Defaults
                                        F7   : Load Optimal Defaults
```

IDE HDD Block Mode

Block mode is also called block transfer, multiple commands, or multiple sector read / write. If the IDE hard drive supports block mode, select Enabled for automatic detection of the optimal number of block read / writes per sector the drive can support. The default is Enabled.

IDE Primary Master PIO

This IDE PIO (Programmed Input / Output) field allows setting a PIO mode (0-4) for the IDE devices that the onboard IDE interface supports. Modes 0 through 4 provide successively increased performance. In Auto mode, the system automatically determines the best mode for each device. The default is Auto.

IDE Primary Slave PIO

This IDE PIO (Programmed Input / Output) field allows setting a PIO mode (0-4) for the IDE devices that the onboard IDE interface supports. Modes 0 through 4 provide successively increased performance. In Auto mode, the system automatically determines the best mode for each device. The default is Auto.

IDE Primary/Secondary Master UDMA

Ultra DMA/33 implementation is possible only if the IDE hard drive supports it and the operating environment includes a DMA driver (Windows 95 OSR2 or a third-party IDE bus master driver). If the hard drive and the operating system both support Ultra DMA/33, select Auto to enable BIOS support. The default is Auto.

IDE Primary/Secondary Slave UDMA

Ultra DMA/33 implementation is possible only if the IDE hard drive supports it and the operating system includes a DMA driver (Windows 95 OSR2 or a third-party IDE bus master driver). If the hard drive and the operating system both support Ultra DMA/33, select Auto to enable BIOS support. The default is Auto.

On-Chip Primary PCI IDE

The integrated peripheral controller contains an IDE interface with support for two IDE channels. Select Enabled to activate each channel separately. The default is Enabled.

On-Chip Secondary PCI IDE

The integrated peripheral controller contains an IDE interface with support for two IDE channels. Select Enabled to activate each channel separately. When this option is disabled, the IDE Primary Master PIO and IDE Primary Slave PIO options in this screen are removed. The default is Disabled.

Onboard PCI SCSI Chip

Select Enabled if your system contains a built-in PCI SCSI controller. The default is Enabled.

NOTE: Although the Onboard PCI SCSI Chip is listed in the BIOS, and the default is enabled, it is NOT supported on the standard VMICPCI-7697.

USB Keyboard Support

Select Enabled if the operating system contains a Universal Serial Bus (USB) controller and you have a USB keyboard. The default is Disabled.

Init Display First

Initialize the AGP video display before initializing any other display device on the system. Thus the AGP display becomes the primary display. The default is PCI Slot.

Onboard FDC Controller

Select Enabled to activate the floppy disk controller (FDC) installed on the system board. If an add-in FDC was installed or the system has no floppy drive, select Disabled in this field. The default is Enabled.

Onboard Serial Port 1/2

Select an address and corresponding interrupt for the first and second serial ports. The default for both ports is Auto.

UART 2 Mode

Select an operating mode for the second serial port. The infrared options listed below are not supported by the VMICPCI-7697. The options are:

• Standard	RS-232C serial port
• IrDA 1.0	N / A
• MIR 0.57M	N / A
• MIR 1.15M	N / A
• FIR	N / A
• ASK IR	N / A

The default for this setting is Standard.

Onboard Parallel Port

Select an address and corresponding interrupt for the physical parallel (printer) port. The default setting is 378/IRQ7.

Parallel Port Mode

Select an operating mode for the onboard parallel (printer) port. The default setting is Normal.

Appendix

E

Device Configuration: I/O and Interrupt Control

Contents

Introduction

This appendix provides the user with the information needed to develop custom applications for the VMICPCI-7697. The CPU board on the VMICPCI-7697 is unique in that the BIOS can not be removed; it must be used in the initial boot cycle. A custom application, like a revised operating system for example, can only begin to operate after the BIOS has finished initializing the CPU. The VMICPCI-7697 will allow the user to either maintain the current BIOS configuration or alter this configuration to be more user specific, but this alteration can only be accomplished after the initial BIOS boot cycle has completed.

BIOS Operations

When the VMICPCI-7697 is powered on, control immediately jumps to the BIOS. The BIOS initiates a Power-on Self-Test (POST) program which instructs the microprocessor to initialize system memory as well as the rest of the system. The BIOS establishes the configuration of all on-board devices by initializing their respective I/O and Memory addresses and interrupt request lines. The BIOS then builds an interrupt vector table in main memory, which is used for interrupt handling. The default interrupt vector table and the default address map is described in Chapter 3 of this manual. Finally, the BIOS jumps to the selected boot device. This is the point at which a custom operating system could take over control of the board and proceed with a custom configuration and/or custom application. A user application could override the configuration set by the BIOS and reconfigure the system or it could accept what the BIOS initialized.

BIOS Control Overview

There are two areas on the VMICPCI-7697 in which the user must be familiar with in order to override the initial BIOS configuration. These include the device addresses and the device interrupts. This appendix reviews the details of these addresses and interrupts, and provides a reference list for the individual devices used on the board.

The VMICPCI-7697 utilizes the high-performance Peripheral Component Interconnect (PCI) bus along with the Industrial Standard Architecture (ISA) bus. In general, the PCI bus is plug-and-play compatible. The components that are connected to the PCI bus are not always placed at a standard I/O or Memory address, nor are they connected to a standard interrupt request line as is the case with ISA bus devices. These PCI bus devices are re-established by the BIOS during every boot cycle, meaning that these devices will not always be located at the same address or connected to the same interrupt request line every time the CPU is booted. This appendix lists the defaults that are found by powering up a specific VMICPCI-7697.

Functional Overview

The block diagram included in Figure E-1 on page 129 illustrates the VMICPCI-7697 emphasizing the I/O features, including the Embedded PCI bridge.

The circled number in the upper right corner of a function block references the appropriate data book necessary for the programming of the function block.

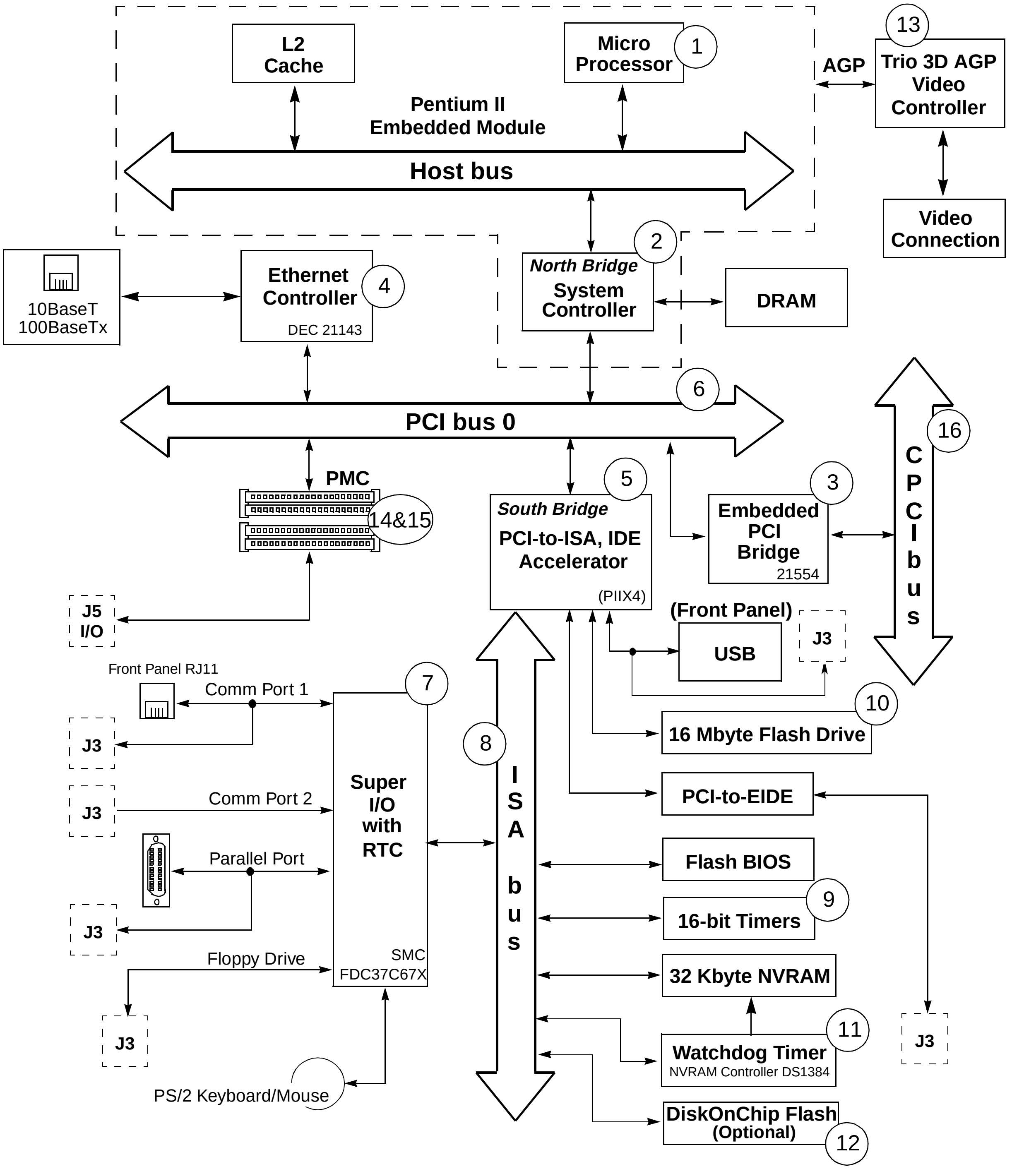


Figure E-1 VMICPCI-7697 Block Diagram

Data Book References

1. *Intel Pentium® II Processor at 233 MHz, 266 MHz, 300 MHz, and 33MHz*

   January 1998, Order Number: 243335-003
   Intel Literature Sales
   P.O. Box 7641
   Mt. Prospect, IL 60056-7641
   Santa Clara, CA 95052-8119
   (800) 548-4752
   www.intel.com

2. *Intel 440BX AGP set: 82443BX Host Bridge/Controller*

   April 1998, Order Number: 290633-001
   Intel Corporation
   P.O. Box 58119
   Santa Clara, CA 95052-8119
   (408) 765-8080
   www.intel.com

3. *Intel 21554 Embedded PCI Bridge*

   Intel Corporation
   P.O. Box 58119
   Santa Clara, CA 95052-8119
   (408) 765-8080
   www.intel.com

4. *Intel 21143 PCI/CardBUS10/100 Mb/s Ethernet LAN Controller*

   October 1998, Order Number 278074-001
   Intel Corporation
   P.O. Box 58119
   Santa Clara, CA 95052-8119
   (408) 765-8080
   www.intel.com

5. *Intel 82440TX PCIset ISA Bridge*

   82371EB PCI ISA IDE Xcellerator (PIIX4E)
   2200 Mission College Boulevard
   P.O. Box 58119
   Santa Clara, CA 95052-8119

6. *PCI Local Bus Specification, Rev. 2.1*

   PCI Special Interest Group

P.O. Box 14070
Portland, OR 97214
(800) 433-5177 (U.S.)
(503) 797-4207 (International)
(503) 234-6762 (FAX)

7. *SMC FDC37C67X Enhanced Super I/O Controller*

SMC Component Products Division
300 Kennedy Drive
Hauppauge, NY 11788
(516) 435-6000
(516) 231-6004 (FAX)

8. *ISA & EISA, Theory and Operation*

Solari, Edward
Annabooks
15010 Avenue of Science, Suite 101
San Diego, CA 92128 USA
ISBN 0-929392 -15-9

9. *82C54 CHMOS Programmable Internal Timer*

Intel Corporation
2200 Mission College Boulevard
P.O. Box 58119
Santa Clara, CA 95052-8119

10. *Flash ChipSet Product Manual*

SanDisk Corporation
140 Caspian Court
Sunnyvale, CA 94089-9820

11. *DS 1384 Watchdog Timekeeping Controller*

Dallas Semiconductor
4461 South Beltwood Pwky.
Dallas, TX 75244-3292

12. *M-Systems Corporate Headquarters*

    USA Office
    39899 Balentine Dr.
    Suite 335
    Newark, CA 94560
    Tel: 510-413-5950
    Fax: 510-413-5980
    Email: info@m-sys.com

