ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Emerson / Motorola MVME1600-011* at our website: ***Click HERE***

MVME1603/MVME1604 Single Board Computer Programmer's Reference Guide

V1600-1A/PGX

Early Access

This Early Access document is a preliminary draft of a manual that is not complete and may not have been technically reviewed or approved. All material herein is subject to change. Some sections may be incomplete or not yet available.

You will receive a copy of the released, printed manual with your product upgrade.

We welcome your comments and suggestions about the documentation. Please mail your comments using the form included with this manual, or send your comments via e-mail to pcdocs@phx.mcd.mot.com, or FAX them to us at (602) 438-3534. Please be sure to include the part number and date shown on this document.

April 13, 1998

PRELIMINARY

Notice

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

No part of this material may be reproduced or copied in any tangible medium, or stored in a retrieval system, or transmitted in any form, or by any means, radio, electronic, mechanical, photocopying, recording or facsimile, or otherwise, without the prior written permission of Motorola, Inc.

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

Restricted Rights Legend

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

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

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

Preface

The *MVME1603/MVME1604 Single Board Computer Programmer's Reference Guide* provides brief board level information, complete memory maps, and detailed ASIC chip information including register bit descriptions for the XVME160x Single Board Computers (called MVME1603 and MVME1604 in this manual). The information contained in this manual applies to the single board computers built from some of the plug-together components listed in the following table.

MVME1600-001A	PM603-001A	PM604-001A	RAM104-001A
MVME1600-011A	PM603-002A	PM604-002A	RAM104-002A
	PM603-003A	PM604-003A	RAM104-003A
	PM603-004A	PM604-004A	RAM104-004A

This manual is intended for anyone who wants to program these boards in order to design OEM systems, supply additional capability to an existing compatible system, or work in a lab environment for experimental purposes.

A basic knowledge of computers and digital logic is assumed.

To use this manual, you should be familiar with the publications listed in *Appendix A, Related Documentation*.

The following conventions are used in this document:

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

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

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

<RETURN> or **<CR>**
: represents the carriage return key.

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

The computer programs stored in the Read Only Memory of this device contain material copyrighted by Motorola Inc., first published 1990, and may be used only under a license such as the License for Computer Programs (Article 14) contained in Motorola's Terms and Conditions of Sale, Rev. 1/79.

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



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

Motorola® and the Motorola symbol are registered trademarks of Motorola, Inc.

PowerStack™, VMEexec, VMEmodule™, and VMEsystem™ are trademarks of Motorola, Inc.

PowerPC™, PowerPC 603™, and PowerPC 604™are trademarks of IBM Corp, and are used by Motorola, Inc. under license from IBM Corp.

AIX™ is a trademark of IBM Corp.

Timekeeper™ and Zeropower™ are trademarks of Thompson Components.

All other products mentioned in this document are trademarks or registered trademarks of their respective holders.

©Copyright Motorola 1994

All Rights Reserved

Printed in the United States of America

November 1994

Safety Summary
Safety Depends On You

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

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

Ground the Instrument.

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

Do Not Operate in an Explosive Atmosphere.

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

Keep Away From Live Circuits.

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

Do Not Service or Adjust Alone.

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

Use Caution When Exposing or Handling the CRT.

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

Do Not Substitute Parts or Modify Equipment.

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

Dangerous Procedure Warnings.

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



WARNING

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

Contents

Introduction 1-1
Manual Terminology 1-1
Overview 1-3
Features 1-3
Block Diagram 1-4
Functional Description 1-6
Programming Model 1-6
- Memory Maps 1-6
  - Processor View of the Memory Map 1-7
  - PCI View of the Memory Map 1-12
- Control and Status Registers (CSR) 1-12
  - CPU Configuration Register 1-13
  - Software Readable Header J8 or J14 1-14
  - Board Configuration Register 1-15
  - DRAM Size Register 1-16
- Z8536 CIO Port Pins 1-18
  - Serial Port Module ID 1-19
- VMEchip2 Memory Maps 1-19
- BBRAM,TOD Clock Memory Map 1-27

Introduction 2-1
- Summary of Major Features 2-1

Functional Blocks 2-4
- Local Bus to VMEbus Interface 2-4
  - Local Bus to VMEbus Requester 2-8
- VMEbus to Local Bus Interface 2-9
- Local Bus to VMEbus DMA Controller 2-11
  - DMAC VMEbus Requester 2-13
- Tick and Watchdog Timers 2-14
  - Prescaler 2-14
  - Tick Timer 2-14
  - Watchdog Timer 2-15
- VMEbus Interrupter 2-15
- VMEbus System Controller 2-16
  - Arbiter 2-17
  - IACK Daisy-Chain Driver 2-17
  - Bus Timer 2-17

Reset Driver 2-18
Local Bus Interrupter and Interrupt Handler 2-18
Global Control and Status Registers 2-20
LCSR Programming Model 2-20
Programming the VMEbus Slave Map Decoders 2-26
VMEbus Slave Ending Address Register 1 2-28
VMEbus Slave Starting Address Register 1 2-28
VMEbus Slave Ending Address Register 2 2-29
VMEbus Slave Starting Address Register 2 2-29
VMEbus Slave Address Translation Address Offset Register 1 2-29
VMEbus Slave Address Translation Select Register 1 2-30
VMEbus Slave Address Translation Address Offset Register 2 2-31
VMEbus Slave Address Translation Select Register 2 2-31
VMEbus Slave Write Post and Snoop Control Register 2 2-32
VMEbus Slave Address Modifier Select Register 2 2-32
VMEbus Slave Write Post and Snoop Control Register 1 2-34
VMEbus Slave Address Modifier Select Register 1 2-34
Programming the Local Bus to VMEbus Map Decoders 2-35
Local Bus Slave (VMEbus Master) Ending Address Register 1 2-38
Local Bus Slave (VMEbus Master) Starting Address Register 1 2-38
Local Bus Slave (VMEbus Master) Ending Address Register 2 2-38
Local Bus Slave (VMEbus Master) Starting Address Register 2 2-39
Local Bus Slave (VMEbus Master) Ending Address Register 3 2-39
Local Bus Slave (VMEbus Master) Starting Address Register 3 2-39
Local Bus Slave (VMEbus Master) Ending Address Register 4 2-40
Local Bus Slave (VMEbus Master) Starting Address Register 4 2-40
Local Bus Slave (VMEbus Master) Address Translation Address Register 4 2-40
Local Bus Slave (VMEbus Master) Address Translation Select Register 4 2-41
Local Bus Slave (VMEbus Master) Attribute Register 4 2-41
Local Bus Slave (VMEbus Master) Attribute Register 3 2-42
Local Bus Slave (VMEbus Master) Attribute Register 2 2-42
Local Bus Slave (VMEbus Master) Attribute Register 1 2-43
VMEbus Slave GCSR Group Address Register 2-44
VMEbus Slave GCSR Board Address Register 2-44
Local Bus To VMEbus Enable Control Register 2-45
Local Bus To VMEbus I/O Control Register 2-46
ROM Control Register 2-47
Programming the VMEchip2 DMA Controller 2-48
DMAC Registers 2-49
PROM Decoder, SRAM and DMA Control Register 2-49

Local Bus To VMEbus Requester Control Register 2-50
DMAC Control Register 1 (bits 0-7) 2-51
DMAC Control Register 2 (bits 8-15) 2-53
DMAC Control Register 2 (bits 0-7) 2-54
DMAC Local Bus Address Counter 2-55
DMAC VMEbus Address Counter 2-55
DMAC Byte Counter 2-56
Table Address Counter 2-56
VMEbus Interrupter Control Register 2-56
VMEbus Interrupter Vector Register 2-57
MPU Status and DMA Interrupt Count Register 2-58
DMAC Status Register 2-59
Programming the Tick and Watchdog Timers 2-59
VMEbus Arbiter Time-out Control Register 2-60
DMAC Ton/Toff Timers and VMEbus Global Time-out Control Register 2-60
VME Access, Local Bus and Watchdog Time-out Control Register 2-61
Prescaler Control Register 2-62
Tick Timer 1 Compare Register 2-63
Tick Timer 1 Counter 2-64
Tick Timer 2 Compare Register 2-64
Tick Timer 2 Counter 2-65
Board Control Register 2-65
Watchdog Timer Control Register 2-66
Tick Timer 2 Control Register 2-67
Tick Timer 1 Control Register 2-68
Prescaler Counter 2-68
Programming the Local Bus Interrupter 2-69
Local Bus Interrupter Status Register (bits 24-31) 2-72
Local Bus Interrupter Status Register (bits 16-23) 2-72
Local Bus Interrupter Status Register (bits 8-15) 2-73
Local Bus Interrupter Status Register (bits 0-7) 2-74
Local Bus Interrupter Enable Register (bits 24-31) 2-74
Local Bus Interrupter Enable Register (bits 16-23) 2-75
Local Bus Interrupter Enable Register (bits 8-15 2-76
Local Bus Interrupter Enable Register (bits 0-7) 2-77
Software Interrupt Set Register (bits 8-15) 2-78
Interrupt Clear Register (bits 24-31) 2-78
Interrupt Clear Register (bits 16-23) 2-79
Interrupt Clear Register (bits 8-15) 2-80
Interrupt Level Register 1 (bits 24-31) 2-80
Interrupt Level Register 1 (bits 16-23) 2-81

Interrupt Level Register 1 (bits 8-15) 2-81
Interrupt Level Register 1 (bits 0-7) 2-82
Interrupt Level Register 2 (bits 24-31) 2-82
Interrupt Level Register 2 (bits 16-23) 2-83
Interrupt Level Register 2 (bits 8-15) 2-83
Interrupt Level Register 2 (bits 0-7) 2-84
Interrupt Level Register 3 (bits 24-31) 2-84
Interrupt Level Register 3 (bits 16-23) 2-85
Interrupt Level Register 3 (bits 8-15) 2-85
Interrupt Level Register 3 (bits 0-7) 2-86
Interrupt Level Register 4 (bits 24-31) 2-86
Interrupt Level Register 4 (bits 16-23) 2-87
Interrupt Level Register 4 (bits 8-15) 2-87
Interrupt Level Register 4 (bits 0-7) 2-88
Vector Base Register 2-88
I/O Control Register 1 2-89
I/O Control Register 2 2-90
I/O Control Register 3 2-91
Miscellaneous Control Register 2-91
GCSR Programming Model 2-93
Programming the GCSR 2-95
VMEchip2 Revision Register 2-97
VMEchip2 ID Register 2-97
VMEchip2 LM/SIG Register 2-97
VMEchip2 Board Status/Control Register 2-99
General Purpose Register 0 2-99
General Purpose Register 1 2-100
General Purpose Register 2 2-100
General Purpose Register 3 2-101
General Purpose Register 4 2-101
General Purpose Register 5 2-102
Introduction 3-1
Major Featuresof VME2PCI ASIC 3-1
VME2PCI ASIC Programming Model 3-2
VME2PCI Configuration Registers 3-3
Vendor ID Register 3-6
Device ID Register 3-6
PCI Command Register 3-7
PCI Status Register 3-8
Revision ID Register 3-9
PCI Class Code Register 3-9

PCI Cache Line Size Register 3-10
PCI Latency Timer Register 3-11
PCI Header Type Register 3-11
Built-In-Self-Test Register 3-12
PCI IO Map Base Register 3-12
PCI Memory Map Base Register 3-13
Interrupt Line Register 3-14
PCI Interrupt Pin Register 3-15
PCI Minimum Grant Register 3-15
PCI Maximum Latency Register 3-16
PCI Slave Start Address 1 Register 3-16
PCI Slave End Address 1 Register 3-17
PCI Slave Address Offset 1 Register 3-17
PCI Slave Control 1 Register 3-18
PCI Slave Start Address 2 Register 3-18
PCI Slave End Address 2 Register 3-19
PCI Slave Address Offset 2 Register 3-19
PCI Slave Control 2 Register 3-20
Interrupt Control and Status Register 3-20
VMEchip2 CSR Window 3-21
VME2PCI IACK Registers 3-21
VMEbus Mapping 3-22
Processor's Point of View 3-22
VMEbus Point of View 3-24
Introduction 4-1
PCI Arbitration 4-1
Interrupt Supports 4-4
Machine Check Interrupt (MCP_) 4-5
Maskable Interrupts 4-5
Handling VMEchip2 Interrupts 4-8
Handling Z8536 and Z85230 Interrupts 4-8
ABT (abort) Interrupt 4-9
DMA Channels 4-9
Sources of Reset 4-10
Endian Issues 4-10
Processor/Memory Domain 4-13
MPC105's Involvement 4-13
PCI Domain 4-13
53C825 or 53C810(SCSI) 4-13
Am79C970 (Ethernet) 4-14
GD5434 (Graphics) 4-14

VME2PCI's Involvement 4-14
VMEbus Domain 4-14
Motorola Computer Group Documents A-1
Manufacturers' Documents A-2
Related Specifications A-7
Abbreviations, Acronyms, and Terms to Know GL-1

List of Figures

MVME1603/MVME1604 Block Diagram 1-5
VMEchip2 Block Diagram 2-5
VME2PCI's CSR Mapping in PCI Memory Space 3-2
VME2PCI's CSR Mapping in PCI I/O Space 3-3
VME2PCI Configuration Registers 3-4
VMEbus Mapping Example 3-23
IBC Arbiter Configuration Diagram 4-2
MVME1603/MVME1604 Interrupt Architecture 4-4
IBC Interrupt Handler Block Diagram 4-6
Big-Endian Mode 4-11
Little-Endian Mode 4-12

List of Tables

Processor View of the Memory Map 1-7
PCI Configuration Space Map 1-8
ISA I/O Space Map 1-9
PCI View of the PCI Memory Map 1-12
Z8536 CIO Port Pins Assignment 1-18
Interpretation of MID3-MID0 1-19
VMEchip2 Memory Map (Sheet 1 of 3) 1-20
VMEchip2 Memory Map (Sheet 2 of 3) 1-22
VMEchip2 Memory Map (Sheet 3 of 3) 1-25
M48T18 BBRAM/TOD Clock Memory Map 1-25
BBRAM Configuration Area Memory Map 1-26
TOD Clock Memory Map 1-26
VMEchip2 Memory Map - LCSR Summary (Sheet 1 of 2) 2-22
VMEchip2 Memory Map - LCSR Summary (Sheet 2 of 2) 2-24
DMAC Command Table Format 2-49
Local Bus Interrupter Summary 2-70
VMEchip2 Memory Map (GCSR Summary) 2-96
VME2PCI ASIC's PCI Configuration Registers Summary 3-5
VME2PCI ASIC's IACK Registers 3-22
PCI Arbitration Assignments 4-2
Arbitration Priority in Fixed Mode 4-3
IBC PCI/ISA Interrupt Assignments 4-7
IBC DMA Channel Assignments 4-9
Glossary of Terms GL-1

Board Description and Memory Maps

1

Introduction

This manual provides programming information for the MVME1603 and MVME1604 Single Board Computers, that are based on the MVME1600-001 or MVME1600-011 base modules and the PM603, PM604, and RAM104 mezzanine modules.. Extensive programming information is provided for several Application-Specific Integrated Circuit (ASIC) devices used on the boards. Reference information is included in Appendix A for the Large Scale Integration (LSI) devices used on the boards and sources for additional information are listed.

This chapter briefly describes the board level hardware features of the MVME1603/MVME1604 Single Board Computers. The chapter begins with a board level overview and features list. Memory maps are next, and are the major feature of this chapter.

Programmable registers in the MVME1603/MVME1604 that reside in ASICs are covered in the chapters on those ASICs. Chapter 2 covers the VMEchip2, Chapter 3 covers the VME2PCI, and Chapter 4 covers certain programming features, such as DMA and interrupts. Appendix A lists all related documentation.

Manual Terminology

Throughout this manual, a convention is used which precedes data and address parameters by a character identifying the numeric format as follows:

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

For example, “12” is the decimal number twelve, and “$12” is the decimal number eighteen.

Unless otherwise specified, all address references are in hexadecimal.

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

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

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

Data and address sizes are defined as follows:

- A *byte* is eight bits, numbered 0 through 7, with bit 0 being the least significant.
- A *half-word* is 16 bits, numbered 0 through 15, with bit 0 being the least significant.
- A *word* or *single word* is 32 bits, numbered 0 through 31, with bit 0 being the least significant.
- A *double word* is 64 bits, numbered 0 through 63, with bit 0 being the least significant.

Refer to Chapter 4 for *Endian Issues*, which covers which parts of the MVME1603/MVME1604 use *big-endian* byte ordering, and which use *small-endian* byte ordering.

The terms *control bit* and *status bit* are used extensively in this document. The term control bit is used to describe a bit in a register that can be set and cleared under software control. The term *true* is used to indicate that a bit is in the state that enables the function it controls. The term *false* is used to indicate that the bit is in the state that disables the function it controls. In all tables, the terms 0 and 1 are used to describe the actual value that should be written to the bit, or the value that it yields when read. The term *status bit* is used

to describe a bit in a register that reflects a specific condition. The status bit can be read by software to determine operational or exception conditions.

Overview

The MVME1603/MVME1604 SBC provides many standard features required by a computer system: SCSI, Ethernet interface, Graphics (MVME1600-001 only), keyboard interface, mouse interface, sync and async serial ports, parallel port, boot ROM, and up to 128MB of DRAM in one or two VME slot(s). The flexible mezzanine architecture of the MVME1603/MVME1604 SBC allows relatively easy upgrades for the processor and/or memory.

Features

- MPC603 PowerPC™ processor with up to 128MB of DRAM in a Single-Slot VME
  or MPC604 PowerPC™ processor with up to 128MB of DRAM (requires 2 VME Slots)
- 8 to 128 MB of DRAM options
- 1MB of FLASH
- RTC with 8KB of NVRAM implemented with SGS-Thomson M48T18 device
- Two synchronous serial ports implemented with the Zilog 85230 ESCC
- Four 16-bit timers (one in the S82378ZB device and three in the Z8536 device)
- PC87303 Super IO Device to provide:
  - parallel port (meets IEEE1284 Bidirectional requirements when using the MVME1600-00x base board)
  - Two asynchronous serial ports
  - keyboard and mouse (when using the MVME1600-00x base module)

- VGA-compatible and high-resolution color graphics implemented with Cirrus CL-GD5434 Graphics Accelerator (only on systems using the MVME1600-001; NOT on those using the MVME1600-011).
- High-performance 16-bit SCSI Bus interface(using the MVME1600-001 with the NCR 53C825 device) or 8-bit SCSI Bus interface (using the MVME1600-011 with NCR 53C810 device)
- Ethernet transceiver interface for both AUI and 10base-T, implemented with AMD Am79C970 device
- VMEbus interface implemented with VMEchip2 ASIC
  - VMEbus system controller functions
  - VMEbus interface to local bus (A24/A32, D8/D16/D32/BLT(D8/D16/D32/D64))
  - Local bus to VMEbus interface (A16/A24/A32, D8/D16/D32)
  - VMEbus interrupter
  - VMEbus interrupt handler
  - Global CSR for interprocessor communications
  - DMA for fast local memory - VMEbus transfers (A16/A24/A32, D16/D32/D64).
- Interface to one P1386.1 PCI Mezzanine Card (PMC)
- 8-bit Software Readable Header
- RST (reset) Switch
- ABT (abort) Switch
- Status LEDs

Block Diagram

Figure 1-1 is a general block diagram of the MVME1600-001-based version of the MVME1603/MVME1604.

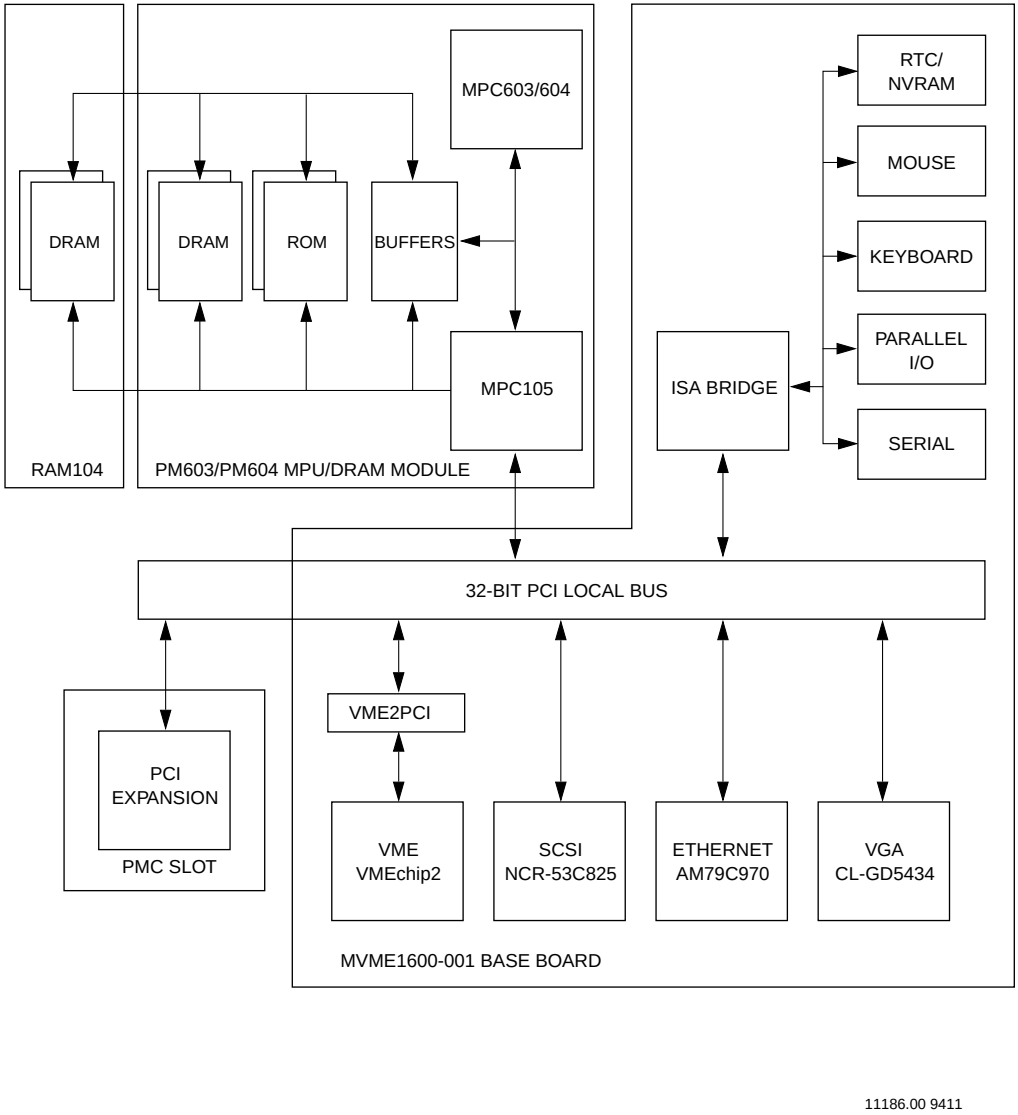


Figure 1-1. MVME1603/MVME1604 Block Diagram

Functional Description

A complete functional description of the major blocks on the MVME1603/MVME1604 Single Board Computers is provided in the *MVME1603/MVME16904 Single Board Computers Installation and Use* manual

Programming Model

The following sections describe the programming model for the MVME1603/MVME1604

Memory Maps

There are multiple buses on the MVME1603/MVME1604 and each bus domain has its own view of the memory map. Some of these are shown in the following sections. The remaining ones are given in the documents listed in Appendix A, *Related Documentation*.

The MVME1603/MVME1604 will make every attempt to conform to the PowerPC Reference Platform (PRP)Specification.

Processor View of the Memory Map

The following Table shows the Memory Map of the MVME1603/MVME1604 from the point of view of the Processor.

Table 1-1. Processor View of the Memory Map

Processor Address		Size	PCI Address Generated		Definition	Notes
Start	End		Start	End		
00000000	7FFFFFFF	2G			DRAM - Not Forwarded to PCI	
80000000	807FFFFF	8M	00000000	007FFFFF	ISA/PCI I/O Space	1, 2, 6
80800000	80FFFFFF	8M	00800000	00FFFFFF	Direct Map PCI Configuration Space	3
81000000	BF7FFFFF	1000M	01000000	3F7FFFFF	PCI I/O Space	
BF8FFFFF	BFFFFFEF	8M -16M			Reserved	
BFFFFFF0	BFFFFFFF	16	3FFFFFF0	3FFFFFFF	PCI IACK/Special Cycles	
C0000000	C0FFFFFF	16M	00000000	00FFFFFF	PCI/ISA Memory Space	
C0000000	FEFFFFFF	1G -32M	01000000	3EFFFFFF	PCI Memory Space	
FF000000	FF07FFFF	512KB			EPROM/FLASH Bank 0	4
FF080000	FF0FFFFF	512KB			EPROM/FLASH Bank 1	4
FF100000	FFEFFFFF	14MB			Reserved	4, 5
FFF00000	FFF7FFFF	512KB			EPROM/FLASH Bank 0 Repeat	4
FFF80000	FFFFFFFF	512KB			EPROM/FLASH Bank 1 Repeat	4

Notes:

1. PCI configuration accesses to CF8 (Configuration Address) and CFC (Configuration Data) are supported by the MPC105 as specified in the PCI Specification Revision 2.0.
2. Both Contiguous and Discontiguous mappings are supported by the MVME1603/MVME1604 See *ISA I/O Space Mapping* section for more details.
3. This space is used for Direct Mapped PCI Configuration Space accesses. See *PCI Configuration Space Mapping* section for more details.
4. EPROM/FLASH decoding repeats every 1MB for this entire 16MB range.
5. The usage of this address range for EPROM/FLASH is not recommended since future PowerPC products will redefine this 14MB area.
6. The M48T18 RTC & NVRAM device is mapped in this area. See *ISA I/O Space Mapping* section for more details.

Direct Mapped PCI Configuration Space

The following Table shows the mapping of the Direct Mapped PCI Configuration Space on the MVME1603/MVME1604:

Table 1-2. PCI Configuration Space Map

IDSEL	Processor Address		PCI Configuration Space Address		Definition
	Start	End	Start	End	
	80800000	808007FF	00800000	008007FF	*Reserved*
A11	80800800	808008FF	00800800	008008FF	IBC Configuration Registers
	80800900	80800FFF	00800900	00800FFF	*Reserved*
A12	80801000	808010FF	00801000	008010FF	53C810/825 Configuration Registers
	80801100	80801FFF	00801100	00801FFF	*Reserved*
A13	80802000	808020FF	00802000	008020FF	VME2PCI Configuration Registers
	80802100	80803FFF	00802100	00803FFF	*Reserved*
A14	80804000	808040FF	00804000	008040FF	Am79C970 Configuration Registers
	80804100	80807FFF	00804100	00807FFF	*Reserved*
A15	80808000	808080FF	00808000	008080FF	GD5434 Configuration Registers
	80808100	8080FFFF	00808100	0080FFFF	*Reserved*
A16	80810000	808100FF	00810000	008100FF	PMC Slot Configuration Registers
	80810100	80FFFFF	00810100	00FFFFFF	*Reserved*

Notes:

1. Accessing *Reserved* space may cause unpredictable results because multiple devices may be selected.

ISA I/O Space

The following Table shows the mapping of the PCI Configuration Space on the MVME1603/MVME1604:

Table 1-3. ISA I/O Space Map

ISA I/O Address	Processor Address		Function	Notes
	Contiguous	Discontiguous		
0000 - 000F	8000 0000 - 8000 000F	8000 0000 - 8000 000F	IBC: DMA1 Registers & Control	2
0020 - 0021	8000 0020 - 8000 0021	8000 1000 - 8000 1001	IBC: Interrupt 1 Control & Mask	2
0040 - 0043	8000 0040 - 8000 0043	8000 2000 - 8000 2003	IBC: Timer Counter 1 Registers	2
0060	8000 0060	8000 3000	IBC: Reset Ubus IRQ12	2
0061	8000 0061	8000 3001	IBC: NMI Status and Control	2
0064	8000 0064	8000 3004	ISASIO: Keyboard Controller Port	3
0074	8000 0074	8000 3014	NVRAM/RTC Address Strobe 0	
0075	8000 0075	8000 3015	NVRAM/RTC Address Strobe 1	
0077	8000 0077	8000 3017	NVRAM/RTC Data Port	
0080 - 0090	8000 0080 - 8000 0090	8000 4000 - 8000 4010	IBC: DMA Page Registers	2
0092	8000 0092	8000 4012	IBC: Port 92 Register	2
0094 - 009F	8000 0094 - 8000 009F	8000 4012 - 8000 401F	IBC: DMA Page Registers	2
00A0 - 00A1	8000 00A0 - 8000 00A1	8000 5000 - 8000 5001	IBC: Interrupt 2 Control & Mask	2
00C0 - 00CF	8000 00C0 - 8000 00CF	8000 6000 - 8000 600F	IBC: DMA2 Address Registers	2
00D0 - 00DF	8000 00D0 - 8000 00DF	8000 7000 - 8000 700F	IBC: DMA2 Control Registers	2
02F8 - 02FF	8000 02F8 - 8000 02FF	8001 7018 - 8001 701F	ISASIO: Serial Port 2 (COM2)	3
0398	8000 0398	8001 C018	ISASIO Index Register	2
0399	8000 0399	8001 C019	ISASIO Data Register	2
03BC - 03BF	8000 03BC - 8000 03BF	8001 D01C - 8001 D01F	ISASIO: Parallel Port (LPT1)	3
03F0 - 03F7	8000 03F0 - 8000 03F7	8001 F010 - 8001 F017	Reserved for Floppy Drive Controller (FDC)	3

Table 1-3. ISA I/O Space Map (Continued)

ISA I/O Address	Processor Address		Function	Notes
	Contiguous	Discontiguous		
03F8 - 03FF	8000 03F8 - 8000 03FF	8001 F018 - 8001 F01F	ISASIO: Serial Port 1 (COM1)	3
040A	8000 040A	8002 000A	IBC: Scatter/Gather Interrupt Status Register	2
040B	8000 040B	8002 000B	IBC: DMA1 Extended Mode Register	2
0410 - 041F	8000 0410 - 8000 041F	8002 0010 - 8002 001F	IBC: DMA Scatter/Gather Command and Status Registers	2
0420 - 042F	8000 0420 - 8000 042F	8002 1000 - 8002 100F	IBC: DMA CH0-CH3 Scatter/Gather Descriptor Table Pointers	2
0430 - 043F	8000 0430 - 8000 043F	8002 1010 - 8002 101F	IBC: DMA CH4-CH7 Scatter/Gather Descriptor Table Pointers	2
0481 - 048B	8000 0481 - 8000 048B	8002 4000 - 8002 400B	IBC: DMA High Page Registers	2
04D0	8000 04D0	8002 6010	IBC: INT1 Edge Level Control	2
04D1	8000 04D1	8002 6011	IBC: INT2 Edge Level Control	2
04D6	8000 04D6	8002 6016	IBC: DMA2 Extended Mode Register	2
0C04	8000 0C04	8006 0004	IBC: Power Control Output Port	2, 4
0C01	8000 0C01	8006 0001	IBC: Test Mode Control Port/Shadow Register of Port 70	2, 4
0800	8000 0800	8004 0000	CPU Configuration Register	4, 6
0801	8000 0801	8004 0001	Software Readable Header	4
0802	8000 0802	8004 0002	Board Configuration Register	4
0803	8000 0803	8004 0003	Reserved	4
0804	8000 0804	8004 0004	DRAM Size Register	4, 6
0805	8000 0805	8004 0005	Reserved	4
0806	8000 0807	8004 0006	Reserved	4
0807	8000 0807	8004 0007	Reserved	4
0820	8000 0820	8004 1000	Reserved for Cooling Monitor	4
0830	8000 0830	8004 1010	Reserved for Audio	4
0840	8000 0840	8004 2000	Z85230: Port B (Serial Port 4) Control	4
0841	8000 0841	8004 2001	Z85230: Port B (Serial Port 4) Data	4
0842	8000 0842	8004 2002	Z85230: Port A (Serial Port 3) Control	4
0843	8000 0F43	8004 2003	Z85230: Port A (Serial Port 3) Data	4

Table 1-3. ISA I/O Space Map (Continued)

ISA I/O Address	Processor Address		Function	Notes
	Contiguous	Discontiguous		
0844	8000 0844	8004 2004	Z8536 CIO: Port C's Data Register	4
0845	8000 0845	8004 2005	Z8536 CIO: Port B's Data Register	4
0846	8000 0846	8004 2006	Z8536 CIO: Port A's Data Register	4
0847	8000 0847	8004 2007	Z8536 CIO: Control Register	4
084F	8000 084F	8004 200F	Z85230/Z8536 Pseudo IACK	4, 5

Notes:

1. All ISA I/O locations not specified in this table are reserved.
2. These locations are internally decoded by the IBC
3. These locations are internally decoded by the ISASIO.
4. These locations are either not specified by the PRP Specification or not PRP-compliant and may overlap some other functions specified by the PRP specification.
5. IACK vector is returned from either the Z8536 or the Z85230 when this location is read.
6. These registers physically reside on the PM603/PM604 module.

PCI View of the Memory Map

The following Table shows the PCI Memory Map of the MVME1603/MVME1604 from the point of view of the PCI Local Bus.

Table 1-4. PCI View of the PCI Memory Map

PCI Address		Size	Processor Bus Address		Definition	Notes
Start	End		Start	End		
00000000	00FFFFFF	16M	Not forwarded to MPU bus		PCI/ISA Memory Space	1, 2
01000000	7FFFFFFF	1G - 16M	Not forwarded to MPU bus		PCI Memory Space	2
80000000	FFFFFFFF	2G	00000000	7FFFFFFF	Onboard DRAM (via MPC105)	
00000000	FFFFFFFF	4G	Not forwarded to MPU bus		PCI/ISA I/O Space	

Notes:

1. IBC performs subtractive decoding for in this range and forward the PCI memory cycle to ISA if DEVSEL_ is not detected.
2. VME2PCI can be programmed to claim some of this address range to forward the PCI memory cycle to the VMEchip2.

Control and Status Registers (CSR)

The CSR on MVME1603/MVME1604 consists of: the CPU Configuration Register, the Software Readeable Header, the Board Configuration Register, and the DRAM Size Register. These registers are accessible in ISA I/O space.

Note that in the OPER row, R = read only bit, R/W = read or write bit, and W = write only bit.

CPU Configuration Register

The CPU Configuration Register provides the configuration information about the PM603/PM604 module.This register resides on the PM603/PM604 mezzanine module, but actual decoding is done by the MVME160X board.

REG	CPU Configuration Register - 0800 (hex)							
BIT	SD7	SD6	SD5	SD4	SD3	DS2	SD1	SD0
FIELD	CPUTYPE				CKM1	CKM0	L2P1	L2P0
OPER	R				R	R	R	R
RESET	0001 (binary)				N/A	N/A	N/A	N/A

L2P1-L2P0 L2 Cache Present. These bits are defined as follows:

L2P1	L2P0	L2 Cache Size
0	0	512KB
0	1	256KB
1	0	1MB
1	1	L2 Cache Not Present

CKM1-CKM0 Clocking Configuration. These bits reflect the clocking configuration of the PM603/PM604. The encoding for these bits is as follows:

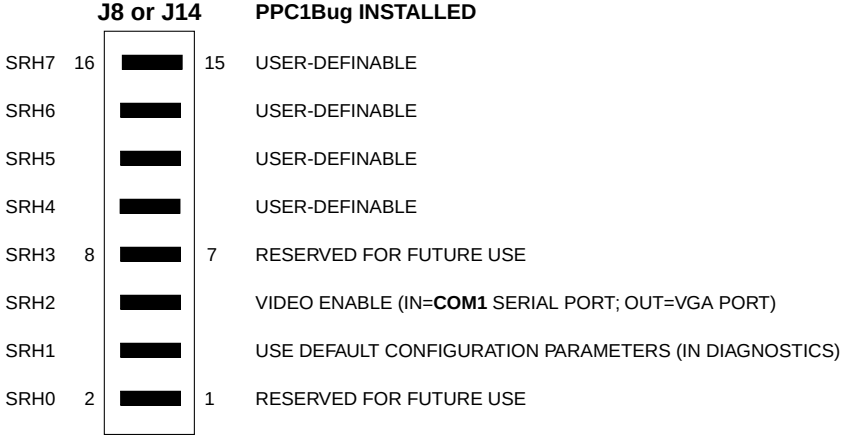
CKM1	CKM0	PCI Bus Clock	CPU External Bus Clock
0	0	33MHz	33MHz
0	1	20MHz	40MHz
1	0	25MHz	50MHz
1	1	33MHz	66MHz

CPUTYPE CPU Type. These four bits reflect the CPU type information. For the PM603/PM604, this field is hardwired to 0001 (binary).

Software Readable Header J8 or J14

A 2x8 header is provided as the Software Readable Header (SRH). A logic 0 means a jumper is installed for that particular bit and a logic 1 means the jumper is not installed. SRH Bit 0 is associated with Pin 1 and Pin 2 of the SRH, and SRH Bit 7 is associated with Pin 15 and Pin 16 of the SRH. The SRH is a read-only register located at ISA I/O address x801 (hex).This register is J8 on the MVME1600-001 main module, but it is J14 on the MVME1600-011 main module. The default (as shipped) configuration is with the first four jumpers, SRH0 through SRH3, installed. They are used by the debug monitor, PPCBug .

