VMICPCI-7696

Single Board Pentium II® Embedded Module
Processor-Based System Slot CompactPCI® CPU

Product Manual
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


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.
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.
Table of Contents

Overview .................................................................................................................. 15
  Organization of the Manual .................................................................................... 16
  References ................................................................................................................ 17
  Safety Summary ...................................................................................................... 20
  Safety Symbols Used in This Manual ..................................................................... 21

Chapter 1 - Features and Options ........................................................................... 23
  CompactPCI® Features ........................................................................................... 26
  VMICPCI-7696 Product Options ........................................................................... 27

Chapter 2 - Installation and Setup ............................................................................ 29
  Unpacking Procedures ........................................................................................... 30
  Hardware Setup ....................................................................................................... 31
  Installation ................................................................................................................ 36
    BIOS Setup ............................................................................................................ 37
    Front Panel Connectors ....................................................................................... 37
    PMC Site Connector .............................................................................................. 38
    LED Definition ...................................................................................................... 39

Chapter 3 - PC/AT Functions .................................................................................... 41
  CPU Socket ............................................................................................................. 42
  Physical Memory ...................................................................................................... 42
  I/O Port Map ............................................................................................................ 43
    PCI-to-PCI Bridge ................................................................................................. 45
    PC/AT Interrupts .................................................................................................... 45
    PCI Interrupts ........................................................................................................ 48
    Integrated Peripherals ........................................................................................... 51
  Ethernet Controller ................................................................................................. 52
Chapter 5 - Maintenance

Maintenance Prints .................................................. 79
J1 Connector Pinout .................................................. 83
J2 Connector and Pinout ............................................. 84
J3 Connector Pinout .................................................. 85
J5 Connector and Pinout ............................................. 86
Ethernet Connector Pinout (J12) .................................. 87
Video Connector Pinout (J11) ...................................... 88
## Table of Contents

- Parallel Port Connector Pinout (P2) .............................................. 89
- Keyboard and Mouse Connectors and Pinout (J10) .......................... 90
- USB Connector (J9) ........................................................................ 91
- PMC Connector Pinout ................................................................. 92
  - PMC J7 Connector and Pinout ..................................................... 92
  - PMC J6 Connector and Pinout ..................................................... 93
  - PMC J4 Connector and Pinout ..................................................... 94
- Driver Software Installation ............................................................. 95
- Windows for Workgroups (Version 3.11) ....................................... 96
- Windows 95 ................................................................................. 98
- Windows 95 INF Update Utility for Intel(TM) Chipsets ..................... 100
- Bus Master IDE Driver for Windows 95 .......................................... 101
- Windows NT (Version 4.0) ............................................................. 102
- System BIOS Setup Utility ............................................................ 106
- BIOS Features Setup .................................................................... 107
- LANWorks BIOS Setup ................................................................ 108
  - Boot Protocol ........................................................................... 108
  - Default Boot ............................................................................ 108
  - Local Boot .............................................................................. 108
- System BIOS Setup Utility ............................................................. 110
- Standard CMOS Setup .................................................................. 111
  - Setting The Date ...................................................................... 111
  - Setting The Time ...................................................................... 111
  - Primary Master/Slave ............................................................... 111
  - Secondary Master/Slave ............................................................ 112
  - Floppy Disk Drive ................................................................. 112
    - Floppy Drive A .................................................................... 112
    - Floppy Drive B .................................................................... 112
  - Video ....................................................................................... 112
  - Halt On ................................................................................... 112
  - Memory .................................................................................. 113
- BIOS Features Setup .................................................................... 114
  - Virus Warning .......................................................................... 114
  - CPU Internal Cache ................................................................... 114
  - External Cache ......................................................................... 115
  - CPU L2 Cache ECC Checking .................................................... 115
  - Quick Power On Self Test ......................................................... 115
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modem Use IRQ</td>
<td>122</td>
</tr>
<tr>
<td>Doze Mode</td>
<td>122</td>
</tr>
<tr>
<td>Standby Mode</td>
<td>122</td>
</tr>
<tr>
<td>Suspend Mode</td>
<td>122</td>
</tr>
<tr>
<td>HDD Power Down</td>
<td>122</td>
</tr>
<tr>
<td>Throttle Duty Cycle</td>
<td>123</td>
</tr>
<tr>
<td>VGA Active Monitor</td>
<td>123</td>
</tr>
<tr>
<td>Resume By Ring</td>
<td>123</td>
</tr>
<tr>
<td>Resume By Alarm</td>
<td>123</td>
</tr>
<tr>
<td>Wake Up On LAN</td>
<td>123</td>
</tr>
<tr>
<td>CPU Clock Throttle</td>
<td>123</td>
</tr>
<tr>
<td>IRQ 8 Break Suspend</td>
<td>123</td>
</tr>
<tr>
<td>Reload Global Timer Events</td>
<td>123</td>
</tr>
<tr>
<td>PnP/PCI Configuration</td>
<td>124</td>
</tr>
<tr>
<td>PNP OS Installed</td>
<td>124</td>
</tr>
<tr>
<td>Resources Controlled By</td>
<td>124</td>
</tr>
<tr>
<td>Reset Configuration Data</td>
<td>124</td>
</tr>
<tr>
<td>IRQ n Assigned to</td>
<td>125</td>
</tr>
<tr>
<td>DMA n Assigned to</td>
<td>125</td>
</tr>
<tr>
<td>Used Mem Base Addr</td>
<td>125</td>
</tr>
<tr>
<td>Integrated Peripherals</td>
<td>126</td>
</tr>
<tr>
<td>IDE HDD Block Mode</td>
<td>126</td>
</tr>
<tr>
<td>IDE Primary Master PIO</td>
<td>126</td>
</tr>
<tr>
<td>IDE Primary Slave PIO</td>
<td>126</td>
</tr>
<tr>
<td>IDE Primary/Secondary Master UDMA</td>
<td>127</td>
</tr>
<tr>
<td>IDE Primary/Secondary Slave UDMA</td>
<td>127</td>
</tr>
<tr>
<td>On-Chip Primary PCI IDE</td>
<td>127</td>
</tr>
<tr>
<td>On-Chip Secondary PCI IDE</td>
<td>127</td>
</tr>
<tr>
<td>Onboard PCI SCSI Chip</td>
<td>127</td>
</tr>
<tr>
<td>USB Keyboard Support</td>
<td>127</td>
</tr>
<tr>
<td>Init Display First</td>
<td>127</td>
</tr>
<tr>
<td>Onboard FDC Controller</td>
<td>127</td>
</tr>
<tr>
<td>Onboard Serial Port 1/2</td>
<td>128</td>
</tr>
<tr>
<td>UART 2 Mode</td>
<td>128</td>
</tr>
<tr>
<td>Onboard Parallel Port</td>
<td>128</td>
</tr>
<tr>
<td>Parallel Port Mode</td>
<td>128</td>
</tr>
<tr>
<td>BIOS Operations</td>
<td>130</td>
</tr>
<tr>
<td>BIOS Control Overview</td>
<td>130</td>
</tr>
</tbody>
</table>
# Functional Overview

---

# Data Book References

---

# Device Address Definition

---

# ISA Devices

---

# PCI Devices

---

# Device Interrupt Definition

---

# PC/AT Interrupt Definition

---

# ISA Device Interrupt Map

---

# PCI Device Interrupt Map

---

# Directory SRAM

---

**File:** T_SRAM.C

---

# Directory Timers

---

**File:** CPU.H

---

**File:** PCI.H

---

**File:** PCI.C

---

**File:** T_Timers.C

---

**File:** Timer.C

---

# Directory WATCHDOG

---

**File:** Watchdog.H

---

**File:** WDT0_RST.C

---
List of Figures

Figure 1-1  VMICPCI-7696 Block Diagram .................................................. 25
Figure 2-1  VMICPCI-7696 Embedded Module, and Jumper Locations ................. 32
Figure 2-2  Installing a PMC Card on the VMICPCI-7696 ............................ 38
Figure 2-3  Front Panel LED Positions ...................................................... 39
Figure 3-1  Connections for the PC Interrupt Logic Controller ....................... 50
Figure 4-1  Timer Interrupt Status Register Read/Steps .............................. 56
Figure 4-2  Timer Interrupt Status Register ............................................. 57
Figure 4-3  Clearing the Timer Interrupt Status Register ............................. 57
Figure 4-4  82C54 Diagram ................................................................. 58
Figure 4-5  Internal Timer Diagram ......................................................... 59
Figure 4-6  Typical System Configuration ............................................... 65
Figure 4-7  Watchdog Alarm Block ......................................................... 71
Figure A-1  VMICPCI-7696 Connector and Jumper Locations ....................... 82
Figure A-2  J1 Connector and Pinout ..................................................... 83
Figure A-3  J2 Connector and Pinout ..................................................... 84
Figure A-4  J3 Connector and Pinout ..................................................... 85
Figure A-5  J5 Connector and Pinout ..................................................... 86
Figure A-6: Ethernet Connector and Pinout ............................................. 87
Figure A-7  Video Connector and Pinout ............................................... 88
Figure A-8  P2 Connector and Pinout ..................................................... 89
Figure A-9  Keyboard/Mouse Connector and Pinout .................................. 90
Figure A-10 USB Connector Pinout ...................................................... 91
Figure E-1  VMICPCI-7696 Block Diagram ............................................. 131
Figure E-2  BIOS Default Connections for the PC Interrupt Logic Controller ....... 138
# List of Tables

| Table 1-1 | PC/AT I/O Features | 24 |
| Table 2-1 | CPU Board Connectors | 33 |
| Table 2-2 | Password Clear (User Configurable) - Jumper (E3) | 33 |
| Table 2-3 | Factory Configured - AGP Enable Jumper E4 (Not Populated) | 34 |
| Table 2-4 | Factory Configured - Host Frequency Jumper E5 (Not Populated) | 34 |
| Table 2-5 | Factory Configured - PIIX4E Battery Jumper (E6) | 34 |
| Table 2-6 | Factory Configured - Watchdog Timer jumper (E7) | 34 |
| Table 2-7 | Factory Configured - Clock Speed Select Jumper (E8) | 34 |
| Table 2-8 | Factory Configured - SRAM Battery Jumper (E10) | 35 |
| Table 2-9 | Programmable Timer Clock Select (User Configurable) Jumper (E11) | 35 |
| Table 2-10 | Factory Configured - Reserved Jumper (E12) | 35 |
| Table 2-11 | Factory Configured - DiskOnChip Address SEL Jumper (E13) | 35 |
| Table 2-12 | Factory Configured - Boot Block Lock Jumper (E17) | 35 |
| Table 2-13 | Factory Configured - Boot Block Lock (E18) | 35 |
| Table 3-1 | VMICPCI-7696 I/O Address Map | 43 |
| Table 3-2 | PC/AT Hardware Interrupt Line Assignments | 45 |
| Table 3-3 | PC/AT Interrupt Vector Table | 46 |
| Table 3-4 | NMI Register Bit Descriptions | 49 |
| Table 3-5 | Supported Graphics Video Resolutions | 53 |
| Table 4-1 | I/O Address of the Control Word Register and Timers | 57 |
| Table 4-2 | Control Word Format | 60 |
| Table 4-3 | ST - Select Timer | 60 |
| Table 4-4 | RW - Read/Write | 60 |
Table 4-5 M - Mode .......................................................... 61
Table 4-6 BCD .............................................................. 61
Table 4-7 Read-Back Command Format ................................ 62
Table 4-8 Read-Back Command Description ............................ 62
Table 4-9 Status Byte ..................................................... 63
Table 4-10 Status Byte Description ..................................... 63
Table 4-11 LOAD Bit Operation .......................................... 63
Table 4-12 Watchdog Registers .......................................... 72
Table 4-13 Time of Day Alarm Registers ................................. 74
Table A-14 Keyboard/Mouse Y Splitter Cable ......................... 90
Table A-1 PMC J7 Connector Pinout ................................... 92
Table A-2 PMC J6 Connector Pinout ................................... 93
Table A-3 PMC J4 Connector Pinout ................................... 94
Table E-1 ISA Device Mapping Configuration ......................... 135
Table E-2 PCI Device Mapping Configuration ......................... 136
Table E-3 PCI Device Interrupt Mapping by the BIOS ............... 139
Table E-4 Default PIRQx to IRQx BIOS Mapping with all Devices Loaded ................................. 140
Overview

Introduction

VMIC’s VMICPCI-7696 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-7696 with advanced CPCI interface and SDRAM that is dual-ported to the CPCI bus, is ideal for CPCI system controller 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, Windows NT, or any other PC/AT-compatible operating system. The PC/AT mode of the VMICPCI-7696 is discussed in Chapter 3 of this manual.