13. *S3 Trio 3d AGP Video Controller*

    P.O. Box 58058
    Santa Clara, CA 95052-8058
    (408) 588-8000

14. *CMC Specification, P1386/Draft 2.0*

    IEEE Standards Department
    Copyrights and Permissions
    445 Hoes Lanes, P.O. Box 1331
    Piscataway, NJ 08855-1331, USA

15. *PMC Specification, P1386.1/Draft 2.0*

    IEEE Standards Department
    Copyrights and Permissions
    445 Hoes Lanes, P.O. Box 1331
    Piscataway, NJ 08855-1331, USA

16. *CompactPCI™ Specification PICMG 2.0 R2.1*

    PCI Industrial Computer Manufactures' Group
    301 Edgewater Place
    Suite 220
    Wakefield, MA 01880
    (617) 224-1100
    (617) 224-1239 (FAX)
    www.picmg.org (Web)

17. *VMICPCI-7697 Specification*

    SSS-657697-000 Product Specification
    VMIC
    12090 South Memorial Parkway
    Huntsville, Alabama 35803-3308
    (800) 322-3616
    (256) 880-0444
    (256) 882-0859 (FAX)

Device Address Definition

The standard PC/AT architecture defines two distinctive types of address spaces for the devices and peripherals on the CPU board. These spaces have typically been named Memory address space and I/O address space. The boundaries for these areas are limited to the number of address bus lines that are physically located on the CPU board. The VMICPCI-7697 has 32 address bus lines located on the board, thereby defining the limit of the address space as 4 Gbyte. The standard PC/AT architecture defines Memory address space from zero to 4 Gbyte and the separate I/O address space from zero to 64 Kbyte.

ISA Devices

The ISA devices on the VMICPCI-7697 are configured by the BIOS at boot-up and fall under the realm of the standard PC/AT architecture. They are mapped in I/O address space within standard addresses and their interrupts are mapped to standard interrupt control registers. However, all of the ISA devices with the exception of the real-time clock, keyboard, and programmable timer are relocatable to almost anywhere within the standard 1 Kbyte of I/O address space. Table E-1 below defines the spectrum of addresses available for reconfiguration of ISA devices.

As previously stated, in the standard PC/AT system, all I/O devices are mapped in I/O address space; however, one exception exists. The Dynamic Random Access Memory (DRAM), Battery-Backed SRAM, and Watchdog Timer are addressed in Memory address space. The BIOS places DRAM at address zero and extends to the physical limit of the on-board DRAM.

Table E-1 ISA Device Mapping Configuration

Device	Memory Space	I/O Address Space	PIC Interrupt Options	Byte Address Boundary	Default
Floppy	N/A	[0x100 - 0xFF8]	IRQ1 - IRQ15	8	$3F0
Parallel Port	N/A	[0x100 - 0xFFC] [0x100 - 0xFF8]	IRQ1 - IRQ15	4 8	$378
Serial Port 1	N/A	[0x100 - 0xFF8]	IRQ1 - IRQ15	8	$3F8
Serial Port 2	N/A	[0x100 - 0xFF8]	IRQ1 - IRQ15	8	$2F8
Real-Time Clock	Nonrelocatable				
Keyboard	Nonrelocatable				
Auxiliary I/O	N/A	- Primary I/O [0x000 - 0xFFF] - Secondary I/O [0x000 - 0xFFF]	IRQ1, IRQ3-IRQ15	1 1	
Programmable Timer		Nonrelocatable 0X500	IRQ5	4	0X500
Watchdog Timer	Nonrelocatable 0XD8000	N/A	N/A	0XD	0XD8000
Battery-Backed SRAM	Nonrelocatable 0XD800E	N/A	N/A	(32K-0XD)	0XD800E

PCI Devices

PCI devices are fully configured under I/O and/or Memory address space. Table E-1 describes the PCI bus devices that are on-board the VMICPCI-7697 along with each device's configuration spectrum.

The PCI bus includes three physical address spaces. As with ISA bus, PCI bus supports Memory and I/O address space, but PCI bus includes an additional Configuration address space. This address space is defined to support PCI bus hardware configuration (refer to the PCI bus Specification for complete details on the configuration address space). PCI bus targets are required to implement Base Address registers in configuration address space to access internal registers or functions. The BIOS uses the Base Address register to determine how much space a device requires in a given address space and then assigns where in that space the device will reside. This functionality enables PCI devices to be located in either Memory or I/O address space.

Table E-2 PCI Device Mapping Configuration

Device	Memory Space	I/O Address Space	PIC Interrupt Options
Embedded PCI Bridge	Anywhere	Anywhere	PCI Defined
Ethernet	Anywhere	Anywhere	PCI Defined
Timer Interrupt Registers	N/A	Fixed	N/A
PMC	Anywhere	Anywhere	PCI Defined

Device Interrupt Definition

PC/AT Interrupt Definition

The interrupt hardware implementation on the VMICPCI-7697 is standard for computers built around the PC/AT architecture. The PC/AT evolved from the IBM PC/XT architecture. In the IBM PC/XT systems, only eight Interrupt Request (IRQ) lines exist, numbered from IRQ0 to IRQ7. These interrupt lines were included originally on a 8259A Priority Interrupt Controller (PIC) chip.

The IBM PC/AT computer added eight more IRQx lines, numbered IRQ8 to IRQ15, by cascading a second slave 8259A PIC into the original master 8259A PIC. The interrupt line IRQ2 at the master PIC was committed as the cascade input from the slave PIC. This master/slave architecture, the standard PC/AT interrupt mapping, is illustrated in Figure E-2 on page 136 within the PCI-to-ISA Bridge PIIX4 82371AB section of the diagram.

To maintain backward compatibility with PC/XT systems, IBM chose to use the new IRQ9 input on the slave PIC to operate as the old IRQ2 interrupt line on the PC/XT Expansion Bus. Thus, in AT systems, the IRQ9 interrupt line connects to the old IRQ2 pin on the AT Expansion Bus (or ISA bus).

The BIOS defines the PC/AT interrupt line to be used by each device. The BIOS writes to each of the two cascaded 8259A PIC chips an 8-bit vector which maps each IRQx to its corresponding interrupt vector in memory.

ISA Device Interrupt Map

The VMICPCI-7697 BIOS maps the IRQx lines to the appropriate device per the standard ISA architecture. Reference Figure E-2 on page 136. This initialization operation cannot be changed; however, a custom application could reroute the interrupt configuration after the BIOS has completed the initial configuration cycle.

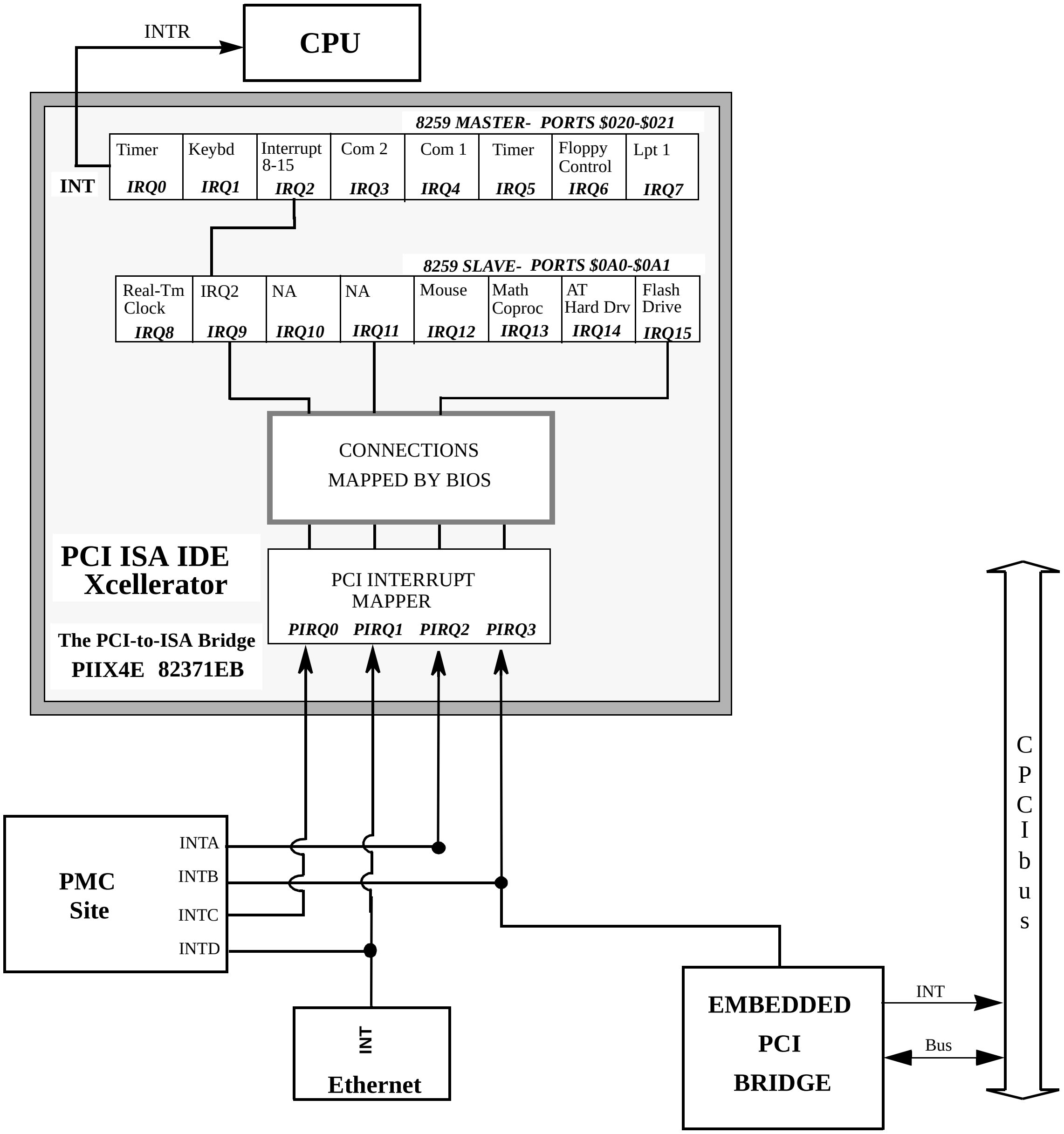


Figure E-2 BIOS Default Connections for the PC Interrupt Logic Controller

NOTE: See Table B-3 for the PCI Device Interrupt Mapping by the BIOS.

PCI Device Interrupt Map

The PCI bus-based external devices include the PMC site, Ethernet Controller, and the PCI-to-PCI bridge. The default BIOS maps these external devices to the PCI Interrupt Request (PIRQx) lines of the PIIX4. This mapping is illustrated in Figure E-2 on page 136 and is defined in Table E-3.

The device PCI interrupt lines (INTA through INTD) that are present on each device *cannot* be modified.

Table E-3 PCI Device Interrupt Mapping by the BIOS

DEVICE	COMPONENT	VENDOR ID	DEVICE ID	CPU ADDRESS MAP ID SELECT	DEVICE PCI INTERRUPT	MOTHER-BOARD PCI INTERRUPT MAPPER	DATA BOOK REF. #	Arbitration Level
Power Management	PIIX4 82371EB Function	0x8086	0x7113	AD18	N / A	N / A	5	N / A
PCI-to-ISA Bridge	PIIX4 82371EB Function 00	0x8086	0x7000	AD18	N / A	N / A	5	N / A
Ethernet Controller	DEC 21143	0x1011	0x0019	AD22	INTA	PIRQ1	4	REQ1
PCI IDE Controller	PIIX4 82371EB Function 01	0x8086	0x7111	AD18	N / A	N / A	5	N / A
PCI Host Bridge	Intel 440BX	0x8086	0x7190	N / A	N / A	N / A	2	N / A
PMC	N / A	N / A	N / A	AD31	INTA	PIRQ2	14	REQ3
Compact PCI Bridge	DEC 21554	0x1011	0x0046	AD26	INTA	PIRQ3	3	REQ0

****** Not connected, for reference only
******* PIRQ4 interrupt is not enabled by the BIOS.

The motherboard accepts these PCI device interrupts through the PCI interrupt mapper function. The BIOS default maps the PCI Interrupt Request (PIRQx) external device lines to one of the available slave PIC Interrupt Request lines, IRQ (9, 10, 11, or 15). The BIOS default mapping of the PIRQx to the slave PIC is defined in Table E-4.

Table E-4 Default PIRQx to IRQx BIOS Mapping with all Devices Loaded

PCI INTx	PIC IRQx
PIRQ0	IRQ11
PIRQ1	IRQ10
PIRQ2	IRQ9
PIRQ3	IRQ9

Using the interrupt steering registers of the 82371EB PIIX4E, the user can override the BIOS defaults and map any of the PCI interrupts (PIRQ0-3) to any of the following PIC IRQx (ISA) interrupts: IRQ15, 14, 12-9, or 7-3.

CAUTION: If PCI interrupts are remapped by the user, care must be taken to ensure that all ISA and PCI functions that require an interrupt are included.

Appendix

F

Sample C Software

Contents

Introduction

This appendix provides listings of a library of sample code that the programmer may utilize to build applications. These files are provided to the VMICPCI-7697 user on disk 320-500040-007, Sample Application C Code for the VMICPCI-7697, included in the distribution disk set.

Because of the wide variety of environments in which the VMICPCI-7697 operates, the samples provided in this appendix are not necessarily intended to be verbatim boilerplates. Rather, they are intended to give the end user an example of the standard structure of the operating code.

Directory SRAM

**File: T_SRAM.C