REG	Software Readable Header Register - 0801 (hex)							
BIT	SD7	SD6	SD5	SD4	SD3	DS2	SD1	SD0
FIELD	SRH7	SRH6	SRH5	SRH4	SRH3	SRH2	SRH1	SRH0
OPER	R	R	R	R	R	R	R	R
RESET	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A



Board Configuration Register

The Board Configuration Register is an 8-bit register providing the configuration information about the MVME1603/MVME1604 Single Board Computer. This read-only register is located at located at ISA I/O address x0F02. This register is on the MVME1600-001 main module

REG	Board Configuration Register - 0802 (hex)							
BIT	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
FIELD	GIOP_	SCCP_		PMCP_	VMEP_	GFXP_	LANP_	SCSIP_
OPER	R	R	R	R	R	R	R	R
RESET	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A

GIOP_ MVME760 module Present. If set, MVME760 transition module is not connected. If cleared, MVME760 module is connected. (This applies only to MVME1600-00x boards, NOT MVME1600-0xx.)

SCCP_ Z85230 ESCC Present. If set, there is no on-board sync serial support (ESCC not present). If cleared, there is on-board support for sync serial interface via Z85230 ESCC.

PMCP_ PMC Present. If set, there is no PCI Mezzanine Card installed in the PMC Slot. If cleared, the PMC slot contains a PMC.

VMEP_ VMEbus Present. If set, there is no VMEbus interface. If cleared, VMEbus interface is supported.

GFXP_ Graphics Present. If set, there is no onboard graphics interface. If cleared, there is an onboard graphics capability (MVME1600-0xx has no graphics).

LANP_ Ethernet Present. If set, there is no Ethernet transceiver interface. If cleared, there is on-board Ethernet support.

SCSIP_ SCSI Present. If set, there is no on-board SCSI interface. If cleared, on-board SCSI is supported.

DRAM Size Register

The DRAM Size Register is an 8-bit register providing the DRAM size information. Banks 0 and 1 are on the PM603/PM604; Banks 2 and 3 reside on the RAM104 DRAM Add-on module. This register resides on the PM603/PM604 module but the actual address decoding is done by the MVME160X board.

REG	DRAM Size Register - 0804h							
BIT	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
FIELD	B2/B3 ASYM_	B2/B3 SIZ2	B2/B3 SIZ1	B2/B3 SIZ0	B0/B1 ASYM_	B0/B1 SIZ2	B0/B1 SIZ1	B0/B1 SIZ0
OPER	R	R	R	R	R	R	R	R
RESET	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

SIZ2-SIZ0 DRAM Size. These bits provide the DRAM size information for the four banks of DRAM supported by the PM603/PM604. The encoding for these size bits is as follows:

B0/B1 (B2/B3)			DRAM Size	
SIZ2	SIZ1	SIZ0	Bank 0 (Bank 2)	Bank 1 (Bank 3)
0	1	1	Not Present	Not Present
0	1	0	8 MB	Not Present
0	0	1	32 MB	Not Present
0	0	0	128MB	Not Present
1	1	1	Not Present	Not Present
1	1	0	8 MB	8 MB
1	0	1	32 MB	32 MB
1	0	0	128 MB	128 MB

ASYM_ Asymmetric Refresh Mode. When cleared, this bit indicates that the DRAM devices installed for Bank 0 and Bank 1 (Bank 2 and Bank 3) have more row address bits than column address bits. This bit is used to determine how to program the MPC105 chip appropriately. Note that, at this time, only the 4M x 4 DRAM devices (32MB banks) have this option. For 4M x4 DRAM, the asymmetric refresh mode is also referred to as the 4K refresh mode. For these devices, there would be 12 row addresses and 10 column addresses.

Z8536 CIO Port Pins

The Z8536 CIO is used to provide: Software Reset function, ABORT interrupt status, FUSE status, SCSI Terminators mode status & control, modem control lines not provided by the Z85230 ESCC, and a method to inquire the module ID of the two sync/async serial ports that reside on the MVME760 module (NOT the MVME712). The assignment for the Port pins of the Z8536 CIO is as follows:

Table 1-5. Z8536 CIO Port Pins Assignment

Port Pin	Signal Name	Direction	Signal Name
PA0	TM3_ MID0	Input	Port 3 Test Mode when IDREQ_ = 1; Module ID Bit 0 when IDREQ_ = 0.
PA1	DSR3_ MID1	Input	Port 3 Data Set Ready when IDREQ_ = 1; Module ID Bit 1 when IDREQ_ = 0.
PA2	RI3_	Input	Port 3 Ring Indicator
PA3	LLB3_ MODSEL	Output	Port 3 Local Loopback (IDREQ_ = 1) or Port Select (IDREQ_ = 0): IDREQ_ = 0 & MODSEL = 0 => Port 3 ID Select IDREQ_ = 0 & MODSEL = 1 => Port 4 ID Select
PA4	RLB3_	Output	Port 3 Remote Loopback
PA5	DTR3_	Output	Port 3 Data Terminal Ready
PA6	BRDFAIL	Output	Board Fail: When set will cause FAIL LED to be lit.
PA7	IDREQ_	Output	Module ID Request - low true
PB0	TM4_ MID2	Input	Port 4 Test Mode when IDREQ_ = 1; Module ID Bit 2 when IDREQ_ = 0.
PB1	DSR4_ MID3	Input	Port 4 Data Set Ready when IDREQ_ = 1; Module ID Bit 3 when IDREQ_ = 0.
PB2	RI4_	Input	Port 4 Ring Indicator
PB3	LLB4_	Output	Port 4 Local Loopback
PB4	RLB4_	Output	Port 4 Remote Loopback
PB5	DTR4_	Output	Port 4 Data Terminal Ready
PB6	FUSE	Input	FUSE = 1 means at least one of the polyswitches on the MVME1600-001 board is opened.
PB7	ABORT_	Input	Status of the ABORT_ signal.
PC0	SCSITERM_	Input	Input: If cleared, on-board SCSI terminators are enabled. If set, SCSI terminators are disabled.
PC1	TBENDIS_	Output	TBEN Disable. If cleared, TBENDIS_ will drive processor's TBEN pin low to disable its internal timebase.
PC2	BASETYP0	Input	Genesis Base Board Type: 00 (binary) = Reserved; 01 (binary) = MVME1600-002; 10 (binary) = Reserved; 11 (binary) = MVME1600-001;
PC3	BASETYP1	Input	

NOTE: The direction and the polarity of the Z8536's port pins are software programmable.

Serial Port Module ID

The module ID signals, which are only valid when IDREQ_ is asserted, indicate the type of the serial module that is installed on either Port 3 or Port 4. The following table shows how to interpret the MID3-MID0 signals:

Table 1-6. Interpretation of MID3-MID0

IDREQ_	LLB3/ MODSEL	MID3	MID2	MID1	MID0	Serial Module Type	Module Assembly Number
1	X	X	X	X	X	Invalid module ID	
0	0	0	0	0	0	Module 3: EIA232 DCE	01-W3876B01
0	0	0	0	0	1	Module 3: EIA232 DTE	01-W3877B01
0	0	0	0	1	0	Module 3: EIA530 DCE	01-W3878B01
0	0	0	0	1	1	Module 3: EIA530 DTE	01-W3879B01
0	0	1	1	1	1	Module 3 Not Installed	
0	1	0	0	0	0	Module 4: EIA232 DCE	01-W3876B01
0	1	0	0	0	1	Module 4: EIA232 DTE	01-W3877B01
0	1	0	0	1	0	Module 3: EIA530 DCE	01-W3878B01
0	1	0	0	1	1	Module 3: EIA530 DTE	01-W3879B01
0	1	1	1	1	1	Module 4 Not Installed	

Note Because IDREQ_ and MID3-MID0 signals go through the P2MX Function, software must wait for the MID3-MID0 to become valid after asserting IDREQ_. The waiting time should be about 4 microseconds because the sampling rate is about 1.6 microsecond with a 10MHz MXCLK.

VMEchip2 Memory Maps

The general memory maps for the VMEchip2 are shown next. Refer to Chapter 2 for details.

Table 1-7. VMEchip2 Memory Map (Sheet 1 of 3)

chip2 LCSR Base Address = $FFF40000
SET:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLAVE ENDING ADDRESS 1															
SLAVE ENDING ADDRESS 2															
SLAVE ADDRESS TRANSLATION ADDRESS 1															
SLAVE ADDRESS TRANSLATION ADDRESS 2															
				ADDER 2	SNP 2			WP 2	SUP 2	USR 2	A32 2	A24 2	BLK D64 2	BLK 2	PRGM 2	DATA2
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MASTER ENDING ADDRESS 1															
MASTER ENDING ADDRESS 2															
MASTER ENDING ADDRESS 3															
MASTER ENDING ADDRESS 4															
MASTER ADDRESS TRANSLATION ADDRESS 4															
MAST D16 EN	MAST WP EN	MASTER AM 4						MAST D16 EN	MAST WP EN	MASTER AM 3					
GCSR GROUP SELECT								GCSR BOARD SELECT				MAST 4 EN	MAST 3 EN	MAST 2 EN	MAST 1 EN
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										WAIT RMW	ROM ZERO	DMA TB SNP MODE		SRAM SPEED	
DMA CONTROLLER															
DMA CONTROLLER															
DMA CONTROLLER															
DMA CONTROLLER															
	TICK 2/1	TICK IRQ 1 EN	CLR IRQ	IRQ STAT	VMEBUS INTERRUPT LEVEL			VMEBUS INTERRUPT VECTOR							

This sheet continues on facing page. ⟶

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLAVE STARTING ADDRESS 1															
SLAVE STARTING ADDRESS 2															
SLAVE ADDRESS TRANSLATION SELECT 1															
SLAVE ADDRESS TRANSLATION SELECT 2															
				ADDER 1	SNP 1			WP 1	SUP 1	USR 1	A32 1	A24 1	BLK D64 1	BLK 1	PRGM 1	DATA1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASTER STARTING ADDRESS 1															
MASTER STARTING ADDRESS 2															
MASTER STARTING ADDRESS 3															
MASTER STARTING ADDRESS 4															
MASTER ADDRESS TRANSLATION SELECT 4															
MAST D16 EN	MAST WP EN	MASTER AM 2						MAST D16 EN	MAST WP EN	MASTER AM 1					
IO2 EN	IO2 WP EN	IO2 S/U	IO2 P/D	IO1 EN	IO1 D16 EN	IO1 WP EN	IO1 S/U	ROM SIZE		ROM BANK B SPEED			ROM BANK A SPEED		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARB ROBN	MAST DHB	MAST DWB		MST FAIR	MST RWD	MASTER VMEBUS		DMA HALT	DMA EN	DMA TBL	DMA FAIR	DM RELM		DMA VMEBUS	
DMA TBL INT	DMA LB SNP MODE			DMA INC VME	DMA INC LB	DMA WRT	DMA D16	DMA D64 BLK	DMA BLK	DMA AM 5	DMA AM 4	DMA AM 3	DMA AM 2	DMA AM 1	DMA AM 0
LOCAL BUS ADDRESS COUNTER															
VMEBUS ADDRESS COUNTER															
BYTE COUNTER															
TABLE ADDRESS COUNTER															
DMA TABLE INTERRUPT COUNT				MPU CLR STAT	MPU LBE ERR	MPU LPE ERR	MPU LOB ERR	MPU LTO ERR	DMA LBE ERR	DMA LPE ERR	DMA LOB ERR	DMA LTO ERR	DMA TBL ERR	DMA VME ERR	DMA DONE

1360 9403

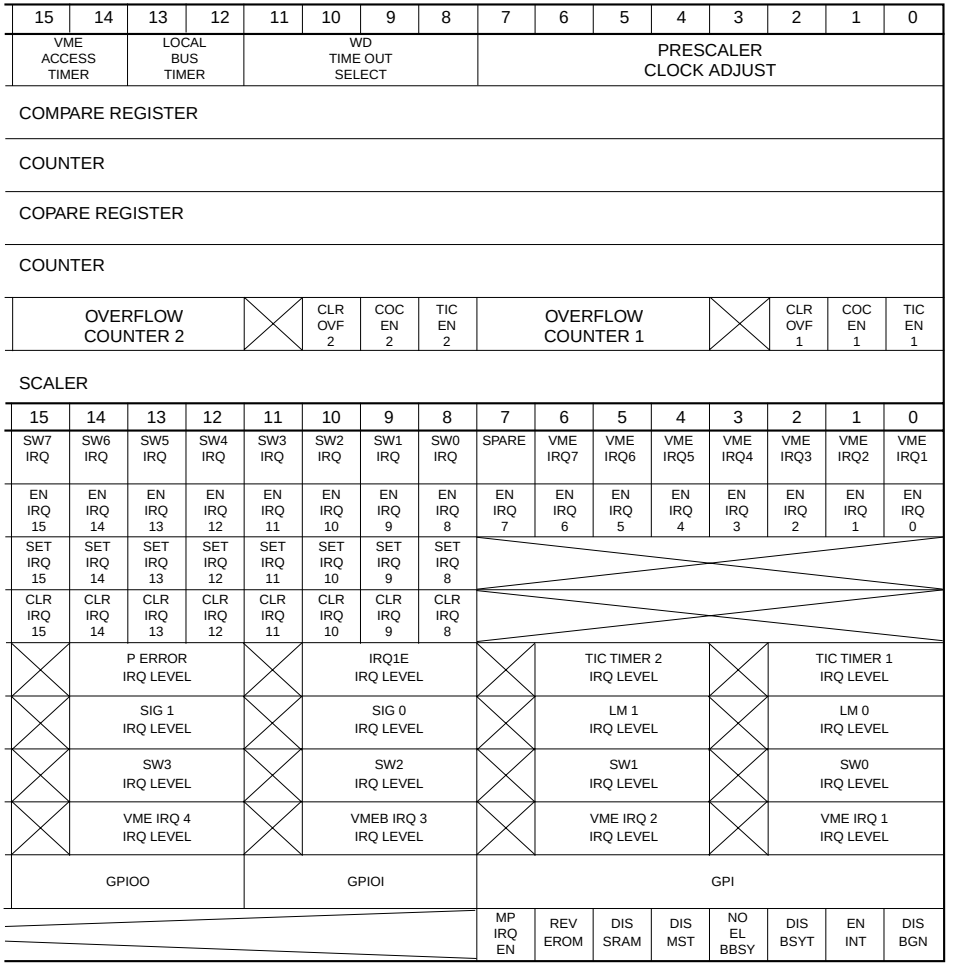
← This sheet begins on facing page.

Table 1-7. VMEchip2 Memory Map (Sheet 2 of 3)

VMEchip2 LCSR Base Address = $FFF40000
OFFSET:

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
4C								ARB BGTO EN	DMA TIME OFF			DMA TIME ON			VME GLOBAL TIMER	
50	TICK TIMER 1															
54	TICK TIMER 1															
58	TICK TIMER 2															
5C	TICK TIMER 2															
60		SCON	SYS FAIL	BRD FAIL STAT	PURS STAT	CLR PURS STAT	BRD FAIL OUT	RST SW EN	SYS RST	WD CLR TO	WD CLR CNT	WD TO STAT	TO BF EN	WD SRST LRST	WD RST EN	WD EN
64	PRE															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
68	AC FAIL IRQ	AB IRQ	SYS FAIL IRQ	MWP BERR IRQ	PE IRQ	IRQ1E IRQ	TIC2 IRQ	TIC1 IRQ	VME IACK IRQ	DMA IRQ	SIG3 IRQ	SIG2 IRQ	SIG1 IRQ	SIG0 IRQ	LM1 IRQ	LM0 IRQ
6C	EN IRQ 31	EN IRQ 30	EN IRQ 29	EN IRQ 28	EN IRQ 27	EN IRQ 26	EN IRQ 25	EN IRQ 24	EN IRQ 23	EN IRQ 22	EN IRQ 21	EN IRQ 20	EN IRQ 19	EN IRQ 18	EN IRQ 17	EN IRQ 16
70																
74	CLR IRQ 31	CLR IRQ 30	CLR IRQ 29	CLR IRQ 28	CLR IRQ 27	CLR IRQ 26	CLR IRQ 25	CLR IRQ 24	CLR IRQ 23	CLR IRQ 22	CLR IRQ 21	CLR IRQ 20	CLR IRQ 19	CLR IRQ 18	CLR IRQ 17	CLR IRQ 16
78		AC FAIL IRQ LEVEL				ABORT IRQ LEVEL				SYS FAIL IRQ LEVEL				MST WP ERROR IRQ LEVEL		
7C		VME IACK IRQ LEVEL				DMA IRQ LEVEL				SIG 3 IRQ LEVEL				SIG 2 IRQ LEVEL		
80		SW7 IRQ LEVEL				SW6 IRQ LEVEL				SW5 IRQ LEVEL				SW4 IRQ LEVEL		
84		SPARE IRQ LEVEL				VME IRQ 7 IRQ LEVEL				VME IRQ 6 IRQ LEVEL				VME IRQ 5 IRQ LEVEL		
88	VECTOR BASE REGISTER 0				VECTOR BASE REGISTER 1				MST IRQ EN	SYS FAIL LEVEL	AC FAIL LEVEL	ABORT LEVEL	GPIOEN			
8C																

This sheet continues on facing page. ⟶



1361 9403

← This sheet begins on facing page.

Table 1-7. VMEchip2 Memory Map (Sheet 3 of 3)

VMEchip2 GCSR Base Address = $FFF40100

Offsets																		
VME -bus	**Local Bus**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	0	CHIP REVISION								CHIP ID							
2	4	LM3	LM2	LM1	LM0	SIG3	SIG2	SIG1	SIG0	RST	ISF	BF	SCON	SYSFL	X	X	X
4	8	GENERAL PURPOSE CONTROL AND STATUS REGISTER 0															
6	C	GENERAL PURPOSE CONTROL AND STATUS REGISTER 1															
8	10	GENERAL PURPOSE CONTROL AND STATUS REGISTER 2															
A	14	GENERAL PURPOSE CONTROL AND STATUS REGISTER 3															
C	18	GENERAL PURPOSE CONTROL AND STATUS REGISTER 4															
E	1C	GENERAL PURPOSE CONTROL AND STATUS REGISTER 5															

Table 1-8. M48T18 BBRAM/TOD Clock Memory Map

Offset (start/end)	Description	Size (Bytes)
$00000000 - $00000FFF	NVRAM per the PRP specification	4096
$00001000 - $000010FF	open	1784
$000016F8 - $00001EF7	Debugger Area	2048
$00001EF8 - $00001FF7	Configuration Area (see Table 1-9)	256
$00001FF8 - $00001FFF	TOD Clock (see Table 1-10)	8

Table 1-9. BBRAM Configuration Area Memory Map

Address Range	Description	Size (Bytes)
$00001EF8 - $00001EFB	Version	4
$00001EFC - $00001F07	Serial Number	12
$00001F08 - $00001F17	Board ID	16
$00001F18 - $00001F27	PWA	16
$00001F28 - $00001F2B	Reserved_0	4
$00001F2C - $00001F31	Ethernet Address	6
$00001F32 - $00001F33	Reserved_1	2
$00001F34 - $00001F35	Local SCSI ID	2
$00001F36 - $00001F38	MPU Speed in MHz	3
$00001F39 - $00001F3B	Bus Speed in MHz	3
$00001F3C - $00001FF6	Reserved	187
$00001FF7	Checksum	1

Table 1-10. TOD Clock Memory Map

Address	Data Bits								Function	
	D7	D6	D5	D4	D3	D2	D1	D0		
$00001FF8	W	R	S	--	--	--	--	--	CONTROL	
$00001FF9	ST	--	--	--	--	--	--	--	SECONDS	00
$00001FFA	x	--	--	--	--	--	--	--	MINUTES	00
$00001FFB	x	x	--	--	--	--	--	--	HOUR	00
$00001FFC	x	FT	x	x	x	--	--	--	DAY	01
$00001FFD	x	x	--	--	--	--	--	--	DATE	01
$00001FFE	x	x	x	--	--	--	--	--	MONTH	01
$00001FFF	--	--	--	--	--	--	--	--	YEAR	00

NOTES: W = Write Bit R = Read Bit S = Sign Bit

ST = Stop Bit FT = Frequency Test x = Unused

BBRAM,TOD Clock Memory Map

The M48T18 BBRAM (also called Non-Volatile RAM or NVRAM) is divided into five areas as shown in Table 1-8. The first four areas are defined by software, while the fifth area, the time-of-day (TOD) clock, is defined by the chip hardware. The first area is reserved for user data. The second area is open. The third area is used by the MVME160x board debugger (PPCBug). The fourth area, detailed in Table 1-9, is the configuration area. The fifth area, the TOD clock, detailed in Table 1-10, is defined by the chip hardware.

The data structure of the configuration bytes starts at offset $00001EF8 and is as follows.

```
struct brdi_cnfg {
        char    version[4];
        char    serial[12];
        char    id[16];
        char    pwa[16];
        char    reserved_0[4];
        char    ethernet_adr[6];
        char    reserved_1[2];
        char    lscsiid[2];
        char    speed_mpu[3];
        char    speed_bus[3];
        char    reserved[187];
        char    cksum[1];
}
```

The fields are defined as follows:

1. Four bytes are reserved for the revision or version of this structure. This revision is stored in ASCII format, with the first two bytes being the major version numbers and the last two bytes being the minor version numbers. For example, if the version of this structure is 1.0, this field contains:

   ```
   0100
   ```

2. Twelve bytes are reserved for the serial number of the board in ASCII format. For example, this field could contain:

```
000000470476
```

3. Sixteen bytes are reserved for the board ID in ASCII format. For example, for a 16 MB,66 MHz MVME1603 board, this field contains:

   ```
   MVME1603-002A
   ```

   (The 13 characters are followed by three blanks.)

4. Sixteen bytes are reserved for the printed wiring assembly (PWA) number assigned to this board in ASCII format. This includes the `01-W` prefix. This is for the main logic board if more than one board is required for a set. Additional boards in a set are defined by a structure for that set. For example, for a 16 MB, 66 MHz MVME1603 board at revision A, the PWA field contains:

   ```
   01-W3015B02A
   ```

   (The 12 characters are followed by four blanks.)

5. These four bytes are reserved.

6. Six bytes are reserved for the Ethernet address. The address is stored in hexadecimal format. (Refer to the detailed description earlier in this chapter.) If the board does not support Ethernet, this field is filled with zeros.

7. These two bytes are reserved.

8. Two bytes are reserved for the local SCSI ID. The SCSI ID is stored in ASCII format.

9. Three bytes contain the speed of the MPU in MHz. For example, for a 66 MHz MPU, this field contains:

   ```
   066
   ```

10. Three bytes contain the speed of the BUS in MHz. For example, for a 33 MHz processor bus, this field contains:

    ```
    033
    ```

11. Growth space (187 bytes) is reserved. This pads the structure to an even 256 bytes. System-specific items may go here.

12. The final one byte of the area is reserved for a checksum (as defined in the *PPCBug Debugging Package User's Manual* for security and data integrity of the configuration area of the NVRAM. This data is stored in hexadecimal format.

Introduction

This chapter defines the VMEchip2 ASIC. The VMEchip2 interfaces an MC68040 style local bus to the VMEbus. The VME2PCI ASIC interfaces the PCI bus to an MC68040 style local bus. When the VMEchip2 and the VME2PCI chips are used together, they form a PCI bus to VMEbus interface. This chapter describes the local bus to VMEbus interface and the local bus is the bus between the VME2PCI chip and the VMEchip2. The local bus clock and the PCI clock are the same frequency.

The VMEchip2 interfaces the local bus to the VMEbus. In addition to the VMEbus defined functions, the VMEchip2 includes a local bus to VMEbus DMA controller, and Global Control and Status Registers (GCSR) for interprocessor communications.

Summary of Major Features

- Local Bus to VMEbus Interface:
  - Programmable local bus map decoder.
  - Programmable short, standard and extended VMEbus addressing.
  - Programmable AM codes.
  - Programmable 16-bit and 32-bit VMEbus data width.
  - Software-enabled write posting mode.
  - Write post buffer (one cache line or one four-byte).
  - Automatically performs dynamic bus sizing for VMEbus cycles.
  - Software-configured VMEbus access timers.

  - Local bus to VMEbus Requester:

    Software-enabled FAIR request mode.

    Software-configured release modes:
    Release-When-Done (RWD).
    Release-On-Request (ROR).

    Software-configured BR0*-BR3* request levels.

- VMEbus Bus to Local Bus Interface:
  - Programmable VMEbus map decoder.
  - Programmable AM decoder.
  - Simple VMEbus to local bus address translation.
  - 8-bit, 16-bit, and 32-bit VMEbus data width.
  - 8-bit, 16-bit, and 32-bit block transfer.
  - Standard and extended VMEbus addressing.
  - Software-enabled write posting mode.
  - Write post buffer (17 four-bytes in BLT mode, two four-bytes in non-BLT mode).
  - An eight four-byte read ahead buffer (BLT mode only).
- 32-Bit Local - VMEbus DMA Controller:
  - Programmable 16-bit, 32-bit, and 64-bit VMEbus data width.
  - Programmable short, standard and extended VMEbus addressing.
  - Programmable AM code.
  - A 16 four-byte FIFO data buffer.
  - Supports up to 4 GB of data per DMA request.
  - Automatically adjusts transfer size to optimize bus utilization.
  - DMA complete interrupt.
  - DMAC command chaining is supported by a singly-linked list of DMA commands.

- VMEbus DMA controller requester:

  Software-enabled FAIR request modes.

  Software-configured release modes:
  Release-On-Request (ROR).
  Release-On-End-Of-Data (ROEOD).

  Software-configured BR0-BR3 request levels.

  Software enabled bus-tenure timer.

- VMEbus Interrupter:
  - Software-configured IRQ1-IRQ7 interrupt request level.
  - 8-bit software-programmed status/ID register.
- VMEbus System Controller:
  - Auto SCON detect.
  - Arbiter with software-configured arbitration modes:
    Priority (PRI).
    Round-Robin-Select (RRS).
    Single-level (SGL).
  - Programmable arbitration timer.
  - IACK daisy-chain driver.
  - Programmable bus timer.
  - SYSRESET logic.
- Global Control Status Register Set:
  - Four location monitors.
  - Global control of locally detected failures.
  - Global control of local reset.
  - Four global attention interrupt bits.
  - A chip ID and revision register.
  - Four 16-bit dual-ported general purpose registers.
- Interrupt Handler:
  - All interrupts are level-programmable.
  - All interrupts are maskable.

  - All interrupts provide a unique vector.
  - Software and external interrupts.
- Watchdog timer.
- Two 32-bit tick timers.

Functional Blocks

The following sections provide an overview of the functions provided by the VMEchip2. See Figure 2-1 for a block diagram of the VMEchip2. A detailed programming model for the local control and status registers (LCSR) is provided in the following section. A detailed programming model for the global control and status registers (GCSR) is provided in the next section.

Local Bus to VMEbus Interface

The local bus to VMEbus interface allows local bus masters access to global resources on the VMEbus. This interface includes a *local bus slave*, a *write post buffer*, and a *VMEbus master*.

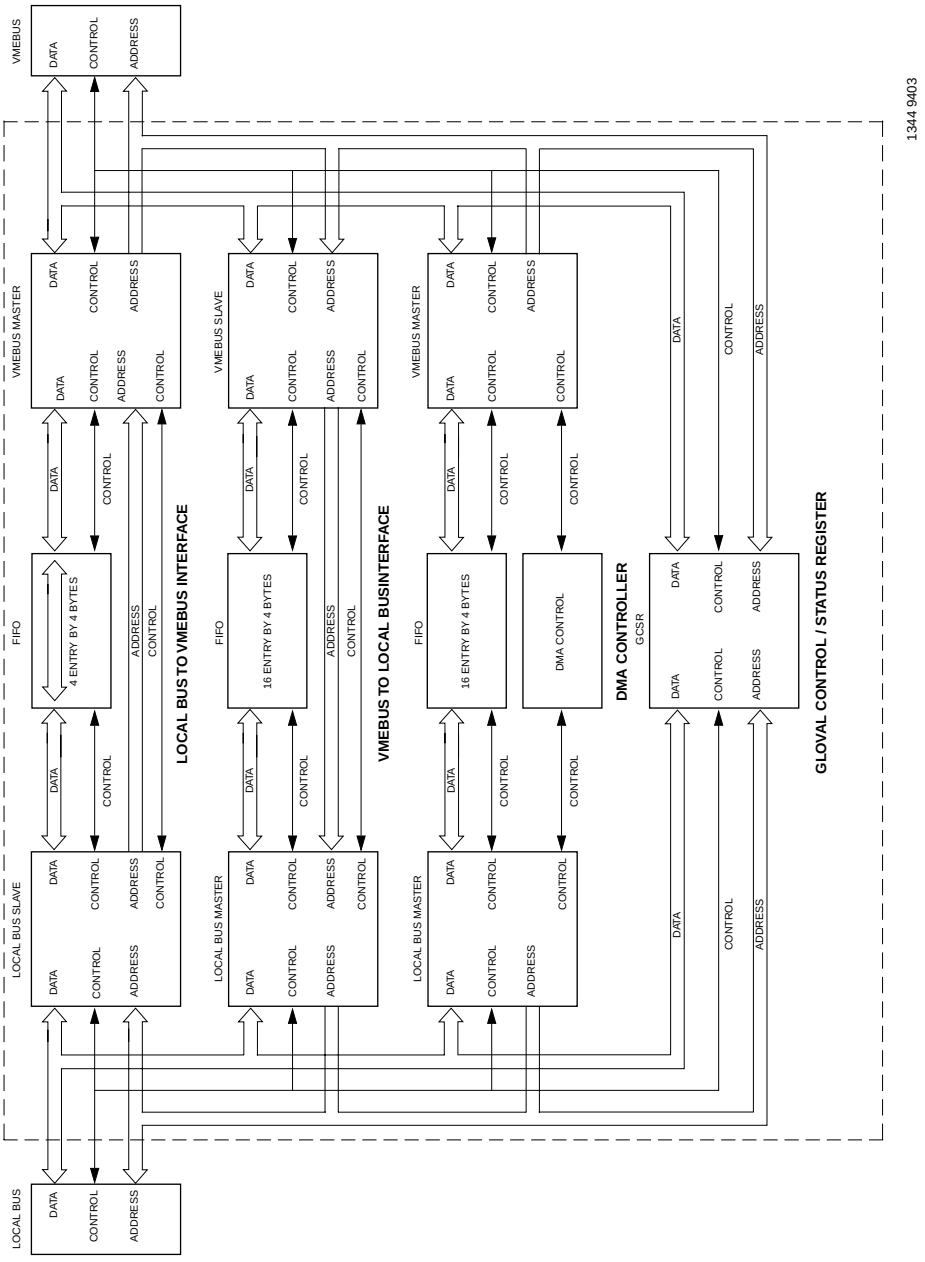


Figure 2-1. VMEchip2 Block Diagram

Using programmable map decoders with programmable attribute bits, the local bus to VMEbus interface can be configured to provide the following VMEbus capabilities:

Addressing capabilities: A16, A24, A32
Data transfer capabilities: D08, D16, D32

The *local bus slave* includes six local bus map decoders for accessing the VMEbus. The first four map decoders are general purpose programmable decoders, while the other two are fixed and are dedicated for I/O decoding.

The first four map decoders compare local bus address lines A31 through A16 with a 16-bit start address and a 16-bit end address. When an address in the selected range is detected, a VMEbus select is generated to the VMEbus master. Each map decoder also has eight attribute bits and an enable bit. The attribute bits are for VMEbus AM codes, D16 enable, and write post (WP) enable.

The fourth map decoder also includes a 16-bit alternate address register and a 16-bit alternate address select register. This allows any or all of the upper 16 address bits from the local bus to be replaced by bits from the alternate address register. The feature allows the local bus master to access any VMEbus address.

Using the four programmable map decoders, separate VMEbus maps can be created, each with its own attributes. For example, one map can be configured as A32, D32 with write posting enabled while a second map can be A24, D16 with write posting disabled.

The first I/O map decoder decodes local bus addresses $FFFF0000 through $FFFFFFFF as the short I/O A16/D16 or A16/D32 area, and the other provides an A24/D16 space at $F0000000 to $F0FFFFFF and an A32/D16 space at $F1000000 to $FF7FFFFF.

Supervisor/non-privileged and program/data space is determined by attribute bits. Write posting may be enabled or disabled for each decoder I/O space and this map decoder may be enabled or disabled.

When *write posting* is enabled, the VMEchip2 stores the local bus address and data and then acknowledges the local bus master. The local bus is then free to perform other operations while the VMEbus master requests the VMEbus and performs the requested operation.

The write post buffer stores one byte, two-byte, four-byte or one cache line four four-bytes). Write posting should only be enabled when bus errors are not expected. If a bus error is returned on a write posted cycle, the local processor is interrupted, if the interrupt is enabled. The address of the error is not saved. Normal memory never returns a bus error on a write cycle. However, some VMEbus ECC memory cards perform a read-modify-write operation and therefore may return a bus error if there is an error on the read portion of a read-modify-write. Write posting should not be enabled when this type of memory card is used. Also, memory should not be sized using write operations if write posting is enabled. I/O areas that have holes should not be write posted if software may access non-existent memory. Using the programmable map decoders, write posting can be enabled for "safe" areas and disabled for areas which are not "safe".

Using programmable map decoders with programmable attribute bits, the local bus to VMEbus interface can be configured to provide the following VMEbus capabilities:

Addressing capabilities: A16, A24, A32
Data transfer capabilities: D08, D16, D32

The *VMEbus master* supports dynamic bus sizing. When a local device initiates a quad-byte access to a VMEbus slave that only has the D16 data transfer capability, the chip executes two double-byte cycles on the VMEbus, acknowledging the local device after all requested four-bytes have been accessed. This enhances the portability of software because it allows software to run on the system regardless of the physical organization of global memory.

Using the local bus map decoder attribute register, the AM code that the master places on the VMEbus can be programmed under software control.

The VMEchip2 includes a software-controlled VMEbus access timer, and it starts ticking when the chip is requested to do a VMEbus data transfer or an interrupt acknowledge cycle. The timer stops ticking once the chip has started the data transfer on the VMEbus. If the data transfer does not begin before the timer times out, the timer drives the local bus error signal, and sets the appropriate status bit in the Local Control and Status Register (LCSR). Using control bits in the LCSR, the timer can be disabled, or it can be enabled to drive the local bus error signal after 64 µs, 1 ms, or 32 ms.

The VMEchip2 includes a software-controlled VMEbus write post timer, and it starts ticking when a data transfer to the VMEbus is write posted. The timer stops ticking once the chip has started the data transfer on the VMEbus. If this does not happen before the timer times out, the chip aborts the write posted cycle and send an interrupt to the local bus interrupter. If the write post bus error interrupt is enabled in the local bus interrupter, the local processor is interrupted to indicate a write post time-out has occurred. The write post timer has the same timing as the VMEbus access timer.

Local Bus to VMEbus Requester

The requester provides all the signals necessary to allow the local bus to VMEbus master to request and be granted use of the VMEbus. The chip connects to all signals that a VMEbus requester is required to drive and monitor.

Requiring no external jumpers, the chip provides the means for software to program the requester to request the bus on any one of the four bus request levels, automatically establishing the bus grant daisy-chains for the three inactive levels.

The requester requests the bus if any of the following conditions occur:

1. The local bus master initiates either a data transfer cycle or an interrupt acknowledge cycle to the VMEbus.
2. The chip is requested to acquire control of the VMEbus as signaled by the DWB input signal pin.

3. The chip is requested to acquire control of the VMEbus as signaled by the DWB control bit in the LCSR.

The local bus to VMEbus requester in the VMEchip2 implements a FAIR mode. By setting the LVFAIR bit, the requester refrains from requesting the VMEbus until it detects its assigned request line in its negated state.

The local bus to VMEbus requester attempts to release the VMEbus when the requested data transfer operation is complete, the DWB pin is negated, the DWB bit in the LCSR is negated and the bus is not being held by a lock cycle. The requester releases the bus as follows:

1. When the chip is configured in the release-when-done (RWD) mode, the requester releases the bus when the above conditions are satisfied.

2. When the chip is configured in the release-on-request (ROR) mode, the requester releases the bus when the above conditions are satisfied and there is a bus request pending on one of the VMEbus request lines.

To minimize the timing overhead of the arbitration process, the local bus to VMEbus requester in the VMEchip2 executes an early release of the VMEbus. If it is about to release the bus and it is executing a VMEbus cycle, the requester releases BBSY before its associated master completes the cycle. This allows the arbiter to arbitrate any pending requests, and grant the bus to the next requester, at the same time that the active master completes its cycle.

VMEbus to Local Bus Interface

The VMEbus to local bus interface allows an off-board VMEbus master access to onboard resources. The VMEbus to local bus interface includes the V*MEbus slave*, *write post buffer*, and *local bus master*.

Adhering to the IEEE 1014-87 VMEbus Standard, the *slave* can withstand address-only cycles, as well as address pipelining, and respond to unaligned transfers. Using programmable map decoders, it can be configured to provide the following VMEbus capabilities:

Addressing capabilities:	A24, A32
Data transfer capabilities:	D08(EO), D16, D32, D8/BLT, D16/BLT, D32/BLT, D64/BLT (BLT = block transfer)

The slave can be programmed to perform *write posting* operations. When in this mode, the chip latches incoming data and addressing information into a staging FIFO and then acknowledges the VMEbus write transfer by asserting DTACK. The chip then requests control of the local bus and independently accesses the local resource after it has been granted the local bus. The write-posting pipeline is two deep in the non-block transfer mode and 16 deep in the block transfer mode.

To significantly improve the access time of the slave when it responds to a VMEbus block read cycle, the VMEchip2 contains a 16 four-byte deep read-ahead pipeline. When responding to a block read cycle, the chip performs block read cycles on the local bus to keep the FIFO buffer full. Data for subsequent transfers is then retrieved from the on-chip buffer, significantly improving the response time of the slave in the block transfer mode.