The VMICPCI-7696 also operates as a CPCI peripheral slot CPU and interacts with other CPCI modules via the on-board embedded bridge.

The VMICPCI-7696 programmer may quickly and easily control CPCI 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-7696 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-7696 Features and Options describes the features of the base unit followed by descriptions of the associated features of the unit in operation on a CPCI bus.

Chapter 2 - Installation and Setup describes unpacking, inspection, hardware jumper settings, connector definitions, installation, system setup, and operation of the VMICPCI-7696.

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

Appendix C - Basic Input/Output System describes the menus and options associated with the Award (system) BIOS.

Appendix D - 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.
For the most up-to-date specifications for the VMICPCI-7696, please refer to:

**VMIC specification number 800-657602-000**

**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

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 Balfentine 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-7696:

**Microsoft Windows NT Server Resource Kit**  
Microsoft Corporation  
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).

The STOP symbol informs the operator 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.

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.

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.

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

Contents

CompactPCI® Features .................................................. 26
VMICPCI-7696 Product Options ........................................ 27

Introduction

The VMICPCI-7696 performs all the functions of a standard IBM PC/AT motherboard with the following features:

• Single-slot CPCI bus 6U Eurocard form factor
• Includes a high-performance Intel Pentium II® Embedded Module processor
• Up to 256 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 CPCI 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 M-Systems DiskOnChip flash memory (12 to 72 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 CPCI J3 connector
• One enhanced parallel port available on the front panel and through the CPCI J3 connector
• Two USB Ports: one available on the front panel, and one through the CPCI J3 connector (rear I/O)
The VMICPCI-7696 supports standard PC/AT I/O features such as those listed in Table 1-1. Figure 1-1 on page 25 shows a block diagram of the VMICPCI-7696 emphasizing the I/O features, including the PCI-to-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

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

The VMIACC-0576 CompactPCI Rear Transition Utility Board provides USB, RJ-45 serial port connectors (2), standard (DB25) parallel port connector, IDE and Floppy connectors, by way of the CompactPCI backplane Connector (J3).
Figure 1-1 VMICPCI-7696 Block Diagram
CompactPCI® Features

In addition to its PC/AT functions, the VMICPCI-7696 has the following CompactPCI features:

- Single-slot, 6U height CPCI board
- Complies fully with Revision 2.1 of the PCI Local Bus Specification
- Complies fully with Revision 1.1 of the PCI-to-PCI Bridge Architecture Specification
- Complies with Revision 2.1 of the CompactPCI Specification
- Implements delayed transactions for all PCI configuration, I/O, and memory read commands—up to three transactions simultaneously in each direction
- Allows 152 bytes of buffering (data and address) for upstream posted memory write commands and 88 bytes of buffering for downstream posted memory write commands—up to nine upstream and five downstream posted write transactions simultaneously
- Allows 152 bytes of read data buffering upstream and 72 bytes of read data buffering downstream
- Provides concurrent primary and secondary bus operation to isolate traffic
- Provides enhanced address decoding
- Includes addressing and VGA palette snooping support
- Supports PCI transaction forwarding for the following commands
  - All I/O and memory commands
  - Type 1 to Type 1 configuration commands
  - Type 1 to Type 0 configuration commands (downstream only)
  - All Type 1 to special cycle configuration commands
- Includes downstream lock support
- Supports both 5 and 3.3 V signaling environments

The VMICPCI-7696 is a versatile single-board solution for CPCI control with familiar PC/AT operation.
VMICPCI-7696 Product Options

VMIC’s VMICPCI-7696 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-7696 are defined in the device specification sheet available from your VMIC representative.
Installation and Setup

Contents

- Unpacking Procedures .................................................. 30
- Hardware Setup .......................................................... 31
- Installation .................................................................. 36

Introduction

This chapter describes the hardware jumper settings, connector descriptions, installation, system setup, and operation of the VMICPCI-7696.
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).

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-7696 is factory populated with user-specified options as part of the VMICPCI-7696 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-7696 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 32. The definitions of the CPU board jumpers and connectors are included in Table 2-1 through .

All jumpers are factory configured and should not be modified by the user. There are three exceptions: the Programmable Timer Clock Select (E11), the Password Clear (E3), and the Watchdog Timer (E7). Modifying any other jumper will void the Warranty and may damage the unit. The default jumper condition of the VMICPCI-7696 is expressed in Table 2-1 through Table 2-3 with bold text in the table cells.
Figure 2-1 VMICPCI-7696 Embedded Module, and Jumper Locations
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 for a means to clear the password feature, this might be necessary to perform 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 2-1 CPU Board Connectors

<table>
<thead>
<tr>
<th>Connector</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>J10</td>
<td>Mouse/Keyboard Connector</td>
</tr>
<tr>
<td>J9</td>
<td>USB Connector</td>
</tr>
<tr>
<td>J13</td>
<td>COM1</td>
</tr>
<tr>
<td>J3</td>
<td>COM1/COM2</td>
</tr>
<tr>
<td>J3, P2</td>
<td>Parallel Port Connector</td>
</tr>
<tr>
<td>J12</td>
<td>Ethernet Connector</td>
</tr>
<tr>
<td>J11</td>
<td>Video Connector</td>
</tr>
<tr>
<td>J3</td>
<td>Floppy Interface Connector</td>
</tr>
<tr>
<td>E15</td>
<td>ITP Test Header</td>
</tr>
<tr>
<td>E16</td>
<td>Port 80 Test Header</td>
</tr>
<tr>
<td>J3</td>
<td>EIDE Connector</td>
</tr>
<tr>
<td>E2</td>
<td>Fan Connector</td>
</tr>
<tr>
<td>J6, J7, J4</td>
<td>PMC Slot</td>
</tr>
<tr>
<td>J1, J2</td>
<td>Compact PCI Connectors</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Normal</th>
<th>Jumper Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear NVRAM/CMOS/Password</td>
<td>In</td>
</tr>
</tbody>
</table>

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 for a means to clear the password feature, this might be necessary to perform in the case of a forgotten password.
### Table 2-3  Factory Configured - AGP Enable Jumper E4 (Not Populated)

<table>
<thead>
<tr>
<th>AGP Enable</th>
<th>Jumper Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGP Enable</td>
<td>Out</td>
</tr>
<tr>
<td>AGP Disable</td>
<td>In</td>
</tr>
</tbody>
</table>

### Table 2-4  Factory Configured - Host Frequency Jumper E5 (Not Populated)

<table>
<thead>
<tr>
<th>Host Frequency</th>
<th>Jumper Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>66 Mhz</td>
<td>Out</td>
</tr>
<tr>
<td>100 Mhz</td>
<td>In</td>
</tr>
</tbody>
</table>

### Table 2-5  Factory Configured - PIIX4E Battery Jumper (E6)

<table>
<thead>
<tr>
<th>PIIX4E Battery</th>
<th>Jumper Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>In</td>
</tr>
<tr>
<td>Disabled</td>
<td>Out</td>
</tr>
</tbody>
</table>

### Table 2-6  Factory Configured - Watchdog Timer jumper (E7)

<table>
<thead>
<tr>
<th>Select</th>
<th>Jumper Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Watchdog Timer Reset</td>
<td>1-2</td>
</tr>
<tr>
<td>No Watchdog Timer Reset or SERR#</td>
<td>Out</td>
</tr>
<tr>
<td>Watchdog Timer SERR#</td>
<td>2-3</td>
</tr>
</tbody>
</table>

### Table 2-7  Factory Configured - Clock Speed Select Jumper (E8)

<table>
<thead>
<tr>
<th>CPU Bus Frequency (MHz)</th>
<th>PCI Bus Frequency (MHz)</th>
<th>E8 Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>66.6</td>
<td>33.3</td>
<td>2-3</td>
</tr>
<tr>
<td>100</td>
<td>25</td>
<td>1-2</td>
</tr>
</tbody>
</table>
### Table 2-8 Factory Configured - SRAM Battery Jumper (E10)

<table>
<thead>
<tr>
<th>SRAM Battery</th>
<th>Jumper Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>In</td>
</tr>
<tr>
<td>Disabled</td>
<td>Out</td>
</tr>
</tbody>
</table>

### Table 2-9 Programmable Timer Clock Select (User Configurable) Jumper (E11)

<table>
<thead>
<tr>
<th>Select</th>
<th>Jumper Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 MHz</td>
<td>1-2</td>
</tr>
<tr>
<td>1 MHz</td>
<td>2-3</td>
</tr>
</tbody>
</table>

### Table 2-10 Factory Configured - Reserved Jumper (E12)

<table>
<thead>
<tr>
<th>Reserved</th>
<th>Jumper Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 2</td>
<td>Out</td>
</tr>
<tr>
<td>2 - 3</td>
<td>Out</td>
</tr>
</tbody>
</table>

### Table 2-11 Factory Configured - DiskOnChip Address SEL Jumper (E13)

<table>
<thead>
<tr>
<th>DiskOnChip</th>
<th>Jumper Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>D000</td>
<td>1 - 2 In</td>
</tr>
<tr>
<td>E000</td>
<td>2 - 3 In</td>
</tr>
</tbody>
</table>

### Table 2-12 Factory Configured - Boot Block Lock Jumper (E17)

<table>
<thead>
<tr>
<th>Select</th>
<th>Jumper Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boot Block Programming</td>
<td>1 - 2</td>
</tr>
<tr>
<td>No Program</td>
<td>2 - 3</td>
</tr>
</tbody>
</table>

### Table 2-13 Factory Configured - Boot Block Lock (E18)

<table>
<thead>
<tr>
<th>Select</th>
<th>Jumper Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td>1-2</td>
</tr>
<tr>
<td>Reserved</td>
<td>2-3</td>
</tr>
<tr>
<td>Reserved</td>
<td>Omitted</td>
</tr>
</tbody>
</table>
Installation

The VMICPCI-7696 conforms to the CompactPCI physical specification for a 6U board. The VMICPCI-7696 is a system 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 CPCI chassis.

![Symbol Diagram]

This symbol identifies the System Controller slot

This symbol identifies peripheral slot

CAUTION: Do not install or remove the board while power is applied.

The following steps describe the VMIC recommended method for installation and powerup of the VMICPCI-7696:

1. Make sure power to the equipment is off.
2. If a PMC module such as VMIC’s VMIPMC-7441 is to be used, connect it to the VMICPCI-7696 prior to board installation. Refer to the Product Manual for that particular board for configuration and setup.
3. The VMICPCI-7696 must be installed in the designated system slot of the CompactPCI backplane (See symbols above for selection of the correct slot).

The VMICPCI-7696 requires forced air cooling. It is advisable to install blank panels over any exposed slots. This will allow for better air flow over the VMICPCI-7696 board.