```
/****************************************************************************/
/* FILE: T_SRAM.C                                                           */
/*                                                                          */
/* Test battery backed SRAM with patterns and data=address.                 */
/*                                                                          */
/*                                                                          */
/****************************************************************************/
#include    <stdlib.h>
#include    <stdio.h>
#include    <dos.h>
unsigned char far * b_ptr;
unsigned int  far * w_ptr;
unsigned long far * l_ptr;
unsigned int  far * buf_ptr;
static unsigned long pat[4] = {
  0x55555555,
  0xCCCCCCCC,
  0x66666666,
  0xFFFFFFFF
};
void main( void ) {
unsigned long i, x;
unsigned char bdat;
unsigned char brd;
unsigned int  wdat;
unsigned int  wrd;
unsigned long ldat;
unsigned long lrd;
  printf("\nTesting 32K SRAM DATA/~DATA B/W/L/ADDR ............");
  buf_ptr = (unsigned int far *) MK_FP( 0xD800, 0x18 );
  /* fill and test buf with DATA/~DATA BYTES 4 patterns */
  for( x = 0; x < 4; x++ ) {
    b_ptr = (unsigned char far *) buf_ptr;
    bdat = (unsigned char) pat[x];
    for( i = 0x18; i < 0x8000; i++ ) {
      *b_ptr++ = bdat;
      bdat = ~bdat;
    }
    b_ptr = (unsigned char far *) buf_ptr;
    bdat = (unsigned char) pat[x];
    for( i = 0x18; i < 0x8000; i++ ) {
      brd = *b_ptr++;
      if( bdat != brd ) {
        printf("FAILED\nBYTE DATA @ ADDR: %Fp  WR: %.2X  RD: %.2X\n",
                --b_ptr, bdat, brd );
        exit( 1 );
      }
      bdat = (~bdat) & 0xFF;
    }
  }
  /* fill and test buf with DATA/~DATA WORDS 4 patterns */
  for( x = 0; x < 4; x++ ) {
    w_ptr = (unsigned int far *) buf_ptr;
    wdat = (unsigned int) pat[x];
    for( i = 0x18; i < 0x8000; i+=2 ) {
      *w_ptr++ = wdat;
      wdat = ~wdat;
    }
    w_ptr = (unsigned int far *) buf_ptr;
    wdat = (unsigned int) pat[x];
    for( i = 0x18; i < 0x8000; i+=2 ) {
      wrd = *w_ptr++;
      if( wdat != wrd ) {
        printf("FAILED\nWORD DATA @ ADDR: %Fp  WR: %.4X  RD: %.4X\n",
                --w_ptr, wdat, wrd );
        exit( 1 );
      }
      wdat = ~wdat;
    }
  }
  /* fill and test buf with DATA/~DATA LONGS 4 patterns */
  for( x = 0; x < 4; x++ ) {
    l_ptr = (unsigned long far *) MK_FP( 0xD800, 0x18 ); ;
    ldat = (unsigned long) pat[x];
    for( i = 0x18; i < 0x8000; i+=4 ) {
      *l_ptr++ = ldat;
      ldat = ~ldat;
    }
    l_ptr = (unsigned long far *) MK_FP( 0xD800, 0x18 );
    ldat = (unsigned long) pat[x];
    for( i = 0x18; i < 0x8000; i+=4 ) {
      lrd = *l_ptr++;
      if( ldat != lrd ) {
        printf("FAILED\nLONG DATA @ ADDR: %Fp  WR: %.8X  RD: %.8X\n",
                --l_ptr, ldat, lrd );
        exit( 1 );
      }
      ldat = ~ldat;
    }
  }
  /* fill and test buf with DATA = ADD LONG */
  for( x = 0; x < 4; x++ ) {
    l_ptr = (unsigned long far *) MK_FP( 0xD800, 0x18 );
    for( i = 0x18; i < 0x8000; i+=4 ) {
      *l_ptr++ = i;
    }
    l_ptr = (unsigned long far *) MK_FP( 0xD800, 0x18 );
    for( i = 0x18; i < 0x8000; i+=4 ) {
      lrd = *l_ptr++;
      if( lrd != i ) {
        printf("FAILED\nDATA=ADDR @ ADDR: %Fp  WR: %.8X  RD: %.8X\n",
                --l_ptr, i, lrd );
        exit( 1 );
      }
    }
  }
  printf("PASSED\n\n");
  exit( 0 );
} /* end main */
```

Directory Timers

This directory contains sample code useful in the creation of applications involving the VMICPCI-7697's three software controlled 16-bit timers. The code is written for the control of a single timer, but can be utilized in generating code for any timer configuration. The timers are described in Chapter 4 of the manual.

**File: CPU.H