The VMEchip2 includes an on-chip map decoder that allows software to configure the global addressing range of onboard resources. The decoder allows the local address range to be partitioned into two separate banks, each with its own start and end address (in increments of 64KB), as well as set each bank's address modifier codes and write post enable.

Each map decoder includes an alternate address register and an alternate address select register. These registers allow any or all of the upper 16 VMEbus address lines to be replaced by signals from the alternate address register. This allows the address of local resources to be different from their VMEbus address.

The alternate address register also provides the upper eight bits of the local address when the VMEbus slave cycle is A24.

The *local bus master* requests the local bus and executes cycles as required. To reduce local bus loading and improve performance it always attempts to transfer data using a burst transfer as defined by the MC68040.

Local Bus to VMEbus DMA Controller

The DMA Controller (DMAC) operates in conjunction with the local bus master, the VMEbus master, and a 16 four-byte FIFO buffer. The DMA controller has a 32-bit local address counter, 32-bit table address counter, a 32-bit VMEbus address counter, a 32-bit byte counter, and control and status registers. The Local Control and Status Register (LCSR) provides software with the ability to control the operational modes of the DMAC. Software can program the DMAC to transfer up to 4GB of data in the course of a single DMA operation. The DMAC supports transfers from any local bus address to any VMEbus address. The transfers may be from one byte to 4GB in length.

To optimize local bus use, the DMAC automatically adjusts the size of individual data transfers until 32-bit transfers can be executed. Based on the address of the first byte, the DMAC transfers a single-byte, a double-byte, or a mixture of both, and then continues to execute quad-byte block transfer cycles. When the DMAC is set for 64-bit transfers, the octal-byte transfers takes place. Based on the address of the last byte, the DMAC transfers a single-byte, a double-byte, or a mixture of both to end the transfer.

Using control register bits in the LCSR, the DMAC can be configured to provide the following VMEbus capabilities:

Addressing capabilities:	A16, A24, A32
Data transfer capabilities:	D16, D32, D16/BLT, D32/BLT, D64/BLT (BLT = block transfer)

Using the DMA AM control register, the address modifier code that the VMEbus DMA controller places on the VMEbus can be programmed under software control. In addition, the DMAC can be programmed to execute block-transfer cycles over the VMEbus.

Complying with the VMEbus specification, the DMAC automatically terminates block-transfer cycles whenever a 256-byte (D32/BLT) or 2-KB (D64/BLT) boundary is crossed. It does so by momentarily releasing AS and then, in accordance with its bus release/bus request configuration, initiating a new block-transfer cycle.

To optimize VMEbus use, the DMAC automatically adjusts the size of individual data transfers until 64-bit transfers (D64/BLT mode), 32-bit transfers (D32 mode) or 16-bit transfers (D16 mode) can be executed. Based on the address of the first byte, the DMAC transfers single-byte, double-byte, or a mixture of both, and then continues to execute transfer cycles based on the programmed data width. Based on the address of the last byte, the DMAC transfers single-byte, double-byte, or a mixture of both to end the transfer.

To optimize local bus use when the VMEbus is operating in the D16 mode, the data FIFO converts D16 VMEbus transfers to D32 local bus transfers. The FIFO also aligns data if the source and destination addresses are not aligned so the local bus and VMEbus can operate at their maximum data transfer sizes.

To allow other boards access to the VMEbus, the DMAC has bus tenure timers to limit the time the DMAC spends on the VMEbus and to ensure a minimum time off the VMEbus. Since the local bus is generally faster than the VMEbus, other local bus masters may use the local bus while the DMAC is waiting for the VMEbus.

The DMAC also supports command chaining through the use of a singly-linked list built in local memory. Each entry in the list includes a VMEbus address, a local bus address, a byte count, a control word, and a pointer to the next entry. When the command chaining mode is enabled, the DMAC reads and executes commands from the list in local memory until all commands are executed.

The DMAC can be programmed to send an interrupt request to the local bus interrupter when any specific table entry has completed. In addition the DMAC always sends an interrupt request at the normal completion of a request or when an error is detected. If the DMAC interrupt is enabled in the DMAC, the local bus is interrupted.

To allow increased flexibility in managing the bus tenure to optimize bus usage as required by the system configuration, the chip contains control bits that allow the DMAC time on and off the bus to be programmed. Using these control bits, software can instruct the DMA Controller to acquire the bus, maintain mastership for a specific amount of time, and then, after relinquishing it, refrain from requesting it for another specific amount of time.

DMAC VMEbus Requester

The chip contains an independent VMEbus requester associated with the DMA Controller. This allows flexibility in instituting different bus tenure policies for the single-transfer oriented master, and the block-transfer oriented DMA controller. The DMAC requester provides all the signals necessary to allow the on-chip DMA Controller to request and be granted use of the VMEbus.

Requiring no external jumpers, the chip provides the means for software to program the DMAC requester to request the bus on any one of the four bus request levels, automatically establishing the bus grant daisy-chains for the three inactive levels.

The DMAC requester requests the bus as required to transfer data to or from the FIFO buffer.

The requester implements a FAIR mode. By setting the DFAIR bit, the requester refrains from requesting the bus until it detects its assigned request line in its negated state.

The requester releases the bus when requested to by the DMA controller. The DMAC always releases the VMEbus when the FIFO is full (VMEbus to local bus) or empty (local bus to VMEbus). The DMAC can also be programmed to release the VMEbus when

another VMEbus master requests the bus, when the time on timer has expired, or when the time on timer has expired and another VMEbus master is requesting the bus. To minimize the timing overhead of the arbitration process, the DMAC requester executes an early release of the bus. If it is about to release the bus and it is executing a VMEbus cycle, the requester releases BBSY before its associated VMEbus master completes the cycle. This allows the arbiter to arbitrate any pending requests, and grant the bus to the next requester, at the same time that the DMAC completes its cycle.

Tick and Watchdog Timers

The VMEchip2 has two 32-bit tick timers and a watchdog timer. The tick timers run on a 1 MHz clock which is derived from the PCI bus clock by the prescaler.

Prescaler

The prescaler is used to derive the various clocks required by the tick timers, VME access timers, reset timer, bus arbitration timer, and VMEbus timer. The prescaler divides the local bus clock to produce the constant-frequency clocks required. Software is required to load the appropriate constant, depending upon the PCI bus clock, following reset to ensure proper operation of the prescaler.

Tick Timer

The VMEchip2 includes two general purpose tick timers. These timers can be used to generate interrupts at various rates or the counters can be read at various times for interval timing. The timers have a resolution of 1 µs and when free running, they roll over every 71.6 minutes.

Each tick timer has a 32-bit counter, a 32-bit compare register, a 4-bit overflow register, an enable bit, an overflow clear bit, and a clear-on-compare enable bit. The counter is readable and writable at any time and when enabled in the free run mode, it increments every 1 µs. When the counter is enabled in the clear-on-compare

mode, it increments every 1 µs until the counter value matches the value in the compare register. When a match occurs, the counter is cleared. When a match occurs, in either mode, an interrupt is sent to the local bus interrupter and the overflow counter is incremented. An interrupt to the local bus is only generated if the tick timer interrupt is enabled by the local bus interrupter. The overflow counter can be cleared by writing a one to the overflow clear bit.

Tick timer one or two can be programmed to generate a pulse on the VMEbus IRQ1 interrupt line at the tick timer period. This provides a broadcast interrupt function which allows several VME boards to receive an interrupt at the same time. In certain applications, this interrupt can be used to synchronize multiple processors. This interrupt is not acknowledged on the VMEbus. This mode is intended for specific applications and is not defined in the VMEbus specification.

Watchdog Timer

The watchdog timer has a 4-bit counter, four clock select bits, an enable bit, a local reset enable bit, a SYSRESET enable bit, a board fail enable bit, counter reset bit, WDTO status bit, and WDTO status reset bit.

When enabled, the counter increments at a rate determined by the clock select bits. If the counter is not reset by software, the counter reaches its terminal count. When this occurs, the WDTO status bit is set; and if the local or SYSRESET function is enabled, the selected reset is generated; if the board fail function is enabled, the board fail signal is generated.

VMEbus Interrupter

The interrupter provides all the signals necessary to allow software to request interrupt service from a VMEbus interrupt handler. The chip connects to all signals that a VMEbus interrupter is required to drive and monitor.

Requiring no external jumpers, the chip provides the means for software to program the interrupter to request an interrupt on any one of the seven interrupt request lines. In addition, the chip controls the propagation of the acknowledge on the IACK daisy-chain.

The interrupter operates in the release-on-acknowledge (ROAK) mode. An 8-bit control register provides software with the means to dynamically program the status/ID information. Upon reset, this register is initialized to a status/ID of $0F (the uninitialized vector in the 68K-based environment).

The VMEbus interrupter has an additional feature not defined in the VMEbus specification. The VMEchip2 supports a broadcast mode on the IRQ1 signal line. When this feature is used, the normal IRQ1 interrupt to the local bus interrupter should be disabled and the edge-sensitive IRQ1 interrupt to the local bus interrupter should be enabled. All boards in the system which are not participating in the broadcast interrupt function should not drive or respond to any signals on the IRQ1 signal line.

There are two ways to broadcast an IRQ1 interrupt. The VMEbus interrupter in the VMEchip2 may be programmed to generate a level one interrupt. This interrupt must be cleared using the interrupt clear bit in the control register because the interrupt is never acknowledged on the VMEbus. The VMEchip2 allows the output of one of the tick timers to be connected to the IRQ1 interrupt signal line on the VMEbus. When this function is enabled, a pulse appears on the IRQ1 signal line at the programmed interrupt rate of the tick timer.

VMEbus System Controller

With the exception of the optional SERCLK Driver and the Power Monitor, the chip includes all the functions that a VMEbus System Controller must provide. The System Controller function is controlled with the aid of an external jumper (the only jumper required in a VMEchip2 based VMEbus interface). The jumper can select enabled, disabled, or auto modes. The VME2PCI chip also

provides an auto System Controller detect feature. When the board is installed in slot 1, this feature will automatically enable the System Controller, when the jumper is in the auto position. NOTE: This feature does not work with S900 series chassis. When used with these chassis the jumper must be used to enable or disable the System Controller.

Arbiter

The arbitration algorithm used by the chip arbiter is selected by software. All three arbitration modes defined in the VMEbus Specification are supported: Priority (PRI), Round-Robin-Select (RRS), as well as Single (SGL). When operating in the PRI mode, the arbiter asserts the BCLR line whenever it detects a request for the bus whose level is higher that the one being serviced.

The chip includes an arbitration timer, preventing a bus lock-up when no requester assumes control of the bus after the arbiter has issued a grant. Using a control bit, this timer can be enabled or disabled. When enabled, it assumes control of the bus by driving the BBSY signal after 256 µseconds, releasing it after satisfying the requirements of the VMEbus specification, and then re-arbitrating any pending bus requests.

IACK Daisy-Chain Driver

Complying with the latest revision of the VMEbus specification, the System Controller includes an IACK Daisy-Chain Driver, ensuring that the timing requirements of the IACK daisy-chain are satisfied.

Bus Timer

The Bus Timer is enabled/disabled by software to terminate a VMEbus cycle by asserting BERR if any of the VMEbus data strobes is maintained in its asserted state for longer than the programmed time-out period. The time-out period can be set to 8, 64, or 256 µsecs. The bus timer terminates an unresponded VMEbus cycle only if both it and the system controller are enabled.

Reset Driver

The chip includes both a global and a local reset driver. When the chip operates as the VMEbus system controller, the reset driver provides a global system reset by asserting the VMEbus signal SYSRESET. A SYSRESET may be generated by the reset switch, a power up reset, a watch dog time-out, or by a control bit in the LCSR. SYSRESET remains asserted for at least 200 msec, as required by the VMEbus specification.

Similarly, the chip provides an input signal and a control bit to initiate a local reset operation. By setting a control bit, software can maintain a board in a reset state, disabling a faulty board from participating in normal system operation. The local reset driver is enabled even when the chip is not the system controller. A local reset may be generated by the reset switch, a power up reset, a watch dog time-out, a VMEbus SYSRESET, or a control bit in the GCSR.

Local Bus Interrupter and Interrupt Handler

There are 31 interrupt sources in the VMEchip2: VMEbus ACFAIL, ABORT switch, VMEbus SYSFAIL, write post bus error, external input, VMEbus IRQ1 edge-sensitive, VMEchip2 VMEbus interrupter acknowledge, tick timer 2-1, DMAC done, GCSR SIG3-0, GCSR location monitor 1-0, software interrupts 7-0, and VMEbus IRQ7-1. Each of the 31 interrupts can be enabled to generate a local bus interrupt at any level. For example, VMEbus IRQ5 can be programmed to generate a level 2 local bus interrupt.

The VMEbus AC fail interrupter is an edge-sensitive interrupter connected to the VMEbus ACFAIL signal line. This interrupter is filtered to remove the ACFAIL glitch which is related to the BBSY glitch.

The ABORT switch interrupter is not used.

The SYS fail interrupter is an edge-sensitive interrupter connected to the VMEbus SYSFAIL signal line.

The write post bus error interrupter is an edge-sensitive interrupter connected to the local bus to VMEbus write post bus error signal line.

The external interrupter is not used.

The VMEbus IRQ1 edge-sensitive interrupter is an edge-sensitive interrupter connected to the VMEbus IRQ1 signal line. This interrupter is used when one of the tick timers is connected to the IRQ1 signal line. When this interrupt is acknowledged, the vector is provided by the VMEchip2 and a VMEbus interrupt acknowledge is not generated. When this interrupt is enabled, the VMEbus IRQ1 level-sensitive interrupter should be disabled.

The VMEchip2 VMEbus interrupter acknowledge interrupter is an edge-sensitive interrupter connected to the acknowledge output of the VMEbus interrupter. An interrupt is generated when an interrupt on the VMEbus from VMEchip2 is acknowledged by a VMEbus interrupt handler.

The tick timer interrupters are edge-sensitive interrupters connected to the output of the tick timers.

The DMAC interrupter is an edge-sensitive interrupter connected to the DMAC.

The GCSR SIG3-0 interrupters are edge-sensitive interrupters connected to the output of the signal bits in the GCSR.

The location monitor interrupters are edge-sensitive interrupters connected to the location monitor bits in the GCSR.

The software 7-0 interrupters can be set by software to generate interrupts.

The VMEbus IRQ7-1 interrupters are level-sensitive interrupters connected to the VMEbus IRQ7-1 signal lines.

The interrupt handler provides all logic necessary to identify and handle all local interrupts as well as VMEbus interrupts. When a local interrupt is acknowledged, a unique vector is provided by the chip. Edge-sensitive interrupters are not cleared during the

interrupt acknowledge cycle and must by reset by software as required. If the interrupt source is the VMEbus, the interrupt handler instructs the VMEbus master to execute a VMEbus IACK cycle to obtain the vector from the VMEbus interrupter. The chip connects to all signals that a VMEbus handler is required to drive and monitor. On the local bus, the interrupt handler is designed to comply with the interrupt handling signaling protocol of the MC68040 microprocessor.

Global Control and Status Registers

The VMEchip2 includes a set of registers that are accessible from both the VMEbus and the local bus. These registers are provided to aid in interprocessor communications over the VMEbus. These registers are fully described in a later section.

LCSR Programming Model

This section defines the programming model for the Local Control and Status Registers (LCSR) in the VMEchip2. The local bus map decoder for the LCSR is included in the VMEchip2. The base address of the LCSR is $FFF40000 from the local bus and BASE+0000 from the PCI bus. Base is programmed in the VME2PCI chip. The registers are 32 bits wide. Byte, two-byte and four-byte read operations are permitted: however, byte and two-byte write operations are not permitted. Byte and two-byte write operations return a TEA signal to the local bus. Read-modify-write operations should be used to modify a byte or a two-byte of a register.

Each register definition includes a table with 5 lines:

- Line 1 is the base address of the register and the number of bits defined in the table.
- Line 2 shows the bits defined by this table.
- Line 3 defines the name of the register or the name of the bits in the register.

- Line 4 defines the operations possible on the register bits as follows:

R	This bit is a read-only status bit.
R/W	This bit is readable and writable.
W/AC	This bit can be set and it is automatically cleared. This bit can also be read.
C	Writing a one to this bit clears this bit or another bit. This bit reads zero.
S	Writing a one to this bit sets this bit or another bit. This bit reads zero.

- Line 5 defines the state of the bit following a reset as follows:

P	The bit is affected by powerup reset.
S	The bit is affected by SYSRESET.
L	The bit is affected by local reset.
X	The bit is not affected by reset.

A summary of the LCSR is shown in Table 2-1.

2

Table 2-1. VMEchip2 Memory Map - LCSR Summary (Sheet 1 of 2)

VMEchip2 LCSR Base Address = BASE+0000

OFFSET:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	SLAVE ENDING ADDRESS 1															
4	SLAVE ENDING ADDRESS 2															
8	SLAVE ADDRESS TRANSLATION ADDRESS 1															
C	SLAVE ADDRESS TRANSLATION ADDRESS 2															
10					ADDER 2	SNP 2			WP 2	SUP 2	USR 2	A32 2	A24 2	BLK D64 2	BLK 2	PRGM 2	DATA2
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
14	MASTER ENDING ADDRESS 1															
18	MASTER ENDING ADDRESS 2															
1C	MASTER ENDING ADDRESS 3															
20	MASTER ENDING ADDRESS 4															
24	MASTER ADDRESS TRANSLATION ADDRESS 4															
28	MAST D16 EN	MAST WP EN	MASTER AM 4						MAST D16 EN	MAST WP EN	MASTER AM 3					
2C	GCSR GROUP SELECT								GCSR BOARD SELECT				MAST 4 EN	MAST 3 EN	MAST 2 EN	MAST 1 EN
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
30											WAIT RMW	ROM ZERO	DMA TB SNP MODE		SRAM SPEED	
34																
38	DMA CONTROLLER															
3C	DMA CONTROLLER															
40	DMA CONTROLLER															
44	DMA CONTROLLER															
48		TICK 2/1	TICK IRQ 1 EN	CLR IRQ	IRQ STAT	VMEBUS INTERRUPT LEVEL			VMEBUS INTERRUPT VECTOR							

This sheet continues on facing page. ⟶

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLAVE STARTING ADDRESS 1															
SLAVE STARTING ADDRESS 2															
SLAVE ADDRESS TRANSLATION SELECT 1															
SLAVE ADDRESS TRANSLATION SELECT 2															
				ADDER 1	SNP 1			WP 1	SUP 1	USR 1	A32 1	A24 1	BLK D64 1	BLK 1	PRGM 1	DATA1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASTER STARTING ADDRESS 1															
MASTER STARTING ADDRESS 2															
MASTER STARTING ADDRESS 3															
MASTER STARTING ADDRESS 4															
MASTER ADDRESS TRANSLATION SELECT 4															
MAST D16 EN	MAST WP EN	MASTER AM 2						MAST D16 EN	MAST WP EN	MASTER AM 1					
IO2 EN	IO2 WP EN	IO2 S/U	IO2 P/D	IO1 EN	IO1 D16 EN	IO1 WP EN	IO1 S/U	ROM SIZE		ROM BANK B SPEED			ROM BANK A SPEED		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARB ROBN	MAST DHB	MAST DWB		MST FAIR	MST RWD	MASTER VMEBUS		DMA HALT	DMA EN	DMA TBL	DMA FAIR	DM RELM		DMA VMEBUS	
DMA TBL INT	DMA LB SNP MODE			DMA INC VME	DMA INC LB	DMA WRT	DMA D16	DMA D64 BLK	DMA BLK	DMA AM 5	DMA AM 4	DMA AM 3	DMA AM 2	DMA AM 1	DMA AM 0
LOCAL BUS ADDRESS COUNTER															
VMEBUS ADDRESS COUNTER															
BYTE COUNTER															
TABLE ADDRESS COUNTER															
DMA TABLE INTERRUPT COUNT				MPU CLR STAT	MPU LBE ERR	MPU LPE ERR	MPU LOB ERR	MPU LTO ERR	DMA LBE ERR	DMA LPE ERR	DMA LOB ERR	DMA LTO ERR	DMA TBL ERR	DMA VME ERR	DMA DONE

1360 9403

← This sheet begins on facing page.

2

Table 2-1. VMEchip2 Memory Map - LCSR Summary (Sheet 2 of 2)

VMEchip2 LCSR Base Address = BASE+0000
OFFSET:

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
4C								ARB BGTO EN	DMA TIME OFF			DMA TIME ON			VME GLOBAL TIMER	
50	TICK TIMER 1															
54	TICK TIMER 1															
58	TICK TIMER 2															
5C	TICK TIMER 2															
60		SCON	SYS FAIL	BRD FAIL STAT	PURS STAT	CLR PURS STAT	BRD FAIL OUT	RST SW EN	SYS RST	WD CLR TO	WD CLR CNT	WD TO STAT	TO BF EN	WD SRST LRST	WD RST EN	WD EN
64	PRE															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
68	AC FAIL IRQ	AB IRQ	SYS FAIL IRQ	MWP BERR IRQ	PE IRQ	IRQ1E IRQ	TIC2 IRQ	TIC1 IRQ	VME IACK IRQ	DMA IRQ	SIG3 IRQ	SIG2 IRQ	SIG1 IRQ	SIG0 IRQ	LM1 IRQ	LM0 IRQ
6C	EN IRQ 31	EN IRQ 30	EN IRQ 29	EN IRQ 28	EN IRQ 27	EN IRQ 26	EN IRQ 25	EN IRQ 24	EN IRQ 23	EN IRQ 22	EN IRQ 21	EN IRQ 20	EN IRQ 19	EN IRQ 18	EN IRQ 17	EN IRQ 16
70																
74	CLR IRQ 31	CLR IRQ 30	CLR IRQ 29	CLR IRQ 28	CLR IRQ 27	CLR IRQ 26	CLR IRQ 25	CLR IRQ 24	CLR IRQ 23	CLR IRQ 22	CLR IRQ 21	CLR IRQ 20	CLR IRQ 19	CLR IRQ 18	CLR IRQ 17	CLR IRQ 16
78		AC FAIL IRQ LEVEL				ABORT IRQ LEVEL				SYS FAIL IRQ LEVEL				MST WP ERROR IRQ LEVEL		
7C		VME IACK IRQ LEVEL				DMA IRQ LEVEL				SIG 3 IRQ LEVEL				SIG 2 IRQ LEVEL		
80		SW7 IRQ LEVEL				SW6 IRQ LEVEL				SW5 IRQ LEVEL				SW4 IRQ LEVEL		
84		SPARE IRQ LEVEL				VME IRQ 7 IRQ LEVEL				VME IRQ 6 IRQ LEVEL				VME IRQ 5 IRQ LEVEL		
88	VECTOR BASE REGISTER 0				VECTOR BASE REGISTER 1				MST IRQ EN	SYS FAIL LEVEL	AC FAIL LEVEL	ABORT LEVEL	GPIOEN			
8C																

This sheet continues on facing page. ⟶

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VME ACCESS TIMER		LOCAL BUS TIMER		WD TIME OUT SELECT				PRESCALER CLOCK ADJUST							
COMPARE REGISTER															
COUNTER															
COPARE REGISTER															
COUNTER															
OVERFLOW COUNTER 2				X	CLR OVF 2	COC EN 2	TIC EN 2	OVERFLOW COUNTER 1				X	CLR OVF 1	COC EN 1	TIC EN 1
SCALER															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SW7 IRQ	SW6 IRQ	SW5 IRQ	SW4 IRQ	SW3 IRQ	SW2 IRQ	SW1 IRQ	SW0 IRQ	SPARE	VME IRQ7	VME IRQ6	VME IRQ5	VME IRQ4	VME IRQ3	VME IRQ2	VME IRQ1
EN IRQ 15	EN IRQ 14	EN IRQ 13	EN IRQ 12	EN IRQ 11	EN IRQ 10	EN IRQ 9	EN IRQ 8	EN IRQ 7	EN IRQ 6	EN IRQ 5	EN IRQ 4	EN IRQ 3	EN IRQ 2	EN IRQ 1	EN IRQ 0
SET IRQ 15	SET IRQ 14	SET IRQ 13	SET IRQ 12	SET IRQ 11	SET IRQ 10	SET IRQ 9	SET IRQ 8								
CLR IRQ 15	CLR IRQ 14	CLR IRQ 13	CLR IRQ 12	CLR IRQ 11	CLR IRQ 10	CLR IRQ 9	CLR IRQ 8								
X	P ERROR IRQ LEVEL			X	IRQ1E IRQ LEVEL			X	TIC TIMER 2 IRQ LEVEL			X	TIC TIMER 1 IRQ LEVEL		
X	SIG 1 IRQ LEVEL			X	SIG 0 IRQ LEVEL			X	LM 1 IRQ LEVEL			X	LM 0 IRQ LEVEL		
X	SW3 IRQ LEVEL			X	SW2 IRQ LEVEL			X	SW1 IRQ LEVEL			X	SW0 IRQ LEVEL		
X	VME IRQ 4 IRQ LEVEL			X	VMEB IRQ 3 IRQ LEVEL			X	VME IRQ 2 IRQ LEVEL			X	VME IRQ 1 IRQ LEVEL		
GPIOO				GPIOI				GPI							
								MP IRQ EN	REV EROM	DIS SRAM	DIS MST	NO EL BBSY	DIS BSYT	EN INT	DIS BGN

1361 9403

← This sheet begins on facing page.

2

Programming the VMEbus Slave Map Decoders

This section includes programming information for the VMEbus to local bus map decoders.

The VMEbus to local bus interface allows off-board VMEbus masters access to local onboard resources. The address of the local resources as viewed from the VMEbus is controlled by the VMEbus slave map decoders, which are part of the VMEbus to local bus interface. Two VMEbus slave map decoders in the VMEchip2 allow two segments of the VMEbus to be mapped to the local bus. A segment may vary in size from 64KB to 4GB in increments of 64KB. Address translation is provided by the address translation registers which allow the upper 16 bits of the local bus address to be provided by the address translation address register rather than the upper 16 bits of the VMEbus.

Each VMEbus slave map decoder has the following registers: *address translation address register, address translation select register*, *starting address register, ending address register*, *address modifier select register*, and *attribute register*. The addresses and bit definitions of these registers are shown in the following tables.

The VMEbus slave map decoders described in this section are disabled by local reset, SYSRESET, or power-up reset. Caution must be used when enabling the map decoders or when modifying their registers after they are enabled. The safest time to enable or modify the map decoder registers is when the VMEchip2 is VMEbus master. The following procedure should be used to modify the map decoder registers: Set the DWB bit in the LCSR and then wait for the DHB bit in the LCSR to be set, indicating that VMEbus mastership has been acquired. The map decoder registers can then be modified and the VMEbus released by clearing the DWB bit in the LCSR. Because the VMEbus is held during this programming operation, the registers should be programmed quickly with interrupts disabled.

The VMEbus slave map decoders can be programmed, without obtaining VMEbus mastership, if they are disabled and the following procedure is followed: The address translation registers

and starting and ending address registers should be programmed first, and then the map decoders should be enabled by programming the address modifier select registers.

A VMEbus slave map decoder is programmed by loading the starting address of the segment into the *starting address register* and the ending address of the segment into the *ending address register*. If the VMEbus address modifier codes indicate an A24 VMEbus address cycle, then the upper eight bits of the VMEbus address are forced to zero before the compare. The address modifier select register should be programmed for the required address modifier codes. A VMEbus slave map decoder is disabled when the address modifier select register is cleared.

The *address translation registers* allow local resources to have different VMEbus and local bus addresses. Only address bits A31 through A16 may be modified.

The *address translation registers* also provide the upper eight local bus address lines when an A24 VMEbus cycle is used to accesses a local resource. The address translation register should be programmed with the translated address and the address translation select register should be programmed to enable the translated address. If address translation is not desired, then the address translation registers should be programmed to zero.

The *address translation address register* and the *address translation select register* operate in the following way: If a bit in the address translation select register is set, then the corresponding local bus address line is driven from the corresponding bit in the address translation address register. If the bit is cleared in the address translation select register, then the corresponding local bus address line is driven from the corresponding VMEbus address line. The most significant bit of the address translation select register corresponds to the most significant bit of address translation register and to A32 of the local bus and A32 of the VMEbus.

In addition to the address translation method previously described, the VMEchip2 includes an adder which can be used for address translation. When the adder is enabled, the local bus address is

generated by adding the offset value to the VMEbus address lines VA<31..16>. The offset is the value in the address translation/offset register. If the VMEbus transfer is A24, then the VMEbus address lines VA<31..24> are forced to 0 before the add. The adders are enable by setting bit 11 for map decoder 1 and bit 27 for map decoder 2 in register BASE+0010. The adders allow any size board to be mapped on any 64KB boundary. The adders are disabled and the address replacement method is used following reset.

Write posting is enabled for the segment by setting the write post enable bit in the *attribute register*.

VMEbus Slave Ending Address Register 1

ADR/SIZ	BASE+0000 (16 bits of 32)		
BIT	31	. . .	16
NAME	Ending Address Register 1		
OPER	R/W		
RESET	0 PS		

This register is the ending address register for the first VMEbus to local bus map decoder.

VMEbus Slave Starting Address Register 1

ADR/SIZ	BASE+0000 (16 bits of 32)		
BIT	15	. . .	0
NAME	Starting Address Register 1		
OPER	R/W		
RESET	0 PS		

This register is the starting address register for the first VMEbus to local bus map decoder.

VMEbus Slave Ending Address Register 2

ADR/SIZ	BASE+0004 (16 bits of 32)		
BIT	31	. . .	16
NAME	Ending Address Register 2		
OPER	R/W		
RESET	0 PS		

This register is the ending address register for the second VMEbus to local bus map decoder.

VMEbus Slave Starting Address Register 2

ADR/SIZ	BASE+0004 (16 bits of 32)		
BIT	15	. . .	0
NAME	Starting Address Register 2		
OPER	R/W		
RESET	0 PS		

This register is the starting address register for the second VMEbus to local bus map decoder.

VMEbus Slave Address Translation Address Offset Register 1

ADR/SIZ	BASE+0008 (16 bits of 32)		
BIT	31	. . .	16
NAME	Address Translation Address Offset Register 1		
OPER	R/W		
RESET	0 PS		

This register is the address translation address register for the first VMEbus to local bus map decoder. It should be programmed to the local bus starting address. When the adder is engaged, this register is the offset value.

VMEbus Slave Address Translation Select Register 1

ADR/SIZ	BASE+0008 (16 bits of 32)		
BIT	15	. . .	0
NAME	Address Translation Select Register 1		
OPER	R/W		
RESET	0 PS		

This register is the address translation select register for the first VMEbus to local bus map decoder. The address translation select register value is based on the segment size (the difference between the VMEbus starting and ending addresses). If the segment size is between the sizes shown in the table below, assume the larger size.

Segment Size	Address Translation Select Value	Segment Size	Address Translation Select Value
64KB	FFFF	32MB	FE00
128KB	FFFE	64MB	FC00
256KB	FFFC	128MB	F800
512KB	FFF8	256MB	F000
1MB	FFF0	512MB	E000
2MB	FFE0	1GB	C000
4MB	FFC0	2GB	8000
8MB	FF80	4GB	0000
16MB	FF00		

VMEbus Slave Address Translation Address Offset Register 2

ADR/SIZ	BASE+000C (16 bits of 32)		
BIT	31	. . .	16
NAME	Address Translation Address Offset Register 2		
OPER	R/W		
RESET	0 PS		

This register is the address translation address register for the second VMEbus to local bus map decoder. It should be programmed to the local bus starting address. When the adder is enabled, this register is the offset value.

VMEbus Slave Address Translation Select Register 2

ADR/SIZ	BASE+000C (16 bits of 32)		
BIT	15	. . .	0
NAME	Address Translation Select Register 2		
OPER	R/W		
RESET	0 PS		

This register is the address translation select register for the second VMEbus to local bus map decoder. The address translation select register value is based on the segment size (the difference between the VMEbus starting and ending addresses). If the segment size is between the sizes shown in the table below, assume the larger size.

Segment Size	Address Translation Select Value	Segment Size	Address Translation Select Value
64KB	FFFF	32MB	FE00
128KB	FFFE	64MB	FC00
256KB	FFFC	128MB	F800
512KB	FFF8	256MB	F000
1MB	FFF0	512MB	E000
2MB	FFE0	1GB	C000
4MB	FFC0	2GB	8000
8MB	FF80	4GB	0000
16MB	FF00		

VMEbus Slave Write Post and Snoop Control Register 2

ADR/SIZ	BASE+0010 (8 bits [4 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME					ADDER2	SNP2		WP2
OPER					R/W	R/W		R/W
RESET					0 PS	0 PS		0 PS

This register is the slave write post and snoop control register for the second VMEbus to local bus map decoder.

WP2 When this bit is high, write posting is enabled for the address range defined by the second VMEbus slave map decoder. When this bit is low, write posting is disabled for the address range defined by the second VMEbus slave map decoder.

SNP2 These bits are not used.

ADDER2 When this bit is high, the adder is used for address translation. When this bit is low, the adder is not used for address translation.

VMEbus Slave Address Modifier Select Register 2

ADR/SIZ	BASE+0010 (8 bits of 32)							
BIT	23	22	21	20	19	18	17	16
NAME	SUP	USR	A32	A24	D64	BLK	PGM	DAT
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL

This register is the address modifier select register for the second VMEbus to local bus map decoder. There are three groups of address modifier select bits: DAT, PGM, BLK and D64; A24 and A32; and USR and SUP. At least one bit must be set from each group to enable the map decoder.

DAT When this bit is high, the second map decoder responds to VMEbus data access cycles. When this bit is low, the second map decoder does not respond to VMEbus data access cycles.

PGM When this bit is high, the second map decoder responds to VMEbus program access cycles. When this bit is low, the second map decoder does not respond to VMEbus program access cycles.

BLK When this bit is high, the second map decoder responds to VMEbus block access cycles. When this bit is low, the second map decoder does not respond to VMEbus block access cycles.

D64 When this bit is high, the second map decoder responds to VMEbus D64 block access cycles. When this bit is low, the second map decoder does not respond to VMEbus D64 block access cycles.

A24 When this bit is high, the second map decoder responds to VMEbus A24 (standard) access cycles. When this bit is low, the second map decoder does not respond to VMEbus A24 access cycles.

A32 When this bit is high, the second map decoder responds to VMEbus A32 (extended) access cycles. When this bit is low, the second map decoder does not respond to VMEbus A32 access cycles.

USR When this bit is high, the second map decoder responds to VMEbus user (non-privileged) access cycles. When this bit is low, the second map decoder does not respond to VMEbus user access cycles.

SUP When this bit is high, the second map decoder responds to VMEbus supervisory access cycles. When this bit is low, the second map decoder does not respond to VMEbus supervisory access cycles.

VMEbus Slave Write Post and Snoop Control Register 1

ADR/SIZ	BASE+0010 (8 bits [4 used] of 32)							
BIT	15	14	13	12	11	10	9	8
NAME					ADDER1	SNP1		WP1
OPER					R/W	R/W		R/W
RESET					0 PS	0 PS		0 PS

This register is the slave write post and snoop control register for the first VMEbus to local bus map decoder.

WP1 When this bit is high, write posting is enabled for the address range defined by the first VMEbus slave map decoder. When this bit is low, write posting is disabled for the address range defined by the first VMEbus slave map decoder.

SNP1 These bits are not used.

ADDER1 When this bit is high, the adder is used for address translation. When this bit is low, the adder is not used for address translation.

VMEbus Slave Address Modifier Select Register 1

ADR/SIZ	BASE+0010 (8 bits of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	SUP	USR	A32	A24	D64	BLK	PGM	DAT
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL

This register is the address modifier select register for the first VMEbus to local bus map decoder. There are three groups of address modifier select bits: DAT, PGM, BLK and D64; A24 and A32; and USR and SUP. At least one bit must be set from each group to enable the first map decoder.

DAT When this bit is high, the first map decoder responds to VMEbus data access cycles. When this bit is low, the first map decoder does not responded to VMEbus data access cycles.

PGM When this bit is high, the first map decoder responds to VMEbus program access cycles. When this bit is low, the first map decoder does not respond to VMEbus program access cycles.

BLK When this bit is high, the first map decoder responds to VMEbus block access cycles. When this bit is low, the first map decoder does not respond to VMEbus block access cycles.

D64 When this bit is high, the first map decoder responds to VMEbus D64 block access cycles. When this bit is low, the first map decoder does not respond to VMEbus D64 block access cycles.

A24 When this bit is high, the first map decoder responds to VMEbus A24 (standard) access cycles. When this bit is low, the first map decoder does not respond to VMEbus A24 access cycles.

A32 When this bit is high, the first map decoder responds to VMEbus A32 (extended) access cycles. When this bit is low, the first map decoder does not respond to VMEbus A32 access cycles.

USR When this bit is high, the first map decoder responds to VMEbus user (non-privileged) access cycles. When this bit is low, the first map decoder does not respond to VMEbus user access cycles.

SUP When this bit is high, the first map decoder responds to VMEbus supervisory access cycles. When this bit is low, the first map decoder does not respond to VMEbus supervisory access cycles.

Programming the Local Bus to VMEbus Map Decoders

This section includes programming information on the local bus to VMEbus map decoders and the GCSR base address registers.

The local bus to VMEbus interface allows onboard local bus masters access to off-board VMEbus resources. The address of the VMEbus resources as viewed from the local bus is controlled by the local bus slave map decoders, which are part of the local bus to VMEbus interface. Four of the six local bus to VMEbus map decoders are programmable, while the two I/O map decoders are fixed. The first I/O map decoder provides an A16/D16 or A16/D32 space at $FFFF0000 to $FFFFFFFF which is the VMEbus short I/O

space. The second I/O map decoder provides an A24/D16 space at $F000000 to $F0FFFFFF and an A32/D16 space at $F1000000 to $FF7FFFFF.