4. Insert the VMICPCI-7696 into the CompactPCI chassis system 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.
5. 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.
6. The VMICPCI-7696 features a Flash Disk resident on the board. Refer to Chapter 4 for setup details.
7. If an external drive module is installed, the BIOS Setup program must be run to configure the drive types. See Appendix C to properly configure the system.
8. If a drive module is present, install the operating system according to the
manufacturer’s instructions.

9. A keyboard and a mouse are required if the user has not previously configured
the system

See Appendix B for instructions on installing VMICPCI-7696 peripheral driver
software during operating system installation.

BIOS Setup

The VMICPCI-7696 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-7696 is shipped from the factory with hard drive type configuration set
to AUTO in the CMOS.

Details of the VMICPCI-7696 BIOS setup program are included in Appendix D.

Front Panel Connectors

The front panel connections, including connector pinouts and orientation, for the
VMICPCI-7696 are defined in detail in Appendix A. Rear panel connections are
defined in detail in the Installation Guide for the VMIACC-0576.
The VMICPCI-7696 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-7696.

Figure 2-2 Installing a PMC Card on the VMICPCI-7696
LED Definition

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  Ready - Indicates BIOS Boot is complete, and module is ready

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

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.

Figure 2-3  Front Panel LED Positions
PC/AT Functions

Introduction

The VMICPCI-7696 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-7696 includes a PCI-compatible Ethernet controller.

The following sections describe in detail the PC/AT functions of the VMICPCI-7696.
CPU Socket

The VMICPCI-7696 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-7696 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-7696 provides Synchronous DRAM (SODIMM) as on-board system memory. Memory can be accessed as bytes, words, or longwords.

The VMICPCI-7696 accepts two 144-pin SODIMM DRAM modules for a maximum memory capacity of 256 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.

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-7696 provides 16 Mbyte of flash memory (See Note) accessible through the Secondary IDE port. The VMICPCI-7696 BIOS includes an option to allow the board to boot from the flash memory or from the Ethernet port.

The VMICPCI-7696 is also available with an optional M-systems DiskOnChip flash up to 72 Mbytes. Reference the VMICPCI-7696 Product Specification for available flash memory options.

The VMICPCI-7696 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.

Memory capacity may be extended as parts become available.
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 Kbyte I/O address space which is accessible as bytes, words, or longwords.

Standard PC/AT hardware circuitry reserves only 1,024 byte 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>
<thead>
<tr>
<th>I/O ADDRESS RANGE</th>
<th>SIZE IN BYTES</th>
<th>HW DEVICE</th>
<th>PC/AT FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>$000 - $00F</td>
<td>16</td>
<td>DMA Controller 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Intel 8237A Compatible)</td>
<td></td>
</tr>
<tr>
<td>$010 - $01F</td>
<td>16</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>$020 - $021</td>
<td>2</td>
<td>Master Interrupt Controller</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Intel 8259A Compatible)</td>
<td></td>
</tr>
<tr>
<td>$022 - $03F</td>
<td>30</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>$040 - $043</td>
<td>4</td>
<td>Programmable Timer</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Intel 8254 Compatible)</td>
<td></td>
</tr>
<tr>
<td>$044 - $05F</td>
<td>30</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>$060 - $064</td>
<td>5</td>
<td>Keyboard, Speaker, Eqpt. Configuration</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Intel 8042 Compatible)</td>
<td></td>
</tr>
<tr>
<td>$065 - $06F</td>
<td>11</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>$070 - $071</td>
<td>2</td>
<td>Real-Time Clock,</td>
<td></td>
</tr>
<tr>
<td>$072 - $07F</td>
<td>14</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>$080 - $08F</td>
<td>16</td>
<td>DMA Page Registers</td>
<td></td>
</tr>
<tr>
<td>$090 - $091</td>
<td>2</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>$092</td>
<td>1</td>
<td>Alt. Gate A20/Fast Reset Register</td>
<td></td>
</tr>
<tr>
<td>$093</td>
<td>1</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>
Table 3-1 VMICPCI-7696 I/O Address Map (Continued)

<table>
<thead>
<tr>
<th>I/O ADDRESS RANGE</th>
<th>SIZE IN BYTES</th>
<th>HW DEVICE</th>
<th>PC/AT FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>$094</td>
<td>1</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>$095 - $09F</td>
<td>11</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>$0A0 - $0A1</td>
<td>2</td>
<td>Slave Interrupt Controller (Intel 8259A Compatible)</td>
<td></td>
</tr>
<tr>
<td>$0A2 - $0BF</td>
<td>30</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>$0C0 - $0DF</td>
<td>32</td>
<td>DMA Controller 2 (Intel 8237A Compatible)</td>
<td></td>
</tr>
<tr>
<td>$0E0 - $16F</td>
<td>142</td>
<td>----</td>
<td>Reserved</td>
</tr>
<tr>
<td>$170 - $177</td>
<td>8</td>
<td>PIIX4</td>
<td>Secondary Hard Disk Controller</td>
</tr>
<tr>
<td>$178 - $1EF</td>
<td>120</td>
<td>User I/O</td>
<td></td>
</tr>
<tr>
<td>$1F0 - $1F7</td>
<td>8</td>
<td>PIIX4</td>
<td>Primary Hard Disk Controller</td>
</tr>
<tr>
<td>$1F8 - $277</td>
<td>128</td>
<td>User I/O</td>
<td></td>
</tr>
<tr>
<td>$278 - $27F</td>
<td>8</td>
<td>I/O Chip*</td>
<td>LPT2 Parallel I/O*</td>
</tr>
<tr>
<td>$280 - $2E7</td>
<td>104</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>$2E8 - $2EE</td>
<td>7</td>
<td>UART*</td>
<td>COM4 Serial I/O*</td>
</tr>
<tr>
<td>$2EF - $2F7</td>
<td>9</td>
<td>User I/O</td>
<td></td>
</tr>
<tr>
<td>$2F8 - $2FE</td>
<td>7</td>
<td>Super-I/O Chip</td>
<td>COM2 Serial I/O (16550 Compatible)</td>
</tr>
<tr>
<td>$2FF - $36F</td>
<td>113</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>$370 - $377</td>
<td>8</td>
<td>Super-I/O Chip</td>
<td>Secondary Floppy Disk Controller</td>
</tr>
<tr>
<td>$378 - $37F</td>
<td>8</td>
<td>Super-I/O Chip</td>
<td>LPT1 Parallel I/O</td>
</tr>
<tr>
<td>$380 - $3E7</td>
<td>108</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>$3E8 - $3EE</td>
<td>7</td>
<td>UART*</td>
<td>COM3 Serial I/O*</td>
</tr>
<tr>
<td>$3F0 - $3F7</td>
<td>8</td>
<td>Super-I/O Chip</td>
<td>Primary Floppy Disk Controller</td>
</tr>
<tr>
<td>$3F8 - $3FE</td>
<td>7</td>
<td>Super-I/O Chip</td>
<td>COM1 Serial I/O (16550 Compatible)</td>
</tr>
<tr>
<td>$3FF - $4FF</td>
<td></td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>$500 - $503</td>
<td>4</td>
<td>82C54 Timer</td>
<td>Programmable Internal Timer</td>
</tr>
<tr>
<td>$504 - CFF</td>
<td></td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

* While these I/O ports are reserved for the listed functions, they are not implemented on the VMICPCI-7696. They are listed here to make the user aware of the standard PC/AT usage of these ports.
PCI-to-PCI Bridge

The VMICPCI-7696 uses the Intel 21153 PCI-to-PCI bridge to interface between the primary PCI bus of unit 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 auto detected by the BIOS and respond to normal PCI accesses.

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 46 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 46.

The interrupt hardware implementation on the VMICPCI-7696 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 50.

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

<table>
<thead>
<tr>
<th>IRQ</th>
<th>AT FUNCTION</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMI</td>
<td>Parity Errors (Must be enabled in BIOS Setup)</td>
<td>Used by VMICPCI-7696 PCI bus Interface</td>
</tr>
<tr>
<td>0</td>
<td>System Timer</td>
<td>Set by BIOS Setup</td>
</tr>
<tr>
<td>1</td>
<td>Keyboard</td>
<td>Set by BIOS Setup</td>
</tr>
<tr>
<td>2</td>
<td>Duplexed to IRQ9</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>COM2/COM4</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>COM1/COM3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Timer</td>
<td>Assigned to On-Board Timer</td>
</tr>
<tr>
<td>6</td>
<td>Floppy Controller</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>LPT1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Real-Time Clock</td>
<td></td>
</tr>
</tbody>
</table>
### Table 3-2 PC/AT Hardware Interrupt Line Assignments (Continued)

<table>
<thead>
<tr>
<th>IRQ</th>
<th>AT FUNCTION</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Old IRQ2</td>
<td>SVGA or Network I/O</td>
</tr>
<tr>
<td>10</td>
<td>Not Assigned</td>
<td>Determined by BIOS</td>
</tr>
<tr>
<td>11</td>
<td>Not Assigned</td>
<td>Determined by BIOS</td>
</tr>
<tr>
<td>12</td>
<td>Mouse</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Math Coprocessor</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>AT Hard Drive</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Flash Drive</td>
<td></td>
</tr>
</tbody>
</table>

### Table 3-3 PC/AT Interrupt Vector Table

<table>
<thead>
<tr>
<th>INTERRUPT NO. HEX</th>
<th>INTERRUPT NO. DEC</th>
<th>IRQ LINE</th>
<th>REAL MODE</th>
<th>PROTECTED MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
<td>Divide Error</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>1</td>
<td>Debug Single Step</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>02</td>
<td>2</td>
<td>2</td>
<td>NMI Memory Parity Error, CompactPCI Interrupts</td>
<td>Same as Real Mode (Must be enabled in BIOS Setup)</td>
</tr>
<tr>
<td>03</td>
<td>3</td>
<td>3</td>
<td>Debug Breakpoint</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>04</td>
<td>4</td>
<td>4</td>
<td>ALU Overflow</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>05</td>
<td>5</td>
<td>5</td>
<td>Print Screen</td>
<td>Array Bounds Check</td>
</tr>
<tr>
<td>06</td>
<td>6</td>
<td>6</td>
<td>Print Screen</td>
<td>Invalid OpCode</td>
</tr>
<tr>
<td>07</td>
<td>7</td>
<td>7</td>
<td>Device Not Available</td>
<td></td>
</tr>
<tr>
<td>08</td>
<td>8</td>
<td>8</td>
<td>IRQ0 Timer Tick</td>
<td>Double Exception Detected</td>
</tr>
<tr>
<td>09</td>
<td>9</td>
<td>9</td>
<td>IRQ1 Keyboard Input</td>
<td>Coprocessor Segment Overrun</td>
</tr>
<tr>
<td>0A</td>
<td>10</td>
<td>10</td>
<td>IRQ2 BIOS Reserved</td>
<td>Invalid Task State Segment</td>
</tr>
<tr>
<td>0B</td>
<td>11</td>
<td>11</td>
<td>IRQ3 COM2 Serial I/O</td>
<td>Segment Not Present</td>
</tr>
<tr>
<td>0C</td>
<td>12</td>
<td>12</td>
<td>IRQ4 COM1 Serial I/O</td>
<td>Stack Segment Overrun</td>
</tr>
<tr>
<td>0D</td>
<td>13</td>
<td>13</td>
<td>IRQ5 Timer</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>0E</td>
<td>14</td>
<td>14</td>
<td>IRQ6 Floppy Disk Controller</td>
<td>Page Fault</td>
</tr>
<tr>
<td>0F</td>
<td>15</td>
<td>15</td>
<td>IRQ7 LPT1 Parallel I/O</td>
<td>Unassigned</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
<td>16</td>
<td>BIOS Video I/O</td>
<td>Coprocessor Error</td>
</tr>
</tbody>
</table>
### Table 3-3  PC/AT Interrupt Vector Table (Continued)