```
/*
**  FILE: T7697T.H
**
*/
#define     DID_PWR_MGM     0x7113
#define     VID_PWR_MGM     0x8086
#define     IRQ5            0x0D
#define     GPI_T1          0x80    /* PIX General Purpose Input 15 (tmr 1)      */
#define     GPI_T2          0x40    /* PIX General Purpose Input 14 (tmr 2)      */
#define     GPI_T3          0x20    /* PIX General Purpose Input 13 (tmr 3)      */
#define     GPO_T1          0x40    /* PIX General Purpose Output 30 (tmr 1)     */
#define     GPO_T2          0x10    /* PIX General Purpose Output 28 (tmr 2)     */
#define     GPO_T3          0x08    /* PIX General Purpose Output 27 (tmr 3)     */
#define     GPO_CLR         0xA7    /* PIX General Purpose Output CLR TMRS       */
#define     TIMER_CNTR1     0x00    /* Timer counter 1 offset                    */
#define     TIMER_CNTR2     0x01    /* Timer counter 2 offset                    */
#define     TIMER_CNTR3     0x02    /* Timer counter 3 offset                    */
#define     TIMER_CNTL      0x03    /* Timer control offset                      */
/*****************************************************************************/
/* 8254 Control word                                                         */
/*****************************************************************************/
#define     CW_SC0          0x00    /* W Selcect counter 0                    */
#define     CW_SC1          0x40    /* W Selcect counter 1                    */
#define     CW_SC2          0x80    /* W Selcect counter 2                    */
#define     CW_RBC          0xC0    /* W Read back command                    */
#define     CW_CLC          0x00    /* W Cntr latch command (cnt/stat)        */
#define     CW_SLC          0x00    /* W Status latch command           */
#define     CW_LSB          0x10    /* W LSB only                             */
#define     CW_MSB          0x20    /* W MSB only                             */
#define     CW_LSBMSB       0x30    /* W LSB first then MSB             */
#define     CW_M0           0x00    /* W Mode 0                               */
#define     CW_M1           0x02    /* W Mode 1                               */
#define     CW_M2           0x04    /* W Mode 2                               */
#define     CW_M3           0x06    /* W Mode 3                               */
#define     CW_M4           0x08    /* W Mode 4                               */
#define     CW_M5           0x0A    /* W Mode 5                               */
#define     CW_BCD          0x01    /* W Binary Coded Decimal           */
```