A programmable segment may vary in size from 64KB to 4GB in increments of 64KB. Address translation for the fourth segment is provided by the address translation registers which allow the upper 16 bits of the VMEbus address to be provided by the address translation address register rather than the upper 16 bits of the local bus.

Each of the four programmable local bus map decoders has a starting address, an ending address, an address modifier register with attribute bits, and an enable bit. The fourth decoder also has address translation registers. The addresses and bit definitions for these registers are in the tables below.

A local bus slave map decoder is programmed by loading the starting address of the segment into the starting address register and the ending address of the segment into the ending address register. The address modifier code is programmed in to the address modifier register. Because the local bus to VMEbus interface does not support VMEbus block transfers, block transfer address modifier codes should not be programmed.

The address translation register allows a local bus master to view a portion of the VMEbus that may be hidden by onboard resources or an area of the VMEbus may be mapped to two local address. For example, some devices in the I/O map may support write posting while others do not. The VMEbus area in question may be mapped to two local bus addresses, one with write posting enabled and one with write posting disabled. The address translation registers allow local bus address bits A31 through A16 to be modified. The address translation register should be programmed with the translated address, and the address translation select register should be programmed to enable the translated address. If address translation is not desired, then the address translation registers should be programmed to zero.

The address translation address register and the address translation select register operate in the following way. If a bit in the address translation select register is set, then the corresponding VMEbus address line is driven from the corresponding bit in the address translation address register. If the bit is cleared in the address translation select register, then the corresponding VMEbus address line is driven from the corresponding local bus address line. The most significant bit of the address translation select register corresponds to the most significant bit of address translation address register and to A32 of the local bus and A32 of the VMEbus.

Write posting is enabled for the segment by setting the write post enable bit in the address modifier register. D16 transfers are forced by setting the D16 bit in the address modifier register. A segment is enabled by setting the enable bit. Segments should not be programmed to overlap.

The first I/O map decoder maps the local bus address range $FFFF0000 to $FFFFFFFF to the A16 (short I/O) map of the VMEbus. This segment may be enabled using the enable bit. Write posting may be enabled for this segment using the write post enable bit. The transfer size may be D16 or D32 as defined by the D16 bit in the control register.

The second I/O map decoder provides support for the other I/O map of the VMEbus. This decoder maps the local bus address range $F0000000 to $F0FFFFFF to the A24 map of the VMEbus and the address range $F1000000 to $FF7FFFFF to the A32 map of the VMEbus. The transfer size is always D16. This segment may be enabled using the enable bit. Write posting may be enabled using the write post enable bit.

The local bus map decoders should not be programmed such that more than one map decoder responds to the same local bus address or a map decoder conflicts with on board resources. However, the map decoders may be programmed to allow a VMEbus address to be accessed from more than one local bus address.

Local Bus Slave (VMEbus Master) Ending Address Register 1

ADR/SIZ	BASE+0014 (16 bits of 32)		
BIT	31	. . .	16
NAME	Ending Address Register 1		
OPER	R/W		
RESET	0 PS		

This register is the ending address register for the first local bus to VMEbus map decoder.

Local Bus Slave (VMEbus Master) Starting Address Register 1

ADR/SIZ	BASE+0014 (16 bits of 32)		
BIT	15	. . .	0
NAME	Starting Address Register 1		
OPER	R/W		
RESET	0 PS		

This register is the starting address register for the first local bus to VMEbus map decoder.

Local Bus Slave (VMEbus Master) Ending Address Register 2

ADR/SIZ	BASE+0018 (16 bits of 32)		
BIT	31	. . .	16
NAME	Ending Address Register 2		
OPER	R/W		
RESET	0 PS		

This register is the ending address register for the second local bus to VMEbus map decoder.

Local Bus Slave (VMEbus Master) Starting Address Register 2

ADR/SIZ	BASE+0018 (16 bits of 32)		
BIT	15	. . .	0
NAME	Starting Address Register 2		
OPER	R/W		
RESET	0 PS		

This register is the starting address register for the second local bus to VMEbus map decoder.

Local Bus Slave (VMEbus Master) Ending Address Register 3

ADR/SIZ	BASE+001C (16 bits of 32)		
BIT	31	. . .	16
NAME	Ending Address Register 3		
OPER	R/W		
RESET	0 PS		

This register is the ending address register for the third local bus to VMEbus map decoder.

Local Bus Slave (VMEbus Master) Starting Address Register 3

ADR/SIZ	BASE+001C (16 bits of 32)		
BIT	15	. . .	0
NAME	Starting Address Register 3		
OPER	R/W		
RESET	0 PS		

This register is the starting address register for the third local bus to VMEbus map decoder.

Local Bus Slave (VMEbus Master) Ending Address Register 4

ADR/SIZ	BASE+0020 (16 bits of 32)		
BIT	31	. . .	16
NAME	Ending Address Register 4		
OPER	R/W		
RESET	0 PS		

This register is the ending address register for the fourth local bus to VMEbus map decoder.

Local Bus Slave (VMEbus Master) Starting Address Register 4

ADR/SIZ	BASE+0020 (16 bits of 32)		
BIT	15	. . .	0
NAME	Starting Address Register 4		
OPER	R/W		
RESET	0 PS		

This register is the starting address register for the fourth local bus to VMEbus map decoder.

Local Bus Slave (VMEbus Master) Address Translation Address Register 4

ADR/SIZ	BASE+0024 (16 bits of 32)		
BIT	31	. . .	16
NAME	Address Translation Address Register 4		
OPER	R/W		
RESET	0 PS		

This register is the address translation address register for the fourth local bus to VMEbus bus map decoder.

Local Bus Slave (VMEbus Master) Address Translation Select Register 4

ADR/SIZ	BASE+0024 (16 bits of 32)		
BIT	15	. . .	0
NAME	Address Translation Select Register 4		
OPER	R/W		
RESET	0 PS		

This register is the address translation select register for the fourth local bus to VMEbus bus map decoder.

Local Bus Slave (VMEbus Master) Attribute Register 4

ADR/SIZ	BASE+0028 (8 bits of 32)							
BIT	31	30	29	28	27	26	25	24
NAME	D16	WP	AM					
OPER	R/W	R/W	R/W					
RESET	0 PS	0 PS	0 PS					

This register is the attribute register for the fourth local bus to VMEbus bus map decoder.

AM These bits define the VMEbus address modifier codes the VMEbus master uses for the segment defined by map decoder 4. Because the local bus to VMEbus interface does not support block transfers, the block transfer address modifier codes should not be used.

WP When this bit is high, write posting is enabled to the segment defined by map decoder 4. When this bit is low, write posting is disabled to the segment defined by map decoder 4.

D16 When this bit is high, D16 data transfers are performed to the segment defined by map decoder 4. When this bit is low, D32 data transfers are performed to the segment defined by map decoder 4.

Local Bus Slave (VMEbus Master) Attribute Register 3

ADR/SIZ	BASE+0028 (8 bits of 32)							
BIT	23	22	21	20	19	18	17	16
NAME	D16	WP	AM					
OPER	R/W	R/W	R/W					
RESET	0 PS	0 PS	0 PS					

This register is the attribute register for the third local bus to VMEbus bus map decoder.

AM These bits define the VMEbus address modifier codes the VMEbus master uses for the segment defined by map decoder 3. Because the local bus to VMEbus interface does not support block transfers, the block transfer address modifier codes should not be used.

WP When this bit is high, write posting is enabled to the segment defined by map decoder 3. When this bit is low, write posting is disabled to the segment defined by map decoder 3.

D16 When this bit is high, D16 data transfers are performed to the segment defined by map decoder 3. When this bit is low, D32 data transfers are performed to the segment defined by map decoder 3.

Local Bus Slave (VMEbus Master) Attribute Register 2

ADR/SIZ	BASE+0028 (8 bits of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	D16	WP	AM					
OPER	R/W	R/W	R/W					
RESET	0 PS	0 PS	0 PS					

This register is the attribute register for the second local bus to VMEbus bus map decoder.

AM These bits define the VMEbus address modifier codes the VMEbus master uses for the segment defined by map decoder 2. Since the local bus to VMEbus interface does not support block transfers, the block transfer address modifier codes should not be used.

WP When this bit is high, write posting is enabled to the segment defined by map decoder 2. When this bit is low, write posting is disabled to the segment defined by map decoder 2.

D16 When this bit is high, D16 data transfers are performed to the segment defined by map decoder 2. When this bit is low, D32 data transfers are performed to the segment defined by map decoder 2.

Local Bus Slave (VMEbus Master) Attribute Register 1

ADR/SIZ	BASE+0028 (8 bits of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	D16	WP	AM					
OPER	R/W	R/W	R/W					
RESET	0 PS	0 PS	0 PS					

This register is the attribute register for the first local bus to VMEbus bus map decoder.

AM These bits define the VMEbus address modifier codes the VMEbus master uses for the segment defined by map decoder 1. Because the local bus to VMEbus interface does not support block transfers, the block transfer address modifier codes should not be used.

WP When this bit is high, write posting is enabled to the segment defined by map decoder 1. When this bit is low, write posting is disabled to the segment defined by map decoder 1.

D16 When this bit is high, D16 data transfers are performed to the segment defined by map decoder 1. When this bit is low, D32 data transfers are performed to the segment defined by map decoder 1.

VMEbus Slave GCSR Group Address Register

ADR/SIZ	BASE+002C (8 bits of 32)		
BIT	31	. . .	24
NAME	GCSR Group Address Register		
OPER	R/W		
RESET	$00 PS		

This register defines the group address of the GCSR as viewed from the VMEbus. The GCSR address is defined by the group address and the board address. Once enabled, the GCSR register should not be reprogrammed unless the VMEchip2 is VMEbus master.

GCSR Group These bits define the group portion of the GCSR address. These bits are compared with VMEbus address lines A8 through A15. The recommended group address for the MVME1603 and MVME1604 is unknown at this time.

VMEbus Slave GCSR Board Address Register

ADR/SIZ	BASE+002C (4 bits of 32)						
BIT	23	. . .	20				
NAME	GCSR Board Address						
OPER	R/W						
RESET	$F PS						

This register defines the board address of the GCSR as viewed from the VMEbus. The GCSR address is defined by the group address and the board address. Once enabled, the GCSR register should not be reprogrammed unless the VMEchip2 is VMEbus master. The value $F in the GCSR board address register disables the map decoder. The map decoder is enabled when the board address is not $F.

GCSR Board These bits define the board number portion of the GCSR address. These bits are compared with VMEbus address lines A4 through A7. The GCSR is enabled by values $0 through $E. The address

$XXFY in the VMEbus A16 space is reserved for the location monitors LM0 through LM3. Note: XX is the group address and Y is the location monitor (1,LM0; 3,LM1; 5,LM2; 7,LM3).

Local Bus To VMEbus Enable Control Register

ADR/SIZ	BASE+002C (4 bits of 32)							
					19	18	17	16
NAME					EN4	EN3	EN2	EN1
OPER					R/W	R/W	R/W	R/W
RESET					0 PSL	0 PSL	0 PSL	0 PSL

This register is the map decoder enable register for the four programmable local bus to VMEbus map decoders.

EN1 When this bit is high, the first local bus to VMEbus map decoder is enabled. When this bit is low, the first local bus to VMEbus map decoder is disabled.

EN2 When this bit is high, the second local bus to VMEbus map decoder is enabled. When this bit is low, the second local bus to VMEbus map decoder is disabled.

EN3 When this bit is high, the third local bus to VMEbus map decoder is enabled. When this bit is low, the third local bus to VMEbus map decoder is disabled.

EN4 When this bit is high, the fourth local bus to VMEbus map decoder is enabled. When this bit is low, the fourth local bus to VMEbus map decoder is disabled.

Local Bus To VMEbus I/O Control Register

ADR/SIZ	BASE+002C (8 bits of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	I2EN	I2WP	I2SU	I2PD	I1EN	I1D16	I1WP	I1SU
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PSL	0 PS	0 PS	0 PS	0 PSL	0 PS	0 PS	0 PS

This register controls the VMEbus short I/O map and the F page ($F0000000 through $FF7FFFFF) I/O map.

I1SU When this bit is high, the VMEchip2 drives a supervisor address modifier code when the short I/O space is accessed. When this bit is low, the VMEchip2 drives a user address modifier code when the short I/O space is accessed.

I1WP When this bit is high, write posting is enabled to the VMEbus short I/O segment. When this bit is low, write posting is disabled to the VMEbus short I/O segment.

I1D16 When this bit is high, D16 data transfers are performed to the VMEbus short I/O segment. When this bit is low, D32 data transfers are performed to the VMEbus short I/O segment.

I1EN When this bit is high, the VMEbus short I/O map decoder is enabled. When this bit is low, the VMEbus short I/O map decoder is disabled.

I2PD When this bit is high, the VMEchip2 drives a program address modifier code when the F page is accessed. When this bit is low, the VMEchip2 drives a data address modifier code when the F page is accessed.

I2SU When this bit is high, the VMEchip2 drives a supervisor address modifier code when the F page is accessed. When this bit is low, the VMEchip2 drives a user address modifier code when the F page is accessed.

I2WP When this bit is high, write posting is enabled to the local bus F page. When this bit is low, write posting is disabled to the local bus F page.

I2EN When this bit is high, the F page ($F0000000 through $FF7FFFFF) map decoder is enabled. The F0 page is defined as A24/D16 on the VMEbus while the F1-FE pages are defined as A32/D16. When this bit is low, the F page is disabled.

ROM Control Register

ADR/SIZ	BASE+002C (8 bits of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	SIZE		BSPD			ASPD		
OPER	R/W		R/W			R/W		
RESET	0 PS		0 PS			0 PS		

The ROM Control Register is not used.

2

Programming the VMEchip2 DMA Controller

This section includes programming information on the DMA controller, VMEbus interrupter, MPU status register, and local bus to VMEbus requester register.

The VMEchip2 features a local bus - VMEbus DMA controller (DMAC). The DMAC has two modes of operation: command chaining, and direct. In the direct mode, the local bus address, the VMEbus address, the byte count, and the control register of the DMAC are programmed and the DMAC is enabled. The DMAC transfers data, as programmed, until the byte count is zero or an error is detected. When the DMAC stops, the status bits in the DMAC status register are set and an interrupt is sent to the local bus interrupter. If the DMAC interrupt is enabled in the local bus interrupter, the local bus is interrupted. The time on and time off timers should be programmed to control the VMEbus bandwidth used by the DMAC.

A maximum of 4GB of data may be transferred with one DMAC command. Larger transfers can be accomplished using the command chaining mode. In the command chaining mode, a singly-linked list of commands is built in local memory and the table address register in the DMAC is programmed with the starting address of the list of commands. The DMAC control register is programmed and the DMAC is enabled. The DMAC executes commands from the list until all commands are executed or an error is detected. When the DMAC stops, the status bits are set in the DMAC status register and an interrupt is sent to the local bus interrupter. If the DMAC interrupt is enabled in the local bus interrupter, the local bus is interrupted. When the DMAC finishes processing a command in the list, and interrupts are enabled for that command, the DMAC sends an interrupt to the local bus interrupter. If the DMAC interrupt is enabled in the local bus interrupter, the local bus is interrupted.

The DMAC control is divided into two registers. The first register is only accessible by the processor. The second register can be loaded by the processor in the direct mode and by the DMAC in the command chaining mode.

Once the DMAC is enabled, the counter and control registers should not be modified by software. When the command chaining mode is used, the list of commands must be in local 32-bit memory and the entries must be four-byte aligned.

A DMAC command list includes one or more DMAC command packets. A DMAC command packet includes a control word that defines the VMEbus AM code, the VMEbus transfer size, the VMEbus transfer method, the DMA transfer direction, and the VMEbus and local bus address counter operation. The format of the control word is the same as the lower 16 bits of the control register. The command packet also includes a local bus address, a VMEbus address, a byte count, and a pointer to the next command packet in the list. The end of a command is indicated by setting bit 0 or 1 of next command address. The command packet format is shown in Table 2-2.

Table 2-2. DMAC Command Table Format

Entry	Function	
0 (bits 0-15)	--	Control Word
1 (bits 0-31)	Local Bus Address	
2 (bits 0-31)	VMEbus Address	
3 (bits 0-31)	Byte Count	
4 (bits 0-31)	Address of Next Command Packet	

DMAC Registers

This section provides addresses and bit level descriptions of the DMAC counters, control registers, and status registers. Other control functions are also included in this section.

PROM Decoder, SRAM and DMA Control Register

This register is not used.

ADR/SIZ	BASE+0030 (8 bits [6 used] of 32)							
BIT	23	22	21	20	19	18	17	16
NAME			WAIT RMW	ROM0	TBLSC		SRAMS	
OPER			R/W	R/W	R/W		R/W	
RESET			0 PSL	1 PSL	0 PS		0 PS	

Local Bus To VMEbus Requester Control Register

ADR/SIZ	BASE+0030 (8 bits [7 used] of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	ROBN	DHB	DWB		LVFAIR	LVRWD	LVREQL	
OPER	R/W	R	R/W		R/W	R/W	R/W	
RESET	0 PS	0 PS	0 PSL		0 PS	0 PS	0 PS	

This register controls the VMEbus request level, the request mode, and release mode for the local bus to VMEbus interface.

LVREQL These bits define the VMEbus request level. The request is only changed when the VMEchip2 is bus master. The VMEchip2 always requests at the old level until it becomes bus master and the new level takes effect. If the VMEchip2 is bus master when the level is changed, the new level does not take effect until the bus has been released and rerequested at the old level. The requester always requests the VMEbus at level 3 the first time following a SYSRESET.

0 Request level is 0
1 Request level is 1
2 Request level is 2
3 Request level is 3

LVRWD When this bit is high, the requester operates in the release-when-done mode. When this bit is low, the requester operates in the release-on-request mode.

LVFAIR When this bit is high, the requester operates in the fair mode. When this bit is low, the requester does not operate in the fair mode. In the fair mode, the requester waits until the request signal line for the selected level is inactive before requesting the VMEbus.

DWB When this bit is high, the VMEchip2 requests the VMEbus and does not release it. When this bit is low, the VMEchip2 releases the VMEbus according to the release mode programmed in the LVRWD bit. When the VMEbus has been acquired, the DHB bit is set.

DHB When this bit is high, the VMEbus has been acquired in response to the DWB bit being set. When the DWB bit is cleared, this bit is cleared.

ROBN When this bit is high, the VMEbus arbiter operates in the round robin mode. When this bit is low, the arbiter operates in the priority mode.

DMAC Control Register 1 (bits 0-7)

ADR/SIZ	BASE+0030 (8 bits of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	DHALT	DEN	DTBL	DFAIR	DRELM		DRELQ	
OPER	S	S	R/W	R/W	R/W		R/W	
RESET	0 PS	0 PS	0 PS	0 PS	0 PS		0 PS	

This control register is loaded by the processor; it is not modified when the DMAC loads new values from the command packet.

DREQL These bits define the VMEbus request level for the DMAC requester. The request is only changed when the VMEchip2 is bus master. The VMEchip2 always requests at the old level until it becomes bus master and the new level takes effect. If the VMEchip2 is bus master when the level is changed, the new level does not take effect until the bus has been released and rerequested at the old level. The requester always requests the VMEbus at level 3 the first time following a SYSRESET.

0 VMEbus request level 0
1 VMEbus request level 1
2 VMEbus request level 2
3 VMEbus request level 3

DRELM These bits define the VMEbus release mode for the DMAC requester. The DMAC always releases the bus when the FIFO is full (VMEbus to local bus) or empty (local bus to VMEbus).

0 Release when the time on timer has expired and a BRx* signal is active on the VMEbus
1 Release when the time on timer has expired
2 Release when a BRx* signal is active on the VMEbus
3 Release when a BRx* signal is active on the VMEbus or the time on timer has expired

DFAIR When this bit is high, the DMAC requester operates in the fair mode. It waits until its request level is inactive before requesting the VMEbus. When this bit is low, the DMAC requester does not operate in the fair mode.

DTBL The DMAC operates in the direct mode when this bit is low, and it operates in the command chaining mode when this bit is high.

DEN The DMAC is enabled when this bit is set high. This bit always reads 0.

DHALT When this bit is high, the DMAC halts at the end of a command when the DMAC is operating in the command chaining mode. When this bit is low, the DMAC executes the next command in the list.

DMAC Control Register 2 (bits 8-15)

ADR/SIZ	BASE+0034 (8 bits [7 used] of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	INTE	SNP			VINC	LINC	TVME	D16
OPER	R/W	R/W			R/W	R/W	R/W	R/W
RESET	0 PS	0 PS			0 PS	0 PS	0 PS	0 PS

This portion of the control register is loaded by the processor or by the DMAC when it loads the command word from the command packet. Because this register is loaded from the command packet in the command chaining mode, the descriptions here also apply to the control word in the command packet.

D16 When this bit is high, the DMAC executes D16 cycles on the VMEbus. When this bit is low, the DMAC executes D32 cycles on the VMEbus.

TVME This bit defines the direction in which the DMAC transfers data. When this bit is high, data is transferred to the VMEbus. When it is low, data is transferred to the local bus.

LINC When this bit is high, the local bus address counter is incremented during DMA transfers. When this bit is low, the counter is not incremented. This bit should normally be set high. In special situations such as transferring data to or from a FIFO, it may be desirable to not increment the counter.

VINC When this bit is high, the VMEbus address counter is incremented during DMA transfers. When this bit is low, the counter is not incremented. This bit should normally be set high. In special situations such as transferring data to or from a FIFO, it may be desirable to not increment the counter.

SNP These bits are not used.

INTE This bit is used only in the command chaining mode and it is only modified when the DMAC loads the control register from the control word in the command packet. When this bit in the command packet is set, an interrupt is sent to the local bus interrupter when the command in the packet has been executed. The local bus is interrupted if the DMAC interrupt is enabled.

DMAC Control Register 2 (bits 0-7)

ADR/SIZ	BASE+0034 (8 bits of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	BLK		VME AM					
OPER	R/W		R/W					
RESET	0 PS		0 PS					

This portion of the control register is loaded by the processor or the DMAC when it loads the command word from the command packet. Because this byte is loaded from the command packet in the command chaining mode, the descriptions here also apply to the control word in the command packet.

VME AM These bits define the address modifier codes the DMAC drives on the VMEbus when it is bus master. During non-block transfer cycles, bits 0-5 define the VMEbus address modifiers. During block transfers, bits 2-5 define VMEbus address modifier bits 2-5, and address modifier bits 0 and 1 are provided by the DMAC to indicate a block transfer. Block transfer mode should not be set in the address modifier codes. The special block transfer bits should be set to enable block transfers. If non-block cycles are required to reach a 32- or 64-bit boundary, bits 0 and 1 are used during these cycles.

BLK These bits control the block transfer modes of the DMAC:

0 Block transfers disabled

1 The DMAC executes D32 block transfer cycles on the VMEbus. In the block transfer mode, the DMAC may execute byte and two-byte cycles at the beginning and ending of a transfer in non-block transfer mode.

2 Block transfers disabled

3 The DMAC executes D64 block transfer cycles on the VMEbus. In the block transfer mode, the DMAC may execute byte, two-byte and four-byte cycles at the beginning and ending of a transfer in non-block transfer mode.

DMAC Local Bus Address Counter

ADR/SIZ	BASE+0038 (32 bits)		
BIT	31	. . .	0
NAME	DMAC Local Bus Address Counter		
OPER	R/W		
RESET	0 PS		

In the direct mode, this counter is programmed with the starting address of the data in local bus memory.

DMAC VMEbus Address Counter

ADR/SIZ	BASE+003C (32 bits)		
BIT	31	. . .	0
NAME	DMAC VMEbus Address Counter		
OPER	R/W		
RESET	0 PS		

In the direct mode, this counter is programmed with the starting address of the data in VMEbus memory.

DMAC Byte Counter

ADR/SIZ	BASE+0040 (32 bits)		
BIT	31	. . .	0
NAME	DMAC Byte Counter		
OPER	R/W		
RESET	0 PS		

In the direct mode, this counter is programmed with the number of bytes of data to be transferred.

Table Address Counter

ADR/SIZ	BASE+0044 (32 bits)		
BIT	31	. . .	0
NAME	Table Address Counter		
OPER	R/W		
RESET	0 PS		

In the command chaining mode, this counter should be loaded by the processor with the starting address of the list of commands. This register gets reloaded by the DMAC with the starting address of the current command. The last command in a list should have bits 0 and 1 set in the next command pointer.

VMEbus Interrupter Control Register

ADR/SIZ	BASE+BASE+0048 (8 bits [7 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME		IRQ1S		IRQC	IRQS	IRQL		
OPER		R/W		S	R	S		
RESET		0 PS		0 PS	0 PS	0 PS		

This register controls the VMEbus interrupter.

IRQL These bits define the level of the VMEbus interrupt generated by the VMEchip2. A VMEbus interrupt is generated by writing the desired level to these bits. These bits always read 0 and writing 0 to these bits has no effect.

IRQS This bit is the IRQ status bit. When this bit is high, the VMEbus interrupt has not been acknowledged. When this bit is low, the VMEbus interrupt has been acknowledged. This is a read-only status bit.

IRQC This bit is VMEbus interrupt clear bit. When this bit is set high, the VMEbus interrupt is removed. This feature is only used when the IRQ1 broadcast mode is used. Normal VMEbus interrupts should never be cleared. This bit always reads 0 and writing a 0 to this bit has no effect.

IRQ1S These bits control the function of the IRQ1 signal line on the VMEbus:

0 The IRQ1 signal from the interrupter is connected to the IRQ1 signal line on the VMEbus.

1 The output from tick timer 1 is connected to the IRQ1 signal line on the VMEbus.

2 The IRQ1 signal from the interrupter is connected to the IRQ1 signal line on the VMEbus.

3 The output from tick timer 2 is connected to the IRQ1 signal line on the VMEbus.

VMEbus Interrupter Vector Register

ADR/SIZ	BASE+0048 (8 bits of 32)		
BIT	23	. . .	16
NAME	INTERRUPTER VECTOR		
OPER	R/W		
RESET	$0F PS		

This register controls the VMEbus interrupter vector.

MPU Status and DMA Interrupt Count Register

ADR/SIZ	BASE+0048 (8 bits of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	DMAIC				MCLR	MLBE	MLPE	MLOB
OPER	R				C	R	R	R
RESET	0 PS				0 PS	0 PS	0 PS	0 PS

This is the MPU status register and DMAC interrupt counter.

MLOB This bit is not used.

MLPE This bit is not used.

MLBE This bit is not used

MCLR Writing a one to this bit clears the MPU status bits 7, 8, 9 and 10 (MLTO, MLOB, MLPE, and MLBE) in this register.

DMAIC The DMAC interrupt counter is incremented when an interrupt is sent to the local bus interrupter. The value in this counter indicates the number of commands processed when the DMAC is operated in the command chaining mode. If interrupt count exceeds 15, the counter rolls over. This counter operates regardless of whether the DMAC interrupts are enabled. This counter is cleared when the DMAC is enabled.

DMAC Status Register

ADR/SIZ	BASE+0048 (8 bits of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	MLTO	DLBE	DLPE	DLOB	DLTO	TBL	VME	DONE
OPER	R	R	R	R	R	R	R	R
RESET	0 PS	0 PS	0 PS	0 PS	0 PS	0 PS	0 PS	0 PS

This is the DMAC status register.

DONE This bit is set when the DMAC has finished executing commands and there were no errors or the DMAC has finished executing command because the halt bit was set. This bit is cleared when the DMAC is enabled.

VME When this bit is set, the DMAC received a VMEbus BERR during a data transfer. This bit is cleared when the DMAC is enabled.

TBL When this bit is set, the DMAC received an error on the local bus while it was reading commands from the command packet. Additional information is provided in bits 3 - 6 (DLTO, DLOB, DLPE, and DLBE). This bit is cleared when the DMAC is enabled.

DLTO This bit is not used.

DLOB When this bit is set, the DMAC received a TEA and the status indicated offboard. This bit is cleared when the DMAC is enabled.

DLPE When this bit is set, the DMAC received a TEA and the status indicated a parity error during a DRAM data transfer. This bit is cleared when the DMAC is enabled.

DLBE When this bit is set, the DMAC received a TEA and additional status was not provided. This bit is cleared when the DMAC is enabled.

MLTO This bit is notused.

Programming the Tick and Watchdog Timers

The VMEchip2 has two 32-bit tick timers and one watchdog timer. This section provides addresses and bit level descriptions of the prescaler, tick timer, watchdog timer registers and various other timer registers.

VMEbus Arbiter Time-out Control Register

ADR/SIZ	BASE+004C (8 bits [1 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME								ARBTO
OPER								R/W
RESET								0 PS

This register controls the VMEbus arbiter time-out timer.

ARBTO When this bit is high, the VMEbus grant time-out timer is enabled. When this bit is low, the VMEbus grant timer is disabled. When the timer is enabled and the arbiter does not receive a BBSY signal within 256 µs after a grant is issued, the arbiter asserts BBSY and removes the grant. The arbiter then re-arbitrates any pending requests.

DMAC Ton/Toff Timers and VMEbus Global Time-out Control Register

ADR/SIZ	BASE+004C (8 bits of 32)							
BIT	23	22	21	20	19	18	17	16
NAME	TIME OFF			TIME ON			VGTO	
OPER	R/W			R/W			R/W	
RESET	0 PS			0 PS			0 PS	

This register controls the DMAC time off timer, the DMAC time on timer, and the VMEbus global time-out timer.

VGTO These bits define the VMEbus global time-out value. When DS0 or DS1 is asserted on the VMEbus, the timer begins timing. If the timer times out before the data strobes are removed, a BERR signal is sent to the VMEbus. The global time-out timer is disabled when the VMEchip2 is not system controller.

0	8 µs
1	64 µs
2	256 µs
3	The timer is disabled

TIME ON These bits define the maximum time the DMAC spends on the VMEbus:

0	16 µs	4	256 µs
1	32 µs	5	512 µs
2	64 µs	6	1024 µs
3	128 µs	7	When done (or no data)

TIME OFF These bits define the minimum time the DMAC spends off the VMEbus:

0	0 µs	4	128 µs
1	16 µs	5	256 µs
2	32 µs	6	512 µs
3	64 µs	7	1024 µs

VME Access, Local Bus and Watchdog Time-out Control Register

ADR/SIZ	BASE+004C (8 bits of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	VATO		LBTO		WDTO			
OPER	R/W		R/W		R/W			
RESET	0 PS		0 PS		0 PS			

WDTO These bits define the watchdog time-out period:

0	512 µs	4	8 ms
1	1 ms	5	16 ms
2	2 ms	6	32 ms
3	4 ms	7	64 ms
8	128 ms	12	4 s
9	256 ms	13	16 s
10	512 ms	14	32 s
11	1 s	15	64 s

LBTO These bits are not used.

VATO These bits define the VMEbus access time-out value. When a transaction is headed to the VMEbus and the VMEchip2 is not the current VMEbus master, the access timer begins timing. If the VMEchip2 has not received bus mastership before the timer times out and the transaction is not write posted, a TEA signal is sent to the local bus. If the transaction is write posted, a write post error interrupt is sent to the local bus interrupter.

0 64 µs
1 1 ms
2 32 ms
3 The timer is disabled

Prescaler Control Register

ADR/SIZ	BASE+004C (8 bits of 32)		
BIT	7	. . .	0
NAME	Prescaler Adjust		
OPER	R/W		
RESET	$DF P		

The prescaler provides the various clocks required by the counters and timers in the VMEchip2. In order to specify absolute times from these counters and timers, the prescaler must be adjusted for different PCI bus clocks. The prescaler register should be programmed based on the following equation. This provides a 1MHz clock to the tick timers.

$$\textit{prescaler register} = 256 - \textit{PCI clock}\ (\text{MHz})$$

For example, for operation at 20 MHz the prescaler value is $EC, at 25 MHz it is $E7, and at 33 MHz it is $DF.

Non-integer local bus clocks introduce an error into the specified times for the various counters and timers. This is most notable in the tick timers. The tick timer clock can be derived by the following equation.

tick timer clock = *PCI clock* / (256 - *prescaler value*)

If the prescaler is not correctly programmed, the bus timers do not generate their specified values and the VMEbus reset time may be violated. The maximum clock frequency for the tick timers is the B clock divided by two. The prescaler register control logic does not allow the value 255 ($FF) to be programmed.

Tick Timer 1 Compare Register

ADR/SIZ	BASE+0050 (32 bits)		
BIT	31	. . .	0
NAME	Tick timer 1 Compare Register		
OPER	R/W		
RESET	0 P		

The tick timer 1 counter is compared to this register. When they are equal, an interrupt is sent to the local bus interrupter and the overflow counter is incremented. If the clear-on-compare mode is enabled, the counter is also cleared. For periodic interrupts, the following equation should be used to calculate the compare register value for a specific period (T).

compare register value = T (μs)

When programming the tick timer for periodic interrupts, the counter should be cleared to zero by software and then enabled. If the counter does not initially start at zero, the time to the first interrupt may be longer or shorter than expected. Remember the rollover time for the counter is 71.6 minutes.

Tick Timer 1 Counter

ADR/SIZ	BASE+0054 (32 bits)		
BIT	31	. . .	0
NAME	Tick timer 1 Counter		
OPER	R/W		
RESET	0 P		

This is the tick timer 1 counter. When enabled, it increments every microsecond. Software may read or write the counter at any time.

Tick Timer 2 Compare Register

ADR/SIZ	BASE+0058 (32 bits)		
BIT	31	. . .	0
NAME	Tick timer 2 Compare Register		
OPER	R/W		
RESET	0 P		

The tick timer 2 counter is compared to this register. When they are equal, an interrupt is sent to the local bus interrupter and the overflow counter is incremented. If the clear-on-compare mode is enabled, the counter is also cleared. For periodic interrupts, the following equation should be used to determine the compare register value for a specific period.

$$\textit{compare register value} = \mathrm{T}\ (\mu s)$$

When programming the tick timer for periodic interrupts, the counter should be cleared to zero by software and then enabled. If the counter does not initially start at zero, the time to the first interrupt may be longer or shorter than expected. Remember the rollover time for the counter is 71.6 minutes.

Tick Timer 2 Counter

ADR/SIZ	BASE+005C (32 bits)		
BIT	31	. . .	0
NAME	Tick timer 2 Counter		
OPER	R/W		
RESET	0 P		

This is the tick timer 2 counter. When enabled, it increments every microsecond. Software may read or write the counter at any time.

Board Control Register

ADR/SIZ	BASE+0060 (8 bits [7 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME		SCON	SFFL	BRFLI	PURS	CPURS	BDFLO	RSWE
OPER		R	R	R	R	C	R/W	R/W
RESET		X	X	1 PSL	1 P	0 PS	1 PSL	1 P

RSWE When this bit is high, the RESET switch will cause a SYSRESET. When this bit is low, the RESET will not cause a SYSRESET.

BDFLO When this bit is high, the VMEchip2 asserts the BRDFAIL signal pin. When this bit is low, this bit does not contribute to the BRDFAIL signal on the VMEchip2. NOTE: The BRDFAIL pin is not connected to the FAIL LED.

CPURS When this bit is set high, the power-up reset status bit is cleared. This bit is always read zero.

PURS This bit is set by a power-up reset. It is cleared by a write to the CPURS bit.

BRFLI When this status bit is high, the BRDFAIL signal pin on the VMEchip2 is asserted. When this status bit is low, the BRDFAIL signal pin on the VMEchip2 is not asserted. The BRDFAIL pin may be asserted by the BDFLO bit in this register, or a watchdog time-out.

SFFL When this status bit is high, the SYSFAIL signal line on the VMEbus is asserted. When this status bit is low, the SYSFAIL signal line on the VMEbus is not asserted.

SCON When this status bit is high, the VMEchip2 is configured as system controller. When this status bit is low, the VMEchip2 is not configured as system controller.

Watchdog Timer Control Register

ADR/SIZ	BASE+0060 (8 bits of 32)							
BIT	23	22	21	20	19	18	17	16
NAME	SRST	WDCS	WDCC	WDTO	WDBFE	WDS/L	WDRSE	WDEN
OPER	S	C	C	R	R/W	R/W	R/W	R/W
RESET	0 PS	0	0	0 P	0 PSL	0 PSL	0 PSL	0 PSL

WDEN When this bit is high, the watchdog timer is enabled. When this bit is low, the watchdog timer is not enabled.

WDRSE When this bit is high, and a watchdog time-out occurs, a SYSRESET or LRESET is generated. The WDS/L bit in this register selects the reset. When this bit is low, a watchdog time-out does not cause a reset.

WDS/L When this bit is high and the watchdog timer has timed out and the watchdog reset enable (WDRSE bit in this register) is high, a SYSRESET signal is generated on the VMEbus which in turn causes LRESET to be asserted. When this bit is low and the watchdog timer has timed out and the watchdog reset enable (WDRSE bit in this register) is high, an LRESET signal is generated on the local bus.

WDBFE When this bit is high and the watchdog timer has timed out, the VMEchip2 asserts the BRDFAIL signal pin. When this bit is low, the watchdog timer does not contribute to the BRDFAIL signal on the VMEchip2.

WDTO When this status bit is high, a watchdog time-out has occurred. When this status bit is low, a watchdog time-out has not occurred. This bit is cleared by writing a one to the WDCS bit in this register.

WDCC — When this bit is set high, the watchdog counter is reset. The counter must be reset within the time-out period or a watchdog time-out occurs.

WDCS — When this bit is set high, the watchdog time-out status bit (WDTO bit in this register) is cleared.

SRST — When this bit is set high, a SYSRESET signal is generated on the VMEbus. SYSRESET resets the VMEchip2 and clears this bit.