<table>
<thead>
<tr>
<th>INTERRUPT NO.</th>
<th>IRQ LINE</th>
<th>REAL MODE</th>
<th>PROTECTED MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEX</td>
<td>DEC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>17</td>
<td>Eqpt Configuration Check</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>12</td>
<td>18</td>
<td>Memory Size Check</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>13</td>
<td>19</td>
<td>XT Floppy/Hard Drive</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>14</td>
<td>20</td>
<td>BIOS Comm I/O</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>15</td>
<td>21</td>
<td>BIOS Cassette Tape I/O</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>16</td>
<td>22</td>
<td>BIOS Keyboard I/O</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>17</td>
<td>23</td>
<td>BIOS Printer I/O</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>18</td>
<td>24</td>
<td>ROM BASIC Entry Point</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>19</td>
<td>25</td>
<td>Bootstrap Loader</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>1A</td>
<td>26</td>
<td>IRQ8 Real-Time Clock</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>1B</td>
<td>27</td>
<td>Control/Break Handler</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>1C</td>
<td>28</td>
<td>Timer Control</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>1D</td>
<td>29</td>
<td>Video Parameter Table Pntr</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>1E</td>
<td>30</td>
<td>Floppy Parm Table Pntr</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>1F</td>
<td>31</td>
<td>Video Graphics Table Pntr</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>20</td>
<td>32</td>
<td>DOS Terminate Program</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>21</td>
<td>33</td>
<td>DOS Function Entry Point</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>22</td>
<td>34</td>
<td>DOS Terminate Handler</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>23</td>
<td>35</td>
<td>DOS Control/Break Handler</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>24</td>
<td>36</td>
<td>DOS Critical Error Handler</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>25</td>
<td>37</td>
<td>DOS Absolute Disk Read</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>26</td>
<td>38</td>
<td>DOS Absolute Disk Write</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>27</td>
<td>39</td>
<td>DOS Program Terminate, Stay Resident</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>28</td>
<td>40</td>
<td>DOS Keyboard Idle Loop</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>29</td>
<td>41</td>
<td>DOS CON Dev. Raw Output</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>2A</td>
<td>42</td>
<td>DOS 3.x+ Network Comm</td>
<td>Same as Real Mode</td>
</tr>
</tbody>
</table>
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 50 depicts the VMICPCI-7696 interrupt logic pertaining to CompactPCI operations and the PMC site.

<table>
<thead>
<tr>
<th>INTERRUPT NO.</th>
<th>IRQ LINE</th>
<th>REAL MODE</th>
<th>PROTECTED MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2B</td>
<td>43</td>
<td>DOS Internal Use</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>2C</td>
<td>44</td>
<td>DOS Internal Use</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>2D</td>
<td>45</td>
<td>DOS Internal Use</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>2E</td>
<td>46</td>
<td>DOS Internal Use</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>2F</td>
<td>47</td>
<td>DOS Print Spooler Driver</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>30-60</td>
<td>48-96</td>
<td>Reserved by DOS</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>61-66</td>
<td>97-102</td>
<td>User Available</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>67-71</td>
<td>103-113</td>
<td>Reserved by DOS</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>72</td>
<td>114</td>
<td>IRQ10</td>
<td>Not Assigned</td>
</tr>
<tr>
<td>73</td>
<td>115</td>
<td>IRQ11</td>
<td>Not Assigned</td>
</tr>
<tr>
<td>74</td>
<td>116</td>
<td>IRQ12</td>
<td>Mouse</td>
</tr>
<tr>
<td>75</td>
<td>117</td>
<td>IRQ13</td>
<td>Math Coprocessor</td>
</tr>
<tr>
<td>76</td>
<td>118</td>
<td>IRQ14</td>
<td>AT Hard Drive</td>
</tr>
<tr>
<td>77</td>
<td>119</td>
<td>IRQ15</td>
<td>Flash Drive</td>
</tr>
<tr>
<td>78-7F</td>
<td>120-127</td>
<td>Reserved by DOS</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>80-F0</td>
<td>128-240</td>
<td>Reserved for BASIC</td>
<td>Same as Real Mode</td>
</tr>
<tr>
<td>F1-FF</td>
<td>241-255</td>
<td>Reserved by DOS</td>
<td>Same as Real Mode</td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Table 3-4  NMI Register Bit Descriptions</th>
</tr>
</thead>
</table>
| **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-7696 incorporates the SMC Super-I/O chip. The SMC chip provides the VMICPCI-7696 with a standard floppy drive controller, USB port, two 16550 UART-compatible serial ports, and one enhance 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.
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-7696 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-7696 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-7696 is capable of booting from a server over the Ethernet. Utilizing Lanworks BootWare®. BootWare gives you the ability to remotely boot the VMICPCI-7696 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-7696 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

<table>
<thead>
<tr>
<th>Screen Resolution</th>
<th>Maximum Colors</th>
<th>Maximum Refresh Rates (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>640 x 480</td>
<td>16 M</td>
<td>85</td>
</tr>
<tr>
<td>800 x 600</td>
<td>16 M</td>
<td>85</td>
</tr>
<tr>
<td>1,024 x 768</td>
<td>16 M</td>
<td>85</td>
</tr>
<tr>
<td>1,280 x 1,024</td>
<td>64 K</td>
<td>60</td>
</tr>
<tr>
<td>1600 x 1200</td>
<td>64 K</td>
<td>60</td>
</tr>
</tbody>
</table>

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-7696’s processor includes a 64-bit access to video memory with no-wait states. Video I/O registers are accessed using the AGP bus.

The floppy disks supplied with the VMICPCI-7696 contain drivers for Windows, Windows 95, and Windows NT (4.0) operating systems. Appendix B contains instruction on the incorporation of the drivers during system installation.
# Embedded PC/RTOS Features

## Contents

<table>
<thead>
<tr>
<th>Feature</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timers</td>
<td>56</td>
</tr>
<tr>
<td>Flash Disk</td>
<td>65</td>
</tr>
<tr>
<td>Flash Disk (Optional)</td>
<td>69</td>
</tr>
<tr>
<td>Watchdog Timer</td>
<td>71</td>
</tr>
<tr>
<td>Battery Backed SRAM</td>
<td>77</td>
</tr>
</tbody>
</table>

## Introduction

VMIC’s VMICPCI-7696 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-7696 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-7696 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 Table 2-9 on page 35 of 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 circuitry external to the 82C54. The external circuitry 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.

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.

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timer 0</td>
<td>Timer 1</td>
<td>Timer 2</td>
<td>Unused</td>
<td>Unused</td>
<td>Unused</td>
<td>Unused</td>
<td>Unused</td>
</tr>
</tbody>
</table>

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. 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.

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unused</td>
<td>Timer 0</td>
<td>Unused</td>
<td>Timer 1</td>
<td>Timer 2</td>
<td>Unused</td>
<td>Unused</td>
<td>Unused</td>
</tr>
</tbody>
</table>

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-7696 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 58) 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

<table>
<thead>
<tr>
<th>I/O Address</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>$500</td>
<td>Timer 0</td>
</tr>
<tr>
<td>$501</td>
<td>Timer 1</td>
</tr>
<tr>
<td>$502</td>
<td>Timer 2</td>
</tr>
<tr>
<td>$503</td>
<td>control word Register</td>
</tr>
</tbody>
</table>
Table 4-1 on page 57 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 61).

![82C54 Diagram](image)

The three Timers, Timer 0, 1, and 2, are functionally equivalent. Therefore only a single Timer will be described. Figure 4-5 on page 59 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, as shown in Figure 4-5 on page 59, when latched, 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 61). 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

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST1</td>
<td>ST0</td>
<td>RW1</td>
<td>RW0</td>
<td>M2</td>
<td>M1</td>
<td>M0</td>
<td>BCD</td>
</tr>
</tbody>
</table>

Table 4-3 ST - Select Timer

<table>
<thead>
<tr>
<th>ST1*</th>
<th>ST0*</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Select Timer 0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Select Timer 1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Select Timer 2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Read-Back Command (See Reading section on page 1-61)</td>
</tr>
</tbody>
</table>

* 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

<table>
<thead>
<tr>
<th>RW1*</th>
<th>RW0*</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Timer Latch Command (see Reading section)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Read/Write least significant byte only</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Read/Write most significant byte only</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Read/Write least significant byte first, then most significant</td>
</tr>
</tbody>
</table>

* The RW bits specify whether this is a Timer Latch command or the byte ordering of the Read/Write transaction.
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 57). The Select Timer bits (ST1, ST0, see Table 4-3 on page 60) select one of the three timers while the Read /Write bits (RW1,RW0, see Table 4-4 on page 60) 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>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Count</td>
<td>Status</td>
<td>Cnt2</td>
<td>Cnt1</td>
<td>Cnt0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4-7 Read-Back Command Format

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D5: Count</td>
<td>Latch count of selected Timer(s)</td>
</tr>
<tr>
<td>D4: Status</td>
<td>Latch status of selected Timer(s)</td>
</tr>
<tr>
<td>D3: Cnt2</td>
<td>Select Timer 2</td>
</tr>
<tr>
<td>D2: Cnt1</td>
<td>Select Timer 1</td>
</tr>
<tr>
<td>D1: Cnt0</td>
<td>Select Timer 0</td>
</tr>
<tr>
<td>D0</td>
<td>Reserved, must be 0</td>
</tr>
</tbody>
</table>

Table 4-8 Read-Back Command Description

The Read-Back Command can be used to latch several Timer counts by setting the 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 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 57). 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.
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 60.

**Table 4-9** Status Byte

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D7</td>
<td>OUT</td>
</tr>
<tr>
<td>D6</td>
<td>LOAD</td>
</tr>
<tr>
<td>D5</td>
<td>RW1</td>
</tr>
<tr>
<td>D4</td>
<td>RW0</td>
</tr>
<tr>
<td>D3</td>
<td>M2</td>
</tr>
<tr>
<td>D2</td>
<td>M1</td>
</tr>
<tr>
<td>D1</td>
<td>M0</td>
</tr>
<tr>
<td>D0</td>
<td>BCD</td>
</tr>
</tbody>
</table>

**Table 4-10** Status Byte Description

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D7</td>
<td>OUT: Current state of Timers OUT pin</td>
</tr>
<tr>
<td>D6</td>
<td>LOAD: Count loaded into Timer</td>
</tr>
<tr>
<td>D5-D0</td>
<td>Timer Programmed Mode</td>
</tr>
</tbody>
</table>

**Table 4-11** LOAD Bit Operation

<table>
<thead>
<tr>
<th>Action</th>
<th>Causes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Write to the control word Register ¹</td>
<td>LOAD bit = 1</td>
</tr>
<tr>
<td>2. Write count to Timer ²</td>
<td>LOAD bit = 1</td>
</tr>
<tr>
<td>3. New count loaded into Timer</td>
<td>LOAD bit = 0</td>
</tr>
</tbody>
</table>

¹Only the Timer specified in the control word will have its LOAD bit set to 1. LOAD bits of other Timers are not affected.

²If 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 Count bit (D5) and 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-7696 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-7696 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-7696 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-7696 BIOS includes an option to allow the board to boot from the Flash Disk.

Configuration

The Flash Disk resides on the VMICPCI-7696 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-7696 with a resident Flash Disk, a hard drive attached to the Primary IDE interface, and a floppy drive attached to the floppy interface.

![Typical System Configuration](image)

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.
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-7696 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 one 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-7696 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-7696, 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, 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.

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 65. 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.

Drive letter changes caused by adding an addition 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-7696 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, which is located between address 0xD0000 to 0xD1FFF.

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, just 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”.

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 need to 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).