```
#define     CW_RB_CNT       0x00    /* W Read back count                     */
#define     CW_RB_STAT      0x00    /* W Read back status              */
#define     CW_RB_C0        0x02    /* W Read back counter 0           */
#define     CW_RB_C1        0x04    /* W Read back counter 1           */
#define     CW_RB_C2        0x08    /* W Read back counter 2           */
```

** File: PCI.H

```
define TRUE   1
#define FALSE 0

#define CARRY_FLAG               0x01

/* PCI Functions */
#define PCI_FUNCTION_ID          0xB1
#define PCI_BIOS_PRESENT         0x01
#define FIND_PCI_DEVICE          0x02
#define FIND_PCI_CLASS_CODE      0x03
#define READ_CONFIG_BYTE         0x08
#define READ_CONFIG_WORD         0x09
#define READ_CONFIG_DWORD        0x0A
#define WRITE_CONFIG_BYTE        0x0B
#define WRITE_CONFIG_WORD        0x0C
#define WRITE_CONFIG_DWORD       0x0D

/* PCI Return codes */

#define SUCCESSFUL               0x00
#define NOT_SUCCESSFUL           0x01

/* PCI Config Space Regs */
#define PCI_CS_VENDOR_ID
#define PCI_CS_DEVICE_ID
#define PCI_CS_COMMAND
#define PCI_CS_STATUS
#define PCI_CS_REVISION_ID
#define PCI_CS_CLASS_CODE
#define PCI_CS_BASE_ADDRESS_0
#define PCI_CS_BASE_ADDRESS_1
#define PCI_CS_BASE_ADDRESS_2
#define PCI_CS_BASE_ADDRESS_3
#define PCI_CS_BASE_ADDRESS_4
#define PCI_CS_BASE_ADDRESS_5
#define PCI_CS_INTERRUPT_LINE
#define PCI_CS_INTERRUPT_PIN

/* Prototypes */

int find_pci_device(unsigned short device_id,
                    unsigned short vendor_id,
                    unsigned short index,
                    unsigned char *bus_number,
                    unsigned char *device_and_function);

int read_configuration_area(unsigned char function,
                            unsigned char bus_number,
                            unsigned char device_and_function,
                            unsigned char register_number,
                            unsigned long *data);

int write_configuration_area(unsigned char function,
                             unsigned char bus_number,
                             unsigned char device_and_function,
                             unsigned char register_number,
                             unsigned long value);

void outpd(unsigned short port, unsigned long value);

unsigned long inpd(unsigned short port);
```

**File: PCI.C

```
#include <dos.h>
#include <stddef.h>
#include "pci.h"

#define HIGH_BYTE(ax) (ax >> 8)
#define LOW_BYTE(ax) (ax & 0xff)

int find_pci_device(unsigned short device_id,
            unsigned short vendor_id,
            unsigned short index,
            unsigned char *bus_number,
            unsigned char *device_and_function)
{
   int ret_status;
   unsigned short ax, bx, flags;
   _CX = device_id;
   _DX = vendor_id;
   _SI = index;
   _AH = PCI_FUNCTION_ID;
   _AL = FIND_PCI_DEVICE;

   geninterrupt(0x1a);

   ax = _AX;
   bx = _BX;
   flags = _FLAGS;

   if ((flags & CARRY_FLAG) == 0)
   {
     ret_status = HIGH_BYTE(ax);
     if (ret_status == SUCCESSFUL)
     {
       if (bus_number != NULL) *bus_number = HIGH_BYTE(bx);
       if (device_and_function != NULL) *device_and_function = LOW_BYTE(bx);
     }
   }
   else
   {
     ret_status = NOT_SUCCESSFUL;
   }
   return(ret_status);
}
```

```
int read_configuration_area(unsigned char function,
                            unsigned char bus_number,
                            unsigned char device_and_function,
                            unsigned char register_number,
                            unsigned long *data)
{
   int ret_status;
   unsigned short ax, flags;
   unsigned long ecx;

   _BH = bus_number;
   _BL = device_and_function;
   _DI = register_number;
   _AH = PCI_FUNCTION_ID;
   _AL = function;

   geninterrupt(0x1a);

   ecx = _ECX;
   ax = _AX;
   flags = _FLAGS;
   if ((flags & CARRY_FLAG) == 0)
   {
     ret_status = HIGH_BYTE(ax);
     if (ret_status == SUCCESSFUL)
     {
       *data = ecx;
     }
   }
   else
   {
     ret_status = NOT_SUCCESSFUL;
   }
   return(ret_status);
}

int write_configuration_area(unsigned char function,
                    unsigned char bus_number,
                    unsigned char device_and_function,
                    unsigned char register_number,
                    unsigned long value)
{
   int ret_status;
   unsigned short ax, flags;
   _BH = bus_number;
   _BL = device_and_function;
   _ECX = value;
   _DI = register_number;
   _AH = PCI_FUNCTION_ID;
   _AL = function;

   geninterrupt(0x1a);

   ax = _AX;
   flags = _FLAGS;
   if ((flags & CARRY_FLAG) == 0)
   {
     ret_status = HIGH_BYTE(ax);
   }
   else
   {
     ret_status = NOT_SUCCESSFUL;
   }
   return(ret_status);
}
void outpd(unsigned short port, unsigned long value)
{
   _DX = port;
   _EAX = value;
   __emit__(0x66, 0xEF);
}
unsigned long inpd(unsigned short port)
{
   _DX = port;
   __emit__(0x66, 0xED);
   return(_EAX);
}
/*
```

**File: T_Timers.C

```
*****************************************************************************/
/*  FILE:   T_TIMERS.C                                                      */
/*          Test Timers (TIC is jumper selectable for 500 ns or 1 us)       */
/*                                                                          */
/****************************************************************************/
#include  <stdlib.h>
#include  <stdio.h>
#include  <string.h>
#include  <conio.h>
#include  <ctype.h>
#include  <dos.h>
#include  "pci.h"
#include  "cpu.h"
/* TT.C function prototypes */
void do_exit( int );
/* TIMERS.C function prototypes */
void far interrupt irq_rcvd( void );
void init_timer_int( void );
void restore_orig_int( void );
void load_counter( int, unsigned int );
void read_counter( int, unsigned int *, unsigned char * );
/* global variables */
unsigned char bus, dev_func;
/* the following globals are used in other files as ’extern’ variables */
unsigned char tmr_status, t1_stat, t2_stat, t3_stat;
unsigned int tmr_cnt, t1_cnt, t2_cnt, t3_cnt;
unsigned long t1_count;      /* counts no. of times timer 1 ISR entered */
unsigned long t2_count;      /* counts no. of times timer 2 ISR entered */
unsigned long t3_count;      /* counts no. of times timer 3 ISR entered */
unsigned int pwr_mgm_base;
unsigned int gpi_base;
unsigned int gpo_base;
unsigned int timer_base;
unsigned char pic1_org;
unsigned char gpo_org;
void main( int argc, char * argv[] )
{
  unsigned long t1, t2, t3;
  int test_int;
  unsigned long temp_dword;
  char user[80];
  timer_base = 0x500;
```

```
 /* try to locate the power management device on the PCI bus */
 test_int = find_pci_device(DID_PWR_MGM, VID_PWR_MGM, 0,
                &bus, &dev_func);
 if(test_int != SUCCESSFUL)
 {
    printf("\nUnable to locate power management device on PCI bus\n");
    do_exit( 1 );
 }
 /* get base address from config area */
 test_int = read_configuration_area(READ_CONFIG_DWORD,
                                    bus, dev_func, 0x40, &temp_dword);
 if(test_int != SUCCESSFUL)
 {
  printf("\nUnable to read POWER MGM. BASE ADDRESS @ 0x40 in config space\n");
    do_exit( 1 );
 }
 pwr_mgm_base = temp_dword & 0x0000FFC0;
 gpi_base = pwr_mgm_base + 0x31; /* PIX general purpose input bits 8-15 */
 gpo_base = pwr_mgm_base + 0x37; /* PIX general purpose output bits 24-31*/
 disable();
/* disable timers by reloading the control word */
 outp( timer_base + TIMER_CNTL, (CW_SC0 | CW_LSBMSB | CW_M2) );
 outp( timer_base + TIMER_CNTL, (CW_SC1 | CW_LSBMSB | CW_M2) );
 outp( timer_base + TIMER_CNTL, (CW_SC2 | CW_LSBMSB | CW_M2) );
 /* Read 8259 slave Programmable Interrupt controller */
 pic1_org = inp(0x21) & 0xFF; /* slave mask bits */
 /* disable interrupt 5 */
 outp(0x21, (pic1_org | 0x20) ); /* 0 = enable 1 = disable */
 enable();
 gpo_org = inp( gpo_base ) & 0xFF;
 /* setup timers interrupt service routine */
 init_timer_int();
 /*
 ** verify all three counters can generate an interrupt (counters 1,2,3)
 */
 printf("\nTesting all three 16 bit counters for interrupt ....");
 /* setup for interrupts to occur */
 t1_count = 0;
 t2_count = 0;
 t3_count = 0;
 t1 = 0;
 t2 = 0;
 t3 = 0;
 tmr_status = 0;
```

```
test_int = 100;
/* load counters */
load_counter( 1, 0xFFFF );
do
{
  if( t1_count ) {
    t1++;
    break;
  }
  test_int--;
  delay( 1 );
} while( test_int );
outp( timer_base + TIMER_CNTL, (CW_SC0 | CW_LSBMSB | CW_M2) );
outp( timer_base + TIMER_CNTL, (CW_SC1 | CW_LSBMSB | CW_M2) );
outp( timer_base + TIMER_CNTL, (CW_SC2 | CW_LSBMSB | CW_M2) );
 tmr_status = 0;
test_int = 100;
load_counter( 2, 0xFFFF );
do
{
  if( t2_count ) {
    t2++;
    break;
  }
  test_int--;
  delay( 1 );
} while( test_int );
/* disable timers by reloading the control word */
outp( timer_base + TIMER_CNTL, (CW_SC1 | CW_LSBMSB | CW_M2) );
tmr_status = 0;
test_int = 100;
load_counter( 3, 0xFFFF );
do
{
  if( t3_count ) {
    t3++;
    break;
  }
  test_int--;
  delay( 1 );
} while( test_int );
/* disable timers by reloading the control word */
outp( timer_base + TIMER_CNTL, (CW_SC2 | CW_LSBMSB | CW_M2) );
/* clear all three status bits in GPI */
```

```
  outp( gpo_base, ( gpo_org & GPO_CLR ) );
  /* set all three GPO outputs to 1 to allow int status registers to function */
  outp( gpo_base, ( gpo_org | GPO_T1 | GPO_T2 | GPO_T3 ) );
  if( t1 && t2 && t3 )
  {
    printf("PASSED\n");
  }
  else
  {
    printf("FAILED\n");
    if( !t1 ) printf("TIMER 1 failed\n");
    if( !t2 ) printf("TIMER 2 failed\n");
    if( !t3 ) printf("TIMER 3 failed\n");
    do_exit( 2 );
  }
  /* do orderly exit */
  do_exit( 3 );
} /* end main */
void do_exit( int xit_code )
{
  if( xit_code > 1 )  restore_orig_int();
  outp( gpo_base, gpo_org );
  if( xit_code == 3 ) xit_code = 0;
  exit( xit_code );
} /* do_exit */
```

**File: Timer.C

```
/*
**  FILE: TIMERS.C
**
*/
#include  <stdlib.h>
#include  <stdio.h>
#include  <dos.h>
#include  <ctype.h>
#include  <conio.h>
#include  "cpu.h"
/* function prototypes */
void far interrupt irq_rcvd( void );
void init_timer_int( void );
void restore_orig_int( void );
void load_counter( int, unsigned int );
void read_counter( int, unsigned int *, unsigned char * );
/* global variables */
extern unsigned long t1_count;    /* timer 1 count */
extern unsigned long t2_count;    /* timer 2 count */
extern unsigned long t3_count;    /* timer 3 count */
extern unsigned char tmr_status;
extern unsigned int gpi_base;
extern unsigned int gpo_base;
extern unsigned int timer_base;
extern unsigned char pic1_org;
extern unsigned char gpo_org;
void far interrupt (* old_vect)(void);
/*******************************************************************/
/*  init_timer_int()                                               */
/*                                                                 */
/*  purpose: Using the interrupt assigned, the original vector is  */
/*           saved and the vector to the new ISR is installed. The */
/*           programmable-interrupt-controller (PIC) is enabled.   */
/*                                                                 */
/*  Prerequisite: The interrupt line to be used must have          */
/*                already been loaded in the global variable.      */
/*                                                                 */
/*******************************************************************/
/*  parameters: none                                               */
/*******************************************************************/
/*  return value: none                                             */
/*******************************************************************/
```

```
void init_timer_int( void )
{
  disable();
  old_vect = getvect( IRQ5 ); /* save vector for IRQ5 */
  setvect( IRQ5, irq_rcvd );
  /* enable interrupt 5 */
  outp(0x21, (pic1_org & 0xDF) ); /* 0 = enable 1 = disable */
  /* clear all three GPO inputs */
  outp( gpo_base, ( gpo_org & GPO_CLR ) );
  /* set all three GPO outputs to 1 to allow int status registers to function */
  outp( gpo_base, (gpo_org | GPO_T1 | GPO_T2 | GPO_T3 ) );
  enable();
} /* init_timer_int */
/*******************************************************************/
/*  restore_orig_int()                                             */
/*                                                                 */
/*  purpose: Using the interrupt assigned, the original vector is  */
/*           restored and the programmable-interrupt-controller    */
/*           is disabled.                                          */
/*                                                                 */
/*  Prerequisite: The interrupt line to be used must have          */
/*                already been loaded in the global variable.      */
/*                                                                 */
/*******************************************************************/
/*  parameters: none                                               */
/*******************************************************************/
/*  return value: none                                             */
/*******************************************************************/
void restore_orig_int( void )
{
  disable();
  outp(0x21, pic1_org);
  setvect( IRQ5, old_vect );
  enable();
} /* restore_orig_int */
/*******************************************************************/
/*  load_counter()                                                 */
/*                                                                 */
/*  purpose: Loads the appropriate counter with the count passed   */
/*                                                                 */
/*                                                                 */
/*******************************************************************/
/*  parameters: int counter = 1, 2, 3 for COUNTER 1, 2, or 3       */
/*              unsigned int count = count to be loaded            */
/*******************************************************************/
/*  return value: none                                             */
/*******************************************************************/
```

```
void load_counter( int counter, unsigned int count )
{
  int lsb, msb;
  lsb = count & 0xff;
  msb = count >> 8;
  switch( counter )
  {
    case 1: /* select counter 1, LSB then MSB, mode 2 */
      outp( timer_base + TIMER_CNTL, (CW_SC0 | CW_LSBMSB | CW_M2) );
      outp( timer_base + TIMER_CNTR1, (unsigned char) lsb );
      outp( timer_base + TIMER_CNTR1, (unsigned char) msb );
    break;
    case 2: /* select counter 2, LSB then MSB, mode 2 */
      outp( timer_base + TIMER_CNTL, (CW_SC1 | CW_LSBMSB | CW_M2) );
      outp( timer_base + TIMER_CNTR2, (unsigned char) lsb );
      outp( timer_base + TIMER_CNTR2, (unsigned char) msb );
    break;
    case 3: /* select counter 3, LSB then MSB, mode 2 */
      outp( timer_base + TIMER_CNTL, (CW_SC2 | CW_LSBMSB | CW_M2) );
      outp( timer_base + TIMER_CNTR3, (unsigned char) lsb );
      outp( timer_base + TIMER_CNTR3, (unsigned char) msb );
    break;
  }
} /* load_counter */
/*******************************************************************/
/*  read_counter()                                                 */
/*                                                                 */
/*  purpose: Reads the appropriate counter in the appropriate      */
/*           bank with the remainin count and status.              */
/*                                                                 */
/*                                                                 */
/*******************************************************************/
/*  parameters: int counter = 1, 2, 3 for COUNTER 1, 2, or 3       */
/*              unsigned int * count = remaining count             */
/*              unsigned char * status = counter status            */
/*******************************************************************/
/*  return value: none                                             */
/*******************************************************************/
void read_counter( int counter,
                   unsigned int * count, unsigned char * status )
{
  int lsb, msb;
  switch( counter )
  {
    case 1: /* select counter 1, LSB then MSB */
      outp( timer_base + TIMER_CNTL, ( CW_RBC | CW_RB_CNT | CW_RB_STAT | CW_RB_C0 )
);
```

```
      *status = inp( timer_base + TIMER_CNTR1 ) & 0xFF;
      lsb = inp( timer_base + TIMER_CNTR1 ) & 0xFF;
      msb = inp( timer_base + TIMER_CNTR1 ) & 0xFF;
      msb = msb << 8;
      *count = ( lsb | msb );
    break;
    case 2: /* select counter 2, LSB then MSB */
      outp( timer_base + TIMER_CNTL, ( CW_RBC | CW_RB_CNT | CW_RB_STAT | CW_RB_C1 )
);
      *status = inp( timer_base + TIMER_CNTR2 ) & 0xFF;
      lsb = inp( timer_base + TIMER_CNTR2 ) & 0xFF;
      msb = inp( timer_base + TIMER_CNTR2 ) & 0xFF;
      msb = msb << 8;
      *count = ( lsb | msb );
    break;
    case 3: /* select counter 3, LSB then MSB */
      outp( timer_base + TIMER_CNTL, ( CW_RBC | CW_RB_CNT | CW_RB_STAT | CW_RB_C2 )
);
      *status = inp( timer_base + TIMER_CNTR3 ) & 0xFF;
      lsb = inp( timer_base + TIMER_CNTR3 ) & 0xFF;
      msb = inp( timer_base + TIMER_CNTR3 ) & 0xFF;
      msb = msb << 8;
      *count = ( lsb | msb );
    break;
  }
} /* read_counter */
/*******************************************************************/
/*  irq_rcvd()                                                     */
/*                                                                 */
/*  purpose: Interrupt service routine used to service any of the  */
/*           counters on the 7697.                                 */
/*                                                                 */
/*                                                                 */
/*******************************************************************/
/*  parameters: none                                               */
/*******************************************************************/
/*  return value: none                                             */
/*******************************************************************/
void interrupt irq_rcvd(void)
{
  disable();
  asm {
    .386P
    push  eax
    push  ebx
  }
  tmr_status = inp( gpi_base ) & 0xFF;
  /* increment counts and clear status */
  if( tmr_status & GPI_T1 ) {
    t1_count++;
    outp( gpo_base, (gpo_org & (~GPO_T1)) );  /* clear timer 1 status bit */
  }
  if( tmr_status & GPI_T2 ) {
    t2_count++;
    outp( gpo_base, (gpo_org & (~GPO_T2)) );  /* clear timer 2 status bit */
  }
  if( tmr_status & GPI_T3 ) {
    t3_count++;
    outp( gpo_base, (gpo_org & (~GPO_T3)) );  /* clear timer 3 status bit */
  }
  outp( gpo_base, (gpo_org | GPO_T1 | GPO_T2 | GPO_T3) ); /* enable status */
  /* Non specific end of interrupt to PIC */
  outp(0x20, 0x20);  /* Master end of irq command */
  asm {
    .386P
    pop  ebx
    pop  eax
  }
  enable();
}
```

Directory WATCHDOG

This directory contains sample code useful in the creation of applications involving the VMICPCI-7697’s watchdog timer function as described in Chapter 4.

**File: Watchdog.H

```
/*
** DS1384 REGISTER OFFSETS
*/
                                  /* 7  6  5  4  3  2  1  0             */
#define      CLK_MSEC    0x00     /* 00-99                              */
#define      CLK_SEC     0x01     /* 00-59 0                            */
#define      CLK_MIN     0x02     /* 00-59 0                            */
#define      CLK_MINAL   0x03     /* 00-59 M                            */
#define      CLK_HRS     0x04     /* 01-12+A/P OR 00-23                 */
#define      CLK_HRSAL   0x05     /* 01-12+A/P OR 00-23                 */
#define      CLK_DAY     0x06     /* 01-07 0  0  0  0  0                */
#define      CLK_DAYAL   0x07     /* 01-07 M  0  0  0  0                */
#define      CLK_DATE    0x08     /* 01-31 0  0                         */
#define      CLK_MONTH   0x09     /* 01-12       0                      */
#define      CLK_YRS     0x0A    /* 00-99                               */
#define      WD_CMD      0x0B    /* command register                    */
#define      WD_MSEC     0x0C    /* milli second watchdog time          */
#define      WD_SEC      0x0D    /* seconds watchdog time               */

/*
** DS1384 COMMAND REGSITER BIT DEFINITIONS
*/

#define      WD_TE       0x80     /* transfer enable 1 - allow updates  */
#define      WD_IPSW     0x40    /* interrupt switch 0 - WD out INTA    */
#define      WD_IBHL     0x20    /* int. B output 0 - current sink      */
#define      WD_PU       0x10    /* pulse/level 1 - 3 ms pulse          */
#define      WD_WAM      0x08    /* watchdog alarm mask 0 - active      */
#define      WD_TDM      0x04    /* time-of-day alarm mask 0 - active   */
#define      WD_WAF      0x02     /* watchdog alram flag                */
#define      WD_TDF      0x01    /* time-of-day flag                    */
```

**File: WDT0_RST.C

```
/****************************************************************************/
/* FILE: WDTO_RST.C                                                         */
/*                                                                          */
/* Setup watchdog to issue reset on time out.                               */
/*                                                                          */
/*                                                                          */
/****************************************************************************/
#include    <stdlib.h>
#include    <stdio.h>
#include    <dos.h>
#include    <time.h>
#include    <conio.h>
#include    <ctype.h>
#include    "watchdog.h"
unsigned char far * wd_ptr;
time_t t;
char usr[80];
char reg_b;
void main( void ) {
  /* Install Jumper E6 */
  wd_ptr = (unsigned char far *) MK_FP( 0xD800, 0 );
  /* set WatchDog Alarm Mask 1 - deactivated and update with 0 time */
  *(wd_ptr + WD_CMD) = ( WD_TE | WD_WAM );
  *(wd_ptr + WD_MSEC) = 0; /* load with 0 to disable */
  *(wd_ptr + WD_SEC) = 0;  /* load with 0 to disable */
  *(wd_ptr + WD_CMD) = ( WD_TE | WD_WAM ); /* allow update with 0 time */
  *(wd_ptr + WD_CMD) = WD_WAM; /* set watchdog alarm mask to 1 */
  *(wd_ptr + WD_MSEC) = 0x99; /* 00.99 seconds */
  *(wd_ptr + WD_SEC) = 0x99;  /* 99.00 seconds */
  *(wd_ptr + WD_CMD) = ( WD_TE | WD_PU ); /* set for 3 ms pulse */
  printf("Reset time out in 99.99 seconds\n\n");
  time(&t);
  printf("START DATE & TIME: %.24s\n\n", ctime(&t) );
  do {
    time(&t);
    printf("CURRENT DATE & TIME: %.24s\r", ctime(&t) );
    delay( 250 );
  } while( !kbhit() );
} /* end main */
```

Index