Tick Timer 2 Control Register

ADR/SIZ	BASE+0060 (8 bits [7 used] of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	OVF					COVF	COC	EN
OPER	R					C	R/W	R/W
RESET	0 PS					0 PS	0 PS	0 PS

EN — When this bit is high, the counter increments. When this bit is low, the counter does not increment.

COC — When this bit is high, the counter is reset to zero when it compares with the compare register. When this bit is low, the counter is not reset.

COVF — The overflow counter is cleared when a one is written to this bit.

OVF — These bits are the output of the overflow counter. The overflow counter is incremented each time the tick timer sends an interrupt to the local bus interrupter. The overflow counter can be cleared by writing a one to the COVF bit.

2

Tick Timer 1 Control Register

ADR/SIZ	BASE+0060 (8 bits [7 used] of 32)							
BIT	7	6	5	4	31	2	1	0
NAME	OVF					COVF	COC	EN
OPER	R					C	R/W	R/W
RESET	0 PS					0 PS	0 PS	0 PS

EN When this bit is high, the counter increments. When this bit is low, the counter does not increment.

COC When this bit is high, the counter is reset to zero when it compares with the compare register. When this bit is low, the counter is not reset.

COVF The overflow counter is cleared when a one is written to this bit.

OVF These bits are the output of the overflow counter. The overflow counter is incremented each time the tick timer sends an interrupt to the local bus interrupter. The overflow counter can be cleared by writing a one to the COVF bit.

Prescaler Counter

ADR/SIZ	BASE+0064 (32 bits)		
BIT	31	. . .	0
NAME	Prescaler Counter		
OPER	R/W		
RESET	0 P		

The VMEchip2 has a 32-bit prescaler that provides the clocks required by the various timers in the chip. Access to the prescaler is provided for test purposes. The counter is described here because it may be useful in other applications. The lower 8 bits of the prescaler counter increment to $FF at the local bus clock rate and then they are loaded from the prescaler adjust register. When the load occurs, the upper 24 bits are incremented. When the prescaler adjust

register is correctly programmed, the lower 8 bits increment at the local bus clock rate and the upper 24 bits increment every microsecond. The counter may be read at any time.

Programming the Local Bus Interrupter

The local bus interrupter is used by devices that wish to interrupt the local bus. There are 31 devices that can interrupt the local bus through the VMEchip2. In the general case, each interrupter has a level select register, an enable bit, a status bit, a clear bit, and for the software interrupts, a set bit. Each interrupter also provides a unique interrupt vector to the processor. The upper four bits of the vector are programmable in the vector base registers. The lower four bits are unique for each interrupter. There are two base registers, one for the first 16 interrupters, and one for the next 8 interrupters. The VMEbus interrupters provide their own vectors. A summary of the interrupts is shown in Table 2-3.

The status bit of an interrupter is affected by the enable bit. If the enable bit is low, the status bit is also low. Interrupts may be polled by setting the enable bit and programming the level to zero. This enables the status bit and prevents the local bus from being interrupted. The enable bit does not clear edge-sensitive interrupts. If necessary, edge-sensitive interrupts should be cleared, in order to remove any old interrupts, and then enabled. The master interrupt enable (MIEN) bit must be set before the VMEchip2 can generate any interrupts. The MIEN bit is in the I/O Control Register 1.

Table 2-3. Local Bus Interrupter Summary

Interrupt	Vector	Priority for Simultaneous Interrupts
VMEbus IRQ1	External	Lowest
VMEbus IRQ2	External	
VMEbus IRQ3	External	
VMEbus IRQ4	External	
VMEbus IRQ5	External	
VMEbus IRQ6	External	
VMEbus IRQ7	External	
Spare	$Y7	
Software 0	$Y8	
Software 1	$Y9	
Software 2	$YA	
Software 3	$YB	
Software 4	$YC	
Software 5	$YD	
Software 6	$YE	
Software 7	$YF	

Table 2-3. Local Bus Interrupter Summary (Continued)

Interrupt	Vector	Priority for Simultaneous Interrupts
GCSR LM0	$X0	
GCSR LM1	$X1	
GCSR SIG0	$X2	
GCSR SIG1	$X3	
GCSR SIG2	$X4	
GCSR SIG3	$X5	
DMAC	$X6	
VMEbus Interrupter Acknowledge	$X7	
Tick Timer 1	$X8	
Tick Timer 2	$X9	
VMEbus IRQ1 Edge-Sensitive	$XA	
External Input (parity error)	$XB	
VMEbus Master Write Post Error	$XC	
VMEbus SYSFAIL	$XD	
Abort Switch	$XE	
VMEbus ACFAIL	$XF	Highest

NOTES: X = The contents of vector base register 0.

Y = The contents of vector base register 1.

Refer to the Vector Base Register description later in this chapter for recommended Vector Base Register values.

Local Bus Interrupter Status Register (bits 24-31)

ADR/SIZ	BASE+0068 (8 bits of 32)							
BIT	31	30	29	28	27	26	25	24
NAME	ACF	AB	SYSF	MWP	PE	VI1E	TIC2	TIC1
OPER	R	R	R	R	R	R	R	R
RESET	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL

This register is the local bus interrupter status register. When an interrupt status bit is high, a local bus interrupt is being generated. When an interrupt status bit is low, a local interrupt is not being generated. The interrupt status bits are:

TIC1 Tick timer 1 interrupt

TIC2 Tick timer 2 interrupt

VI1E VMEbus IRQ1 edge-sensitive interrupt

PE This interrupt is not used

MWP VMEbus master write post error interrupt

SYSF VMEbus SYSFAIL interrupt

AB This interrupt is not used

ACF VMEbus ACFAIL interrupt

Local Bus Interrupter Status Register (bits 16-23)

ADR/SIZ	BASE+0068 (8 bits of 32)							
BIT	23	22	21	20	19	18	17	16
NAME	VIA	DMA	SIG3	SIG2	SIG1	SIG0	LM1	LM0
OPER	R	R	R	R	R	R	R	R
RESET	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL

This register is the local bus interrupter status register. When an interrupt status bit is high, a local bus interrupt is being generated. When an interrupt status bit is low, a local interrupt is not being generated. The interrupt status bits are:

LM0 GCSR LM0 interrupt

LM1 GCSR LM1 interrupt

SIG0 GCSR SIG0 interrupt

SIG1 GCSR SIG1 interrupt

SIG2 GCSR SIG2 interrupt

SIG3 GCSR SIG3 interrupt

DMA DMAC interrupt

VIA VMEbus interrupter acknowledge interrupt

Local Bus Interrupter Status Register (bits 8-15)

ADR/SIZ	BASE+0068 (8 bits of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	SW7	SW6	SW5	SW4	SW3	SW2	SW1	SW0
OPER	R	R	R	R	R	R	R	R
RESET	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL

This register is the local bus interrupter status register. When an interrupt status bit is high, a local bus interrupt is being generated. When an interrupt status bit is low, a local interrupt is not being generated. The interrupt status bits are:

SW0 Software 0 interrupt

SW1 Software 1 interrupt

SW2 Software 2 interrupt

SW3 Software 3 interrupt

SW4 Software 4 interrupt

SW5 Software 5 interrupt

SW6 Software 6 interrupt

SW7 Software 7 interrupt

Local Bus Interrupter Status Register (bits 0-7)

ADR/SIZ	BASE+0068 (8 bits of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	SPARE	VME7	VME6	VME5	VME4	VME3	VME2	VME1
OPER	R	R	R	R	R	R	R	R
RESET	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL

This register is the local bus interrupter status register. When an interrupt status bit is high, a local bus interrupt is being generated. When an interrupt status bit is low, a local interrupt is not being generated. The interrupt status bits are:

VME1 VMEbus IRQ1 Interrupt

VME2 VMEbus IRQ2 Interrupt

VME3 VMEbus IRQ3 Interrupt

VME4 VMEbus IRQ4 Interrupt

VME5 VMEbus IRQ5 Interrupt

VME6 VMEbus IRQ6 Interrupt

VME7 VMEbus IRQ7 Interrupt

SPARE This bit is not used

Local Bus Interrupter Enable Register (bits 24-31)

ADR/SIZ	BASE+006C (8 bits of 32)							
BIT	31	30	29	28	27	26	25	24
NAME	EACF	EAB	ESYSF	EMWP	EPE	EVI1E	ETIC2	ETIC1
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL

This register is the local bus interrupter enable register. When an enable bit is high, the corresponding interrupt is enabled. When an enable bit is low, the corresponding interrupt is disabled. The

enable bit does not clear edge-sensitive interrupts or prevent the flip flop from being set. If necessary, edge-sensitive interrupters should be cleared to remove any old interrupts and then enabled.

ETIC1 Enable tick timer 1 interrupt

ETIC2 Enable tick timer 2 interrupt

EVI1E Enable VMEbus IRQ1 edge-sensitive interrupt

EPE This interrupt is not used

EMWP Enable VMEbus master write post error interrupt

ESYSF Enable VMEbus SYSFAIL interrupt

EAB This interrupt is not used

EACF Enable VMEbus ACFAIL interrupt

Local Bus Interrupter Enable Register (bits 16-23)

ADR/SIZ	BASE+006C (8-bits)							
BIT	23	22	21	20	19	18	17	16
NAME	EVIA	EDMA	ESIG3	ESIG2	ESIG1	ESIG0	ELM1	ELM0
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL

This register is the local bus interrupter enable register. When an enable bit is high, the corresponding interrupt is enabled. When an enable bit is low, the corresponding interrupt is disabled. The enable bit does not clear edge-sensitive interrupts or prevent the flip flop from being set. If necessary, edge-sensitive interrupters should be cleared to remove any old interrupts and then enabled.

ELM0 Enable GCSR LM0 interrupt

ELM1 Enable GCSR LM1 interrupt

ESIG0 Enable GCSR SIG0 interrupt

ESIG1 Enable GCSR SIG1 interrupt

ESIG2 Enable GCSR SIG2 interrupt

ESIG3 Enable GCSR SIG3 interrupt

EDMA Enable DMAC interrupt

EVIA VMEbus interrupter acknowledge interrupt

Local Bus Interrupter Enable Register (bits 8-15

ADR/SIZ	BASE+006C (8 bits of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	ESW7	ESW6	ESW5	ESW4	ESW3	ESW2	ESW1	ESW0
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL

This is the local bus interrupter enable register. When an enable bit is high, the corresponding interrupt is enabled. When an enable bit is low, the corresponding interrupt is disabled. The enable bit does not clear edge-sensitive interrupts or prevent the flip flop from being set. If necessary, edge-sensitive interrupters should be cleared to remove any old interrupts and then enabled.

ESW0 Enable software 0 interrupt

ESW1 Enable software 1 interrupt

ESW2 Enable software 2 interrupt

ESW3 Enable software 3 interrupt

ESW4 Enable software 4 interrupt

ESW5 Enable software 5 interrupt

ESW6 Enable software 6 interrupt

ESW7 Enable software 7 interrupt

Local Bus Interrupter Enable Register (bits 0-7)

ADR/SIZ	BASE+006C (8 bits of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	SPARE	EIRQ7	EIRQ6	EIRQ5	EIRQ4	EIRQ3	EIRQ2	EIRQ1
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL

This is the local bus interrupter enable register. When an enable bit is high, the corresponding interrupt is enabled. When an enable bit is low, the corresponding interrupt is disabled. The enable bit does not clear edge-sensitive interrupts or prevent the flip flop from being set. If necessary, edge-sensitive interrupters should be cleared to remove any old interrupts and then enabled.

EIRQ1 Enable VMEbus IRQ1 interrupt

EIRQ2 Enable VMEbus IRQ2 interrupt

EIRQ3 Enable VMEbus IRQ3 interrupt

EIRQ4 Enable VMEbus IRQ4 interrupt

EIRQ5 Enable VMEbus IRQ5 interrupt

EIRQ6 Enable VMEbus IRQ6 interrupt

EIRQ7 Enable VMEbus IRQ7 interrupt

SPARE SPARE

Software Interrupt Set Register (bits 8-15)

ADR/SIZ	BASE+0070 (8 bits of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	SSW7	SSW6	SSW5	SSW4	SSW3	SSW2	SSW17	SSW07
OPER	S	S	S	S	S	S	S	S
RESET	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL

This register is used to set the software interrupts. An interrupt is set by writing a one to it. The software interrupt set bits are:

SSW0 Set software 0 interrupt

SSW1 Set software 1 interrupt

SSW2 Set software 2 interrupt

SSW3 Set software 3 interrupt

SSW4 Set software 4 interrupt

SSW5 Set software 5 interrupt

SSW6 Set software 6 interrupt

SSW7 Set software 7 interrupt

Interrupt Clear Register (bits 24-31)

ADR/SIZ	BASE+0074 (8 bits of 32)							
BIT	31	30	29	28	27	26	25	24
NAME	CACF	CAB	CSYSF	CMWP	CPE	CVI1E	CTIC2	CTIC1
OPER	C	C	C	C	C	C	C	C
RESET	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL	0 PSL

This register is used to clear the edge-sensitive interrupts. An interrupt is cleared by writing a one to its clear bit. The clear bits are defined below.

CTIC1 Clear tick timer 1 interrupt

CTIC2 Clear tick timer 2 interrupt

CVI1E Clear VMEbus IRQ1 edge-sensitive interrupt

CPE This interrupt is not used

CMWP Clear VMEbus master write post error interrupt

CSYSF Clear VMEbus SYSFAIL interrupt

CAB This interrupt is not used

CACF Clear VMEbus ACFAIL interrupt

Interrupt Clear Register (bits 16-23)

ADR/SIZ	BASE+0074 (8 bits of 32)							
BIT	23	22	21	20	19	18	17	16
NAME	CVIA	CDMA	CSIG3	CSIG2	CSIG1	CSIG0	CLM1	CLM0
OPER	C	C	C	C	C	C	C	C
RESET	X	X	X	X	X	X	X	X

This register is used to clear the edge sensitive-interrupts. An interrupt is cleared by writing a one to its clear bit. The clear bits are defined below.

CLM0 Clear GCSR LM0 interrupt

CLM1 Clear GCSR LM1 interrupt

CSIG0 Clear GCSR SIG0 interrupt

CSIG1 Clear GCSR SIG1 interrupt

CSIG2 Clear GCSR SIG2 interrupt

CSIG3 Clear GCSR SIG3 interrupt

CDMA Clear DMA controller interrupt

CVIA Clear VMEbus interrupter acknowledge interrupt

Interrupt Clear Register (bits 8-15)

ADR/SIZ	BASE+0074 (8 bits of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	CSW7	CSW6	CSW57	CSW4	CSW3	CSW2	CSW1	CSW0
OPER	C	C	C	C	C	C	C	C
RESET	X	X	X	X	X	X	X	X

This register is used to clear the edge software interrupts. An interrupt is cleared by writing a one to its clear bit. The clear bits are:

CSW0 Clear software 0 interrupt

CSW1 Clear software 1 interrupt

CSW2 Clear software 2 interrupt

CSW3 Clear software 3 interrupt

CSW4 Clear software 4 interrupt

CSW5 Clear software 5 interrupt

CSW6 Clear software 6 interrupt

CSW7 Clear software 7 interrupt

Interrupt Level Register 1 (bits 24-31)

ADR/SIZ	BASE+0078 (8 bits [6 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME		ACF LEVEL				AB LEVEL		
OPER		R/W				R/W		
RESET		0 PSL				0 PSL		

This register is used to define the level of the abort interrupt and the ACFAIL interrupt.

AB LEVEL This bit is not used.

ACF LEVEL These bits define the level of the ACFAIL interrupt.

Interrupt Level Register 1 (bits 16-23)

ADR/SIZ	BASE+0078 (8 bits [6 used] of 32)							
BIT	23	22	21	20	19	18	17	16
NAME		SYSF LEVEL				WPE LEVEL		
OPER		R/W				R/W		
RESET		0 PSL				0 PSL		

This register is used to define the level of the SYSFAIL interrupt and the master write post bus error interrupt.

WPE LEVEL These bits define the level of the master write post bus error interrupt.

SYSF LEVEL These bits define the level of the SYSFAIL interrupt.

Interrupt Level Register 1 (bits 8-15)

ADR/SIZ	BASE+0078 (8 bits [6 used] of 32)							
BIT	15	14	13	12	11	10	9	8
NAME		PE LEVEL				IRQ1E LEVEL		
OPER		R/W				R/W		
RESET		0 PSL				0 PSL		

This register is used to define the level of the VMEbus IRQ1 edge-sensitive interrupt and the level of the external (parity error) interrupt.

IRQ1E LEVEL These bits define the level of the VMEbus IRQ1 edge-sensitive interrupt.

PE LEVEL This interrupt is not used.

Interrupt Level Register 1 (bits 0-7)

ADR/SIZ	BASE+0078 (8 bits [6 used] of 32)							
BIT	7	6	5	4	3	2	1	0
NAME		TICK2 LEVEL				TICK1 LEVEL		
OPER		R/W				R/W		
RESET		0 PSL				o PSL		

This register is used to define the level of the tick timer 1 interrupt and the tick timer 2 interrupt.

TICK1 LEVEL These bits define the level of the tick timer 1 interrupt.

TICK2 LEVEL These bits define the level of the tick timer 2 interrupt.

Interrupt Level Register 2 (bits 24-31)

ADR/SIZ	BASE+007C (8 bits [6 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME		VIA LEVEL				DMA LEVEL		
OPER		R/W				R/W		
RESET		0 PSL				0 PSL		

This register is used to define the level of the DMA controller interrupt and the VMEbus acknowledge interrupt.

DMA LEVEL These bits define the level of the DMA controller interrupt.

VIA LEVEL These bits define the level of the VMEbus interrupter acknowledge interrupt.

Interrupt Level Register 2 (bits 16-23)

ADR/SIZ	BASE+007C (8 bits [6 used] of 32)							
BIT	23	22	21	20	19	18	17	16
NAME		SIG3 LEVEL				SIG2 LEVEL		
OPER		R/W				R/W		
RESET		0 PSL				0 PSL		

This register is used to define the level of the GCSR SIG2 interrupt and the GCSR SIG3 interrupt.

SIG2 LEVEL These bits define the level of the GCSR SIG2 interrupt.

SIG3 LEVEL These bits define the level of the GCSR SIG3 interrupt.

Interrupt Level Register 2 (bits 8-15)

ADR/SIZ	BASE+007C (8 bits [6 used] of 32)							
BIT	15	14	13	12	11	10	9	8
NAME		SIG1 LEVEL				SIG0 LEVEL		
OPER		R/W				R/W		
RESET		0 PSL				0 PSL		

This register is used to define the level of the GCSR SIG0 interrupt and the GCSR SIG1 interrupt.

SIG0 LEVEL These bits define the level of the GCSR SIG0 interrupt.

SIG1 LEVEL These bits define the level of the GCSR SIG1 interrupt.

Interrupt Level Register 2 (bits 0-7)

ADR/SIZ	BASE+007C (8 bits [6 used] of 32)							
BIT	7	6	5	4	3	2	1	0
NAME		LM1 LEVEL				LM0 LEVEL		
OPER		R/W				R/W		
RESET		0 PSL				0 PSL		

This register is used to define the level of the GCSR LM0 interrupt and the GCSR LM1 interrupt.

LM0 LEVEL These bits define the level of the GCSR LM0 interrupt.

LM1 LEVEL These bits define the level of the GCSR LM1 interrupt.

Interrupt Level Register 3 (bits 24-31)

ADR/SIZ	BASE+0080 (8 bits [6 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME		SW7 LEVEL				SW6 LEVEL		
OPER		R/W				R/W		
RESET		0 PSL				0 PSL		

This register is used to define the level of the software 6 interrupt and the software 7 interrupt.

SW6 LEVEL These bits define the level of the software 6 interrupt.

SW7 LEVEL These bits define the level of the software 7 interrupt.

Interrupt Level Register 3 (bits 16-23)

ADR/SIZ	BASE+0080 (8 bits [6 used] of 32)							
BIT	23	22	21	20	19	18	17	16
NAME		SW5 LEVEL				SW4 LEVEL		
OPER		R/W				R/W		
RESET		0 PSL				0 PSL		

This register is used to define the level of the software 4 interrupt and the software 5 interrupt.

SW4 LEVEL These bits define the level of the software 4 interrupt.

SW5 LEVEL These bits define the level of the software 5 interrupt.

Interrupt Level Register 3 (bits 8-15)

ADR/SIZ	BASE+0080 (8 bits [6 used] of 32)							
BIT	15	14	13	12	11	10	9	8
NAME		SW3 LEVEL				SW2 LEVEL		
OPER		R/W				R/W		
RESET		0 PSL				0		

This register is used to define the level of the software 2 interrupt and the software 3 interrupt.

SW2 LEVEL These bits define the level of the software 2 interrupt.

SW3 LEVEL These bits define the level of the software 3 interrupt.

Interrupt Level Register 3 (bits 0-7)

ADR/SIZ	BASE+0080 (8 bits [6 used] of 32)							
BIT	7	6	5	4	3	2	1	0
NAME		SW1 LEVEL				SW0 LEVEL		
OPER		R/W				R/W		
RESET		0 PSL				0 PSL		

This register is used to define the level of the software 0 interrupt and the software 1 interrupt.

SW0 LEVEL These bits define the level of the software 0 interrupt.

SW1 LEVEL These bits define the level of the software 1 interrupt.

Interrupt Level Register 4 (bits 24-31)

ADR/SIZ	BASE+0084 (8 bits [6 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME		SPARE LEVEL				VIRQ7 LEVEL		
OPER		R/W				R/W		
RESET		0 PSL				0 PSL		

This register is used to define the level of the VMEbus IRQ7 interrupt and the spare interrupt. The VMEbus level 7 (IRQ7) interrupt may be mapped to any local bus interrupt level.

VIRQ7 LEVEL These bits define the level of the VMEbus IRQ7 interrupt.

SPARE LEVEL These bits define the level of the spare interrupt.

Interrupt Level Register 4 (bits 16-23)

ADR/SIZ	BASE+0084 (8 bits [6 used] of 32)							
BIT	23	22	21	20	19	18	17	16
NAME		VIRQ6 LEVEL				VIRQ5 LEVEL		
OPER		R/W				R/W		
RESET		0 PSL				0 PSL		

This register is used to define the level of the VMEbus IRQ5 interrupt and the VMEbus IRQ6 interrupt. The VMEbus level 5 (IRQ5) interrupt and the VMEbus level 6 (IRQ6) interrupt may be mapped to any local bus interrupt level.

VIRQ5 LEVEL These bits define the level of the VMEbus IRQ5 interrupt.

VIRQ6 LEVEL These bits define the level of the VMEbus IRQ6 interrupt.

Interrupt Level Register 4 (bits 8-15)

ADR/SIZ	BASE+0084 (8 bits [6 used] of 32)							
BIT	15	14	13	12	11	10	9	8
NAME		VIRQ4 LEVEL				VIRQ3 LEVEL		
OPER		R/W				R/W		
RESET		0 PSL				0 PSL		

This register is used to define the level of the VMEbus IRQ3 interrupt and the VMEbus IRQ4 interrupt. The VMEbus level 3 (IRQ3) interrupt and the VMEbus level 4 (IRQ4) interrupt may be mapped to any local bus interrupt level.

VIRQ3 LEVEL These bits define the level of the VMEbus IRQ3 interrupt.

VIRQ4 LEVEL These bits define the level of the VMEbus IRQ4 interrupt.

Interrupt Level Register 4 (bits 0-7)

ADR/SIZ	BASE+0084 (8 bits [6 used] of 32)							
BIT	7	6	5	4	3	2	1	0
NAME		VIRQ2 LEVEL				VIRQ1 LEVEL		
OPER		R/W				R/W		
RESET		0 PSL				0 PSL		

This register is used to define the level of the VMEbus IRQ1 interrupt and the VMEbus IRQ2 interrupt. The VMEbus level 1 (IRQ1) interrupt and the VMEbus level 2 (IRQ2) interrupt may be mapped to any local bus interrupt level.

VIRQ1 LEVEL These bits define the level of the VMEbus IRQ1 interrupt.

VIRQ2 LEVEL These bits define the level of the VMEbus IRQ2 interrupt.

Vector Base Register

ADR/SIZ	BASE+0088 (8 bits of 32)							
BIT	31	30	29	28	27	26	25	24
NAME	VBR 0				VBR 1			
OPER	R/W				R/W			
RESET	0 PSL				0 PSL			

This register is used to define the interrupt base vectors.

VBR 1 These bits define the interrupt base vector 1.

VBR 0 These bits define the interrupt base vector 0.

NOTE: Refer to Table 2-3, Local Bus Interrupter Summary, earlier in this chapter, for further information.
A suggested setting for the Vector Base Register is not known at this time.

I/O Control Register 1

ADR/SIZ	BASE+0088 (8 bits of 32)							
BIT	23	22	21	20	19	18	17	16
NAME	MIEN	SYSFL	ACFL	ABRTL	GPOEN3	GPOEN2	GPOEN1	GPOEN0
OPER	R/W	R	R	R	R/W	R/W	R/W	R/W
RESET	0 PSL	X	X	X	0 PS	0 PS	0 PS	0 PS

This register is a general purpose I/O control register.

Bits 16-19 control the direction of the fourGeneral Purpose I/O pins (GPIO0-3). The general purpose I/O pins are not used.

GPOEN1 This bit is not used.

GPOEN2 This bit is not used.

GPOEN3 This bit is not used.

ABRTL This bit is not used.

ACFL This bit indicates the status of the ACFAIL signal line on the VMEbus. When this bit is high, the ACFAIL signal line is active. When this bit is low, the ACFAIL signal line is not active.

SYSFL This bit indicates the status of the SYSFAIL signal line on the VMEbus. When this bit is high, the SYSFAIL signal line is active. When this bit is low, the SYSFAIL signal line is not active.

MIEN When this bit is low, all interrupts controlled by the VMEchip2 are masked. When this bit is high, all interrupts controlled by the VMEchip2 are not masked.

I/O Control Register 2

ADR/SIZ	BASE+0088 (8 bits of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	GPIO O3	GPIO O2	GPIO O1	GPIO O0	GPIOI 3	GPIOI 2	GPIOI 1	GPIOI 0
OPER	R/W	R/W	R/W	R/W	R	R	R	R
RESET	0 PSL	0 PS	0 PS	0 PS	X	X	X	X

This register is a general purpose I/O control register.

Bits 8-11 reflect the status of the four General Purpose I/O pins (GPIO0-3).

The GPIO pins are not used.

GPIOI0 This bit is not used.

GPIOI1 This bit is not used.

GPIOI2 This bit is not used.

GPIOI3 This bit is not used.

Bits 12-15 determine the driven level of the four General Purpose I/O pins (GPIO0-3) when they are defined as outputs.

GPIOO0 This bit is not used.

GPIOO1 This bit is not used.

GPIOO2 This bit is not used.

GPIOO3 This bit is not used.

I/O Control Register 3

ADR/SIZ	BASE+0088 (8 bits of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	GPI7	GPI6	GPI5	GPI4	GPI3	GPI2	GPI1	GPI0
OPER	R	R	R	R	R	R	R	R
RESET	X	X	X	X	X	X	X	X

This register reflects the status of the eight General Purpose Input pins (GPI0-7). The General Purpose Input pins are not used.

Miscellaneous Control Register

ADR/SIZ	BASE+008C (8 bits of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	MPIR QEN	REVE ROM	DISSR AM	DISMS T	NOEL BBSY	DISBS YT	ENINT	DISBG N
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PSL	0 PSL	0 PSL	0 PS	0 PS	0 PS	0 PS	0 PS

DISBGN When this bit is high, the VMEbus BGIN filters are disabled. When this bit is low, the VMEbus BGIN filters are enabled. This bit should not be set.

ENINT When this bit is high, the local bus interrupt filters are enabled. When this bit is low, the local bus interrupt filters are disabled. This bit should not be set.

DISBSYT When this bit is low, the minimum VMEbus BBSY* time when the local bus master has been retried off the local bus is 32 local bus clocks. When this bit is high, the minimum VMEbus BBSY* time when the local bus master has been retried off the local bus is 3 local bus clocks.
When a local bus master attempts to access the VMEbus and a VMEbus master attempts to access the local bus, a deadlock is created. The VMEchip2 detects this condition and requests the local bus master to give up the local bus and retry the cycle. This allows the VMEbus master to complete the cycle to the local bus.

If the VMEchip2 receives VMEbus mastership, the local master has not returned from the retry, and this bit is high, VMEchip2 drives VMEbus BBSY* for the minimum time (about 90 ns) and then releases the VMEbus. If the local master does not return from the retry within this 90 ns window, the board loses its turn on the VMEbus. If the VMEchip2 receives VMEbus mastership, the local master has not returned from the retry, and this bit is low, VMEchip2 drives VMEbus BBSY* for a minimum of 32 local bus clocks, which allows the local bus master time to return from the retry and the board does not lose its turn on the VMEbus. For this reason, it is recommended that this bit remain low.

NOELBBSY When this bit is high, the early release feature of bus busy feature on the VMEbus is disabled. The VMEchip2 drives BBSY* low whenever VMEbus AS* is low. When this bit is low, the early release feature of bus busy feature on the VMEbus is not disabled.

DISMST This bit is not used.

DISSRAM This bit is not used.

REVEROM This bit is not used.

MPIRQEN This function is not used. This bit must not be set.

GCSR Programming Model

This section describes the programming model for the Global Control and Status Registers (GCSR) in the VMEchip2. The local bus map decoder for the GCSR registers is included in the VMEchip2. The local bus base address for the GCSR is BASE+0100. The registers in the GCSR are 16 bits wide and they are byte accessible from both the VMEbus and the local bus. The GCSR is located in the 16-bit VMEbus short I/O space and it responds to address modifier codes $29 or $2D. The address of the GCSR as viewed from the VMEbus depends upon the GCSR group select value XX and GCSR board select value Y programmed in the LCSR. The board value Y may be $0 through $E, allowing 15 boards in one group. The value $F is reserved for the location monitors.

The VMEchip2 includes four location monitors (LM0-LM3). The location monitors provide a broadcast signaling capability on the VMEbus. When a location monitor address is generated on the VMEbus, all location monitors in the group are cleared. The signal interrupts SIG0-SIG3 should be used to signal individual boards. The location monitors are located in the VMEbus short I/O space and the specific address is determined by the VMEchip2 group address. The location monitors LM0-LM3 are located at addresses $XXF1, $XXF3, $XXF5, and $XXF7 respectively. A location monitor cycle on the VMEbus is generated by a read or write to VMEbus short I/O address $XXFN, where XX is the group address and N is the specific location monitor address. When the VMEchip2 generates a location monitor cycle to the VMEbus, within its own group, the VMEchip2 DTACKs itself. A VMEchip2 cannot DTACK location monitor cycles to other groups.

The GCSR section of the VMEchip2 contains the following registers: a *chip ID register*, a *chip revision register*, a *location monitor status registe*r, an *interrupt control register*, a *board control register*, and six *general purpose registers*.

The *chip ID* and *revision registers* are provided to allow software to determine the ID of the chip and its revision level. The VMEchip2 has a chip ID of ten. ID codes zero and one are used by the old VMEchip. The initial revision of the VMEchip2 is zero. If mask changes are required, the revision level is incremented.

The *location monitor status register* provides the status of the location monitors. A location monitor bit is cleared when the VMEchip2 detects a VMEbus cycle to the corresponding location monitor address. When the LM0 or LM1 bits are cleared, an interrupt is set to the local bus interrupter. If the LM0 or LM1 interrupt is enabled in the local bus interrupter, then a local bus interrupt is generated. The location monitor bits are set by writing a one to the corresponding bit in the location monitor register. LM0 and LM1 can also be set by writing a one to the corresponding clear bits in the local interrupt clear register.

The *interrupt control register* provides four bits that allow the VMEbus to interrupt the local bus. An interrupt is sent to the local bus interrupter when one of the bits is set. If the interrupt is enabled in the local bus interrupter, then a local bus interrupt is generated. The interrupt bits are cleared by writing a one to the corresponding bit in the interrupt clear register.

The *board control register* allows a VMEbus master to reset the local bus, prevent the VMEchip2 from driving the SYSFAIL signal line, and detect if the VMEchip2 wants to drive the SYSFAIL signal line.

The six *general purpose registers* can be read and written from both the local bus and the VMEbus. These registers are provided to allow local bus masters to communicate with VMEbus masters. The function of these registers is not defined by this specification. The GCSR supports read-modify-write cycles such as TAS.



Caution

The GCSR allows a VMEbus master to reset the local bus. This feature is very dangerous and should be used with caution. The local reset feature is a partial system

reset, not a complete system reset such as power-up reset or SYSRESET. When the local bus reset signal is asserted, a local bus cycle may be aborted. The VMEchip2 is connected to both the local bus and the VMEbus and if the aborted cycle is bound for the VMEbus, erratic operation may result. Communications between the local processor and a VMEbus master should use interrupts or mailbox locations; reset should not be used in normal communications. Reset should be used only when the local processor is halted or the local bus is hung and reset is the last resort.

Programming the GCSR

A complete description of the GCSR is provided in the following tables. Each register definition includes a table with 5 lines:

- Line 1 is the base address of the register as viewed from the local bus and as viewed from the VMEbus, and the number of bits defined in the table.
- Line 2 shows the bits defined by this table.
- Line 3 defines the name of the register or the name of the bits in the register.
- Line 4 defines the operations possible on the register bits as follows:

R	This bit is a read-only status bit.
R/W	This bit is readable and writable.
S/R	Writing a one to this bit sets it. Reading it returns its current status.

- Line 5 defines the state of the bit following a reset as defined below:

P This bit is affected by power-up reset.
S The bit is affected by SYSRESET.
L The bit is affected by local bus reset.
X The bit is not affected by reset.

A summary of the GCSR is shown in Table 2-4.

Table 2-4. VMEchip2 Memory Map (GCSR Summary)

Offsets																	
VME-bus	**Local Bus**	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	CHIP REVISION								CHIP ID							
2	4	LM3	LM2	LM1	LM0	SIG3	SIG2	SIG1	SIG0	RST	ISF	BF	SCON	SYSFL	X	X	X
4	8	GENERAL PURPOSE CONTROL AND STATUS REGISTER 0															
6	C	GENERAL PURPOSE CONTROL AND STATUS REGISTER 1															
8	10	GENERAL PURPOSE CONTROL AND STATUS REGISTER 2															
A	14	GENERAL PURPOSE CONTROL AND STATUS REGISTER 3															
C	18	GENERAL PURPOSE CONTROL AND STATUS REGISTER 4															
E	1C	GENERAL PURPOSE CONTROL AND STATUS REGISTER 5															

VMEchip2 Revision Register

ADR/SIZ	Local bus: BASE+0100/VMEbus: $XXY0 (8 bits)		
BIT	15	. . .	8
NAME	VMEchip2 Revision Register		
OPER	R		
RESET	01 PS		

This register is the VMEchip2 revision register. The revision level for the VMEchip2 starts at zero and is incremented if mask changes are required.

VMEchip2 ID Register

ADR/SIZ	Local bus: BASE+0100/VMEbus: $XXY0 (8 bits)		
BIT	7	. . .	0
NAME	VMEchip2 ID Register		
OPER	R		
RESET	10 PS		

This register is the VMEchip2 ID register. The ID for the VMEchip2 is 10.

VMEchip2 LM/SIG Register

ADR/SIZ	Local bus: BASE+0104/VMEbus: $XXY2 (8 bits)							
BIT	15	14	13	12	11	10	9	8
NAME	LM3	LM2	LM1	LM0	SIG3	SIG2	SIG1	SIG0
OPER	R	R	R	R	S/R	S/R	S/R	S/R
RESET	1 PS	1 PS	1 PS	1 PS	0 PS	0 PS	0 PS	0 PS

This register is the VMEchip2 location monitor register and the interrupt register.

SIG0 The SIG0 bit is set when a VMEbus master writes a one to it. When the SIG0 bit is set, an interrupt is sent to the local bus interrupter. The SIG0 bit is cleared when the local processor writes a one to the SIG0 bit in this register or the CSIG0 bit in the local interrupt clear register.

SIG1 The SIG1 bit is set when a VMEbus master writes a one to it. When the SIG1 bit is set, an interrupt is sent to the local bus interrupter. The SIG1 bit is cleared when the local processor writes a one to the SIG1 bit in this register or the CSIG1 bit in the local interrupt clear register.

SIG2 The SIG2 bit is set when a VMEbus master writes a one to it. When the SIG2 bit is set, an interrupt is sent to the local bus interrupter. The SIG2 bit is cleared when the local processor writes a one to the SIG2 bit in this register or the CSIG2 bit in the local interrupt clear register.

SIG3 The SIG3 bit is set when a VMEbus master writes a one to it. When the SIG3 bit is set, an interrupt is sent to the local bus interrupter. The SIG3 bit is cleared when the local processor writes a one to the SIG3 bit in this register or the CSIG3 bit in the local interrupt clear register.

LM0 This bit is cleared by an LM0 cycle on the VMEbus. When this bit is cleared, an interrupt is set to the local bus interrupter. This bit is set when the local processor or a VMEbus master writes a one to the LM0 bit in this register or the CLM0 bit in local interrupt clear register.

LM1 This bit is cleared by an LM1 cycle on the VMEbus. When this bit is cleared, an interrupt is set to the local bus interrupter. This bit is set when the local processor or a VMEbus master writes a one to the LM1 bit in this register or the CLM1 bit in local interrupt clear register.

LM2 This bit is cleared by an LM2 cycle on the VMEbus. This bit is set when the local processor or a VMEbus master writes a one to the LM0 bit in this register.

LM3 This bit is cleared by an LM3 cycle on the VMEbus. This bit is set when the local processor or a VMEbus master writes a one to the LM3 bit in this register.