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 on the disk provided, labeled DiskOnChip Utilities. 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 (320-500039-006) into the floppy drive and type A:[e].
3. At the DOS prompt type: DUPDATE D:/FIRST/S:DOC2000.EXE.
4. After rebooting the system, the DiskOnChip will appear as drive C.
Watchdog Timer

The VMICPCI-7696 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-7696 are located at the standard PC/AT I/O address. The Time of Day feature in the DS1384 Watchdog Timer is available for use by the user at their 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.

Figure 4-7 shows a generalized block diagram of how the Watchdog Timer is used in the VMICPCI-7696. 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 72 shows the address, content and the range of each Watchdog Register.

![Figure 4-7 Watchdog Alarm Block](image-url)
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.

<table>
<thead>
<tr>
<th>Register</th>
<th>Address</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$D8000</td>
<td>0.1 S</td>
<td>0.01 S</td>
<td>00 - 99</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$D8001</td>
<td>10 S</td>
<td>Seconds</td>
<td>00 - 59</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$D8002</td>
<td>10 M</td>
<td>Minutes</td>
<td>00 - 59</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$D8003</td>
<td>M</td>
<td>10 Minute Alarm</td>
<td>00 - 59</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$D8004</td>
<td>0</td>
<td>12/24</td>
<td>Hours (BCD)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>$D8005</td>
<td>M</td>
<td>12/24</td>
<td>Hour Alarm (BCD)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>$D8006</td>
<td>0</td>
<td>0</td>
<td>Days (BCD)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>$D8007</td>
<td>M</td>
<td>0</td>
<td>Day Alarm (BCD)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>$D8008</td>
<td>0</td>
<td>10 Date (BCD)</td>
<td>Date (BCD)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>$D8009</td>
<td>Esc</td>
<td>1**</td>
<td>Months (BCD)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>$D800A</td>
<td>10 Years (BCD)</td>
<td>Years (BCD)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>$D800B</td>
<td>Te</td>
<td>Ipasw</td>
<td>Wam Tdm Waf Tdf</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>$D800C</td>
<td>0.1 S</td>
<td>0.01 S</td>
<td>00 - 99</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>$D800D</td>
<td>10 S</td>
<td>Seconds</td>
<td>00 - 99</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* 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.
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 while 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 while 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 0.01 Seconds value with a range of 0 to 9 in BCD while 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. 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 and 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 while 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, $\text{Eosc}$, 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>
<thead>
<tr>
<th>Register</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minutes</td>
<td>Hours</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The Time of Day Alarm registers are read and written 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 while 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 while 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 71. 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 that is 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-7696 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.
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

J1 Connector Pinout .......................................................... 83
J2 Connector and Pinout ...................................................... 84
J3 Connector Pinout .......................................................... 85
J5 Connector and Pinout ...................................................... 86
Keyboard and Mouse Connectors and Pinout (J10) .................... 90
PMC Connector Pinout ......................................................... 92

Introduction

The VMICPCI-7696 PC/AT-Compatible CompactPCI® Controller has several connectors for its I/O ports. Figure A-1 shows the locations of the connectors on the VMICPCI-7696. Wherever possible, the VMICPCI-7696 uses connectors and pinouts typical for any desktop PC. This ensures maximum compatibility with a variety of systems.

Connector diagrams in this appendix are generally shown in a natural orientation with the controller board mounted in a CPCI bus chassis.
### VMICPCI-7696 Jumper Configuration

<table>
<thead>
<tr>
<th>Jumper</th>
<th>Configuration</th>
</tr>
</thead>
</table>
| E18: Boot Block Lock | Pins 1 and 2 Reserved  
Pins 2 and 3 Reserved  
Omitted = Reserved |
| E7: Watchdog Timer SEL | Pins 1 and 2 IN = RTCRESET#  
Pins 2 and 3 IN = SERR# |
| E17: BIOS Mode Option | Pins 1 and 2 = Boot Block Prgm  
Pins 2 and 3 = Boot Block No Prgm |
| E6: PIIX4E Battery | Pins 1 and 2 IN = Battery Enabled  
Pins 2 and 3 IN = Battery Disabled |
| E13: DiskOnChip Address SEL | Pins 1 and 2 IN = D000  
Pins 2 and 3 IN = E000 |
| E5: Host Frequency | Pins 1 and 2 IN = 100 MHz  
OUT = 66 MHz |
| E12: Reserved |  |
| E4: AGP Enable | Pins 1 and 2 IN = Disable AGP  
OUT = Enable AGP |
| E11: Prog Timer Clock SEL | Pins 1 and 2 IN = 2 MHz  
Pins 2 and 3 IN = 1 MHz |
| E3: Password Clear | Pins 1 and 2 IN = Clear Password |
| E10: SRAM Battery SEL | Pins 1 and 2 IN = Enabled |
| E2: Fan Connector |  |
| E8: CPU Clock Speed SEL | Pins 1 and 2 IN = 100 MHz  
Pins 2 and 3 IN = 66 MHz |

**Figure A-1** VMICPCI-7696 Connector and Jumper Locations

**Bold = Default Factory Configuration**
The VMICPCI-7696 utilizes 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-2 below depicts the J1 connector and the connector pinout.

<table>
<thead>
<tr>
<th>Pin No.</th>
<th>Row A</th>
<th>Row B</th>
<th>Row C</th>
<th>Row D</th>
<th>Row E</th>
<th>Row F</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>+5 V</td>
<td>C_REQ64#</td>
<td>C_ENUM</td>
<td>PCI_3.3</td>
<td>+5 V</td>
<td>GND</td>
</tr>
<tr>
<td>24</td>
<td>C_AD[1]</td>
<td>+5 V</td>
<td>PCI_VIO</td>
<td>C_AD[0]</td>
<td>C_ACK64#</td>
<td>N/C</td>
</tr>
<tr>
<td>21</td>
<td>PCI_3.3</td>
<td>C_AD[9]</td>
<td>C_AD[8]</td>
<td>GND</td>
<td>C_C/BE0#</td>
<td>GND</td>
</tr>
<tr>
<td>18</td>
<td>C_SERR#</td>
<td>GND</td>
<td>PCI_3.3</td>
<td>C_PAR</td>
<td>C_C/BE1#</td>
<td>N/C</td>
</tr>
<tr>
<td>17</td>
<td>PCI_3.3</td>
<td>C_SDONE</td>
<td>C_SR0#</td>
<td>GND</td>
<td>C_PER#</td>
<td>GND</td>
</tr>
<tr>
<td>16</td>
<td>C_DEVSEL#</td>
<td>GND</td>
<td>PCI_VIO</td>
<td>C_STOP#</td>
<td>C_LOCK#</td>
<td>N/C</td>
</tr>
<tr>
<td>15</td>
<td>PCI_3.3</td>
<td>C_FRAME</td>
<td>C_IRDY#</td>
<td>GND</td>
<td>C_TRDY#</td>
<td>GND</td>
</tr>
<tr>
<td>14</td>
<td>lost to the keying area</td>
<td>N/C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>lost to the keying area</td>
<td>GND</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>lost to the keying area</td>
<td>N/C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>C_CBE3#</td>
<td>N/C</td>
<td>C_AD[23]</td>
<td>GND</td>
<td>C_AD[22]</td>
<td>GND</td>
</tr>
<tr>
<td>6</td>
<td>C_REQ#</td>
<td>GND</td>
<td>PCI_3.3</td>
<td>C_CLK</td>
<td>C_AD[31]</td>
<td>N/C</td>
</tr>
<tr>
<td>5</td>
<td>N/C</td>
<td>N/C</td>
<td>C_RST#</td>
<td>GND</td>
<td>C_GNT#</td>
<td>GND</td>
</tr>
<tr>
<td>4</td>
<td>N/C</td>
<td>GND</td>
<td>PCI_VIO</td>
<td>N/C</td>
<td>N/C</td>
<td>N/C</td>
</tr>
<tr>
<td>3</td>
<td>C_INTA#</td>
<td>C_INTB#</td>
<td>C_INTC#</td>
<td>+5 V</td>
<td>C_INTD#</td>
<td>GND</td>
</tr>
<tr>
<td>2</td>
<td>C_TCK</td>
<td>+5 V</td>
<td>C_TMS</td>
<td>N/C</td>
<td>C_TDI</td>
<td>N/C</td>
</tr>
<tr>
<td>1</td>
<td>+5 V</td>
<td>-12 V</td>
<td>C_TRST#</td>
<td>+12 V</td>
<td>+5 V</td>
<td>GND</td>
</tr>
</tbody>
</table>

Figure A-2 J1 Connector and Pinout
J2 Connector and Pinout

The VMICPCI-7696 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 is also used, labeled row F. This connector’s controlled impedance minimizes unwanted signal reflections. Figure A-3 illustrates the J2 connector and the connector pinout.

<table>
<thead>
<tr>
<th>Pin No.</th>
<th>Row A</th>
<th>Row B</th>
<th>Row C</th>
<th>Row D</th>
<th>Row E</th>
<th>Row F</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>N/C</td>
<td>N/C</td>
<td>N/C</td>
<td>N/C</td>
<td>N/C</td>
<td>GND</td>
</tr>
<tr>
<td>21</td>
<td>C_CLK[6]</td>
<td>GND</td>
<td>N/C</td>
<td>N/C</td>
<td>N/C</td>
<td>N/C</td>
</tr>
<tr>
<td>20</td>
<td>C_CLK[5]</td>
<td>GND</td>
<td>N/C</td>
<td>GND</td>
<td>N/C</td>
<td>GND</td>
</tr>
<tr>
<td>19</td>
<td>GND</td>
<td>GND</td>
<td>N/C</td>
<td>N/C</td>
<td>N/C</td>
<td>N/C</td>
</tr>
<tr>
<td>18</td>
<td>N/C</td>
<td>N/C</td>
<td>N/C</td>
<td>GND</td>
<td>N/C</td>
<td>GND</td>
</tr>
<tr>
<td>17</td>
<td>N/C</td>
<td>GND</td>
<td>C-PRST#</td>
<td>C_REQ6#</td>
<td>C_GNT6#</td>
<td>N/C</td>
</tr>
<tr>
<td>16</td>
<td>N/C</td>
<td>C_REQ#</td>
<td>C_REQ6#</td>
<td>GND</td>
<td>N/C</td>
<td>GND</td>
</tr>
<tr>
<td>15</td>
<td>N/C</td>
<td>GND</td>
<td>C_FAL#</td>
<td>C_REQ5#</td>
<td>C_GNT5#</td>
<td>N/C</td>
</tr>
<tr>
<td>14</td>
<td>N/C</td>
<td>N/C</td>
<td>N/C</td>
<td>GND</td>
<td>N/C</td>
<td>GND</td>
</tr>
<tr>
<td>13</td>
<td>N/C</td>
<td>GND</td>
<td>CPC_VIO</td>
<td>N/C</td>
<td>N/C</td>
<td>N/C</td>
</tr>
<tr>
<td>12</td>
<td>N/C</td>
<td>N/C</td>
<td>N/C</td>
<td>GND</td>
<td>N/C</td>
<td>GND</td>
</tr>
<tr>
<td>11</td>
<td>N/C</td>
<td>GND</td>
<td>CPC_VIO</td>
<td>N/C</td>
<td>N/C</td>
<td>N/C</td>
</tr>
<tr>
<td>10</td>
<td>N/C</td>
<td>N/C</td>
<td>N/C</td>
<td>GND</td>
<td>N/C</td>
<td>GND</td>
</tr>
<tr>
<td>9</td>
<td>N/C</td>
<td>GND</td>
<td>CPC_VIO</td>
<td>N/C</td>
<td>N/C</td>
<td>N/C</td>
</tr>
<tr>
<td>8</td>
<td>N/C</td>
<td>N/C</td>
<td>N/C</td>
<td>GND</td>
<td>N/C</td>
<td>GND</td>
</tr>
<tr>
<td>7</td>
<td>N/C</td>
<td>GND</td>
<td>CPC_VIO</td>
<td>N/C</td>
<td>N/C</td>
<td>N/C</td>
</tr>
<tr>
<td>6</td>
<td>N/C</td>
<td>N/C</td>
<td>N/C</td>
<td>GND</td>
<td>N/C</td>
<td>GND</td>
</tr>
<tr>
<td>5</td>
<td>C_C/BE5#</td>
<td>GND</td>
<td>CPC_VIO</td>
<td>C_C/BE4#</td>
<td>C_PAR64</td>
<td>N/C</td>
</tr>
<tr>
<td>4</td>
<td>CPC_VIO</td>
<td>N/C</td>
<td>C_C/BE7#</td>
<td>GND</td>
<td>C_C/BE6#</td>
<td>GND</td>
</tr>
<tr>
<td>3</td>
<td>C_CLK[4]</td>
<td>GND</td>
<td>C_GNT3#</td>
<td>C_REQ4#</td>
<td>C_GNT4#</td>
<td>N/C</td>
</tr>
<tr>
<td>1</td>
<td>C_CLK[1]</td>
<td>GND</td>
<td>C_REQ1#</td>
<td>C_GNT1#</td>
<td>C_REQ2#</td>
<td>N/C</td>
</tr>
</tbody>
</table>

* CPC_VIO - The VMICPCI-7696 is a universal VIO design.
* The VMICPCI-7696 supports a 32-bit PCI bus.
* J2 Row C pins 15 and 16 are not supported.