VMEchip2 Board Status/Control Register

ADR/SIZ	Local bus: BASE+0104/VMEbus: $XXY2 (8 bits [5 used])							
BIT	7	6	5	4	3	2	1	0
NAME	RST	ISF	BF	SCON	SYSFL			
OPER	S/R	R/W	R	R	R			
RESET	0 PSL	0 PSL	1 PS	X	1 PSL			

This register is the VMEchip2 board status/control register.

SYSFL This bit is set when the VMEchip2 is driving the SYSFAIL signal.

SCON This bit is set if the VMEchip2 is system controller.

BF When this bit is high, the Board Fail signal is active. When this bit is low, the Board Fail signal is inactive. When this bit is set, the VMEchip2 drives SYSFAIL if the inhibit SYSFAIL bit is not set.

ISF When this bit is set, the VMEchip2 is prevented from driving the VMEbus SYSFAIL signal line. When this bit is cleared, the VMEchip2 is allowed to drive the VMEbus SYSFAIL signal line.

RST This bit allows a VMEbus master to reset the local bus. Refer to the note on local reset in the *GCSR Programming Model* section, earlier in this chapter. When this bit is set, a local bus reset is generated. This bit is cleared by the local bus reset.

General Purpose Register 0

ADR/SIZ	Local bus: BASE+0108/VMEbus: $XXY4 (16 bits)		
BIT	15	. . .	0
NAME	General Purpose Register 0		
OPER	R/W		
RESET	0 PS		

This register is a general purpose register that allows a local bus master to communicate with a VMEbus master. The function of this register is not defined by the hardware specification.

General Purpose Register 1

ADR/SIZ	Local bus: BASE+010C/VMEbus: $XXY6 (16 bits)		
BIT	15	. . .	0
NAME	General Purpose Register 1		
OPER	R/W		
RESET	0 PS		

This register is a general purpose register that allows a local bus master to communicate with a VMEbus master. The function of this register is not defined by the hardware specification.

General Purpose Register 2

ADR/SIZ	Local bus: BASE+0110/VMEbus: $XXY8 (16 bits)		
BIT	15	. . .	0
NAME	General Purpose Register 2		
OPER	R/W		
RESET	0 PS		

This register is a general purpose register that allows a local bus master to communicate with a VMEbus master. The function of this register is not defined by the hardware specification.

General Purpose Register 3

ADR/SIZ	Local bus: BASE+0114/VMEbus: $XXYA (16 bits)		
BIT	15	. . .	0
NAME	General Purpose Register 3		
OPER	R/W		
RESET	0 PS		

This register is a general purpose register that allows a local bus master to communicate with a VMEbus master. The function of this register is not defined by the hardware specification.

General Purpose Register 4

ADR/SIZ	Local bus: BASE+0118/VMEbus: $XXYC (16 bits)		
BIT	15	. . .	0
NAME	General Purpose Register 4		
OPER	R/W		
RESET	0 PS		

This register is a general purpose register that allows a local bus master to communicate with a VMEbus master. The function of this register is not defined by the hardware specification.

General Purpose Register 5

ADR/SIZ	Local bus: BASE+011C/VMEbus: $XXYE (16 bits)		
BIT	15	. . .	0
NAME	General Purpose Register 5		
OPER	R/W		
RESET	0 PS		

This register is a general purpose register that allows a local bus master to communicate with a VMEbus master. The function of this register is not defined by the hardware specification.

Introduction

This chapter defines the VME2PCI, PCI local bus to VMEbus interface chip.

There are four internal standard buses on the MVME1603/MVME1604: PowerPC 603/604 Processor bus, PCI Local Bus, ISA Bus, and ‘040bus (VMEchip2’s local bus). The MPC105 PCI Bridge/Memory Controller provides the interface from the processor bus to PCI. The S82378ZB device performs the bridge function between PCI and ISA. Two ASIC devices (VME2PCI and VMEchip2) are used to interface the PCI Local Bus to ‘040bus.

Major Featuresof VME2PCI ASIC

The VME2PCI ASIC is a 225-pin OMPAC device that interfaces between the PCI Local Bus and the ‘040bus which is the local bus of the VMEchip2. The VME2PCI performs address translation from PCI Memory space so that the MPC603/604 processor can get to the VMEchip2 internal registers, the VMEbus F-page, the VMEbus Short I/O area, and to perform pseudo IACK cycles to fetch interrupt vectors from the VMEchip2 and the VMEbus.

The VME2PCI ASIC also performs byte swapping between PCI and the VMEchip2 since PCI is little-endian and VMEbus is big-endian. Little-endian software may have to manipulate multi-byte data when communicating to the VMEbus.

AD13 is routed to the IDSEL pin on the VME2PCI chip; therefore the base address of the VME2PCI Configuration Space is at x00802000 in the PCI Configuration area. Refer to the *Programming Model* section for additional information.

VME2PCI ASIC Programming Model

The VME2PCI ASIC provides the necessary interface between PCI Local Bus and the VMEchip2.

The Control and Status Registers of the VME2PCI ASIC consist of: The VMEchip2 CSR window, the Pseudo IACK Registers, and the PCI Configuration Registers. All these registers can be mapped anywhere in the PCI Memory Space and/or PCI I/O Space. The mapping is done by programming the appropriate base address registers of the VME2PCI Configuration Registers. In addition, the PCI Configuration Registers are also accessible from the PCI Configuration Space. The figures below illustrate how the VME2PCI Control and Status Registers can be mapped:

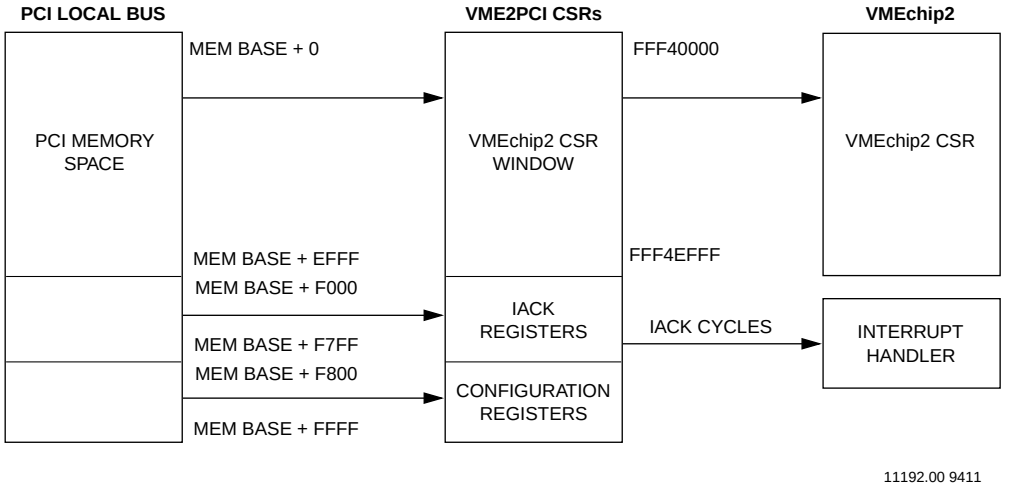


Figure 3-1. VME2PCI's CSR Mapping in PCI Memory Space

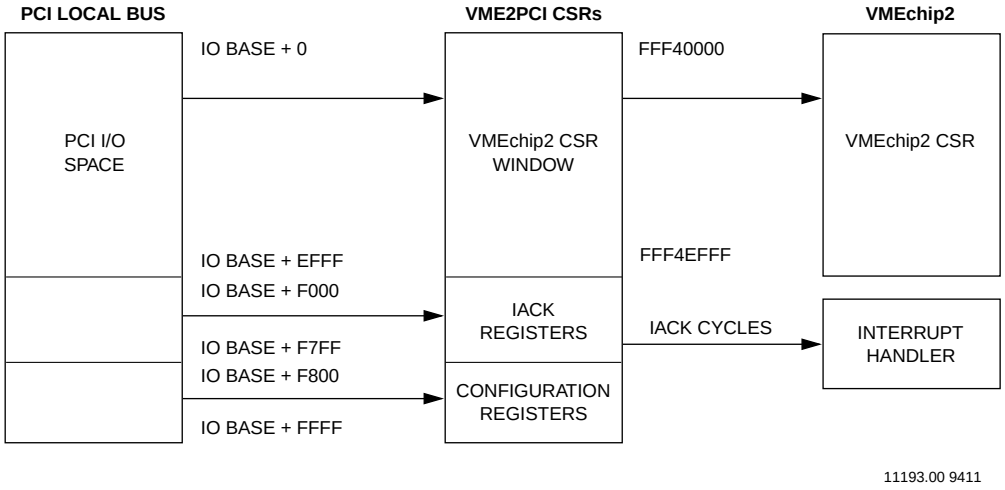


Figure 3-2. VME2PCI's CSR Mapping in PCI I/O Space

VME2PCI Configuration Registers

The PCI Configuration Registers for the VME2PCI ASIC are shown by the following figure:

Device ID 4800 (hex)		**Vendor ID** 1057 (hex)		**00 (hex)**
Status		**Command**		**04 (hex)**
Class Code 068000 (hex)			**Revision ID** 01 (hex)	**08 (hex)**
BIST 00 (hex)	**Header Type** 00 (hex)	**Latency Timer** 00 (hex)	**Cache Line Size** 00 (hex)	**0C (hex)**
IO Map Base Address (Bits 31-16)		00000000000000 (binary)	0 1	**10 (hex)**
Memory Map Base Address (Bits 31-16)		000000000000 (binary)	0 00 0	**14 (hex)**
Not Supported				**18 (hex)**
Not Supported				**1C (hex)**
Not Supported				**20 (hex)**
Not Supported				**24 (hex)**
Reserved				**28 (hex)**
Reserved				**2C (hex)**
Not Supported				**30 (hex)**
Reserved				**34 (hex)**
Not Supported				**38 (hex)**
Max Latency 00 (hex)	**Minimum Grant** 00 (hex)	**Interrupt Pin** 01 (hex)	**Interrupt Line**	**3C (hex)**
PCI Slave End Address 1		**PCI Slave Start Address 1**		**40 (hex)**
Reserved	**EN1**	**Address Offset 1**		**44 (hex)**
PCI Slave End Address 2		**PCI Slave Start Address 2**		**48 (hex)**
Reserved	**EN2**	**Address Offset 2**		**4C (hex)**
Reserved			**IEN** **ILVL**	**50 (hex)**

Figure 3-3. VME2PCI Configuration Registers

Note: Writes to Reserved locations are don't care, reads to Reserved locations return zeros.

Table 3-1. VME2PCI ASIC's PCI Configuration Registers Summary

Processor Address	PCI Configuration Address	Register Name	R/W	Reset Value (hex)
80802000	00802000	PCI Vendor ID	R	1057 (hex)
80802002	00802002	PCI Device ID	R	4800 (hex)
80802004	00802004	PCI Command	R/W	0000 (hex)
80802006	00802006	PCI Status	R/W	0000 (hex)
80802008	00802008	PCI Revision ID	R	01
80802009	00802009	PCI Class Code	R	068000 (hex)
8080200C	0080200C	PCI Cache Line Size	R/W	00 (hex)
8080200D	0080200D	PCI Latency Timer	R/W	00 (hex)
8080200E	0080200E	PCI Header Type	R	00 (hex)
80802010	00802010	PCI I/O Base Address	R/W	00000001 (hex)
80802014	00802014	PCI Memory Base Address	R/W	00000000 (hex)
8080203C	0080203C	PCI Interrupt Line	R/W	00 (hex)
8080203D	0080200D	PCI Interrupt Pin	R	01 (hex)
8080203E	0080200E	PCI Minimum Grant	R	00
8080203F	0080200F	PCI Maximum Latency	R	00
80802040	00802040	Slave Starting Address 1	R/W	0000 (hex)
80802042	00802042	Slave Ending Address 1	R/W	0000 (hex)
80802044	00802044	Slave Address Offset 1	R/W	0000 (hex)
80802046	00802046	Slave Address Enable 1	R/W	00 (hex)
80802048	00802048	Slave Starting Address 2	R/W	0000 (hex)
8080204A	0080204A	Slave Ending Address 2	R/W	0000 (hex)
8080204C	0080204C	Slave Address Offset 2	R/W	0000 (hex)
8080204D	0080204D	Slave Address Enable 2	R/W	00 (hex)
80802050	00802050	Interrupt Status and Control	R/W	0000 (hex)

Vendor ID Register

REG	Vendor ID Register - 80802000 (hex) (16 bits)		
BIT	AD15		AD00
FIELD	VENDID		
OPER	R		
RESET	1057 (hex)		

The Vendor ID Register identifies the unique manufacturer number assigned to Motorola by the PCI SIG. This register is accessible in the VME2PCI Memory and/or I/O Space at offset xF800 (hex).

Device ID Register

REG	Device ID Register - 80802002 (hex) (16 bits)		
BIT	AD31		AD16
FIELD	DEVID		
OPER	R		
RESET	4800 (hex)		

The Device ID Register identifies the unique ID number of this particular device. This number must be between 4800 (hex) and 4BFF (hex) and is assigned by MCG. This register is accessible in the VME2PCI Memory and/or I/O Space at offset xF802 (hex).

PCI Command Register

REG	PCI Command Register - 80802004 (hex) (16 bits)															
BIT	AD 15	AD 14	AD 13	AD 12	AD 11	AD 10	AD 09	AD 08	AD 07	AD 06	AD 05	AD 04	AD 03	AD 02	AD 01	AD 00
FIELD														MASTR	MEMSP	IOSP
OPER	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is accessible in the VME2PCI Memory and/or I/O Space at offset xF804 (hex) if enabled.

IOSP I/O Space Enable. The VME2PCI device will respond to the I/O space accesses when this bit is set.

MEMSP Memory Space Enable. The VME2PCI device will respond to the Memory space accesses when this bit is set.

MASTR PCI Mastership Enable. The VME2PCI device is allowed to become PCI bus master when this bit is set.

PCI Status Register

REG	PCI Status Register - 80802006 (hex) (16 bits)															
BIT	AD31	AD30	AD29	AD28	AD27	AD26	AD25	AD24	AD23	AD22	AD21	AD20	AD19	AD18	AD17	AD16
FIELD			RCVMA	RCVTA	SIGTA	SELTIM			FSST2B							
OPER	R	R	RC	RC	RC	R	R	R	R	R	R	R	R	R	R	R
RESET	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

This register is accessible in the VME2PCI Memory and/or I/O Space at offset xF806 (hex).

FSTB2B Fast Back-to-Back Capable. This bit is always read as zero which means that the VME2PCI device does not support fast back-to-back capability.

SELTIM Device Select (DEVSEL_) Timing. The VME2PCI device hardwires these two bits to 01 to indicate medium (1 clock) DEVSEL_ timing.

SIGTA Signaled Target Abort. Asserted if the VME2PCI device ends a slave PCI bus cycle with Target Abort. This bit is cleared by writing a 1 to the bit; writing a 0 has no effect.

RCVTA Received Target Abort. Asserted if the VME2PCI device receives a Target Abort while performing PCI bus cycle as PCI master. This bit is cleared by writing a 1 to the bit; writing a 0 has no effect.

RCVMA Received Master Abort. Asserted if the VME2PCI device terminates the PCI bus cycle with a Master Abort while performing PCI bus cycle as PCI master. This bit is cleared by writing a 1 to the bit; writing a 0 has no effect.

Revision ID Register

REG	Revision ID Register - 80802008 (hex) (8 bits)		
BIT	AD07		AD00
FIELD	REVID		
OPER	R		
RESET	01 (hex)		

The Revision ID Register identifies the revision of the VME2PCI device. This register is internally hardwired by the VME2PCI device. This register is accessible in the VME2PCI Memory and/or I/O Space at offset xF808 (hex).

PCI Class Code Register

REG	Class Code Register - 8080200B (hex) (8 of 24 bits)		
BIT	AD31		AD24
FIELD	BASE		
OPER	R		
RESET	06 (hex)		

REG	Class Code Register - 8080200A (hex) (8 of 24 bits)		
BIT	AD23		AD16
FIELD	SUB		
OPER	R		
RESET	80 (hex)		

REG	Class Code Register - 80802009 (hex) (8 of 24 bits)		
BIT	AD15		AD08
FIELD	PROG		
OPER	R		
RESET	00 (hex)		

This register is accessible in the VME2PCI Memory and/or I/O Space at offset xF809 (hex).

BASE Base Class Code. These bits are hardwired with a value of $06 to identify the VME2PCI device as a PCI Bridge device.

SUB Sub Class Code. These bits are hardwired with a value of $80 to identify the VME2PCI device as a VMEbus Bridge device since the VMEbus falls into the “OTHER” bridge category.

PROG Programming Interface Class Code. This field is always $00.

PCI Cache Line Size Register

REG	Cache Line Size Register - 8080200C (hex) (8 bits)		
BIT	AD07		AD00
FIELD	CLINE		
OPER	R		
RESET	00 (hex)		

This register is accessible in the VME2PCI Memory and/or I/O Space at offset xF80C (hex). This register is hard-wired to 00 (hex).

PCI Latency Timer Register

REG	Latency Timer Register - 8080200D(hex) (8 bits)		
BIT	AD15		AD08
FIELD	LTIME		
OPER	R		
RESET	00 (hex)		

This register is also accessible in the VME2PCI Memory and/or I/O Space at offset xF80D (hex). This register is not implemented by the VME2PCI device. It is hardwired to 00 (hex).

PCI Header Type Register

REG	Header Type Register - 8080200E (hex) (8 bits)		
BIT	AD23		A16
FIELD	HEADER		
OPER	R		
RESET	00 (hex)		

The Header Type Register defines the format for bytes 10 (hex) to 3F (hex) of the VME2PCI's configuration header. It is hardwired to 00 (hex). This register is also accessible in the VME2PCI Memory and/or I/O Space at offset xF80E (hex).

Built-In-Self-Test Register

REG	BIST Register - 8080200F (hex) (8 bits)		
BIT	AD31		AD24
FIELD	BIST		
OPER	R		
RESET	00 (hex)		

The VME2PCI device does not support BIST; therefore, this register is hardwired to 00 (hex). This register is accessible in the VME2PCI Memory and/or I/O Space at offset xF80F (hex).

PCI IO Map Base Register

REG	IO Map Base Register - 80802010 (hex) (16 of 32bits)															
BIT	AD15	AD14	AD13	AD12	AD11	AD10	AD09	AD08	AD07	AD06	AD05	AD04	AD03	AD02	AD01	AD00
FIELD															Reserved	IO/MEM–
OPER	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

REG	PCI Status Register - 80802012 (hex) (16 of 32bits)															
BIT	AD31	AD30	AD29	AD28	AD27	AD26	AD25	AD24	AD23	AD22	AD21	AD20	AD19	AD18	AD17	AD16
FIELD	IOBASE															
OPER	R/W															

REG	PCI Status Register - 80802012 (hex) (16 of 32bits)
RESET	0000 (hex)

This register controls the mapping of the VME2PCI Control and Status Registers in PCI I/O Space. This register is also accessible in the VME2PCI Memory and/or I/O Space at offset xF810 (hex).

IO/MEM_ I/O Space or Memory Space Indicator. This bit is hardwired to a logic one to indicate I/O Space.

Reserved This bit is hardwired to zero.

IOBASE I/O Space Base Address. These bits define the I/O Space base address for the VME2PCI CSRs.

PCI Memory Map Base Register

REG	Memory Map Base Register - 80802014 (hex) (16 of 32bits)															
BIT	AD15	AD14	AD13	AD12	AD11	AD10	AD09	AD08	AD07	AD06	AD05	AD04	AD03	AD02	AD01	AD00
FIELD													PREFETCH	MEMTYPE		IO/MEM_
OPER	R	R	R	R	R	R	R	R	R	R	R	R	R	R		R
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0

REG	PCI Status Register - 80802016 (hex) (16 of 32bits)															
BIT	AD31	AD30	AD29	AD28	AD27	AD26	AD25	AD24	AD23	AD22	AD21	AD20	AD19	AD18	AD17	AD16
FIELD	MEMBASE															

REG	PCI Status Register - 80802016 (hex) (16 of 32bits)
OPER	R/W
RESET	0000 (hex)

This register controls the mapping of the VME2PCI Registers in PCI Memory Space. This register is also accessible in the VME2PCI Memory and/or I/O Space at offset xF814 (hex).

IO/MEM_ I/O Space or Memory Space Indicator. This bit is hardwired to a logic zero to indicate Memory Space.

MEMTYPE These bits are hardwired to zero to indicate that the VME2PCI CSRs can be located anywhere in the 32-bit address space.

PREFETCH This bit is hardwired to zero to indicate that the VME2PCI CSRs are not prefetchable.

MEMBASE Memory Space Base Address. These bits define the Memory Space base address for the VME2PCI CSRs.

Interrupt Line Register

REG	Interrupt Line Register - 8080203C (hex) (8 bits)		
BIT	AD07		AD00
FIELD	ILINE		
OPER	R/W		
RESET	00 (hex)		

This register is used by software to communicate about the interrupt line routing information. This register is also accessible in the VME2PCI Memory and/or I/O Space at offset xF83C (hex).

PCI Interrupt Pin Register

REG	Interrupt Pin Register - 8080203D (hex) (8 bits)		
BIT	AD15		AD08
FIELD	IPIN		
OPER	R		
RESET	01 (hex)		

This register is hardwired to 01 (hex) to indicate that the VME2PCI device interrupts on the INTA_ signal line. This register is also accessible in the VME2PCI Memory and/or I/O Space at offset xF83D (hex).

PCI Minimum Grant Register

REG	Minimum Grant Register - 8080203E (hex) (8 bits)		
BIT	AD23		AD16
FIELD	MINGNT		
OPER	R		
RESET	00 (hex)		

This register is hardwired to 00 (hex) by the VME2PCI device. This register is also accessible in the VME2PCI Memory and/or I/O Space at offset xF83E (hex).

PCI Maximum Latency Register

REG	Maximum Latency Register - 8080203F (hex) (8 bits)		
BIT	AD31		AD24
FIELD	MAXLAT		
OPER	R		
RESET	00 (hex)		

This register is hardwired to 00 (hex) by the VME2PCI device. This register is also accessible in the VME2PCI Memory and/or I/O Space at offset xF83F (hex).

PCI Slave Start Address 1 Register

REG	PCI Slave Start Address 1 - 80802040 (hex) (16 bits)		
BIT	AD15		AD00
FIELD	SLVSTART1		
OPER	R/W		
RESET	0000 (hex)		

This register is also accessible in the VME2PCI Memory and/or I/O Space at offset xF840 (hex). This register defines the most significant 16-bit starting address of the first PCI Memory Space for the VME2PCI's slave interface. PCI memory accesses within the range of the starting and ending address are passed on to the '040bus (VMEchip2). The address presented on the '040bus is controlled by the Address Offset 1 Register.

PCI Slave End Address 1 Register

REG	PCI Slave End Address 1 - 80802042 (hex) (16 bits)		
BIT	AD 31		AD 16
FIELD	SLVEND1		
OPER	R/W		
RESET	0000 (hex)		

This register is also accessible in the VME2PCI Memory and/or I/O Space at offset xF842 (hex). This register defines the most significant 16-bit ending address of the first PCI Memory Space for the VME2PCI's slave interface. PCI memory accesses within the range of the starting and ending address are passed on to the '040bus (VMEchip2). The address presented on the '040bus is controlled by the Address Offset 1 Register.

PCI Slave Address Offset 1 Register

REG	PCI Slave Address Offset 1 - 80802044 (hex) (16 bits)		
BIT	AD 15		AD 00
FIELD	SLVADO1		
OPER	R/W		
RESET	0000 (hex)		

This register is also accessible in the VME2PCI Memory and/or I/O Space at offset xF844 (hex). This register is used to translate the most significant 16 bits of the address to be presented to the '040bus. '040bus address (A31-A16) is equal to the sum of PCI address (AD31-AD16) and the value of this SLVADO1 Register.

PCI Slave Control 1 Register

REG	Slave Control 1 - 80802046 (hex) (8 bits)							
BIT	AD23	AD22	AD22	AD20	AD19	AD18	AD17	AD16
FIELD								EN1
OPER	R	R	R	R	R	R	R	R/W
RESET	0	0	0	0	0	0	0	0

This register is accessible in the VME2PCI Memory and/or I/O Space at offset xF846 (hex).

EN1 PCI Slave Enable 1. When set, this bit enables the VME2PCI to respond to PCI Memory Space accesses that fall between the PCI Slave Starting Address 1 Register and the PCI Slave Ending Address 1 Register.

PCI Slave Start Address 2 Register

REG	PCI Slave Start Address 2 - 80802048 (hex) (16 bits)	
BIT	AD15	AD00
FIELD	SLVSTART2	
OPER	R/W	
RESET	0000 (hex)	

This register is also accessible in the VME2PCI Memory and/or I/O Space at offset xF848 (hex). This register defines the most significant 16-bit starting address of the second PCI Memory Space for the VME2PCI's slave interface. PCI memory accesses within the range of the starting and ending address are passed on to the '040bus (VMEchip2). The address presented on the '040bus is controlled by the Address Offset 2 Register.

PCI Slave End Address 2 Register

REG	PCI Slave End Address 2 - 8080204A (hex) (16 bits)		
BIT	AD31		AD16
FIELD	SLVEND2		
OPER	R/W		
RESET	0000 (hex)		

This register is also accessible in the VME2PCI Memory and/or I/O Space at offset xF84A (hex). This register defines the most significant 16-bit ending address of the second PCI Memory Space for the VME2PCI's slave interface. PCI memory accesses within the range of the starting and ending address are passed on to the '040bus (VMEchip2). The address presented on the '040bus is controlled by the Address Offset 2 Register.

PCI Slave Address Offset 2 Register

REG	PCI Slave Address Offset 2 - 8080204C (hex) (16 bits)		
BIT	AD15		AD00
FIELD	SLVADO1		
OPER	R/W		
RESET	0000 (hex)		

This register is also accessible in the VME2PCI Memory and/or I/O Space at offset xF84C (hex). This register is used to translate the most significant 16 bits of the address to be presented to the '040bus. '040bus address (A31-A16) is equal to the sum of PCI address (AD31-AD16) and the value of this SLVADO2 Register.

PCI Slave Control 2 Register

REG	Slave Control 2 - 8080204E (hex) (8 bits)							
BIT	AD23	AD22	AD22	AD20	AD19	AD18	AD17	AD16
FIELD								EN2
OPER	R	R	R	R	R	R	R	R/W
RESET	0	0	0	0	0	0	0	0

This register is accessible in the VME2PCI Memory and/or I/O Space at offset xF84E (hex).

EN2 PCI Slave Enable 2. When set, this bit enables the VME2PCI to respond to PCI Memory Space accesses that fall between the PCI Slave Starting Address 2 Register and the PCI Slave Ending Address 2 Register.

Interrupt Control and Status Register

REG	Slave Control 2 - 80802050 (hex) (8 bits)							
BIT	AD07	AD06	AD05	AD04	AD03	AD02	AD01	AD00
FIELD					IEN	ILVL		
OPER	R	R	R	R	R/W	R		
RESET	0	0	0	0	0	0	0	0

This register is accessible in the VME2PCI Memory and/or I/O Space at offset xF850 (hex).

IEN Interrupt Enable. When set, this bit enables the VME2PCI assert the PCI interrupt line if there is an interrupt request from the VMEchip2 (i.e. IPL2-IPL0 > 0). When cleared, VME2PCI will not assert PCI interrupt.

ILVL Pending Interrupt Level. These three bits reflect the state of the IPL2-IPL0 lines from the VMEchip2. Software read these bits to determine which IACK Register to read to obtain the appropriate interrupt vector from the VMEchip2.

VMEchip2 CSR Window

The VMEchip2 CSR Window provides a means to access the VMEchip2 Control and Status Registers. The VMEchip2 CSR Window occupies from offset x0000 (hex) to xEFFF (hex) of the VME2PCI Memory Space and/or the VME2PCI I/O Space. Accesses to this range will be translated to the xFFF40000 (hex) to xFFF4EFFF (hex) address range in the '040bus. Note that the VMEchip2 actually only responds to '040 address range from FFF40000 (offset x0000 (hex)) to FFF40FFF (offset x0FFF (hex)).

Because the VMEbus interface is optional, some VMEchip2 functions are not used by the MVME1603/MVME1604 SBC. These unused functions are: ABT (abort) switch support, RST (RESET) switch support, BOARD FAIL status/control, ROM support, and SRAM support.

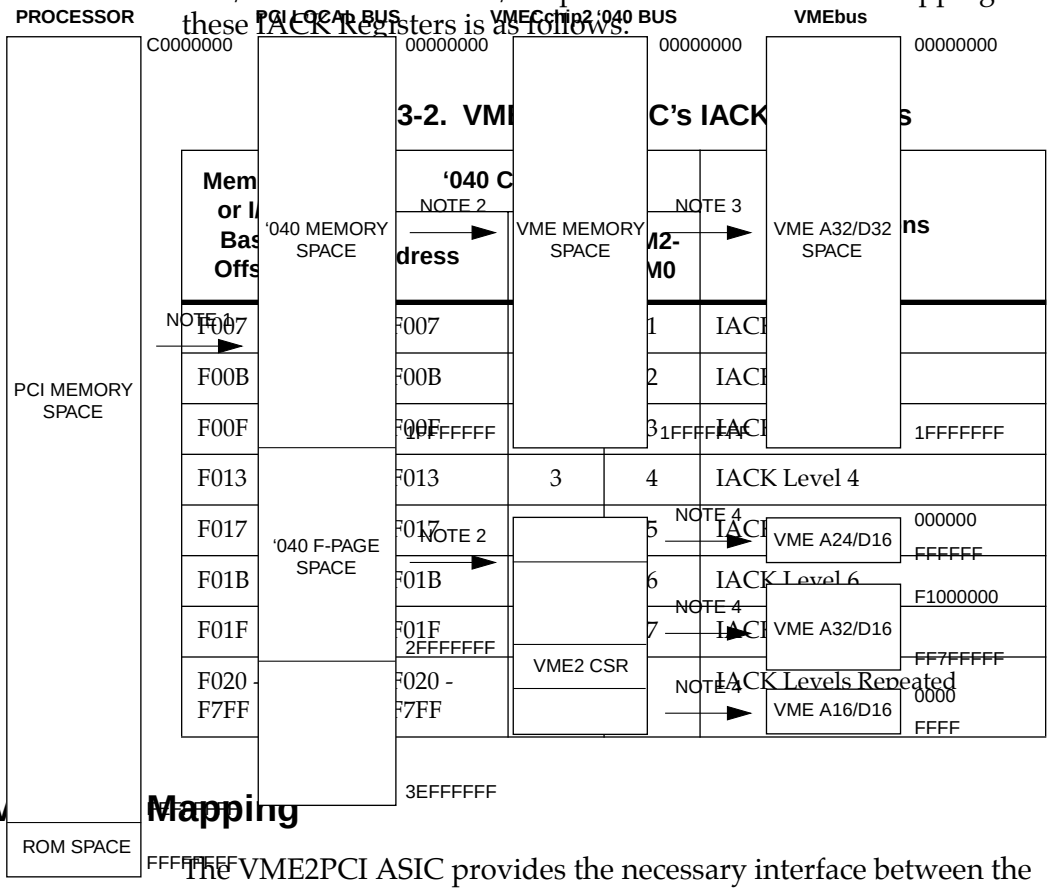
The VMEchip2 also provides two 32-bit tick timers and a watchdog timer. Software should not use these timers for system timing functions since there may be versions of the MVME1603/MVME1604 SBC without the VMEchip2 (and without the VME2PCI ASIC).

Refer to Chapter 2 for programming information on the VMEchip2.

VME2PCI IACK Registers

The VME2PCI device provides means to obtain interrupt vectors from the VMEchip2 via the IACK Registers. Accessing these registers causes the VME2PCI to perform pseudo IACK bus cycles to the VMEchip2 to obtain the appropriate interrupt vectors.

These IACK Registers can appear in the VME2PCI's Memory Space and/or the VME2PCI's I/O Space at offset xF000. The mapping for these IACK Registers is as follows:



V Mapping

The VME2PCI ASIC provides the necessary interface between the PCI Local Bus and the VMEchip2 ASIC. 11194.00 9411

Processor's Point of View

The processor can access any address range in the VMEbus with the help from the address translation capabilities of the VME2PCI and the VMEchip2. The following figure illustrates a mapping example for the VMEbus from the view of the processor:

Figure 3-4. VMEbus Mapping Example

Notes:

1. Fixed mapping done by MPC105
2. Programmable mapping done by VME2PCI ASIC
3. Programmable mapping done by VMEchip2 ASIC
4. Fixed mappings done by VMEchip2 ASIC
5. VME2 CSR space includes address space for 1x7 local resources which are not implemented on the MVME1603/MVME1604.

VMEbus Point of View

No address translation is done by the VME2PCI ASIC for accesses initiated by the VMEchip2; therefore, all accesses from the VMEbus may need to be translated by the VMEchip2 because the onboard DRAM resides in the upper 2GB address range in the PCI memory space (from 80000000 to FFFFFFFF). Refer to Chapter 2 for additional information on programming the VMEchip2.

Programming Details 4

Introduction

This chapter contains details of several programming functions that are not tied to any specific ASIC chip.

PCI Arbitration

There are 6 potential PCI bus masters on the MVME1603/MVME1604 SBC. They are: MPC105 (CPU), IBC, Am79C970 Ethernet Controller, 53C825 (or 53C810) SCSI Controller, VME2PCI ASIC, and the PMC Slot. The PCI arbitration support for these six devices are provided by the IBC which supports flexible arbitration modes: fixed priority scheme, rotating scheme, and mixed priority scheme. The IBC registers that control the arbitration mode are the PCI Arbiter Priority Control (PAPC) Register and the PCI Arbiter Priority Control Extension (ARBPRIX) Register. The PAPC register and the ARBPRIX register default to 04 (hex) and 00 (hex), respectively. This default configuration puts the CPU (MPC105) at the highest priority level. Refer to S82378ZB Reference Manual for programming information.

The following figure shows the arbitration configuration diagram of the IBC:

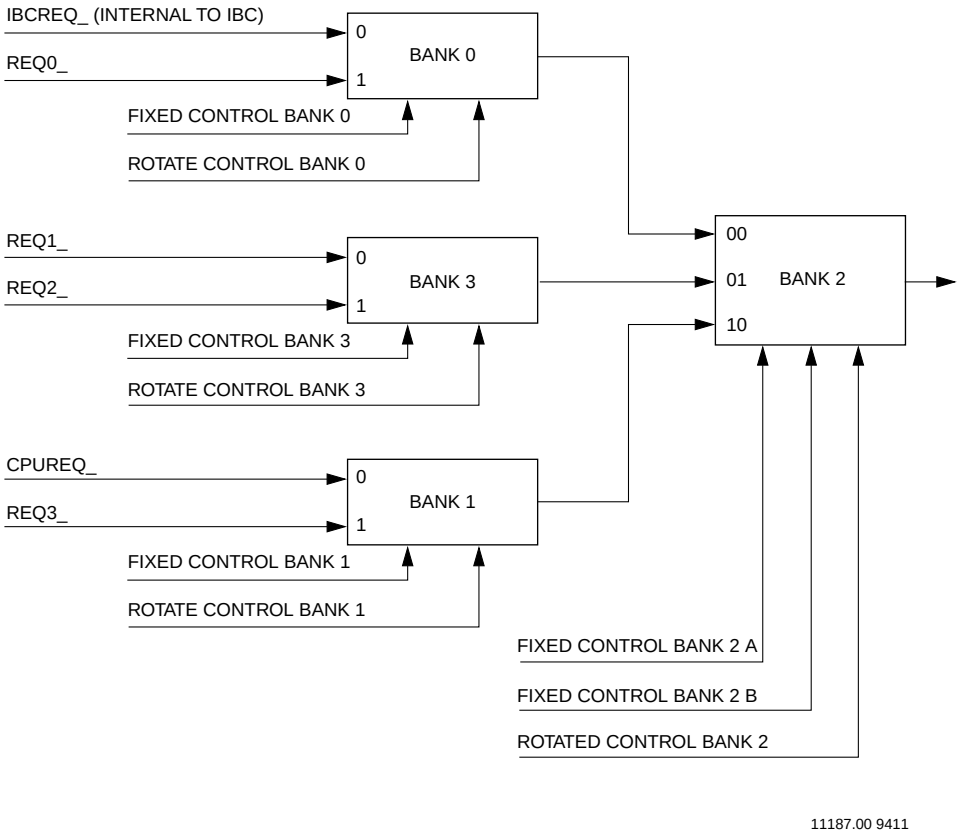


Figure 4-1. IBC Arbiter Configuration Diagram

The PCI arbitration assignments for all PCI masters on the MVME1603/MVME1604 are as follows:

Table 4-1. PCI Arbitration Assignments

PCI BUS REQUEST	CPUREQ_	IBCREQ_	REQ0_	REQ1_	REQ2_	REQ3_
PCI MASTER	CPU (MPC105)	IBC (Internal)	SCSI (53C825)	LANC (Am79C970)	VME (VME2PCI)	PMC Slot

The following table shows the fixed arbitration modes supported by the MVME1603/MVME1604 SBC:

Table 4-2. Arbitration Priority in Fixed Mode

MODE	Bank Fixed Priority Mode Select Bits					PRIORITY						NOTE
	3	2b	2a	1	0	Highest					Lowest	
00	0	0	0	0	0	IBC	SCSI	VME	PMC	CPU	LANC	
01	0	0	0	0	1	SCSI	IBC	VME	PMC	CPU	LANC	
02	0	0	0	1	0	IBC	SCSI	VME	PMC	LANC	CPU	
03	0	0	0	1	1	SCSI	IBC	VME	PMC	LANC	CPU	
04	0	0	1	0	0	CPU	LANC	IBC	SCSI	VME	PMC	1
05	0	0	1	0	1	CPU	LANC	SCSI	IBC	VME	PMC	
06	0	0	1	1	0	LANC	CPU	IBC	SCSI	VME	PMC	
07	0	0	1	1	1	LANC	CPU	SCSI	IBC	VME	PMC	
08	0	1	0	0	0	VME	PMC	CPU	LANC	IBC	SCSI	
09	0	1	0	0	1	VME	PMC	CPU	LANC	SCSI	IBC	
0A	0	1	0	1	0	VME	PMC	LANC	CPU	IBC	SCSI	
0B	0	1	0	1	1	VME	PMC	LANC	CPU	SCSI	IBC	
0C-0F	0	1	1	X	X	Reserved						
10	1	0	0	0	0	IBC	SCSI	PMC	VME	CPU	LANC	
11	1	0	0	0	1	SCSI	IBC	PMC	VME	CPU	LANC	
12	1	0	0	1	0	IBC	SCSI	PMC	VME	LANC	CPU	
13	1	0	0	1	1	SCSI	IBC	PMC	VME	LANC	CPU	
14	1	0	1	0	0	CPU	LANC	IBC	SCSI	PMC	VME	
15	1	0	1	0	1	CPU	LANC	SCSI	IBC	PMC	VME	
16	1	0	1	1	0	LANC	CPU	IBC	SCSI	PMC	VME	
17	1	0	1	1	1	LANC	CPU	SCSI	IBC	PMC	VME	
18	1	1	0	0	0	PMC	VME	CPU	LANC	IBC	SCSI	
19	1	1	0	0	1	PMC	VME	CPU	LANC	SCSI	IBC	
1A	1	1	0	1	0	PMC	VME	LANC	CPU	IBC	SCSI	
1B	1	1	0	1	1	PMC	VME	LANC	CPU	SCSI	IBC	
1C-1F	1	1	1	X	X	Reserved						

Notes:

1. Default configuration after a reset.

Interrupt Supports

There are both maskable interrupts and non-maskable interrupts on MVME1603/MVME1604. The interrupt architecture of the MVME1603/MVME1604 SBC is shown in the following figure:

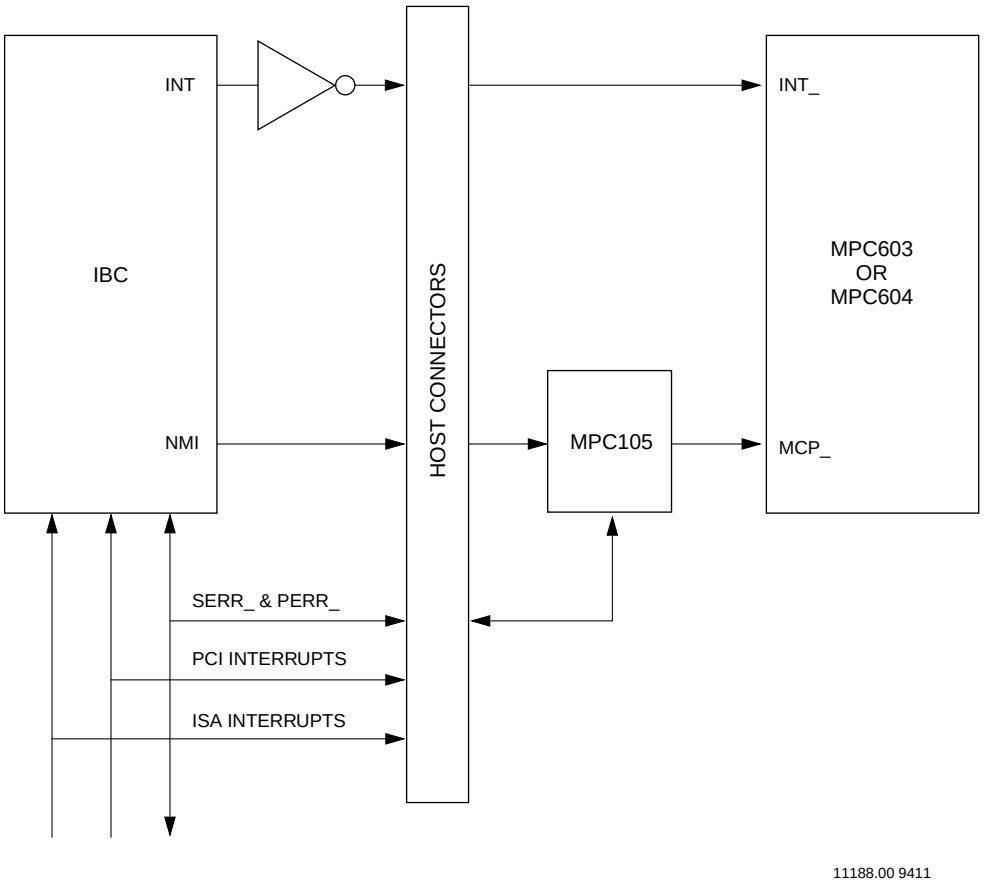


Figure 4-2. MVME1603/MVME1604 Interrupt Architecture

Machine Check Interrupt (MCP_)

The IBC can be programmed to assert NMI when it detects either SERR_ low on the PCI Local Bus or IOCHK_ low on ISA bus. However, IOCHK_ is not used on the MVME1603/MVME1604 SBC. The MPC105 will assert MCP_ to the processor upon detecting a high level on NMI from the IBC.

Note that MPC105 also monitors SERR_ and PERR_ and can be programmed to asserted MCP_ when it detects a low level on either SERR_ or PERR_.The MPC105 can also be programmed to assert MCP_ under many other conditions. Refer to the MPC105 Eagle Specification for additional information on MCP_ interrupt.

Maskable Interrupts

There are 15 interrupts requests supported by the IBC. These 15 interrupts are ISA-type interrupts that are functionally equivalent to two 82C59 interrupt controllers. Except for IRQ0, IRQ1, IRQ2, IRQ8_, and IRQ13, each of the interrupt lines can be configured for either edge-sensitive mode or level-sensitive mode by programming the appropriate ELCR registers in the IBC.

There is also support for four PCI interrupts, INT3_-INT0_. The IBC has four PIRQ Route Control Registers to allow each of the PCI interrupt lines to be routed to any of eleven ISA interrupt lines (IRQ0, IRQ1, IRQ2, IRQ8_, and IRQ13 are reserved for ISA system interrupts). Since PCI interrupts are defined as level-sensitive, software must program the selected IRQ(s) for level-sensitive mode. Note that more than one PCI interrupts can be routed to the same ISA IRQ line.

The following figure shows the interrupt structure of the IBC.

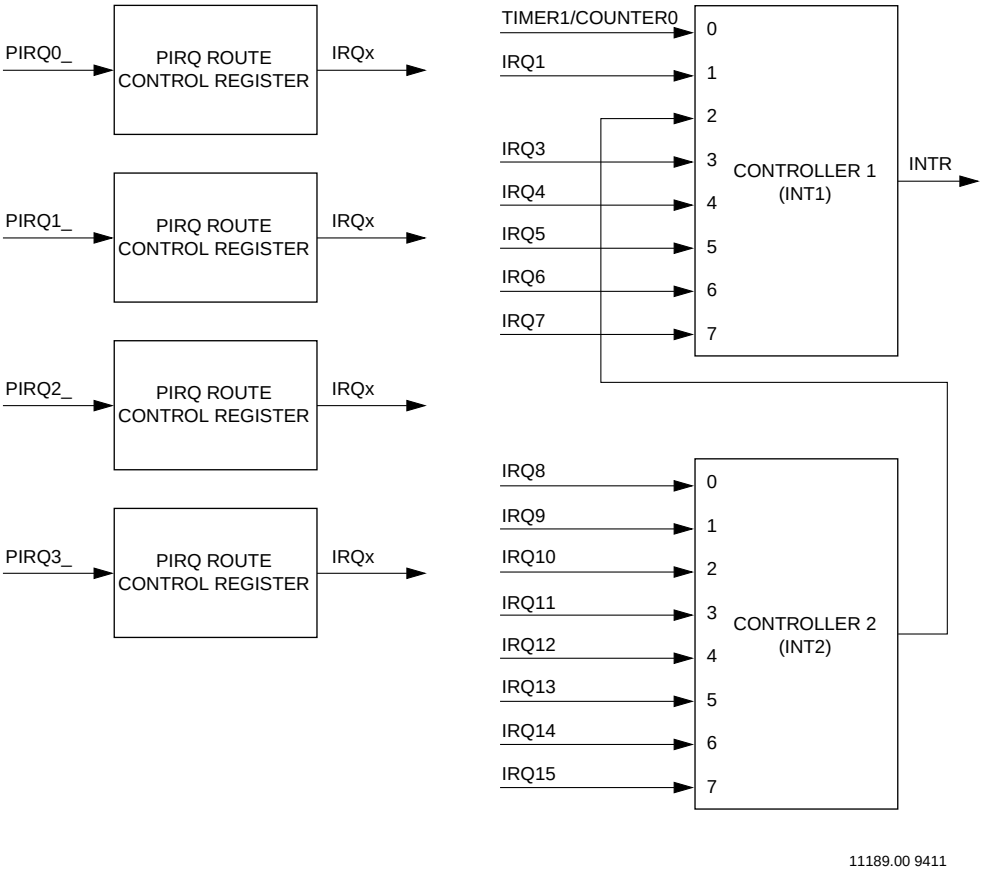


Figure 4-3. IBC Interrupt Handler Block Diagram

The assignments of the PCI and ISA interrupts supported by the IBC are as follows:

Table 4-3. IBC PCI/ISA Interrupt Assignments

PRI	ISA IRQ	PCI IRQ	Con-troller	Edge/Level	Polar-ity	Interrupt Source	Notes
1	IRQ0		INT1	Edge	High	Timer 1 / Counter 0	1
2	IRQ1			Edge	High	Keyboard	2
3-10	IRQ2			Edge	High	Cascade Interrupt from INT2	
3	IRQ8_		INT2	Edge	Low	ABT (abort) Switch Interrupt	
4	IRQ9			Level	Low	Z8536 CIO	3, 4
						Z85230 ESCC	
5	IRQ10	PIRQ0_		Level	Low	Am79C970 Interrupt	3, 5
6	IRQ11	PIRQ1_		Level	Low	VME2PCI Interrupt	3, 5
7	IRQ12			Edge	High	Mouse	
8	IRQ13			Edge	High	Reserved	
9	IRQ14	PIRQ2_		Level	Low	53C825 Interrupt	3, 5
10	IRQ15	PIRQ3_		Level	Low	GD5434 Interrupt	3, 5, 6
						PMC Interrupt	3, 5, 6
11	IRQ3		INT1	Edge	High	COM2 (Async Serial Port 2)	
12	IRQ4			Edge	High	COM1 (Async Serial Port 1)	
13	IRQ5			Edge	High	Not Used (Reserved for Audio)	
14	IRQ6			Edge	High	Reserved for Floppy Interrupt	
15	IRQ7			Edge	High	Parallel Port Interrupt	

Notes:

1. Internally generated by the IBC.
2. Bit 4 of ISA Clock Divisor Register in the IBC must be set to 0 to support external keyboard interrupt (from the ISASIO device).
3. After a reset, all ISA IRQ interrupt lines default to edge-sensitive mode.
4. Interrupts from Z8536 and Z85230 devices are externally wire-ORed. External logic will determine which device to acknowledge during a pseudo IACK cycle. The Z8536 CIO

has higher priority than the Z85230 ESCC. This IRQ MUST be programmed for level-sensitive mode.

5. These PCI interrupts are routed to the ISA interrupts by programming the PRIQ Route Control Registers in the IBC. The PCI to ISA interrupt assignments in this table are suggested. Each ISA IRQ to which a PCI interrupt is routed to MUST be programmed for level-sensitive mode.
6. PIRQ3_ are shared by the GD5434 device and the PMC Slot.

Handling VMEchip2 Interrupts

VMEchip2 interrupts consist of interrupts from the VMEbus IRQ lines and from the VMEchip2 internal resources (i.e. DMA and Timers). Program the VMEchip2 interrupt control registers as if the system is MC68040-base, i.e. with interrupt priority levels from 1 through 7. When an interrupt is pending, the VMEchip2 asserts three encoded interrupt request lines (IPL2_-IPL0_) to the VME2PCI device. An interrupt then is issued by the VME2PCI device to the processor through the IBC.

After the software is informed by the IBC that the source of the interrupt is the VME2PCI, it determines the interrupt level to acknowledge the VMEchip2 by examining the ILVL status bits of the Interrupt Control and Status Register in the VME2PCI ASIC. Finally, to get the interrupt vector from the VMEchip2, the interrupt handling routine must read the appropriate Pseudo IACK Registers.

Handling Z8536 and Z85230 Interrupts

After the software is informed by the IBC that the source of the interrupt is the Z85230/Z8536 devices, it can either poll the two devices or perform an 8-bit read access to the Z85230/Z8536 Pseudo IACK Register to get the interrupt vector. Refer to the Z85230 and the Z8536 Data Sheets for programming information and additional information about their interrupt structures.

ABT (abort) Interrupt

The MVME1600-0001 board can be programmed to generate an interrupt to the processor via ISA Interrupt IRQ8_ when the ABT (abort) switch is activated (See also the ABT Switch Section). The ABORT_ signal is also routed to pin PB7 of the Z8536 device. Refer to the 82C378ZB and the Z8536 Data Sheets for programming information.

DMA Channels

There are seven DMA channels supported by the IBC. These DMA channels are assigned as follows:

Table 4-4. IBC DMA Channel Assignments

IBC Priority	IBC Label	Controller	DMA Assignment	DMA Request Polarity
1	Channel 0	DMA1	Serial Port 3 Receiver (Z85230 Port A Rx)	High
2	Channel 1		Serial Port 3 Transmitter (Z85230 Port A Tx)	High
3	Channel 2		Reserved for Floppy Drive Controller	High
4	Channel 3		Parallel Port	High
5	Channel 4	DMA2	Not available - Cascaded from DMA1	N/A
6	Channel 5		Serial Port 4 Receiver (Z85230 Port B Rx)	High
7	Channel 6		Serial Port 4 Transmitter (Z85230 Port B Tx)	High
8	Channel 7		Not Used	High

Sources of Reset

There are six potential sources of reset on the MVME1603/MVME1604 SBC. They are:

1. Power-On Reset
2. RST (reset) Switch
3. ALT_RST_ function controlled by Port 92 Register in the IBC
4. Keyboard RESET function from the Keyboard Controller in the ISASIO device
5. Reset sources from the VMEchip2: VMEbus SYSRESET_, Watchdog Reset, and Software Reset Function.

When the MVME1603/MVME1604 is the VMEbus System Controller, an HRESET_ will also cause a VMEbus SYSRESET_.

Endian Issues

The MVME1603/MVME1604 SBC intends to support both little endian software (e.g. NT) and big-endian software (e.g. AIX). Since PowerPC processor is inherently big endian, PCI is inherently little-endian, and the VMEbus is big endian, things do get rather confusing. The following figures shows how the MVME1603/MVME1604 handles the endian issue in big-endian and little-endian modes:

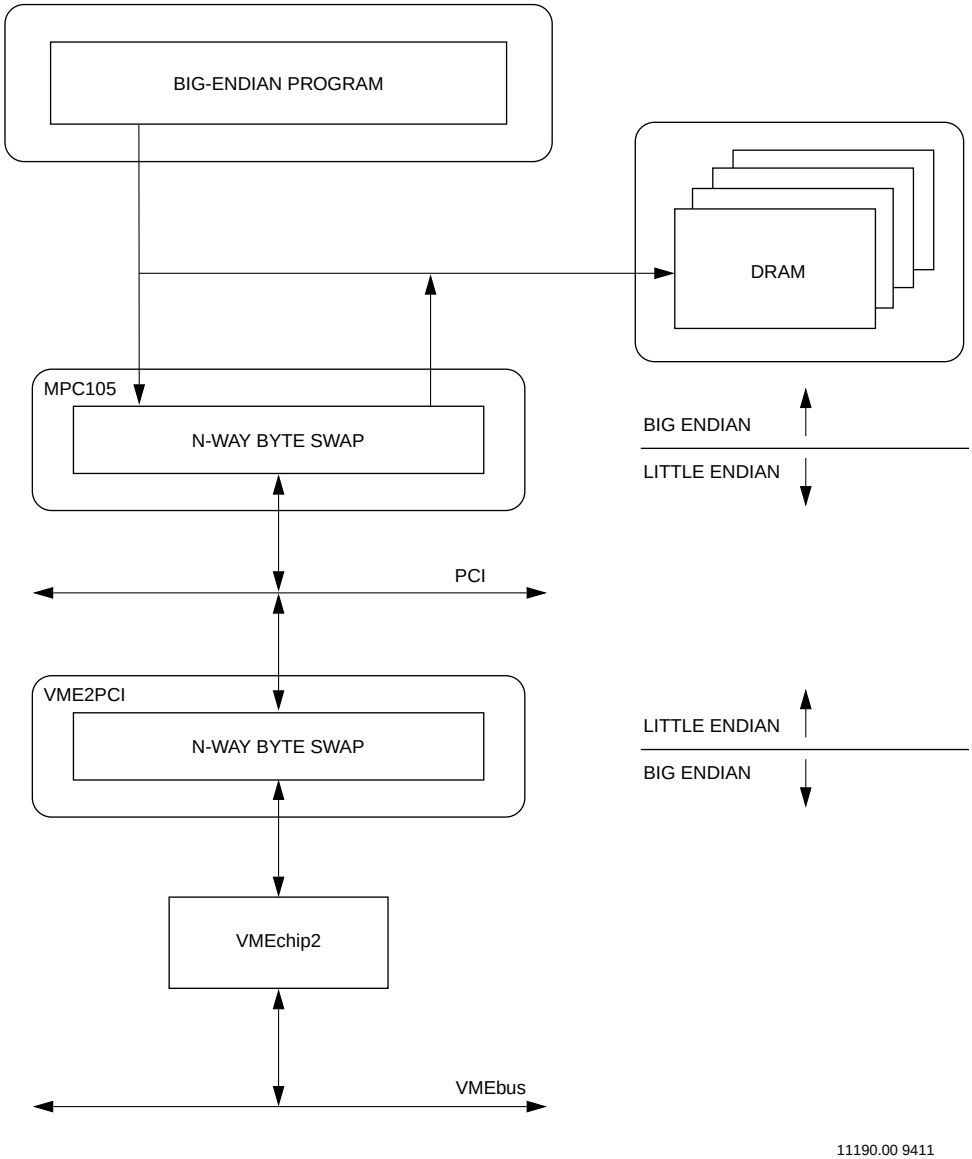


Figure 4-4. Big-Endian Mode

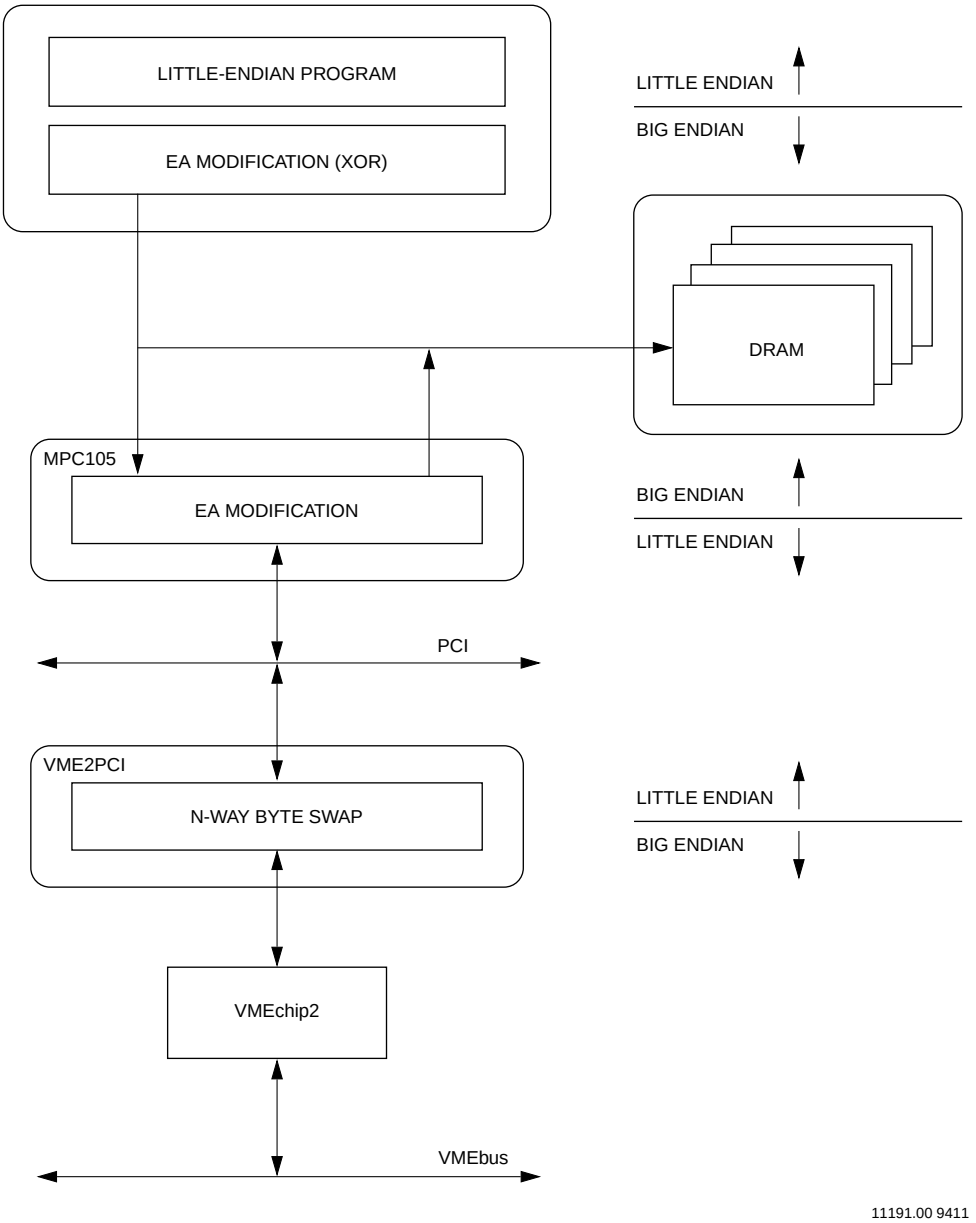


Figure 4-5. Little-Endian Mode

Processor/Memory Domain

The MPC603/604 processor can operate in both big-endian and little-endian mode. However, it always treats the external processor/memory bus as big-endian by performing *address rearrangement and reordering* when running in little-endian mode.

MPC105's Involvement

Because PCI is little-endian, the MPC105 performs byte swapping in both directions (from PCI to memory and from the processor to PCI) to maintain address invariance when it is programmed to operate in big-endian mode with the processor and the memory sub-system.

In little-endian mode, it *reverse-rearranges* the address for PCI-bound accesses and *rearranges* the address for memory-bound accesses (from PCI). In this case, no byte swapping is done.

PCI Domain

The PCI bus is inherently little-endian and all devices connected directly to PCI will operate in little-endian mode, regardless of the mode of operation in the processor's domain.

53C825 or 53C810(SCSI)

SCSI is byte stream oriented with the byte having the lowest address in memory being the first one to be transferred regardless of the endian mode. Since address invariance is maintained by the MPC105 in both little-endian and big-endian mode, there should be no endian issues for the SCSI data. Big-endian software must still however be aware of the byte-swapping effect when accessing the registers of the 53C825 or 53C810.

Am79C970 (Ethernet)

Ethernet is byte stream oriented with the byte having the lowest address in memory being the first one to be transferred regardless of the endian mode. Since address invariance is maintained by the MPC105 in both little-endian and big-endian mode, there should be no endian issues for the Ethernet data. Big-endian software must still however be aware of the byte-swapping effect when accessing the registers of the AM79C970.

GD5434 (Graphics)

The effects of byte swapping on big-endian software must be considered by big-endian software.

VME2PCI's Involvement

Because PCI is little-endian and the VMEbus is big-endian, the VME2PCI performs byte swapping in both directions (from PCI to VMEbus and from VMEbus to PCI) to maintain address invariance, regardless of the mode of operation in the processor's domain.

VMEbus Domain

The VMEbus is inherently big-endian and all devices connected directly to VMEbus are expected to operate in big-endian mode, regardless of the mode of operation in the processor's domain.

In big-endian mode byte-swapping is performed by the VME2PCI and then by the MPC105. The result has the desirable effect by being transparent to the big-endian software.

In little-endian mode, however, software must be aware of the byte-swapping effect from the VME2PCI and the address *reverse-rearranging* effect of the MPC105.

Related Documentation A

Motorola Computer Group Documents

The publications listed below are referenced in this document. If not shipped with this product, manuals may be purchased by contacting your local Motorola sales office.

Table A-1. Motorola Computer Group Documents

Document Title	Publication Number
MVME1603/MVME1604 Single Board Computer Installation and Use	V1600-1A/IH
MVME1603/MVME1604 Single Board Computer Programmer's Reference Guide	V1600-1A/PG
MVME1603/MVME1604 Support Information	V1600-1A/SC
PM603/PM604 Processor/Memory Mezzanine Module User's Manual	PM603A/UM
RAM104 DRAM Memory Module User's Manual	RAM104A/UM
PPCBug Debugging Package User's Manual (Parts 1 and 2)	PPCBUGA1/UM PPCBUGA2/UM
PPC1Bug Diagnostics Manual	PPC1DIAA/UM
MVME712M Transition Module and P2 Adapter Board User's Manual	MVME712M/D
MVME712-12, MVME712-13, MVME712A, MVME712AM, and MVME712B Transition Modules and LCP2 Adapter Board User's Manual	MVME712A/D
MVME760 Transition Module User's Manual	VME760A/UM
SIM705 Serial Interface Module Installation Guide	SIM705A/IH
Environmental Monitor User's Guide	ENVMONA/UG
Ultra 603/Ultra 604 Installation and Hardware User's Manual	ULMB60XA/IH
Ultra 603/Ultra 604 Programmer's Reference Guide	ULMB60XA/PG
Ultra 603/Ultra 604 Support Information	ULMB60XA/SC
PowerStack Series E System Installation Guide	SYSEIA/D

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

Manufacturers' Documents

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

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

Table A-2. Manufacturers' Documents

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

Table A-2. Manufacturers' Documents (Continued)

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

Table A-2. Manufacturers' Documents (Continued)

Document Title and Source	Publication Number
Alpine™ VGA Family - CL-GD543X Technical Reference Manual Third Edition Cirrus Logic, Inc. (or nearest Sales Office) 3100 West Warren Avenue Fremont, California 94538-6423 Telephone: (510) 623-8300 FAX: (510) 226-2180	GD543X-TRM-003 (part number 385439-003)
Am79C970 - PCnet™-PCI Single-Chip Ethernet Controller for PCI Local Bus Advance Micro Devices, Inc. 901 Thompson Place P.O. Box 3453 Sunnyvale, California 94088-3453 Applications Hotline and Literature Ordering Telephone: 1-800-222-9323	AMC79C970 (part number 18220, Rev B)
Am79C974 - PCnet™-SCSI Combination Ethernet and SCSI Controller for PCI Systems Advance Micro Devices, Inc. 901 Thompson Place P.O. Box 3453 Sunnyvale, California 94088-3453 Applications Hotline and Literature Ordering Telephone: 1-800-222-9323	18681, Rev A
DECchip 21040 Ethernet LAN Controller for PCI Hardware Reference Manual Digital Equipment Corporation Maynard, Massachusetts DECchip Information Line Telephone (United States and Canada): 1-800-332-2717 TTY (United States only): 1-800-332-2515 Telephone (outside North America): +1-508-568-6868	EC-N0752-72

Table A-2. Manufacturers' Documents (Continued)

Document Title and Source	Publication Number
PC87303VUL (Super I/O™ Sidewinder Lite) Floppy Disk Controller, Keyboard Controller, Real-Time Clock, Dual UARTs, IEEE 1284 Parallel Port, and IDE Interface National Semiconductor Corporation Customer Support Center (or nearest Sales Office) 2900 Semiconductor Drive P.O. Box 58090 Santa Clara, California 95052-8090 Telephone: 1-800-272-9959	PC87303VUL
PC87323VF (Super I/O™ Sidewinder) Floppy Disk Controller, Keyboard Controller, Real-Time Clock, Dual UARTs, IEEE 1284 Parallel Port, and IDE Interface National Semiconductor Corporation Customer Support Center (or nearest Sales Office) 2900 Semiconductor Drive P.O. Box 58090 Santa Clara, California 95052-8090 Telephone: 1-800-272-9959	PC87323VF
M48T18 CMOS 8K x 8 TIMEKEEPER™ SRAM Data Sheet SGS-Thomson Microelectronics Group Marketing Headquarters (or nearest Sales Office) 1000 East Bell Road Phoenix, Arizona 85022 Telephone: (602) 967-6100	M48T18
82378 System I/O (SIO) PCI-to-ISA Bridge Controller Intel Corporation Literature Sales P.O. Box 7641 Mt. Prospect, Illinois 60056-7641 Telephone: 1-800-548-4725	290473-003
NCR 53C8XX Family PCI-SCSI I/O Processors Programming Guide NCR Corporation Microelectronics Products Division 1635 Aeroplaza Drive Colorado Springs, Colorado 80916 Telephone: (719) 596-5795 NCR Hotline: 1-800-334-5454 FAX: (719) 527-8225	J10931I

Table A-2. Manufacturers' Documents (Continued)

Document Title and Source	Publication Number
SCC (Serial Communications Controller) User's Manual (for Z85230 and other Zilog parts) Zilog, Inc. 210 East Hacienda Ave., mail stop C1-0 Campbell, California 95008-6600 Telephone: (408) 370-8016 FAX: (408) 370-8056	DC-8293-02
Z8536 CIO Counter/Timer and Parallel I/O Unit Product Specification and User's Manual (in Z8000® Family of Products Data Book) Zilog, Inc. 210 East Hacienda Ave., mail stop C1-0 Campbell, California 95008-6600 Telephone: (408) 370-8016 FAX: (408) 370-8056	DC-8319-00
CS4231 Parallel Interface, Multimedia Audio Codec Data Sheet Crystal Semiconductor Corporation 4210 South Industrial Drive P.O. Box 17847 Austin, Texas 78744-7847 Telephone: 1-800-888-5016 Telephone: (512) 445-7222 FAX: (512) 445-7581	DS111PP4
CSB4231/4248 Evaluation Board Data Sheet Crystal Semiconductor Corporation 4210 South Industrial Drive P.O. Box 17847 Austin, Texas 78744-7847 Telephone: 1-800-888-5016 Telephone: (512) 445-7222 FAX: (512) 445-7581	DS111DB4
Award Classic KB42 Keyboard Controller Firmware for the National Semiconductor PC87323VUL-IAB SuperI/O™ Device Award Software International, Inc. Sales and Marketing 777 E. Middlefield Road Mountain View, California 94043 Telephone: (415) 968-4433	Award Classic KB42

Related Specifications

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

Table A-3. Related Specifications

Document Title and Source	Publication Number
ANSI Small Computer System Interface-2 (SCSI-2), Draft Document Global Engineering Documents P.O. Box 19539 Irvine, California 92713-9539 Telephone: 1-800-854-7179 or (714) 979-8135	X3.131.1990
Versatile Backplane Bus: VMEbus Institute of Electrical and Electronics Engineers, Inc. Publication and Sales Department 345 East 47th Street New York, New York 10017-21633 Telephone: 1-800-678-4333 OR Microprocessor system bus for 1 to 4 byte data Bureau Central de la Commission Electrotechnique Internationale 3, rue de Varembé Geneva, Switzerland	ANSI/IEEE Standard 1014-1987 IEC 821 BUS
IEEE - Common Mezzanine Card Specification Institute of Electrical and Electronics Engineers, Inc. Publication and Sales Department 345 East 47th Street New York, New York 10017-21633 Telephone: 1-800-678-4333	P1386 Draft 1.3
IEEE - PCI Mezzanine Card Specification (PMC) Institute of Electrical and Electronics Engineers, Inc. Publication and Sales Department 345 East 47th Street New York, New York 10017-21633 Telephone: 1-800-678-4333	P1386.1 Draft 1.3

Table A-3. Related Specifications (Continued)

Document Title and Source	Publication Number
Bidirectional Parallel Port Interface Specification Institute of Electrical and Electronics Engineers, Inc. Publication and Sales Department 345 East 47th Street New York, New York 10017-21633 Telephone: 1-800-678-4333	IEEE Standard 1284
Peripheral Component Interconnect (PCI) Local Bus Specification, Revision 2.0 PCI Special Interest Group P.O. Box 14070 Portland, Oregon 97214-4070 Marketing/Help Line Telephone: (503) 696-6111 Document/Specification Ordering Telephone: 1-800-433-5177or (503) 797-4207 FAX: (503) 234-6762	PCI Local Bus Specification
PowerPC Reference Platform (PRP) Specification, Third Edition, Version 1.0, Volumes I and II International Business Machines Corporation Power Personal Systems Architecture 11400 Burnet Rd. Austin, TX 78758-3493 Document/Specification Ordering Telephone: 1-800-PowerPC Telephone: 1-800-769-3772 Telephone: 708-296-9332	MPR-PPC-RPU-02

Glossary

Abbreviations, Acronyms, and Terms to Know

This section provides the descriptions of some of the abbreviations and acronyms used in this document, as well as a word definition for some of the key terms used in this document.

Table GL-1.Glossary of Terms

Term	Definition
10base-5	See thick Ethernet.
10base-2	See thin Ethernet.
10base-T	See twisted-pair Ethernet.
ACIA	**A**synchronous **C**ommunications **I**nterface **A**dapter
AIX	**A**dvanced **I**nteractive e**X**ecutive (**IBM** version of UNIX)
architecture	The main overall design in which each individual hardware component of the computer system is interrelated. The most common uses of this term are 8-bit, 16-bit, or 32-bit architectural design systems.
ASCII	**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange. This is a 7-bit code used to encode alphanumeric information. In the IBM-compatible world, this is expanded to 8-bits to encode a total of 256 alphanumeric and control characters.
ASIC	**A**pplication-**S**pecific **I**ntegrated **C**ircuit
AUI	**A**ttachment **U**nit **I**nterface
BBRAM	**B**attery **B**acked Up **R**andom **A**ccess **M**emory
bi-endian	Having big-endian and little-endian byte ordering capability.
big-endian	A byte-ordering method in memory where the address n of a word corresponds to the most significant byte. In an addressed memory word, the bytes are ordered (left to right) 0, 1, 2, 3, with 0 being the most significant byte.

Table GL-1.Glossary of Terms (Continued)

Term	Definition
BIOS	**B**asic **I**nput/**O**utput **S**ystem. This is the built-in program that controls the basic functions of communications between the processor and the I/O (peripherals) devices. Also referred to as ROM BIOS.
BitBLT	**Bit B**oundary B**L**ock **T**ransfer. A type of graphics drawing routine that moves a rectangle of data from one area of display memory to another. The data specifically need not have any particular alignment.
BLT	**BL**ock **T**ransfer
board	The term more commonly used to refer to a PCB (printed circuit board). Basically, a flat board made of nonconducting material, such as plastic or fiberglass, on which chips and other electronic components are mounted. Also referred to as a circuit board or card.
bpi	**b**its **p**er **i**nch
bps	**b**its **p**er **s**econd
bus	The pathway used to communicate between the CPU, memory, and various input/output devices, including floppy and hard disk drives. Available in various widths (8-, 16-, and 32-bit), with accompanying increases in speed.
cache	A high-speed memory that resides logically between a central processing unit (CPU) and the main memory. This temporary memory holds the data and/or instructions that the CPU is most likely to use over and over again and avoids accessing the slower hard or floppy disk drive.
CAS	**C**olumn **A**ddress **S**trobe. The clock signal used in dynamic RAMs to control the input of column addresses.
CD	**C**ompact **D**isc**.** A hard, round, flat portable storage unit that stores information digitally.
CD-ROM	**C**ompact **D**isk - **R**ead-**O**nly **M**emory

Table GL-1.Glossary of Terms (Continued)

Term	Definition
CISC	**C**omplex-**I**nstruction-**S**et **C**omputer**.** A computer whose processor is designed to sequentially run variable-length instructions, many of which require several clock cycles, that perform complex tasks and thereby simplify programming.
CODEC	**CO**der/**DEC**oder
Color **D**ifference (**CD**)	The signals of (R-Y) and (B-Y) without the luminance (-Y) signal. The Green signals (G-Y) can be extracted by these two signals.
Composite **V**ideo **S**ignal (**CVS**/**CVBS**)	Signal that carries video picture information for color, brightness and synchronizing signals for both horizontal and vertical scans. Sometimes referred to as “Baseband Video”.
cpi	**c**haracters **p**er **i**nch
cpl	**c**haracters **p**er **l**ine
CPU	**C**entral **P**rocessing **U**nit. The master computer unit in a system.
DCE	**D**ata **C**ircuit-terminating **E**quipment.
DLL	**D**ynamic **L**ink **L**ibrary. A set of functions that are linked to the referencing program at the time it is loaded into memory.
DMA	**D**irect **M**emory **A**ccess. A method by which a device may read or write to memory directly without processor intervention. DMA is typically used by block I/O devices.
DOS	**D**isk **O**perating **S**ystem
dpi	**d**ots **p**er **i**nch
DRAM	**D**ynamic **R**andom **A**ccess **M**emory. A memory technology that is characterized by extreme high density, low power, and low cost. It must be more or less continuously refreshed to avoid loss of data.
DTE	**D**ata **T**erminal **E**quipment.
ECC	**E**rror **C**orrection **C**ode

Table GL-1.Glossary of Terms (Continued)

Term	Definition
ECP	**E**xtended **C**apability **P**ort
EEPROM	**E**lectrically **E**rasable **P**rogrammable **R**ead-**O**nly **M**emory. A memory storage device that can be written repeatedly with no special erasure fixture. EEPROMs do not lose their contents when they are powered down.
EISA (bus)	**E**xtended **I**ndustry **S**tandard **A**rchitecture (bus) (**IBM**). An architectural system using a 32-bit bus that allows data to be transferred between peripherals in 32-bit chunks instead of 16-bit or 8-bit that most systems use. With the transfer of larger bits of information, the machine is able to perform much faster than the standard ISA bus system.
EPP	**E**nhanced **P**arallel **P**ort
EPROM	**E**rasable **P**rogrammable **R**ead-**O**nly **M**emory. A memory storage device that can be written once (per erasure cycle) and read many times.
ESCC	**E**nhanced **S**erial **C**ommunication **C**ontroller
ESD	**E**lectro-**S**tatic **D**ischarge/Damage
Ethernet	A local area network standard that uses radio frequency signals carried by coaxial cables.
FDC	**F**loppy **D**isk **C**ontroller
FDDI	**F**iber **D**istributed **D**ata **I**nterface. A network based on the use of optical-fiber cable to transmit data in non-return-to-zero, invert-on-1s (NRZI) format at speeds up to 100 Mbps.
FIFO	**F**irst-**I**n, **F**irst-**O**ut. A memory that can temporarily hold data so that the sending device can send data faster than the receiving device can accept it. The sending and receiving devices typically operate asynchronously.