Figure A-3  J2 Connector and Pinout
J3 Connector Pinout

The J3 connector is 5 rows of 19 pins each, 2mm “Hard Metric” CompactPCI connector. An additional external metal shield is also used, labeled row F. Figure A-4 illustrates the J3 connector and the connector pinout. This connector is also used to route the serial, floppy, and harddrive signals to the backplane I/O.

<table>
<thead>
<tr>
<th>Pin No.</th>
<th>Row A</th>
<th>Row B</th>
<th>Row C</th>
<th>Row D</th>
<th>Row E</th>
<th>Row F</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>GND</td>
<td>ACK#</td>
<td>BUSY</td>
<td>PE</td>
<td>SLCT</td>
<td>GND</td>
</tr>
<tr>
<td>18</td>
<td>PD4</td>
<td>PD5</td>
<td>PD6</td>
<td>PD7</td>
<td>GND</td>
<td>N/C</td>
</tr>
<tr>
<td>17</td>
<td>INIT#</td>
<td>PD2</td>
<td>SLIN#</td>
<td>PD3</td>
<td>N/C</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>N/C</td>
<td>STB#</td>
<td>AFD#</td>
<td>PD0</td>
<td>ERR#</td>
<td>N/C</td>
</tr>
<tr>
<td>15</td>
<td>USBP1+RTC</td>
<td>USBP1-RTC</td>
<td>USB0C1#</td>
<td>DATA</td>
<td>CLK</td>
<td>GND</td>
</tr>
<tr>
<td>14</td>
<td>GND</td>
<td>SP0_R1#B</td>
<td>SP0_DTR#B</td>
<td>SP0_CTS#B</td>
<td>SP0_TXB</td>
<td>N/C</td>
</tr>
<tr>
<td>13</td>
<td>SP0_RTS</td>
<td>SP0_RXB</td>
<td>SP0_DSR#B</td>
<td>SP0_DCD#B</td>
<td>N/C</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>SP1_RTS</td>
<td>N/C</td>
<td>+5 V</td>
<td>N/C</td>
<td>+12 V</td>
<td>N/C</td>
</tr>
<tr>
<td>11</td>
<td>GND</td>
<td>SP1_R1#B</td>
<td>SP1_DTR#B</td>
<td>SP1_CTS#B</td>
<td>SP1_TXB</td>
<td>N/C</td>
</tr>
<tr>
<td>10</td>
<td>DAP0</td>
<td>SP1_RXB</td>
<td>SP1_DSR#B</td>
<td>SP1_DCD#</td>
<td>DAP2PR</td>
<td>N/C</td>
</tr>
<tr>
<td>9</td>
<td>DAP1</td>
<td>REDWC#</td>
<td>IDEIORDY</td>
<td>IDEDELA</td>
<td>IDEION0#</td>
<td>GND</td>
</tr>
<tr>
<td>8</td>
<td>DIR#</td>
<td>INDEX#</td>
<td>IDECS01#</td>
<td>DOP[8]</td>
<td>DOP[6]</td>
<td>N/C</td>
</tr>
<tr>
<td>7</td>
<td>SIDE1#</td>
<td>IDEIOR0#</td>
<td>HD_ACTA#</td>
<td>DOP[9]</td>
<td>DOP[7]</td>
<td>GND</td>
</tr>
<tr>
<td>6</td>
<td>RDATA#</td>
<td>TRK0#</td>
<td>IDECS03#</td>
<td>DOP[10]</td>
<td>DOP[2]</td>
<td>N/C</td>
</tr>
<tr>
<td>4</td>
<td>DRVSB#</td>
<td>STEP#</td>
<td>IDEOACK0#</td>
<td>DOP[12]</td>
<td>DOP[4]</td>
<td>N/C</td>
</tr>
<tr>
<td>2</td>
<td>WPT#</td>
<td>DRVSA#</td>
<td>IDEDRQ0</td>
<td>DOP[14]</td>
<td>DOP[6]</td>
<td>N/C</td>
</tr>
</tbody>
</table>

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

**Figure A-4** J3 Connector and Pinout
J5 Connector and Pinout

The VMICPCI-7696 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-5 illustrates the J5 connector and the connector pinout.

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

Figure A-5  J5 Connector and Pinout
Ethernet Connector Pinout (J12)

The pinout diagram for the Ethernet 10BaseT and 100BaseTx connector is shown in Figure A-6.

**Figure A-6: Ethernet Connector and Pinout**

<table>
<thead>
<tr>
<th>PIN</th>
<th>Signal Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TD+ Transmit Data</td>
</tr>
<tr>
<td>2</td>
<td>TD- Transmit Data</td>
</tr>
<tr>
<td>3</td>
<td>RD+ Receive Data</td>
</tr>
<tr>
<td>4</td>
<td>NC No Connection</td>
</tr>
<tr>
<td>5</td>
<td>NC No Connection</td>
</tr>
<tr>
<td>6</td>
<td>RD- Receive Data</td>
</tr>
<tr>
<td>7</td>
<td>NC No Connection</td>
</tr>
<tr>
<td>8</td>
<td>NC No Connection</td>
</tr>
</tbody>
</table>
Video Connector Pinout (J11)

The video port uses a standard high-density DB15 SVGA connector. Figure A-7 illustrates the connector and pinout.

![Video Connector Diagram]

<table>
<thead>
<tr>
<th>Pin</th>
<th>Direction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Out</td>
<td>Red</td>
</tr>
<tr>
<td>2</td>
<td>Out</td>
<td>Green</td>
</tr>
<tr>
<td>3</td>
<td>Out</td>
<td>Blue</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Ground</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Ground</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Ground</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Ground</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>5 V</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>Ground</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>12</td>
<td>I2 Data</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Out</td>
<td>Horizontal SYNC</td>
</tr>
<tr>
<td>14</td>
<td>Out</td>
<td>Vertical SYNC</td>
</tr>
<tr>
<td>15</td>
<td>I2 Clock</td>
<td></td>
</tr>
<tr>
<td>Shield</td>
<td></td>
<td>Chassis Ground</td>
</tr>
</tbody>
</table>

*Figure A-7 Video Connector and Pinout*
Parallel Port Connector Pinout (P2)

The parallel port shown in Figure A-8 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-7696 Product Specification” for ordering information.

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

Figure A-8  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-9 and Table A-14.

Figure A-9  Keyboard/Mouse Connector and Pinout

Table A-14  Keyboard/Mouse Y Splitter Cable

<table>
<thead>
<tr>
<th>Pin</th>
<th>Dir</th>
<th>Function</th>
<th>Pin</th>
<th>Dir</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>In/Out</td>
<td>Keyboard Data</td>
<td>1</td>
<td>In/Out</td>
<td>Mouse Data</td>
</tr>
<tr>
<td>2</td>
<td>Unused</td>
<td></td>
<td>2</td>
<td>Unused</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Ground</td>
<td></td>
<td>3</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>+5 V</td>
<td></td>
<td>4</td>
<td>+5 V</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Out</td>
<td>Keyboard Clock</td>
<td>5</td>
<td>Out</td>
<td>Mouse Clock</td>
</tr>
<tr>
<td>6</td>
<td>Unused</td>
<td></td>
<td>6</td>
<td>Unused</td>
<td></td>
</tr>
<tr>
<td>Shield</td>
<td>Chassis Ground</td>
<td>Shield</td>
<td>Shield</td>
<td>Chassis Ground</td>
<td></td>
</tr>
</tbody>
</table>
USB Connector (J9)

The Universal Serial Bus (USB) port uses an industry standard four position shielded connector. Figure A-10 shows the pinout of the USB connector.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>USBV</td>
<td>USB Power</td>
</tr>
<tr>
<td>2</td>
<td>USB-</td>
<td>USB Data -</td>
</tr>
<tr>
<td>3</td>
<td>USB+</td>
<td>USB Data +</td>
</tr>
<tr>
<td>4</td>
<td>USBG</td>
<td>USB Ground</td>
</tr>
</tbody>
</table>

**Figure A-10** 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. Tables A-1 through A-3 are the pinouts for the PMC connectors (J6, J7, and J4).

![PMC J7 Connector Diagram]

Table A-1 PMC J7 Connector Pinout

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Pin</th>
<th>Name</th>
<th>Pin</th>
<th>Name</th>
<th>Pin</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GND</td>
<td>2</td>
<td>-12</td>
<td>33</td>
<td>FRAME#</td>
<td>34</td>
<td>GND</td>
</tr>
<tr>
<td>3</td>
<td>GND</td>
<td>4</td>
<td>INTC#</td>
<td>35</td>
<td>GND</td>
<td>36</td>
<td>IRDY#</td>
</tr>
<tr>
<td>5</td>
<td>INTD#</td>
<td>6</td>
<td>INTA#</td>
<td>37</td>
<td>DEVSEL#</td>
<td>38</td>
<td>+5 V</td>
</tr>
<tr>
<td>7</td>
<td>BMODE1A</td>
<td>8</td>
<td>+5 V</td>
<td>39</td>
<td>GND</td>
<td>40</td>
<td>LOCK#</td>
</tr>
<tr>
<td>9</td>
<td>INTB#</td>
<td>10</td>
<td>NC</td>
<td>41</td>
<td>SDONE#</td>
<td>42</td>
<td>NC</td>
</tr>
<tr>
<td>11</td>
<td>GND</td>
<td>12</td>
<td>NC</td>
<td>43</td>
<td>PAR</td>
<td>44</td>
<td>GND</td>
</tr>
<tr>
<td>13</td>
<td>CLK</td>
<td>14</td>
<td>GND</td>
<td>45</td>
<td>+5 V</td>
<td>46</td>
<td>AD[15]</td>
</tr>
<tr>
<td>17</td>
<td>REQ#</td>
<td>18</td>
<td>+5 V</td>
<td>49</td>
<td>AD[9]</td>
<td>50</td>
<td>+5 V</td>
</tr>
<tr>
<td>19</td>
<td>+5 V</td>
<td>20</td>
<td>AD[31]</td>
<td>51</td>
<td>GND</td>
<td>52</td>
<td>C/BE3#</td>
</tr>
<tr>
<td>25</td>
<td>GND</td>
<td>26</td>
<td>C/BE3#</td>
<td>57</td>
<td>+5 V</td>
<td>58</td>
<td>AD[3]</td>
</tr>
<tr>
<td>29</td>
<td>AD[19]</td>
<td>30</td>
<td>+5 V</td>
<td>61</td>
<td>AD[0]</td>
<td>62</td>
<td>+5 V</td>
</tr>
<tr>
<td>31</td>
<td>+5 V</td>
<td>32</td>
<td>AD[17]</td>
<td>63</td>
<td>GND</td>
<td>64</td>
<td>REQ64#</td>
</tr>
</tbody>
</table>
## PMC J6 Connector and Pinout

![PMC J6 Connector Image](image)

### Table A-2 PMC J6 Connector Pinout

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Pin</th>
<th>Name</th>
<th>Pin</th>
<th>Name</th>
<th>Pin</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+12 V</td>
<td>2</td>
<td>+5 V</td>
<td>33</td>
<td>GND</td>
<td>34</td>
<td>NC</td>
</tr>
<tr>
<td>3</td>
<td>GND</td>
<td>4</td>
<td>NC</td>
<td>35</td>
<td>TRDY</td>
<td>36</td>
<td>+3.3 V</td>
</tr>
<tr>
<td>5</td>
<td>+5 V</td>
<td>6</td>
<td>GND</td>
<td>37</td>
<td>GND</td>
<td>38</td>
<td>STOP#</td>
</tr>
<tr>
<td>7</td>
<td>GND</td>
<td>8</td>
<td>NC</td>
<td>39</td>
<td>PERR#</td>
<td>40</td>
<td>GND</td>
</tr>
<tr>
<td>9</td>
<td>NC</td>
<td>10</td>
<td>NC</td>
<td>41</td>
<td>+3.3 V</td>
<td>42</td>
<td>SERR#</td>
</tr>
<tr>
<td>11</td>
<td>PRSNT2</td>
<td>12</td>
<td>+3.3 V</td>
<td>43</td>
<td>C/BE1#</td>
<td>44</td>
<td>GND</td>
</tr>
<tr>
<td>15</td>
<td>+3.3 V</td>
<td>16</td>
<td>GND</td>
<td>47</td>
<td>GND</td>
<td>48</td>
<td>AD[10]</td>
</tr>
<tr>
<td>17</td>
<td>NC</td>
<td>18</td>
<td>GND</td>
<td>49</td>
<td>AD[8]</td>
<td>50</td>
<td>+3.3 V</td>
</tr>
<tr>
<td>21</td>
<td>GND</td>
<td>22</td>
<td>AD[26]</td>
<td>53</td>
<td>+3.3 V</td>
<td>54</td>
<td>NC</td>
</tr>
<tr>
<td>23</td>
<td>AD[24]</td>
<td>24</td>
<td>+3.3 V</td>
<td>55</td>
<td>NC</td>
<td>56</td>
<td>GND</td>
</tr>
<tr>
<td>25</td>
<td>IDSEL</td>
<td>26</td>
<td>AD[23]</td>
<td>57</td>
<td>NC</td>
<td>58</td>
<td>NC</td>
</tr>
<tr>
<td>27</td>
<td>+3.3 V</td>
<td>28</td>
<td>AD[20]</td>
<td>59</td>
<td>GND</td>
<td>60</td>
<td>NC</td>
</tr>
<tr>
<td>29</td>
<td>AD[18]</td>
<td>30</td>
<td>GND</td>
<td>61</td>
<td>ACK64#</td>
<td>62</td>
<td>+3.3 V</td>
</tr>
<tr>
<td>31</td>
<td>AD[16]</td>
<td>32</td>
<td>C/BE2#</td>
<td>63</td>
<td>GND</td>
<td>64</td>
<td>NC</td>
</tr>
</tbody>
</table>
### PMC J4 Connector and Pinout

A visual representation of the PMC J4 connector is shown, indicating the physical layout of the pins. The diagram illustrates the connector with numbers marking each pin.

#### Table A-3 PMC J4 Connector Pinout

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Pin</th>
<th>Name</th>
<th>Pin</th>
<th>Name</th>
<th>Pin</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>J5_PMC_E22_1</td>
<td>2</td>
<td>J5_PMC_D22_2</td>
<td>33</td>
<td>J5_PMC_C16_33</td>
<td>34</td>
<td>J5_PM_C16_34</td>
</tr>
<tr>
<td>3</td>
<td>J5_PM_C22_3</td>
<td>4</td>
<td>J5_PM_B22_4</td>
<td>35</td>
<td>J5_PM_A16_35</td>
<td>36</td>
<td>J5_PM16_36</td>
</tr>
<tr>
<td>5</td>
<td>J5_PM_A22_5</td>
<td>6</td>
<td>J5_PM_E21_6</td>
<td>37</td>
<td>J5_PM_D15_37</td>
<td>38</td>
<td>J5_PM C15_38</td>
</tr>
<tr>
<td>7</td>
<td>J5_PM D21_7</td>
<td>8</td>
<td>J5_PM_C21_8</td>
<td>39</td>
<td>J5_PM_B15_39</td>
<td>40</td>
<td>J5_PM A15_40</td>
</tr>
<tr>
<td>9</td>
<td>J5_PM B21_9</td>
<td>10</td>
<td>J5_PM_A21_10</td>
<td>41</td>
<td>J5_PM E14_41</td>
<td>42</td>
<td>J5_PM D14_42</td>
</tr>
<tr>
<td>11</td>
<td>J5_PM E20_11</td>
<td>12</td>
<td>J5_PM D20_12</td>
<td>43</td>
<td>J5_PM C14_43</td>
<td>44</td>
<td>J5_PM B14_44</td>
</tr>
<tr>
<td>13</td>
<td>J5_PM C20_13</td>
<td>14</td>
<td>J5_PM B20_14</td>
<td>45</td>
<td>J5_PM A14_45</td>
<td>46</td>
<td>J5_PM E13_46</td>
</tr>
<tr>
<td>15</td>
<td>J5_PM A20_15</td>
<td>16</td>
<td>J5_PM E19_16</td>
<td>47</td>
<td>J5_PM D13_47</td>
<td>48</td>
<td>J5_PM C13_48</td>
</tr>
<tr>
<td>17</td>
<td>J5_PM D19_17</td>
<td>18</td>
<td>J5_PM C19_18</td>
<td>49</td>
<td>J5_PM B13_49</td>
<td>50</td>
<td>J5_PM A13_50</td>
</tr>
<tr>
<td>19</td>
<td>J5_PM B19_19</td>
<td>20</td>
<td>J5_PM A19_20</td>
<td>51</td>
<td>J5_PM E12_51</td>
<td>52</td>
<td>J5_PM D12_52</td>
</tr>
<tr>
<td>21</td>
<td>J5_PM E18_21</td>
<td>22</td>
<td>J5_PM D18_22</td>
<td>53</td>
<td>J5_PM C12_53</td>
<td>54</td>
<td>J5_PM B12_54</td>
</tr>
<tr>
<td>23</td>
<td>J5_PM C18_23</td>
<td>24</td>
<td>J5_PM B18_24</td>
<td>55</td>
<td>J5_PM A12_55</td>
<td>56</td>
<td>J5_PM E11_56</td>
</tr>
<tr>
<td>25</td>
<td>J5_PM A18_25</td>
<td>26</td>
<td>J5_PM E17_26</td>
<td>57</td>
<td>J5_PM D11_57</td>
<td>58</td>
<td>J5_PM C11_58</td>
</tr>
<tr>
<td>27</td>
<td>J5_PM D17_27</td>
<td>28</td>
<td>J5_PM C17_28</td>
<td>59</td>
<td>J5_PM B11_59</td>
<td>60</td>
<td>J5_PM A11_60</td>
</tr>
<tr>
<td>29</td>
<td>J5_PM B17_29</td>
<td>30</td>
<td>J5_PM A17_30</td>
<td>61</td>
<td>J5_PM E10_61</td>
<td>62</td>
<td>J5_PM D10_62</td>
</tr>
<tr>
<td>31</td>
<td>J5_PM E16_31</td>
<td>32</td>
<td>J5_PM D16_32</td>
<td>63</td>
<td>J5_PM C10_63</td>
<td>64</td>
<td>J5_PM B10_64</td>
</tr>
</tbody>
</table>
System Driver Software

Contents

Windows for Workgroups (Version 3.11) ............................................. 96
Windows 95 .................................................................................. 98
Windows NT (Version 4.0) ............................................................... 102

Introduction

The VMICPCI-7696 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-7696 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-7696 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-7696, 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.

3. If you do not require LAN operation, click ‘CONTINUE’ and skip steps 3 through 12.

4. 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.


6. Then click ‘OK’.

7. 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’.

8. Again at the ‘NETWORK SETUP’ screen, click ‘CONTINUE’.

9. Under ‘ADD NETWORK ADAPTER’ click on ‘UNLISTED or UPDATED NETWORK ADAPTER’. Then click ‘OK’.

10. Insert the VMICPCI-7696 distribution disk marked 320-500039-004 into drive A: and type: A:\WFW311\,. Then click ‘OK’.

11. Under ‘UNLISTED’ or ‘UPDATED NETWORK ADAPTER’, choose ‘Intel 21143-based 10/100 mpbs Ethernet Controller (NDIS31)’. Then click ‘OK’.

12. Under ‘MICROSOFT WINDOWS NETWORK NAMES’, enter the network names you want for computer name, group name, etc. Click ‘OK’.

13. 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.

14. 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.

15. Insert the VMICPCI-7696 distribution diskette marked 320-500039-001 into drive A:

16. From the Program Manager Screen double-click the mouse on the ‘Windows Setup’ icon.

17. 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:\32bit\ 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
collection type for your system.

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

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.

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 marked 320-500039-004 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’.
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 320-500039-002 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(TM) Chipsets

This update allows the operating system to correctly identify the Intel(TM) 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 marked 320-50039-003 into drive A: and run A:\95\INF_UP\SETUP.EXE.
4. Click Next on 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 when 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 DE-INSTALLED. Insert the Diskette marked 320-500039-003 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(r) 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. Place a dot next to ‘THIS COMPUTER WILL PARTICIPATE ON A NETWORK’.

3. Place a check mark 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 320-500039-004 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 on ‘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 ‘MY COMPUTER’ window.


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 320-500039-002 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 on ‘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 on ‘OK’.

27. On the ‘DISPLAY PROPERTIES’ screen click on ‘SETTINGS’, then click on ‘TEST’.

28. The ‘TESTING MODE’ screen will be displayed. Click on ‘OK’. If the bitmap test image is displayed correctly, click on ‘OK’.

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

Contents

System BIOS Setup Utility ..................................................... 106
BIOS Features Setup .......................................................... 107
LANWorks BIOS Setup ....................................................... 108

Introduction

The VMICPCI-7696 includes the LANWorks option which allows the VMICPCI-7696 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-7696 and when prompted, press the Delete key to access the System BIOS Setup Utility screen shown below.

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

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

When prompted, press “Control-Alt-B” as the VMICPCI-7696 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.

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
Introduction

The VMICPCI-7696 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-7696 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.
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.

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-7696 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-7696.
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-7696 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-7696 does not support a second floppy drive. The default is None.

Video

The VMICPCI-7696 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.
This screen, selected from the CMOS Setup Utility screen, allows the user to configure options that are in addition to the basic BIOS features.

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 enabled 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-7696 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 is enabled this 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 is enabled this 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.

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 AGP 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.

** ACPI Function **

<table>
<thead>
<tr>
<th>Option</th>
<th>Setting</th>
<th>** Reload Global Timer Events **</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACPI Function</td>
<td>Disabled</td>
<td>IRQ [3-7,9-15],NMI : Disabled</td>
</tr>
<tr>
<td>Power Management</td>
<td>User Defined</td>
<td>Primary IDE 0 : Disabled</td>
</tr>
<tr>
<td>PM Control by APM</td>
<td>Yes</td>
<td>Primary IDE 1 : Disabled</td>
</tr>
<tr>
<td>Video-off After</td>
<td>Standby</td>
<td>Secondary IDE 0 : Disabled</td>
</tr>
<tr>
<td>Modem Use IRQ</td>
<td>3</td>
<td>Secondary IDE 1 : Disabled</td>
</tr>
<tr>
<td>Doze Mode</td>
<td>Disabled</td>
<td>Floppy disk : Disabled</td>
</tr>
<tr>
<td>Standby Mode</td>
<td>Disabled</td>
<td>Serial Port : Enabled</td>
</tr>
<tr>
<td>Suspend Mode</td>
<td>Disabled</td>
<td>Parallel : Disabled</td>
</tr>
<tr>
<td>HDD Power Down</td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td>Throttle Duty Cycle</td>
<td>62.5%</td>
<td></td>
</tr>
<tr>
<td>VGA Active Monitor</td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td>Resume By Ring</td>
<td>Enabled</td>
<td></td>
</tr>
<tr>
<td>Resume By Alarm</td>
<td>Disabled</td>
<td></td>
</tr>
<tr>
<td>Wake Up On LAN</td>
<td>Enabled</td>
<td></td>
</tr>
<tr>
<td>CPU Clock Throttle</td>
<td>Enabled</td>
<td></td>
</tr>
<tr>
<td>IRQ 8 Break Suspend</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

** 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.

**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 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.

<table>
<thead>
<tr>
<th>IDE HDD Block Mode</th>
<th>Enabled</th>
<th>Onboard Parallel Port</th>
<th>378/IRQ7</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDE Primary Master PIO</td>
<td>Auto</td>
<td>Parallel Port Mode</td>
<td>Normal</td>
</tr>
<tr>
<td>IDE Primary Slave PIO</td>
<td>Auto</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDE Primary Master UDMA</td>
<td>Auto</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDE Secondary Master UDMA</td>
<td>Auto</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDE Secondary Slave UDMA</td>
<td>Auto</td>
<td></td>
<td></td>
</tr>
<tr>
<td>On-Chip Primary PCI IDE</td>
<td>Enabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>On-Chip Secondary PCI IDE</td>
<td>Disabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Onboard PCI SCSI Chip</td>
<td>Enabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USB Keyboard Support</td>
<td>Disabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Init Display First</td>
<td>PCI Slot</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Onboard FDC Controller</td>
<td>Enabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Onboard Serial Port 1</td>
<td>Auto</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Onboard Serial Port 2</td>
<td>Auto</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UART2 Mode</td>
<td>Standard</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

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-7696. 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.
Device Configuration: I/O and Interrupt Control

Contents

BIOS Operations ......................................................... 130
Device Address Definition ........................................... 135
Device Interrupt Definition ........................................... 137

Introduction

This appendix provides the user with the information needed to develop custom applications for the VMICPCI-7696. The CPU board on the VMICPCI-7696 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-7696 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.
When the VMICPCI-7696 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.

There are two areas on the VMICPCI-7696 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-7696 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-7696.

The block diagram included in Figure E-1 on page 131 illustrates the VMICPCI-7696 emphasizing the I/O features, including the PCI-to-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-7696 Block Diagram
Data Book References

1. Intel Pentium® II Processor at 233 MHz, 266 MHz, 300 MHz, and 333MHz
   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 21153 PCI-to-PCI Bridge
   September 1998, Order Number: 278220-001
   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 82430TX 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

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 from:
   
   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 from:
   
   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-7696 Specification
   
   800-657696-000 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-7696 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-7696 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>
<thead>
<tr>
<th>Device</th>
<th>Memory Space</th>
<th>I/O Address Space</th>
<th>PIC Interrupt Options</th>
<th>Byte Address Boundary</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Floppy</td>
<td>N/A</td>
<td>[0x100 - 0xFF8]</td>
<td>IRQ1 - IRQ15</td>
<td>8</td>
<td>$3F0</td>
</tr>
<tr>
<td>Parallel Port</td>
<td>N/A</td>
<td>[0x100 - 0xFFC]</td>
<td>IRQ1 - IRQ15</td>
<td>4</td>
<td>$378</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[0x100 - 0xFF8]</td>
<td></td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Serial Port 1</td>
<td>N/A</td>
<td>[0x100 - 0xFF8]</td>
<td>IRQ1 - IRQ15</td>
<td>8</td>
<td>$3F8</td>
</tr>
<tr>
<td>Serial Port 2</td>
<td>N/A</td>
<td>[0x100 - 0xFF8]</td>
<td>IRQ1 - IRQ15</td>
<td>8</td>
<td>$2F8</td>
</tr>
<tr>
<td>Real-Time Clock</td>
<td>Nonrelocatable</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keyboard</td>
<td>Nonrelocatable</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Auxiliary I/O</td>
<td>N/A</td>
<td>[0x000 - 0xFFF]</td>
<td>IRQ1, IRQ3-IRQ15</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programmable Timer</td>
<td>Nonrelocatable</td>
<td></td>
<td>IRQ5</td>
<td>4</td>
<td>0X500</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Watchdog Timer</td>
<td>Nonrelocatable</td>
<td></td>
<td>N/A</td>
<td>0XD</td>
<td>0XD8000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Battery-Backed SRAM</td>
<td>Nonrelocatable</td>
<td></td>
<td>N/A</td>
<td>(32K-0XD)</td>
<td>0XD800E</td>
</tr>
</tbody>
</table>

Table E-1 ISA Device Mapping Configuration
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-7696 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

<table>
<thead>
<tr>
<th>Device</th>
<th>Memory Space</th>
<th>I/O Address Space</th>
<th>PIC Interrupt Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI-to-PCI Bridge</td>
<td>Anywhere</td>
<td>N/A</td>
<td>PCI Defined</td>
</tr>
<tr>
<td>Ethernet</td>
<td>Anywhere</td>
<td>Anywhere</td>
<td>PCI Defined</td>
</tr>
<tr>
<td>Timer Interrupt Registers</td>
<td>N/A</td>
<td>Fixed</td>
<td>N/A</td>
</tr>
<tr>
<td>PMC</td>
<td>Anywhere</td>
<td>Anywhere</td>
<td>PCI Defined</td>
</tr>
</tbody>
</table>
Device Interrupt Definition

PC/AT Interrupt Definition

The interrupt hardware implementation on the VMICPCI-7696 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 138 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-7696 BIOS maps the IRQx lines to the appropriate device per the standard ISA architecture. Reference Figure E-2 on page 138. This initialization operation cannot be changed; however, a custom application could reroute the interrupt configuration after the BIOS has completed the initial configuration cycle.
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 138 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

<table>
<thead>
<tr>
<th>DEVICE COMPONENT</th>
<th>VENDOR ID</th>
<th>DEVICE ID</th>
<th>CPU ADDRESS</th>
<th>DEVICE PCI INTERRUPT</th>
<th>MOTHERBOARD PCI INTERRUPT MAPPER</th>
<th>DATA BOOK REF. #</th>
<th>Arbitration Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Management</td>
<td>PIIX4 82371EB Function</td>
<td>0x8086</td>
<td>0x7113</td>
<td>AD18</td>
<td>N/A</td>
<td>N/A</td>
<td>5</td>
</tr>
<tr>
<td>PCI-to-ISA Bridge</td>
<td>PIIX4 82371EB Function 00</td>
<td>0x8086</td>
<td>0x7000</td>
<td>AD18</td>
<td>N/A</td>
<td>N/A</td>
<td>5</td>
</tr>
<tr>
<td>Ethernet Controller</td>
<td>DEC 21140</td>
<td>0x1011</td>
<td>0x0019</td>
<td>AD22</td>
<td>INTA</td>
<td>PIRQ1</td>
<td>4</td>
</tr>
<tr>
<td>PCI IDE Controller</td>
<td>PIIX4 82371EB Function 01</td>
<td>0x8086</td>
<td>0x7111</td>
<td>AD18</td>
<td>N/A</td>
<td>N/A</td>
<td>5</td>
</tr>
<tr>
<td>PCI Host Bridge</td>
<td>Intel 440BX</td>
<td>0x8086</td>
<td>0x7190</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>2</td>
</tr>
<tr>
<td>PMC</td>
<td>N/A</td>
<td>N/A</td>
<td>AD31</td>
<td>INTA</td>
<td>PIRQ2</td>
<td>14</td>
<td>REQ3</td>
</tr>
<tr>
<td>Compact PCI Bridge</td>
<td>DEC 21153</td>
<td>0x1011</td>
<td>0x0258</td>
<td>AD26</td>
<td>N/A</td>
<td>N/A</td>
<td>3</td>
</tr>
</tbody>
</table>

** 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

<table>
<thead>
<tr>
<th>PCI INTx</th>
<th>PIC IRQx</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIRQ0</td>
<td>IRQ9</td>
</tr>
<tr>
<td>PIRQ1</td>
<td>IRQ9</td>
</tr>
<tr>
<td>PIRQ2</td>
<td>IRQ10</td>
</tr>
<tr>
<td>PIRQ3</td>
<td>IRQ11</td>
</tr>
</tbody>
</table>

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.

⚠️ 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.
Sample C Software

Contents

Directory SRAM ................................................................. 142
**File: T_SRAM.C .......................................................... 142
Directory Timers ................................................................. 145
**File: CPU.H ................................................................. 145
** File: PCI.H ................................................................. 147
**File: PCI.C ................................................................. 149
**File: T_Timers.C .......................................................... 152
**File: Timer.C ............................................................... 156
Directory WATCHDOG ......................................................... 161
**File: Watchdog_H ......................................................... 161
**File: WDT0_RST.C ......................................................... 162

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-7696 user on disk 320-500039-xxx, Sample Application C Code for the VMICPCI-7696, included in the distribution disk set.

Because of the wide variety of environments in which the VMICPCI-7696 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.
**File: T_SRAM.C**

```c
#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 .............\n");
  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-7696'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

```c
/*
**  FILE: T7696T.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 Select counter 0 */
#define     CW_SC1          0x40    /* W Select counter 1 */
#define     CW_SC2          0x80    /* W Select counter 2 */
#define     CW_RBC          0xC0    /* W Read back command */
#define     CW_CLC          0x00    /* W Cntr latch command (cnt/stat) */
#define     CW_SCL          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**

```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;
_BLE = 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**

```
#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 */
#define do_exit( int )
/* TIMERS.C function prototypes */
#define far interrupt irq_rcvd( void )
#define init_timer_int( void )
#define restore_orig_int( void )
#define load_counter( int, unsigned int )
#define 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("Unable 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("Unable 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 ....\n");
/* 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 */

void restore_orig_int( void )
{
    disable();
    outp(0x21, pic1_org);
    setvect( IRQ5, old_vect );
    enable();
} /* restore_orig_int */

void load_counter( void )
{
    /* purpose: Loads the appropriate counter with the count passed */
    /* */
    /* */
    /* Parameters: none */
    /* */
    /* 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() */

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 ) );
    } /* read_counter */
}
*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 7696. */
/* */
/* */
/**************************************************************************/
/* 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-7696's watchdog timer function as described in Chapter 4.

**File: Watchdog.H**

/*
** DS1384 REGISTER OFFSETS
*/

#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 */
#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 REGISTER 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 alaram flag */
#define WD_TDF 0x01 /* time-of-day flag */
**File: WDT0_RST.C**

```c
#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;

char usr[80];

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", ctime(&t) );
    do {
        time(&t);
        printf("CURRENT DATE & TIME: %.24s", ctime(&t) );
        delay( 250 );
    } while( !kbhit() );
} /* end main */
```