Table GL-1.Glossary of Terms (Continued)

Term	Definition
firmware	The term, usually related to microprogramming, used to describe the program or specific software instructions that have been more or less permanently burned into an electronic component, such as a ROM (read-only memory) or an EPROM (erasable programmable read-only memory).
frame	One complete television picture frame consists of 525 horizontal lines with the NTSC system. One frame consists of two Fields.
graphics controller	On EGA and VGA, a section of circuitry that can provide hardware assist for graphics drawing algorithms by performing logical functions on data written to display memory.
HAL	**H**ardware **A**bstraction **L**ayer. The lower level hardware interface module of the Windows NT operating system. It contains platform specific functionality.
hardware	A computing system is normally spoken of as having two major components: hardware and software. Hardware is the term used to describe any of the physical embodiments of a computer system, with emphasis on the electronic circuits (the computer) and electromechanical devices (peripherals) that make up the system.
HCT	**H**ardware **C**onformance **T**est. A test used to ensure that both hardware and software conform to the Windows NT interface.
I/O	**I**nput/**O**utput
IBC	PCI/**I**SA **B**ridge **C**ontroller
IDE	**I**ntelligent **D**evice **E**xpansion
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers

Table GL-1.Glossary of Terms (Continued)

Term	Definition
interlaced	A graphics system in which the even scanlines are refreshed in one vertical cycle (field), and the odd scanlines are refreshed in another vertical cycle. The advantage is that the video bandwidth is roughly half that required for a non-interlaced system of the same resolution. This results in less costly hardware. It also may make it possible to display a resolution that would otherwise be impossible on given hardware. The disadvantage of an interlaced system is flicker, especially when displaying objects that are only a few scanlines high.
IQ Signals	Similar to the color difference signals (R-Y), (B-Y) but using different vector axis for encoding or decoding. Used by some USA TV and IC manufacturers for color decoding.
ISA (bus)	**I**ndustry **S**tandard **A**rchitecture (bus). The de facto standard system bus for IBM-compatible computers until the introduction of VESA and PCI. Used in the reference platform specification. (**IBM**)
ISASIO	**ISA S**uper **I**nput/**O**utput device
ISDN	**I**ntegrated **S**ervices **D**igital **N**etwork. A standard for digitally transmitting video, audio, and electronic data over public phone networks.
LAN	**L**ocal **A**rea **N**etwork
LED	**L**ight-**E**mitting **D**iode
little-endian	A byte-ordering method in memory where the address n of a word corresponds to the least significant byte. In an addressed memory word, the bytes are ordered (left to right) 3, 2, 1, 0, with 3 being the most significant byte.
MBLT	**M**ultiplexed **BL**ock **T**ransfer
MCA (bus)	**M**icro **C**hannel **A**rchitecture
MCG	**M**otorola **C**omputer **G**roup
MFM	**M**odified **F**requency **M**odulation

Table GL-1.Glossary of Terms (Continued)

Term	Definition
MIDI	**M**usical **I**nstrument **D**igital **I**nterface. The standard format for recording, storing, and playing digital music.
MPC	**M**ultimedia **P**ersonal **C**omputer
MPC105	The PowerPC-to-PCI bus bridge chip being developed by Motorola for the Ultra 603/Ultra 604 system board. It provides the necessary interface between the MPC603/MPC604 processor and the Boot ROM (secondary cache), the DRAM (system memory array), and the PCI bus.
MPC601	Motorola's component designation for the PowerPC 601 microprocessor.
MPC603	Motorola's component designation for the PowerPC 603 microprocessor.
MPC604	Motorola's component designation for the PowerPC 604 microprocessor.
MPU	**M**icro**P**rocessing **U**nit
MTBF	**M**ean **T**ime **B**etween **F**ailures. A statistical term relating to reliability as expressed in power on hours (poh). It was originally developed for the military and can be calculated several different ways, yielding substantially different results. The specification is based on a large number of samplings in one place, running continuously, and the rate at which failure occurs. MTBF is not representative of how long a device, or any individual device is likely to last, nor is it a warranty, but rather, of the relative reliability of a family of products.
multisession	The ability to record additional information, such as digitized photographs, on a CD-ROM after a prior recording session has ended.
non-interlaced	A video system in which every pixel is refreshed during every vertical scan. A non-interlaced system is normally more expensive than an interlaced system of the same resolution, and is usually said to have a more pleasing appearance.

Table GL-1.Glossary of Terms (Continued)

Term	Definition
nonvolatile memory	A memory in which the data content is maintained whether the power supply is connected or not.
NTSC	**N**ational **T**elevision **S**tandards **C**ommittee (USA)
NVRAM	**N**on-**V**olatile **R**andom **A**ccess **M**emory
OEM	**O**riginal **E**quipment **M**anufacturer
OMPAC	**O**ver - **M**olded **P**ad **A**rray **C**arrier
OS	**O**perating **S**ystem. The software that manages the computer resources, accesses files, and dispatches programs.
OTP	**O**ne **T**ime **P**rogrammable
palette	The range of colors available on the screen, not necessarily simultaneously. For VGA, this is either 16 or 256 simultaneous colors out of 262,144.
parallel port	A connector that can exchange data with an I/O device eight bits at a time. This port is more commonly used for the connection of a printer to a system.
PCI (local bus)	**P**eripheral **C**omponent **I**nterconnect (local bus) (**Intel**). A high-performance, 32-bit internal interconnect bus used for data transfer to peripheral controller components, such as those for audio, video, and graphics.
PCMCIA (bus)	**P**ersonal **C**omputer **M**emory **C**ard **I**nternational **A**ssociation (bus). A standard external interconnect bus which allows peripherals adhering to the standard to be plugged in and used without further system modification.
PDS	**P**rocessor **D**irect **S**lot
physical address	A binary address that refers to the actual location of information stored in secondary storage.
PIB	**P**CI-to-**I**SA **B**ridge

Table GL-1.Glossary of Terms (Continued)

Term	Definition
pixel	An acronym for picture element, and is also called a pel. A pixel is the smallest addressable graphic on a display screen. In RGB systems, the color of a pixel is defined by some Red intensity, some Green intensity, and some Blue intensity.
PLL	**P**hase-**L**ocked **L**oop
PMC	**P**CI **M**ezzanine **C**ard
POWER	**P**erformance **O**ptimized **W**ith **E**nhanced **R**ISC architecture (**IBM**)
PowerPC™	The trademark used to describe the **P**erformance **O**ptimized **W**ith **E**nhanced **R**ISC microprocessor architecture for **P**ersonal **C**omputers developed by the IBM Corporation. PowerPC is superscalar, which means it can handle more than one instruction per clock cycle. Instructions can be sent simultaneously to three types of independent execution units (branch units, fixed-point units, and floating-point units), where they can execute concurrently, but finish out of order. PowerPC is used by Motorola, Inc. under license from IBM.
PowerPC 601™	The first implementation of the PowerPC family of microprocessors. This CPU incorporates a memory management unit with a 256-entry buffer and a 32KB unified (instruction and data) cache. It provides a 64-bit data bus and a separate 32-bit address bus. PowerPC 601 is used by Motorola, Inc. under license from IBM.
PowerPC 603™	The second implementation of the PowerPC family of microprocessors. This CPU incorporates a memory management unit with a 64-entry buffer and an 8KB (instruction and data) cache. It provides a selectable 32-bit or 64-bit data bus and a separate 32-bit address bus. PowerPC 603 is used by Motorola, Inc. under license from IBM.
PowerPC 604™	The third implementation of the PowerPC family of microprocessors currently under development. PowerPC 604 is used by Motorola, Inc. under license from IBM.

Table GL-1.Glossary of Terms (Continued)

Term	Definition
PowerPC Reference Platform (**PRP**)	A specification published by the IBM Power Personal Systems Division which defines the devices, interfaces, and data formats that make up a PRP-compliant system using a PowerPC processor.
PowerStack™ RISC PC (System Board)	A PowerPC-based computer board platform currently in development by the Motorola Computer Group-Austin. It will be able to support Microsoft's Windows NT and IBM's AIX operating systems.
PRP	See PowerPC Reference Platform (PRP).
PRP-compliant	See PowerPC Reference Platform (PRP).
PRP Spec	See PowerPC Reference Platform (PRP).
PROM	**P**rogrammable **R**ead-**O**nly **M**emory
PS/2	**P**ersonal **S**ystem/**2** (**IBM**)
QFP	**Q**uad **F**lat **P**ackage
RAM	**R**andom-**A**ccess **M**emory. The temporary memory that a computer uses to hold the instructions and data currently being worked with. All data in RAM is lost when the computer is turned off.
RAS	**R**ow **A**ddress **S**trobe. A clock signal used in dynamic RAMs to control the input of the row addresses.
Reduced-Instruction-Set Computer (**RISC**)	A computer in which the processor's instruction set is limited to constant-length instructions that can usually be executed in a single clock cycle.
RFI	**R**adio **F**requency **I**nterference
RGB	The three separate color signals**: R**ed, **G**reen, and **B**lue. Used with color displays, an interface that uses these three color signals as opposed to an interface used with a monochrome display that requires only a single signal. Both digital and analog RGB interfaces exist.

Table GL-1.Glossary of Terms (Continued)

Term	Definition
RISC	See Reduced Instruction Set Computer (RISC).
ROM	**R**ead-**O**nly **M**emory
RTC	**R**eal-**T**ime **C**lock
SBC	**S**ingle **B**oard **C**omputer
SCSI	**S**mall **C**omputer **S**ystems **I**nterface. An industry-standard high-speed interface primarily used for secondary storage. SCSI-1 provides up to 5 Mbps data transfer.
SCSI-2 (Fast/Wide)	An improvement over plain SCSI; and includes command queuing. Fast SCSI provides 10 Mbps data transfer on an 8-bit bus. Wide SCSI provides up to 40 Mbps data transfer on a 16- or 32-bit bus.
serial port	A connector that can exchange data with an I/O device one bit at a time. It may operate synchronously or asynchronously, and may include start bits, stop bits, and/or parity.
SIM	**S**erial **I**nterface **M**odule
SIMM	**S**ingle **I**n-Line **M**emory **M**odule. A small circuit board with RAM chips (normally surface mounted) on it designed to fit into a standard slot.
SIO	**S**uper **I**/**O** controller
SMP	**S**ymmetric **M**ulti**P**rocessing. A computer architecture in which tasks are distributed among two or more local processors.
SMT	**S**urface **M**ount **T**echnology. A method of mounting devices (such as integrated circuits, resistors, capacitors, and others) on a printed circuit board, characterized by not requiring mounting holes. Rather, the devices are soldered to pads on the printed circuit board. Surface-mount devices are typically smaller than the equivalent through-hole devices.

Table GL-1.Glossary of Terms (Continued)

Term	Definition
software	A computing system is normally spoken of as having two major components: hardware and software. Software is the term used to describe any single program or group of programs, languages, operating procedures, and documentation of a computer system. Software is the real interface between the user and the computer.
SRAM	**S**tatic **R**andom **A**ccess **M**emory
SSBLT	**S**ource **S**ynchronous **BL**ock **T**ransfer
standard(s)	A set of detailed technical guidelines used as a means of establishing uniformity in an area of hardware or software development.
SVGA	**S**uper **V**ideo **G**raphics **A**rray (**IBM**). An improved VGA monitor standard that provides at least 256 simultaneous colors and a screen resolution of 800 x 600 pixels.
Teletext	One way broadcast of digital information. The digital information is injected in the broadcast TV signal, VBI, or full field, The transmission medium could be satellite, microwave, cable, etc. The display medium is a regular TV receiver.
thick Ethernet (10base-5)	An Ethernet in which the physical medium is a doubly shielded, 50-ohm coaxial cable capable of carrying data at 10 Mbps for a length of 500 meters (also referred to as thicknet).
thin Ethernet (10base-2)	An Ethernet in which the physical medium is a single-shielded, 50-ohm RG58A/U coaxial cable capable of carrying data at 10 Mbps for a length of 185 meters (also referred to as AUI or thinnet).
twisted-pair Ethernet (10base-T)	An Ethernet in which the physical medium is an unshielded pair of entwined wires capable of carrying data at 10 Mbps for a maximum distance of 185 meters.
UART	**U**niversal **A**synchronous **R**eceiver/**T**ransmitter
UV	**U**ltra**V**iolet

Table GL-1.Glossary of Terms (Continued)

Term	Definition
UVGA	**U**ltra **V**ideo **G**raphics **A**rray. An improved VGA monitor standard that provides at least 256 simultaneous colors and a screen resolution of 1024 x 768 pixels.
Vertical **B**lanking **I**nterval (**VBI**)	The time it takes the beam to fly back to the top of the screen in order to retrace the opposite field (odd or even). VBI is in the order of 20 TV lines.Teletext information is transmitted over 4 of these lines (lines 14-17).
VESA (bus)	**V**ideo **E**lectronics **S**tandards **A**ssociation (or VL bus). A internal interconnect standard for transferring video information to a computer display system.
VGA	**V**ideo **G**raphics **A**rray (**IBM**). The third and most common monitor standard used today. It provides up to 256 simultaneous colors and a screen resolution of 640 x 480 pixels.
virtual address	A binary address issued by a CPU that indirectly refers to the location of information in primary memory, such as main memory. When data is copied from disk to main memory, the physical address is changed to the virtual address.
VL bus	See **V**ESA **L**ocal **bus** (**VL bus**).
VMEchip2	MCG second generation VMEbus interface ASIC (**Motorola**)
VME2PCI	MCG ASIC that interfaces between the PCI bus and the VMEchip2 device
volatile memory	A memory in which the data content is lost when the power supply is disconnected.
VRAM	**V**ideo (Dynamic) **R**andom **A**ccess **M**emory. Memory chips with two ports, one used for random accesses and the other capable of serial accesses. Once the serial port has been initialized (with a transfer cycle), it can operate independently of the random port. This frees the random port for CPU accesses. The result of adding the serial port is a significantly reduced amount of interference from screen refresh. VRAMs cost more per bit than DRAMs.

Table GL-1.Glossary of Terms (Continued)

Term	Definition
Windows NT™	The trademark representing **Windows N**ew **T**echnology, the computer operating system developed by the Microsoft Corporation.
XGA	E**X**tended **G**raphics **A**rray. An improved IBM VGA monitor standard that provides at least 256 simultaneous colors and a screen resolution of 1024 x 768 pixels.
Y Signal	Luminance. This determines the brightness of each spot (pixel) on a CRT screen either color or B/W systems, but not the color.

Index

Symbols

+12 Vdc power 2-90

Numerics

53C825 or 53C810(SCSI) 4-13

A

A24 (standard) access cycles 2-33, 2-35
A32 (extended) access cycles 2-33, 2-35
abbreviations, acronyms, and terms to know GL-1
abort interrupt 2-80
ABORT switch
  interrupter 2-18
ABT (abort) interrupt 4-9
access timer
  VMEbus 2-8
ACFAIL interrupt 2-80
ACFAIL signal 2-89
adder 2-32, 2-34
adders, VMEchip2 2-27
address modifier codes 2-12, 2-41, 2-42, 2-43, 2-54
Address Modifier Select Register 2-32, 2-34
address offsets
  VMEchip2 1-20
Address Translation Address Register 2-29, 2-31, 2-40
Address Translation Select Register 2-30, 2-31, 2-41
addresses
  BBRAM configuration area 1-26
  GCSR 2-44
  location monitors 2-93
  M48T18 registers 1-25
  TOD clock 1-26
  VMEbus resources 2-35
  VMEchip2 LCSR 2-20
Am79C970 (Ethernet) 4-14
Application-Specific Integrated Circuit (ASIC) 1-1
arbiter time-out timer 2-60
arbitration
  algorithm 2-17
  mode, PRI 2-17
arbitration priority in fixed mode 4-3
assertion, definition 1-2
asterisk (*) 1-2
Attribute Register 2-41, 2-42, 2-43

B

BBRAM Configuration Area memory map 1-26
BBRAM/TOD clock memory map 1-25, 1-27
BBSY* time 2-91
BGIN filters 2-91
big-endian 1-2
big-endian mode 4-11
binary number 1-1
block access cycles 2-33, 2-35
block diagram
  MVME1603/MVME1604 1-4
  VMEchip2 2-5
block transfer cycles 2-12

block transfer mode 2-54
Board 1-15
board configuration register 1-15
Board Control Register 2-65
board description and memory maps 1-1
board failure 2-65, 2-66
board serial number 1-27
board status/control register, VMEchip2 2-99
BRDFAIL signal 2-65
Built-In-Self-Test register 3-12
bus requester, VMEchip2 2-8
bus timers 2-17
byte ordering 1-2
byte, definition 1-2

C

chip arbiter 2-17
clear MPU status bits 2-58
clear-on-compare 2-63, 2-64
clock prescaler 2-14, 2-62, 2-68
command packet, DMAC 2-49
configuration registers
  VME2PCI 3-3
control and status registers (CSR) 1-12
Control and Status Registers (CSRs)
  VMEchip2 2-20
control bit, definition 1-2
conventions, manual 1-1
counter, tick timer 1 2-64
counter, tick timer 2 2-65
CPU configuration register 1-13
CSR (control and status registers) 1-12

D

D16, D32 cycles 2-53
D32, D64 block transfer cycles 2-54
D64 block access cycles 2-33, 2-35
daisy-chain, IACK 2-17
data access cycles 2-32, 2-34
data transfers 2-41, 2-42, 2-43, 2-46, 2-53
decimal number 1-1
decoders 2-6, 2-10
  enable 2-45
  local bus to VMEbus map 2-35
  VMEchip2 2-26
description
  VMEchip2 2-4
description, functional 1-6
Device ID register 3-6
direct mapped PCI configuration space 1-8
direct mode 2-52
DMA channels 4-9
DMA Control Register 2-49
DMA Controller (DMAC) 2-11, 2-48
DMAC
  command table format 2-49
  done 2-59
  enable 2-52
  halt 2-52
  interrupt 2-73, 2-76, 2-79, 2-82
  interrupt counter 2-58
  interrupter 2-19
  registers 2-49
  requester 2-51
  VMEbus Address Counter 2-55
  VMEbus requester 2-13
DMAC Byte Counter 2-56
DMAC Control Register 1 (bits 0-7) 2-51
DMAC Control Register 2 (bits 0-7) 2-54
DMAC Control Register 2 (bits 8-15) 2-53
DMAC Local Bus Address Counter 2-55
DMAC Status Register 2-59
DMAC Ton/Toff Timers Control Register 2-60
don't release mode 2-51
double word, definition 1-2
DRAM Size Register 1-16
DRAM size register 1-16

E

early release, bus busy 2-92

edge-sensitive interrupt 2-69, 2-75, 2-77, 2-78, 2-79
endian issues 4-10
Ending Address Register 2-28, 2-29, 2-38, 2-39, 2-40
Ethernet
address 1-28
Ethernet (Am79C970) 4-14
external interrupt (parity error) 2-81
external interrupter 2-19

F

F page map decoder 2-47
fair mode 2-51, 2-52
false, definition 1-2
features
hardware 1-3
VMEchip2 2-1, 3-1
functional blocks, VMEchip2 2-4
functional description 1-6

G

GCSR 2-20
base address registers 2-35
board address 2-44
group address 2-44
LM interrupt 2-73, 2-75, 2-79, 2-84
programming model 2-93
SIG interrupt 2-73, 2-75, 2-79, 2-83
SIG3-0 interrupters 2-19
GD5434 (graphics) 4-14
General Purpose I/O (GPIO) pins 2-90
General Purpose Input (GPI) pins 2-91
General Purpose Register 0 2-99
General Purpose Register 1 2-100
General Purpose Register 2 2-100
General Purpose Register 3 2-101
General Purpose Register 4 2-101
General Purpose Register 5 2-102
Global Control and Status Registers 2-20
Global Control and Status Registers (GCSR) 2-93
global reset driver 2-18
global time-out period 2-60
glossary GL-1
glossary of terms GL-1
graphics (GD5434) 4-14

H

half-word, definition 1-2
handling VMEchip2 interrupts 4-8
handling Z8536 and Z85230 interrupts 4-8
header, software readable 1-14
hexadecimal character 1-1

I

I/O Control Register 1 2-89
I/O Control Register 2 2-90
I/O Control Register 3 2-91
IACK daisy-chain driver 2-17
IBC arbiter configuration diagram 4-2
IBC DMA channel assignments 4-9
IBC interrupt handler block diagram 4-6
IBC PCI/ISA interrupt assignments 4-7
ID register, VMEchip2 2-97
increment local bus address counter 2-53
increment VMEbus address counter 2-53
interpretation of MID3-MID0 1-19
interrupt base vectors 2-88
interrupt clear 2-57
Interrupt Clear Register (bits 16-23) 2-79
Interrupt Clear Register (bits 24-31) 2-78
Interrupt Clear Register (bits 8-15) 2-80
Interrupt Control and Status register 3-20
interrupt counter 2-58
interrupt handler 2-18, 2-19
VMEchip2 2-18
interrupt level 2-57
Interrupt Level Register 1 (bits 0-7) 2-82
Interrupt Level Register 1 (bits 16-23) 2-81
Interrupt Level Register 1 (bits 24-31) 2-80

Interrupt Level Register 1 (bits 8-15) 2-81
Interrupt Level Register 2 (bits 0-7) 2-84
Interrupt Level Register 2 (bits 16-23) 2-83
Interrupt Level Register 2 (bits 24-31) 2-82
Interrupt Level Register 2 (bits 8-15) 2-83
Interrupt Level Register 3 (bits 0-7) 2-86
Interrupt Level Register 3 (bits 16-23) 2-85
Interrupt Level Register 3 (bits 24-31) 2-84
Interrupt Level Register 3 (bits 8-15) 2-85
Interrupt Level Register 4 (bits 0-7) 2-88
Interrupt Level Register 4 (bits 16-23) 2-87
Interrupt Level Register 4 (bits 24-31) 2-86
Interrupt Level Register 4 (bits 8-15) 2-87
Interrupt Line register 3-14
interrupt register, VMEchip2 2-97
interrupt sources
  VMEchip2 2-18
interrupt status bits 2-72, 2-73, 2-74
interrupt supports 4-4
interrupter enable register 2-75, 2-76, 2-77
interrupter vector, VMEbus 2-58
interrupter, VMEbus 2-15
interrupters, local bus 2-69
interrupts 2-14, 2-53
  DMAC and interrupter acknowledge 2-82
  edge-sensitive 2-78
  GCSR LM 2-84
  GCSR SIG 2-83
  IRQ1 and parity error 2-81
  IRQ1,2 2-88
  IRQ5,6 2-87
  IRQ7 and spare 2-86
  software 2-78, 2-84
  SYSFAIL and master write post bus error 2-81
  tick timer 2-82
introduction 1-1, 4-1
  VMEchip2 2-1
IRQ
  enable interrupt 2-77
  function 2-57
  interrupt clear 2-57
  interrupt level 2-57
  interrupts 2-86, 2-87, 2-88
  status 2-57
ISA I/O space 1-9
ISA I/O space map 1-9

J

J14 1-14
J8 1-14

L

Large Scale Integration (LSI) 1-1
LCSR programming model 2-20
little-endian mode 4-12
LM cycles 2-98
local bus
  clock prescaler 2-69
  error 2-59
  interrupt filters 2-91
  interrupter summary 2-70
  interrupter, VMEchip2 2-69
  master 2-9, 2-11, 2-99
  reset 2-99
  slave 2-6
  slave map decoders 2-35
  to VMEbus DMAC 2-11
  to VMEbus interface 2-4
  to VMEbus requester 2-8
Local Bus Control Register 2-61
Local Bus Interrupter Enable Register (bits 0-7) 2-77
Local Bus Interrupter Enable Register (bits 16-23) 2-75

INDEX

Local Bus Interrupter Enable Register (bits 24-31) 2-74
Local Bus Interrupter Enable Register (bits 8-15 2-76
Local Bus Interrupter Status Register (bits 0-7) 2-74
Local Bus Interrupter Status Register (bits 16-23) 2-72
Local Bus Interrupter Status Register (bits 24-31) 2-72
Local Bus Interrupter Status Register (bits 8-15) 2-73
Local Bus Slave (VMEbus Master) Address Translation Address Register 4 2-40
Local Bus Slave (VMEbus Master) Address Translation Select Register 4 2-41
Local Bus Slave (VMEbus Master) Attribute Register 1 2-43
Local Bus Slave (VMEbus Master) Attribute Register 2 2-42
Local Bus Slave (VMEbus Master) Attribute Register 3 2-42
Local Bus Slave (VMEbus Master) Attribute Register 4 2-41
Local Bus Slave (VMEbus Master) Ending Address Register 1 2-38
Local Bus Slave (VMEbus Master) Ending Address Register 2 2-38
Local Bus Slave (VMEbus Master) Ending Address Register 3 2-39
Local Bus Slave (VMEbus Master) Ending Address Register 4 2-40
Local Bus Slave (VMEbus Master) Starting Address Register 1 2-38
Local Bus Slave (VMEbus Master) Starting Address Register 2 2-39
Local Bus Slave (VMEbus Master) Starting Address Register 3 2-39
Local Bus Slave (VMEbus Master) Starting Address Register 4 2-40
Local Bus to VMEbus Enable Control Register 2-45
Local Bus to VMEbus I/O Control Register 2-46
Local Bus to VMEbus Requester Control Register 2-50
Local Control and Status Register (LCSR) 2-8, 2-11
local reset 2-66
local reset driver 2-18
location monitors 2-93
  interrupters 2-19
  register, VMEchip2 2-97
LRESET signal 2-66

M

M48T18
  BBRAM 1-27
  BBRAM/TOD Clock memory map 1-25
  memory map 1-25
machine check interrupt (MCP_) 4-5
major featuresof VME2PCI ASIC 3-1
manual terminology 1-1, 1-2
manufacturers' documents A-2
map decoder 2-6, 2-10
  addresses 2-35
  enable 2-45
  enable register 2-45
  local bus slave 2-35
mask interrupts 2-89
maskable interrupts 4-5
master interrupt enable 2-69, 2-89
master write post bus error interrupt 2-81
MCP_ (machine check interrupt) 4-5
memory map
  BBRAM configuration area 1-26
  M48T18 registers 1-25
  PCI configuration space 1-8
  PCI view 1-12
  processor's view 1-7
  TOD clock 1-26

INDEX

VMEchip GCSR 1-25
VMEchip2 1-19
VMEchip2 LCSR 1-20, 2-22
memory map, ISA I/O space 1-9
memory maps 1-6
Miscellaneous Control Register 2-91
Motorola Computer Group documents A-1
MPC105's involvement 4-13
MPU
status register 2-58
MPU Status and DMA Interrupt Count Register 2-58
MVME1603/MVME1604 interrupt architecture 4-4

N

negation, definition 1-2
number of bytes of data 2-56
NVRAM 1-27

O

offboard status 2-59
overflow counter
tick timer 1 2-68
tick timer 2 2-67
overview of hardware 1-3

P

parity
error 2-59
error interrupt 2-81
PCI arbitration 4-1
PCI arbitration assignments 4-2
PCI Cache Line Size register 3-10
PCI Class Code register 3-9
PCI Command register 3-7
PCI configuration space map 1-8
PCI domain 4-13
PCI Header Type register 3-11
PCI Interrupt Pin register 3-15
PCI IO Map Base register 3-12
PCI Latency Timer register 3-11
PCI Maximum Latency register 3-16
PCI Memory Map Base register 3-13
PCI Minimum Grant register 3-15
PCI Slave Address Offset 1 register 3-17
PCI Slave Address Offset 2 register 3-19
PCI Slave Control 1 register 3-18
PCI Slave Control 2 register 3-20
PCI Slave End Address 1 register 3-17
PCI Slave End Address 2 register 3-19
PCI Slave Start Address 1 register 3-16
PCI Slave Start Address 2 register 3-18
PCI Status register 3-8
PCI view of the memory map 1-12
periodic interrupts 2-63
power-up reset status 2-65
prescaler 2-14
Prescaler Control Register 2-62
Prescaler Counter 2-68
priority arbitration mode (PRI) 2-17
processor view of the memory map 1-7
processor/memory domain 4-13
processor's point of view 3-22
program access cycles 2-33, 2-35
program address modifier code 2-46
programmable map decoders 2-6
programming
local bus interrupter 2-69
local bus to VMEbus map decoders 2-35
VMEbnus slave map decoders 2-26
VMEchip2 DMA controller 2-48
VMEchip2 GCSR 2-95
VMEchip2 tick and watchdog timers 2-59
programming details 4-1
programming model 1-6
VME2PCI 3-2
VMEchip2 GCSR 2-93
VMEchip2 LCSR 2-20
PROM Decoder, SRAM, and DMA Control Register 2-49

R

registers
- DMAC 2-48
- global control and status 2-20
- VMEchip2 GCSR 2-95
- VMEchip2 LCSR 2-20

related documentation A-1
related specifications A-7
release modes 2-50
release-on-request 2-50
release-when-done 2-50
request mode 2-50
reset drivers 2-18
reset local bus 2-99
RESET switch 2-65
Revision ID register 3-9
revision register, VMEchip2 2-97
ROM Control Register 2-47
round robin mode 2-51
Round-Robin-Select (RRS) arbitration mode 2-17

S

SCSI
- ID 1-28

SCSI (53C825 or 53C810) 4-13
serial port module ID 1-19
short I/O
- map 2-46
- map decoder 2-46
- space, address 2-35

SIG bits 2-98
single (SGL) arbitration mode 2-17
single word, definition 1-2
Slave Map Decoder Registers 2-26
slave map decoders, VMEchip2 2-26
Slave Write Post Control Register 2-32, 2-34
small-endian 1-2
snoop
- control register 2-32
- enable lines 2-34

Snoop Control Register 2-34
software
- interrupt 2-73, 2-76, 2-78, 2-80, 2-84, 2-85
- interrupters 2-19

Software Interrupt Set Register (bits 8-15) 2-78
software readable header 1-14
sources of reset 4-10
spare interrupt 2-86
SRAM, PROM Decoder, and DMA Control Register 2-49
starting address of command list 2-56
starting address of data 2-55
Starting Address Register 2-28, 2-29, 2-38, 2-39, 2-40
STAT LED 2-90
static RAM (SRAM) 2-49
status bit, definition 1-2
supervisor address modifier code 2-46
supervisory access cycles 2-33, 2-35
SYS fail interrupter 2-18
SYSFAIL interrupt 2-81
SYSFAIL signal 2-66, 2-89, 2-99
SYSRESET 2-15, 2-18
SYSRESET signal 2-66
system controller 2-66, 2-99
- VMEbus 2-16

system reset 2-66

T

Table Address Counter 2-56
Tick Timer 1 Compare Register 2-63
Tick Timer 1 Counter 2-64
Tick Timer 2 Compare Register 2-64
Tick Timer 2 Control Register 2-67
Tick Timer 2 Counter 2-65
tick timers
- interrupt 2-72, 2-75, 2-78, 2-82
- interrupters 2-19
- VMEchip2 2-14, 2-59

time-off period 2-61

time-on period 2-61
time-out
  timer 2-60
  watchdog 2-66
time-out period
  access 2-62
  global 2-60
  watchdog 2-61
timers
  tick and watchdog 2-14
  VMEbus 2-8
  watchdog 2-66
TOD clock memory map 1-25, 1-26
true, definition 1-2

U

user (non-privileged) access cycles 2-33, 2-35
user address modifier code 2-46

V

Vector Base Register (VBR) 2-71, 2-88
Vendor ID register 3-6
VME Access Control Register 2-61
VME2PCI 3-1
  configuration registers 3-3
  introduction 3-1
  major features 3-1
VME2PCI ASIC
  programming model 3-2
VME2PCI ASIC's IACK registers 3-22
VME2PCI ASIC's PCI configuration registers summary 3-5
VME2PCI IACK registers 3-21
VME2PCI's CSR mapping in PCI memory space 3-2
VME2PCI's CSR mMapping in PCI I/O space 3-3
VME2PCI's involvement 4-14
VMEbus
  AC fail interrupter 2-18
  ACFAIL interrupt 2-72, 2-75, 2-79
  acknowledge interrupt 2-82
  BERR 2-59
  grant time-out timer 2-60
  interrupter 2-15
  interrupter acknowledge interrupt 2-73, 2-76, 2-79, 2-82
  IRQ1 edge-sensitive interrupt 2-72, 2-75, 2-79, 2-81
  IRQ1 edge-sensitive interrupter 2-19
  IRQ1,2 interrupt 2-88
  IRQ1-7 interrupt 2-74, 2-77
  IRQ3,4 interrupt 2-87
  IRQ5,6 interrupt 2-87
  IRQ7 interrupt 2-86
  IRQ7-1 interrupters 2-19
  map decoder enable 2-45
  master 2-7, 2-11, 2-99
  master write post error interrupt 2-72, 2-75, 2-79
  release mode 2-52
  request level 2-50, 2-51
  requester 2-13
  slave 2-9
  slave map decoders 2-26
  SYSFAIL interrupt 2-72, 2-75, 2-79
  to local bus interface 2-9
VMEbus Arbiter Time-out Control Register 2-60
VMEbus domain 4-14
VMEbus Global Time-out Control Register 2-60
VMEbus Interrupter Control Register 2-56
VMEbus Interrupter Vector Register 2-57
VMEbus mapping 3-22
VMEbus mapping example 3-23
VMEbus point of view 3-24
VMEbus Slave Address Modifier Select Register 1 2-34
VMEbus Slave Address Modifier Select Register 2 2-32

VMEbus Slave Address Translation Address Offset Register 1 2-29
VMEbus Slave Address Translation Address Offset Register 2 2-31
VMEbus Slave Address Translation Select Register 1 2-30
VMEbus Slave Address Translation Select Register 2 2-31
VMEbus Slave Ending Address Register 1 2-28
VMEbus Slave Ending Address Register 2 2-29
VMEbus Slave GCSR Board Address Register 2-44
VMEbus Slave GCSR Group Address Register 2-44
VMEbus Slave Starting Address Register 2-29
VMEbus Slave Starting Address Register 1 2-28
VMEbus Slave Starting Address Register 2 2-29
VMEbus Slave Write Post and Snoop Control Register 1 2-34
VMEbus Slave Write Post and Snoop Control Register 2 2-32
VMEbus System Controller 2-16
VMEchip2
  block diagram 2-5
  features 2-1
  functional blocks 2-4
  GCSR programming model 2-93
  GCSR registers 2-95
  global control and status registers 2-20
  introduction 2-1
  LCSR programming model 2-20
  LCSR registers 2-20
  local bus interrupter and interrupt handler 2-18
  local bus to VMEbus DMA controller 2-11
  local bus to VMEbus interface 2-4
  memory map 1-20
  memory map, LCSR Summary 2-22
  programming GCSR 2-95
  programming local bus interrupter 2-69
  programming local bus to VMEbus map decoders 2-35
  programming tick and watchdog timers 2-59
  programming VMEbus slave map decoders 2-26
  programming VMEchip2 DMA controller 2-48
  tick and watchdog timers 2-14
  VMEboard functions 2-20
  VMEbus interrupter 2-15, 2-19
  VMEbus system controller 2-16
  VMEbus to local bus interface 2-9
VMEchip2 Board Status/Control Register 2-99
VMEchip2 CSR window 3-21
VMEchip2 ID Register 2-97
VMEchip2 LM/SIG Register 2-97
VMEchip2 Memory Map (GCSR Summary) 2-96
VMEchip2 memory maps 1-19
VMEchip2 Revision Register 2-97

W

Watchdog Time-out Control Register 2-61
watchdog time-out period 2-61
watchdog timer 2-14, 2-15, 2-59
Watchdog Timer Control Register 2-66
WDTO bits 2-15
word, definition 1-2
write post 2-34, 2-41, 2-42, 2-43, 2-46
  buffer 2-7, 2-9
  bus error interrupter 2-19
  enable/disable 2-32
  timer 2-8

write posting 2-7, 2-10, 2-37
    definition 2-7

Z

Z8536 CIO port pins assignment 1-18

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP