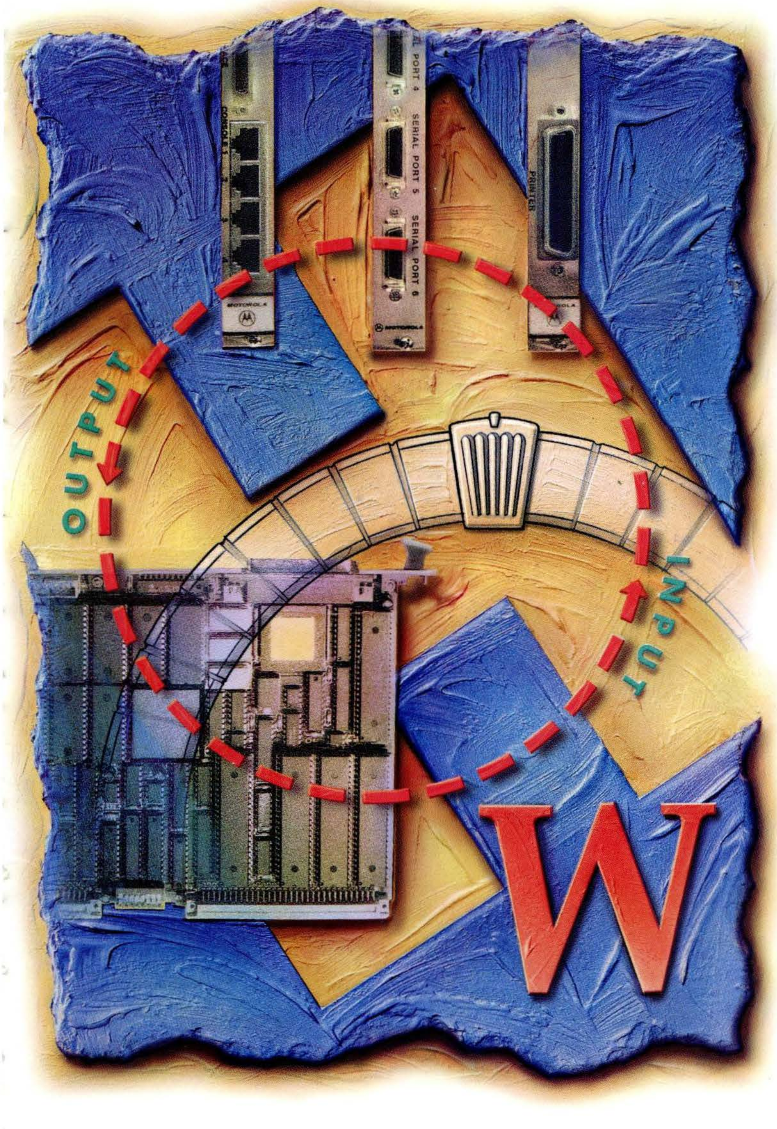
MOTOROLA

MVME166/167/187 Single Board Computers Programmer's Reference Guide



HARDWARE

MVME166/MVME167/MVME187

Single Board Computers

Programmer's Reference Guide

(MVME187PG/D3)

Notice

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

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

Restricted Rights Legend

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

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

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

PREFACE

This manual provides board level information and detailed ASIC chip information including register bit descriptions for the MVME166, MVME167, and MVME187 Single Board Computers. The information in this manual applies to the single board computers listed in the following table.

Single Board Computers Supported

MVME166 Models	MVME167 Models	MVME187 Models
MVME166-11A	MVME167-001B	MVME187-001B
MVME166-12A	MVME167-002B	MVME187-002B
MVME166-13A	MVME167-003B	MVME187-003B
MVME166-14A	MVME167-004B	MVME187-004B
MVME166-15A	MVME167-031B	MVME187-023B
MVME166-16A	MVME167-032B	MVME187-024B
	MVME167-033B	MVME187-031B
	MVME167-034B	MVME187-032B
	MVME167-035B	MVME187-033B
	MVME167-036B	MVME187-034B
		MVME187-035B
		MVME187-036B

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 the *Related Documentation* section in Chapter 1 of this manual.

WARNING

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the documentation for this product, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A Computing Device pursuant to Subpart J of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user, at the user's own expense, will be required to take whatever measures necessary to correct the interference.

Printed in the United States of America

May 1994

Contents

CHAPTER 1 BOARD LEVEL HARDWARE DESCRIPTION

CHAPTER 3 PCCchip2

CHAPTER 6 VSBchip2

CHAPTER 7 PRINTER AND SERIAL PORT CONNECTIONS

CHAPTER 8 USING INTERRUPTS ON THE MVME166/167/187

INDEX

LIST OF FIGURES

LIST OF TABLES

BOARD LEVEL HARDWARE DESCRIPTION 1

Introduction

This manual provides programming information for the MVME166, MVME167, and MVME187 Single Board Computers. Extensive programming information is provided for the Application-Specific Integrated Circuit (ASIC) devices used on the boards. Reference information is included for the Large Scale Integration (LSI) devices used on the boards and sources for additional information are provided.

This chapter describes the board level hardware features of the MVME166/167/187 Single Board Computers. The chapter is organized with a board level overview and features list in this introduction, followed by a more detailed hardware functional description. Front panel switches and indicators are included in the detailed hardware functional description. Memory maps are next, and the chapter closes with some general software considerations such as cache coherency, interrupts, and bus errors.

All programmable registers in the MVME166/167/187 that reside in ASICs are covered in the chapters on those ASICs. Chapter 2 covers the VMEchip2, Chapter 3 covers the PCCchip2, Chapter 4 covers the MEMC040 memory controller, Chapter 5 covers the MCECC memory controller, and Chapter 6 covers the VSBchip2. Chapter 7 covers certain printer and serial port connections. Appendix A details using interrupts. For those interested in programmable register bit definitions and less interested in hardware functionality, focus on Chapters 2, 3, 4, 5, and 6. In some cases, however, Chapter 1 gives related background information.

Overview

The MVME166/167 is based on the MC68040 microprocessor. The MVME187 is based on the M88000 RISC microprocessor. The MVME166 has 4/8/16/32/64/128/256 MB of ECC-protected DRAM.The MVME167/187 has 4/8/16/32/64/128/256 MB of ECC-protected DRAM or 4/8/16/32/64 MB of parity-protected DRAM. The MVME166/167/187 also has 8KB of static RAM and time of day clock (with battery backup), Ethernet transceiver interface, four serial ports with EIA-232-D interface (MVME167/187), four serial ports with TTL interface (MVME166), four tick timers, watchdog timer, four ROM sockets (MVME167/187), four Flash memory devices (MVME166), SCSI bus interface with DMA, Centronics printer port, VMEbus controller, 128KB of static RAM (with battery backup on MVME166, optionally available on MVME167/187), and a VSB interface (MVME166 only).

The I/O on the MVME167/187 is connected to the VMEbus P2 connector. The main board is connected through a P2 transition board and cables to the transition boards. The MVME167/187 supports the transition boards MVME712-12, MVME712-13, MVME712M, MVME712A, MVME712AM, and MVME712B (referred to in this manual as MVME712X, unless separately specified). The MVME712X transition boards provide configuration headers and provide industry standard connectors for the I/O devices.

The I/O connection for the MVME166 is provided by two high density shielded front panel I/O connectors. The SCSI bus is connected through a 68 pin connector. The printer, four serial ports and Ethernet interface are connected through a 100 pin connector. The MVME712-10 transition module and the MVME712-06/07/09 I/O distribution board set were designed to support the MVME166 boards. These transition boards provide configuration headers, serial port drivers and industry standard connectors for the I/O devices.

The VSB is connected to the P2 connector rows A and C on the MVME166.

The VMEbus interface is provided by an ASIC called the VMEchip2. The VMEchip2 includes two tick timers, a watchdog timer, programmable map decoders for the master and slave interfaces, and a VMEbus to/from local bus DMA controller, a VMEbus to/from local bus non-DMA programmed access interface, a VMEbus interrupter, a VMEbus system controller, a VMEbus interrupt handler, and a VMEbus requester.

Processor-to-VMEbus transfers can be D8, D16, or D32. VMEchip2 DMA transfers to the VMEbus, however, can be D16, D32, D16/BLT, D32/BLT, or D64/MBLT.

The PCCchip2 ASIC provides two tick timers and the interface to the LAN chip, SCSI chip, serial port chip, printer port, and BBRAM.

The MEMC040 memory controller ASIC provides the programmable interface for the parity-protected DRAM mezzanine board.

The MCECC memory controller ASIC provides the programmable interface for the ECC-protected DRAM mezzanine board.

The VSBchip2 provides the VSB interface on the MVME166. The VSBchip2 includes programmable map decoders for the master and slave interfaces, a VSB master interface, a VSB slave interface, a VSB interrupter, a VSB interrupt handler, a VSB serial requester, a VSB serial arbiter, and a VSB parallel requester.

Related Documentation

The following publications are applicable to the MVME166/167/187 and may provide additional helpful information. If not shipped with this product, they may be purchased by contacting your local Motorola sales office. Non-Motorola documents may be obtained from the sources listed.

Document Title	Motorola Publication Number
MVME166 Single Board Computer User's Manual	MVME166
MVME167 Single Board Computer User's Manual	MVME167
MVME167Bug Debugging Package User's Manual	MVME167BUG
Debugging Package for Motorola 68K CISC CPUs User's Manual	68KBUG
MVME187 RISC Single Board Computer User's Manual	MVME187
MVME187Bug Debugging Package User's Manual	MVME187BUG
Debugging Package for Motorola 88K RISC CPUs User's Manual	88KBUG
Single Board Computers SCSI Software User's Manual	SBCSCSI
MVME712-06/07/09 I/O Distribution Board Set User's Manual	MVME712IO
MVME712-10 Transition Module User's Manual	MVME712-10
MVME712M Transition Module and P2 Adapter Board User's Manual	MVME712M
MVME712-12, MVME712-13, MVME712A, MVME712AM, and MVME712B Transition Modules and LCP2 Adapter Board User's Manual	MVME712A
MC88100 RISC Microprocessor User's Manual	MC88100UM
MC88200 Cache/Memory Management Unit (CMMU) User's Manual	MC88200UM
M68040 Microprocessors User's Manual	M68040UM
M68000 Family Reference Manual	M68000FR

Note

Although not shown in the above list, each Motorola Computer Group manual publication number is suffixed with characters which represent the revision level of the document, such as "/D2" (the second revision of a manual); a supplement bears the same number as a manual but has a suffix such as "/D2A1" (the first supplement to the second edition of the manual).

To assist your development effort, Motorola has collected technical literature for each of the peripheral controllers used on the MVME166, MVME167, and MVME187 from the suppliers. This bundle, which can be ordered as part number **68-1X7DS**, includes the following:

NCR 53C710 SCSI Controller Data Manual and Programmer's Guide
Intel i82596 Ethernet Controller User's Manual
Cirrus Logic CD2401 Serial Controller User's Manual
SGS-Thompson MK48T08 NVRAM/TOD Clock Data Sheet
Zilog Z85230 Serial Communications Controller Data Sheet
Intel i28F008 andi28F020 Flash Memory Data Sheets
Intel i28F008SA Software Drivers and Automation and Algorithms Application Notes
Motorola MC68230 Parallel Interface Timer Data Sheet

The following publications are available from the sources indicated:

Versatile Backplane Bus: VMEbus, ANSI/IEEE Std 1014-1987, The Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, New York, NY 10017 (VMEbus Specification). (This is also *Microprocessor System Bus for 1 to 4 Byte Data,* IEC 821 BUS, Bureau Central de la Commission Electrotechnique Internationale; 3,rue de Varembé, Geneva, Switzerland.)

IEEE Standard for Multiplexed High-Performance Bus Structure: VSB, ANSI/IEEE Std 1096-1988, The Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, New York, NY 10017 (VSB Specification). (This is also *Parallel Sub-system Bus of the IEC 821 VMEbus,* IEC 822 VSB, Bureau Central de la Commission Electrotechnique Internationale; 3,rue de Varembé, Geneva, Switzerland.)

ANSI Small Computer System Interface-2 (SCSI-2), Draft Document X3.131-198X, Revision 10c; Global Engineering Documents, P.O. Box 19539, Irvine, CA 92714.

CL-CD2400/2401 Four-Channel Multi-Protocol Communications Controller Data Sheet, order number 542400-003; Cirrus Logic, Inc., 3100 West Warren Ave., Fremont, CA 94538.

82596CA Local Area Network Coprocessor Data Sheet, order number 290218; and *82596 User's Manual,* order number 296853; Intel Corporation, Literature Sales, P.O. Box 58130, Santa Clara, CA 95052-8130.

NCR 53C710 SCSI I/O Processor Data Manual, order number NCR53C710DM, and *NCR 53C710 SCSI I/O Processor Programmer's Guide,* order number NCR53C710PG; NCR Corporation, Microelectronics Products Division, Colorado Springs, CO.

MK48T08(B) Timekeeper™ and 8Kx8 Zeropower™ RAM data sheet in *Static RAMs Databook,* order number DBSRAM71; SGS-THOMPSON Microelectronics Group; North & South American Marketing Headquarters, 1000 East Bell Road, Phoenix, AZ 85022-2699.

i28F008 Flash Memory Data Sheet, order number 290435, *i28F020 Flash Memory Data Sheet,* order number 290245, *i28F008SA Software Drivers Application Note,* order number 292095, and *i28F008SA Automation and Algorithms Application Note,* order number 292099; Intel Literature Sales, P.O. Box 7641, Mt. Prospect, IL 60056-7641.

MC68230 Parallel Interface Timer (PI/T) Data Sheet, order number MC68230/D, Motorola Semiconductor Products, Inc., LDC, Broadway Bldg. BB100, P.O. Box 20924, Phoenix, AZ 85036-0924.

Z85230 Serial Communications Controller Data Sheet, order number DC-8293-02, Zilog Inc., 210 East Hacienda Drive, Campbell, CA 95008-6600.

Requirements

These boards are designed to conform to the requirements of the following documents:

❑ VMEbus Specification (IEEE 1014-87)

❑ EIA-232-D Serial Interface Specification, EIA

❑ SCSI Specification, ANSI

❑ VSB Specification (IEEE 1096-1988) (MVME166 only)

Features

❑ MC68040 Microprocessor (MVME166/167)

❑ M88000 Microprocessor (MVME187)

❑ 4/8/16/32/64/128/256MB of 32-bit DRAM; with ECC protection up to 256MB (MVME166/167/187), or parity protection up to 64MB (MVME167/187).

❑ Four 44-pin PLCC ROM sockets (two 32-bit wide banks) (MVME167/187)

❑ Four Flash memory devices and a download EPROM (MVME166 only)

❑ 128KB SRAM with battery backup (MVME166; battery backup is optional on MVME167/187) .

❑ Status LEDs for FAIL, STAT, RUN, SCON, LAN, +12V (LAN power), SCSI, and VME (MVME167/187)

- ❑ Status LEDs for FAIL, STAT, RUN, SCON, LAN, RPWR, SCSI, VME, TPWR and VSB (MVME166)
- ❑ 8K by 8 RAM and time of day clock with battery backup
- ❑ RESET and ABORT switches
- ❑ Four 32-bit tick timers for periodic interrupts
- ❑ Watchdog timer
- ❑ Eight software interrupts
- ❑ I/O
  - SCSI Bus interface with DMA
  - Four serial ports with EIA-232-D buffers with DMA (MVME167/187)
  - Four serial ports with TTL buffers (MVME166)
  - Centronics printer port
  - Ethernet transceiver interface with DMA
- ❑ VMEbus interface
  - VMEbus system controller functions
  - VMEbus to local bus interface (A24/A32, D8/D16/D32 (D8/D16/D32/D64 BLT) (BLT = Block Transfer)
  - 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 (D16/D32/D64 BLT)
- ❑ VSB interface (MVME166 only)
  - Local bus to VSB interface (A16/A24/A32, D8/D16/D32)
  - VSB to local bus interface (A16/A24/A32, D8/D16/D32)
  - Control and Status Register sets (Board CSRs accessible from both local bus and VSB; local CSRs accessible from local bus) (Includes Global CSR for IPC (General Purpose Registers 1 and 2))
  - Local bus interrupter
  - VSB interrupter and VSB interrupt handler
  - Bidirectional write posting - local bus to VSB and VSB to local bus
  - EVSB compatible

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 *two-byte* is 16 bits, numbered 0 through 15, with bit 0 being the least significant. For the MVME166, MVME167, and other CISC boards, this is called a *word*. For the MVME187 and other RISC boards, this is called a *half-word.*
- ❏ A *four-byte* is 32 bits, numbered 0 through 31, with bit 0 being the least significant. For the MVME166, MVME167, and other CISC boards, this is called a *longword*. For the MVME187 and other RISC boards, this is called a *word.*

Throughout this manual, it is assumed that the MPU on the MVME187 always programs the CMMUs with *big-endian* byte ordering, as shown below. Any attempt to use small-endian byte ordering immediately renders the MVME187Bug debugger unusable.

```
BIT                                             BIT
31        24  23         16  15        08  07   00
+------------+------------+------------+------------+
|    ADR0    |    ADR1    |    ADR2    |    ADR3    |
+------------+------------+------------+------------+
```

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.

Block Diagram

Figure 1-1 is a general block diagram of the MVME166/167/187.

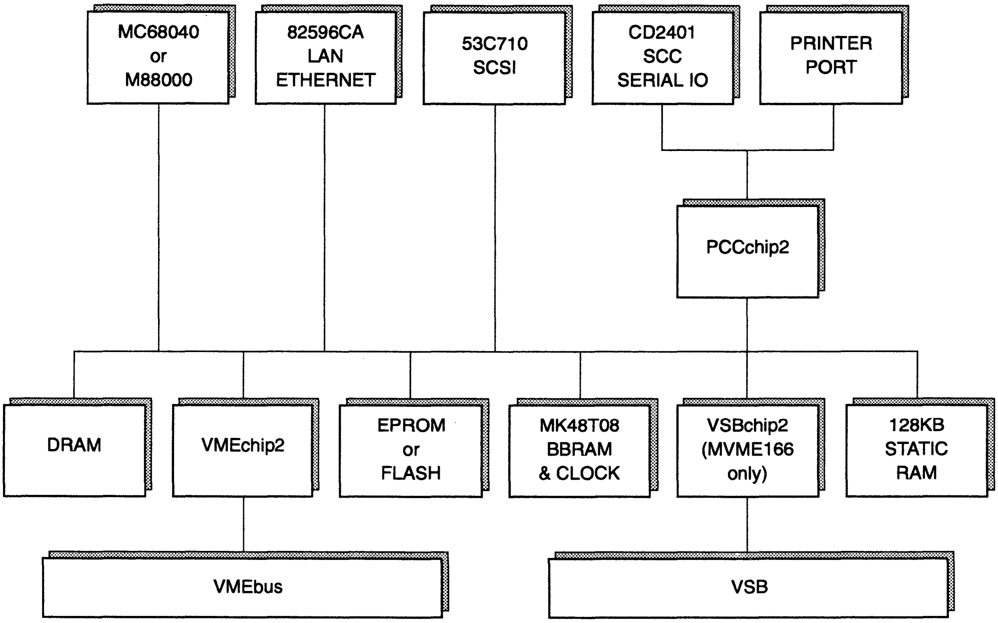


Figure 1-1. MVME166/167/187 Block Diagram

Functional Description

This section contains a functional description of the major blocks on the MVME166/167/187 Single Board Computers.

Front Panel Switches and Indicators

The front panel of the MVME166/167/187 contains two switches and several LEDs, as listed in the following tables. The MVME166 has ten LEDs. The MVME167 and MVME187 each have eight LEDs.

Table 1-1. Front Panel Switches

Switch Name	Description
RESET	The RESET switch resets all onboard devices and drives SYSRESET* if the board is system controller. The RESET switch may be disabled by software. Refer to the VMEchip2 description in Chapter 2.
ABORT	When enabled by software, the ABORT switch generates an interrupt at a user-programmable level. It is normally used to abort program execution and return to the debugger. Refer to the VMEchip2 description in Chapter 2 for more information.

Table 1-2. Front Panel LEDs

LED Name	Color	VMEmodule	Description
FAIL	Red	MVME166/167/187	The red FAIL LED (part of DS1) lights when the BRDFAIL signal line is active.
STAT	Yellow	MVME166/167	The MC68040 status lines are decoded, on the MVME166/167 to drive the yellow STAT (status) LED (part of DS1). In this case, a halt condition from the processor lights the LED.
		MVME187	The STAT LED is controlled by software on the MVME187.
RUN	Green	MVME166/167/187	The green RUN LED (part of DS2) lights when the local bus TIP* signal line is low. This indicates one of the local bus masters is executing a local bus cycle.
SCON	Green	MVME166/167/187	The green SCON LED (part of DS2) lights when the VMEchip2 is the VMEbus system controller.
LAN	Green	MVME166/167/187	The green LAN LED (part of DS3) lights when the LAN chip is local bus master.

Table 1-2. Front Panel LEDs (Continued)

LED Name	Color	VMEmodule	Description
RPWR	Green	MVME166	The MVME166 supplies +5V, +12V, and -12V power to the transition board through fuses. There is one fuse for each voltage. The green RPWR (remote power) LED (part of DS3) lights when all three voltages are available to the transition board interface.
+12V	Green	MVME167/187	The MVME167/187 supplies +12V power to Ethernet transceiver interface through a fuse. The green +12V (LAN power) LED (part of DS3) lights when power is available to the transceiver interface.
SCSI	Green	MVME166/167/187	The green SCSI LED (part of DS4) lights when the SCSI chip is local bus master.
VME	Green	MVME166/167/187	The green VME LED (part of DS4) lights when the board is using the VMEbus (VMEbus AS* is asserted by the VMEchip2) or when the board is accessed by the VMEbus (VMEchip2 is the local bus master).
TPWR	Green	MVME166	The MVME166 supplies +5V to the SCSI bus for terminator power through a fuse. The green TPWR (term power) LED (part of DS5) lights when TERMPWR is available to the SCSI bus. SCSI bus TERMPWR may be supplied by other devices on the SCSI bus.
VSB	Green	MVME166	The green VSB LED (part of DS5) lights when the MVME166 is using the VSB (VSB PAS* is asserted by the VSBchip2) or when the MVME166 is accessed by the VSB (VSBchip2 is the local bus master).

Data Bus Structure

The local bus on the MVME166/167/187 is a 32-bit synchronous bus that is based on the MC68040 bus, and supports burst transfers and cache coherency snooping (MVME166/167 only). The various local bus master and slave devices use the local bus to communicate. The local bus is arbitrated by priority type arbiter and the priority of the local bus masters from highest to lowest is: 82596CA LAN, CD2401 serial (through the PCCchip2), 53C710 SCSI, VSB (MVME166 only), VMEbus, and MPU. In the general case, any master can access any slave; however, not all combinations pass the common sense test.

Refer to the specific section of this manual and to the user's guide for each device to determine its port size, data bus connection, and any restrictions that apply when accessing the device.

MC68040 MPU

The MC68040 processor is used on the MVME166/167. The MC68040 has on-chip instruction and data caches and a floating point processor. Refer to the M68040 user's manual for more information.

M88000 MPU

The MVME187 is based on the M88000 family and uses one MC88100 MPU and two MC88200 or MC88204 CMMUs. One CMMU is used for the data cache and one is used for the instruction cache. Refer to the MC88100 and MC88200 user's manuals for more information.

EPROM

There are four 44-pin PLCC/CLCC EPROM sockets for 27C102JK or 27C202JK type EPROMs. (The MVME166 has Flash memories.) They are organized as two 32-bit wide banks that support 8-, 16-, and 32-bit read accesses. The EPROMs are mapped to local bus address 0 following a local bus reset. This allows the MC68040 to access the stack pointer and execution address following a reset. It also allows the MC88100 to start executing code at address 0 following a reset. The EPROMs are controlled by the VMEchip2. The map decoder, access time, and when they appear at address 0, is programmable. Refer to the VMEchip2 in Chapter 2 for more detail.

Flash Memory and Download EPROM

The MVME166 includes four 28F020 Flash memory devices and a download EPROM. These parts replace the four EPROM sockets used on the MVME167/187. The Flash parts are programmable on the MVME166 board and the programming code is provided in the download EPROM. The Flash devices provide 1 MB of ROM at address $FF800000-$FF8FFFFF. The download EPROM provides 128 KB of ROM at $FFF80000-$FFF9FFFF. The download EPROM is mapped to local bus address 0 following a local bus reset. This allows the MC68040 to access the stack pointer and execution address following a reset. The download EPROM appears at 0 until the DR0 bit is cleared in the PCCchip2 chip. The Flash devices are controlled by the VMEchip2 and the download EPROM is controlled by the PCCchip2. The PC0 bit in the MC68230 PI/T chip must be low to enable writes to Flash.

The EPROM contains the BootBug product (166BBug). Because Flash memory can be electronically erased, the EPROM firmware is a subset of the regular debugger product. It contains enough functionality from the debugger to permit downloading of object code (via VMEbus, serial port, SCSI bus, or the network) and reprogramming of the Flash memory.

A jumper on the MVME166 (J3, pins 7 and 8) controls the operation of the BootBug. If the jumper is in place, the BootBug (which always executes at power-up and reset) passes execution to the full debugger contained in Flash memory. If the jumper is removed, execution continues (with diminished functionality) in the BootBug.

Before you perform any SCSI, VMEbus, or Ethernet I/O with the MVME166, it may be necessary to define some parameters (e.g., SCSI ID, Ethernet address, VMEbus mapping). For details on configuring the MVME166, refer to the **setup** command description in the *MVME167Bug Debugging Package User's Manual.*

SRAM

The boards include 128KB of 32-bit wide static RAM that supports 8-, 16-, and 32-bit wide accesses. The SRAM allows the debugger to operate and limited diagnostics to be executed without the DRAM mezzanine. The SRAM is controlled by the VMEchip2, and the access time is programmable. Refer to the VMEchip2 in Chapter 2 for more detail. The boards are populated with 100 ns SRAMs.

The SRAM is also battery backed up on the MVME166, and battery backup is optionally available on the MVME167 and MVME187. The battery backup function is provided by a Dallas DS1210S. On the MVME166 and MVME187, the DS1210S supports primary and secondary power sources; when the main board power fails, the DS1210S selects the source with the highest voltage. If one source should fail, the DS1210S switches to the redundant source. Only one backup power source is supported on the MVME167.

Each time the MVME166/167/187 is powered, the DS1210S checks the power source(s) and if the voltage of the backup source(s) is less than two volts, the second memory cycle is blocked. This allows software to provide an early warning to avoid data loss. Because the DS1210S may block the second access, the software should do at least two accesses before relying on the data.

The MVME166 (and optionally, the MVME187) provides jumpers that allow either power source of the DS1210S to be connected to the VMEbus +5 V STDBY pin or one cell of the onboard battery. For example, the primary system backup source may be a battery connected to the VMEbus +5 V STDBY pin and the secondary source may be the onboard battery. If the system source should fail or the board is removed from the chassis, the onboard battery takes over. On the MVME167, only the +5V standby or the optional onboard battery is jumper selectable. The jumpers for each board are described in the *Configuration Jumpers, MVME1xx* sections later in this chapter.

The optional power source for the MVME167 SRAM is a socketed Sanyo CR2430 battery. The power source for the MVME166 (onboard) and the MVME187 (optional) is a RAYOVAC FB1225 battery which has two BR1225 type lithium cells and is socketed for easy removal and replacement. Small capacitors are provided to allow the batteries to be quickly replaced without data loss.

The lifetime of the batteries is very dependent on the ambient temperature of the board and the power-on duty cycle. The FB1225 and CR2430 lithium batteries should provide at least two years of backup time with the board powered off and the board at 40° C. If the power-on duty cycle is 50% (the board is powered on half of the time), the battery lifetime is four years. At lower ambient temperatures the backup time is greatly extended and may approach the shelf life of the battery.

When a board is stored, the battery should be disconnected to prolong battery life. This is especially important at high ambient temperatures. MVME166/167/187 boards with battery backup are shipped with the batteries disconnected.

The power leads from the battery are exposed on the solder side of the board, therefore the board should not be placed on a conductive surface or stored in a conductive bag unless the battery is removed.

WARNING

Lithium batteries incorporate inflammable materials such as lithium and organic solvents. If lithium batteries are mistreated or handled incorrectly, they may burst open and ignite, possibly resulting in injury and/or fire. When dealing with lithium batteries, carefully follow the precautions listed below in order to prevent accidents.

- ❏ Do not short circuit.
- ❏ Do not disassemble, deform, or apply excessive pressure.
- ❏ Do not heat or incinerate.
- ❏ Do not apply solder directly.
- ❏ Do not use different models, or new and old batteries together.

❑ Do not charge.

❑ Always check proper polarity.

To remove the battery from the module, carefully pull the battery from the socket.

Onboard DRAM

The MVME166/167/187 onboard DRAM is located on mezzanine boards. The mezzanine boards are available in different sizes and with ECC protection (MVME166) or with parity protection *or* ECC protection (MVME167/MVME187). Mezzanine board sizes are 4, 8, 16, or 32MB (with parity protection), or 4, 8, 16, 32, or 128 (with ECC protection); two mezzanine boards may be stacked to provide up to 256MB of onboard RAM. The main board and a single mezzanine board together take one slot. The stacked configuration requires two VMEboard slots. Motorola software *does* support mixed parity and ECC memory boards on the same main board. The DRAM is four-way interleaved to efficiently support cache burst cycles. The parity mezzanines are only supported on 25 MHz main boards.

The DRAM map decoder can be programmed to accommodate different base address(es) and sizes of mezzanine boards. The onboard DRAM is disabled by a local bus reset and must be programmed before the DRAM can be accessed. Refer to the MEMC040 in Chapter 4 for detailed programming information on the parity board. Refer to the MCECC in Chapter 5 for detailed programming information on the ECC board. Most DRAM devices require some number of access cycles before the DRAMs are fully operational. Normally this requirement is met by the onboard refresh circuitry and normal DRAM initialization. However, software should insure a minimum of 10 initialization cycles are performed to each bank of RAM.

Battery Backed Up RAM and Clock

The MK48T08 RAM and clock chip is used on the MVME166/167/187. This chip provides a time of day clock, oscillator, crystal, power fail detection, memory write protection, 8KB of RAM, and a battery in one 28-pin package. The clock provides seconds, minutes, hours, day, date, month, and year in BCD 24-hour format. Corrections for 28-, 29- (leap year), and 30-day months are automatically made. No interrupts are generated by the clock. The MK48T08 is an 8 bit device; however, the interface provided by the PCCchip2 supports 8-, 16-, and 32-bit accesses to the MK48T08. Refer to the PCCchip2 in Chapter 3 and to the MK48T08 data sheet for detailed programming and battery life information.

VMEbus Interface

The local bus to VMEbus interface, the VMEbus to local bus interface, and the local-VMEbus DMA controller functions on the MVME166/167/187 are provided by the VMEchip2. The VMEchip2 can also provide the VMEbus system controller functions. Refer to the VMEchip2 in Chapter 2 for detailed programming information.

VME Subsystem Bus (VSB) Interface

The local bus to VSB interface and the VSB to local bus interface are provided by the VSBchip2, only on the MVME166 board. The VSB uses the P2 connector of the MVME166. Refer to Chapter 6 for detailed programming of the VSBchip2.

I/O Interfaces

The MVME166/167/187 provides onboard I/O for many system applications. The I/O functions include serial ports, printer port, Ethernet transceiver interface, and SCSI mass storage interface.

Serial Port Interface

The CD2401 serial controller chip (SCC) is used to implement the four serial ports. The serial ports support the standard baud rates (110 to 38.4K baud). The four serial ports on the MVME167/187 are different functionally because of the limited number of pins on the P2 I/O connector. Serial port 1 is a minimum function asynchronous port. It uses RXD, CTS, TXD, and RTS. Serial ports 2 and 3 are full function asynchronous ports. They use RXD, CTS, DCD, TXD, RTS, and DTR. Serial port 4 is a full function asynchronous or synchronous port. It can operate at synchronous bit rates up to 64 k bits per second. It uses RXD, CTS, DCD, TXD, RTS, and DTR. It also interfaces to the synchronous clock signal lines. Refer to Chapter 7 for drawings of the serial port interface connections.

All four serial ports on the MVME167/187 use EIA-232-D drivers and receivers located on the main board, and all the signal lines are routed to the I/O connector. The configuration headers are located on the main board and the MVME712X transition board. An external I/O transition board such as the MVME712X should be used to provide configuration headers and industry-standard connectors.

Note

The MVME167/MVME187 board hardware ties the DTR signal from the CD2401 to the pin labeled RTS at connector P2. Likewise, RTS from the CD2401 is tied to DTR on P2.

Therefore, when programming the CD2401, assert DTR when you want RTS, and RTS when you want DTR.

The four serial ports on the MVME166 are functionally the same. All serial ports are full function asynchronous or synchronous ports. They can operate at synchronous bit rates up to 64 k bits per second. They use RXD, CTS, DCD, TXD, RTS, DTR, and DSR. They also interface to the synchronous clock signal lines. Additional control signals are provided for each serial port by the MC68230. These include local loopback control, self test control, and ring indicator. The ring indicator signal can be programmed to generate a local bus interrupt. Refer to the MC68230 section for additional information. Refer to Chapter 7 for drawings of the serial port interface connections. Note that the usable functionality of the serial ports depends on the transition module used.

All four serial ports on the MVME166 use a TTL interface to the transition board. This allows the interface specific drivers to be located on the transition board. This allows more flexibility in configuring the serial ports for different interfaces like EIA-232-D or V.35. An external I/O transition module such as the MVME712-10 should be used to provide configuration headers, interface drivers, and industry-standard connectors.

On the MVME166/167/187, the interface provided by the PCCchip2 allows the 16-bit CD2401 to appear at contiguous addresses; however, accesses to the CD2401 must be 8 or 16 bits. 32-bit accesses are not permitted. Refer to the CD2401 data sheet and to the PCCchip2 in Chapter 3 for detailed programming information.

On the MVME166/167/187, the CD2401 supports DMA operations to local memory. Because the CD2401 does not support a retry operation necessary to break VMEbus or VSB dual port lockup conditions, the CD2401 DMA controllers should not be programmed to access the VMEbus or VSB. The hardware does not restrict the CD2401 to onboard DRAM.

MC68230 Parallel Interface/Timer

The MVME166 provides an MC68230 parallel interface/timer (PI/T) chip. When the MVME166 is used with the MVME712-10 transition module or the MVME712-06/07/09 I/O distribution board set, the MC68230 is used to provide additional control lines for the serial ports. These include local loopback, self test, and ring indicator. The ring indicator signals can be programmed to generate local bus interrupts. Refer to Chapter 7 in this manual, and to the MVME712-10 transition module manual, for more information.

The base address of the MC68230 is $FFF45E00, and because it is an 8-bit device it appears only at odd addresses. Space for the MC68230 was created by dividing the area occupied by redundant copies of the CD2401 registers into eight segments. The CD2401 is still addressed at $FFF45000 to $FFF451FF. Addresses $FFF45200 to $FFF45BFF are reserved, and if accessed on an MVME166 cause a local bus time-out error, if the local bus timer is enabled. The address range from $FFF45C00 to $FFF45DFF always returns a local bus time-out error if the local bus timer is enabled. The CD2401 appears redundantly from $FFF45200 to $FFF45FFF on the MVME167/187.

The presence of the MC68230 can be determined by reading address $FFF45C00. If a time-out error occurs, then the board is an MVME166 and the MC68230 is present. If a time-out does not occur, then the board is an MVME167/187 and the MC68230 is not present. The local bus time-out timer in the VMEchip2 must be enabled for this test.

The MC68230 may be used for general purpose I/O when the MVME166 is not used with the MVME712 family of transition modules. Because the outputs are unbuffered and unprotected, these signals should be used with caution. The port A signal lines PA<7..0> are connected to the front panel connector J9. The port A signal lines can be programmed as inputs or outputs. The port B signal lines PB<3..0> are connected to the port H signal lines H<4..1> and the front panel connector J9. This allows these four lines to be inputs or outputs or receive interrupts. The port B signal line PB<7> is also connected to the front panel connector J9. When used with the MVME712 family of transition modules, the PB<7> signal line is used to read the configuration of the serial ports. Timer interrupts from the MC68230 are not supported on the MVME166. The MC68230 is connected to a 10 MHz clock. The PC0 bit in the MC68230 PI/T chip must be low to enable writes to Flash memory.

Printer Interface

The MVME166/167/187 has a Centronics-compatible printer interface. The printer interface is provided by the PCCchip2. Refer to the PCCchip2 in Chapter 3 for detailed programming information. Refer to Chapter 7 for drawings of the printer port interface connections.

Ethernet Interface

The 82596CA is used to implement the Ethernet transceiver interface. The 82596CA accesses local RAM using DMA operations to perform its normal functions. Because the 82596CA has small internal buffers and the VMEbus has an undefined latency period, buffer overrun may occur if the DMA is programmed to access the VMEbus or VSB. Therefore, the 82596CA should not be programmed to access the VMEbus or VSB.

Every MVME166/167/187 is assigned an Ethernet Station Address. The address is $08003E2*xxxxx* where *xxxxx* is the unique 5-nibble number assigned to the board (i.e., every MVME166/167/187 has a different value for *xxxxx*).

Each board has an Ethernet Station Address displayed on a label attached to the VMEbus P2 connector. In addition, the six bytes including the Ethernet address are stored in the configuration area of the BBRAM. That is, 08003E2*xxxxx* is stored in the BBRAM. At an address of $FFFC1F2C, the upper four bytes (08003E2X) can be read. At an address of $FFFC1F30, the lower two bytes (*xxxx*) can be read. Refer to the BBRAM, TOD Clock memory map description later in this chapter. The MVME166/167/187 debugger has the capability to retrieve or set the Ethernet address, as does the MVME166 BootBug.

If the data in the BBRAM is lost, the user should use the number on the VMEbus P2 connector label to restore it.

The Ethernet transceiver interface is located on the MVME166/167/187 main board, and the industry standard connector is located on the MVME712X transition board.

Support functions for the 82596CA are provided by the PCCchip2. Refer to the 82596CA user's guide and to the PCCchip2 in Chapter 3 for detailed programming information.

SCSI Interface

The MVME166/167/187 provides for mass storage subsystems through the industry-standard SCSI bus. These subsystems may include hard and floppy disk drives, streaming tape drives, and other mass storage devices. The SCSI interface is implemented using the NCR 53C710 SCSI I/O controller.

Support functions for the 53C710 are provided by the PCCchip2. Refer to the 53C710 user's guide and to the PCCchip2 in Chapter 3 for detailed programming information.

SCSI Termination

The system configurer must ensure that the SCSI bus is properly terminated at both ends. On the MVME167/187, sockets are provided for the terminators on the P2 transition board. If the SCSI bus ends at the P2 transition board, then termination resistors must be installed on the P2 transition board. +5V power to the SCSI bus TERM power line and termination resistors is provided through a fuse located on the P2 transition board.

On the MVME166, the SCSI bus termination is provided on the main board. The terminators are enable/disabled by a jumper. If the SCSI bus ends at the MVME166, the SCSI terminators must be enabled by installing the jumper. Refer to the jumper configuration tables in this chapter.

Local Resources

The MVME166/167/187 includes many resources for the local processor. These include tick timers, software programmable hardware interrupts, watchdog timer, and local bus time-out.

Programmable Tick Timers

Four 32-bit programmable tick timers with 1 µsec resolution are provided, two in the VMEchip2 and two in the PCCchip2. The tick timers can be programmed to generate periodic interrupts to the processor. Refer to the VMEchip2 and PCCchip2 in Chapters 2 and 3, respectively, for detailed programming information.

Watchdog Timer

A watchdog timer function is provided in the VMEchip2. When the watchdog timer is enabled, it must be reset by software within the programmed time or it times out. The watchdog timer can be programmed to generate a SYSRESET signal, local reset signal, or board fail signal if it times out. Refer to the VMEchip2 in Chapter 2 for detailed programming information.

Software-Programmable Hardware Interrupts

Eight software-programmable hardware interrupts are provided by the VMEchip2. These interrupts allow software to create a hardware interrupt. Refer to the VMEchip2 in Chapter 2 for detailed programming information.

Local Bus Time-out

The MVME166/167/187 provides a time-out function for the local bus. When the timer is enabled and a local bus access times out, a Transfer Error Acknowledge (TEA) signal is sent to the local bus master. The time-out value is selectable by software for 8 µsec, 64 µsec, 256 µsec, or infinite. The local bus timer does not operate during VMEbus or VSB bound cycles. VMEbus bound cycles are timed by the VMEbus access timer and the VMEbus global timer. Refer to the VMEchip2 in Chapter 2 for detailed programming information. VSB bound cycles are timed by the VSB access timer, the VSB transfer timer, and if its serial arbiter is enabled, by the VSB arbitration timer. Refer to the VSBchip2 in Chapter 6 for detailed programming information.

Connectors

The MVME166/167/187 has two 96-position DIN connectors: P1 and P2. P1 rows A, B, C, and P2 row B provide the VMEbus interconnection. P2 rows A and C, on the MVME167/187, provide the connection to the SCSI bus, serial ports, Ethernet, and printer. P2 rows A and C, on the MVME166, provide the connection to the VSB. The MVME166/167/187 has a 20-pin connector mounted behind the front panel. When the MVME166/167/187 board is enclosed in a chassis and the front panel is not visible, this connector allows the reset, abort and LED functions to be extended to the control panel of the system, where they are visible. The MVME166 has a 68-pin mini D ribbon shielded connector for the SCSI bus interface. The MVME166 has a 100-pin mini D ribbon shielded connector for the serial ports, Ethernet, and printer. Refer to the support manual for the specific board for detailed connector pinout information.

Fuses

The MVME166/167/187 boards supply power to various I/O devices. The power sources are fused by pico fuses on the main board. The fuses are included to protect the board in case any of the power sources are shorted. If a fuse is blown, the board may behave in a erratic manner or stop functioning completely. If any of the onboard I/O devices behave this way, the fuses should be checked. There are several LEDs to monitor the status of the fuses.

MVME167/187 Fuses

The MVME167/187 provides +5V power to the 20-pin remote reset connector J3 through fuse F1 located near the connector. This voltage source is only used when the LED functions are extended and there is no onboard monitor.

The MVME167/187 provides +12V power to the transition boards through fuse F2 located between the VMEbus P1 and P2 connectors. This voltage source is used to power LAN transceivers connected to the MVME712X transition boards and for pullup resistors on some serial port lines. The +12V LED on the front panel of the MVME167/187 is used to monitor this voltage source.

The P2 transition board used with the MVME167/187 provides +5V to SCSI bus TERMPWR signal through a fuse. There is no monitor LED on the MVME167/187 for this fuse. The MVME712M transition board and some SCSI bus terminators provide monitor LEDs.

MVME166 Fuses

The MVME166 provides +12V, -12V, and +5V to the transition boards through fuses F1, F3, and F4. The fused +5V is also provided to the 20-pin remote reset connector. These voltage sources are used by the transition boards to power the serial port drivers and any LAN transceivers connected to the transition board. The RPWR LED is lit when all three voltages are available.

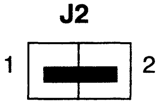
The MVME166 provides +5V to the SCSI bus TERMPWR signal through fuse F2 located near the front panel SCSI bus connector. The TPWR LED monitors the SCSI bus TERMPWR signal. Because any devices on the SCSI bus can provide TERMPWR, the LED does not directly indicate the condition of the fuse. If the LED is not illuminated, the fuse should be checked.

Configuration Jumpers, MVME166

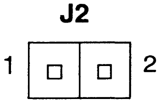
The MVME166 was designed to provide software control for most options. Some options cannot be done in software, so are done by jumpers on headers. This section describes the jumpers used on the MVME166.

SCSI Terminator Enable Header J2

The MVME166 provides terminators for the SCSI bus. The SCSI terminators are enabled/disabled by jumpers on header J2. The SCSI terminators may be configured as follows.



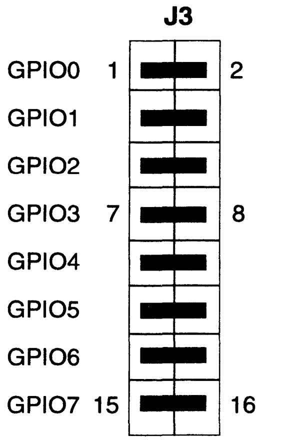
Onboard SCSI Bus Terminator Enabled
(Factory Configuration)



Onboard SCSI Bus Terminator Disabled

General Purpose Readable Jumpers on Header J3

Each MVME166 may be configured with readable jumpers. These jumpers can be read as a register (at $FFF40088) in the VMEchip2 LCSR. The bit values are read as a one when the jumper is off, and as a zero when the jumper is on.

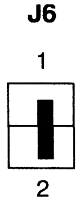


IN = Normal Debugger
OUT = BootBug

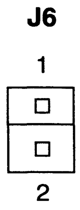
All Zeros
(Factory Configuration)

System Controller Header J6

The MVME166 can be VMEbus system controller. The system controller function is enabled/disabled by jumpers on header J6. When the MVME166 is system controller, the SCON LED is turned on. The VMEchip2 may be configured as a system controller as follows.



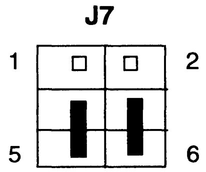
System Controller
(Factory Configuration)



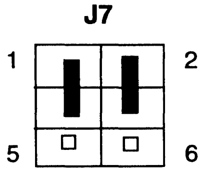
Not System Controller

SRAM Backup Power Source Select Header J7

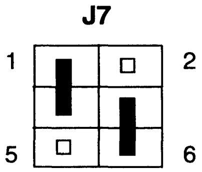
Header J7 is used to select the power source used to back up the SRAM on the MVME166.



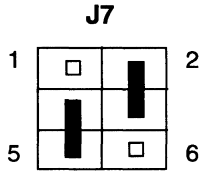
Primary Source Onboard Battery
Secondary Source Onboard Battery



Primary Source VMEbus +5V STBY
Secondary Source VMEbus +5V STBY
(Factory Configuration)



Primary Source VMEbus +5V STBY
Secondary Source Onboard Battery



Primary Source Onboard Battery
Secondary Source VMEbus +5V STBY

Caution

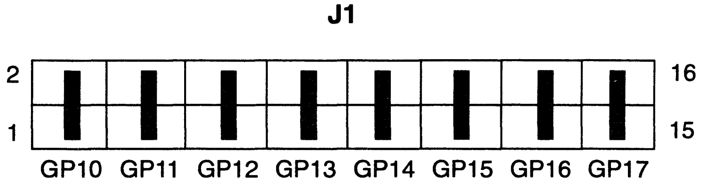
Do not remove all jumpers from J7. This may disable the SRAM.
If the battery is removed, jumpers must be installed on J7, between pins 1 to 3 and pins 2 to 4, as shown in the Factory Configuration drawing above.

Configuration Jumpers, MVME167

The MVME167 was designed to provide software control for most options. Some options can not be done in software, so are done by jumpers on headers. This section describes the jumpers used on the MVME167.

General Purpose Readable Jumpers on Header J1

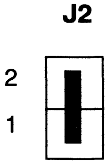
Each MVME167 may be configured with readable jumpers. These jumpers can be read as a register (at $FFF40088) in the VMEchip2 LCSR. The bit values are read as a one when the jumper is off, and as a zero when the jumper is on.



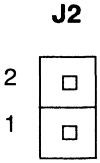
All Zeros (Factory Configuration)

System Controller Header J2

The MVME167 can be VMEbus system controller. The system controller function is enabled/disabled by jumpers on header J2. When the MVME167 is system controller, the SCON LED is turned on. The VMEchip2 may be configured as a system controller as follows.



System Controller
(Factory Configuration)

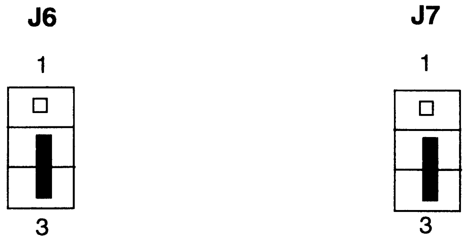


Not System Controller

Serial Port 4 Clock Configuration Select Headers J6 and J7

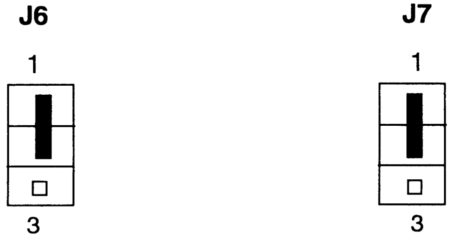
Serial port 4 can be configured to use clock signals provided by the RTXC4 and TRXC4 signal lines. Headers J6 and J7 on the MVME167 configure serial port 4 to drive or receive RTXC4 and TRXC4, respectively. Factory configuration is with port 4 set to receive both signals.

The remaining configuration of the clock lines is accomplished using the Serial Port 4 Clock Configuration Select header on the MVME712M transition board. Refer to the *MVME712M Transition Module and P2 Adapter Board User's Manual* for configuration of that header.



Receive RTXC4 Receive TRXC4

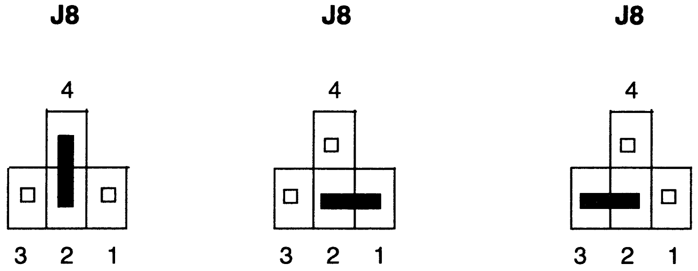
(Factory Configurations)



Drive RTXC4 Drive TRXC4

SRAM Backup Power Source Select Header J8

Header J8 is an optional header that is used to select the power source used to back up the SRAM on the MVME167, if the optional battery and circuitry are present.



Backup Power Disabled

VMEbus +5V STBY (Factory Configuration, when Optional Battery Is Present)

Optional Battery

Caution

Do not remove all jumpers from J8. This may disable the SRAM.

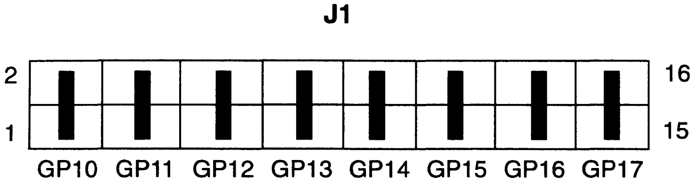
If your board contains the optional header J8, but the optional battery is removed, jumpers must be installed on J8 between pins 2 and 4, as shown in the Backup Power Disabled drawing above.

Configuration Jumpers, MVME187

The MVME187 was designed to provide software control for most options. Some options can not be done in software, so are done by jumpers on headers. This section describes the jumpers used on the MVME187.

General Purpose Readable Jumpers on Header J1

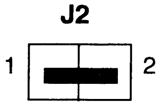
Each MVME187 may be configured with readable jumpers. These jumpers can be read as a register (at $FFF40088) in the VMEchip2 LCSR. The bit values are read as a one when the jumper is off, and as a zero when the jumper is on.



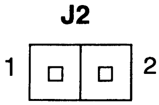
All Zeros
(Factory Configuration)

System Controller Header J2

The MVME187 can be VMEbus system controller. The system controller function is enabled/disabled by jumpers on header J2. When the MVME187 is system controller, the SCON LED is turned on. The VMEchip2 may be configured as a system controller as follows.



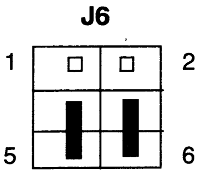
System Controller
(Factory Configuration)



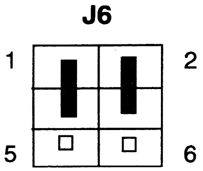
Not System Controller

SRAM Backup Power Source Select Header J6

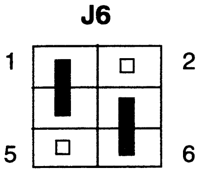
Header J6 is an optional header that is used to select the power source used to back up the SRAM on the MVME187, if the optional battery and circuitry are present.



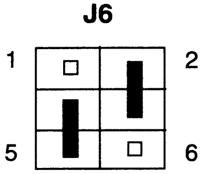
Primary Source Optional Battery
Secondary Source Optional Battery



Primary Source VMEbus +5V STBY
Secondary Source VMEbus +5V STBY
(Factory Configuration, when
Optional Battery Is Present)



Primary Source VMEbus +5V STBY
Secondary Source Optional Battery



Primary Source Optional Battery
Secondary Source VMEbus +5V STBY

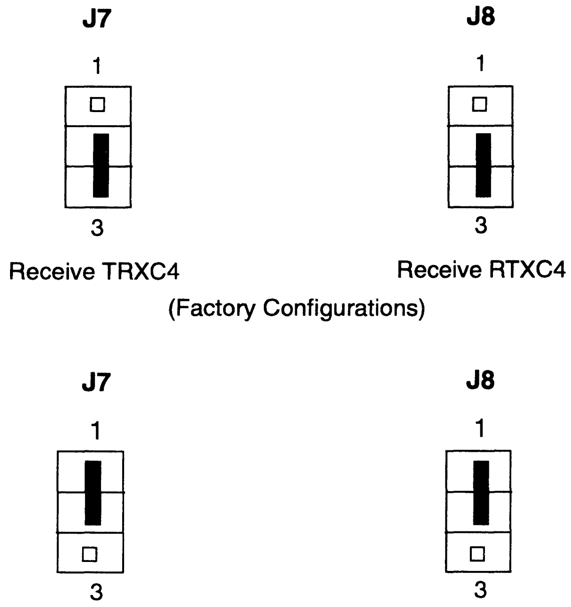
Caution

Do not remove all jumpers from J6. This may disable the SRAM.

If your board contains the optional header J6 but the optional battery is removed, jumpers must be installed on J6, between pins 1 to 3 and pins 2 to 4, as shown in the Factory Configuration drawing above.

Serial Port 4 Clock Configuration Select Headers J7 and J8

Serial port 4 can be configured to use clock signals provided by the TRXC4 and RTXC4 signal lines. Headers J7 and J8 on the MVME187 configure serial port 4 to drive or receive TRXC4 and RTXC4, respectively. Factory configuration is with port 4 set to receive both signals. The remaining configuration of the clock lines is accomplished using the Serial Port 4 Clock Configuration Select header on the MVME712M transition board. Refer to the *MVME712M Transition Module and P2 Adapter Board User's Manual* for configuration of that header.



Receive TRXC4

Receive RTXC4

(Factory Configurations)

Drive TRXC4

Drive RTXC4

Memory Maps

There are two points of view for memory maps: 1) the mapping of all resources as viewed by local bus masters (local bus memory map), and 2) the mapping of onboard resources as viewed by external masters (VMEbus memory map or VSB memory map)

Local Bus Memory Map

The local bus memory map is split into different address spaces by the transfer type (TT) signals. The local resources respond to the normal access and interrupt acknowledge codes.

Normal Address Range

The memory map of devices that respond to the normal address range is shown in the following tables. The normal address range is defined by the Transfer Type (TT) signals on the local bus. On the MVME166/167, Transfer Types 0, 1, and 2 define the normal address range. On the MVME187, Transfer Types 0 and 1 define the normal address range.

Table 1-3 is the entire map from $00000000 to $FFFFFFFF. Many areas of the map are user-programmable, and suggested uses are shown in the table. The cache inhibit function is programmable in the MMUs. The onboard I/O space must be marked cache inhibit and serialized in its page table.

Table 1-4 further defines the map for the local I/O devices portion of the local bus Main Memory Map.

Detailed I/O Memory Maps

Tables 1-5 through 1-17 give the detailed memory maps for:

1-5	VMEchip2
1-6	VSBchip2
1-7	PCCchip2
1-8	Printer
1-9	MEMC040 memory controller chip
1-10	MCECC memory controller chip
1-11	CD2401 serial chip
1-12	MC68230 PI/T chip
1-13	82596CA Ethernet chip
1-14	53C710 SCSI chip
1-15	MK48T08 BBRAM/TOD clock
1-16	BBRAM configuration area
1-17	TOD clock

Note: Manufacturers' errata sheets for the various chips are available by contacting your local Motorola sales representative. A non-disclosure agreement may be required.

Table 1-3. Local Bus Memory Map

Address Range		Devices Accessed	Port Size	Size	Software Cache Inhibit	Notes
$00000000 - DRAMSIZE		User Programmable (Onboard DRAM)	D32	DRAMSIZE	N	1, 2
DRAMSIZE - $FF7FFFFF		User Programmable (VMEbus or VSB)	D32/D16	3GB	?	3, 4
$FF800000 -	$FF8FFFFF	FLASH (MVME166)	D32	1MB	N	1
	$FFBFFFFF	ROM (MVME167/187)	D32	4MB	N	1
$FFC00000 - $FFDFFFFF		Reserved	--	2MB	--	5
$FFE00000 - $FFE1FFFF		SRAM	D32	128KB	N	--
$FFE20000 - $FFEFFFFF		SRAM (repeated)	D32	896KB	N	--
$FFF00000 - $FFFEFFFF		Local I/O Devices (Refer to next table)	D32-D8	1MB	Y	3
$FFFF0000 - $FFFFFFFF		User Programmable (VMEbus A16)	D32/D16	64KB	?	2, 4

NOTES:

1. Onboard EPROM (Download EPROM on the MVME166) appears at $00000000 - ROMSIZE following a local bus reset. The EPROM appears at 0 until the ROM0 bit is cleared in the VMEchip2. The ROM0 bit is located at address $FFF40030 bit 20. The EPROM must be disabled at 0 before the DRAM is enabled. The VMEchip2, VSBchip2, and DRAM map decoders are disabled by a local bus reset. (The Download EPROM appears at 0 until the DR0 bit is cleared in the PCCchip2. The DR0 bit is located at address $FFF42000 bit 15.)
2. This area is user-programmable. The suggested use is shown in the table. The DRAM decoder is programmed in the MEMC040 or MCECC chip, and the local-to-VMEbus decoders are programmed in the VMEchip2. The local-to-VSB decoders are programmed in the VSBchip2.
3. Size is approximate.
4. Cache inhibit depends on devices in area mapped.
5. This area is not decoded. If these locations are accessed and the local bus timer is enabled, the cycle times out and is terminated by a TEA signal.

Table 1-4. Local I/O Devices Memory Map

Address Range	Devices Accessed	Port Size	Size	Notes
$FFF00000 - $FFF3FFFF	reserved	--	256KB	5
$FFF40000 - $FFF400FF	VMEchip2 (LCSR)	D32	256B	1,4
$FFF40100 - $FFF401FF	VMEchip2 (GCSR)	D32-D8	256B	1,4
$FFF40200 - $FFF40FFF	reserved	--	3.5KB	5,7
$FFF41000 - $FFF41FFF	VSBchip2	D32-D8	4KB	1,10
$FFF42000 - $FFF42FFF	PCCchip2	D32-D8	4KB	1
$FFF43000 - $FFF430FF	MEMC040/MCECC #1	D8	256B	1
$FFF43100 - $FFF431FF	MEMC040/MCECC #2	D8	256B	1
$FFF43200 - $FFF43FFF	MEMC040s/MCECCs (repeated)	--	3.5KB	1,7
$FFF44000 - $FFF44FFF	reserved	--	4KB	5
$FFF45000 - $FFF451FF	CD2401 (Serial Comm. Cont.)	D16-D8	512B	1,9
$FFF45200 - $FFF45DFF	reserved	--	3KB	7,9
$FFF45E00 - $FFF45FFF	MC68230	--	512B	1,9
$FFF46000 - $FFF46FFF	82596CA (LAN)	D32	4KB	1,8
$FFF47000 - $FFF47FFF	53C710 (SCSI)	D32/D8	4KB	1
$FFF48000 - $FFF4FFFF	reserved	--	32KB	5
$FFF50000 - $FFF6FFFF	reserved	--	128KB	5
$FFF70000 - $FFF76FFF	reserved	--	28KB	6
$FFF77000 - $FFF77FFF	CODE CMMU	D32	4KB	1,2
$FFF78000 - $FFF7EFFF	reserved	--	28KB	6
$FFF7F000 - $FFF7FFFF	DATA CMMU	D32	4KB	1,2
$FFF80000 - $FFF9FFFF	Download EPROM	--	128KB	11
$FFFA0000 - $FFFBFFFF	reserved	--	128KB	5
$FFFC0000 - $FFFCFFFF	MK48T08 (BBRAM, TOD Clock)	D32-D8	64KB	1
$FFFD0000 - $FFFDFFFF	reserved	--	64KB	5
$FFFE0007	IACK LEVEL 1	D8	1 byte	2,3
$FFFE000B	IACK LEVEL 2	D8	1 byte	2,3
$FFFE000F	IACK LEVEL 3	D8	1 byte	2,3
$FFFE0013	IACK LEVEL 4	D8	1 byte	2,3
$FFFE0017	IACK LEVEL 5	D8	1 byte	2,3
$FFFE001B	IACK LEVEL 6	D8	1 byte	2,3
$FFFE001F	IACK LEVEL 7	D8	1 byte	2,3
$FFFE0020 - $FFFEFFFF	IACK LEVELS (repeated)	--	64KB	2,7

NOTES:

1. For a complete description of the register bits, refer to the data sheet for the specific chip. For a more detailed memory map refer to the following detailed peripheral device memory maps.
2. MVME187 only. On the MVME166/167 this area does not return an acknowledge signal. If the local bus timer is enabled, the access times out and is terminated by a TEA signal.
3. Byte reads should be used to read the interrupt vector. These locations do not respond when an interrupt is not pending. If the local bus timer is enabled, the access times out and is terminated by a TEA signal.
4. Writes to the LCSR in the VMEchip2 must be 32 bits. LCSR writes of 8 or 16 bits terminate with a TEA signal. Writes to the GCSR may be 8, 16 or 32 bits. Reads to the LCSR and GCSR may be 8, 16 or 32 bits.
5. This area does not return an acknowledge signal. If the local bus timer is enabled, the access times out and is terminated by a TEA signal.
6. This area does return an acknowledge signal.
7. Size is approximate.
8. Port commands to the 82596CA must be written as two 16-bit writes: upper word (two-byte) first and lower word (two-byte) second.
9. The MC68230 is included only on the MVME166. The CD2401 appears repeatedly from $FFF45200 to $FFF45FFF on the MVME167/187. The area from $FFF45200 to $FFF45DFF does not return an acknowledge on the MVME166. If the local bus timer is enabled, the access times out and is terminated by a TEA signal.
10. The VSBchip2 is included only on the MVME166. On the MVME167/187 this area does not return an acknowledge signal. If the local bus timer is enabled, the access times out and is terminated by a TEA signal.
11. The Download EPROM is only on the MVME166. On the MVME167/187, this area does return an acknowledge signal.

Table 1-5. VMEchip2 Memory Map (Sheet 1 of 3)

VMEchip2 LCSR Base Address = $FFF40000

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	DATA 2
	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	DATA 1
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 1-5. 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. ⟶

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				☒	CLR OVF 2	COC EN 2	TIC EN 2	OVERFLOW COUNTER 1				☒	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	☒							
☒	P ERROR IRQ LEVEL			☒	IRQ1E IRQ LEVEL			☒	TIC TIMER 2 IRQ LEVEL			☒	TIC TIMER 1 IRQ LEVEL		
☒	SIG 1 IRQ LEVEL			☒	SIG 0 IRQ LEVEL			☒	LM 1 IRQ LEVEL			☒	LM 0 IRQ LEVEL		
☒	SW3 IRQ LEVEL			☒	SW2 IRQ LEVEL			☒	SW1 IRQ LEVEL			☒	SW0 IRQ LEVEL		
☒	VME IRQ 4 IRQ LEVEL			☒	VMEB IRQ 3 IRQ LEVEL			☒	VME IRQ 2 IRQ LEVEL			☒	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.

This page intentionally left blank.

Table 1-5. 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 1-6. VSBchip2 Memory Map (Sheet 1 of 2)

LCSR Base Address =$FFF41000
ADDRESS OFFSETS:

	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
0000	PURS			VLED	VACTO	VTXTO	VARTO	VBE		VGA2-VGA0				SGA2-SGA0		
0004		LWPIF	VWPIF	VSBIF		ATTIF	VIAIF									
0008						LWPIL2-LWPIL0				VWPIL2-VWPIL0				VSBIL2-VSBIL0		
000C																
0010				DHB	DWB	PARMD	LVFAIR	LVRWD					VARBID3-VARBID0			
0014				VARBTD	VATS1 VATS0		VTXS1 VTXS0									
0018	LOCAL SLAVE 1 END ADDRESS															
001C	LOCAL SLAVE 1 ADDRESS OFFSET															
0020	LOCAL SLAVE 2 END ADDRESS															
0024	LOCAL SLAVE 2 ADDRESS OFFSET															
0028	LOCAL SLAVE 3 END ADDRESS															
002C	LOCAL SLAVE 3 ADDRESS OFFSET															
0030	LOCAL SLAVE 4 END ADDRESS															
0034	LOCAL SLAVE 4 ADDRESS OFFSET															
0038 - 0070																
0074	LOCAL ERROR															
0078									PRESCALER							
007C	TESTEN								CNTR63-CNTR60				CNTR53-CNTR50			
	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16

This sheet continues on facing page. →

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
								LVEC7-LVEC4							
GIE	LWPIE	VWPIE	VSBIE		ATTIE	VIAIE									
					ATTIL2-ATTIL0				VIAIL2-VIAIL0						
								CLOCK PRESCALER							
LOCAL SLAVE 1 START ADDRESS															
REN 1	WEN 1			WPE 1				BNCEN 1		VSP1-VSP0 1					
LOCAL SLAVE 2 START ADDRESS															
REN 2	WEN 2			WPE 2				BNCEN 2		VSP1-VSP0 2					
LOCAL SLAVE 3 START ADDRESS															
REN 3	WEN 3			WPE 3				BNCEN 3		VSP1-VSP0 3					
LOCAL SLAVE 4 START ADDRESS															
REN 4	WEN 4			WPE 4				BNCEN 4		VSP1-VSP0 4					
ADDRESS															
CURRENT COUNT															
CNTR43-CNTR40				CNTR33-CNTR30				CNTR23-CNTR20				CNTR13-CNTR10			
D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00

1364 9403

← This sheet begins on facing page.

Table 1-6. VSBchip2 Memory Map (Sheet 2 of 2)

Local Bus BCSR Base Address = $FFF41100 **VSB BCSR Base Address = $E0n00000**
ADDRESS OFFSETS
LOCAL:
VSB:

	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
00000 00000	READY	RESET	ATTN	ERR	IRQ											
00004 00004	TAS															
00008 00008																GENERAL PURPOSE
0000C 0000C																GENERAL PURPOSE
00010 FFFE0					LBTE	LBPE	LBXE	LBE								
00014 FFFE4		VEN	VIFAIR	IHV	INTERRUPT ARBITRATION ID				VSB INTERRUPT VECTOR							
00018 FFFE8	VSB SLAVE 1 END ADDRESS															
0001C FFFEC	VSB SLAVE 1 ADDRESS OFFSET															
00020 FFFF0	VSB SLAVE 2 END ADDRESS															
00024 FFFF4	VSB SLAVE 2 ADDRESS OFFSET															
00028 FFFF8																
0002C FFFFC																VSB ERROR
	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16

This sheet continues on facing page. ⟶

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
VSBCHIP2 VERSION								VSBCHIP2 ID							
CONTROL AND STATUS															
CONTROL AND STATUS															
VGIE						VSWIE	VWPIE							VSWIF	VWPIF
VSB SLAVE 1 START ADDRESS															
REN 1	WEN 1	POR 1	POW 1	WPE 1	SAS 1	ALTAS 1	IOAS 1		LOCK			LBTS 1		LBSC 1	
VSB SLAVE 2 START ADDRESS															
REN 2	WEN 2	POR 2	POW 2	WPE 2	SAS 2	ALTAS 2	IOAS 2		LOCK			LBTS 2		LBSC 2	
ADDRESS															
D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00

1363 9403

← This sheet begins on facing page.

Table 1-7. PCCchip2 Memory Map

PCCchip2 Base Address = $FFF42000
OFFSET:

OFFSET:	D31							D24	D23							D16
00	CHIP ID								CHIP REVISION							
04	TIC TIMER 1															
08	TIC TIMER 1															
0C	TIC TIMER 2															
10	TIC TIMER 2															
14	PRESCALER COUNT REGISTER								PRESCALER CLOCK ADJUST							
18	GPI PLTY	GPI E/L*	GPI INT	GPI IEN	GPI ICLR	GPI IRQ LEVEL								GPI	GPOE	GPO
1C				SCC RTRY ERR	SCC PAR ERR	SCC EXT ERR	SCC LTO ERR	SCC SCLR			SCC MDM ERR	SCC MDM IEN	SCC MDM AVEC	SCC MODEM IRQ LEVEL		
20																
24									SCC TRANSMIT PIACK							
28					LAN PAR ERR	LAN EXT ERR	LAN LTO ERR	LAN SCLR								
2C					SCSI PAR ERR	SCSI EXT ERR	SCSI LTO ERR	SCSI SCLR								
30	PRTR ACK PLTY	PRTR ACK E/L*	PRTR ACK INT	PRTR ACK IEN	PRTR ACK ICLR	PRTR ACK IRQ LEVEL			PRTR FLT PLTY	PRTR FLT E/L*	PRTR FLT INT	PRTR FLT IEN	PRTR FLT ICLR	PRTR FAULT IRQ LEVEL		
34	PRTR BSY PLTY	PRTR BSY E/L*	PRTR BSY INT	PRTR BSY IEN	PRTR BSY ICLR	PRTR BSY IRQ LEVEL										
38	CHIP SPEED															
3C																

SCC PROVIDES ITS OWN VECTORS

This sheet continues on facing page. ⟶

<table>
<tr><th>D15</th><th></th><th></th><th></th><th></th><th></th><th></th><th>D8</th><th>D7</th><th></th><th></th><th></th><th></th><th></th><th></th><th>D0</th></tr>
<tr><td>DRO</td><td colspan="3">☒</td><td>☒</td><td>CPU 040</td><td>MSTR INT EN</td><td>FAST BRAM</td><td colspan="8">VECTOR BASE REGISTER</td></tr>
<tr><td colspan="16">COMPARE REGISTER</td></tr>
<tr><td colspan="16">COUNTER REGISTER</td></tr>
<tr><td colspan="16">COMPARE REGISTER</td></tr>
<tr><td colspan="16">COUNTER REGISTER</td></tr>
<tr><td colspan="4">OVERFLOW COUNTER 2</td><td>☒</td><td>CLR OVF 2</td><td>COC EN 2</td><td>TIC EN 2</td><td colspan="4">OVERFLOW COUNTER 1</td><td>☒</td><td>CLR OVF 1</td><td>COC EN 1</td><td>TIC EN 1</td></tr>
<tr><td colspan="2">☒</td><td>TIC2 INT</td><td>TIC2 IEN</td><td>TIC2 ICLR</td><td colspan="3">TIC TIMER 2 IRQ LEVEL</td><td colspan="2">☒</td><td>TIC1 INT</td><td>TIC1 IEN</td><td>TIC1 ICLR</td><td colspan="3">TIC TIMER 1 IRQ LEVEL</td></tr>
<tr><td colspan="2">☒</td><td>SCC TX IRQ</td><td>SCC TX IEN</td><td>SCC TX AVEC</td><td colspan="3">SCC TRANSMIT IRQ LEVEL</td><td>SCC SC1</td><td>SCC SC0</td><td>SCC RX IRQ</td><td>SCC RX IEN</td><td>SCC RX AVEC</td><td colspan="3">SCC RECEIVE IRQ LEVEL</td></tr>
<tr><td colspan="8">☒</td><td colspan="8">SCC MODEM PIACK</td></tr>
<tr><td colspan="8">☒</td><td colspan="8">SCC RECEIVE PIACK</td></tr>
<tr><td>LAN INT PLTY</td><td>LAN INT E/L*</td><td>LAN INT</td><td>LAN IEN</td><td>LAN ICLR</td><td colspan="3">LAN INT IRQ LEVEL</td><td>LAN SC1</td><td>LAN SC0</td><td>LAN ERR INT</td><td>LAN ERR IEN</td><td>LAN ERR ICLR</td><td colspan="3">LAN ERR IRQ LEVEL</td></tr>
<tr><td colspan="8">☒</td><td colspan="2">☒</td><td>SCSI IRQ</td><td>SCSI IEN</td><td>☒</td><td colspan="3">SCSI INT IRQ LEVEL</td></tr>
<tr><td>PRTR SEL PLTY</td><td>PRTR SEL E/L*</td><td>PRTR SEL INT</td><td>PRTR SEL IEN</td><td>PRTR SEL ICLR</td><td colspan="3">PRTR SEL IRQ LEVEL</td><td>PRTR PE PLTY</td><td>PRTR PE E/L*</td><td>PRTR PE INT</td><td>PRTR PE IEN</td><td>PRTR PE ICLR</td><td colspan="3">PRTR PE IRQ LEVEL</td></tr>
<tr><td>PRTR ANY INT</td><td colspan="2">☒</td><td>PRTR ACK</td><td>PRTR FLT</td><td>PRTR SEL</td><td>PRTR PE</td><td>PRTR BSY</td><td colspan="3">☒</td><td>PRTR DAT ENBL</td><td>PRTR INP</td><td>PRTR STB</td><td>PRTR FAST ASTB</td><td>PRTR MAN STB</td></tr>
<tr><td colspan="16">PRINTER DATA</td></tr>
<tr><td colspan="4">☒</td><td>☒</td><td colspan="3">INTERRUPT IPL LEVEL</td><td colspan="4">☒</td><td>☒</td><td colspan="3">INTERRUPT MASK LEVEL</td></tr>
</table>

1362 9403

← This sheet begins on facing page.

Table 1-8. Printer Memory Map

Printer ACK Interrupt Control Register **$FFF42030**

BIT	31	30	29	28	27	26	25	24
NAME	PLTY	E/L*	INT	IEN	ICLR	IL2	IL1	IL0

Printer FAULT Interrupt Control Register **$FFF42031**

BIT	23	22	21	20	19	18	17	16
NAME	PLTY	E/L*	INT	IEN	ICLR	IL2	IL1	IL0

Printer SEL Interrupt Control Register **$FFF42032**

BIT	15	14	13	12	11	10	9	8
NAME	PLTY	E/L*	INT	IEN	ICLR	IL2	IL1	IL0

Printer PE Interrupt Control Register **$FFF42033**

BIT	7	6	5	4	3	2	1	0
NAME	PLTY	E/L*	INT	IEN	ICLR	IL2	IL1	IL0

Printer BUSY Interrupt Control Register **$FFF42034**

BIT	31	30	29	28	27	26	25	24
NAME	PLTY	E/L*	INT	IEN	ICLR	IL2	IL1	IL0

Printer Input Status Register **$FFF42036**

BIT	15	14	13	12	11	10	9	8
NAME	PLTY			ACK	FLT	SEL	PE	BSY

Printer Port Control Register **$FFF42037**

BIT	7	6	5	4	3	2	1	0
NAME				DOEN	INP	STB	FAST	MAN

Printer Data Register 16 bits **$FFF4203A**

BIT	15-0
NAME	PD15 - PD0

Table 1-9. MEMC040 Internal Register Memory Map

2nd MEMC040	1st MEMC040	Data Bits							
		D31	D30	D29	D28	D27	D26	D25	D24
$FFF43100	$FFF43000	CID7	CID6	CID5	CID4	CID3	CID2	CID1	CID0
$FFF43104	$FFF43004	REV7	REV6	REV5	REV4	REV3	REV2	REV1	REV0
$FFF43108	$FFF43008			FSTRD	EXTPEN	WPB*	MSIZ2	MSIZ1	MSIZ0
$FFF4310C	$FFF4300C	STS7	STS6	STS5	STS4	STS3	STS2	STS1	STS0
$FFF43110	$FFF43010	OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1	OUT0
$FFF43114	$FFF43014	BAD31	BAD30	BAD29	BAD28	BAD27	BAD26	BAD25	BAD24
$FFF43118	$FFF43018	BAD23	BAD22	DMCTL	SWAIT	WWP	PARINT	PAREN	RAMEN
$FFF4311C	$FFF4301C	BCK7	BCK6	BCK5	BCK4	BCK3	BCK2	BCK1	BCK0

Table 1-10. MCECC Internal Register Memory Map

MCECC Base Address = $FFF43000 (1st); $FFF43100 (2nd)

Register Offset	Register Name	Register Bit Names							
		D31	D30	D29	D28	D27	D26	D25	D24
$00	CHIP ID	CID7	CID6	CID5	CID4	CID3	CID2	CID1	CID0
$04	CHIP REVISION	REV7	REV6	REV5	REV4	REV3	REV2	REV1	REV0
$08	MEM CONFIG			FSTRD	1	0	MSIZ2	MSIZ1	MSIZ0
$0C	DUMMY 0	0	0	0	0	0	0	0	0
$10	DUMMY 1	0	0	0	0	0	0	0	0
$14	BASE ADDRESS	BAD31	BAD30	BAD29	BAD28	BAD27	BAD26	BAD25	BAD24
$18	DRAM CONTRL	BAD23	BAD22	RWB5	SWAIT	RWB3	NCEIEN	NCEBEN	RAMEN
$1C	BCLK FREQ	BCK7	BCK6	BCK5	BCK4	BCK3	BCK2	BCK1	BCK0
$20	DATA CONTRL	0	0	DERC	ZFILL	RWCKB	0	0	0
$24	SCRUB CNTRL	RACODE	RADATA	HITDIS	SCRB	SCRBEN	0	SBEIEN	IDIS
$28	SCRUB PERIOD	SBPD15	SBPD14	SBPD13	SBPD12	SBPD11	SBPD10	SBPD9	SBPD8
$2C	SCRUB PERIOD	SBPD7	SBPD6	SBPD5	SBPD4	SBPD3	SBPD2	SBPD1	SBPD0
$30	CHIP PRESCALE	CPS7	CPS6	CPS5	CPS4	CPS3	CPS2	CPS1	CPS0
$34	SCRUB TIME ON/OFF	SRDIS	0	STON2	STON1	STON0	STOFF2	STOFF1	STOFF0
$38	SCRUB PRESCALE	0	0	SPS21	SPS20	SPS19	SPS18	SPS17	SPS16
$3C	SCRUB PRESCALE	SPS15	SPS14	SPS13	SPS12	SPS11	SPS10	SPS9	SPS8
$40	SCRUB PRESCALE	SPS7	SPS6	SPS5	SPS4	SPS3	SPS2	SPS1	SPS0
$44	SCRUB TIMER	ST15	ST14	ST13	ST12	ST11	ST10	ST9	ST8

Table 1-10. MCECC Internal Register Memory Map (Continued)

MCECC Base Address = $FFF43000 (1st); $FFF43100 (2nd)

Register Offset	Register Name	Register Bit Names							
		D31	D30	D29	D28	D27	D26	D25	D24
$48	SCRUB TIMER	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
$4C	SCRUB ADDR CNTRL	0	0	0	0	0	SAC26	SAC25	SAC24
$50	SCRUB ADDR CNTRL	SAC23	SAC22	SAC21	SAC20	SAC19	SAC18	SAC17	SAC16
$54	SCRUB ADDR CNTRL	SAC15	SAC14	SAC13	SAC12	SAC11	SAC10	SAC9	SAC8
$58	SCRUB ADDR CNTRL	SAC7	SAC6	SAC5	SAC4	0	0	0	0
$5C	ERROR LOGGER	ERRLOG	ERD	ESCRB	ERA	EALT	0	MBE	SBE
$60	ERROR ADDRESS	EA31	EA30	EA29	EA28	EA27	EA26	EA25	EA24
$64	ERROR ADDRESS	EA23	EA22	EA21	EA20	EA19	EA18	EA17	EA16
$68	ERROR ADDRESS	EA15	EA14	EA13	EA12	EA11	EA10	EA9	EA8
$6C	ERROR ADDRESS	EA7	EA6	EA5	EA4	0	0	0	0
$70	ERROR SYNDROME	S7	S6	S5	S4	S3	S2	S1	S0
$74	DEFAULTS1	WRHDIS	STATCOL	FSTRD	SELI1	SELI0	RSIZ2	RSIZ1	RSIZ0
$78	DEFAULTS2	FRC_OPN	XY_FLIP	REFDIS	TVECT	NOCACHE	RESST2	RESST1	RESST0

Table 1-11. Cirrus Logic CD2401 Serial Port Memory Map

Base Address = $FFF45000

Register Description	Register Name	Offsets	Size	Access
Global Registers				
Global Firmware Revision Code Register	GFRCR	81	B	R
Channel Access Register	CAR	EE	B	R/W
Option Registers				
Channel Mode Register	CMR	1B	B	R/W
Channel Option Register 1	COR1	10	B	R/W
Channel Option Register 2	COR2	17	B	R/W
Channel Option Register 3	COR3	16	B	R/W
Channel Option Register 4	COR4	15	B	R/W
Channel Option Register 5	COR5	14	B	R/W
Channel Option Register 6	COR6	18	B	R/W

Table 1-11. Cirrus Logic CD2401 Serial Port Memory Map (Continued)

Base Address = $FFF45000

Register Description	Register Name	Offsets	Size	Access
Channel Option Register 7	COR7	07	B	R/W
Special Character Register 1	SCHR1	1F	B	R/W Async
Special Character Register 2	SCHR2	1E	B	R/W Async
Special Character Register 3	SCHR3	1D	B	R/W Async
Special Character Register 4	SCHR4	1C	B	R/W Async
Special Character Range low	SCRl	23	B	R/W Async
Special Character Range high	SCRh	22	B	R/W Async
LNext Character	LNXT	2E	B	R/W Async
Bit Rate and Clock Option Registers				
Receive Frame Address Register1	RFAR1	1F	B	R/W Sync
Receive Frame Address Register2	RFAR2	1E	B	R/W Sync
Receive Frame Address Register3	RFAR3	1D	B	R/W Sync
Receive Frame Address Register4	RFAR4	1C	B	R/W Sync
CRC Polynomial Select Register	CPSR	D6	B	R/W Sync
Receive Baud Rate Period Register	RBPR	CB	B	R/W
Receive Clock Option Register	RCOR	C8	B	R/W
Transmit Baud Rate Period Register	TBPR	C3	B	R/W
Transmit Clock Option Register	TCOR	C0	B	R/W
Channel Command and Status Registers				
Channel Command Register	CCR	13	B	R/W
Special Transmit Command Register	STCR	12	B	R/W
Channel Status Register	CSR	1A	B	R
Modem Signal Value Registers	MSVR-RTS	DE	B	R/W
	MSVR-DTR	DF	B	R/W
Interrupt Registers				
Local Interrupt Vector Register	LIVR	09	B	R/W
Interrupt Enable Register	IER	11	B	R/W
Local Interrupting Channel Register	LICR	26	B	R/W
Stack Register	STK	E2	B	R

Table 1-11. Cirrus Logic CD2401 Serial Port Memory Map (Continued)

Base Address = $FFF45000

Register Description	Register Name	Offsets	Size	Access
Receive Interrupt Registers				
Receive Priority Interrupt Level Register	RPILR	E1	B	R/W
Receive Interrupt Register	RIR	ED	B	R
Receive Interrupt Status Register	RISR	88	W (NOTE)	R/W
Receive Interrupt Status Register low	RISRl	89	B	R
Receive Interrupt Status Register high	RISRh	88	B	R
Receive FIFO Output Count	RFOC	30	B	R
Receive Data Register	RDR	F8	B	R
Receive End Of Interrupt Register	REOIR	84	B	W
Transmit Interrupt Registers				
Transmit Priority Interrupt Level Register	TPILR	E0	B	R/W
Transmit Interrupt Register	TIR	EC	B	R
Transmit Interrupt Status Register	TISR	8A	B	R
Transmit FIFO Transfer Count	TFTC	80	B	R
Transmit Data Register	TDR	F8	B	W
Transmit End Of Interrupt Register	TEOIR	85	B	W
Modem Interrupt Registers				
Modem Priority Interrupt Level Register	MPILR	E3	B	R/W
Modem Interrupt Register	MIR	EF	B	R
Modem (/Timer) Interrupt Status Register	MISR	8B	B	R
Modem End Of Interrupt Register	MEOIR	86	B	W
DMA Registers				
DMA Mode Register (write only)	DMR	F6	B	W
Bus Error Retry Count	BERCNT	8E	B	R/W
DMA Buffer Status	DMABSTS	19	B	R
DMA Receive Registers				
A Receive Buffer Address Lower	ARBADRL	42	W	R/W
A Receive Buffer Address Upper	ARBADRU	40	W	R/W
B Receive Buffer Address Lower	BRBADRL	46	W	R/W

Table 1-11. Cirrus Logic CD2401 Serial Port Memory Map (Continued)

Base Address = $FFF45000

Register Description	Register Name	Offsets	Size	Access
B Receive Buffer Address Upper	BRBADRU	44	W	R/W
A Receive Buffer Byte Count	ARBCNT	4A	W	R/W
B Receive Buffer Byte Count	BRBCNT	48	W	R/W
A Receive Buffer Status	ARBSTS	4F	B	R/W
B Receive Buffer Status	BRBSTS	4E	B	R/W
Receive Current Buffer Address Lower	RCBADRL	3E	W	R
Receive Current Buffer Address Upper	RCBADRU	3C	W	R
DMA Transmit Registers				
A Transmit Buffer Address Lower	ATBADRL	52	W	R/W
A Transmit Buffer Address Upper	ATBADRU	50	W	R/W
B Transmit Buffer Address Lower	BTBADRL	56	W	R/W
B Transmit Buffer Address Upper	BTBADRU	54	W	R/W
A Transmit Buffer Byte Count	ATBCNT	5A	W	R/W
B Transmit Buffer Byte Count	BTBCNT	58	W	R/W
A Transmit Buffer Status	ATBSTS	5F	B	R/W
B Transmit Buffer Status	BTBSTS	5E	B	R/W
Transmit Current Buffer Address Lower	TCBADRL	3A	W	R
Transmit Current Buffer Address Upper	TCBADRU	38	W	R
Timer Registers				
Timer Period Register	TPR	DA	B	R/W
Receive Time-out Period Register	RTPR	24	W	R/W Async
Receive Time-out Period Regis low	RTPRl	25	B	R/W Async
Receive Time-out Period Register high	RTPRh	24	B	R/W Async
General Timer 1	GT1	2A	W	R Sync
General Timer 1 low	GT1l	2B	B	R Sync
General Timer 1 high	GT1h	2A	B	R Sync
General Timer 2	GT2	29	B	R Sync
Transmit Timer Register	TTR	29	B	R Async

NOTE: This is a 16-bit register.

Table 1-12. MC68230 PI/T Register Map

MC68230 Base Address = $FFF45E00

Offset	Physical Address	Register Name	Register Description
$1	$FFF45E01	PGCR	Port General Control Register
$3	$FFF45E03	PSRR	Port Service Request Register
$5	$FFF45E05	PADDR	Port A Data Direction Register
$7	$FFF45E07	PBDDR	Port B Data Direction Register
$9	$FFF45E09	PCDDR	Port C Data Direction Register
$B	$FFF45E0B	PIVR	Port Interrupt Vector Register
$D	$FFF45E0D	PACR	Port A Control Register
$F	$FFF45E0F	PBCR	Port B Control Register
$11	$FFF45E11	PADR	Port A Data Register
$13	$FFF45E13	PBDR	Port B Data Register
$15	$FFF45E15	PAAR	Port A Alternate Register
$17	$FFF45E17	PBAR	Port B Alternate Register
$19	$FFF45E19	PCDR	Port C Data Register
$1B	$FFF45E1B	PSR	PortStatus Register
$21	$FFF45E21	TCR	Timer Control Register
$23	$FFF45E23	TIVR	Timer Interrupt Vector Register
$27	$FFF45E27	CPRH	Counter Preload Register High
$29	$FFF45E29	CPRM	Counter Preload Register Middle
$2B	$FFF45E2B	CPRL	Counter Preload Register Low
$2F	$FFF45E2F	CNTRH	Counter Register High
$31	$FFF45E31	CNTRM	Counter Register Middle
$33	$FFF45E33	CNTRL	Counter Register Low
$35	$FFF45E35	TSR	Timer Status Register

Table 1-13. 82596CA Ethernet LAN Memory Map

82596CA Ethernet LAN Directly Accessible Registers

Address	Data Bits	
	D31 . . . D16	D15 . . . D0
$FFF46000	Upper Command Word	Lower Command Word
$FFF46004	MPU Channel Attention (CA)	

NOTES: 1. Refer to the MPU Port and MPU Channel Attention registers in Chapter 3.

2. After resetting, you must write the System Configuration Pointer to the command registers before writing to the MPU Channel Attention register. Writes to the System Configuration Pointer must be upper word first, lower word second.

Table 1-14. 53C710 SCSI Memory Map

Base Address = $FFF47000

Big Endian Mode	53C710 Register Address Map				SCRIPTs Mode and Little Endian Mode
00	SIEN	SDID	SCNTL1	SCNTL0	00
04	SOCL	SODL	SXFER	SCID	04
08	SBCL	SBDL	SIDL	SFBR	08
0C	SSTAT2	SSTAT1	SSTAT0	DSTAT	0C
10	DSA				10
14	CTEST3	CTEST2	CTEST1	CTEST0	14
18	CTEST7	CTEST6	CTEST5	CTEST4	18
1C	TEMP				1C
20	LCRC	CTEST8	ISTAT	DFIFO	20
24	DCMD	DBC			24
28	DNAD				28
2C	DSP				2C
30	DSPS				30
34	SCRATCH				34
38	DCNTL	DWT	DIEN	DMODE	38
3C	ADDER				3C

NOTE: Accesses may be 8-bit or 32-bit, but not 16-bit.

Table 1-15. MK48T08 BBRAM/TOD Clock Memory Map

Address Range	Description	Size (Bytes)
$FFFC0000 - $FFFC0FFF	User Area	4096
$FFFC1000 - $FFFC10FF	Networking Area	256
$FFFC1100 - $FFFC16F7	Operating System Area	1528
$FFFC16F8 - $FFFC1EF7	Debugger Area	2048
$FFFC1EF8 - $FFFC1FF7	Configuration Area	256
$FFFC1FF8 - $FFFC1FFF	TOD Clock	8

Table 1-16. BBRAM Configuration Area Memory Map

Address Range	Description	Size (Bytes)
$FFFC1EF8 - $FFFC1EFB	Version	4
$FFFC1EFC - $FFFC1F07	Serial Number	12
$FFFC1F08 - $FFFC1F17	Board ID	16
$FFFC1F18 - $FFFC1F27	PWA	16
$FFFC1F28 - $FFFC1F2B	Speed	4
$FFFC1F2C - $FFFC1F31	Ethernet Address	6
$FFFC1F32 - $FFFC1F33	Reserved	2
$FFFC1F34 - $FFFC1F35	SCSI ID	2
$FFFC1F36 - $FFFC1F3D	System ID	8
$FFFC1F3E - $FFFC1F45	Mezzanine Board 1 PWB	8
$FFFC1F46 - $FFFC1F4D	Mezzanine Board 1 Serial Number	8
$FFFC1F4E - $FFFC1F55	Mezzanine Board 2 PWB	8
$FFFC1F56 - $FFFC1F5D	Mezzanine Board 2 Serial Number	8
$FFFC1F5E - $FFFC1FF6	Reserved	153
$FFFC1FF7	Checksum	1

Table 1-17. TOD Clock Memory Map

Address	Data Bits								Function	
	D7	D6	D5	D4	D3	D2	D1	D0		
$FFFC1FF8	W	R	S	--	--	--	--	--	CONTROL	
$FFFC1FF9	ST	--	--	--	--	--	--	--	SECONDS	00
$FFFC1FFA	x	--	--	--	--	--	--	--	MINUTES	00
$FFFC1FFB	x	x	--	--	--	--	--	--	HOUR	00
$FFFC1FFC	x	FT	x	x	x	--	--	--	DAY	01
$FFFC1FFD	x	x	--	--	--	--	--	--	DATE	01
$FFFC1FFE	x	x	x	--	--	--	--	--	MONTH	01
$FFFC1FFF	--	--	--	--	--	--	--	--	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 MK48T08 BBRAM (also called Non-Volatile RAM or NVRAM) is divided into six areas as shown in Table 1-15. The first five areas are defined by software, while the sixth 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 used by Motorola networking software. The third area is used by the SYSTEM V/68 or SYSTEM V/88 operating system. The fourth area is used by the MVME166/MVME167/MVME187 board debugger (MVME166Bug, MVME167Bug, or MVME187Bug, respectively). The fifth area, detailed in Table 1-16, is the configuration area. The sixth area, the TOD clock, detailed in Table 1-17, is defined by the chip hardware.

The data structure of the configuration bytes starts at $FFFC1EF8 and is as follows.

```
struct brdi_cnfg {
    char      version[4];
    char      serial[12];
    char      id[16];
    char      pwa[16];
    char      speed[4];
    char      ethernet_adr[6];
    char      fill[2];
```

```
        char      lscsiid[2];
        char      sysid[8];
        char      brd1_pwb[8];
        char      brd1_serial[8];
        char      brd2_pwb[8];
        char      brd2_serial[8];
        char      reserved[153];
        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, 25 MHz MVME167 board, this field contains:

   `MVME167-003B`

   (The 12 characters are followed by four 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, 25 MHz MVME167 board at revision A, the PWA field contains:

   `01-W3899B03A`

   (The 12 characters are followed by four blanks.)

5. Four bytes contain the speed of the board in MHz. The first two bytes are the whole number of MHz and the second two bytes are fractions of MHz. For example, for a 25.00 MHz board, this field contains:

   `2500`

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. Eight bytes are reserved for the systems serial ID, for boards used in a system.
10. Eight bytes are reserved for the printed wiring board (PWB) number assigned to the first mezzanine board in ASCII format. This does *not* include the `01-W` prefix. For example, for a 16MB parity mezzanine at revision E, the PWB field contains:

    ```
    3690B03E
    ```
11. Eight bytes are reserved for the serial number assigned to the first mezzanine board in ASCII format.
12. Eight bytes are reserved for the printed wiring board (PWB) number assigned to the optional second mezzanine board in ASCII format.
13. Eight bytes are reserved for the serial number assigned to the optional second mezzanine board in ASCII format.
14. Growth space (153 bytes) is reserved. This pads the structure to an even 256 bytes. System-specific items, such as size of system side, and systems side version, may go here.
15. The final one byte of the area is reserved for a checksum (as defined in the *MVME167Bug Debugging Package User's Manual,* or the *MVME187Bug Debugging Package User's Manual*), for security and data integrity of the configuration area of the NVRAM. This data is stored in hexadecimal format.

Interrupt Acknowledge Map

The local bus distinguishes interrupt acknowledge cycles from other cycles by placing the binary value %11 on TT1-TT0. It also specifies the level that is being acknowledged using TM2-TM0. The interrupt handler selects which device within that level is being acknowledged.

On the MVME187, a read anywhere from location $FFFE0004 through $FFFE001C causes an interrupt acknowledge cycle at the specified level. Refer to the PCCchip2 information in Chapter 3 for information on reading the current interrupt level and setting the interrupt mask.

VMEbus Memory Map

This section describes the mapping of local resources as viewed by VMEbus masters.

VMEbus Accesses to the Local Bus

The VMEchip2 includes a user-programmable map decoder for the VMEbus to local bus interface. The map decoder allows you to program the starting and ending address and the modifiers to which the MVME166/167/187 responds.

VMEbus Short I/O Memory Map

The VMEchip2 includes a user-programmable map decoder for the GCSR. The GCSR map decoder allows you to program the starting address of the GCSR in the VMEbus short I/O space.

VSB Memory Map

This section describes the mapping of local resources as viewed by VSB masters. The VSBchip2, on the MVME166, includes a user-programmable map decoder for the VSB to local bus interface. This map decoder allows VSB masters access to devices on the local bus.

Software Support Considerations

The MVME166/167/187 is a complex board that interfaces to the VMEbus, VSB, and SCSI bus. These multiple bus interfaces raise the issue of cache coherency and support of indivisible cycles. There are also many sources of bus error. First, let us consider how interrupts are handled with emphasis on the different ways the MVME187 and MVME166/167 implement interrupts.

Interrupts

Because the MC68040 uses hardware-vectored interrupts while the MC88100 does not, interrupts are handled differently on the MVME166/167 and the MVME187. The C040 bit in the PCCchip2 General Control Register (address $FFF42002) should be set when the board MPU is an MC68040. It should be cleared when the MPU is an MC88100. For more information, refer to the Interrupt Prioritizer section of the PCCchip2 in Chapter 3.

Most interrupt sources are level and base vector programmable. Interrupt vectors from the PCCchip2, VSBchip2, and the VMEchip2 have two sections, a base value which can be set by the processor, usually the upper four bits, and the lower bits which are set according to the particular interrupt source. There is an onboard daisy chain of interrupt sources, with interrupts from the PCCchip2 having highest priority in the daisy chain, followed by interrupt sources from the VSBchip2, and interrupt sources from the VMEchip2 having lowest priority. When operating in the MC68040 mode, a seven-level prioritized, hardware-vectored interrupt scheme is used such as has been standard in MC68000 family product. When operating in the MC88100 mode, all interrupt sources are combined into one interrupt request to the MC88100 by the PCCchip2. When an interrupt occurs, the MC88100 reads the interrupt level from the PCCchip2 Interrupt Priority Level Register. The MC88100 then reads the corresponding IRQ Level register as shown in the table of Local I/O Devices, Table 1-4. This read causes a hardware IACK cycle to be performed and returns the associated vector as the value read. For interrupt programming details, refer to Appendix A.

Cache Coherency, MVME166/167

The MC68040 has the ability to watch local bus cycles executed by other local bus masters such as the SCSI DMA controller, the LAN, the VMEchip2 DMA controller, the VMEbus to local bus controller, and the VSB to local bus controller. This bus snooping capability is described in the *M68040 Microprocessors User's Manual* sections on *Cache Coherency* and *Bus Snooping Operation.*

When snooping is enabled, the MVME166/167 MPU can source data and invalidate cache entries as required by the current cycle. The MPU cannot watch VMEbus or VSB cycles which do not access the local bus on the MVME166/167. Software must ensure that data shared by multiple processors is kept in memory that is not cached. The software must also mark all onboard I/O areas as cache inhibited and serialized.

Cache Coherency, MVME187

Snooping is not supported on the MVME187. Software must ensure that data shared by multiple processors is kept in memory that is not cached. The software must also mark all onboard I/O areas as cache inhibited and serialized. Snoop control bits in the 53C710, VMEchip2, VSBchip2, and PCCchip2 must be set to 0 (snoop disabled).

Sources of Local BERR*

A TEA* signal (indicating a bus error) is returned to the local bus master when a local bus time-out occurs, a DRAM parity error occurs and parity checking is enabled, or a VME bus error occurs during a VMEbus access.

The devices on the MVME166/167/187 that are able to assert a local bus error are described below.

Local Bus Time-out

A Local Bus Time-out occurs whenever a local bus cycle does not complete within the programmed time (VMEbus bound cycles are not timed by the local bus timer). If the system is configured properly, this should only happen if software accesses a non-existent location within the onboard address range.

VMEbus Access Time-out

A VMEbus Access Time-out occurs whenever a VMEbus bound transfer does not receive a VMEbus bus grant within the programmed time. This is usually caused by another bus master holding the bus for an excessive period of time.

VMEbus BERR*

A VMEbus BERR* occurs when the BERR* signal line is asserted on the VMEbus while a local bus master is accessing the VMEbus. VMEbus BERR* should occur only if: an initialization routine samples to see if a device is present on the VMEbus and it is not, software accesses a non-existent device within the VMEbus range, incorrect configuration information causes the VMEchip2 to incorrectly access a device on the VMEbus (such as driving LWORD* low to a 16-bit board), a hardware error occurs on the VMEbus, or a VMEbus slave reports an access error (such as parity error).

Local DRAM Parity Error

When parity checking is enabled, the current bus master receives a bus error if it is accessing the local DRAM and a parity error occurs.

VMEchip2

An 8- or 16-bit write to the LCSR in the VMEchip2 causes a local BERR*.

VSBchip2 BERR*

The VSBchip2 on the MVME166 most likely returns a TEA* at some time. Refer to the VSBchip2 in Chapter 6 for more information on bus errors from the VSB.

Bus Error Processing

Because different conditions can cause bus error exceptions, the software must be able to distinguish the source. To aid in this, status registers are provided for every local bus master. The next section describes the various causes of bus error and the associated status registers.

Generally, the bus error handler can interrogate the status bits and proceed with the result. However, an interrupt can happen during the execution of the bus error handler (before an instruction can write to the status register to raise the interrupt mask). If the interrupt service routine causes a second bus error, the status that indicates the source of the first bus error may be lost. The software must be written to deal with this.

A Description of Error Conditions on the MVME166/167/187

This section lists the various error conditions that are reported by the MVME166/167/187 hardware. A subsection heading identifies each type of error condition. A standard format gives a description of the error, indicates how notification of the error condition is made, indicates which status register(s) have information about the error, and concludes with some comments pertaining to each particular error.

MPU Parity Error

Description:
A DRAM parity error.

MPU Notification:
TEA is asserted during an MPU DRAM access.

Status:
Bit 9 of the MPU Status and DMA Interrupt Count Register in the VMEchip2 at address $FFF40048.

Comments:
After memory has been initialized, this error normally indicates a hardware problem.

MPU Offboard Error

Description:
An error occurred while the MPU was attempting to access an offboard resource.

MPU Notification:
TEA is asserted during offboard access.

Status:
Bit 8 of the MPU Status and DMA Interrupt Count Register.
Address $FFF40048

Comments:
This can be caused by a VMEbus time-out, a VMEbus BERR, or an MVME166/167/187 VMEbus access time-out. The latter is the time from when the VMEbus has been requested to when it is granted.

MPU TEA - Cause Unidentified

Description:
An error occurred while the MPU was attempting an access.

MPU Notification:
TEA is asserted during an MPU access.

Status:
Bit 10 of the MPU Status and DMA Interrupt Count Register.
Address $FFF40048

Comments:
No status was given as to the cause of the TEA assertion.

MPU Local Bus Time-out

Description:
An error occurred while the MPU was attempting to access a local resource.

MPU Notification:
TEA is asserted during the MPU access.

Status:
Bit 7 of the MPU Status and DMA Interrupt Count Register.
(actually in the DMAC Status Register)
Address $FFF40048

Comments:
The local bus timer timed out. This usually indicates the MPU tried to read or write an address at which there was no resource. Otherwise, it indicates a hardware problem.

DMAC VMEbus Error

Description:
The DMAC experienced a VMEbus error during an attempted transfer.

MPU Notification:
DMAC interrupt (when enabled)

Status:
The VME bit is set in the DMAC Status Register
(address $FFF40048 bit 1).

Comments:
This indicates the DMAC attempted to access a VMEbus address at which there was no resource or the VMEbus slave returned a BERR signal.

DMAC Parity Error

Description:
Parity error while the DMAC was reading DRAM.

MPU Notification:
DMAC interrupt (when enabled)

Status:
The DLPE bit is set in the DMAC Status Register
(address $FFF40048 bit 5).

Comments:
If the TBL bit is set (address $FFF40048 bit 2) the error occurred during a command table access, otherwise the error occurred during a data access.

DMAC Offboard Error

Description:
Error encountered while the local bus side of the DMAC was attempting to go to the VMEbus.

MPU Notification:
DMAC interrupt (when enabled)

Status:
The DLOB bit is set in the DMAC Status Register
(address $FFF40048 bit 4).

Comments:

This is normally caused by a programming error.

The local bus address of the DMAC should not be programmed with a local bus address that maps to the VMEbus. If the TBL bit is set (address $FFF40048 bit 2) the error occurred during a command table access, otherwise the error occurred during a data access.

DMAC LTO Error

Description:

A local bus time-out (LTO) occurred while the DMAC was local bus master.

MPU Notification:

DMAC interrupt (when enabled)

Status:

The DLTO bit is set in the DMAC Status Register
(address $FFF40048 bit 3).

Comments:

This indicates the DMAC attempted to access a local bus address at which there was no resource. If the TBL bit is set (address $FFF40048 bit 2) the error occurred during a command table access, otherwise the error occurred during a data access.

DMAC TEA - Cause Unidentified

Description:

An error occurred while the DMAC was local bus master and additional status was not provided.

MPU Notification:

DMAC interrupt (when enabled)

Status:

The DLBE bit is set in the DMAC Status Register
(address $FFF40048 bit 6).

Comments:

An 8- or 16-bit write to the LCSR in the VMEchip2 causes this error. If the TBL bit is set (address $FFF40048 bit 2) the error occurred during a command table access, otherwise the error occurred during a data access.

SCC Retry Error

Description:
Local Bus Retry occurred due to VMEbus Dual Port Lock or LAN-wanted-Bus while the SCC was local bus master.

MPU Notification:
SCC Transmit Interrupt or SCC Receive Interrupt

Status:
SCC Transmit Interrupt Status Register
SCC Transmit Current Buffer Address Register
SCC Receive Interrupt Status Register High
SCC Receive Current Buffer Address Register
PCCchip2 SCC Error Status Register ($FFF4201C)

Comments:
The DMA controllers in the SCC should not be programmed to access the VMEbus. Refer to the *Serial Port Interface* section in this chapter. SCC Transmit and Receive interrupt enables are controlled in the SCC and in the PCCchip2.

SCC Parity Error

Description:
Parity Error detected while the SCC was reading DRAM.

MPU Notification:
SCC Transmit Interrupt or SCC Receive Interrupt

Status:
SCC Transmit Interrupt Status Register
SCC Transmit Current Buffer Address Register
SCC Receive Interrupt Status Register High
SCC Receive Current Buffer Address Register
PCCchip2 SCC Error Status Register ($FFF4201C)

Comments:
SCC Transmit and Receive interrupt enables are controlled in the SCC and in the PCCchip2.

SCC Offboard Error

Description:
Error encountered while the SCC was attempting to go to the VMEbus.

MPU Notification:
SCC Transmit Interrupt or SCC Receive Interrupt

Status:
SCC Transmit Interrupt Status Register
SCC Transmit Current Buffer Address Register
SCC Receive Interrupt Status Register High
SCC Receive Current Buffer Address Register
PCCchip2 SCC Error Status Register ($FFF4201C)

Comments:
SCC Transmit and Receive interrupt enables are controlled in the SCC and in the PCCchip2.

SCC LTO Error

Description:
Local Bus Time-out occurred while the SCC was local bus master.

MPU Notification:
SCC Transmit Interrupt or SCC Receive Interrupt

Status:
SCC Transmit Interrupt Status Register
SCC Transmit Current Buffer Address Register
SCC Receive Interrupt Status Register High
SCC Receive Current Buffer Address Register
PCCchip2 SCC Error Status Register ($FFF4201C)

Comments:
SCC Transmit and Receive interrupt enables are controlled in the SCC and in the PCCchip2.

LAN Parity Error

Description:
Parity error while the LANCE was reading DRAM

MPU Notification:
PCCchip2 Interrupt (LAN ERROR IRQ)

Status:
PCCchip2 LAN Error Status Register ($FFF42028)

Comments:
The LANCE has no ability to respond to TEA so the error interrupt and status are provided in the PCCchip2. Control for the interrupt is in the PCCchip2 LAN Error Interrupt Control Register ($FFF4202B).

LAN Offboard Error

Description:
Error encountered while the LANCE was attempting to go to the VMEbus.

MPU Notification:
PCCchip2 Interrupt (LAN ERROR IRQ)

Status:
PCCchip2 LAN Error Status Register ($FFF42028)

Comments:
The LANCE has no ability to respond to TEA so the error interrupt and status are provided in the PCCchip2. Control for the interrupt is in the PCCchip2 LAN Error Interrupt Control Register ($FFF4202B).

LAN LTO Error

Description:
Local Bus Time-out occurred while the LANCE was local bus master.

MPU Notification:
PCCchip2 Interrupt (LAN ERROR IRQ)

Status:
PCCchip2 LAN Error Status Register ($FFF42028)

Comments:
The LANCE has no ability to respond to TEA so the error interrupt and status are provided in the PCCchip2. Control for the interrupt is in the PCCchip2 LAN Error Interrupt Control Register ($FFF4202B).

SCSI Parity Error

Description:
Parity error detected while the 53C710 was reading DRAM.

MPU Notification:
53C710 Interrupt

Status:
53C710 DMA Status Register
53C710 DMA Interrupt Status Register
PCCchip2 SCSI Error Status Register ($FFF4202C)

Comments:
53C710 interrupt enables are controlled in the 53C710 and in the PCCchip2.

SCSI Offboard Error

Description:
Error encountered while the 53C710 was attempting to go to the VMEbus.

MPU Notification:
53C710 Interrupt

Status:
53C710 DMA Status Register
53C710 DMA Interrupt Status Register
PCCchip2 SCSI Error Status Register ($FFF4202C)

Comments:
53C710 interrupt enables are controlled in the 53C710 and in the PCCchip2.

SCSI LTO Error

Description:
Local Bus Time-out occurred while the 53C710 was local bus master.

MPU Notification:
53C710 Interrupt

Status:
53C710 DMA Status Register
53C710 DMA Interrupt Status Register
PCCchip2 SCSI Error Status Register ($FFF4202C)

Comments:
53C710 interrupt enables are controlled in the 53C710 and in the PCCchip2.

Example of the Proper Use of Bus Timers

In this example, the use of the bus timers is illustrated by describing the sequence of events when the MPU on one MVME166/167/187 accesses the local bus memory on another MVME166/167/187 using the VMEbus. A similar sequence of events could be described if the access occurred over the VSB. In this scenario there are three bus timers involved. These are the local bus timer, the VMEbus access timer, and the Global VMEbus timer. The local bus timer measures the time an access to an onboard resource takes. The VMEbus timer measures the time from when the VMEbus request has been initiated to when a VMEbus grant has been obtained. The global bus timer measures the time from when a VMEbus cycle begins to when it completes. Normally these timers should be set to quite different values.

The sequence begins when the MPU asserts a request for the local bus. The MPU must wait until the local bus is released by the current bus master before its cycle can begin. When the MPU is granted the local bus, it begins its cycle and the local bus timer starts counting. It continues to count until an address decode of the VMEbus address space is detected and then the timer stops. This is normally a very short period of time. In fact, all local bus non-error bus accesses are normally very short, such as the time to access onboard memory. Therefore, it is recommended this timer be set to a small value, such as 8 μsec.

The next timer to take over when one MVME166/167/187 accesses another is the VMEbus access timer. This measures the time between when the VMEbus has been address decoded and hence a VMEbus request has been made, and when VMEbus mastership has been granted. Because we have found in the past that some VME systems can become very busy, we recommend this time-out be set at a large value, such as 32 msec. For debug purposes this value can also be set to infinity.

Once the VMEbus has been granted, a third timer takes over. This is the global VMEbus timer. This timer starts when a transfer actually begins (DS0 or DS1 goes active) and ends when that transfer completes (DS0 or DS1 goes inactive). This time should be longer than any expected legitimate transfer time on the bus. We normally set it to 256 μsec. This timer can also be disabled for debug purposes. Before an MVME166/167/187 access to another MVME166/167/187 can complete, however, the VMEchip2 on the accessed MVME166/167/187 must decode a slave access and request the local bus of the second MVME166/167/187. When the local bus is granted (any in-process onboard transfers have completed) then the local bus timer of the accessed MVME166/167/187 starts. Normally, this is also set to 8 μsec. When the memory has the data available, a transfer acknowledge signal (TA) is given.

This translates into a DTACK signal on the VMEbus which is then translated into a TA signal to the first requesting processor, and the transfer is complete. If the VMEbus global timer expires on a legitimate transfer, the VMEbus to local bus controller in the VMEchip2 may become confused and the VMEchip2 may misbehave. Therefore the bus timers values must be set correctly. The correct settings may depend on the system configuration.

MVME166/167 MC68040 Indivisible Cycles

The MC68040 performs operations that require indivisible read-modify-write (RMW) memory accesses. These RMW sequences occur when the MMU modifies table entries or when the MPU executes a TAS, CAS, or CAS2 instruction. TAS cycles are always single-address RMW operations, while the CAS, CAS2, and MMU operations can be multiple-address RMW cycles. The VMEbus does not support multiple-address RMW cycles and there is no defined protocol for supporting multiple-address RMW cycles which start onboard and then access offboard resources. The MVME166/167/187 does not fully support all RMW operations in all possible cases.

The MVME166/167/187 makes the following assumptions and supports a limited subset of RMW instructions. The MVME166/167/187 supports single-address RMW cycles caused by TAS and CAS instructions. Because it is not possible to tell if the MC68040 is executing a single- or multiple-address read-modify-write cycle, software should only execute single-address RMW instructions. Multiple-address RMW cycles caused by CAS or CAS2 instructions are not guaranteed indivisible and may cause illegal VMEbus cycles. Lock cycles caused by MMU table walks do not cause illegal VMEbus cycles, and they are not guaranteed indivisible.

VMEchip2 2

Introduction

This chapter defines the VMEchip2, local bus to VMEbus interface chip.

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, VME board support features, 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.
  - Programmable local bus snoop enable.
  - 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.
- Programmable local bus snoop enable.
- 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:

- 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.
- ❑ Map decoder and control for two banks of EPROM (Flash memory on the MVME166).
- ❑ Map decoder and control for one bank of static RAM.
- ❑ Support for RESET and ABORT switches.

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.*

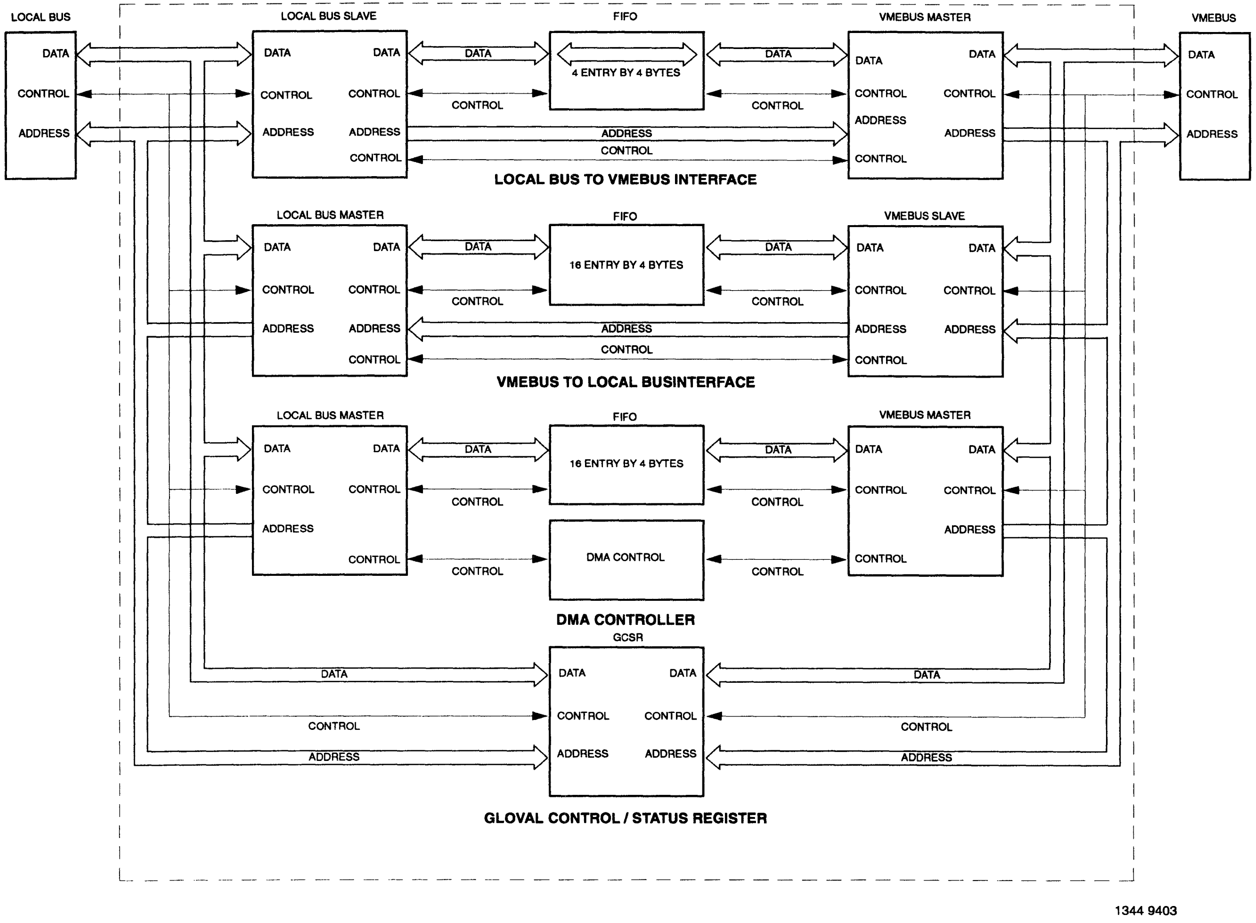


Figure 2-1. VMEchip2 Block Diagram

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

Block transfer is not supported because the MC68040 block transfer capability is not compatible with the VMEbus.

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 *VMEbus 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 and snoop 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.

When snooping is enabled, the local bus master requests the cache controller in the MC68040 to monitor the local bus addresses.

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 local 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, local bus 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 local 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 is enabled/disabled with the aid of an external jumper (the only jumper required in a VMEchip2 based VMEbus interface).

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.

In addition to the VMEbus timer, the chip contains a local bus timer. This timer asserts the local TEA when the local bus cycle maintained in its asserted state for longer that the programmed time-out period. This timer can be enabled or disabled under software control. The time-out period can be programmed for 8, 64, or 256 µsecs.

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 an edge-sensitive interrupter connected to the ABORT switch. This interrupter is filtered to remove switch bounce.

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 an edge-sensitive interrupter connected to a signal pin on the VMEchip2.

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.

VMEboard Functions

The VMEchip2 also includes several functions that are generally used on VMEbus boards. The VMEchip2 includes a local bus map decoder and control logic for two banks of EPROMs (Flash memories on the MVME166). A 4MB area is provided at $FF800000 to $FFBFFFFF for the EPROMs. Following a local bus reset, the EPROMs also respond at $00000000 to $003FFFFF (MVME167/MVME187 only) to allow the processor to fetch the execution address and stack pointer. The EPROMs respond at addresses $00000000 to $003FFFFF until the ROM0 bit is cleared in register $FFF40030.

The VMEchip2 provides a local bus map decoder and control for static RAM. The SRAM space is 1MB and it is located at local bus address $FFE00000 to $FFEFFFFF.

The VMEchip2 provides eight general purpose input signal pins and four general purpose I/O pins.

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 and 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 This bit is a read-only status bit.

  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.

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

VMEchip2 LCSR Base Address = $FFF40000

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

VMEchip2 LCSR Base Address = $FFF40000
OFFSET:

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				☒	CLR OVF 2	COC EN 2	TIC EN 2	OVERFLOW COUNTER 1				☒	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	☒							
☒	P ERROR IRQ LEVEL			☒	IRQ1E IRQ LEVEL			☒	TIC TIMER 2 IRQ LEVEL			☒	TIC TIMER 1 IRQ LEVEL		
☒	SIG 1 IRQ LEVEL			☒	SIG 0 IRQ LEVEL			☒	LM 1 IRQ LEVEL			☒	LM 0 IRQ LEVEL		
☒	SW3 IRQ LEVEL			☒	SW2 IRQ LEVEL			☒	SW1 IRQ LEVEL			☒	SW0 IRQ LEVEL		
☒	VME IRQ 4 IRQ LEVEL			☒	VMEB IRQ 3 IRQ LEVEL			☒	VME IRQ 2 IRQ LEVEL			☒	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.

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 used on the MVME166/167/187 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 $FFF40010. 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*. Local bus snooping for the segment is enabled by setting the snoop bits in the attribute register. The snoop bits in the attribute register are driven on to the local bus when the VMEbus to local bus interface is local bus master.

VMEbus Slave Ending Address Register 1

ADR/SIZ	$FFF40000 (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	$FFF40000 (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	$FFF40004 (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	$FFF40004 (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	$FFF40008 (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	$FFF40008 (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.

	Address		Address
Segment	Translation	Segment	Translation
Size	Select Value	Size	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	$FFF4000C (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	$FFF4000C (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	$FFF40010 (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 control the snoop enable lines to the local bus for the address range defined by the second VMEbus slave map decoder. The snooping functions are:

0 Snoop inhibited

1 Write - Sink data
Read - Supply dirty data and leave dirty

2 Write - Invalidate
Read - Supply dirty data and mark invalid

3 Snoop inhibited

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	$FFF40010 (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	$FFF40010 (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 control the snoop enable lines to the local bus for the address range defined by the first VMEbus slave map decoder. These bits must be 0 on the MVME187. The snooping functions are:

0	Snoop inhibited
1	Write - Sink data Read - Supply dirty data and leave dirty
2	Write - Invalidate Read - Supply dirty data and mark invalid
3	Snoop inhibited

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	$FFF40010 (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	$FFF40014 (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	$FFF40014 (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	$FFF40018 (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	$FFF40018 (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	$FFF4001C (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	$FFF4001C (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	$FFF40020 (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	$FFF40020 (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	$FFF40024 (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	$FFF40024 (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	$FFF40028 (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	$FFF40028 (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	$FFF40028 (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	$FFF40028 (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	$FFF4002C (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 MVME166/167 is $CC, and for the MVME187 is $CE.

VMEbus Slave GCSR Board Address Register

ADR/SIZ	$FFF4002C (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	$FFF4002C (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	$FFF4002C (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/SIZE	$FFF4002C (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		

This register is the ROM control register. The VMEchip2 provides a map decoder and control logic for two banks of ROM. The ROM size and speed are programmable. Bank A is always selected following reset to allow the processor to fetch the program counter and stack pointer. (Refer to the ROM0 bit in the EPROM Decoder, SRAM and DMA Control Register description later in this chapter.) The time from CS* to data valid, for the ROMs used with the VMEchip2, must be less than:

(T * (*local bus clocks* - 1) - 35)

where T is the local bus clock period, and *local bus clocks* is the programmed number of local bus clocks. For example, if the local bus clock is 33 MHz (30 ns), and the number of local bus clocks is 4, the access time of the ROMs must be less than:

(30 * (4-1) -35) = 55 ns

ASPD These bits define the number of local bus clocks for a bank A ROM cycle.

		Maximum EPROM/Flash Access Time	
	Local Bus		
ASPD	Clocks	at 25 MHz	at 33 MHz
0	11	365 ns	265 ns
1	10	325 ns	235 ns
2	9	285 ns	205 ns
3	8	245 ns	175 ns
4	7	205 ns	145 ns
5	6	165 ns	115 ns
6	5	125 ns	85 ns
7	4	85 ns	55 ns

BSPD These bits define the number of local bus clocks for a bank B ROM cycle.

		Maximum EPROM/Flash Access Time	
	Local Bus		
BSPD	Clocks	at 25 MHz	at 33 MHz
0	11	365 ns	265 ns
1	10	325 ns	235 ns
2	9	285 ns	205 ns
3	8	245 ns	175 ns
4	7	205 ns	145 ns
5	6	165 ns	115 ns
6	5	125 ns	85 ns
7	4	85 ns	55 ns

SIZE These bits define the size of the ROM chips. The ending address of bank A and the starting and ending address of bank B is defined by the ROM size. This allows the ROMs to be contiguous when both banks are equal in size.

0 8-Megabit Chips; Bank A $FF800000 to $FF9FFFFF; Bank B $FFA00000 to $FFBFFFFF

1 4-Megabit Chips; Bank A $FF800000 to $FF8FFFFF; Bank B $FF900000 to $FF9FFFFF

2 2-Megabit Chips; Bank A $FF800000 to $FF87FFFF; Bank B $FF880000 to $FF8FFFFF

3 1-Megabit Chips; Bank A $FF800000 to $FF83FFFF; Bank B $FF840000 to $FF87FFFF

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, the VMEbus and local bus address counter operation, and the local bus snoop 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

ADR/SIZ	$FFF40030 (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	

This register controls the EPROM decoder, the snoop control bits used by the DMAC when it is accessing table entries, and the access time of the SRAM (Static RAM, also known as slow RAM). The time from SRAM CS* to data valid, for the SRAMs used with the VMEchip2, must be less than:

(T * (*local bus clocks* -1) - 35)

where T is the local bus clock period, and *local bus clocks* is the programmed number of local bus clocks. For example, if the local bus clock is 33 MHz (30 ns), and the number of local bus clocks is 3, the access time of the SRAMs must be less than:

(30 * (3-1) -35) = 25 ns

SRAMS These bits define the number of local bus clocks for a static RAM cycle.

SRAMS	Local Bus Clocks	Maximum SRAM Access Time at 25 MHz	 at 33 MHz
0	6	165 ns	115 ns
1	5	125 ns	85 ns
2	4	85 ns	55 ns
3	3	45 ns	25 ns

TBLSC These bits control the snoop signal lines on the local bus when the DMAC is table walking. These bits must be 0 on the MVME187.

0 Snoop inhibited

1 Write - Sink data
Read - Supply dirty data and leave dirty

2 Write - Invalidate
Read - Supply dirty data and mark invalid

3 Snoop inhibited

ROM0 When this bit is set to 1, the EPROM decoder responds at $00000000 to $003FFFFF and $FF800000 to $FFBFFFFF. When this bit is set to 0, the EPROM decoder responds only at $FF800000 to $FFBFFFFF.
NOTE: ROM0 is set to 1 by power-up reset, SYSRESET, and local reset, causing ROM BANK A to provide reset vectors for the MPU.

WAIT RMW Not used on the MVME166/167/187.

Local Bus To VMEbus Requester Control Register

ADR/SIZ	$FFF40030 (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	$FFF40030 (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	$FFF40034 (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 control the snoop signal lines on the local bus when the DMAC is local bus master and it is not accessing the command table. These bits must be 0 on the MVME187.

0	Snoop inhibited
1	Write - Sink data Read - Supply dirty data and leave dirty
2	Write - Invalidate Read - Supply dirty data and mark invalid
3	Snoop inhibited

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	$FFF40034 (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	$FFF40038 (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	$FFF4003C (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	$FFF40040 (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	$FFF40044 (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	$FFF40048 (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	$FFF40048 (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	$FFF40048 (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 When this bit is set, the MPU received a TEA and the status indicated offboard. This bit is cleared by writing a one to the MCLR bit in this register.

MLPE When this bit is set, the MPU received a TEA and the status indicated a parity error during a DRAM data transfer. This bit is cleared by writing a one to the MCLR bit in this register.

MLBE When this bit is set, the MPU received a TEA and additional status was not provided. This bit is cleared by writing a one to the MCLR bit in this register.

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	$FFF40048 (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 When this bit is set, the DMAC received a TEA and the status indicated a local bus time-out. This bit is cleared when the DMAC is enabled.

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 When this bit is set, the MPU received a TEA and the status indicated a local bus time-out. This bit is cleared by a writing a one to the MCLR bit in this register.

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	$FFF4004C (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	$FFF4004C (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	$FFF4004C (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 define the local bus time-out value. The timer begins timing when TS is asserted on the local bus. If TA or TAE is not asserted before the timer times out, a TEA signal is sent to the local bus. The timer is disabled if the transfer is bound for the VMEbus.

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

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	$FFF4004C (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 local bus clocks. The prescaler register should be programmed based on the following equation. This provides a 1MHz clock to the tick timers.

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

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 = *B 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	$FFF40050 (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	$FFF40054 (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	$FFF40058 (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.

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

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	$FFF4005C (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	$FFF40060 (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 is enabled. When this bit is low, the RESET switch is disabled.

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.

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 an external device, 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	$FFF40060 (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	$FFF40060 (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.

Tick Timer 1 Control Register

ADR/SIZ	$FFF40060 (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	$FFF40064 (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	$FFF40068 (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	External interrupt (parity error)
MWP	VMEbus master write post error interrupt
SYSF	VMEbus SYSFAIL interrupt
AB	ABORT switch interrupt
ACF	VMEbus ACFAIL interrupt

Local Bus Interrupter Status Register (bits 16-23)

ADR/SIZ	$FFF40068 (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	$FFF40068 (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	$FFF40068 (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	$FFF4006C (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 Enable external interrupt (parity error)

EMWP Enable VMEbus master write post error interrupt

ESYSF Enable VMEbus SYSFAIL interrupt

EAB Enable ABORT switch interrupt

EACF Enable VMEbus ACFAIL interrupt

Local Bus Interrupter Enable Register (bits 16-23)

ADR/SIZ	$FFF4006C (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	$FFF4006C (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	$FFF4006C (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	$FFF40070 (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	$FFF40074 (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	Clear external interrupt (parity error)
CMWP	Clear VMEbus master write post error interrupt
CSYSF	Clear VMEbus SYSFAIL interrupt
CAB	Clear ABORT switch interrupt
CACF	Clear VMEbus ACFAIL interrupt

Interrupt Clear Register (bits 16-23)

ADR/SIZ	$FFF40074 (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	$FFF40074 (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	$FFF40078 (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 These bits define the level of the abort interrupt.

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

Interrupt Level Register 1 (bits 16-23)

ADR/SIZ	$FFF40078 (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	$FFF40078 (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 These bits define the level of the external (parity error) interrupt.

Interrupt Level Register 1 (bits 0-7)

ADR/SIZ	$FFF40078 (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	$FFF4007C (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	$FFF4007C (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	$FFF4007C (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	$FFF4007C (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	$FFF40080 (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	$FFF40080 (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	$FFF40080 (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	$FFF40080 (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	$FFF40084 (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	$FFF40084 (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	$FFF40084 (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	$FFF40084 (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	$FFF40088 (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 for the VMEchip2 is: VBR0 = 6, VBR1 = 7 (i.e., setting the Vector Base Register at address $FFF40088 to $67*xxxxxx*). This produces a Vector Base0 of $60 corresponding to the "X" in Table 2-3, and a Vector Base1 of $70 corresponding to the "Y" in Table 2-3.

I/O Control Register 1

ADR/SIZ	$FFF40088 (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 four General Purpose I/O pins (GPIO0-3).

Caution **GPIO0 should not be programmed as an output. GPIO3 on the MVME166 should not be programmed as an output. Refer to the I/O Control Register 2 description.**

GPOEN0 When this bit is low, the GPIO0 pin is an input. When this bit is high, the GPIO0 pin is an output.

GPOEN1 When this bit is low, the GPIO1 pin is an input. When this bit is high, the GPIO1 pin is an output.

GPOEN2 When this bit is low, the GPIO2 pin is an input. When this bit is high, the GPIO2 pin is an output.

GPOEN3 When this bit is low, the GPIO3 pin is an input. When this bit is high, the GPIO3 pin is an output.

ABRTL This bit indicates the status of the ABORT switch. When this bit is high, the ABORT switch is depressed. When this bit is low, the ABORT switch is not depressed.

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	$FFF40088 (8 bits of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	GPIOO3	GPIOO2	GPIOO1	GPIOO0	GPIOI3	GPIOI2	GPIOI1	GPIOI0
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).

GPIOI0 When this bit is low, the GPIO0 pin is low. When this bit is high, the GPIO0 pin is high.

GPIOI1 When this bit is low, the GPIO1 pin is low. When this bit is high, the GPIO1 pin is high.

GPIOI2 When this bit is low, the GPIO2 pin is low. When this bit is high, the GPIO2 pin is high.

GPIOI3 When this bit is low, the GPIO3 pin is low. When this bit is high, the GPIO3 pin is high.

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

GPIOO0 When this bit is low, the GPIO0 pin is driven low if it is defined as an output. When this bit is high, the GPIO0 pin is driven high if it is defined as an output.

GPIOO1 When this bit is low, the GPIO1 pin is driven low if it is defined as an output. When this bit is high, the GPIO1 pin is driven high if it is defined as an output.

GPIOO2 When this bit is low, the GPIO2 pin is driven low if it is defined as an output. When this bit is high, the GPIO2 pin is driven high if it is defined as an output.

GPIOO3 When this bit is low, the GPIO3 pin is driven low if it is defined as an output. When this bit is high, the GPIO3 pin is driven high if it is defined as an output.

NOTES: The GPIO0 pin on the MVME167/187 is used to monitor the +12 Vdc power to the LAN connector. When the GPIOI0 bit is high, +12 Vdc is not present. When the GPIOI0 bit is low, +12 Vdc is present.

The GPIO0 pin on the MVME166 is used to monitor power to the I/O transition module (MVME712-10). When the GPIO0 bit is low, +5 Vdc, +12 Vdc, and -12 Vdc are available to the transition board. When the GPIO0 bit is high, one of the voltages is not available to the transition board.

Caution

GPIO0 should not be programmed as an output. GPIO3 on the MVME166 should not be programmed as an output.

The GPIO1 pin on the MVME187 is used to control the STAT LED. When the GPIO1 pin is high or programmed as an input, the LED is on. When the GPIO1 pin is low, the LED is off. The GPIO1 pin on the MVME166/167 is connected to the remote reset connector (J3 on the MVME167, J1 on the MVME166) pin 16.

The GPIO2 pin on the MVME166/167/187 is used to control bus error handling by the 82596CA interface logic. Refer to the *82596CA LAN Controller Interface* section in the PCCchip2 description in Chapter 3.

The GPIO3 pin on the MVME167/187 is connected to the remote reset connector J3 pin 18.

The GPIO3 pin on the MVME166 is used to monitor term power on the SCSI bus. When the GPIO3 bit is low, SCSI term power is present on the SCSI bus. When the GPIO3 bit is high, SCSI term power is not present on the SCSI bus.

On the MVME167, the GPIO1 and 3 pins may be used as inputs or outputs. On the MVME187, the GPIO3 pin may be used as an input or output. Because GPIO1 is connected to the STAT LED, it is normally on output. If it is used as an input, the STAT LED follows the input. On the MVME166, the GPIO1 pin may be used as an input or output.

I/O Control Register 3

ADR/SIZ	$FFF40088 (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). On the MVME167/187, the GPI pins are connected to the general purpose jumpers on header J1. (Refer to Chapter 1 for hardware jumpering of these pins.)

GPI0 When this bit is low, the GPI0 pin is low. When this bit is high, the GPI0 pin is high.

GPI1 When this bit is low, the GPI1 pin is low. When this bit is high, the GPI1 pin is high.

GPI2 When this bit is low, the GPI2 pin is low. When this bit is high, the GPI2 pin is high.

GPI3 When this bit is low, the GPI3 pin is low. When this bit is high, the GPI3 pin is high.
On the MVME166 only, if the jumper is in, you can execute 166Bug from the Flash memory; if the jumper is out, the 166Bug stays in the Boot ROM.

GPI4 When this bit is low, the GPI4 pin is low. When this bit is high, the GPI4 pin is high.

GPI5 When this bit is low, the GPI5 pin is low. When this bit is high, the GPI5 pin is high.

GPI6 When this bit is low, the GPI6 pin is low. When this bit is high, the GPI6 pin is high.

GPI7 When this bit is low, the GPI7 pin is low. When this bit is high, the GPI7 pin is high.

Miscellaneous Control Register

ADR/SIZ	$FFF4008C (8 bits of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	MPIRQE N	REVERO M	DISSRA M	DISMST	NOELBB SY	DISBSYT	ENINT	DISBGN
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 When this bit is high, the VME LED on the MVME166/167/187 is lit when local bus reset is asserted or the VMEchip2 is driving local bus busy. When this bit is low, the VME LED on the MVME166/167/187 is lit when local bus reset is asserted, the VMEchip2 is driving local bus busy, or the VMEchip2 is driving the VMEbus address strobe.

DISSRAM When this bit is high, the SRAM decoder in the VMEchip2 is disabled. When this bit is low, the SRAM decoder in the VMEchip2 is enabled. This bit should not be set on the MVME166/167/187.

REVEROM For the MVME167/187: when this bit is high, the EPROM is disabled; when this bit is low, the EPROM is enabled. This bit should not be set on the MVME167/187.

For the MVME166: This bit is used to enable/disable the FLASH ROM on the MVME166. When this bit is high, the FLASH ROM is enabled. When this bit is low, the FLASH ROM is disabled. This bit should be set, after the ROM0 bit is cleared ($FFF40030 bit 20).

MPIRQEN This function is not used on the MVME166/167/187. 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 $FFF40100. 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 register*, 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)

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															

VMEchip2 Revision Register

ADR/SIZ	Local bus: $FFF40100/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: $FFF40100/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: $FFF40104/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: $FFF40104/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: $FFF40108/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: $FFF4010C/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: $FFF40110/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: $FFF40114/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: $FFF40118/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: $FFF4011C/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.

PCCchip2 3

Introduction

This chapter defines the peripheral channel controller ASIC which is referred to as the PCCchip2 hereafter. The PCCchip2 is designed for the MVME166/ 167/187 Single Board Computers to interface the local MC68040-bus to various peripheral devices.

Features

- ❑ BBRAM interface with dynamic sizing support.
- ❑ Map decoder for MEMC040 Memory Controller ASIC.
- ❑ 8-bit parallel I/O port.
- ❑ Master and slave interface for CD2401 Intelligent Multi-Protocol Peripheral.
- ❑ Host interface to Intel 82596CA LAN Coprocessor.
- ❑ Host interface to NCR SCSI I/O Processor.
- ❑ Two 32-bit tick timers.
- ❑ Interrupt handler for tick timers and all peripherals:
  - All interrupts are level-programmable.
  - All interrupts are maskable.
  - All interrupts provide a unique vector.
- ❑ Interrupt Mask Register to help prioritize interrupt requests to the MC88100.

Functional Description

The following sections provide an overview of the functions provided by the PCCchip2. A detailed programming model for the PCCchip2 control and status registers is provided in a later section.

General Description

The PCCchip2 interfaces the MC68040 microprocessor bus to the local peripherals on the MVME166/167/187 Single Board Computers including: battery-backed RAM, Serial Communications Controller (CL-CD2401), LAN controller (82596CA), SCSI controller (NCR53C710), and the Memory

Controller ASIC (MEMC040). The PCCchip2 also provides two 32-bit timers and a parallel I/O port. The block diagram of the PCCchip2 is shown as Figure 3-1.

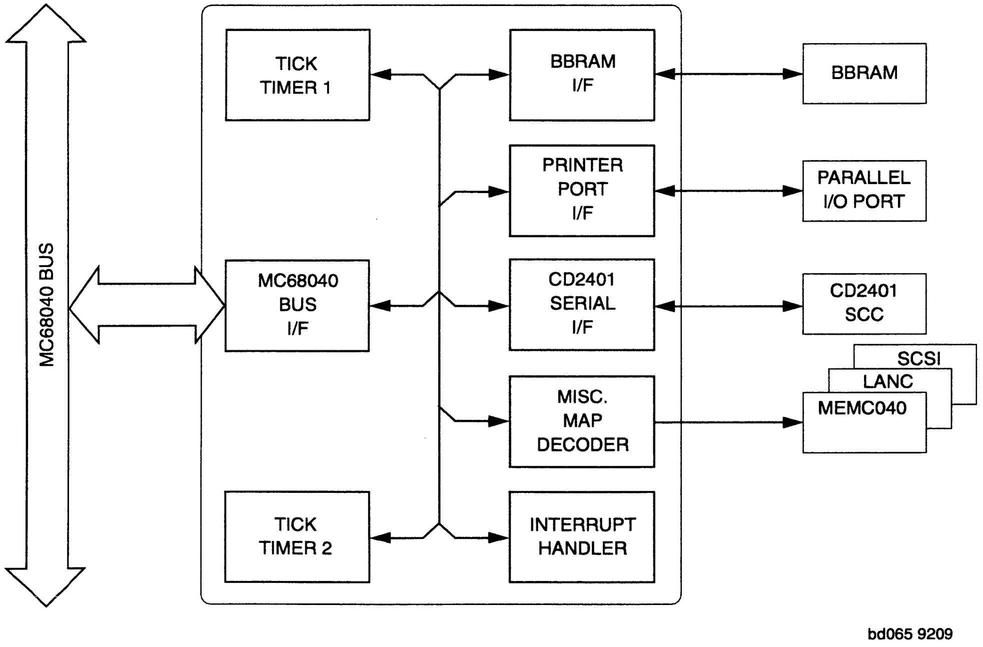


Figure 3-1. PCCchip2 Block Diagram

BBRAM Interface

The PCCchip2 provides a read/write interface to the BBRAM by any bus master on the MC68040 bus. The PCCchip2 performs dynamic sizing for accesses to the 8-bit BBRAM to make it appear contiguous. This feature allows code to be executable from the BBRAM. The BBRAM device access time must be no greater than 5 BCLK periods in fast mode or 9 BCLK periods in slow mode. The BBRAM speed option is controlled by a control bit in the General Control Register.

Download ROM Interface (MVME166 Only)

The PCCchip2 provides a read/write interface to the download ROM (DROM) for any master on the MC68040 bus. The PCCchip2 performs dynamic sizing for accesses to the 8-bit DROM to make it appear contiguous. This feature allows code to be executable from the DROM. The DROM device access time must be no greater than 5 BCLK periods in fast mode or 9 BCLK periods in slow mode. The DROM speed option is controlled by a control bit in the General Control Register.

If the DR0 bit is set in the General Control Register, DROM appears at locations $00000000 through $0001FFFF in addition to its normal address range.

DR0 is normally cleared at local or power-up reset. However, if no other device responds to the first memory access on the local MC68040, after reset, the PCCchip2 sets DR0, causing DROM to respond to the memory access. DR0 remains set until software writes a 0 to it. (The PCCchip2 determines that no device is responding to the vector fetch by detecting the lack of TA* or TEA* for 32 BCLK cycles after the assertion of TS*.)

82596CA LAN Controller Interface

The LAN controller interface is described in the following sections.

MPU Port and MPU Channel Attention

The PCCchip2 allows the MC68040 bus master to communicate directly with the Intel 82596CA LAN Coprocessor by providing a map decoder and required control and timing logic. Two types of direct access are feasible with the 82596CA: MPU Port and MPU Attention.

MPU Port access enables the MPU to write to an internal, 32-bit 82596CA command register. This allows the MPU to do four things:

1. Write an alternate System Configuration Pointer address.
2. Write an alternative dump area pointer and perform a dump.
3. Execute a software reset.
4. Execute a self-test.

Each Port access must consist of two 16-bit writes: Upper Command Word (two bytes) and Lower Command Word (two bytes). The Upper Command Word (two bytes) is mapped at $FFF46000 and the Lower Command Word (two bytes) is mapped at $FFF46002.

The PCCchip2 only supports (decodes) MPU Port writes. It does not decode MPU Port reads. (Nor does the 82596CA support MPU Port reads.)

MPU Channel Attention access is used to cause the 82596CA to begin executing memory resident Command blocks. To execute an MPU Channel Attention, the MC68040-bus master performs a simple read or write to address $FFF46004.

MC68040-Bus Master Support for 82596CA

The 82596CA has DMA capability with an Intel i486-bus interface. When it is the local bus master, external hardware is needed to convert its bus cycles into MC68040-bus cycles. When the 82596CA has local bus mastership, the PCCchip2 drives the following MC68040 signal lines:

- ❑ Snoop Control SC1-SC0. (With the value programmed into the LAN Interrupt Control Register.)
- ❑ Transfer Types TT1-TT0. (With the value of %00.)
- ❑ Transfer Modifiers TM2-TM0. (With the value of %101.)
- ❑ Transfer Acknowledge (TA*) if Transfer Error Acknowledge (TEA*) is detected.

LANC Bus Error

The 82596CA does not provide a way to terminate a bus cycle with an error indication. The interface to the 82596CA on the MVME166/167/187 provides several ways of processing bus errors that occur while the 82596CA is local bus master. These options are controlled by registers in the VMEchip2 and the PCCchip2.

The GPIO2 signal on the VMEchip2 LCSR (address $FFF40088) controls how the 82596CA interface logic responds to bus errors. If the GPIO2 signal is programmed as an input (reset state) or programmed as an output and set high, bus errors are processed in the following way. (NOTE: this option is not supported on Rev. A and Rev. B artwork boards.)

The 82596CA interface logic monitors all bus cycles initiated by the 82596CA, and if a bus error is indicated (TEA* = 0 and TA* = 1), the Back Off signal (BOFF*) to the 82596CA is asserted to keep the 82596CA off the local bus and prevent it from transmitting bad data or corrupting local memory. The LANC Error Status Register in the PCCchip2 is updated and a LANC bus error interrupt is generated if it is enabled in the PCCchip2. The Back Off signal remains asserted until the 82596CA is reset via a port reset command. After the 82596CA is reset, pending operations must be restarted.

If the GPIO2 signal is programmed as an output and set low, bus errors are processed in the following way. The 82596CA interface logic monitors all bus cycles initiated by the 82596CA, and if a bus error is indicated (TEA* = 0 and TA* = 1), the interface logic asserts the TA* signal to terminate the bus cycle. The LANC Error Status Register in the PCCchip2 is updated and a LANC bus error interrupt is generated if it is enabled in the PCCchip2. In this case the 82596CA continues to operate and because the cycle was terminated with an error, the 82596CA may transmit bad data or corrupt memory.

LANC Interrupt

When the PCCchip2 detects a high level on the INT signal from the 82596CA, if such interrupts are enabled, it generates an interrupt to the MPU.

If the C040 bit is set, the interrupt request goes to the MPU via the EIPL* pins at the level that is programmed for LANC interrupts in the LANC Interrupt Control Register.

If the C040 bit is cleared, the interrupt goes to the MPU via the INT pin (if the level that is programmed for LANC interrupts in the LANC Interrupt Control Register is higher than the level set in the Interrupt Mask Level Register).

When the MPU acknowledges the LANC interrupt, the PCCchip2 responds with the vector that corresponds to LANC interrupts.

53C710 SCSI Controller Interface

The PCCchip2 provides a map decoder and an interrupt handler for the NCR-53C710 SCSI I/O Processor. The base address for the 53C710 is $FFF47000.

When the PCCchip2 detects low a level on the IRQ* line from the 53C710, if such interrupts are enabled, it generates an interrupt to the MPU.

If the C040 bit is set, the interrupt request goes to the MPU via the EIPL* pins at the level that is programmed for SCSI interrupts in the SCSI Interrupt Control Register.

If the C040 bit is cleared, the interrupt goes to the MPU via the INT pin (if the level that is programmed for SCSI interrupts in the SCSI Interrupt Control Register is higher than the level set in the Interrupt Mask Level Register).

Memory Controller MEMC040 Interface

The PCCchip2 decodes the address for accesses to the memory controller MEMC040. The base address for the MEMC040 is $FFF43000.

Parallel Port Interface

The PCCchip2 provides an 8/16-bit bidirectional parallel port. All eight/sixteen bits of the port must be either inputs or outputs (no individual selection). In addition to the 8/16 bits of data, there are two control pins and five status pins. Each of the status pins can generate an interrupt to the MPU in any of the following programmable conditions: high level, low level, high-to-low transition, or low-to-high transition. This port may be used as a parallel printer port or as a general parallel I/O port.

When used as a parallel printer port, the five status pins function as: Printer Acknowledge (ACK), Printer Fault (FAULT*), Printer Busy (BSY), Printer Select (SELECT), and Printer Paper Error (PE); while the control pins act as Printer Strobe (STROBE*), and Input Prime (INP*).

The PCCchip2 provides an auto-strobe feature similar to that of the MVME147 PCC. In auto-strobe mode, after a write to the Printer Data Register, the PCCchip2 automatically asserts the STROBE* pin for a selected time specified by the Printer Fast Strobe control bit. In manual mode, the Printer Strobe control bit directly controls the state of the STROBE* pin.

General Purpose I/O Pin

The General Purpose I/O pin can be used as an input pin, as an output pin, or as both. The PCCchip2 has a status bit that reflects the state of the pin. The PCCchip2 also has a control bit that allows it to drive the pin, and another control bit that controls the level that is driven.

The input can be configured to generate an interrupt to the MPU in any of the following programmable conditions: high level, low level, high-to-low transition, or low-to-high transition.

CD2401 SCC Interface

The PCCchip2 provides the required logic to interface the CL-CD2401 (SCC) Intelligent MultiProtocol Peripheral to the MC68040 bus. The interface logic consists of a local master interface, a local slave interface, a CD2401 Host interface, a CD2401 DMA interface, a CD2401 interrupt handler, and a local bus requester.

The base address for the CL-CD2401 is $FFF45000. It has 8- and 16-bit registers only. Consequently it does not respond when accessed with a size of 4 bytes (SIZ1,0 = %00) or with a size of 16 bytes (SIZ1,0 = %11).

There are three interrupts sources from the SCC: receive interrupt, transmit interrupt, and modem interrupt. The PCCchip2 provides the ability to individually program the priority level of each of these interrupt sources.

When the C040 bit is set, these interrupts are sent to the MPU via the EIPL* pins (at the programmed level).

When the C040 bit is cleared, they are sent to the MPU via the INT pin. (The INT pin is only asserted if the programmed level of the interrupt source is higher than the level programmed into the Interrupt Mask Level Register.)

There are two interrupt acknowledge modes supported by the PCCchip2 for the SCC: auto vector and direct. In auto vector mode, the PCCchip2 supplies the interrupt vector to the MPU. (No interrupt acknowledge cycle is seen by the CD2401.) In direct mode, the SCC supplies the vector to the MPU. (The PCCchip2 passes the interrupt acknowledge cycle on through to the CD2401. Note that the PCCchip2 drives the CD2401 A7-A0 pins with $01 for modem interrupt acknowledges, $02 for transmit interrupt acknowledges and $03 for receive interrupt acknowledges.) The use of the auto vector mode is not recommended because the CD2401 can supply the vector and the CD2401 requires an interrupt acknowledge cycle.

In order to support polling with the CD2401, the PCCchip2 supports pseudo interrupt acknowledge (PIACK) cycles to the CD2401. (This is required since the CD2401 has no other way of clearing its interrupt requests.) PIACK cycles happen as follows:

1. The MPU waits for an IRQ bit to be set in one of the three SCC interrupt control registers.
2. The local bus master starts a normal read cycle to one of the three PIACK registers in the PCCchip2. (The three PIACK registers correspond to modem, transmit, and receive interrupts respectively.)
3. The PCCchip2 upon detecting the start of the read, performs an interrupt acknowledge cycle to the CD2401. (The PCCchip2 drives the CD2401 A7 through A0 pins with a value that corresponds to the PIACK register that is being read. If the Modem PIACK Register is being read, then A7 through A0 = $01. If the Transmit PIACK Register is being read, then A7 through A0 = $02. If the Receive PIACK Register is being read, then A7 through A0 = $03.)
4. As the interrupt acknowledge cycle completes, the PCCchip2 places the vector being driven by the CD2401 onto the local bus D0 through D8 and D16 through D23 signals. (From the MPU point of view, the status read from the selected PCCchip2 PIACK register is the vector from the CD2401.)
5. The PCCchip2 signals to the local MPU (via TA*) that the read cycle is complete.

Interrupt Prioritizer

The PCCchip2 provides circuitry to support a seven level, priority, interrupt scheme, when the local MPU is an MC88100. This support circuit is enabled/disabled by the C040 bit in the General Control Register.

When the C040 bit is set, the PCCchip2 drives the level of its highest priority internal interrupt request onto the EIPL<2..0>* pins. It is intended that the EIPL pins be combined outside the chip with any external IPL signals and driven externally to the MC68040. The priority interrupt scheme is assumed to be provided in the MC68040 when the C040 bit is set.

When the C040 bit is cleared, the PCCchip2 receives the level that is being driven onto the EIPL<2..0>* pins by external devices. It is intended that all external interrupt sources be combined externally onto the incoming EIPL pins, and that the INT pin from the PCCchip2 be synchronized externally then connected to the INT pin on the MC88100. The PCCchip2 combines the incoming EIPL level with its internal interrupt levels and compares that combined level with the level that is programmed into the Mask Register. If the combined internal and external interrupt request levels are greater than the mask level, the PCCchip2 asserts the INT pin to the MC88100. There are two actions that cause the INT pin to be asserted:

1. The combined internal and external interrupt request level transitions to a level higher than the mask level programmed in the Mask Register.
2. The MPU writes a mask level into the Mask Register that is lower than the current combined internal and external interrupt request level.

Tick Timer

The PCCchip2 includes two 32-bit general purpose tick timers. The tick timers run on a 1MHz clock which is derived from the processor clock by a prescaler.

Each tick timer has a 32-bit counter, a 32-bit compare register, and a clear-on-compare enable bit. The counter is readable and writable at any time. These timers can be used to generate interrupts at various rates or the counters can be read at various times for interval timing. There are two modes of operation for these timers: free-running and clear-on-compare.

In free-running mode, the timers have a resolution of 1 µs and roll over after the count reaches the maximum value $FFFFFFFF. The rollover period for the timers is 71.6 minutes.

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.

Overall Memory Map

The following memory map includes all devices selected by the PCCchip2 map decoders, including those internal to the chip and those external. These devices respond only when the Transfer Type signals carry the values of %00 or %01 which correspond to Normal and MOVE16 accesses on the MC68040 local bus.

Table 3-1. PCCchip2 Devices Memory Map

Address Range	Selected Device	Comments
$FFF42000-$FFF4203F	PCCchip2 Registers	See Programming Model
$FFF42040-$FFF42FFF	PCCchip2 Registers	Repeated
$FFF43000-$FFF43FFF	MEMC040 (Memory Controller)	External Device
$FFF45000-$FFF450FF	CD2401 (SCC)	External Device
$FFF45100-$FFF45FFF	CD2401 (SCC)	Repeated
$FFF46000-$FFF46FFF	82596CA (LANC)	External Device
$FFF47000-$FFF47FFF	53C710 (SCSI)	External Device
$FFF80000-$FFF9FFFF	DROM (MVME166 only)	External Device
$FFFC0000-$FFFCFFFF	MK48T08 (BBRAM, TOD Clock)	External Device

Programming Model

This section defines the programming model for the control and status registers (CSR) in the PCCchip2. The base address of the CSR is $FFF42000. The PCCchip2 control and status registers can be accessed as bytes (8 bits), two-bytes (16 bits), or four-bytes(32 bits). The possible operations for each bit in the CSR are 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.

0 This bit is read only. It always reads as 0.

The possible states of the bits after local and power-up reset are as defined below.

P The bit is affected by power-up reset.

L The bit is affected by local reset.

X The bit is not affected by reset.

V The effect of reset on this bit is variable.

0 The bit is always 0.

1 The bit is always 1.

A summary of the PCCchip2 CSR is shown in Table 3-2.

Chip ID Register

The Chip ID Register is located at $FFF42000. It is an 8-bit read-only register that is hard-wired to a hexadecimal value of $20. Writes to this register are ignored; however, the PCCchip2 always terminates the cycles properly with TA*.

ADR/SIZ	$FFF42000 (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	CID7	CID6	CID5	CID4	CID3	CID2	CID1	CID0
OPER	R	R	R	R	R	R	R	R
RESET	0	0	1	0	0	0	0	0

Chip Revision Register

The Chip Revision Register is located at $FFF42001. It is an 8-bit read-only register that is hard-wired to reflect the revision level of the PCCchip2 ASIC. The current value of this register is $00. Writes to this register are ignored; however, the PCCchip2 always terminates the cycles properly with TA*.

ADR/SIZ	$FFF42001 (8 bits)							
BIT	23	22	21	20	19	18	17	16
NAME	REV7	REV6	REV5	REV4	REV3	REV2	REV1	REV0
OPER	R	R	R	R	R	R	R	R
RESET	0	0	0	0	0	0	0	0

General Control Register

The General Control Register is located at $FFF42002. It is an 8-bit register that controls chip general functions. The Master Interrupt Enable bit (MIEN) must be set high for any interrupts from the PCCchip2 to be asserted to the processor.

ADR/SIZ	$FFF42002 (8 bits)							
DIR	15	14	13	12	11	10	9	8
NAME	DR0					C040	MIEN	FAST
OPER	R/W	R	R	R	R	R/W	R/W	R/W
RESET	V PL	0	0	0	0	0 P	0 PL	0 P

Note: V=1 if no other device responds to the first memory access after Power-up or Local Reset. Otherwise V=0.

Table 3-2. PCCchip2 Memory Map - Control and Status Registers

PCCchip2 Base Address = $FFF42000
OFFSET:

	D31							D24	D23							D16
00	CHIP ID								CHIP REVISION							
04	TIC TIMER 1															
08	TIC TIMER 1															
0C	TIC TIMER 2															
10	TIC TIMER 2															
14	PRESCALER COUNT REGISTER								PRESCALER CLOCK ADJUST							
18	GPI PLTY	GPI E/L*	GPI INT	GPI IEN	GPI ICLR	GPI IRQ LEVEL								GPI	GPOE	GPO
1C				SCC RTRY ERR	SCC PAR ERR	SCC EXT ERR	SCC LTO ERR	SCC SCLR			SCC MDM ERR	SCC MDM IEN	SCC MDM AVEC	SCC MODEM IRQ LEVEL		
20																
24									SCC TRANSMIT PIACK							
28					LAN PAR ERR	LAN EXT ERR	LAN LTO ERR	LAN SCLR								
2C					SCSI PAR ERR	SCSI EXT ERR	SCSI LTO ERR	SCSI SCLR								
30	PRTR ACK PLTY	PRTR ACK E/L*	PRTR ACK INT	PRTR ACK IEN	PRTR ACK ICLR	PRTR ACK IRQ LEVEL			PRTR FLT PLTY	PRTR FLT E/L*	PRTR FLT INT	PRTR FLT IEN	PRTR FLT ICLR	PRTR FAULT IRQ LEVEL		
34	PRTR BSY PLTY	PRTR BSY E/L*	PRTR BSY INT	PRTR BSY IEN	PRTR BSY ICLR	PRTR BSY IRQ LEVEL										
38	CHIP SPEED															
3C																

SCC PROVIDES ITS OWN VECTORS

This sheet continues on facing page. ⟶

D15							D8	D7							D0
DRO					CPU 040	MSTR INT EN	FAST BRAM	VECTOR BASE REGISTER							
COMPARE REGISTER															
COUNTER REGISTER															
COMPARE REGISTER															
COUNTER REGISTER															
OVERFLOW COUNTER 2					CLR OVF 2	COC EN 2	TIC EN 2	OVERFLOW COUNTER 1					CLR OVF 1	COC EN 1	TIC EN 1
		TIC2 INT	TIC2 IEN	TIC2 ICLR	TIC TIMER 2 IRQ LEVEL					TIC1 INT	TIC1 IEN	TIC1 ICLR	TIC TIMER 1 IRQ LEVEL		
		SCC TX IRQ	SCC TX IEN	SCC TX AVEC	SCC TRANSMIT IRQ LEVEL			SCC SC1	SCC SC0	SCC RX IRQ	SCC RX IEN	SCC RX AVEC	SCC RECEIVE IRQ LEVEL		
								SCC MODEM PIACK							
								SCC RECEIVE PIACK							
LAN INT PLTY	LAN INT E/L*	LAN INT	LAN IEN	LAN ICLR	LAN INT IRQ LEVEL			LAN SC1	LAN SC0	LAN ERR INT	LAN ERR IEN	LAN ERR ICLR	LAN ERR IRQ LEVEL		
										SCSI IRQ	SCSI IEN		SCSI INT IRQ LEVEL		
PRTR SEL PLTY	PRTR SEL E/L*	PRTR SEL INT	PRTR SEL IEN	PRTR SEL ICLR	PRTR SEL IRQ LEVEL			PRTR PE PLTY	PRTR PE E/L*	PRTR PE INT	PRTR PE IEN	PRTR PE ICLR	PRTR PE IRQ LEVEL		
PRTR ANY INT			PRTR ACK	PRTR FLT	PRTR SEL	PRTR PE	PRTR BSY				PRTR DAT ENBL	PRTR INP	PRTR STB	PRTR FAST ASTB	PRTR MAN STB
PRINTER DATA															
					INTERRUPT IPL LEVEL								INTERRUPT MASK LEVEL		

1362 9403

← This sheet begins on facing page.

FAST

This control bit tailors the control circuit for BBRAM to the speed of BBRAM.

When operating at 25 MHz, the FAST bit should be cleared for devices with access times longer than 200 ns (5 CLK cycles). The bit can be set for devices that have access times of 200 ns or faster. It is not allowed to use devices slower than 360 ns (9 CLK cycles), at 25 MHz.

When operating at 33 MHz, the FAST bit should be cleared for devices with access times longer than 150 ns (5 CLK cycles). The bit can be set for devices that have access times 150 ns or faster. It is not allowed to use devices slower than 270 ns (9 CLK cycles), at 33 MHz.

MIEN

Master Interrupt Enable. When this bit is high, interrupts from and via the PCCchip2 are allowed to reach the MPU. When it is low, all interrupts from the PCCchip2 are disabled (this includes both the EIPL* pins and the INT pin). Also, when the bit is low, all interrupt acknowledge cycles to the PCCchip2 are passed on, via the IACKOUT* pin. This bit is cleared by a reset.

C040

CPU040. This bit should be set when the MPU is an MC68040. It should be cleared when the MPU is an MC88100. When the bit is set, EIPL<2..0>* are driven as outputs which carry the priority encoded interrupt request from the PCCchip2 interrupt sources. When the bit is cleared, EIPL<2..0>* are not driven as outputs, but are inputs only.

DR0

Download ROM at 0 (MVME166 only). When this bit is cleared, DROM appears only in its normal address range. When DR0 is set, DROM also appears at $00000000 through $0001FFFF. DR0 is cleared by power-up or local reset, but if no other device responds (within a certain amount of time) to the first memory access after the reset, then the PCCchip2 sets DR0. This causes the DROM to respond to the memory access (and all memory accesses thereafter until software clears DR0).

Vector Base Register

The Interrupt Vector Base Register is located at $FFF42003. It is an 8-bit read/write register that is used to supply the vector to the MPU during an interrupt acknowledge cycle for: the two internal tick timers, LAN interrupt, LAN BERR interrupt, SCSI interrupt, GPIO interrupt, and parallel port interrupts. Only the most significant four bits are used. The least significant four bits encode the interrupt source during the acknowledge cycle. The exception to this is that after reset occurs, the interrupt vector passed is $0F, which remains in effect until a write is generated to the Vector Base Register.

A normal read access to the Vector Base Register yields the value $0F if the read happens before it has been initialized. A normal read access yields all 0s on bits 0-3 and the value that was last written on bits 4-7 if the read happens after the Vector Base Register has been initialized. A suggested setting of the Vector Base Register is $50.

ADR/SIZ	$FFF42003 (8 bits)							
BIT	7	6	5	4	3	2	1	0
NAME	IV7	IV6	IV5	IV4	IV3	IV2	IV1	IV0
OPER	R/W	R/W	R/W	R/W	R	R	R	R
RESET	0 PL	0 PL	0 PL	0 PL	1 PL	1 PL	1 PL	1 PL

The encoding for the interrupt sources is shown below, where IV3-IV0 refer to bits 3-0 of the vector passed during the IACK cycle:

Interrupt Source	IV3-IV0	Priority
Printer Port-BSY	$0	Lowest
Printer Port-PE	$1	
Printer Port-SELECT	$2	
Printer Port-FAULT	$3	
Printer Port-ACK	$4	
SCSI IRQ	$5	
LANC ERR	$6	
LANC IRQ	$7	
Tick Timer 2 IRQ	$8	
Tick Timer 1 IRQ	$9	
GPIO IRQ	$A	
Serial Modem IRQ (auto vector mode only)	$B	
Serial RX IRQ (auto vector mode only)	$C	
Serial TX IRQ (auto vector mode only)	$D	Highest

3

The PCCchip2 supports an auto vector mode for the Cirrus Logic CD2401 SCC serial port. (Refer to the AVEC bit in the following registers: SCC Modem Interrupt Control Register, SCC Transmit Interrupt Control Register, and SCC Receive Interrupt Control Register.) If this mode is disabled by setting the AVEC bits to 0, then the PCCchip2 obtains the vector from the SCC and passes it to the MPU. Using the auto vector mode is *NOT* recommended.

A suggested setting of the Local Interrupt Vector Register in the SCC chip is $5C. This produces the following vectors:

$5C	Serial RX Exception IRQ
$5D	Serial Modem IRQ
$5E	Serial TX IRQ
$5F	Serial RX IRQ

Programming the Tick Timers

This section provides addresses and bit level descriptions of the prescaler, tick timers, and various other timer registers.

Tick Timer 1 Compare Register

The Tick Timer 1 Compare Register is a 32-bit register located at $FFF42004. The count value of Tick Timer 1 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.

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. The rollover time for the counter is 71.6 minutes.

ADR/SIZ	$FFF42004 (32 bits)		
BIT	31	. . .	0
NAME	Tick Timer 1 Compare Register		
OPER	R/W		
RESET	0 P		

Tick Timer 1 Counter

The Tick Timer 1 Counter is a 32-bit read/write register located at address $FFF42008. When enabled, it increments every microsecond. Software may read or write the counter at any time.

ADR/SIZ	$FFF42008 (32 bits)		
BIT	31	. . .	0
NAME	Tick Timer 1 Counter		
OPER	R/W		
RESET	X		

Tick Timer 2 Compare Register

The Tick Timer 2 Compare Register is a 32-bit register located at $FFF4200C. The count value of Tick Timer 2 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.

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. The rollover time for the counter is 71.6 minutes.

ADR/SIZ	$FFF4200C (32 bits)		
BIT	31	. . .	0
NAME	Tick Timer 2 Compare Register		
OPER	R/W		
RESET	0 P		

Tick Timer 2 Counter

The Tick Timer 2 Counter is a 32-bit read/write register located at address $FFF42010. When enabled, it increments every microsecond. Software may read or write the counter at any time.

ADR/SIZ	$FFF42010 (32 bits)		
BIT	31	. . .	0
NAME	Tick Timer 2 Counter		
OPER	R/W		
RESET	X		

Prescaler Count Register

The Prescaler Count Register is an 8-bit counter used to generate the 1 MHz clock for the two tick timers. This register is a read-only register located at address $FFF42014. It increments to $FF at the BCLK frequency, then it is loaded from the Prescaler Clock Adjust Register.

ADR/SIZ	$FFF42014 (8 bits)		
BIT	31	. . .	24
NAME	Prescaler Count		
OPER	R/W		
RESET	X		

Prescaler Clock Adjust Register

The Prescaler Clock Adjust Register is an 8-bit read/write register located at address $FFF42015. It is required to adjust the prescaler so that it maintains a 1 MHz clock source for the tick timers, regardless of what frequency is used for BCLK. To provide a 1 MHz clock to the tick timers, the prescaler adjust register should be programmed based on the following equation:

prescaler clock adjust register = 256 - BCLK (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 tick timers. The tick timer clock can be derived by the following equation.

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

The maximum clock frequency for the tick timers is the BCLK frequency divided by two. The value 255 ($FF) is not allowed to be programmed into this register. If a write with the value of $FF occurs to this register, the PCCchip2 terminates the cycle properly with TA*, but the register remains unchanged.

ADR/SIZ	$FFF42015 (8 bits)		
BIT	23	. . .	16
NAME	Prescaler Clock Adjust		
OPER	R/W		
RESET	$DF P		

Tick Timer 2 Control Register

This is an 8-bit read/write register that controls Tick Timer 2. It is located at address $FFF42016.

ADR/SIZ	$FFF42016 (8 bits)							
BIT	15	14	13	12	11	10	9	8
NAME	OVF3	OVF2	OVF1	OVF0		COVF	COC	CEN
OPER	R	R	R	R	R	C	R/W	R/W
RESET	0 PL	0 PL	0 PL	0 PL	0	0 PL	0 PL	0 PL

CEN Counter Enable. When this bit is high, the counter increments. When this bit is low, the counter does not increment.

COC Clear On Compare. 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 Clear Overflow Counter. The overflow counter is cleared when a one is written to this bit.

OVF3-OVF0 These four bits are the outputs 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 control bit.

Tick Timer 1 Control Register

This is an 8-bit read/write register that controls Tick Timer 1. It is located at address $FFF42017.

ADR/SIZ	$FFF42017 (8 bits)							
BIT	7	6	5	4	3	2	1	0
NAME	OVF3	OVF2	OVF1	OVF0		COVF	COC	CEN
OPER	R	R	R	R	R	C	R/W	R/W
RESET	0 PL	0 PL	0 PL	0 PL	0	0 PL	0 PL	0 PL

CEN Counter Enable. When this bit is high, the counter increments. When this bit is low, the counter does not increment.

COC Clear On Compare. 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 Clear Overflow Counter. The overflow counter is cleared when a one is written to this bit.

OVF3-OVF0 These four bits are the outputs 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 control bit.

General Purpose Input Interrupt Control Register

ADR/SIZ	$FFF42018 (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	PLTY	E/L*	INT	IEN	ICLR	IL2	IL1	IL0
OPER	R/W	R/W	R	R/W	C	R/W	R/W	R/W
RESET	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL

IL2-IL0 These three bits select the interrupt level for the general purpose input/output (GPIO) pin. Level 0 does not generate an interrupt.

ICLR In edge-sensitive mode, writing a logic 1 to this bit clears the INT status bit. This bit has no function in level-sensitive mode. This bit is always read as zero.

IEN When this bit is high, the interrupt is enabled. The interrupt is disabled when this bit is low.

INT When this bit is high, a general purpose input interrupt is being generated at the level programmed in IL2-IL0 (if nonzero).

E/L* When this bit is high, the interrupt is edge-sensitive. The interrupt is level-sensitive when this bit is low.

PLTY When this bit is low, interrupt is activated by either a rising edge on the GPIO pin or a high level on the GPIO pin (depending on the E/L* bit.) When this bit is high, interrupt is activated by either a falling edge on the GPIO pin or a low level of the GPIO pin (depending on the E/L* bit.) Note that if this bit is changed while the E/L* bit is set (or is being set), a GPIO interrupt may be generated. This can be avoided by setting the ICLR bit during write cycles that change the E/L* bit.

General Purpose Input/Output Pin Control Register

ADR/SIZ	$FFF42019 (8 bits)							
BIT	23	22	21	20	19	18	17	16
NAME						GPI	GPOE	GPO
OPER	R	R	R	R	R	R	R/W	R/W
RESET	0	0	0	0	0	X	0 PL	0 PL

GPO When GPO is set, and GPOE is set, the GPIO pin is at a logic high level. When GPO is cleared, and GPOE is set, the GPIO pin is at a logic low level.

GPOE This bit controls whether or not the PCCchip2 drives the GPIO pin. When GPOE is set, the PCCchip2 drives the GPIO pin. When GPOE is cleared, the PCCchip2 does not drive the GPIO pin.

GPI This bit reflects the state of the GPIO pin. It is set when GPIO is high and cleared when GPIO is low. On the MVME166/167/187, the PCCGPIO1 pin is connected to the remote reset connector J3 pin 19.

Tick Timer 2 Interrupt Control Register

ADR/SIZ	$FFF4201A (8 bits)							
BIT	15	14	13	12	11	10	9	8
NAME			INT	IEN	ICLR	IL2	IL1	IL0
OPER	R	R	R	R/W	C	R/W	R/W	R/W
RESET	0	0	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL

IL2-IL0 Interrupt Request Level. These three bits select the interrupt level for Tick Timer 2. Level 0 does not generate an interrupt.

ICLR Writing a logic 1 into this bit clears the INT status bit. This bit is always read as zero.

IEN Interrupt Enable. When this bit is high, the interrupt is enabled. The interrupt is disabled when this bit is low.

INT Interrupt Status. When this bit is high a Tick Timer 2 interrupt is being generated at the level programmed in IL2-IL0 (if nonzero). This bit is edge-sensitive and can be cleared by writing a logic 1 into the ICLR control bit.

Tick Timer 1 Interrupt Control Register

ADR/SIZ	$FFF4201B (8 bits)							
BIT	7	6	5	4	3	2	1	0
NAME			INT	IEN	ICLR	IL2	IL1	IL0
OPER	R	R	R	R/W	C	R/W	R/W	R/W
RESET	0	0	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL

IL2-IL0 Interrupt Request Level. These three bits select the interrupt level for Tick Timer 1. Level 0 does not generate an interrupt.

ICLR Writing a logic 1 into this bit clears the INT status bit. This bit is always read as zero.

IEN Interrupt Enable. When this bit is high, the interrupt is enabled. The interrupt is disabled when this bit is low.

INT Interrupt Status. When this bit is high a Tick Timer 1 interrupt is being generated at the level programmed in IL2-IL0 (if nonzero). This bit is edge-sensitive and can be cleared by writing a logic 1 into the ICLR control bit.

SCC Error Status Register and Interrupt Control Registers

This section provides addresses and bit level descriptions of the SCC interrupt control registers and status registers.

SCC Error Status Register

ADR/SIZ	$FFF4201C (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME				RTRY	PRTY	EXT	LTO	SCLR
OPER				R	R	R	R	W/R-0
RESET	0	0	0	0 PL	0 PL	0 PL	0 PL	0

SCLR Writing a 1 to this bit clears bits 25 through 28 (LTO, EXT, PRTY, and RTRY). Reading this bit always yields 0.

LTO,EXT,PRTY,RTRY

These bits indicate the status of the last local bus error condition encountered by the SCC while performing DMA accesses to the local bus. A local bus error condition is flagged by the assertion of TEA*. When the SCC receives TEA* if the source of the error is local-time-out, then LTO is set and EXT, PRTY, and RTRY are cleared. If the source of the TEA* is due to an error in going to the VMEbus, then EXT is set and the other three status bits are cleared. If the source of the error is DRAM parity check error, then PRTY is set and the other three status bits are cleared. If the source of the TEA* is because a retry was needed, then RTRY is set and the other three status bits are cleared. If the source of the error is none of the above conditions, then all four bits are cleared. Writing a 1 to bit 24 (SCLR) also clears all four bits.

SCC Modem Interrupt Control Register

ADR/SIZ	$FFF4201D (8 bits)							
BIT	23	22	21	20	19	18	17	16
NAME			IRQ	IEN	AVEC	IL2	IL1	IL0
OPER	R	R	R	R/W	R/W	R/W	R/W	R/W
RESET	0	0	X	0 PL	0 PL	0 PL	0 PL	0 PL

IL2-IL0 Interrupt Request Level. These three bits select the interrupt level for SCC modem Interrupt. Level 0 does not generate an interrupt.

AVEC When this bit is high, the PCCchip2 supplies the interrupt vector to the MPU during an IACK for SCC modem interrupt. When this bit is low, the PCCchip2 obtains the vector from the SCC and passes it to the MPU. The use of the AVEC mode is not recommended.

IEN Interrupt Enable. When this bit is high, the interrupt is enabled. The interrupt is disabled when this bit is low.

IRQ Interrupt Status. This status bit reflects the state of the SCC-IRQ1 pin of the CD2401 (qualified by the IEN bit). When this bit is high, an SCC modem interrupt is being generated at the level programmed in IL2-IL0 (if nonzero). This status bit does not need to be cleared, because it is not edge-sensitive.

SCC Transmit Interrupt Control Register

ADR/SIZ	$FFF4201E (8 bits)							
BIT	15	14	13	12	11	10	9	8
NAME			IRQ	IEN	AVEC	IL2	IL1	IL0
OPER	R	R	R	R/W	R/W	R/W	R/W	R/W
RESET	0	0	X	0 PL	0 PL	0 PL	0 PL	0 PL

IL2-IL0 Interrupt Request Level. These three bits select the interrupt level for SCC Transmit Interrupt. Level 0 does not generate an interrupt.

AVEC When this bit is high, the PCCchip2 supplies the interrupt vector to the MPU during an IACK for SCC transmit interrupt. When this bit is low, the PCCchip2 obtains the vector from the SCC and passes it to the MPU. The use of the AVEC mode is not recommended.

IEN Interrupt Enable. When this bit is high, the interrupt is enabled. The interrupt is disabled when this bit is low.

IRQ Interrupt Status. This status bit reflects the state of the SCC-IRQ2 pin of the CD2401 (qualified by the IEN bit). When this bit is high, an SCC Transmit interrupt is being generated at the level programmed in IL2-IL0 (if nonzero). This status bit does not need to be cleared, because it is not edge-sensitive.

SCC Receive Interrupt Control Register

ADR/SIZ	$FFF4201F (8 bits)							
BIT	7	6	5	4	3	2	1	0
NAME	SC1	SC0	IRQ	IEN	AVEC	IL2	IL1	IL0
OPER	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
RESET	0 PL	0 PL	X	0 PL	0 PL	0 PL	0 PL	0 PL

IL2-IL0 Interrupt Request Level. These three bits select the interrupt level for SCC Receive Interrupt. Level 0 does not generate an interrupt.

AVEC When this bit is high, the PCCchip2 supplies the interrupt vector to the MPU during an IACK for SCC receive interrupt. When this bit is low, the PCCchip2 obtains the vector from the SCC and passes it to the MPU. The use of the AVEC mode is not recommended.

IEN Interrupt Enable. When this bit is high, the interrupt is enabled. The interrupt is disabled when this bit is low.

IRQ Interrupt Status. This status bit reflects the state of the SCC-IRQ3 pin of the CD2401 (qualified by the IEN bit). When this bit is high, an SCC receive interrupt is being generated at the level programmed in IL2-IL0 (if nonzero). This status bit does not need to be cleared, because it is not edge-sensitive.

SC1-SC0 Snoop Control. These control bits determine the value that the PCCchip2 drives onto the local MC68040 bus SC1 and SC0 pins, when the CL-CD2401(SCC) performs DMA accesses. During SCC DMA, when bit SC0 is 0 local bus pin SC0 is low, and when bit SC0 is 1, pin SC0 is high. The same relationship holds true for bit and pin SC1. See the M68040 user's manual for details on how it uses the Snoop Control signals. Note that these bits must be 0 on the MVME187.

Modem PIACK Register

ADR/SIZ	$FFF42023 (8 bits)							
BIT	7	6	5	4	3	2	1	0
NAME	MIV7	MIV6	MIV5	MIV4	MIV3	MIV2	MIV1	MIV0
OPER	R	R	R	R	R	R	R	R
RESET	X	X	X	X	X	X	X	X

The Modem PIACK Register is used to execute modem pseudo interrupt acknowledge cycles to the CD2401. When the local bus master initiates a read cycle to this register, the PCCchip2 executes an interrupt acknowledge cycle to the CD2401 with A7-A0 = $01. (Note that the PILR1 register in the CD2401 should be set to the same value ($01) for the interrupt acknowledge cycle to operate properly.) To finish the local read cycle, the PCCchip2 drives the vector received from the CD2401 onto the local data bus, and asserts TA*. Reads to this register are termed pseudo interrupt acknowledge cycles because they are normal read cycles on the local bus side but they are interrupt acknowledge cycles on the CD2401 side of the PCCchip2. They are necessary to support polled mode operation with the CD2401.

Note **If this register is read when an interrupt is not present, the interrupt acknowledge cycle times out with a TEA if the local bus timer is enabled.**

MIV7-MIV0 Modem interrupt vector bits 7-0 reflect the modem interrupt vector driven by the CD2401 to the PCCchip2 during a pseudo interrupt acknowledge cycle.

Transmit PIACK Register

ADR/SIZ	$FFF42025 (8 bits)							
BIT	23	22	21	20	19	18	17	16
NAME	TIV7	TIV6	TIV5	TIV4	TIV3	TIV2	TIV1	TIV0
OPER	R	R	R	R	R	R	R	R
RESET	X	X	X	X	X	X	X	X

The Transmit PIACK Register is used to execute transmit pseudo interrupt acknowledge cycles to the CD2401. When the local bus master initiates a read cycle to this register, the PCCchip2 executes an interrupt acknowledge cycle to the CD2401 with A7-A0 = $02. (Note that the PILR1 register in the CD2401 should be set to the same value ($02) for the interrupt acknowledge cycle to operate properly.) To finish the local read cycle, the PCCchip2 drives the vector received from the CD2401 onto the local data bus, and asserts TA*. Reads to this register are termed pseudo interrupt acknowledge cycles because they are normal read cycles on the local bus side but they are interrupt acknowledge cycles on the CD2401 side of the PCCchip2. They are necessary to support polled mode operation with the CD2401.

Note **If this register is read when an interrupt is not present, the interrupt acknowledge cycle times out with a TEA if the local bus timer is enabled.**

TIV7-TIV0 Transmit Interrupt vector bits 7-0 reflect the transmit interrupt vector driven by the CD2401 to the PCCchip2 during a pseudo interrupt acknowledge cycle.

Receive PIACK Register

ADR/SIZ	$FFF42027 (8 bits)							
BIT	7	6	5	4	3	2	1	0
NAME	RIV7	RIV6	RIV5	RIV4	RIV3	RIV2	RIV1	RIV0
OPER	R	R	R	R	R	R	R	R
RESET	X	X	X	X	X	X	X	X

The Receive PIACK Register is used to execute receive pseudo interrupt acknowledge cycles to the CD2401. When the local bus master initiates a read cycle to this register, the PCCchip2 executes an interrupt acknowledge cycle to the CD2401 with A7-A0 = $03. (Note that the PILR1 register in the CD2401 should be set to the same value ($03) for the interrupt acknowledge cycle to operate properly.) To finish the local read cycle, the PCCchip2 drives the vector received from the CD2401 onto the local data bus, and asserts TA*. Reads to this register are termed pseudo interrupt acknowledge cycles because they are normal read cycles on the local bus side but they are interrupt acknowledge cycles on the CD2401 side of the PCCchip2. They are necessary to support polled mode operation with the CD2401.

Note

If this register is read when an interrupt is not present, the interrupt acknowledge cycle times out with a TEA if the local bus timer is enabled.

RIV7-RIV0 Receive Interrupt vector bits 7-0 reflect the transmit interrupt vector driven by the CD2401 to the PCCchip2 during a pseudo interrupt acknowledge cycle.

LANC Error Status and Interrupt Control Registers

This section provides addresses and bit level descriptions of the LANC interrupt control registers and status register.

LANC Error Status Register

ADR/SIZ	$FFF42028 (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME					PRTY	EXT	LTO	SCLR
OPER	R	R	R	R	R	R	R	W/R-0
RESET	0	0	0	0	0 PL	0 PL	0 PL	0

SCLR Writing a 1 to this bit clears bits 25 through 27 (LTO, EXT, and PRTY). Reading this bit always yields 0.

LTO,EXT,PRTY

These bits indicate the status of the last local bus error condition encountered by the LANC while performing DMA accesses to the local bus. A local bus error condition is flagged by the assertion of TEA*. When the LANC receives TEA* if the source of the error is local time-out, then LTO is set and EXT and PRTY are cleared. If the source of the TEA* is due to an error in going to the VMEbus, then EXT is set and the other two status bits are cleared. If the source of the error is DRAM parity check error, then PRTY is set and the other two status bits are cleared. If the source of the error is none of the above conditions, then all three bits are cleared. Writing a 1 to bit 24 (SCLR) also clears all three bits.

82596CA LANC Interrupt Control Register

ADR/SIZ	$FFF4202A (8 bits)							
BIT	15	14	13	12	11	10	9	8
NAME	PLTY	E/L*	INT	IEN	ICLR	IL2	IL1	IL0
OPER	R/W	R/W	R	R/W	C	R/W	R/W	R/W
RESET	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL

IL2-IL0 Interrupt Request Level. These three bits select the interrupt level for the 82596CA LANC. Level 0 does not generate an interrupt.

ICLR In edge-sensitive mode, writing a logic 1 to this bit clears the INT status bit. This bit has no function in level-sensitive mode. This bit is always read as zero.

IEN Interrupt Enable. When this bit is high, the interrupt is enabled. The interrupt is disabled when this bit is low.

INT This status bit reflects the state of the INT pin from the LANC (qualified by the IEN bit). When this bit is high, a LANC INT interrupt is being generated at the level programmed in IL2-IL0 (if nonzero).

E/L* Edge or Level. When this bit is high, the interrupt is edge-sensitive. The interrupt is level-sensitive when this bit is low.

PLTY Polarity. When this bit is low, interrupt is activated by a rising edge/high level of the LANC INT pin. When this bit is high, interrupt is activated by a falling edge/low level of the LANC INT pin. Note that if this bit is changed while the E/L* bit is set (or is being set), a LANC interrupt may be generated. This can be avoided by setting the ICLR bit during write cycles that change the E/L* bit.

LANC Bus Error Interrupt Control Register

ADR/SIZ	$FFF4202B (8 bits)							
BIT	7	6	5	4	3	2	1	0
NAME	SC1	SC0	INT	IEN	ICLR	IL2	IL1	IL0
OPER	R/W	R/W	R	R/W	C	R/W	R/W	R/W
RESET	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL

IL2-IL0 Interrupt Request Level. These three bits select the interrupt level. Level 0 does not generate an interrupt.

ICLR Writing a logic 1 into this bit clears the INT status bit. This bit is always read as zero.

IEN Interrupt Enable. When this bit is high, the interrupt is enabled. The interrupt is disabled when this bit is low.

IRQ Interrupt Status. When this bit is high, a LANC Bus Error interrupt is being generated at the level programmed in IL2-IL0 (if nonzero).

SC1-SC0 Snoop Control. These control bits determine the value that the PCCchip2 drives onto the local MC68040 bus SC1 and SC0 pins, when the 82596CA (LANC) performs DMA accesses. During LANC DMA, if bit SC0 is 0 local bus pin SC0 is low, and when bit SC0 is 1, pin SC0 is high. The same relationship holds true for bit and pin SC1. See the M68040 user's manual for details on how it uses the Snoop Control signals. Note that these bits must be 0 on the MVME187.

Programming the SCSI Error Status and Interrupt Registers

This section provides address and bit level description of the SCSI interrupt control register and status register.

SCSI Error Status Register

ADR/SIZ	$FFF4202C (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME					PRTY	EXT	LTO	SCLR
OPER	R	R	R	R	R	R	R	W/R-0
RESET	0	0	0	0	0 PL	0 PL	0 PL	0

SCLR Writing a 1 to this bit clears bits 25 through 27 (LTO, EXT, and PRTY). Reading this bit always yields 0.

LTO,EXT,PRTY

These bits indicate the status of the last local bus error condition encountered by the SCSI processor while performing DMA accesses to the local bus. A local bus error condition is flagged by the assertion of TEA*. When the SCSI processor receives TEA* if the source of the error is local time-out, then LTO is set and EXT and PRTY are cleared. If the source of the TEA* is due to an error in going to the VMEbus, then EXT is set and the other two status bits are cleared. If the source of the error is DRAM parity check error, then PRTY is set and the other two status bits are cleared. If the source of the error is none of the above conditions, then all three bits are cleared. Writing a 1 to bit 24 (SCLR) also clears all three bits.

SCSI Interrupt Control Register

ADR/SIZ	$FFF4202F (8 bits)							
BIT	7	6	5	4	3	2	1	0
NAME			IRQ	IEN		IL2	IL1	IL0
OPER	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
RESET	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL

IL2-IL0 Interrupt Request Level. These three bits select the interrupt level for the SCSI Processor. Level 0 does not generate an interrupt.

IEN Interrupt Enable. When this bit is high, the interrupt is enabled. The interrupt is disabled when this bit is low.

IRQ Interrupt Status. This status bit reflects the state of the IRQ* pin of the SCSI Processor (qualified by the IEN bit). When this bit is high, a SCSI processor interrupt is being generated at the level programmed in IL2-IL0 (if nonzero). This status bit does not need to be cleared, because it is not edge-sensitive.

Programming the Printer Port

This section provides addresses and bit level descriptions of the printer port control, status, and data registers.

Printer ACK Interrupt Control Register

ADR/SIZ	$FFF42030 (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	PLTY	E/L*	INT	IEN	ICLR	IL2	IL1	IL0
OPER	R/W	R/W	R	R/W	C	R/W	R/W	R/W
RESET	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL

IL2-IL0 These three bits select the interrupt level for the printer ACK. Level 0 does not generate an interrupt.

ICLR In edge-sensitive mode, writing a logic 1 to this bit clears the INT status bit. This bit has no function in level-sensitive mode. This bit is always read as zero.

IEN When this bit is high, the interrupt is enabled. The interrupt is disabled when this bit is low.

INT When this bit is high, a printer ACK interrupt is being generated at the level programmed in IL2-IL0 (if nonzero).

E/L* When this bit is high, the interrupt is edge-sensitive. The interrupt is level-sensitive when this bit is low.

PLTY When this bit is low, interrupt is activated by a falling edge/low level on the PRACKI* pin. When this bit is high, interrupt is activated by a rising edge/high level on the PRACKI* pin. Note that if this bit is changed while the E/L* bit is set (or is being set), an ACK interrupt may be generated. This can be avoided by setting the ICLR bit during write cycles that change the E/L* bit.

Printer FAULT Interrupt Control Register

ADR/SIZ	$FFF42031 (8 bits)							
BIT	23	22	21	20	19	18	17	16
NAME	PLTY	E/L*	INT	IEN	ICLR	IL2	IL1	IL0
OPER	R/W	R/W	R	R/W	C	R/W	R/W	R/W
RESET	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL

IL2-IL0 These three bits select the interrupt level for the printer FAULT. Level 0 does not generate an interrupt.

ICLR In edge-sensitive mode, writing a logic 1 to this bit clears the INT status bit. This bit has no function in level-sensitive mode. This bit is always read as zero.

IEN When this bit is high, the interrupt is enabled. The interrupt is disabled when this bit is low.

INT When this bit is high, a printer FAULT interrupt is being generated at the level programmed in IL2-IL0 (if nonzero).

E/L* When this bit is high, the interrupt is edge-sensitive. The interrupt is level-sensitive when this bit is low.

PLTY When this bit is low, interrupt is activated by a falling edge/low level of the PRFAULTI* pin. When this bit is high, interrupt is activated by a rising edge /high level of the PRFAULTI* pin. Note that if this bit is changed while the E/L* bit is set (or is being set), a FAULT interrupt may be generated. This can be avoided by setting the ICLR bit during write cycles that change the E/L* bit.

Printer SEL Interrupt Control Register

ADR/SIZ	$FFF42032 (8 bits)							
BIT	15	14	13	12	11	10	9	8
NAME	PLTY	E/L*	INT	IEN	ICLR	IL2	IL1	IL0
OPER	R/W	R/W	R	R/W	C	R/W	R/W	R/W
RESET	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL

IL2-IL0 These three bits select the interrupt level for the printer SEL. Level 0 does not generate an interrupt.

ICLR In edge-sensitive mode, writing a logic 1 to this bit clears the INT status bit. This bit has no function in level-sensitive mode. This bit is always read as zero.

IEN When this bit is high, the interrupt is enabled. The interrupt is disabled when this bit is low.

INT When this bit is high, a printer SEL interrupt is being generated at the level programmed in IL2-IL0 (if nonzero).

E/L* When this bit is high, the interrupt is edge-sensitive. The interrupt is level-sensitive when this bit is low.

PLTY When this bit is low, interrupt is activated by a rising edge/high level of the SEL pin. When this bit is high, interrupt is activated by a falling edge/low level of the SEL pin. Note that if this bit is changed while the E/L* bit is set (or is being set), a SEL interrupt may be generated. This can be avoided by setting the ICLR bit during write cycles that change the E/L* bit.

Printer PE Interrupt Control Register

ADR/SIZ	$FFF42033 (8 bits)							
BIT	7	6	5	4	3	2	1	0
NAME	PLTY	E/L*	INT	IEN	ICLR	IL2	IL1	IL0
OPER	R/W	R/W	R	R/W	C	R/W	R/W	R/W
RESET	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL

IL2-IL0 These three bits select the interrupt level for the printer PE. Level 0 does not generate an interrupt.

ICLR In edge-sensitive mode, writing a logic 1 to this bit clears the INT status bit. This bit has no function in level-sensitive mode. This bit is always read as zero.

IEN When this bit is high, the interrupt is enabled. The interrupt is disabled when this bit is low.

INT When this bit is high, a printer PE interrupt is being generated at the level programmed in IL2-IL0 (if nonzero).

E/L* When this bit is high, the interrupt is edge-sensitive. The interrupt is level-sensitive when this bit is low.

PLTY When this bit is low, interrupt is activated by a rising edge/high level of the PE pin. When this bit is high, interrupt is activated by a falling edge/low level of the PE pin. Note that if this bit is changed while the E/L* bit is set (or is being set), a PE interrupt may be generated. This can be avoided by setting the ICLR bit during write cycles that change the E/L* bit.

Printer BUSY Interrupt Control Register

ADR/SIZ	$FFF42034 (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	PLTY	E/L*	INT	IEN	ICLR	IL2	IL1	IL0
OPER	R/W	R/W	R	R/W	C	R/W	R/W	R/W
RESET	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL

IL2-IL0 These three bits select the interrupt level for the printer BUSY. Level 0 does not generate an interrupt.

ICLR In edge-sensitive mode, writing a logic 1 to this bit clears the INT status bit. This bit has no function in level-sensitive mode. This bit is always read as zero.

IEN When this bit is high, the interrupt is enabled. The interrupt is disabled when this bit is low.

INT When this bit is high, a printer BUSY interrupt is being generated at the level programmed in IL2-IL0 (if nonzero).

E/L* When this bit is high, the interrupt is edge-sensitive. The interrupt is level-sensitive when this bit is low.

PLTY When this bit is low, interrupt is activated by a rising edge/high level of the BUSY pin. When this bit is high, interrupt is activated by a falling edge/low level of the BUSY pin. Note that if this bit is changed while the E/L* bit is set (or is being set), a BUSY interrupt may be generated. This can be avoided by setting the ICLR bit during write cycles that change the E/L* bit.

Printer Input Status Register

ADR/SIZ	$FFF42036 (8 bits)							
BIT	15	14	13	12	11	10	9	8
NAME	PINT			ACK	FLT	SEL	PE	BSY
OPER	R	R	R	R	R	R	R	R
RESET	X	0	0	X	X	X	X	X

BSY This bit reflects the state of the Printer Busy input pin. It is 1 when BSY is high and 0 when BSY is low.

PE This bit reflects the state of the Printer Paper Error input pin. It is 1 when PE is high and 0 when PE is low.

SEL This bit reflects the state of the Printer Select input pin. It is 1 when SELECT is high and 0 when SELECT is low.

FLT This bit reflects the state of the Printer Fault input pin. It is 1 when FAULT* is low and 0 when FAULT* is high.

ACK This bit reflects the state of the Printer Acknowledge input pin. It is 1 when ACK* is low and 0 when ACK* is high.

PINT Printer Interrupt Status - When this bit is high, an interrupt is being generated at the level programmed in one or more of the Printer Interrupt Control Registers. The interrupt may come from one or more printer status pins.

Printer Port Control Register

ADR/SIZ	$FFF42037 (8 bits)							
BIT	7	6	5	4	3	2	1	0
NAME				DOEN	INP	STB	FAST	MAN
OPER	R	R	R	R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	0 PL	0 PL	0 PL	0 PL	0 PL

MAN Manual Strobe Control - This bit selects the auto or manual mode for the printer strobe. When this bit is low, the printer strobe is generated automatically by a write to the Printer Data Register (auto mode). When this bit is high, the strobe pin is directly controlled by the STB control bit (manual mode).

FAST Strobe Timing - In auto mode, this bit controls the printer strobe timing. When this bit is low, the strobe time is 212 BCLK periods (10.6 µs at 20MHz, 8.5 µs at 25MHz and 6.4 µs at 33MHz). When this bit is high, the strobe time is 50 BCLK periods (2.5 µs at 20MHz, 2 µs at 25MHz and 1.5 µs at 33MHz). Note that the strobe time is the width of the low-going pulse generated on the STB* pin. Also note that after a write to the Printer Data Register, the PCCchip2 delays about one strobe time before issuing the STB* pulse. This bit is not used in manual mode.

STB Manual Strobe Control - In the manual mode, the software controls the strobe timing. When this bit is high, the printer strobe is activated. When this bit is low, the printer strobe is not activated. This bit has no function in auto mode.

INP Printer Input Prime - This bit controls the input prime signal. When this bit is high, the input prime signal is activated. When this bit is low, the input prime signal is not activated. Software must control the timing of the printer input prime signal.

DOEN Printer Data Output Enable - This bit controls the external data buffer for the printer port. When this bit is high, the external printer data buffer is enabled. When this bit is low, the external printer data buffer is disabled. For normal connection to a printer, DOEN should be set to 1.

Chip Speed Register

ADR/SIZ	$FFF42038 (16-bits)
BIT	31-16
NAME	CS31 - CS16
OPER	R
RESET	X

CS31-CS16 This read-only register is for factory test purposes only.

Printer Data Register

ADR/SIZ	$FFF4203A (16-bits)
BIT	15-0
NAME	PD15 - PD0
OPER	R/W
RESET	X

PD15-PD0 Writing to these bits causes the PCCchip2 to latch data into the external printer data buffer. Generally the printer data buffer only connects to PD7-PD0, because most printer data paths are 8 bits wide. PD7-PD0 can be accessed as an 8-bit register at location $FFF4203B, or PD15-PD0 can be accessed as a 16-bit register at location $FFF4203A. In auto mode, writing these bits also generates the strobe for the printer. Reading these bits causes the PCCchip2 to read the data from the printer data signal lines (no strobe is generated). When the DOEN bit is set, the printer data signal lines are driven by the external printer data buffer. When the DOEN bit is cleared, they must be terminated to high or to low and/or an external device must drive them.

Interrupt Priority Level Register

ADR/SIZ	$FFF4203E (8 bits)							
BIT	15	14	13	12	11	10	9	8
NAME						IPL2	IPL1	IPL0
OPER	R	R	R	R	R	R	R	R
RESET	0	0	0	0	0	X	X	X

IPL2-IPL0 Interrupt Priority Level - These bits reflect the priority-encoded interrupt request level. This level is a combination of the PCCchip2 interrupt requests and the interrupt requests driven onto the EIPL2-EIPL0 pins. Note that when the C040 bit is cleared, external devices can drive EIPL2-EIPL0 with their interrupt requests. When C040 is set, the PCCchip2 drives EIPL2-EIPL0 with its interrupt requests. In this case (C040 set), IPL2-IPL0 only reflect PCCchip2 interrupt requests. The IPL bits are encoded as shown below:

IPL2	IPL1	IPL0	Priority Level	Comments
0	0	0	0	No Interrupt
0	0	1	1	Lowest Level
0	1	0	2	
0	1	1	3	↑
1	0	0	4	
1	0	1	5	
1	1	0	6	↓
1	1	1	7	Highest Level

Interrupt Mask Level Register

ADR/SIZ	$FFF4203F (8 bits)							
BIT	7	6	5	4	3	2	1	0
NAME						MSK2	MSK1	MSK0
OPER	R	R	R	R	R	R/W	R/W	R/W
RESET	0	0	0	0	0	1 PL	1 PL	1 PL

MSK2-MSK0 Interrupt Mask Level - The interrupt mask level bits determine the level which must be exceeded by IPL2-IPL0 in order for the PCCchip2 to assert its INT pin. The MSK bits are encoded as follows:

MSK2	MSK1	MSK0	Priority Level	Comments
0	0	0	0	Lowest Level
0	0	1	1	
0	1	0	2	↕
0	1	1	3	
1	0	0	4	
1	0	1	5	
1	1	0	6	
1	1	1	7	Highest Level

MEMC040 4

Introduction

This chapter defines the four-way interleaving memory controller ASIC for the MC68040-type bus. This memory controller ASIC is referred to as the MEMC040 hereafter. The MEMC040 is designed for the MVME166/ MVME167/MVME187 Single Board Computers and is to be used in conjunction with the data multiplexer ASIC (MEMMUX) and the address latch/multiplexer ASIC (AMUX) to provide the interface to a 144-bit wide DRAM memory system.

Features

- ❑ Allows 2-1-1-1 memory accesses (sustained) for burst writes.
- ❑ Allows 4-1-1-1 memory accesses (sustained) for burst reads (5-1-1-1 with parity ON).
- ❑ One MEMC040 controls up to two contiguous blocks of 144-bit wide memory array.
- ❑ One MEMC040 controls up to 128MB of memory.
- ❑ Supports 1M, 4M, and 16M DRAM in either x1 or x4 configurations.
- ❑ Supports byte, two-byte, four-byte, and cache line read or write transfers.
- ❑ Supports write-per-bit x4 DRAM for parity memory.
- ❑ Provides an 8-bit status register and an 8-bit control register when write-per-bit DRAMs are not used.
- ❑ Programmable base address for the memory blocks.
- ❑ Programmable parity modes: ON, OFF, interrupt.
- ❑ Built-in refresh timer and refresh controller.
- ❑ Write-wrong parity control bit for test purposes.

Functional Description

This section describes the MEMC040 in general and then in detail.

General Description

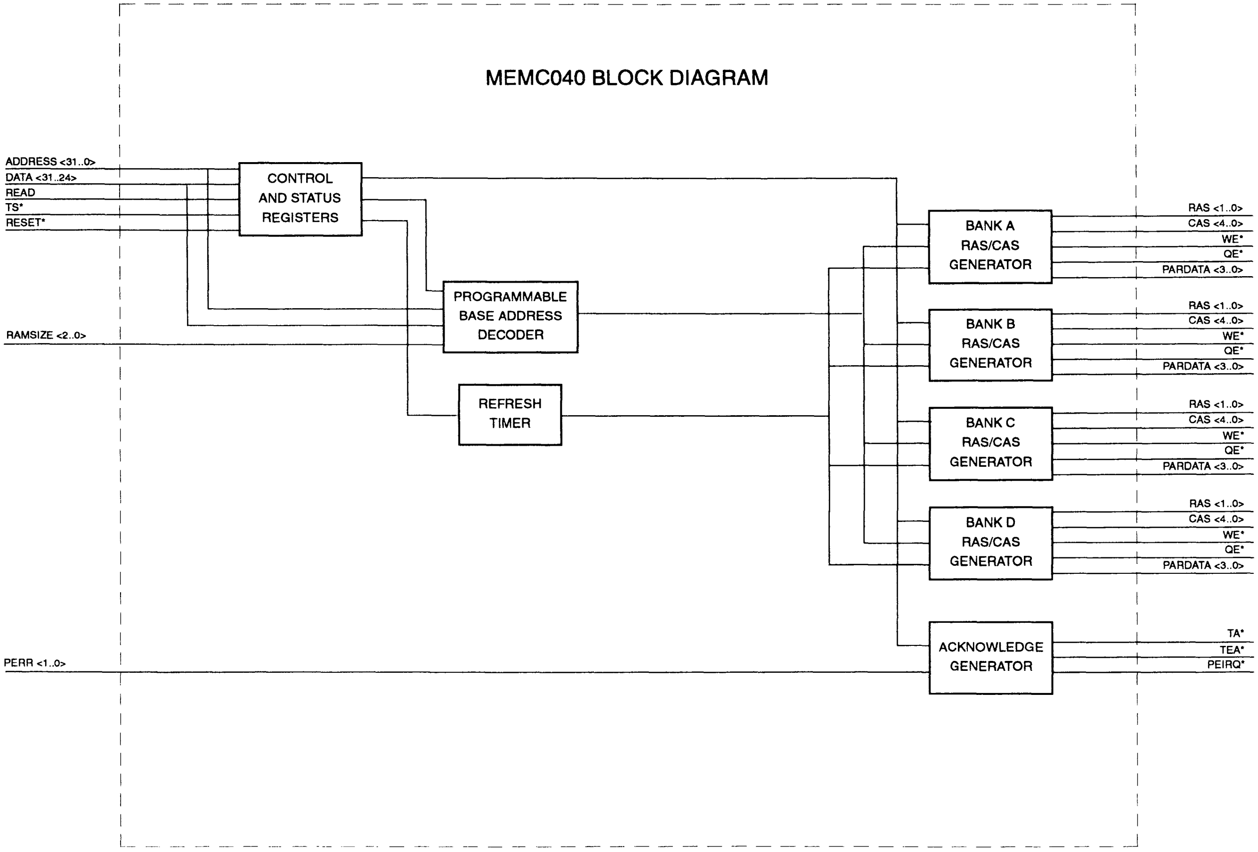
The MEMC040 is designed to be used with one AMUX address multiplexer ASIC, two MEMMUX data multiplexer ASICs, and x1 or x4 DRAM memory chips to form a memory system for an MC68040-type bus. This ASIC is used by the MVME166/167/187 Single Board Computers. The typical block diagram for such a memory scheme is shown in Figure 4-1.

The MEMC040 is specifically designed to provide maximum performance for cache line (burst) cycles to and from the MC68040 bus. This is done by providing a four-way interleave between the 32-bit MC68040 data bus and four separate banks of 32-bit DRAM. This permits burst accesses to be pipelined, giving high performance from standard speed DRAMs. For example, burst reads can be sustained at speeds of 7 clocks per line of four four-bytes (8 clocks per line with parity enabled). This gives an average access time of 1.75 clocks (2.0 clocks) per four-byte, or 70 nsec (80 nsec) at 25 MHz, while using 80 nsec DRAM. Burst writes can be sustained at 5 clocks per line, for an average of 1.25 clocks per four-byte, or 50 nsec at 25 MHz.

Random reads and writes are pipelined to the extent possible. Random reads take four clocks (five clocks with parity ON), while random writes take two to four clocks, based on the amount of pipelining possible. When write-per-bit DRAMs are used for the parity memory, a byte or two-byte write takes one clock longer than a four-byte write, to setup mask data before RAS. For four-byte and burst writes, since all four parity bits are written, the non-write-per-bit memory cycle is used.

The clock cycle counts and timing given above assume that the FASTREAD input pin is at a logic 1. If this pin is at a logic 0, add one clock to all read cycles, making random reads five clocks and line reads eight clocks (5-1-1-1). If parity checking is ON, random reads become six clocks and line reads become nine clocks (6-1-1-1). This permits operation at higher clock frequencies while relaxing memory speed requirements. Write timing is unaffected by the FASTREAD pin.

In addition, the MEMC040 also contains refresh timers and refresh arbitration logic. One CAS-before-RAS refresh cycle is performed nominally every 16 μsec.



1345 9403

Figure 4-1. Block Diagram for Memory Using MEMC040

Status and Control Registers

The MEMC040 contains eight 8-bit registers that appear only at D31-D24. Burst reads and writes are not allowed to be executed to these registers, but byte, two-byte, or four-byte accesses may be used interchangeably. The base address of the first MEMC040 in a system is $FFF43000, and the base address of the second MEMC040 (if used) is $FFF43100.

Each register definition includes a table with 5 lines:

- Line 1 is the two base addresses 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.

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

P	The bit is affected by power-up reset.
S	The bit is affected by SYSRESET.
L	The bit is affected by local reset.
X	The bit is not affected by reset.
0	This bit is always 0.
1	This bit is always 1.

Table 4-1 shows all MEMC040 internal registers.

Table 4-1. MEMC040 Internal Register Memory Map

2nd	1st				Data Bits				
MEMC040	MEMC040	D31	D30	D29	D28	D27	D26	D25	D24
$FFF43100	$FFF43000	CID7	CID6	CID5	CID4	CID3	CID2	CID1	CID0
$FFF43104	$FFF43004	REV7	REV6	REV5	REV4	REV3	REV2	REV1	REV0
$FFF43108	$FFF43008			FSTRD	EXTPEN	WPB*	MSIZ2	MSIZ1	MSIZ0
$FFF4310C	$FFF4300C	STS7	STS6	STS5	STS4	STS3	STS2	STS1	STS0
$FFF43110	$FFF43010	OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1	OUT0
$FFF43114	$FFF43014	BAD31	BAD30	BAD29	BAD28	BAD27	BAD26	BAD25	BAD24
$FFF43118	$FFF43018	BAD23	BAD22	DMCTL	SWAIT	WWP	PARINT	PAREN	RAMEN
$FFF4311C	$FFF4301C	BCK7	BCK6	BCK5	BCK4	BCK3	BCK2	BCK1	BCK0

Register 1 - Chip ID Register

The Chip ID Register is located at offset $00 in the register map of the MEMC040. It is an 8-bit register that is hard-wired to read a hexadecimal value of $80. The MEMC040 can be given a software reset by writing a value of $0F to this register. This write is terminated properly with TA*, and sets most internal registers to their default (power-up) state. Exceptions are noted in the register descriptions. Writes of any value other than $0F to this register are ignored; however, the MEMC040 always terminates the cycles properly with TA*.

ADR/SIZ	1st $FFF43000/2nd $FFF43100 (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	CID7	CID6	CID5	CID4	CID3	CID2	CID1	CID0
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	1	0	0	0	0	0	0	0

Register 2 - Chip Revision Register

The Chip Revision Register is located at offset $04 in the register map of the MEMC040. It is an 8-bit read-only register that is hard-wired to reflect the revision level of the MEMC040 ASIC. Writes to this register are ignored; however, the MEMC040 always terminates the cycles properly with TA*.

ADR/SIZ	1st $FFF43004/2nd $FFF43104 (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	REV7	REV6	REV5	REV4	REV3	REV2	REV1	REV0
OPER	R	R	R	R	R	R	R	R
RESET	0	0	0	0	0	0	0	0

Register 3 - Memory Configuration Register

The Memory Configuration Register is located at offset $08 in the register map of the MEMC040. It is an 8-bit read-only register that reflects the states of the external status pins to specify the memory configuration to the MEMC040. Writes to this register are ignored; however, the MEMC040 always terminates the cycles properly with TA*.

MSIZ2-MSIZ0 Memory Size <2..0>. MSIZ2-MSIZ0 together define the size of the total memory to be controlled by the MEMC040. These bits reflect the actual states of the Memory Size input strap pins and are assigned as follows:

MSIZ2	MSIZ1	MSIZ0	Memory Size
0	0	0	4 MB
0	0	1	8 MB
0	1	0	16 MB
0	1	1	32 MB
1	0	0	64 MB
1	0	1	128 MB
1	1	0	Reserved
1	1	1	Reserved

WPB* Write-Per-Bit mode. This status bit is controlled by the WPB input strap pin. When write-per-bit x4 DRAMs are used for the parity memory (as in the MVME166/167/187), the WPB pin should be pulled up or left unconnected. (An internal pullup is provided on this pin.) This bit is then read as a logic 0, and byte/two-byte writes utilize write-per-bit memory cycles.

If the MEMC040 is used in an application which does not use write-per-bit DRAMs for the parity memory, then the WPB input strap pin should be connected to a logic low or ground. This bit is then read as a logic one, and pins APD3-APD0 and BPD3-BPD0 become input pins and together form Status Register 4. Also, pins CPD3-CPD0 and DPD3-DPD0 are combined to form an 8-bit control register driven by Control Register 5.

EXTPEN External Parity Enable. This status bit reflects the state of the EXTPEN input pin. When EXTPEN is a logic 1, it enables the PAREN and PARINT control bits to determine the parity checking mode for the memory. If EXTPEN is at a logic 0, all parity checking and parity interrupts are disabled. An internal pullup is provided on this input pin, to assure a logic 1 if unconnected.

FSTRD Fast Read. This status bit reflects the state of the FASTREAD input pin. When this input is at a logic 1, the MEMC040 operates in the fast mode. When this input is at a logic 0, the MEMC040 operates in the slow mode. Timing for the fast and slow modes is given in the *General Description* section. Write timing is unaffected by this pin. An internal pullup is provided on this input pin, to assure a logic 1 if unconnected.

Register 4 - Alternate Status Register

Register 4 is a read-only 8-bit status register located at offset $0C in the MEMC040 register map. Writes to this register are ignored; however, the MEMC040 always terminates the cycles properly with TA*. Register 4 is only defined when WPB* status bit is at logic 1. When it is defined, the Alternate Status Register reflects the states of the APD3-APD0 and BPD3-BPD0 pins as follow:

ADR/SIZ	1st $FFF4300C/2nd $FFF4310C (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	STS7 APD3	STS6 APD2	STS5 APD1	STS4 APD0	STS3 BPD3	STS2 BPD2	STS1 BPD1	STS0 BPD0
OPER	R	R	R	R	R	R	R	R
RESET	X	X	X	X	X	X	X	X

Register 5 - Alternate Control Register

Register 5 is an 8-bit read/write register located at offset $10 in the MEMC040 register map. This register is cleared to zero by a reset. When the WPB* status bit in Register 3 is false (logic 1), the contents of the Alternate Control Register are driven to the CPD3-CPD0 and DPD3-DPD0 pins as follows:

ADR/SIZ	1st $FFF43010/2nd $FFF43110 (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	OUT7 CPD3	OUT6 CPD2	OUT5 CPD1	OUT4 CPD0	OUT3 DPD3	OUT2 DPD2	OUT1 DPD1	OUT0 DPD0
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

Register 6 - Base Address Register

Register 6 is an 8-bit read/write register located at offset $14 in the MEMC040 register map. These 8 bits are combined with two most significant bits in Register 7 to form BAD31-BAD22, which defines the base address of the memory. For larger memory sizes, the lower significant bits are ignored. All 8 bits of this register are cleared to zero by a reset. The bit assignments for Base Address Register (BAD) are:

ADR/SIZ	1st $FFF43014/2nd $FFF43114 (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	BAD31	BAD30	BAD29	BAD28	BAD27	BAD26	BAD25	BAD24
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

Register 7 - RAM Control Register

Register 7 is an 8-bit read/write register located at offset $18 in the MEMC040 register map. All 8 bits of this register are cleared to zero by a reset. The bit assignments for the RAM Control Register are:

ADR/SIZ	1st $FFF43018/2nd $FFF43118 (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	BAD23	BAD22	DMCTL	SWAIT	WWP	PARINT	PAREN	RAMEN
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

RAMEN RAM Enable. This control bit is used to enable the MEMC040 to perform read/write accesses to the memory. The memory is enabled when this bit is set and is disabled when this bit is cleared. *This bit should only be set after BAD31-BAD22 have been initialized.*

PAREN Parity Enable. When EXTPEN is at a logic 1 to enable parity checking, this bit and the PARINT bit (below) control the type of parity checking performed for the MPU and alternate bus masters.

PARINT Parity Interrupt. When EXTPEN is at a logic 1 to enable parity checking, this bit and the PAREN bit (above) control the type of parity checking performed for the MPU and alternate bus masters, according to the table below:

PAREN	PARINT	MPU	Alternate
0	0	None	None
0	1	Interrupt	None
1	0	Checked	Checked
1	1	Interrupt	Checked

"None" means no parity checking. Parity errors are not detected or reported. "Interrupt" means that the MPU receives a parity interrupt if a parity error occurs. The bus cycle is terminated with TA, and runs at the same speed as unchecked cycles. "Checked" means that the cycle is terminated by TEA if a parity error occurs, and requires one more clock than unchecked cycles. When EXTPEN is low, parity checking is disabled for all bus masters, regardless of the state of PAREN or PARINT. When the interrupt mode is selected, the Parity Error Interrupt in the VMEchip2 must be enabled.

WWP Write Wrong Parity. The state of this control bit is driven to the WWP pin. A logic 1 means that external logic (i.e., the MEMMUX) should present wrong parity to the DRAM, to test the parity generation/checking circuits.

SWAIT Snoop Wait. When SWAIT is at logic 0, the MEMC040 does not wait for MI* (Memory Inhibit signal from MC68040) to be negated before starting a read access or a line push (burst write). When SWAIT is at logic 1, the MEMC040 waits for MI* (Memory Inhibit signal from MC68040) to be negated before starting any memory accesses.

DMCTL Data Mux Control. This bit is cleared to logic 0 at reset and must remain clear for all memory cycles in the MVME166/167/187 application. If this bit is set, the MEMMUX performs read-modify-write operations to the parity DRAMs for byte or two-byte writes. The MVME166/167/187 uses write-per-bit DRAMs, with the MEMC040 providing the write-per-bit support. This bit may be toggled for testing purposes while the RAMEN bit is cleared.

BAD23-BAD22 These two bits are combined with all eight bits in Register 7 to form BAD31-BAD22 to define the base address of the memory. For larger memory sizes, the lower significant bits are ignored.

Register 8 - Bus Clock Register

Bus Clock Register is an 8-bit read/write register located at offset $1C in the MEMC040 register map. It should be programmed with the hexadecimal value of the operating clock frequency in MHz (i.e. $21 for 33 MHz). The MEMC040 uses the value programmed in this register to control the refresh timer so that the DRAMs are refreshed every 15.6 microseconds. After power-up, this register is initialized to $10 (for 16 MHz).

Note

This register is configured only at the rising edge of the POR* (Power-Up Reset) pin of the MEMC040, and is unchanged by the software reset from the Chip ID Register or the RESET* pin.

ADR/SIZ	1st $FFF4301C/2nd $FFF4311C (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	BCK7	BCK6	BCK5	BCK4	BCK3	BCK2	BCK1	BCK0
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 P	0 P	0 P	0 P	0 P	0 P	0 P	0 P

The refresh rate is defined by the following equation:

refresh rate = **BCK** / *bus clock* * 16

where **BCK** is the value programmed in the Bus Clock Register, and *bus clock* is the board bus clock frequency.

For example, on a 25 MHz board, the refresh rate is:

25 / 25M * 16 = 16 μs

MCECC 5

Introduction

This chapter describes the ECC DRAM Controller ASIC (MCECC) used on the memory mezzanine boards with ECC protection. The MCECC is designed for the MVME166/167/187 families of boards and is used in a set of two, to provide the interface to a 144-bit wide DRAM memory system.

Features

- ❑ Allows 2-1-1-1 memory accesses (sustained) for burst writes.
- ❑ Allows 4-1-1-1 memory accesses (sustained) for burst reads (5-1-1-1 with BERR on or when FSTRD is cleared).
- ❑ Supports byte, two-byte, four-byte, and cache line read or write transfers.
- ❑ Programmable base address for DRAM.
- ❑ Built-in refresh timer and refresh controller.
- ❑ ECC :
  - Single Bit Error Detect and Correct.
  - Software enabled interrupt on Single Bit Error.
  - Address and Syndrome Register for Single Bit Error logging support.
  - Double Bit Error detect.
  - Software programmable bus error and/or interrupt on Double Bit Error.
- ❑ Programmable period automatic scrub operation.

Functional Description

The following sections provide an overview of the functions provided by the MCECC. A detailed programming model for the MCECC control and status registers is provided in the section on *Programming Model.*

General Description

The MCECC is designed to be used as a set of two chips. A pair of MCECCs works with x4 DRAM memory chips to form a memory system for the MVME166/167/187 boards. A pair of MCECCs that is connected to implement a memory control function is referred to as an "MCECC pair". The MCECC pair provides all the functions required to implement a memory system. These include programmable map decoding, memory control, refresh, and a scrubber. The scrubber, when it is enabled, periodically scans memory looking for errors. If the scrubber finds a single bit error in the memory array, it corrects it. This prevents soft single bit errors from becoming double bit errors.

Performance

The MCECC pair is specifically designed to provide maximum performance for cache line (burst) cycles to and from the MC68040 bus. This is done by providing a four-way interleave between the 32-bit MC68040 data bus and the 128 bit (144 with check bits) DRAM. This permits burst accesses to be pipelined, giving high performance from standard speed, static column, DRAMs. For example, burst reads can be sustained at speeds of 7 clocks per line of four four-bytes (8 clocks per line with BERR enabled or FSTRD cleared). If the local MC68040 bus clock frequency is 25MHz, this gives an average access time of 70ns (80ns with BERR or no FSTRD) per four-byte. Burst writes can be sustained at 5 clocks per line, for an average of 50 ns at 33 MHz.

Random (non-burst) reads and writes are pipelined to the extent possible. Random reads take four clocks (five clocks with BERR on or FSTRD cleared).

Random, non-burst writes are the slowest kind of access because they require that the MCECC pair perform a read-modify-write cycle to the DRAM in order to complete. The MCECC pair responds to the local bus in two clocks during random writes, but then it takes another eight clocks for the DRAM read-modify-write cycle to complete, thereby making the effective cycle time 10 clocks if the following access by the local bus master is to DRAM. This boils down to two clocks for one random write, and 10 clocks for sustained random writes.

The performance specifications for the MCECC are shown in Table 5-1.

Table 5-1. MCECC Specifications

Descriptions	Specifications
Reads, BERR off, FSTRD = 1	4 clock cycles for random reads 4-1-1-1 clock cycles for burst reads (sustained)
Reads, FSTRD = 0	5 clock cycles for random reads 5-1-1-1 clock cycles for burst reads (sustained)
Reads, BERR on	5 clock cycles for random reads 5-1-1-1 clock cycles for burst reads (sustained)
Writes	2 to 10 clock cycles for random non-burst writes 2-1-1-1 clock cycles for burst writes (sustained)

Cache Coherency

The MCECC pair supports the MC68040 caching scheme on the local bus by always providing 32 bits of valid data during DRAM read cycles regardless of the number of bytes requested by the local bus master for the cycle. It also supports cache coherency by monitoring the snoop control signal lines on the local bus and behaving appropriately based on their value.

When the snoop control signal lines (SC1, SC0) indicate that snooping is inhibited, the MCECC pair ignores the memory inhibit (MI*) signal line.

When (SC1, SC0) do not indicate that snooping is inhibited, the MCECC pair responds differently to DRAM accesses, based on whether the cycle is a read or a write, and on the snoop wait (SWAIT) control bit.

For a read with SWAIT = 0, the MCECC pair immediately starts a read cycle to the DRAM and latches the data from the DRAMs. It waits, however, for MI* to be negated before it enables the data (that has been latched) onto the local bus and asserts TA* or TEA*. If TA* or TEA* is asserted by another local bus slave before MI* is negated, then the MCECC pair assumes that the cycle is over and that the DRAM is not to participate in that cycle.

For a read with SWAIT = 1, the MCECC pair behaves the same as with SWAIT = 0 except that it does not start the DRAM read cycle until it sees the MI* signal negated. Note that this means that if another local bus slave asserts TA* or TEA* before MI* is negated, then the MCECC pair never starts the DRAM read cycle.

For a write cycle, the MCECC pair always waits for MI* to be negated before it begins a write cycle to the DRAM. If another local bus slave asserts TA* or TEA* before MI* is negated, then the MCECC pair never starts the DRAM write cycle.

ECC

The MCECC pair performs single bit error correction and double bit error detection (SECDED). The 32 bit wide local data bus is divided into lower (D00-D15) and upper (D16-D31) halves. Each half is routed through an MCECC, which multiplexes it with half of the 128 bit wide DRAM. This allows each MCECC to connect to 64 bits of the DRAM. Each MCECC additionally connects to 8 bits of check bit DRAM. This actually makes the DRAM array 144 bits wide (128 bits of normal data and 16 bits of check data).

Cycle Types

To support ECC, the MCECC pair always deals with DRAM using full width (144 bits, 72 bits for each MCECC) accesses. When the local bus master requests any size read of DRAM, the MCECC pair reads 144 bits. When the local bus master requests a line write to DRAM, the MCECC pair writes all 144 bits. When the local bus master requests a byte, word (two-byte), or longword write to DRAM, the MCECC pair performs a 144-bit wide read cycle to DRAM, merges the appropriate local bus write data in, and writes 144 bits to DRAM.

Error Reporting

The MCECCs generate the ECC check bits for write cycles. They also check read data from the DRAM and correct it if it contains a single bit error. If a non-correctable error occurs within either of the MCECC 72 bits of read data, the affected MCECC indicates it by asserting its non-correctable error (NCE*) pin.

The following paragraphs indicate the actions taken by the MCECC pair for different error situations.

Single Bit Error (Cycle Type = Burst Read or Non-Burst Read)

Correct the Data that is driven to the local MC68040 bus.

Do not correct the Data in DRAM. Note that the DRAM is not corrected until the next scrub of that address, which happens only if scrubbing is enabled.

Terminate the cycle normally. (Assert TA to the local bus.)

Log the error if one has not already been logged.

Notify the local MPU via interrupt if so enabled.

Double Bit Error (Cycle Type = Burst Read or Non-Burst Read)

Cannot correct the data that is driven to the local MC68040 bus.

Leave the error in DRAM. (Note that it is not corrected in DRAM during the next scrub of that address.)

Terminate the cycle with Bus Error (assert TEA to the local bus) if so enabled.

Log the error if one has not already been logged.

Notify the local MPU via interrupt if so enabled.

Triple (or Greater) Bit Error (Cycle Type = Burst Read or Non-Burst Read)

Some of these errors are detected correctly and are treated the same as a double bit error. The rest could show up as "no error" or "single bit error", both of which are incorrect.

Cycle Type = Burst Write

Because all of the bits are written during a burst write, no checking is done.

Single Bit Error (Cycle Type = Non-Burst Write)

Correct the data read from the DRAM, merge with the write data, and write the correct, merged data to the DRAM.

Terminate the cycle normally. (Assert TA to the local bus.)

Log the error if one has not already been logged.

Notify the local MPU via interrupt if so enabled.

Double Bit Error (Cycle Type = Non-Burst Write)

Do not perform the write portion of the cycle. This causes the location to continue to indicate non-correctable error when accessed.

Terminate the cycle normally. (Assert TA to the local bus.)

Log the error if one has not already been logged.

Notify the local MPU via interrupt if so enabled.

Triple (or Greater) Bit Error (Cycle Type = Non-Burst Write)

Some of these errors are detected correctly and are treated the same as a double bit error. The rest could show up as "no error" or "single bit error", both of which are incorrect.

Single Bit Error (Cycle Type = Scrub)

Write corrected data to the DRAM.

Log the error if one has not already been logged.

Notify the local MPU via interrupt if so enabled.

Double Bit Error (Cycle Type = Scrub)

Do not perform the write portion of the cycle. This causes the location to continue to indicate non-correctable error when accessed.

Log the error if one has not already been logged.

Notify the local MPU via interrupt if so enabled.

Triple (or Greater) Bit Error (Cycle Type = Scrub)

Some of these errors are detected correctly and are treated the same as a double bit error. The rest could show up as "no error" or "single bit error", both of which are incorrect.

Error Logging

ECC error logging is facilitated by the MCECC because of its internal latches. When an error (single or double bit) occurs in the DRAMs to which an MCECC is connected, it freezes the address of the error and the syndrome bits associated with the data that is in error. Each MCECC performs this logging function independently of the other. Once an MCECC has logged an error, it does not log any new errors that occur until the ERRLOG control/status bit has been cleared by software.

Scrub

The MCECC pair contains programmable registers and circuitry that provide the scrubbing function. Programmable registers determine how often the entire DRAM is scrubbed. During a scrub, the scrubber holds the memory for a programmable amount of time, then releases it for the local bus, or refresher if one of them is requesting local bus mastership. The scrubber then refrains from using the DRAM again for a programmable amount of time. Each scrub cycle is made up of a full 144-bit read of DRAM, a correction of any single bit errors, and a write of the full 144 corrected bits back to the same location. If a single or double bit error occurs, the local bus master is notified if such interrupts are enabled in the control register. A software bit is available to disable the read portion of the scrub cycle.

Refresh

The MCECC pair provides refresh control for the DRAM. It performs a single CAS-before-RAS refresh cycle to the two DRAM blocks approximately once every 15.6 µs. To prevent undue noise generation, the MCECC pair does not refresh both blocks at once, but staggers the refreshes by one clock cycle.

Arbitration

The MCECC pair has 3 different entities that can request use of the DRAM cycle controller: (1) the local bus master, (2) the refresher, and (3) the scrubber.

The MCECC pair arbiter accepts requests and provides grants to the requesting entities as follows:

> Priority is (highest to lowest) refresher, local bus, and scrubber.
>
> When no requests are pending, the arbiter defaults to providing a local bus grant for fast response to local bus cycles.
>
> Although the arbiter operates on a priority basis, it also performs a pseudo round robin algorithm in order to prevent starving any of the requesting entities.

Chip Defaults

Some jumper option kinds of parameters need to be configured in the MCECC pair. These options include DRAM size, DRAM speed, Control and Status Register Selection, etc. Rather than use pins (which are extremely scarce) for each of the options, the MCECC pair is designed to have an external PAL or other equivalent logic provide this information at reset time, using one pin as a serial input. The information provided to this input pin at power-up-reset or local bus reset, is called the "reset serial bit stream". The reset serial bit stream initializes the MCECC pair by setting or resetting the bits that appear in the Defaults 1 and Defaults 2 Registers. Software can override this initial setting by writing to the Defaults Registers. It is not recommended that non-test software alter the bits in the Defaults Registers.

Programming Model

This section defines the programming model for the control and status registers (CSRs) in the MCECC pair. The base address of the CSRs is hard coded to the address $FFF43000 for the MCECC pair on the first mezzanine board and $FFF43100 for the MCECC pair on the second mezzanine board. The CSRs for the two MCECCs appear at the same address, (one on D16-D31, the other on D00-D15). Hardware automatically duplicates the values that are written to the CSRs in the upper MCECC (the one that connects to D16-D31) to the lower MCECC (the one that connects to D0-D15). Hence Software only needs to write to the control registers in the upper MCECC. This duplicating function can be disabled by software for test purposes.

Some effort has gone into making the register map for the first eight registers, of the MCECC pair, look as close as possible to that for the eight registers contained in the MEMC040. Where there are differences, they are noted. The remaining 18 registers contain functions unique to the MCECC pair.

The possible operations for each bit in the CSR are as follows:

R This bit is a read only status bit.

R/W This bit is readable and writable.

R/C This status bit is cleared by writing a one to it.

C Writing a zero 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.

The possible states of the bits after local, software, and power-up reset are as defined below.

P The bit is affected by power-up reset.

L The bit is affected by local reset.

S The bit is affected by software reset. (Writing $0F to the Chip ID Register)

X The bit is not affected by reset.

V The effect of reset on this bit is variable.

A summary of the first eight CSR registers (the ones that correspond to those found in the MEMC040) is shown in Table 5-2, following. Note that even though there are two sets of these registers, one for the lower MCECC and one for the upper MCECC, software should only perform read and write cycles to the control and status registers in the upper MCECC. Hardware takes care of duplicating the information to the lower MCECC. The following descriptions show the upper MCECC bit positions. Upper MCECC bit positions 31-24 correspond to lower MCECC bit positions 15-8. The base address of the CSRs is hard coded to the address $FFF43000 for the MCECC pair on the first mezzanine board and $FFF43100 for the MCECC pair on the second mezzanine board.

Table 5-2. MCECC Internal Register Memory Map, Part 1

MCECC Base Address = $FFF43000 (1st); $FFF43100 (2nd)

Register Offset	Register Name	Register Bit Names							
		D31	D30	D29	D28	D27	D26	D25	D24
$00	CHIP ID	CID7	CID6	CID5	CID4	CID3	CID2	CID1	CID0
$04	CHIP REVISION	REV7	REV6	REV5	REV4	REV3	REV2	REV1	REV0
$08	MEM CONFIG	0	0	FSTRD	1	0	MSIZ2	MSIZ1	MSIZ0
$0C	DUMMY 0	0	0	0	0	0	0	0	0
$10	DUMMY 1	0	0	0	0	0	0	0	0
$14	BASE ADDRESS	BAD31	BAD30	BAD29	BAD28	BAD27	BAD26	BAD25	BAD24
$18	DRAM CONTRL	BAD23	BAD22	RWB5	SWAIT	RWB3	NCEIEN	NCEBEN	RAMEN
$1C	BCLK FREQ	BCK7	BCK6	BCK5	BCK4	BCK3	BCK2	BCK1	BCK0

A summary of the remaining CSR registers is shown in Table 5-3, following. As with the first eight CSR registers, the summary shows the registers for the upper MCECC. The registers for the lower MCECC appear on D8-D15. As with the first eight CSR registers, software should read and write to only the upper MCECC CSRs. The exception to this is the error logger, error address, and error syndrome registers. These registers contain information specific to each MCECC and the DRAMs which it controls, and as such should be treated separately. The base address of the CSRs is hard coded to the address $FFF43000 for the MCECC pair on the first mezzanine board and $FFF43100 for the MCECC pair on the second mezzanine board.

Table 5-3. MCECC Internal Register Memory Map, Part 2

MCECC Base Address = $FFF43000 (1st); $FFF43100 (2nd)

Register Offset	Register Name	Register Bit Names							
		D31	D30	D29	D28	D27	D26	D25	D24
$20	DATA CONTRL	0	0	DERC	ZFILL	RWCKB	0	0	0
$24	SCRUB CNTRL	RACODE	RADATA	HITDIS	SCRB	SCRBEN	0	SBEIEN	IDIS
$28	SCRUB PERIOD	SBPD15	SBPD14	SBPD13	SBPD12	SBPD11	SBPD10	SBPD9	SBPD8
$2C	SCRUB PERIOD	SBPD7	SBPD6	SBPD5	SBPD4	SBPD3	SBPD2	SBPD1	SBPD0
$30	CHIP PRESCALE	CPS7	CPS6	CPS5	CPS4	CPS3	CPS2	CPS1	CPS0
$34	SCRUB TIME ON/OFF	SRDIS	0	STON2	STON1	STON0	STOFF2	STOFF1	STOFF0
$38	SCRUB PRESCALE	0	0	SPS21	SPS20	SPS19	SPS18	SPS17	SPS16
$3C	SCRUB PRESCALE	SPS15	SPS14	SPS13	SPS12	SPS11	SPS10	SPS9	SPS8
$40	SCRUB PRESCALE	SPS7	SPS6	SPS5	SPS4	SPS3	SPS2	SPS1	SPS0
$44	SCRUB TIMER	ST15	ST14	ST13	ST12	ST11	ST10	ST9	ST8
$48	SCRUB TIMER	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
$4C	SCRUB ADDR CNTRL	0	0	0	0	0	SAC26	SAC25	SAC24
$50	SCRUB ADDR CNTRL	SAC23	SAC22	SAC21	SAC20	SAC19	SAC18	SAC17	SAC16
$54	SCRUB ADDR CNTRL	SAC15	SAC14	SAC13	SAC12	SAC11	SAC10	SAC9	SAC8
$58	SCRUB ADDR CNTRL	SAC7	SAC6	SAC5	SAC4	0	0	0	0
$5C	ERROR LOGGER	ERRLOG	ERD	ESCRB	ERA	EALT	0	MBE	SBE
$60	ERROR ADDRESS	EA31	EA30	EA29	EA28	EA27	EA26	EA25	EA24
$64	ERROR ADDRESS	EA23	EA22	EA21	EA20	EA19	EA18	EA17	EA16
$68	ERROR ADDRESS	EA15	EA14	EA13	EA12	EA11	EA10	EA9	EA8
$6C	ERROR ADDRESS	EA7	EA6	EA5	EA4	0	0	0	0
$70	ERROR SYNDROME	S7	S6	S5	S4	S3	S2	S1	S0
$74	DEFAULTS1	WRHDIS	STATCOL	FSTRD	SELI1	SELI0	RSIZ2	RSIZ1	RSIZ0
$78	DEFAULTS2	FRC_OPN	XY_FLIP	REFDIS	TVECT	NOCACHE	RESST2	RESST1	RESST0

Chip ID Register

The Chip ID Register is hard-wired to a hexadecimal value of $81. The MCECC can be given a software reset by writing a value of $0F to this register. This write is terminated properly with TA*, and sets most internal registers to their default (power-up) state. Writes of any value other than $0F to this register are ignored; however, the MCECC always terminates the cycles properly with TA*.

Difference from MEMC040: value = $80 for MEMC040; value = $81 for MCECC.

ADR/SIZ	1st $FFF43000/2nd $FFF43100 (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	CID7	CID6	CID5	CID4	CID3	CID2	CID1	CID0
OPER	R	R	R	R	R	R	R	R
RESET	X	X	X	X	X	X	X	X

Chip Revision Register

The Chip Revision Register is hard-wired to reflect the revision level of the MCECC ASIC. The current value of this register is $00. Writes to this register are ignored; however, the MCECC pair always terminates the cycles properly with TA*.

Difference from MEMC040: none between corresponding revisions of the two parts.

ADR/SIZ	1st $FFF43004/2nd $FFF43104 (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	REV7	REV6	REV5	REV4	REV3	REV2	REV1	REV0
OPER	R	R	R	R	R	R	R	R
RESET	X	X	X	X	X	X	X	X

Memory Configuration Register

ADR/SIZ	1st $FFF43008/2nd $FFF43108 (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	0	0	FSTRD	RB4	RB3	MSIZ2	MSIZ1	MSIZ0
OPER	R	R	R	R	R	R	R	R
RESET	X	X	X	X	X	X	X	X

MSIZ2-MSIZ0 MSIZ2-MSIZ0 together define the size of the total memory to be controlled by the MCECC pair. These bits reflect the RSIZ2-RSIZ0 bits in the Defaults Register 1.

MSIZ2	MSIZ1	MSIZ0	Memory Size
0	0	0	4MB using one 144-bit wide block of 256Kx4 DRAMs
0	0	1	8MB using two 144-bit wide block of 256Kx4 DRAMs
0	1	0	16MB using one 144-bit wide block of 1Mx4 DRAMs
0	1	1	32MB using two 144-bit wide blocks of 1Mx4 DRAMs
1	0	0	64MB using one 144-bit wide block of 4Mx4 DRAMs
1	0	1	128MB using two 144-bit wide blocks of 4Mx4 DRAMs
1	1	0	Reserved
1	1	1	Reserved

Difference from MEMC040: NONE except that they reflect input pins on the MEMC040; while they reflect register bits that are initialized by the reset serial bit stream on the MCECC.

RB3 Read Bit 3 is a read only bit that is always 0.

Difference from MEMC040: bit = WPB (write-per-bit input strap status) for MEMC040; bit = 0 for MCECC (WPB = 0 on current versions of MVME166/167/187).

RB4 Read Bit 4 is a read only bit that is always 1.

Difference from MEMC040: bit = EXTPEN (external parity enable input strap status) for MEMC040; bit = 1 for MCECC (EXTPEN = 1 on current versions of MVME166/167/187).

FSTRD FSTRD reflects the state of the FSTRD bit in the Defaults Register 1. When 1, this bit indicates that DRAM reads are operating at full speed. When 0, it indicates that DRAM read accesses are slowed by one clock cycle to accommodate slower DRAM devices.

Difference from MEMC040: NONE except that it is an input pin on the MEMC040; while it is a register bit that is initialized by the reset serial bit stream on the MCECC.

Dummy Register 0

Dummy Register 0 is hard-wired to all zeros. Writes to this register are ignored; however, the MCECC always terminates the cycles properly with TA*.

Difference from MEMC040: register = Alternate Status for MEMC040; register = $00 for MCECC.

ADR/SIZ	1st $FFF4300C/2nd $FFF4310C (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	0	0	0	0	0	0	0	0
OPER	R	R	R	R	R	R	R	R
RESET	X	X	X	X	X	X	X	X

Dummy Register 1

Dummy Register 1 is hard-wired to all zeros. Writes to this register are ignored; however, the MCECC always terminates the cycles properly with TA*.

Difference from MEMC040: register = Alternate Control for MEMC040; register = $00 for MCECC.

ADR/SIZ	1st $FFF43010/2nd $FFF43110 (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	0	0	0	0	0	0	0	0
OPER	R	R	R	R	R	R	R	R
RESET	X	X	X	X	X	X	X	X

Base Address Register

These eight bits are combined with the two most significant bits in Register 7 (the next register) to form BAD31-BAD22, which defines the base address of the memory. For larger memory sizes, the lower significant bits are ignored.

Difference from MEMC040: none.

The bit assignments for the Base Address Register are:

ADR/SIZ	1st $FFF43014/2nd $FFF43114 (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	BAD31	BAD30	BAD29	BAD28	BAD27	BAD26	BAD25	BAD24
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS

DRAM Control Register

The bit assignments for the DRAM Control Register are:

ADR/SIZ	1st $FFF43018/2nd $FFF43118 (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	BAD23	BAD22	RWB5	SWAIT	RWB3	NCEIEN	NCEBEN	RAMEN
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
RESET	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS

RAMEN RAM Enable. This control bit is used to enable the local bus to perform read/write accesses to the memory. Accesses are enabled when this bit is set and are disabled when this bit is cleared. *This bit should only be set after BAD31-BAD22 have been initialized.*

Difference from MEMC040: none.

NCEBEN Setting the NCEBEN control bit enables the MCECC pair to assert TEA* when a non-correctable error occurs during a local bus access to memory. In some cases setting NCEBEN causes DRAM accesses to be delayed by one clock. This delay is incurred when the access is a local bus (or scrub) read and the FSTRD bit is set.

Difference from MEMC040: bit = PAREN for MEMC040; bit = NCEBEN for MCECC (both accomplish basically the same thing, enabling TEA assertion for non-correctable errors).

NCEIEN When NCEIEN is set, the logging of a non-correctable error causes the INT signal pin to pulse true. Note that NCEIEN has no effect on DRAM access time.

Difference from MEMC040: bit = PARINT for MEMC040; bit = NCEIEN for MCECC.

RWB3 Read/Write Bit 3 is a general purpose read/write bit.

Difference from MEMC040: bit = WWP (write-wrong-parity) for MEMC040; bit = RWB (general purpose read write bit) for MCECC.

SWAIT Setting the SWAIT control bit causes the MCECC pair to wait for MI* to be negated before starting a DRAM cycle in response to a local bus cycle to DRAM that does not have snooping inhibited. Clearing the SWAIT bit causes the

MCECC pair to start a DRAM read cycle even before MI* is negated during a snooped, local bus cycle. Note that the MCECC pair still waits for MI* to be negated before enabling its data onto the local data bus and asserting TA*/TEA*. Additionally, setting the SWAIT bit causes the MCECC pair to wait for LOCKOK to be asserted before starting a DRAM cycle in response to a local bus cycle to DRAM that has LOCKL asserted. Clearing the SWAIT bit causes the MCECC pair to start a DRAM read even before LOCKOK is asserted during a local bus cycle that has LOCKL asserted. As with MI*, the MCECC pair still waits for LOCKOK to be asserted before enabling its data onto the local data bus and asserting TA*/TEA*. SWAIT should normally be cleared, as it can provide a slight performance gain.

Difference from MEMC040: when bit set - no difference for snooping, when bit cleared - MEMC040 REV. 1 no difference, MEMC040 REV. 0 - MCECC pair waits for MI* negated in all cases of snooped writes whereas MEMC040 REV. 0 does not wait if snooped write is a line push Additionally, for the MEMC040, SWAIT does not affect LOCKL, LOCKOK operation. For the MCECC, SWAIT affects LOCKL, LOCKOK operation as explained.

RWB5 Read/Write Bit 5 is a general purpose read/write bit.

Difference from MEMC040: bit = DMCTL (data-mux-control) for MEMC040; bit = RWB (general purpose read write bit) for MCECC (data-mux-control not required for MCECC pair).

BAD22, BAD23

These are the lower two bits of the DRAM base address described in the previous register.

Difference from MEMC040: none.

BCLK Frequency Register

The Bus Clock (BCLK) Frequency Register should be programmed with the hexadecimal value of the operating clock frequency in MHz (i.e., $19 for 25 MHz and $21 for 33 MHz). The MCECC pair uses the value programmed in this register to control the Prescaler Counter. The Prescaler Counter increments to $FF and then it is loaded with the two's compliment of the value in the BCLK Frequency Register. This produces a 1 MHz clock that is used by the refresh timer and the scrubber. When the BCLK Frequency Register is correctly programmed with the BCLK frequency, the DRAMs are refreshed approximately once every 15.6 microseconds. After power-up, this register is initialized to $19 (for 25 MHz). **Difference from MEMC040: none.**

Note **This register is configured only during power-up-reset and is unchanged by software or local reset.**

ADR/SIZ	1st $FFF4301C/2nd $FFF4311C (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	BCK7	BCK6	BCK5	BCK4	BCK3	BCK2	BCK1	BCK0
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 P	0 P	0 P	1 P	1 P	0 P	0 P	1 P

Note **None of the remaining registers have counterparts in the MEMC040 because they are associated with functions contained only in the MCECC pair.**

Data Control Register

ADR/SIZ	1st $FFF43020/2nd $FFF43120 (16 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	0	0	DERC	ZFILL	RWCKB	0	0	0
OPER	R	R	R/W	R/W	R/W	R	R	R
RESET	X	X	1 PLS	0 PLS	0 PLS	X	X	X

RWCKB READ/WRITE CHECKBITS, when set, enables the data from the eight checkbits in this MCECC to be written and read on the local MC68040 data bus (bits 24-31 for upper MCECC, bits 8-15 for lower MCECC). This bit should be cleared for normal system operation. Note that if test software forces a single bit error to a location (line) using this function, the scrubber may correct the location before the test software gets a chance to check for the single bit error at that location. This can be avoided by disabling scrubbing and making sure that all previous scrubs have completed, before performing the test.

Also note that writing bad checkbits can set the ERRLOG bit in the Error Logger Register. The writing of checkbits causes the MCECC to perform a read-modify-write to DRAM. If the location to which check bits are being written, has a single or double bit err, data in the location may be altered by the write checkbits operation. To avoid this, it is recommended that the DERC bit also be set while the RWCKB bit is set. A suggested sequence for performing read-write checkbits is as follows:

1. Stop all scrub operations by clearing all of the STON bits and setting all of the STOFF bits in the Scrub Time On/Time Off Register.
2. Set the DERC and RWCKB bits in the Data Control Register.
3. Perform the desired read and/or write checkbit operations.
4. Clear the DERC and RWCKB bits in the Data Control Register.
5. Perform the desired testing related to the location/locations that have had their checkbits altered.
6. Allow the scrubber to proceed by restoring the STON and STOFF bits to their original state.

ZFILL ZERO FILL memory, when set, forces all zeros to be written to the DRAM during any kind of write cycle or scrub cycle. It is intended to be used with the zero-fill function. Refer to the section on *Initialization* at the end of this chapter. This bit should be cleared for normal system operation.

DERC DISABLE ERROR CORRECTION, when set to one, disables the MCECC from correcting single bit errors. Specifically, read data is presented to the local MC68040 data bus unaltered from the DRAM array. Less-than-line write data performs a read-modify-write without correcting single bit errors that may occur on the read portion of the read-modify-write. Note that DERC does not affect the generation of check bits. DERC should be cleared during normal system operation. DERC also allows the write portion of a read-modify-write to happen regardless of whether or not there is a multiple bit error during the read portion of the read-modify-write. DERC also affects scrub cycles.

Scrub Control Register

ADR/SIZ	1st $FFF43024/2nd $FFF43124 (8 bits)							
BIT	31	30	29	28	27	26	25	24
NAME	RACODE	RADATA	HITDIS	SCRB	SCRBEN	0	SBEIEN	IDIS
OPER	R/W	R/W	R/W	R	R/W	R	R/W	R/W
RESET	V PLS	0 PLS	V PLS	0 PLS	0 PLS	X	0 PLS	0 PLS

IDIS When cleared, the Image DISable bit allows writes to the upper MCECC control registers to duplicate the data to the lower MCECC control registers. When IDIS is set, the lower MCECC control registers are written separately by the data on D00-D16. IDIS should only be set for test purposes.

SBEIEN Setting SBEIEN causes the logging of a single bit error to create a true pulse on the INT signal pin.

SCRBEN This control bit enables the scrubber to operate. When SCRBEN is set, the MCECC immediately performs a scrub of the entire DRAM array. When the scrub is complete, if software has cleared SCRBEN, then scrubbing is not done again, until software sets the SCRBEN bit. If software has not cleared the SCRBEN bit, then when the amount of time indicated in the Scrub Period (SBPD) Register expires, the MCECC scrubs the DRAM array again. It continues to perform scrubs of the entire DRAM array at the frequency indicated in the SBPD Register. The scrubber does not start a new scrub once the SCRBEN bit is cleared. The time between scrubs is approximately two seconds times the value stored in the SBPD Register. Note that power-up, local, or software reset stops the scrubber.

SCRB This status bit reflects the state of the scrubber. When the scrubber is in the process of doing a scrub, this bit is set. When the scrubber is between scrubs, this bit is cleared.

HITDIS This bit controls a function that is not currently used in the MCECC.

RADATA This bit controls a function that is not currently used in the MCECC.

RACODE This bit controls a function that is not currently used in the MCECC.

Scrub Period Register Bits 15-8

The Scrub Period Control Register controls how often a scrub of the entire memory is performed if the SCRBEN bit is set in the Scrub Control Register. The time between scrubs is approximately two seconds times the value programmed into the Scrub Period Register. The scrub period can be programmed from once every four seconds to once every 36 hours. This register contains bits 15-8 of the Scrub Period Register.

ADR/SIZ	1st $FFF43028/2nd $FFF43128 (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	SBPD15	SBPD14	SBPD13	SBPD12	SBPD11	SBPD10	SBPD9	SBPD8
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	1 PLS	1 PLS	1 PLS	1 PLS	1 PLS	1 PLS	1 PLS	1 PLS

Scrub Period Register Bits 7-0

This register contains bits 7-0 of the Scrub Period Register.

ADR/SIZ	1st $FFF4302C/2nd $FFF4312C (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	SBPD7	SBPD67	SBPD5	SBPD4	SBPD3	SBPD2	SBPD1	SBPD0
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	1 PLS	1 PLS	1 PLS	1 PLS	1 PLS	1 PLS	1 PLS	1 PLS

Chip Prescaler Counter

This register reflects the current value in the prescaler counter. The Prescaler Counter is used with the BCLK Frequency Register to produce a 1 MHz clock signal for use by the refresher, and by the scrubber. The register is readable and writable for test purposes. Programming of this register is not recommended.

ADR/SIZ	1st $FFF43030/2nd $FFF43130 (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	CPS7	CPS6	CPS57	CPS4	CPS3	CPS2	CPS1	CPS0
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 P	0 P	0 P	0 P	0 P	0 P	0 P	0 P

Scrub Time On/Time Off Register

ADR/SIZ	1st $FFF43034/2nd $FFF43134 (8-bits)						
BIT	31	30	29	28	27	26	25	24
NAME	SRDIS	0	STON2	STON1	STON0	STOFF2	STOFF1	STOFF0
OPER	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PLS	0	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS

STOFF2-STOFF0

STOFF2-STOFF0 control the amount of time that the scrubber refrains from requesting use of the DRAM each time it gives it up during a scrub. They control the off time as follows:

STOFF2	STOFF1	STOFF0	Scrubber Time Off
0	0	0	Request DRAM immediately
0	0	1	Request DRAM after 16 BCLK cycles
0	1	0	Request DRAM after 32 BCLK cycles
0	1	1	Request DRAM after 64 BCLK cycles
1	0	0	Request DRAM after 128 BCLK cycles
1	0	1	Request DRAM after 256 BCLK cycles
1	1	0	Request DRAM after 512 BCLK cycles
1	1	1	Request DRAM never

STON2-STON0

STON2-STON0 control the amount of time that the scrubber occupies the DRAM before providing a window during which the local bus and refresher might use it. They control the on time as follows:

STON2	STON1	STON0	Scrubber Time On
0	0	0	Keep DRAM for 1 memory cycle
0	0	1	Keep DRAM for 16 BCLK cycles
0	1	0	Keep DRAM for 32 BCLK cycles
0	1	1	Keep DRAM for 64 BCLK cycles
1	0	0	Keep DRAM for 128 BCLK cycles
1	0	1	Keep DRAM for 256 BCLK cycles
1	1	0	Keep DRAM for 512 BCLK cycles
1	1	1	Keep DRAM for TOTAL SCRUB TIME

Note that if STON2-0 is zero, the scrubber always releases the DRAM after one memory cycle, even if neither the local bus nor refresher need it.

SRDIS SRDIS disables the scrubber from performing reads during scrub cycles. This mode should only be used when using the scrub function to perform zero fill of the DRAM. Setting this bit causes the zero fill to happen faster. This bit should not be changed while scrubbing is in process.

Scrub Prescaler Counter (Bits 21-16)

The Scrub Prescaler Counter uses the 1MHz clock as an input to create the .5 Hz clock that is used for the scrub period. Writes to this address update the scrub prescaler. Reads to this address yield the value in the scrub prescaler. The ability to read and write to the scrub prescaler is provided for test purposes. Programming this counter is not recommended. This register reflects the current value in the scrub prescaler bits 21-16.

ADR/SIZ	1st $FFF43038/2nd $FFF43138 (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	0	0	SPS21	SPS20	SPS19	SPS18	SPS17	SPS16
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS

Scrub Prescaler Counter (Bits 15-8)

ADR/SIZ	1st $FFF4303C/2nd $FFF4313C (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	SPS15	SPS14	SPS13	SPS12	SPS11	SPS10	SPS9	SPS8
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS

This register reflects the current value in the scrub prescaler bits 15-8.

Scrub Prescaler Counter (Bits 7-0)

This register reflects the current value in the scrub prescaler bits 7-0.

ADR/SIZ	1st $FFF43040/2nd $FFF43140 (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	SPS7	SPS6	SPS5	SPS4	SPS3	SPS2	SPS1	SPS0
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS

Scrub Timer Counter (Bits 15-8)

This read/write register is the Scrub Timer Counter. If scrubbing is enabled and the Scrub Period Register is non-zero, the Scrub Timer Counter increments approximately once every two seconds until it matches the value programmed into the Scrub Period Register, at which time, it clears and resumes incrementing. Writes to this address update the Scrub Timer Counter, reads to this address yield its value. The ability to read and write this register is provided for test purposes. Programming this counter is not recommended. This register reflects the current value in the Scrub Timer Counter bits 15-8.

ADR/SIZ	1st $FFF43044/2nd $FFF43144 (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	ST15	ST14	ST13	ST12	ST11	ST10	ST9	ST8
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS

Scrub Timer Counter (Bits 7-0)

This register reflects the current value in the Scrub Timer Counter bits 7-0.

ADR/SIZ	1st $FFF43048/2nd $FFF43148 (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS

Scrub Address Counter (Bits 26-24)

This read/write register is the Scrub Address Counter. Each time the scrubber performs a scrub memory cycle, the Scrub Address Counter increments. For an entire scrub, the Scrub Address Counter starts at 0 and increments until it reaches the DRAM size that is indicated by the MEMSIZ pins. Writes to this address update the Scrub Address Counter; reads to this address yield the value in the Scrub Address Counter. The ability to read and write this counter is provided for test purposes. Note that if scrubbing is in process, the Scrub Time On/Time Off Register should be set for the minimum time on and the maximum time off during any writes to this register. This register reflects the current value in the Scrub Address Counter bits 26-24.

ADR/SIZ	1st $FFF4304C/2nd $FFF4314C (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	0	0	0	0	0	SAC26	SAC25	SAC24
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	X	X	X	X	X	0 PLS	0 PLS	0 PLS

Scrub Address Counter (Bits 23-16)

This register reflects the current value in the Scrub Address Counter bits 23-16.

ADR/SIZ	1st $FFF43050/2nd $FFF43150 (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	SAC23	SAC22	SAC21	SAC20	SAC19	SAC18	SAC17	SAC16
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS

Scrub Address Counter (Bits 15-8)

This register reflects the current value in the Scrub Address Counter bits 15-8.

ADR/SIZ	1st $FFF43054/2nd $FFF43154 (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	SAC15	SAC14	SAC13	SAC12	SAC11	SAC10	SAC9	SAC8
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS

Scrub Address Counter (Bits 7-4)

This register reflects the current value in the Scrub Address Counter bits 7-4.

ADR/SIZ	1st $FFF43058/2nd $FFF43158 (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	SAC7	SAC6	SAC5	SAC4	0	0	0	0
OPER	R/W	R/W	R/W	R/W	R	R	R	R
RESET	0 PLS	0 PLS	0 PLS	0 PLS	X	X	X	X

Error Logger Register

ADR/SIZ	1st $FFF4305C/2nd $FFF4315C (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	ERRLOG	ERD	ESCRB	ERA	EALT	0	MBE	SBE
OPER	R/C	R	R	R	R	R	R	R
RESET	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	X	0 PLS	0 PLS

SBE SINGLE BIT ERROR is set when the last error logged was due to a single bit error. It is cleared when a 1 is written to the ERRLOG bit. The syndrome code reflects the bit in error. (Refer to the section on *Syndrome Decode.*)

MBE MULTIPLE BIT ERROR is set when the last error logged was due to a multiple bit error. It is cleared when a 1 is written to the ERRLOG bit. The syndrome code is meaningless if MBE is set.

ERA This bit provides status for a function that is not currently used in the MCECC.

EALT EALT indicates that the last logging of an error occurred on a DRAM access by an alternate (MI* not asserted) local bus master.

ESCRB ESCRB indicates the entity that was accessing DRAM at the last logging of a single or double bit error. If ESCRB is 1, it indicates that the scrubber was accessing DRAM. If ESCRB is 0, it indicates that the local MC68040 bus master was accessing DRAM.

ERD ERD reflects the state of the local bus READ signal pin at the last logging of a single or double bit error. ERD = 1 corresponds to READ = high and ERD = 0 to READ = low. ERD is meaningless if ESCRB is set.

ERRLOG When set, ERRLOG indicates that a single or a double bit error has been logged by this MCECC, and that no more is logged until it is cleared. The bit can only be set by logging an error and cleared by writing a one to it. When ERRLOG is cleared, the MCECC is ready to log a new error. Note that because hardware duplicates control register writes to both MCECCs, clearing ERRLOG in one MCECC clears it in the other. Any available error information in either MCECC should be recovered before clearing ERRLOG

Error Address (Bits 31-24)

ADR/SIZ	1st $FFF43060/2nd $FFF43160 (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	EA31	EA30	EA29	EA28	EA27	EA26	EA25	EA24
OPER	R	R	R	R	R	R	R	R
RESET	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS

This register reflects the value that was on bits 31-24 of the local MC68040 address bus at the last logging of an error.

Error Address (Bits 23-16)

This register reflects the value that was on bits 23-16 of the local MC68040 address bus at the last logging of an error.

ADR/SIZ	1st $FFF43064/2nd $FFF43164 (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	EA23	EA22	EA21	EA20	EA19	EA18	EA17	EA16
OPER	R	R	R	R	R	R	R	R
RESET	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS

Error Address Bits (15-8)

ADR/SIZ	1st $FFF43068/2nd $FFF43168 (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	EA15	EA14	EA13	EA12	EA11	EA10	EA9	EA8
OPER	R	R	R	R	R	R	R	R
RESET	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS

This register reflects the value that was on bits 15-8 of the local MC68040 address bus at the last logging of an error.

Error Address Bits (7-4)

This register reflects the value that was on bits 7-4 of the local MC68040 bus at the last logging of an error.

ADR/SIZ	1st $FFF4306C/2nd $FFF4316C (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	EA7	EA6	EA5	EA4	0	0	0	0
OPER	R	R	R	R	R	R	R	R
RESET	0 PLS	0 PLS	0 PLS	0 PLS	X	X	X	X

Error Syndrome Register

ADR/SIZ	1st $FFF43070/2nd $FFF43170 (16-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	S7	S6	S5	S4	S3	S2	S1	S0
OPER	R	R	R	R	R	R	R	R
RESET	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS	0 PLS

S7-S0 SYNDROME7-0 reflects the syndrome value at the last logging of an error. The eight bit code indicates the position of the data error. When all the bits are zero, there is no error. Note that if the logged error was non-correctable, then these bits are meaningless. Refer to the section on *Syndrome Decode.*

Defaults Register 1

ADR/SIZ	1st $FFF43074/2nd $FFF43174 (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	WRHDIS	STATCOL	FSTRD	SELI1	SELI0	RSIZ2	RSIZ1	RSIZ0
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PL	V PLS	V PLS	V PLS	V PLS	V PLS	V PLS	V PLS

It is not recommended that non-test software write to this register.

RSIZ2-RSIZ0 RSIZ2-RSIZ0 determine the size of the DRAM array that is assumed by the MCECC. They control the size as follows :

RSIZ2	RSIZ1	RSIZ0	DRAM Array Size
0	0	0	4MB using one 144-bit wide block of 256Kx4 DRAMs
0	0	1	8MB using two 144-bit wide blocks of 256Kx4 DRAMs
0	1	0	16MB using one 144-bit wide block of 1Mx4 DRAMs
0	1	1	32MB using two 144-bit wide blocks of 1Mx4 DRAMs

1	0	0	64MB using one 144-bit wide block of 4Mx4 DRAMs
1	0	1	128MB using two 144-bit wide blocks of 4Mx4 DRAMs
1	1	0	Reserved
1	1	1	Reserved

The states of RSIZ2-0 after power-up, soft, or local reset, match those of the RSIZ2-0 bits from the reset serial bit stream.

SELI1, SELI0 The SELI1, SELI0 control bits determine the base address at which the control and status registers respond as shown below:

SELI1	SELI0	Register Base Address
0	0	$FFF43000
0	1	$FFF43100
1	0	$FFF43200
1	1	$FFF43300

The states of SELI1 and SELI0 after power-up, soft, or local reset, match those of the SELI1 and SELI0 bits from the reset serial bit stream.

FSTRD The FSTRD control bit determines the speed at which DRAM reads occur. When it is 1, DRAM reads happen at full speed. When it is 0, DRAM reads are slowed by one clock, unless they are already slowed by NCEBEN being set. FSTRD is cleared by Power-up or Local Reset if the FSTRD bit in the reset serial bit stream is 0. It is set by Power-up, soft, or Local Reset if the FSTRD bit in the reset serial bit stream is 1. Note that this bit can also be read in the Memory Configuration Register.

STATCOL When the STATCOL bit is set, the RACODE and/or RADATA bits in the Scrub Control Register can be set. When it is cleared, they cannot. STATCOL is initialized by power-up, soft, or local reset to match the value of the STATCOL bit in the reset serial bit stream.

WRHDIS This bit controls a function that is not currently used in the MCECC.

Defaults Register 2

ADR/SIZ	1st $FFF43078/2nd $FFF43178 (8-bits)							
BIT	31	30	29	28	27	26	25	24
NAME	FRC_OPEN	XY_FLIP	REFDIS	TVECT	NOCACHE	RESST2	RESST1	RESST0
OPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PLS	0 PLS	0 PLS	V PLS	V PLS	V PLS	V PLS	V PLS

It is not recommended that non-test software write to this register.

RESST2-RESST0

These general purpose read/write bits are initialized by power-up, soft, or local reset, to match the RESST2-RESST0 bits from the reset serial bit stream.

NOCACHE When NOCACHE is cleared, the HITDIS bit in the Scrub Control Register can be cleared by software. When it is set, the HITDIS bit cannot be cleared. NOCACHE is initialized by power-up, soft, or local reset to match the NOCACHE bit in the reset serial bit stream. It should always be left at the default value of 1.

TVECT TVECT makes bidirectional signals work while running the vendors test vectors on this chip. It should be cleared for normal operation. It is initialized by power-up, soft, or local reset, to match the TVECT bit from the reset serial bit stream.

REFDIS When REFDIS is set, refreshing is disabled. This mode should only be used for testing, as DRAM must have refresh to operate correctly. REFDIS is initialized by power-up, soft, or local reset to match the REFDIS bit in the reset serial bit stream.

XY_FLIP When XY_FLIP is set, the opposite internal set of cache latches is selected. This bit should be used with caution and is for test vector coverage improvement.

FRC_OPN When FRC_OPN is set, the internal DRAM read latches are forced continuously open. This bit should be used with caution and is for test vector coverage improvement.

Initialization

Most DRAM vendors require that the DRAMs be subjected to some number of access cycles before the DRAMs are fully operational. The MCECC does not perform this automatically but depends on software to perform enough dummy accesses to DRAM to meet the requirement. The number of required cycles is less than 10. If there are multiple blocks of DRAM, software has to perform at least 10 accesses to each block.

The MCECC pair provides a fast zero fill capability. The sequence shown below performs such a zero fill. It zeros all of the DRAM controlled by this MCECC pair at the rate of 100 MB/second when the BCLK pin is operating at 25 MHz. This sequence may have to be altered to perform the scrub more slowly if the scrub causes the DRAM to consume too much power at full speed.

1. Make sure that the scrubber is disabled by clearing the SCRBEN bit in the Scrub Control Register. (Clear bit 27 of offset $24.)
2. Make sure that the scrubber is done with any old scrub cycles by waiting for the SCRB bit in the Scrub Control Register to be cleared. (Wait for bit 28 of offset $24 = 0.)
3. Discontinue all accesses from the MC68040 bus to the DRAM.
4. Ensure that all accesses have stopped by clearing the RAMEN bit in the DRAM Control Register. (Clear bit 0 of offset $18)
5. Set the ZFILL bit in the MCECC pair. (Set Bit 28 of offset $20)
6. Set the Scrub Time On/Time Off Register for the maximum rate and to do write cycles, by setting the SRDIS bit, setting all of the STON bits, and clearing all of the STOFF bits. (Write $B8 to offset $34)
7. Enable scrubbing by setting the SCRBEN bit in the Scrub Control Register. (Set bit 27 of offset $24.)
8. Ensure that the zero-fill has started by waiting for the SCRB bit in the Scrub Control Register to be set. (Wait for bit 28 of offset $24 = 1.)
9. Ensure that the zero-fill stops after one time through, by clearing the SCRBEN bit in the Scrub Control Register. (Clear bit 27 of offset $24.)
10. Wait for the zero-fill to complete by waiting for the SCRB bit in the Scrub Control Register to be cleared. (Wait for bit 28 of offset $24 = 0.)
11. Clear the ZFILL bit in the MCECC pair. (Clear Bit 28 of offset $20)
12. The entire DRAM that is controlled by this MCECC is now zero-filled. The software can now program the appropriate scrubbing mode and other desired initialization, and enable DRAM for operation.

Syndrome Decode

A syndrome code value of $00 indicates no error found. All other syndrome code values indicate an error with the bit in error decoded as shown in the following table. Note that BANK A corresponds to A3,A2 = 00, BANK B to A3,A2 = 01, BANK C to A3,A2 = 10, and BANK D to A3,A2 = 11.

Bank in Error	Bit in Error	Syndrome Code
BANK D	BIT 0/16	$8C
BANK D	BIT 1/17	$0D
BANK D	BIT 2/18	$0E
BANK D	BIT 3/19	$F4
BANK D	BIT 4/20	$15
BANK D	BIT 5/21	$16
BANK D	BIT 6/22	$26
BANK D	BIT 7/23	$25
BANK D	BIT 8/24	$19
BANK D	BIT 9/25	$1A
BANK D	BIT 10/26	$1C
BANK D	BIT 11/27	$E9
BANK D	BIT 12/28	$2A
BANK D	BIT 13/29	$2C
BANK D	BIT 14/30	$4C
BANK D	BIT 15/31	$4A

Bank in Error	Bit in Error	Syndrome Code
BANK C	BIT 0/16	$23
BANK C	BIT 1/17	$43
BANK C	BIT 2/18	$83
BANK C	BIT 3/19	$3D
BANK C	BIT 4/20	$45
BANK C	BIT 5/21	$85
BANK C	BIT 6/22	$89
BANK C	BIT 7/23	$49
BANK C	BIT 8/24	$46

BANK C	BIT 9/25	$86
BANK C	BIT 10/26	$07
BANK C	BIT 11/27	$7A
BANK C	BIT 12/28	$8A
BANK C	BIT 13/29	$0B
BANK C	BIT 14/30	$13
BANK C	BIT 15/31	$92

Bank in Error	Bit in Error	Syndrome Code
BANK B	BIT 0/16	$C8
BANK B	BIT 1/17	$D0
BANK B	BIT 2/18	$E0
BANK B	BIT 3/19	$4F
BANK B	BIT 4/20	$51
BANK B	BIT 5/21	$61
BANK B	BIT 6/22	$62
BANK B	BIT 7/23	$52
BANK B	BIT 8/24	$91
BANK B	BIT 9/25	$A1
BANK B	BIT 10/26	$C1
BANK B	BIT 11/27	$9E
BANK B	BIT 12/28	$A2
BANK B	BIT 13/29	$C2
BANK B	BIT 14/30	$C4
BANK B	BIT 15/31	$A4

Bank in Error	Bit in Error	Syndrome Code
BANK A	BIT 0/16	$32
BANK A	BIT 1/17	$34
BANK A	BIT 2/18	$38
BANK A	BIT 3/19	$D3
BANK A	BIT 4/20	$54

BANK A	BIT 5/21	$58
BANK A	BIT 6/22	$98
BANK A	BIT 7/23	$94
BANK A	BIT 8/24	$64
BANK A	BIT 9/25	$68
BANK A	BIT 10/26	$70
BANK A	BIT 11/27	$A7
BANK A	BIT 12/28	$A8
BANK A	BIT 13/29	$B0
BANK A	BIT 14/30	$31
BANK A	BIT 15/31	$29

Bank in Error	**Bit in Error**	**Syndrome Code**
UPPER/LOWER CHECKBITS	BIT 0	$01
UPPER/LOWER CHECKBITS	BIT 1	$02
UPPER/LOWER CHECKBITS	BIT 2	$04
UPPER/LOWER CHECKBITS	BIT 3	$08
UPPER/LOWER CHECKBITS	BIT 4	$10
UPPER/LOWER CHECKBITS	BIT 5	$20
UPPER/LOWER CHECKBITS	BIT 6	$40
UPPER/LOWER CHECKBITS	BIT 7	$80

VSBchip2 6

Introduction

This chapter describes the VSB interface chip ASIC (VSBchip2) used only on the MVME166 boards. The VSBchip2 is an ASIC designed to provide a fully functional master/slave interface between the VME Subsystem Bus (VSB) and an MC68040-compatible bus (Local Bus).

Summary of Major Features

- ❑ Local Bus to VSB Interface:
  - Four programmable local bus to VSB map decoders.
    - Each decoder includes a 16-bit address offset register.
    - Independent programmable attributes for each decoder
      - VSB space codes.
      - Separate read and write enables.
      - Write post enable.
      - Bounce mode enable.
  - VSB master generates 8, 16, or 32 bit single- or block-transfer cycles.
  - Local bus slave accepts 8, 16, or 32 bit single- or burst-transfer cycles.
  - Supports dynamic bus sizing on VSB.
  - Single level write post buffer.
  - Programmable timers
    - VSB access timer.
    - VSB Address and Data transfer timer.
  - VSB Requester:
    - Programmable FAIR request mode
    - Programmable release modes (serial mode only):
      - Release When Done (RWD).
      - Release On Request (ROR).
    - Programmable Parallel Arbitration ID.
  - Bounce output pin.

  - Local timer disable output pin.
- VSB to Local Bus Interface:
  - Two programmable VSB to local bus map decoders.
    - Each decoder includes 16-bit address offset register.
    - Independent programmable attributes for each decoder:
      - Participating/Responding slave read and write enables.
      - VSB address space select.
      - Local bus lock on block transfer enable.
      - Write post enable.
      - Snoop attribute select.
      - Local bus transfer size select.
  - Local bus master generates 8, 16, or 32 bit single-transfer cycles.
  - VSB slave accepts 8, 16, or 32 bit single- or block-transfer cycles.
  - Additional VSB cycles supported:
    - Data broadcall.
    - Data broadcast.
    - Interrupt acknowledge.
  - Single level write post buffer.
- Board Control and Status Register (BCSR) Set:
  - Supports EVSB Register Set.
- Local Interrupter:
  - Sources
    - Local Bus Write post error.
    - VSB Write post error.
    - VSB IRQ asserted.
    - VSB serviced locally requested interrupt.
    - EVSB Attention Register ATTN bit set.
  - Independent programmable control over each source:
    - Mask bit.
    - Unique vector.
    - Priority Level.

- ❑ VSB interrupter:
  - Sources
    - VSB Interrupt Status Register VSWIF bit set.
    - VSB Write Post Error.
  - Unique vector for each source.
  - Programmable FAIR request mode.
- ❑ VSB interrupt handler:
  - Parallel multi-source handler with programmable arbitration ID.
  - VSB IACK cycles generated automatically in response to a local IACK cycle servicing the VSB IRQ asserted interrupt.
  - Programmable local vector used if VSB IACK cycle fails.

Functional Description

The following sections provide an overview of the functionality of the VSBchip2. See Figure 6-1 for a block diagram of the VSBchip2. Detailed descriptions of all registers are provided later in this chapter.

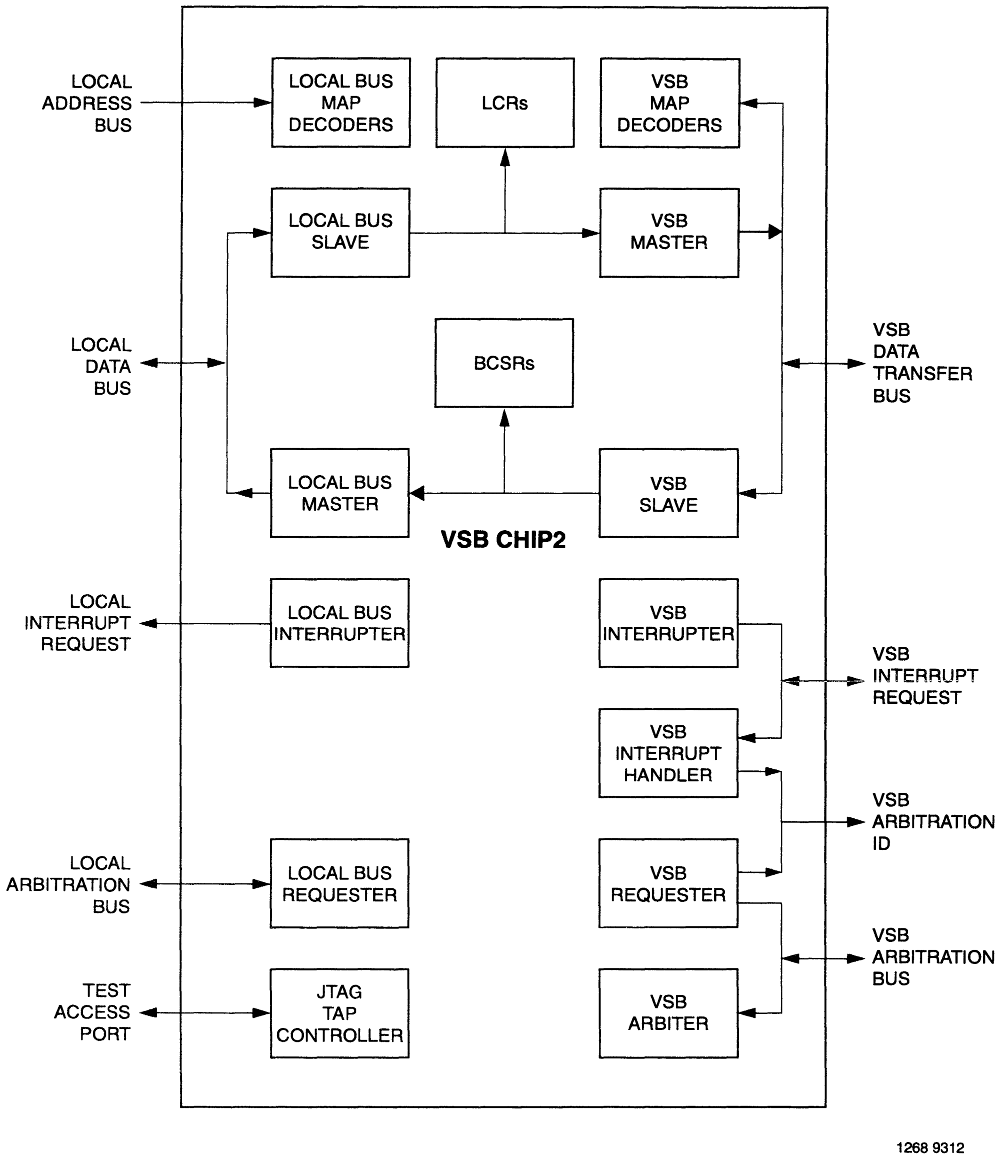


Figure 6-1. VSBchip2 Block Diagram

VSB to Local Bus Interface

The VSB to local bus interface allows a VSB device access to local bus resources. This module includes the VSB slave interface, two programmable map decoders, write post buffer, and local bus master interface.

VSB Slave Interface

The VSB slave interface includes one fixed map decoder, two programmable map decoders, and a write post buffer. To support EVSB, the Board Control and Status Registers (BCSRs) are designed to overlay a non-volatile memory which contains board specific information. The VSBchip2 supports this by giving the fixed map decoder precedence over the programmable map decoders. If one of the programmable map decoders is set to respond to an address also covered by the fixed map decoder, the fixed map decoder is the only one to respond.

In some multi-processing situations, it may be beneficial to perform broadcast and broadcall operations. To support this, the VSBchip2 VSB slave interface can be programmed to act as a participating slave as well as a responding slave.

The VSBchip2 can also be programmed to respond to only read transfers, to only write transfers, or to both. Additionally, it can be programmed to reside in any of the three VSB Address Spaces: System (SAS), Alternate (ALTAS), and/or I/O (IOAS).

Programmable Map Decoders

The VSBchip2 includes two programmable map decoders that allow software to configure the VSB addressing range of local bus resources. The decoders allow the local address range to be partitioned into two separate banks, each with its own start and end address (in increments of 64 KB). Each map decoder includes a 16-bit offset register. The contents of the offset register are added to the upper 16 bits of the incoming VSB address before the address is passed on to the local bus master. This allows the address of local resources to differ from their VSB address. Associated with each decoder is an attribute register which controls each bank's local bus transfer size, local bus snoop codes, local bus lock, VSB participating/responding slave enable, VSB read enable, VSB write enable, VSB Address Space, and VSB write posting capability.

Write Post Buffer

The VSB slave can be programmed to perform write posting operations. When in this mode, the chip latches incoming VSB data and addressing information into a write post buffer and immediately acknowledges the transfer. The VSB is then free for transfers between other devices while the VSBchip2 requests control of the local bus, waits for a local bus grant, and completes the write transfer. The write post buffer stores the data from one byte, word, or longword data transfer. If any VSB to local bus transfer begins before a previous write-posted cycle has completed, that transfer is not acknowledged until the previous write-posted cycle has completed.

Write posting should only be enabled when bus errors are not expected. Using the programmable map decoders, write posting can be enabled for "safe" areas and disabled for areas which are not "safe". If the VSBchip2 detects a bus error during a write posted cycle, this condition is reflected in the Local Interrupt Status Register and the VSB Interrupt Status Register, and a local bus and/or a VSB interrupt may be generated. The address contained in the write post buffer is saved in the VSB Error Address Register, and the specific cause of the error is recorded in the VSB Error Status Register.

Local Bus Master Interface

The local bus master is designed to act exactly as an MC68040 would within the described limits of this chapter. It generates byte, word, and longword single-transfer cycles. It does not generate burst-transfers because there is no equivalent on VSB. The local bus master drives the appropriate local bus snoop control bits and responds to snoop hits correctly.

One of the programmable attributes for each VSB map is the local bus transfer size. This feature was included because the local bus does not directly support dynamic bus sizing. In some applications, there may be an 8- or 16-bit device on the local bus. By programming the local bus transfer size appropriately, a VSB device could communicate with the local device without restricting VSB transfer sizes. The local bus master will take care of translating the VSB transfers to the appropriate size on the local bus as shown in the following table.

Table 6-1. Local Bus Transfer Size

Port Size	LBTS		VSIZE		VAD		VASACK*		LSIZ	
	1	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**
No Response	1	1	X	X	X	X	1	1	X	X
8-bit	1	0	X	X	X	X	1	0	0	1
16-bit	0	1	0	1	X	X	0	1	0	1
	0	1	1	0	X	0	0	1	1	0
	0	1	1	1	X	0	0	1	1	0
	0	1	0	0	X	0	0	1	1	0
	0	1	1	0	X	1	0	1	0	1
	0	1	1	1	X	1	0	1	0	1
	0	1	0	0	X	1	0	1	0	1
32-bit	0	0	0	1	X	X	0	0	0	1
	0	0	1	0	0	0	0	0	1	0
	0	0	1	0	0	1	0	1	0	1
	0	0	1	0	1	0	0	0	1	0
	0	0	1	0	1	1	0	0	0	1
	0	0	1	1	0	0	0	1	1	0
	0	0	1	1	0	1	0	1	0	1
	0	0	1	1	1	0	0	0	1	0
	0	0	1	1	1	1	0	0	0	1
	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	1	0	1	0	1
	0	0	0	0	1	0	0	0	1	0
	0	0	0	0	1	1	0	0	0	1

VSB Block Transfer to a Local Bus Burst

The VSB slave is capable of receiving VSB block transfer cycles. Each data transfer in the VSB block sequence appears on the local bus as an individual transfer. This is not the most efficient use of VSB block transfers, but unfortunately, because there is no way to know how large the block transfer is going to be, it is not possible to translate these to local bus burst transfers.

Each programmable map can be programmed to lock the local bus during VSB block transfer cycles. This mode can improve data throughput by circumventing the need for local bus arbitration between each data transfer. On VSB it is not possible to determine if a block transfer is in progress until after the first data transfer is complete. After the first data transfer, the negation of PAS* can be used to detect the end of a block. When in local bus lock mode, on the first data transfer of a block, the local bus master acquires the local bus, transfers the data, but does not release the local bus. On subsequent data transfers, the local master can perform transfers without the delay normally caused by acquiring the local bus. After the last cycle of the locked transfer has been completed, the local bus is released.

Note **This mode should be used with care. For very long VSB block transfers, local bus devices could be locked off the local bus too long.**

Local Bus to VSB Interface

The Local bus to VSB interface allows local bus devices access to resources on the VSB. This module includes the local bus slave interface, four programmable map decoders, a write post buffer, and a VSB master interface.

Local Bus Slave Interface

The local bus slave includes four independent programmable map decoders and two fixed map decoders. The two fixed map decoders are used to decode the addresses of the Local Control and Status Registers (LCSRs) and the Board Control and Status Registers (BCSRs) respectively.

When a local bus address falls within the range of one of the programmable map decoders, the VSBchip2 assumes control over the local bus time-out using its internal VSB access and VSB transfer timers. The local bus slave asserts the LBTODIS* output pin to turn off any external timers for the remainder of this transfer.

Programmable Map Decoders

The VSBchip2 includes four map decoders that allow software to configure the local bus addressing range of VSB resources. The decoders allow the VSB address range to be partitioned into four separate banks, each with its own start and end address (in increments of 64 KB). Each map decoder includes a 16-bit offset register. The contents of the offset register are added to the upper

16 bits of the incoming local bus address before the address is passed on to the VSB master. This allows the address of VSB resources to differ from their local address. Associated with each decoder is an attribute register, which controls each bank's VSB space codes and write posting capability.

Bounce Mode

Bounce mode is a means of prioritizing transfers over VSB and VME, and allows VME and VSB local bus slave mappings to overlap. When bounce mode is enabled, VSB assumes the higher priority, and each transfer is attempted on VSB first. If the transfer fails on VSB, it is then attempted on VME.

If the VSBchip2 local bus slave receives a "no response" signal back from the VSB master, it can be programmed to carry out one of two courses of action. If bounce mode is enabled, the local bus slave asserts the BOUNCE output pin and negates the LBTODIS* pin until it detects the end of the current local bus transfer. If bounce mode is not enabled, the local bus slave asserts LTEA* to terminate the transfer. The BOUNCE output pin is asserted 1 clock after the local bus TS* is detected for cycles which are not decoded by the VSBchip2.

For local bus burst transfers, BOUNCE is asserted only if the "no response" condition occurred on the first transfer attempt on VSB. On subsequent transfers, the "no response" condition is treated as a bus error, and the local burst is terminated accordingly.

Write Post Buffer

The local bus slave can be programmed to perform write posting operations. When in this mode, the chip latches incoming local bus data and addressing information into a write post buffer and immediately acknowledges the transfer. The local bus is then free to perform transfers between other devices while the VSBchip2 requests control of the VSB, waits for a VSB grant, and completes the write transfer. The write post buffer stores the data from one byte, word, longword, or burst data transfer. If a local bus write transfer begins before a previous write-posted cycle has completed, that transfer is not acknowledged until the previously write-posted cycle has completed.

Write posting should only be enabled when bus errors are not expected. Normal memory cards never return a bus error on a write cycle. However, some ECC memory cards which reside on VSB 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. Using the programmable map decoders, write posting can be enabled for "safe" areas and disabled for areas which are not "safe". If the VSBchip2 detects a bus error during a write-posted cycle, this condition is reflected in the Local Interrupt Status Register, and a local bus interrupt may be generated. The address contained in the write post buffer is saved in the Local Bus Error Address Register, and the specific cause of the error is recorded in the Chip Control/Status Register.

VSB Master Interface

The VSB master supports data broadcast and data broadcall operations on the VSB. If no VSB device is programmed to respond to the current VSB cycle, the VSB master terminates the VSB cycle and passes this information back to the local bus slave.

VSB Dynamic Bus Sizing

The VSBchip2 supports dynamic bus sizing on the VSB. For example, when a local device initiates a D32 access to a VSB slave that only has D16 data transfer capability, the chip executes two word transfer cycles on the VSB and acknowledges the transfer on the local bus side after all requested data has been transferred. This enhances the portability of software because it allows software to run on the system regardless of the physical organization of global memory.

VSB Timers

There are two programmable timers which control the operation of the VSB master. The VSB access timer measures the time from the VSB master bus request until the VSB requester has gained control of the bus. The VSB transfer timer measures two different periods. During the address broadcast phase, it measures the time from the assertion of VSB address until a VSB device has acknowledged receipt of the address. For the data transfer phase, it measures the time from the beginning of a data transfer until a VSB device has acknowledged the data transfer. Note that for block transfers, the VSB transfer timer starts over at the beginning of each data transfer.

The VSB access timer actually measures the time from the assertion of the VSB master's bus request to the assertion of bus busy by the VSB requester. The VSB transfer timer actually measures the time from assertion of the address on VSB to the receipt of AC high OR at least one ASACK* active and WAIT* high. It also measures the time from the assertion of data (write cycle) or assertion of DS* (read cycle) to the receipt of ACK* low.

Normally, if the VSB is not too heavily loaded, the VSB arbiter grants the VSB master the bus before the VSB access timer expires. However, for a heavily loaded bus, or for situations where some circuitry may be broken, the VSB access timer expires, and the current access attempt is suspended. If the VSB access timer expires, the appropriate error bit is set in the Chip Control/Status

Register, and either the local bus TEA* is asserted to terminate the cycle (no write posting) or the LWPIF bit in the Local Interrupt Status Register is set (write posted cycle). System software must then decide whether to retry the cycle or record the error.

The VSB transfer timer is included to guard against lockup due to certain hardware failures. Normally, the VSB address broadcast phase is terminated when each VSB slave releases AC to high. If, however, any slave continues to drive this signal low, the timer expires, and the transfer is aborted. During the VSB data transfer phase, the responding and/or participating slaves assert ACK* and release WAIT*. If, for some reason, one of these signals is stuck or not driven correctly, the VSB transfer timer expires, and the cycle is aborted. If the VSB transfer timer expires, the appropriate error bit is set in the Chip Control/Status Register, and either the local bus TEA* is asserted to terminate the cycle (no write posting) or the LWPIF bit in the Local Interrupt Status Register is set (write posted cycle).

VSB Block Transfers

The VSBchip2 attempts to generate VSB block transfer cycles when multiple VSB transfers are necessary due to a local bus burst transfer.

Local bus burst cycles are not required to be burst aligned (i. e., on even 16-byte boundaries). The local bus address determines the destination of the first longword. The destination of the next longword is determined by incrementing the address by four, unless incrementing by four would cross an even 16-byte boundary. If a boundary would be crossed, the destination address "wraps" back to the previous 16-byte boundary. For example, for a local bus burst which begins at address $00003214, the four longwords would actually be destined for addresses $00003214, $00003218, $0000321C, $00003210 respectively. Because each VSB block transfer must be to the next linear address, it may be necessary to divide local bus burst transfers into at least two VSB blocks. In the example above, the first three longwords could be sent as a block on VSB, but the fourth longword would require the VSB address to be reissued. Each local bus burst transfer is converted into a VSB block transfer sequence until the address "wraps" back to the beginning of the local bus burst boundary. At this point, the VSB address is reissued and a new block begun.

If at any time the responding and/or participating VSB slave wishes to break a block transfer sequence, the VSBchip2 reissues the VSB address and starts another VSB block.

VSB Requester and VSB Serial Arbiter

The VSBchip2 contains all the necessary circuitry to implement a serial VSB requester, a serial VSB arbiter, and a parallel VSB requester. The arbitration mode used is determined by the state of the VPARMD* input pin and the geographical address input pins (VGA2 - VGA0) upon power-up. Parallel VSB requester mode is supported by the MVME166.

VSB Geographical Addressing

The VSB specification assigns each slot in a VSB backplane a unique address using the GA2 - GA0 signals. When a board is installed in a backplane slot, it uses the addresses on these lines as part of its interrupt and parallel arbitration ID's and to determine which board contains the active VSB arbiter. In addition, the VSBchip2 uses the geographical address to determine the placement of the Board Control and Status Registers. The VSB specification defines the geographical addresses as follows:

VSB Slot	GA2	GA1	GA0
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1

Most of Motorola's system products do not have a VSB backplane. Because of this, they do not implement the required geographical addresses or the bus grant/request daisy chain. In order for the VSBchip2 to be compatible with many existing systems as well as the VSB specification, several assumptions are made.

The VSBchip2 VGA2 - VGA0 input pins have internal pull-up resistors. If they are not connected to the appropriate VSB geographical addresses, they will always read as %111, an illegal combination according to the VSB specification. In this case, the VSBchip2 assumes it resides in a system without geographical addresses. Parallel arbitration is disabled regardless of the state of the VPARMD* pin, and the serial requester is enabled by default. The VSBchip2 drives VBGOUT* appropriately to configure the rest of the VSB subsystem requesters. The serial arbiter is enabled if software initializes SGA2 - SGA0 in the Chip Control/Status Register to %000. It is the responsibility of the system software to assure that each board in the VSB subsystem is configured with a unique geographical address and that one of those boards is at %000.

If the VGA2 - VGA0 input pins are not %111 after power-up, the VSBchip2 assumes it is in a system with a fully VSB-compliant backplane. If VGA2 - VGA0 are %000, the VSBchip2 samples the VPARMD* input pin to determine the request mode. It then drives the VBGIN*/VBGOUT* daisy chain appropriately to configure the other requesters in the system. If the VPARMD* pin is low, parallel arbitration mode is selected. Systems integrators must be aware that if the board in VSB slot 1 is configured as a parallel requester, all other boards in that VSB subsystem MUST also have parallel requester capability. If serial only boards are placed in a system configured for parallel request mode, the VSB subsystem may become deadlocked.

VSB Requesters

In parallel mode, the VSB requester that currently has the bus performs an arbitration cycle to determine which requesting device gets it next. All requesters that have a request pending participate in the arbitration cycle. There is no parallel arbiter. In serial mode, each requester submits its bus request to one system arbiter. Only the arbiter in VSB slot 1 (SGA2 - SGA0 are %000) will be the active system arbiter. All other serial arbiters are disabled.

The VSB requester issues a bus request under the following conditions:

The VSB master wishes to perform a VSB cycler,
OR
Some external circuitry has asserted the DWB* input pin,
OR
System software has set the DWB bit in the VSB Requester Control Register.

The VSB requester may be programmed to implement a "fairness" mode to assure that all VSB masters have equal access to the VSB. In fairness mode, any requester which has just released the VSB refrains from requesting it again until VBREQI* is high, indicating no other requests are pending.

When operating in the serial mode, the VSB requester may be programmed to implement one of two different release modes: Release-When-Done (RWD) or Release-On-Request (ROR). Release-When-Done specifies that the requester does not release the bus until its associated master no longer needs it. Release-On-Request means the requester releases the bus only when its associated master no longer needs it AND some other requester has a request pending.

When the VSBchip2 is operating in parallel arbitration mode, the active parallel requester generates a VSB parallel arbitration cycle to transfer bus mastership. Each requester which has a request pending drives a 7-bit arbitration vector onto the VSB data bus. This vector is composed of SGA2 - SGA0 from the Chip Control/Status Register appended to VARBID3 - VARBID0 from the VSB Requester Control/Status Register. The arbitration process is described in the VSB specification section 3.4.2.

VSB Serial Arbiter

Only one serial arbiter may be active in a VSB subsystem. The VSBchip2 serial arbiter is active only if SGA2 - SGA0 in the Chip Control/Status Register are %000, and serial arbitration mode is selected by driving the input pin VPARMD* to a high.

Arbitration Timer

The VSBchip2 includes an arbitration timer which measures the time between when its arbiter asserts bus grant and when a VSB requester assumes control of the bus. This timer prevents a bus lock-up condition caused when no requester assumes control of the bus after a grant was issued. When the timer expires, the arbiter asserts bus busy temporarily as if it is the responding requester and then re-arbitrates any pending bus requests. The VARTO bit in the Chip Control/Status Register is set each time the arbitration timer expires. An arbitration time-out is not normally treated as a fatal error condition because the arbitration is retried, but continued time-outs may be an indication of a bad VSB arbiter/requester or an improper subsystem configuration.

VSB Interrupter

The VSBchip2 has two sources for generating a VSB Interrupt: a VSB Write Post Error, and the VSWIF bit in the VSB Interrupt Status Register.

The VSB interrupter generates a VSB Write Post Error interrupt every time a VSB write posted cycle is aborted because of a local bus time-out or bus error if the VGIE and VWPIE bits are set in the VSB Interrupt Enable Register enabling the VSBchip2 to assert the output pin VIRQO*.

When the VSWIF bit in the VSB Interrupt Status Register is set, and the VGIE and VSWIE bits in the VSB Interrupt Enable Register are set, the VSB interrupter generates a VSB interrupt by asserting its output pin VIRQO*.

When the VSB interrupter within the VSBchip2 detects the VSB master is executing a VSB interrupt acknowledge cycle, this interrupter responds with the 8-bit vector contained in the VSB Interrupt Vector Register and then clears the interrupt request. The lowest order bit of the interrupt vector is unique for each interrupt source. The VSBchip2 only responds to an interrupt-acknowledge cycle if the VEN bit in the VSB Interrupt Control Register is set.

Interrupt Vector Bit 0	**Source**
0	VSB Write Post Error Interrupt
1	Software Interrupt

If there is no active VSB interrupt handler, the interrupt bits may be polled and cleared by system software.

The VSB interrupter may be programmed to implement a "fairness" mode to assure that all VSB interrupters have an equal opportunity to be serviced. In fairness mode, any interrupter which has just been serviced refrains from generating another interrupt until VIRQI* is high, indicating no other interrupt requests are pending.

The address broadcast portion of a VSB interrupt acknowledge is used to determine which interrupt is to be serviced. Each interrupt which has a request pending drives a 7-bit arbitration vector onto the VSB data bus. This vector is composed of SGA2 - SGA0 from the Chip Control/Status Register appended to VINTID3 - VINTID0 from the VSB Interrupt Control Register. The interrupt arbitration process is described in the VSB specification section 2.5.4.

After the highest priority interrupt requester has been selected, that requester uses the data transfer portion of the VSB interrupt acknowledge to pass its vector back to the VSB interrupt handler.

VSB Interrupt Handler

The VSB interrupt handler will generate a VSB interrupt acknowledge cycle automatically when it is enabled, a VSB interrupt is pending, and a local bus interrupt acknowledge cycle is performed to service the VSB interrupt. If no VSB interrupter responds to the VSB interrupt acknowledge cycle, an 8-bit vector derived from the Local Interrupt Vector Base Register will be returned. (Refer to the Local Bus Interrupter description that follows.)

System software is responsible for assuring that only one VSB interrupt handler is enabled in a VSB subsystem at any given time.

Local Bus Interrupter

There are five sources of local bus interrupts: Local Write Post Error, VSB Write Post Error, VSB Interrupt Pending, EVSB Attention Interrupt, and VSB Interrupt Acknowledge Complete. Any of these sources can be programmed to generate a local bus interrupt at any level.

When an interrupt acknowledge cycle is executed to service these interrupts, the vector driven onto the local bus is derived from the Local Bus Interrupt Vector Register as shown below.

Interrupt Vector Bits 3-0	Source
$0	Local Write Post Error Interrupt
$1	VSB Write Post Interrupt
$2	VSB Interrupt Pending
$4	EVSB Attention Interrupt
$5	VSB Interrupt Acknowledge Complete

A Local Write Post Error interrupt is generated any time an error is detected during completion of a local bus transfer which has been write posted. This interrupt is cleared either automatically when serviced by a local bus interrupt acknowledge cycle or under software control.

A VSB Write Post Error interrupt is generated any time an error is detected during completion of a VSB transfer which has been write posted. This interrupt is cleared either automatically when serviced by a local bus interrupt acknowledge cycle or under software control.

A VSB IRQ Pending interrupt is generated any time the VSB IRQ signal is asserted. This interrupt can be cleared only by clearing the source of the interrupt on VSB. When this interrupt is serviced by a local bus interrupt acknowledge cycle, the source of the returned vector is programmable. If the VSB interrupt handler is enabled, it performs an interrupt acknowledge cycle on the VSB and passes the resulting vector back to the local bus. If the VSB interrupt handler is disabled or the VSB interrupt acknowledge cycle is unsuccessful, the vector driven onto the local bus is derived from the Local Bus Interrupt Vector Register as shown above.

An EVSB Attention interrupt is generated when the ATTN bit in the EVSB Attention Register is set. This interrupt is cleared either automatically when serviced by a local bus interrupt acknowledge cycle or under software control.

A VSB Interrupt Acknowledge Complete Interrupt is generated when the VSWIF bit in the VSB Interrupt Status Register is cleared. This interrupt is cleared either automatically when serviced by a local bus interrupt acknowledge cycle or under software control.

The VSBchip2 includes the means to merge an externally generated prioritized interrupt with those generated internally. When an external interrupt is detected on pins LIPLI2 - LIPLI0, its priority level is compared to any pending internal interrupts, and the highest priority level is output on the local bus interrupt level pins LIPLO2 - LIPLO0. This interrupt must be serviced and cleared at its source.

Control and Status Registers

The VSBchip2 includes two sets of registers. The Local Control/Status Registers are accessible only from the local bus. These registers are described in detail next in this chapter. The Board Control/Status Registers are accessible from both the local bus and VSB. These registers are described in detail later in this chapter.

Local Control and Status Registers Programming Model

The VSBchip2 contains twenty-three Local Bus Control and Status Registers (LCSRs). These LCSRs are accessible only through the local bus interface. Each register can be read or written by a byte, word, or longword single-transfer cycle. If a burst transfer is used to read from or write to these registers, the first transfer completes successfully and the VSBchip2 asserts LTBI* on the local bus to indicate it cannot complete the rest of the request. Table 6-2 summarizes this register set.

Each register is defined by a table with five lines: an ADR/SIZ field, BIT field, NAME field, OPER field, and RESET field. The ADR/SIZ field defines the base address of the register and the number of bits defined in the table. The BIT field specifies the function's bit location in the register, and the NAME field is the name of the function. Unused bits have the word 'Reserved' in their NAME field. For these bits, writes have no effect and reads always return a zero. The OPER field specifies the allowed operations on that function. These operations are:

R	This bit is read only.
R/W	This bit is read and write.

R This bit is read only.

R/C This bit is read and clear only.

R/S This bit is read and set only.

The last field, RESET, specifies both the state the bit enters upon application of a reset, and by which reset signal(s) it is affected. The three reset states are 0, 1, or the letter `X' (not affected). The two reset signals are power-up reset (PURST*) signified by the letter 'P', or a local reset (LBRSTI*) signified by the letter 'L'.

Table 6-2. VSBchip2 Local Control and Status Registers Memory Map

Local Address	31 ... 24	23 ... 16	15 ... 8	7 ... 0
$FFF41000	Chip Control/Status Register		Local Interrupt Vector Base Register	
$FFF41004	Local Interrupt Status Register		Local Interrupt Enable Register	
$FFF41008	Local Interrupt Level Register			
$FFF4100C	Reserved			
$FFF41010	VSB Requester Control/Status Register			
$FFF41014	Timer Control Register		Clock Prescaler Register	
$FFF41018	Local Slave 1 Address Range Register (NOTES 1,2)			
$FFF4101C	Local Slave 1 Address Offset Register (NOTE 1)		Local Slave 1 Attribute Register (NOTE 1)	
$FFF41020	Local Slave 2 Address Range Register (NOTES 1,2)			
$FFF41024	Local Slave 2 Address Offset Register (NOTE 1)		Local Slave 2 Attribute Register (NOTE 1)	
$FFF41028	Local Slave 3 Address Range Register (NOTES 1,2)			
$FFF4102C	Local Slave 3 Address Offset Register (NOTE 1)		Local Slave 3 Attribute Register (NOTE 1)	
$FFF41030	Local Slave 4 Address Range Register (NOTES 1,2)			
$FFF41034	Local Slave 4 Address Offset Register (NOTE 1)		Local Slave 4 Attribute Register (NOTE 1)	
$FFF41038	Reserved			
$FFF4103C	Reserved			
$FFF41040	Reserved			
$FFF41044	Reserved			
$FFF41048	Reserved			
$FFF4104C	Reserved			
$FFF41050	Reserved			
$FFF41054	Reserved			
$FFF41058	Reserved			

Table 6-2. VSBchip2 Local Control and Status Registers Memory Map (Continued)

Local Address	31 ... 24	23 ... 16	15 ... 8	7 ... 0
$FFF4105C	Reserved			
$FFF41060	Reserved			
$FFF41064	Reserved			
$FFF41068	Reserved			
$FFF4106C	Reserved			
$FFF41070	Reserved			
$FFF41074	Local Error Address Register			
$FFF41078	Prescaler Current Count Register			
$FFF4107C	Prescaler Test Register			

NOTES:

1. Registers listed as "Slave" 1, 2, 3, or 4 in this memory map are listed as "Master" 1, 2, 3, or 4 in the **ENV** command parameters configurable by MVME166BUG (166Bug).

2. Registers listed as "Slave Address Range" in this memory map are listed as "Master Starting Address" and "Master Ending Address" in the **ENV** command parameters configurable by MVME166BUG (166Bug).

Chip Control/Status Register

ADR/SIZ	$FFF41000 (8 bits [6 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME	PURS	Reserved		VLED	VACTO	VTXT	VARTO	VBE
OPER	R/C	R		R/W	R/C	R/C	R/C	R/C
RESET	1 P	0	0	0 P	0 PL	0 PL	0 PL	0 PL

ADR/SIZ	$FFF41000 (8 bits [6 used] of 32)							
BIT	23	22	21	20	19	18	17	16
NAME	Reserved	VGA2	VGA1	VGA0	Reserved	SGA2	SGA1	SGA0
OPER	R	R	R	R	R	R/W	R/W	R/W
RESET	0	X	X	X	0	VGA2 P	VGA1 P	VGA0 P

PURS Power-up Reset Status. This status bit is set when the VSBchip2 undergoes a power-up reset (PURST* asserted). Writing a one clears this bit, and writing a zero does not have an effect.

VLED VSB LED Control. When this bit is cleared, the VSBLED* output pin is driven when either the VSB master has asserted VPAS* or the VSB slave is a responding or participating slave and has obtained control of the local bus. When this bit is set, the VSBLED* output pin is driven only in the latter case - when the VSBchip2 is the local bus master.

VACTO VSB Access Timer Time-out. This bit is set when the VSB Access Timer times out. Writing a one clears this bit, and writing a zero does not have an effect.

VTXTO VSB Transfer Timer Time-out. This bit is set when the VSB Transfer Timer times out. Writing a one clears this bit, and writing a zero does not have an effect.

VARTO VSB Arbitration Timer Time-out. This bit is set when the VSB Arbitration Timer times out. Writing a one clears this bit, and writing a zero does not have an effect.

VBE VSB Bus Error. This bit is set when the VSB master detects one of the following three conditions:

1. The VSB responding slave answers by asserting VERRI*.
2. A VSB responding slave is not found at the requested address when bounce mode is disabled.
3. A VSB responding slave does not continue to respond at the requested address after some part of the requested transfer has been completed.

Writing a one clears this bit, and writing a zero does not have an effect.

VGA2 - VGA0

VSB Geographical Address. These bits reflect the status of the VGA2 - VGA0 input pins. An address of %111 indicates that the board is not plugged into a VSB backplane.

SGA2 - SGA0 Programmable Geographical Address. Software can change the board's geographical address by programming these bits. When changing a board's geographical address, the burden is now on the system programmer to ensure that each board in the VSB subsystem is assigned a unique geographical address. On power-up, these bits revert to state of the VGA2 - VGA0 input pins.

Local Interrupt Vector Base Register

ADR/SIZ	$FFF41002 (8 bits [0 used] of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	Reserved							
OPER	R							
RESET	0	0	0	0	0	0	0	0

ADR/SIZ	$FFF41002 (8 bits [4 used] of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	LVEC7	LVEC6	LVEC5	LVEC4	Reserved			
OPER	R/W	R/W	R/W	R/W	R			
RESET	0 P	0 P	0 P	0 P	0	0	0	0

Each interrupt source provides a unique interrupt vector in response to a local bus interrupt acknowledge cycle. LVEC7 - LVEC4 comprise the upper four bits of the vector. LVEC7 is the most significant bit, and LVEC4 is the least. The lower four bits are unique for each interrupt source. These bits are encoded as shown below:

Local Interrupt Source	**LVEC3**	**LVEC2**	**LVEC1**	**LVEC0**	**Priority**
Local Write Post Error	0	0	0	0	Highest
VSB Write Post Error	0	0	0	1	
VSB	0	0	1	0	▲
Reserved	0	0	1	1	
EVSB Attention Interrupt	0	1	0	0	
VSB Interrupter Acknowledge	0	1	0	1	
Reserved	0	1	1	0	
Reserved	0	1	1	1	▼
Reserved	1	X	X	X	Lowest

If the VSBchip2 is programmed as the VSB interrupt handler, it attempts to perform a VSB interrupt-acknowledge cycle on the VSB to obtain the interrupt vector. If, however, the VSBchip2 is not programmed as the VSB interrupt handler, or the VSB device requesting the interrupt is not capable of providing a vector, the VSBchip2 returns the locally generated vector. Software is then responsible for polling the VSB interrupt requesters to determine which requester is to be serviced.

Local Interrupt Status Register

ADR/SIZ	$FFF41004 (8 bits [5 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME	Reserved	LWPIF	VWPIF	VSBIF	Reserved	ATTIF	VIAIF	Reserved
OPER	R	R/C	R/C	R	R	R/C	R/C	R
RESET	0	0 PL	0 PL	X	0	0 PL	0 PL	0

ADR/SIZ	$FFF41004 (8 bits [0 used] of 32)							
BIT	23	22	21	20	19	18	17	16
NAME	Reserved							
OPER	R							
RESET	0	0	0	0	0	0	0	0

Reading this register returns the status of each interrupt. When a bit is set, it signifies that a local bus interrupt is pending. If that interrupt is enabled through the Local Interrupt Enable Register, a hardware interrupt request is generated. If the interrupt is not enabled, its flag bit can be polled. Once an interrupt flag is set, it can only be cleared by PURST* or LBRSTI* being asserted, software writing a one to it, or a local bus IACK cycle servicing the interrupt.

LWPIF Local Write Post Error Interrupt Flag. This bit is set when an error is detected during completion of a write posted Local Bus cycle. When this flag is set, the Local Bus Error Address Register contains the address at which the write post error occurred. (Refer to this register later in this chapter.)

VWPIF VSB Write Post Error Interrupt Flag. This bit is set when an error is detected during completion of a write posted VSB cycle.

VSBIF VSB Interrupt Flag. This bit reflects the state of the VIRQI* pin.

ATTIF EVSB Attention Interrupt Flag. This bit reflects the state of the BCSR EVSB Attention Register ATTN bit.

VIAIF VSB Interrupt Acknowledge Complete Interrupt Flag. This bit is set only when the VSB Interrupt Status Register VSWIF bit is cleared indicating that the interrupt has been serviced.

Local Interrupt Enable Register

ADR/SIZ	$FFF41006 (8 bits [6 used] of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	GIE	LWPIE	VWPIE	VSBIE	Reserved	ATTIE	VIAIE	Reserved
OPER	R/W	R/W	R/W	R/W	R	R/W	R/W	R
RESET	0 PL	0 PL	0 PL	0 PL	0	0 PL	0 PL	0

6

ADR/SIZ	$FFF41006 (8 bits [0 used] of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	Reserved							
OPER	R							
RESET	0	0	0	0	0	0	0	0

This register is the local bus interrupt enable register. When an enable bit is set, the corresponding interrupt is enabled. When an enable bit is cleared, the corresponding interrupt is disabled. The enable does not clear the interrupt source. If necessary, interrupters should be cleared to remove any old interrupts before being enabled.

GIE Global Interrupt Enable. When this bit is cleared, the interrupts controlled by this register (Local Write Post Error, VSB Write Post Error, VSB Interrupt, EVSB Attention, and VSB Interrupt Acknowledge) are masked, regardless of the state of their individual enable bits. When this bit is set, the five interrupts are not masked and can be enabled by setting their enable bit.

LWPIE Local Write Post Error Interrupt Enable. When this bit and the GIE bit are set, whenever the LWPIF bit in the Local Interrupt Status Register is set, an interrupt is generated on the local bus by asserting its interrupt level programmed in the Local Interrupt Level Register on the output pins LIPLO2*-LIPLO0*.

VWPIE VSB Write Post Error Interrupt Enable. When this bit and the GIE bit are set, whenever the VWPIF bit in the Local Interrupt Status Register is set, an interrupt is generated on the local bus by asserting its interrupt level programmed in the Local Interrupt Level Register on the output pins LIPLO2*-LIPLO0*.

VSBIE VSB Interrupt Enable. When this bit and the GIE bit are set, whenever the VSBIF bit in the Local Interrupt Status Register is set, an interrupt is generated on the local bus by asserting its interrupt level programmed in the Local Interrupt Level Register on the output pins LIPLO2*-LIPLO0*.

ATTIE EVSB Attention Interrupt Enable. When this bit and the GIE bit are set, whenever the ATTIF bit in the Local Interrupt Status Register is set, an interrupt is generated on the local bus by asserting its interrupt level programmed in the Local Interrupt Level Register on the output pins LIPLO2*-LIPLO0*.

VIAIE VSB Interrupt Acknowledge Complete Interrupt Enable. When this bit and the GIE bit are set, whenever the VIAIF bit in the Local Interrupt Status Register is set, an interrupt is generated on the local bus by asserting its interrupt level programmed in the Local Interrupt Level Register on the output pins LIPLO2*-LIPLO0*.

Local Interrupt Level Register

ADR/SIZ	$FFF41008 (8 bits [3 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME	Reserved					LWPIL2	LWPIL1	LWPIL0
OPER	R/W	R/W	R/W	R/W	R	R/W	R/W	R
RESET	0 PL	0 PL	0 PL	0 PL	0	0 PL	0 PL	0

ADR/SIZ	$FFF41008 (8 bits [6 used] of 32)							
BIT	23	22	21	20	19	18	17	16
NAME	Reserved	VWPIL2	VWPIL1	VWPIL0	Reserved	VSBIL2	VSBIL1	VSBIL0
OPER	R	R/W	R/W	R/W	R	R/W	R/W	R/W
RESET	0	0 PL	0 PL	0 PL	0	0 PL	0 PL	0 PL

ADR/SIZ	$FFF41008 (8 bits [3 used] of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	Reserved					ATTIL2	ATTIL1	ATTIL0
OPER	R					R/W	R/W	R/W
RESET	0	0	0	0	0	0 PL	0 PL	0 PL

ADR/SIZ	$FFF41008 (8 bits [3 used] of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	Reserved	VIAIL2	VIAIL1	VIAIL0	Reserved			
OPER	R	R/W	R/W	R/W	R			
RESET	0	0 PL	0 PL	0 PL	0	0	0	0

These bits define the interrupt level driven onto the output pins LIPLO2*-LIPLO0* to request an interrupt on the local bus. Interrupt level bit 2 (suffix `IL2') is the most significant bit and interrupt level bit 0 (suffix `IL0') is the least significant. There are seven possible levels. Level 7 (%111) is the highest priority level and level 1 ($%001) is the lowest. Level 0 (%000) means the interrupt is disabled.

Note **The binary levels here are the complement of that driven onto LIPLO2*-LIPLO0*. A programmed level of %101 (five) translates to %010 (two) asserted on LIPLO2*-LIPLO0*.**

LWPIL2-LWPIL0

Local Bus Write Post Error Interrupt Level. These bits define the level of the Local Bus Write Post Error Interrupt.

VWPIL2-VWPIL0

VSB Write Post Error Interrupt Level. These bits define the level of the VSB Write Post Error Interrupt.

VSBIL2-VSBIL0

VSB Interrupt Level. These bits define the level of the VSB Interrupt.

ATTIL2-ATTIL0

EVSB Attention Interrupt Level. These bits define the level of the EVSB Attention Interrupt.

VIAIL2-VIAIL0

VSB Interrupt Acknowledge Complete Interrupt Level. These bits define the level of the VSB Interrupt Acknowledge Complete Interrupt.

Reserved Register

ADR/SIZ	$FFF4100C (32 bits [0 used])		
BIT	31	. . .	0
NAME	Reserved		
OPER	R		
RESET	$00000000		

This 32-bit register is currently undefined but is reserved for future VSBchip2 enhancements. Reading from this address always returns the value $00000000, and writing to this address does not have an effect.

VSB Requester Control/Status Register

ADR/SIZ	$FFF41010 (8 bits [5 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME	Reserved			DHB	DWB	PARMD	LVFAIR	LVRWD
OPER	R			R	R/W	R	R/W	R/W
RESET	0	0	0	0	0 PL	1 or 0 P	0 P	0 P

ADR/SIZ	$FFF41010 (8 bits [4 used] of 32)							
BIT	23	22	21	20	19	18	17	16
NAME	Reserved				VARBID3	VARBID2	VARBID1	VARBID0
OPER	R				R/W	R/W	R/W	R/W
RESET	0	0	0	0	0 P	0 P	0 P	0 P

ADR/SIZ	$FFF41010 (16 bits [0 used] of 32)		
BIT	15	...	0
NAME	Reserved		
OPER	R		
RESET	$0000		

This register controls the operation of the VSB serial requester, the VSB serial arbiter, and the VSB parallel requester.

DHB Device Has the Bus. Whenever the VSBchip2 has obtained VSB mastership in response to setting DWB, this status bit is set. It maintains DHB set as long as it has the bus. This bit applies whether the VSBchip2 requester is operating in either the serial or the parallel mode.

DWB Device Wants the Bus. When software sets this control bit, the chip's VSB requester tries to obtain VSB mastership, unless of course it already has it. Once VSB mastership is obtained, the DHB bit in this register is set. The requester maintains bus ownership, or keeps trying to acquire the VSB, as long as DWB remains set. When DWB is cleared, the requester relinquishes control of the VSB. Refer to the section on the VSB Requester and VSB Serial Arbiter (earlier in this chapter) for a discussion of how the VSBchip2 releases the bus when operating in serial or parallel arbitration mode.

PARMD Parallel Arbitration Mode. This status bit reflects the VSB arbitration mode selected on power-up. If the bit is set, the arbitration mode on the VSB is parallel. If the bit is cleared, the arbitration mode is serial. Refer to the section on VSB Geographical Addressing (earlier in this chapter) for the methodology used in selecting the VSB arbitration mode.

LVFAIR VSB Requester Fair Mode. When LVFAIR is set, the VSB Requester waits to assert its contribution to VBREQO* until it detects VBREQI* high for at least 1.5 LBCLK periods since it was last VSB master. When this bit is cleared, the requester does not wait. This bit is applicable only when the requester is operating in the serial arbitration mode.

LVRWD VSB Requester Release When Done. When this bit is set, the requester operates in the Release When Done (RWD) mode. When this bit is cleared, the requester operates in the Release On Request (ROR) mode. This bit is applicable only when the VSBchip2 is operating in the serial arbitration mode.

VARBID3 - VARBID0

VSB Requester Arbitration ID. These four bits are the upper four bits of the seven bit arbitration ID that the VSB requester places on the VSB when it contends for VSB mastership. The lower 3 bits of the arbitration ID are this board's geographical address. These bits apply only when the requester is operating in the parallel mode. VARBID3 is the most significant bit; and VARBID0 is the least. On power-up reset, these bits are cleared.

Timer Control Register

ADR/SIZ	$FFF41014 (8 bits [5 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME	Reserved			VARBTD	VATS1	VATS0	VTSX1	VTSX0
OPER	R			R/W	R/W	R/W	R/W	R/W
RESET	0	0	0	1 P	0 P	0 P	0 P	0 P

ADR/SIZ	$FFF41014 (8 bits [0 used] of 32)							
BIT	23	22	21	20	19	18	17	16
NAME	Reserved							
OPER	R							
RESET	0	0	0	0	0	0	0	0

VARBTD VSB Arbitration Timer Disable. When this bit is set, the VSB arbitration time-out timer is disabled; when cleared, the timer is enabled. When the timer is enabled and the arbiter does not receive VBUSYI* asserted within 256µs after a grant is issued, the arbiter removes the grant. The arbiter then re-arbitrates any pending requests. This bit is relevant only if the board is installed in slot 1 (VGA2 - VGA0=%000) of the VSB backplane, and only if the VSBchip2 is operating in serial arbitration mode. Alternately, VARBTD is relevant if the VSBchip2 is the active serial arbiter (SGA2 - SGA0=%000). It is recommended that this feature always be enabled in order to prevent lockups on the VSB.

VATS1 - VATS0

VSB Access Timer Select. These bits select the VSB access time-out value. When a transaction is headed to the VSB and the VSBchip2 is not the current VSB master, the access timer begins counting. If the VSBchip2 has not received bus mastership before the timer times out and the transaction is not write posted, the LTEA* signal is asserted on the local bus. If the transaction is write posted, a write post error interrupt is sent to the local bus interrupter instead. VATS1 - VATS0 are encoded as follows:

VATS1	VATS0	VSB Access Time-out
0	0	64 µs
0	1	1 ms
1	0	32 ms
1	1	Timer disabled

VTXS1 - VTXS0

VSB Transfer Timer Select. These bits select the VSB transfer time-out value. When the VSBchip2 asserts VPAS*, the timer begins timing. If the timer times out before a VACKI* is received, the VSBchip2 negates VPAS* and terminates the cycle. The transfer time-out timer is disabled when the VSBchip2 is not the current VSB master. VTXS1 - VTXS0 are encoded as follows:

VTXS1	VTXS0	VSB Transfer Time-out
0	0	8 µs
0	1	64 µs
1	0	256 µs
1	1	Timer disabled

Timer Clock Prescaler Register

<table>
<tr><td>ADR/SIZ</td><td colspan="8">$FFF41016 (8 bits [0 used] of 32)</td></tr>
<tr><td>BIT</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>NAME</td><td colspan="8">Reserved</td></tr>
<tr><td>OPER</td><td colspan="8">R</td></tr>
<tr><td>RESET</td><td colspan="8">0</td></tr>
</table>

<table>
<tr><td>ADR/SIZ</td><td colspan="8">$FFF41016 (8 bits of 32)</td></tr>
<tr><td>BIT</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>NAME</td><td colspan="8">Reserved</td></tr>
<tr><td>OPER</td><td colspan="8">R/W</td></tr>
<tr><td>RESET</td><td colspan="8">$DF P</td></tr>
</table>

The prescaler adjust provides the various clocks required by the timers and counters in the VSBchip2. In order to specify absolute times for these timers and counters, the prescaler value must be adjusted for different bus clocks. The prescaler register should be programmed based on the following equation:

Prescale Adjust = 256 - *LBCLK (MHz)*

Non-integer bus clocks introduce an error into the specified times for the various counters. The default value is $DF = 223 which assumes LBCLK is 33 MHz. If a 25 MHz clock is used, then this register needs to be programmed to $E7 = 231.

Local Bus Slave 1 Address Range Register

(called VSBC2 Master Ending Address #1 and VSBC2 Master Starting Address #1 in ENV command in 166Bug)

ADR/SIZ	$FFF41018 (16 bits of 32)		
BIT	31	. . .	16
NAME	Ending Address		
OPER	R/W		
RESET	$0000 P		

ADR/SIZ	$FFF41018 (16 bits of 32)		
BIT	15	. . .	0
NAME	Starting Address		
OPER	R/W		
RESET	$0000 P		

This register provides the address range for the first local bus to VSB map decoder. The ending address is in the first 16 bits and the starting address is in the second. Before this register can be programmed, the first local bus to VSB map decoder must be disabled by clearing the REN and WEN bits in the Local Bus Slave 1 Attribute Register.

Local Bus Slave 1 Address Offset Register

(called VSBC2 Master Address Offset #1 in **ENV** command in 166Bug)

ADR/SIZ	$FFF4101C (16 bits of 32)		
BIT	31	. . .	16
NAME	Offset Address		
OPER	R/W		
RESET	$0000 P		

This register is the address offset register for the first local bus to VSB map decoder. The contents of this register are added to the most significant bits of the local bus address received (LA31 - LA16). This sum is the address driven onto the VSB address lines VAD31 - VAD16. Before this register can be programmed, the first local bus to VSB map decoder must be disabled by clearing the REN and WEN bits in the Local Bus Slave 1 Attribute Register.

Local Bus Slave 1 Attribute Register

(called VSBC2 Master Attributes #1 in **ENV** command in 166Bug)

ADR/SIZ	$FFF4101E (8 bits [3 used] of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	REN	WEN	Reserved		WPE	Reserved		
OPER	R/W	R/W	R		R/W	R		
RESET	0 PL	0 PL	0	0	0 P	0	0	0

ADR/SIZ	$FFF4101E (8 bits [3 used] of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	BNCEN	Reserved	VSP1	VSP0	Reserved			
OPER	R/W	R	R/W	R/W	R			
RESET	0 P	0	1 P	1 P	0	0	0	0

REN Read Enable. When this bit is set, the first local bus to VSB map decoder is enabled for read cycles.

WEN Write Enable. When this bit is set, the first local bus to VSB map decoder is enabled for write cycles.

WPE Write Post Enable. When this bit is high, write posting is enabled for the address segment defined by the Local Bus Slave 1 Address Range Register.

BNCEN Bounce Mode Enable. If this bit is set, whenever the VSBchip2 performs a VSB address broadcast in which no slave responds, it asserts the BOUNCE output pin for one LBCLK, terminates the VSB cycle, and waits for an alternate device to terminate the local bus cycle. When bounce is disabled, if there is no response from a VSB slave, the VSBchip2 terminates the local bus transfer by asserting LTEA*.

VSP1 - VSP0 VSB Space Codes. These bits control the space codes asserted by the VSBchip2 when functioning as the VSB master in the address range defined by the Local Bus Slave 1 Address Range Register. VSP1 and VSP0 are encoded as follows:

Address Space	**VSP1**	**VSP0**
Reserved - System Address Space Selected	0	0
Alternate Address Space	0	1
I/O Address Space	1	0
System Address Space	1	1

Local Bus Slave 2 Address Range Register

(called VSBC2 Master Ending Address #2 and VSBC2 Master Starting Address #2 in **ENV** command in 166Bug)

ADR/SIZ	$FFF41020 (16 bits of 32)		
BIT	31	. . .	16
NAME	Ending Address		
OPER	R/W		
RESET	$0000 P		

ADR/SIZ	$FFF41020 (16 bits of 32)		
BIT	15	. . .	0
NAME	Starting Address		
OPER	R/W		
RESET	$0000 P		

This register provides the address range for the second local bus to VSB map decoder. The ending address is in the first 16 bits and the starting address is in the second. Before this register can be programmed, the second local bus to VSB map decoder must be disabled by clearing the REN and WEN bits in the Local Bus Slave 2 Attribute Register.

Local Bus Slave 2 Address Offset Register

(called VSBC2 Master Address Offset #2 in **ENV** command in 166Bug)

ADR/SIZ	$FFF41024 (16 bits of 32)		
BIT	31	. . .	16
NAME	Offset Address		
OPER	R/W		
RESET	$0000 P		

This register is the address offset register for the second local bus to VSB map decoder. The contents of this register are added to the most significant bits of the local bus address received (LA31 - LA16). This sum is the address driven onto the VSB address lines VAD31 - VAD16. Before this register can be programmed, the second local bus to VSB map decoder must be disabled by clearing the REN and WEN bits in the Local Bus Slave 2 Attribute Register.

Local Bus Slave 2 Attribute Register

(called VSBC2 Master Attributes #2 in **ENV** command in 166Bug)

ADR/SIZ	$FFF41026 (8 bits [3 used] of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	REN	WEN	Reserved		WPE	Reserved		
OPER	R/W	R/W	R		R/W	R		
RESET	0 PL	0 PL	0	0	0 P	0	0	0

ADR/SIZ	$FFF41026 (8 bits [3 used] of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	BNCEN	Reserved	VSP1	VSP0	Reserved			
OPER	R/W	R	R/W	R/W	R			
RESET	0 P	0	1 P	1 P	0	0	0	0

REN Read Enable. When this bit is set, the second local bus to VSB map decoder is enabled for read cycles.

WEN Write Enable. When this bit is set, the second local bus to VSB map decoder is enabled for write cycles.

WPE Write Post Enable. When this bit is high, write posting is enabled for the address segment defined by the Local Bus Slave 2 Address Range Register.

BNCEN Bounce Mode Enable. If this bit is set, whenever the VSBchip2 performs a VSB address broadcast in which no slave responds, it asserts the BOUNCE output pin for one LBCLK, terminates the VSB cycle, and waits for an alternate device to terminate the local bus cycle. When bounce is disabled, if there is no response from a VSB slave, the VSBchip2 terminates the local bus transfer by asserting LTEA*.

VSP1 - VSP0 VSB Space Codes. These bits control the space codes asserted by the VSBchip2 when functioning as the VSB master in the address range defined by the Local Bus Slave 2 Address Range Register. VSP1 and VSP0 are encoded as follows:

Address Space	VSP1	VSP0
Reserved - System Address Space Selected	0	0
Alternate Address Space	0	1
I/O Address Space	1	0
System Address Space	1	1

Local Bus Slave 3 Address Range Register

(called VSBC2 Master Ending Address #3 and VSBC2 Master Starting Address #3 in **ENV** command in 166Bug)

ADR/SIZ	$FFF41028 (16 bits of 32)		
BIT	31	. . .	16
NAME	Ending Address		
OPER	R/W		
RESET	$0000 P		

ADR/SIZ	$FFF41028 (16 bits of 32)		
BIT	15	. . .	0
NAME	Starting Address		
OPER	R/W		
RESET	$0000 P		

This register provides the address range for the third local bus to VSB map decoder. The ending address is in the first 16 bits and the starting address is in the second. Before this register can be programmed, the third local bus to VSB map decoder must be disabled by clearing the REN and WEN bits in the Local Bus Slave 3 Attribute Register.

Local Bus Slave 3 Address Offset Register

(called VSBC2 Master Address Offset #3 in **ENV** command in 166Bug)

ADR/SIZ	$FFF4102C (16 bits of 32)		
BIT	31	. . .	16
NAME	Offset Address		
OPER	R/W		
RESET	$0000 P		

This register is the address offset register for the third local bus to VSB map decoder. The contents of this register are added to the most significant bits of the local bus address received (LA31 - LA16). This sum is the address driven onto the VSB address lines VAD31 - VAD16. Before this register can be programmed, the third local bus to VSB map decoder must be disabled by clearing the REN and WEN bits in the Local Bus Slave 3 Attribute Register.

Local Bus Slave 3 Attribute Register

(called VSBC2 Master Attributes #3 in **ENV** command in 166Bug)

ADR/SIZ	$FFF4102E (8 bits [3 used] of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	REN	WEN	Reserved		WPE	Reserved		
OPER	R/W	R/W	R		R/W	R		
RESET	0 PL	0 PL	0	0	0 P	0	0	0

ADR/SIZ	$FFF4102E (8 bits [3 used] of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	BNCEN	Reserved	VSP1	VSP0	Reserved			
OPER	R/W	R	R/W	R/W	R			
RESET	0 P	0	1 P	1 P	0	0	0	0

REN Read Enable. When this bit is set, the third local bus to VSB map decoder is enabled for read cycles.

WEN Write Enable. When this bit is set, the third local bus to VSB map decoder is enabled for write cycles.

WPE Write Post Enable. When this bit is high, write posting is enabled for the address segment defined by the Local Bus Slave 3 Address Range Register.

BNCEN Bounce Mode Enable. If this bit is set, whenever the VSBchip2 performs a VSB address broadcast in which no slave responds, it asserts the BOUNCE output pin for one LBCLK, terminates the VSB cycle, and waits for an alternate device to terminate the local bus cycle. When bounce is disabled, if there is no response from a VSB slave, the VSBchip2 terminates the local bus transfer by asserting LTEA*.

VSP1 - VSP0 VSB Space Codes. These bits control the space codes asserted by the VSBchip2 when functioning as the VSB master in the address range defined by the Local Bus Slave 3 Address Range Register. VSP1 and VSP0 are encoded as follows:

Address Space	VSP1	VSP0
Reserved - System Address Space Selected	0	0
Alternate Address Space	0	1
I/O Address Space	1	0
System Address Space	1	1

Local Bus Slave 4 Address Range Register

(called VSBC2 Master Ending Address #4 and VSBC2 Master Starting Address #4 in ENV command in 166Bug)

ADR/SIZ	$FFF41030 (16 bits of 32)		
BIT	31	. . .	16
NAME	Ending Address		
OPER	R/W		
RESET	$0000 P		

ADR/SIZ	$FFF41030 (16 bits of 32)		
BIT	15	. . .	0
NAME	Starting Address		
OPER	R/W		
RESET	$0000 P		

This register provides the address range for the fourth local bus to VSB map decoder. The ending address is in the first 16 bits and the starting address is in the second. Before this register can be programmed, the fourth local bus to VSB map decoder must be disabled by clearing the REN and WEN bits in the Local Bus Slave 4 Attribute Register.

Local Bus Slave 4 Address Offset Register

(called VSBC2 Master Address Offset #4 in **ENV** command in 166Bug)

ADR/SIZ	$FFF41034 (16 bits of 32)		
BIT	31	. . .	16
NAME	Offset Address		
OPER	R/W		
RESET	$0000 P		

This register is the address offset register for the fourth local bus to VSB map decoder. The contents of this register are added to the most significant bits of the local bus address received (LA31 - LA16). This sum is the address driven onto the VSB address lines VAD31 - VAD16. Before this register can be programmed, the fourth local bus to VSB map decoder must be disabled by clearing the REN and WEN bits in the Local Bus Slave 4 Attribute Register.

Local Bus Slave 4 Attribute Register

(called VSBC2 Master Attributes #4 in **ENV** command in 166Bug)

ADR/SIZ	$FFF41036 (8 bits [3 used] of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	REN	WEN	Reserved		WPE	Reserved		
OPER	R/W	R/W	R		R/W	R		
RESET	0 PL	0 PL	0	0	0 P	0	0	0

ADR/SIZ	$FFF41036 (8 bits [3 used] of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	BNCEN	Reserved	VSP1	VSP0	Reserved			
OPER	R/W	R	R/W	R/W	R			
RESET	0 P	0	1 P	1 P	0	0	0	0

REN Read Enable. When this bit is set, the fourth local bus to VSB map decoder is enabled for read cycles.

WEN Write Enable. When this bit is set, the fourth local bus to VSB map decoder is enabled for write cycles.

WPE Write Post Enable. When this bit is high, write posting is enabled for the address segment defined by the Local Bus Slave 4 Address Range Register.

BNCEN Bounce Mode Enable. If this bit is set, whenever the VSBchip2 performs a VSB address broadcast in which no slave responds, it asserts the BOUNCE output pin for one LBCLK, terminates the VSB cycle, and waits for an alternate device to terminate the local bus cycle. When bounce is disabled, if there is no response from a VSB slave, the VSBchip2 terminates the local bus transfer by asserting LTEA*.

VSP1 - VSP0 VSB Space Codes. These bits control the space codes asserted by the VSBchip2 when functioning as the VSB master in the address range defined by the Local Bus Slave 4 Address Range Register. VSP1 and VSP0 are encoded as follows:

Address Space	VSP1	VSP0
Reserved - System Address Space Selected	0	0
Alternate Address Space	0	1
I/O Address Space	1	0
System Address Space	1	1

Reserved Registers

ADR/SIZ	$FFF41038 through $FFF41070 (32 bits [0 used] each)		
BIT	31	. . .	0
NAME	Reserved		
OPER	R		
RESET	$00000000		

These 32-bit registers are currently undefined but are reserved for future VSBchip2 enhancements. Reading from this area always returns the value $00000000 and writing to this area has no effect.

Local Error Address Register

ADR/SIZ	$FFF41074 (32 bits)		
BIT	31	. . .	0
NAME	Error Address		
OPER	R		
RESET	$00000000 P		

If the LWPIF bit in the Local Interrupt Status Register is set, then this register contains the address stored in the local bus write post buffer at the time the last write post error was detected. This register does not change until the next Local Bus Write Post Error is detected.

Prescaler Current Count Register

ADR/SIZ	$FFF41078 (8 bits [0 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME	Reserved							
OPER	R							
RESET	$00							

ADR/SIZ	$FFF41078 (24 bits of 32)		
BIT	23	. . .	0
NAME	Prescaler Count		
OPER	R		
RESET	$000000 P		

Access to the prescaler is provided to verify the counter is operational. The VSBchip2 has a 24-bit prescaler that provides the clocks required by the various timers in the chip. The lower 8 bits of the prescaler counter increment to $FF at the bus clock rate (LBCLK) and then they are loaded from the Timer Prescaler Register. When the load occurs, the upper 16 bits are incremented. When the Timer Prescaler Register is correctly programmed, the lower 8 bits increment at the bus clock rate and the upper 16 bits increment every microsecond. The prescaler count register may be read at any time.

Prescaler Test Register

ADR/SIZ	$FFF4107C (8 bits [1 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME	TESTEN	Reserved						
OPER	R/W	R						
RESET	0 P	0	0	0	0	0	0	0

ADR/SIZ	$FFF4107C (8 bits of 32)							
BIT	23	22	21	20	19	18	17	16
NAME	CNTR63	CNTR62	CNTR61	CNTR60	CNTR53	CNTR52	CNTR51	CNTR50
OPER	R/W				R/W			
RESET	0 P	0 P	0 P	0 P	0 P	0 P	0 P	0 P

ADR/SIZ	$FFF4107C (8 bits of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	CNTR43	CNTR42	CNTR41	CNTR40	CNTR33	CNTR32	CNTR31	CNTR30
OPER	R/W				R/W			
RESET	0 P	0 P	0 P	0 P	0 P	0 P	0 P	0 P

ADR/SIZ	$FFF4107C (8 bits of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	CNTR23	CNTR22	CNTR21	CNTR20	CNTR13	CNTR12	CNTR11	CNTR10
OPER	R/W				R/W			
RESET	0 P	0 P	0 P	0 P	0 P	0 P	0 P	0 P

TESTEN Prescaler Test Mode Enable. Setting this bit places the prescaler in test mode. The 24-bit counter is broken into six separate 4-bit binary counters. The value written into this register is then loaded into the six counters. On each LBCLK, the six counters increment. This enables software to quickly insure that segment of the prescaler is operational without waiting 2^{24} clocks.

CNTR13-CNTR10
Counter 1.

CNTR23-CNTR20
Counter 2.

CNTR33-CNTR30
Counter 3.

CNTR43-CNTR40
Counter 4.

CNTR53-CNTR50
Counter 5.

CNTR63-CNTR60
Counter 6.

Board Control and Status Registers Programming Model

This section details the Board Control and Status Registers (BCSRs). These sixteen registers are accessible from both the local bus and the VSB. Each register can be read from or written to by a byte, word, triple-byte (VSB only), or longword transfer cycle. VSB block transfers are not supported when accessing these registers. If a burst transfer is used to read from or write to these registers, the first transfer completes successfully and the VSBchip2 asserts LTBI* on the local bus to indicate it cannot complete the rest of the request. There are no restrictions as to when these registers may be accessed; they may be read from or written to at any time.

The BCSRs are fully compliant with the Extensible VME Subsystem Bus Proposal published April 24, 1990.

Table 6-3 shows the memory map of the BCSRs. All registers are accessible through VSB System Address Space.

Table 6-3. VSBchip2 Board Control and Status Registers Memory Map

VSB Address	Local Address	31 ... 24	23 ... 16	15 ... 8	7 ... 0
$E0n00000	$FFF41100	EVSB Attention Register			
$E0n00004	$FFF41104	EVSB Test-And-Set (TAS) Register			
$E0n00008	$FFF41108	General Purpose Register 1			
$E0n0000C	$FFF4110C	General Purpose Register 2			
$E0nFFFE0	$FFF41110	VSB Error Status Register			
$E0nFFFE4	$FFF41114	VSB Interrupt Control Register	VSB Interrupt Vector Register	VSB Interrupt Enable Register	VSB Interrupt Status Register
$E0nFFFE8	$FFF41118	VSB Slave 1 Address Range Register			
$E0nFFFEC	$FFF4111C	VSB Slave 1 Address Offset Register		VSB Slave 1 Attribute Register	
$E0nFFFF0	$FFF41120	VSB Slave 2 Address Range Register			
$E0nFFFF4	$FFF41124	VSB Slave 2 Address Offset Register		VSB Slave 2 Attribute Register	
$E0nFFFF8	$FFF41128	Reserved			
$E0nFFFFC	$FFF4112C	VSB Error Address Register			

NOTE: n = value in SGA2 - SGA0 (Chip Control/Status Register).

Each register is defined by a table with six lines: an ADR/SIZ field, a BIT field, a NAME field, a LOPER field, a VOPER field, and a RESET field. The ADR/SIZ field defines the base addresses of the register and the number of bits defined in the table. The BIT field specifies the function's bit location in the register, and the NAME field is the name of the function. Unused bits are designated 'Reserved' in their NAME field. For these bits, writes have no effect and reads always return a zero. The LOPER field specifies the operations allowed on that bit from the local bus. The VOPER field specifies the operations allowed on that bit from the VSB.

These operations are:

R	This bit is read only.
R/W	This bit is read and write.
R/C	This bit is read and clear only.
R/S	This bit is read and set only.

The last field, RESET, specifies the state the bit enters upon application of a reset, and by which reset signal(s) it is affected. The three reset states are 0, 1, or `X' (not affected). The two reset signals are power-up reset (PURST*) signified by the letter `P', and local reset (LBRSTI*) signified by the letter `L'.

EVSB Attention Register

ADR/SIZ	$E0n00000/$FFF41100 (8 bits [5 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME	READY	RESET	ATTN	ERR	IRQ	Reserved		
LOPER	R/W	R	R	R	R	R		
VOPER	R	R/S	R/S	R	R	R		
RESET	0 PL	0 P	0 PL	0 PL	0 PL	0	0	0

ADR/SIZ	$E0n00000/$FFF41100 (8 bits [0 used] of 32)		
BIT	23	. . .	16
NAME	Reserved		
LOPER	R		
VOPER	R		
RESET	$00		

ADR/SIZ	$E0n00000/$FFF41100 (8 bits of 32)		
BIT	15	. . .	8
NAME	VSBchip2 Version		
LOPER	R		
VOPER	R		
RESET	$01		

ADR/SIZ	$E0n00000/$FFF41100 (8 bits of 32)		
BIT	7	. . .	0
NAME	VSBchip2 ID		
LOPER	R		
VOPER	R		
RESET	$11		

READY Device Ready. This bit is set by a local bus device to inform all VSB devices that it has completed initialization of all local bus resources. The contents of all EVSB Registers should be considered invalid until this bit is set.

RESET Software Reset.
Not used on the MVME166.

ATTN Local Interrupt Request. This bit is set by a VSB device to force a local bus interrupt (provided this interrupt has been enabled in the Local Interrupt Enable Register). From the local bus, this bit reflects the status of the ATTNIF bit in the Local Interrupt Status Register. Writing a one to this bit from the VSB sets it; writing a zero does not have an effect. This bit is cleared when the ATTNIF bit in the Local Interrupt Status Register is cleared.

ERR VSB Error. This status bit is set when any of the error bits in the VSB Error Status Register are set. ERR remains set until all the error bits in the VSB Error Status Register are cleared.

IRQ VSB Interrupt Request. This bit is set when either the VSWIF or the VWPIF bits in the VSB Interrupt Status Register are set. IRQ remains set until both the VSWIF and VWPIF bits in the VSB Interrupt Status Register are cleared.

VSBchip2 Version
VSBchip2 Version Number. These eight bits are the VSBchip2 version number. This field is incremented each time a mask change is made to the device. The initial mask is version $01. The next mask will be version $02.

VSBchip2 ID VSBchip2 Identification Number. These eight bits are the VSBchip2 unique part number. This field is always $11.

EVSB Test and Set (TAS) Register

ADR/SIZ	$E0n00004/$FFF41104 (8 bits [1 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME	TAS	Reserved						
LOPER	R/W	R						
VOPER	R/W	R						
RESET	0 PL	0	0	0	0	0	0	0

ADR/SIZ	$E0n00004/$FFF41104 (24 bits [0 used] of 32)		
BIT	23	. . .	0
NAME	Reserved		
LOPER	R		
VOPER	R		
RESET	$000000		

This register contains a single bit used by software to lock resources during access by multiple VSB and/or local bus devices. TAS is set at the end of any read to this register, or it can be written by software to a one or zero. When accessed with a locked test-and-set instruction, TAS can be used as a semaphore among competing devices. For example, if two VSB devices read this register successively, and the bit was originally a zero, the first reads a zero, and the second reads a one. This register does not actually interlock any resource in hardware. Software must be written to check this bit before accessing any shared resources.

General Purpose Register 1

ADR/SIZ	$E0n00008/$FFF41108 (32 bits)		
BIT	31	. . .	0
NAME	User Defined		
LOPER	R/W		
VOPER	R/W		
RESET	$00000000 P		

This register is a general purpose register that allows VSB and local bus devices to share some information about a resource. The function of this register is not defined by the hardware specification. It may be used as a message mailbox in conjunction with the Test and Set Register previously described.

General Purpose Register 2

ADR/SIZ	$E0n0000C/$FFF4110C (32 bits)		
BIT	31	. . .	0
NAME	User Defined		
LOPER	R/W		
VOPER	R/W		
RESET	$00000000 P		

This register is a general purpose register that allows VSB and local bus devices to share some information about a resource. The function of this register is not defined by the hardware specification. It may be used as a message mailbox in conjunction with the Test and Set Register previously described.

VSB Error Status Register

ADR/SIZ	$E0nFFFE0/$FFF41110 (8 bits [4 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME	Reserved				LBTE	LBPE	LBXE	LBE
LOPER	R				R	R	R	R
VOPER	R				R/C	R/C	R/C	R/C
RESET	0	0	0	0	0 PL	0 PL	0 PL	0 PL

ADR/SIZ	$E0nFFFE0/$FFF41110 (24 bits [0 used] of 32)		
BIT	23	. . .	0
NAME	Reserved		
LOPER	R		
VOPER	R		
RESET	$000000		

This status register is updated only when the VSBchip2 is functioning as the local bus master and receives a local bus error (LTEA* asserted and LTA* negated) in response to a transfer cycle. This register records the decoded state of the LST1-LST0 input/output status pins; therefore, only one bit can be set. Until this register is cleared, it contains the cause of the last bus error received by the VSBchip2. The contents of the register can be cleared by asserting PURST* or LBRSTI*, or by a VSB device writing a one to the set bit. Writing a zero does not have an effect.

LBTE Local Bus Time-out Error. This bit is set when the status lines indicate a local bus time-out (LST1 - LST0 = %00).

LBPE Local Bus RAM Parity Error. This bit is set when the status lines indicate a RAM parity error (LST1 - LST0 = %10).

LBXE Local Bus External Error. This bit is set when the status lines indicate an external bus error (LST1 - LST0 = %01).

LBE Local Bus Error. This bit is set when the status lines indicate an error of unknown origin (LST1 - LST0 = %11).

VSB Interrupt Control Register

ADR/SIZ	$E0nFFFE4/$FFF41114 (8 bits [7 used] of 32)							
BIT	31	30	29	28	27	26	25	24
NAME	Reserved	VEN	VIFAIR	IHV	VINTID3	VINTID2	VINTID1	VINTID0
LOPER	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
VOPER	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL

VEN VSB INTV Capability Enable. When this bit is set, the VSBchip2 can function both as a VSB INTV (Interrupt Vector) slave participating in VSB interrupt-acknowledge cycles, and as a VSB INTP (Interrupt Poll) slave having its interrupts serviced by polling. When VEN is cleared, the VSBchip2 ignores VSB interrupt-acknowledge cycles and functions only as a VSB INTP slave.

VIFAIR VSB Interrupter FAIR Mode. When this bit is set, the interrupter operates in the fairness mode: the VSBchip2 does not reassert VIRQO* until VIRQI* has been negated for a minimum of 1.5 LBCLKs. This fair mode enables lower priority interrupting devices the opportunity to have their interrupts serviced. When VIFAIR is cleared, the VSBchip2 can assert VIRQO* as soon as it detects an interrupt condition.

IHV VSB Interrupt Handler Enable. When this bit is set, the VSBchip2 will act as the VSB Interrupt Handler. It generates a VSB interrupt-acknowledge cycle in response to a local bus interrupt-acknowledge cycle which is servicing the VSB interrupt.

VINTID3 - VINTID0

VSB Interrupt Arbitration ID. These four bits are the upper four bits of the seven bit arbitration ID that the VSBchip2 places on the VSB during the arbitration portion of a VSB interrupt-acknowledge cycle, provided the VEN bit is set and the VSBchip2 has a VSB interrupt pending (VIRQO* asserted). VINTID3 is the most-significant bit; and VINTID0, the least-significant.

VSB Interrupt Vector Register

ADR/SIZ	$E0nFFFE5/$FFF41115 (8 bits of 32)							
BIT	23	22	21	20	19	18	17	16
NAME	VIVEC7	VIVEC6	VIVEC5	VIVEC4	VIVEC3	VIVEC2	VIVEC1	VIVEC0
LOPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
VOPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
RESET	0 PL	0 PL	0 PL	0 PL	1 PL	1 PL	1 PL	0 PL

If the VSBchip2 wins interrupt arbitration, it passes this vector back to the VSB master during the Status/ID transfer phase of the interrupt-acknowledge cycle. The upper seven bits of this vector, VIVEC7-VIVEC1, are software selectable to any value. VIVEC0, the lowest order bit of this vector, identifies the interrupting source. If VIVEC0 is cleared, a VSB Write Post Error Interrupt is being serviced; and if VIVEC1 is set, a Software interrupt is being serviced.

VSB Interrupt Enable Register

ADR/SIZ	$E0nFFFE6/$FFF41116 (8 bits [3 used] of 32)						
BIT	15	14	13	12	11	10	9	8
NAME	VGIE	Reserved					VSWIE	VWPIE
LOPER	R/W	R					R/W	R/W
VOPER	R/W	R					R/W	R/W
RESET	0 PL	0	0	0	0	0	0 PL	0 PL

VGIE VSB Global Interrupt Enable. This bit is set to enable all VSB interrupts. Clearing VGIE disables all interrupts regardless of the state of the individual interrupt enable bits.

VSWIE VSB Software Interrupt Enable. When this bit and the VGIE bit are set, setting the VSWIF bit in the VSB Interrupt Status Register generates an interrupt on the VSB (VIRQO* is asserted).

VWPIE VSB Write Post Interrupt Enable. When this bit and the VGIE bit are set, an interrupt is generated on VSB (the VSBchip2 asserts VIRQO*) each time an error is detected during completion of a write posted VSB cycle.

VSB Interrupt Status Register

ADR/SIZ	$E0nFFFE7/$FFF41117 (8 bits [2 used] of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	Reserved						VSWIF	VWPIF
LOPER	R						R/S	R
VOPER	R						R/C	R/C
RESET	0	0	0	0	0	0	0 PL	0 PL

VSWIF VSB Software Interrupt Flag. Software writing a one to this bit generates a VSB interrupt. The IRQ bit in the EVSB Attention Register is set, and if the VGIE and VSWIE bits in the VSB Interrupt Enable Register have been set, the VSBchip2 asserts VIRQO*. Once VSWIF is set via the local bus it can only be cleared by PURST* or LBRSTI* being asserted, a VSB device writing a one to it, or the interrupt being serviced by a VSB interrupt-acknowledge cycle.

VWPIF VSB Write Post Interrupt Flag. This bit is only set when an error is detected during completion of a write posted VSB cycle. VWPIF set, generates a VSB interrupt. Therefore, the IRQ bit in the EVSB Attention Register is also set, and if the VGIE and VWPIE bits in the VSB Interrupt Enable Register have been set, the VSBchip2 asserts VIRQO*. Once VWPIF is set, it can only be cleared by PURST* or LBRSTI* being asserted, a VSB device writing a one to it, or the interrupt being serviced by a VSB interrupt-acknowledge cycle.

Note **The VSB address at which the write post error occurred is stored in the VSB Error Address Register. Refer to its description later in this chapter.**

VSB Slave 1 Address Range Register

ADR/SIZ	$E0nFFFE8/$FFF41118 (16 bits of 32)		
BIT	31	. . .	16
NAME	Ending Address		
LOPER	R/W		
VOPER	R/W		
RESET	$0000 P		

ADR/SIZ	$E0nFFFE8/$FFF41118 (16 bits of 32)		
BIT	15	. . .	0
NAME	Starting Address		
LOPER	R/W		
VOPER	R/W		
RESET	$0000 P		

This register provides the address range for the first VSB to local bus map decoder. The ending address is in the first 16 bits and the starting address is in the second.

VSB Slave 1 Address Offset Register

ADR/SIZ	$E0nFFFEC/$FFF4111C (16 bits of 32)		
BIT	31	. . .	16
NAME	Offset Address		
LOPER	R/W		
VOPER	R/W		
RESET	$0000 P		

This register is the address offset register for the first VSB to local bus map decoder. The contents of this register are added to the most significant bits of the VSB address received (VAD31 - VAD16). This sum is then the address driven onto the local bus address lines LA31 - LA16.

VSB Slave 1 Attribute Register

ADR/SIZ	$E0nFFFEE/$FFF4111E (8 bits of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	REN	WEN	POR	POW	WPE	SAS	ALTAS	IOAS
LOPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
VOPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL	0 PL

ADR/SIZ	$E0nFFFEE/$FFF4111E (8 bits [5 used] of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	Reserved	LOCK	Reserved		LBTS1	LBTS0	LBSC1	LBSC0
LOPER	R	R/W	R		R/W	R/W	R/W	R/W
VOPER	R	R/W	R		R/W	R/W	R/W	R/W
RESET	0	0 PL	0	0	0 PL	0 PL	0 PL	0 PL

REN Read Enable. When this bit is set, read access to the address range programmed in the VSB Slave 1 Address Range Register is allowed and the VSBchip2 either responds to or participates in read cycles depending upon the state of the POR bit.

WEN Write Enable. When this bit is set, write access to the address range programmed in the VSB Slave 1 Address Range Register is allowed and the VSBchip2 either responds to or participates in write cycles depending upon the state of the POW bit.

POR Participate on Read. When the REN bit is cleared this bit is not relevant. However, when the REN bit is set, this bit defines whether the VSBchip2 is the responding slave or a participating slave in read cycles to the address range programmed in the VSB Slave 1 Address Range Register. When this bit is set, the VSBchip2 is a participator; and when this bit is cleared, the responder. The default state of this bit is cleared - the VSBchip2 is a responding slave.

POW Participate on Write. When the WEN bit is cleared this bit is not relevant. However, when the WEN bit is set, this bit defines whether the VSBchip2 is the responding slave or a participating slave in write cycles to the address range programmed in the VSB Slave 1 Address Range Register. When this bit is set, the VSBchip2 is a participator; and when this bit is cleared, the responder. The default state of this bit is cleared - the VSBchip2 is a responding slave.

WPE Write Post Enable. When this bit is set, write posting is enabled for the address range defined by the VSB Slave 1 Address Range Register.

SAS System Address Space. When access to the local bus is permitted (responding or participating capability must be enabled), setting this bit defines the VSB Slave 1 Address Range to be in the VSB System Address Space. This is the default location.

ALTAS Alternate Address Space. When access to the local bus is permitted (responding or participating capability must be enabled), setting this bit defines the VSB Slave 1 Address Range to be in the VSB Alternate Address Space.

IOAS I/O Address Space. When access to the local bus is permitted (responding or participating capability must be enabled), setting this bit defines the VSB Slave 1 Address Range to be in the VSB I/O Address Space.

LOCK

Lock Local Bus on Block Transfers. The Lock bit, if set, causes the VSBchip2 local master to assert LBB* on the start of a VSB block transfer. This effectively prevents any other local bus master from taking the bus back, and allows higher speed block transfers. Only when the VSB master removes VPAS*, is the bus released.

This software Lock contrasts with the VSB lock signal (VLOCK*) which may be used to keep the bus locked between VSB transfers as well as within VSB transfers. Using VLOCK* allows an external VSB master to perform read-modify-write cycles, for example, which is not possible with this software Lock bit.

LBTS1 - LBTS0

Local Bus Transfer Size. These bits define the port size of the VSB Slave 1 address range to be 8-bits, 16-bits, or 32-bits. The port size programmed is reflected by the value of VASACK1* - VASACK0* driven onto the VSB, and by LSIZ1 and LSIZ0 driven onto the local bus. Refer to the table in the section on Local Bus Master Interface at the beginning of this chapter for this encoding. LBTS1 and LBTS0 are encoded as follows:

LBTS1	LBTS0	Port Size
0	0	32-bits
0	1	16-bits
1	0	8-bits
1	1	No Responder

LBSC1 - LBSC0

Local Bus Snoop Control. These bits control the snoop enable lines to the local bus for the address range defined by the VSB Slave 1 Address Range Register. LBSC1 and LBSC0 are encoded as follows:

LBSC1	LBSC0	Snoop Function
0	0	Snoop Inhibited
0	1	Write - Sink data Read - Supply dirty data and leave dirty
1	0	Write - Invalidate Read - Supply dirty data and mark invalid
1	1	Snoop Inhibited

VSB Slave 2 Address Range Register

ADR/SIZ	$E0nFFFF0/$FFF41120 (16 bits of 32)		
BIT	31	. . .	16
NAME	Ending Address		
LOPER	R/W		
VOPER	R/W		
RESET	$0000 P		

ADR/SIZ	$E0nFFFF0/$FFF41120 (16 bits of 32)		
BIT	15	. . .	0
NAME	Starting Address		
LOPER	R/W		
VOPER	R/W		
RESET	$0000 P		

This register provides the address range for the second VSB to local bus map decoder. The ending address is in the first 16 bits and the starting address is in the second.

VSB Slave 2 Address Offset Register

ADR/SIZ	$E0nFFFF4/$FFF41124 (16 bits of 32)		
BIT	31	. . .	16
NAME	Offset Address		
LOPER	R/W		
VOPER	R/W		
RESET	$0000 P		

This register is the address offset register for the second VSB to local bus map decoder. The contents of this register are added to the most significant bits of the VSB address received (VAD31 - VAD16). This sum is then the address driven onto the local bus address lines LA31 - LA16.

VSB Slave 2 Attribute Register

ADR/SIZ	$E0nFFFF6/$FFF41126 (8 bits of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	REN	WEN	POR	POW	WPE	SAS	ALTAS	IOAS
LOPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
VOPER	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RESET	0 PL	0 PL	0 PL	0 PL	0 PL	1PL	0 PL	0 PL

ADR/SIZ	$E0nFFFF6/$FFF41126 (8 bits [5 used] of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	Reserved	LOCK	Reserved		LBTS1	LBTS0	LBSC1	LBSC0
LOPER	R	R/W	R		R/W	R/W	R/W	R/W
VOPER	R	R/W	R		R/W	R/W	R/W	R/W
RESET	0	0 PL	0	0	0 PL	0 PL	0 PL	0 PL

REN Read Enable. When this bit is set, read access to the address range programmed in the VSB Slave 2 Address Range Register is allowed and the VSBchip2 either responds to or participates in read cycles depending upon the state of the POR bit.

WEN Write Enable. When this bit is set, write access to the address range programmed in the VSB Slave 2 Address Range Register is allowed and the VSBchip2 either responds to or participates in write cycles depending upon the state of the POW bit.

POR Participate on Read. When the REN bit is cleared this bit is not relevant. However, when the REN bit is set, this bit defines whether the VSBchip2 is the responding slave or a participating slave in read cycles to the address range programmed in the VSB Slave 2 Address Range Register.

When this bit is set, the VSBchip2 is a participator; and when this bit is cleared, the responder. The default state of this bit is cleared - the VSBchip2 is a responding slave.

POW Participate on Write. When the WEN bit is cleared this bit is not relevant. However, when the WEN bit is set, this bit defines whether the VSBchip2 is the responding slave or a participating slave in write cycles to the address range programmed in the VSB Slave 2 Address Range Register. When this bit is set, the VSBchip2 is a participator; and when this bit is cleared, the responder. The default state of this bit is cleared - the VSBchip2 is a responding slave.

WPE Write Post Enable. When this bit is set, write posting is enabled for the address range defined by the VSB Slave 2 Address Range Register.

SAS System Address Space. When access to the local bus is permitted (responding or participating capability must be enabled), setting this bit defines the VSB Slave 2 Address Range to be in the VSB System Address Space. This is the default location.

ALTAS Alternate Address Space. When access to the local bus is permitted (responding or participating capability must be enabled), setting this bit defines the VSB Slave 2 Address Range to be in the VSB Alternate Address Space.

IOAS I/O Address Space. When access to the local bus is permitted (responding or participating capability must be enabled), setting this bit defines the VSB Slave 2 Address Range to be in the VSB I/O Address Space.

LOCK Lock Local Bus on Block Transfers. The Lock bit, if set, causes the VSBchip2 local master to assert LBB* on the start of a VSB block transfer. This effectively prevents any other local bus master from taking the bus back, and allows higher speed block transfers. Only when the VSB master removes VPAS*, is the bus released.

This software Lock contrasts with the VSB lock signal (VLOCK*) which may be used to keep the bus locked between VSB transfers as well as within VSB transfers. Using VLOCK* allows an external VSB master to perform read-modify-write cycles, for example, which is not possible with this software Lock bit.

LBTS1 - LBTS0

Local Bus Transfer Size. These bits define the port size of the VSB Slave 2 address range to be 8-bits, 16-bits, or 32-bits. The port size programmed is reflected by the value of VASACK1* - VASACK0* driven onto the VSB, and by LSIZ1 and LSIZ0 driven onto the local bus. Refer to the table in the section on Local Bus Master Interface at the beginning of this chapter for this encoding. LBTS1 and LBTS0 are encoded as follows:

LBTS1	LBTS0	Port Size
0	0	32-bits
0	1	16-bits
1	0	8-bits
1	1	No Responder

LBSC1 - LBSC0

Local Bus Snoop Control. These bits control the snoop enable lines to the local bus for the address range defined by the VSB Slave 2 Address Range Register. LBSC1 and LBSC0 are encoded as follows:

LBSC1	LBSC0	Snoop Function
0	0	Snoop Inhibited
0	1	Write - Sink data Read - Supply dirty data and leave dirty
1	0	Write - Invalidate Read - Supply dirty data and mark invalid
1	1	Snoop Inhibited

Reserved Register

ADR/SIZ	$E0nFFFF8/$FFF41128 (32 bits [0 used])		
BIT	31	. . .	0
NAME	Reserved		
OPER	R		
RESET	$00000000		

This register is reserved for future expansion.

VSB Error Address Register

ADR/SIZ	$E0nFFFFC/$FFF4112C (32 bits)		
BIT	31	. . .	0
NAME	VSB Error Address		
OPER	R		
RESET	$00000000 P		

If the VWPIF bit in the VSB Interrupt Status Register is set, then this register contains the address stored in the VSB write post buffer at the time the last write post error was detected. This register does not change until another VSB write post error is detected.

If the VWPIF interrupt is not handled quickly, a subsequent write post error overwrites the original contents of this register.

PRINTER AND SERIAL PORT CONNECTIONS 7

Introduction

This chapter has connection diagrams for the printer port and the four serial ports on the MVME167/187, and for the serial ports on the MVME166. These ports are connected to external devices through the MVME712 series of transition modules.

The configuration of the serial ports as Data Terminal Equipment (DTE) or Data Circuit-terminating Equipment (DCE) is accomplished by jumpers on the transition modules. For more information, refer to the *MVME712-10 Transition Module User's Manual,* the *MVME712-06/07/09 I/O Distribution Board Set User's Manual,* the *MVME712-12, MVME712-13, MVME712A, MVME712AM, and MVME712B Transition Modules and LCP2 Adapter Board User's Manual,* or the *MVME712M Transition Module and P2 Adapter Board User's Manual.*

Connection Diagrams

The MVME712X transition module connection diagrams are shown in the following figures:

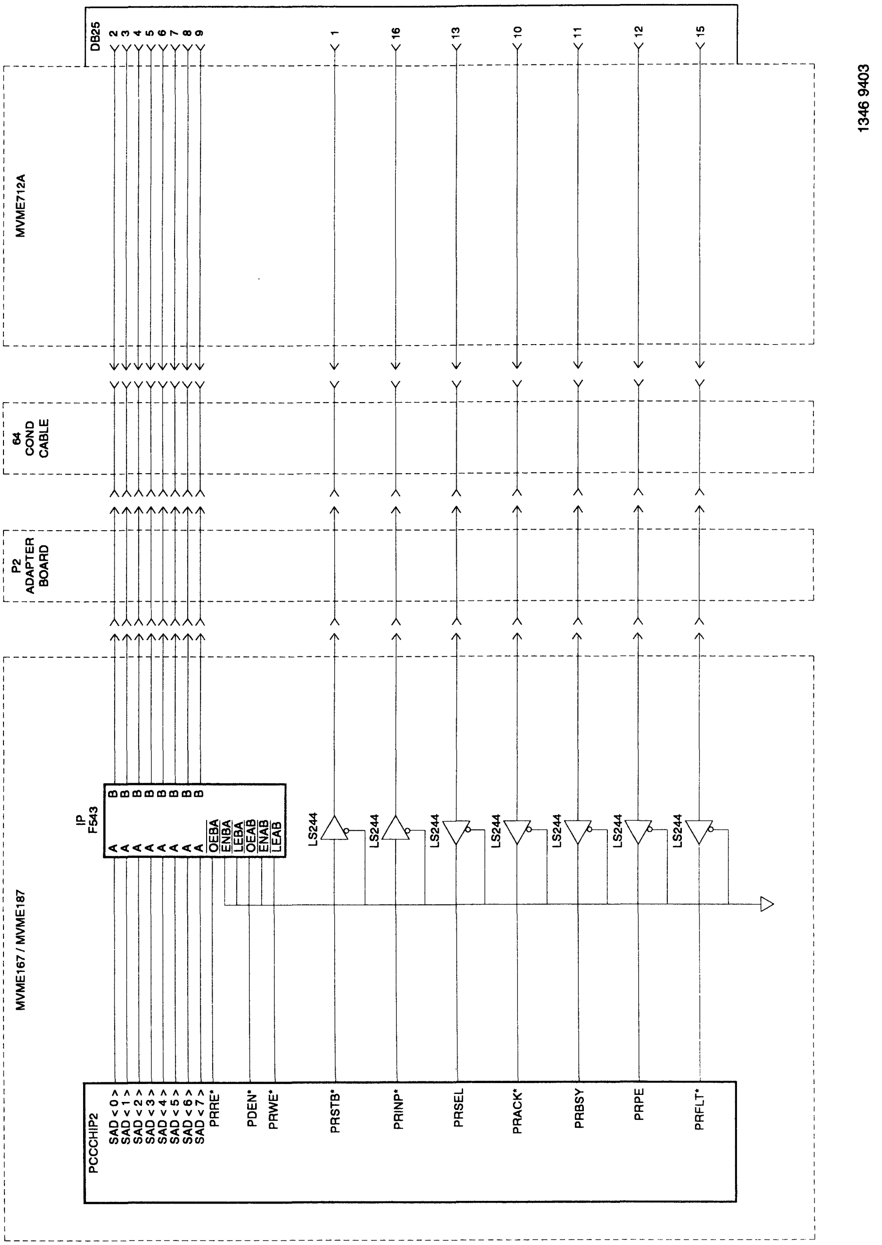


Figure 7-1. MVME167/187 Printer Port with MVME712A

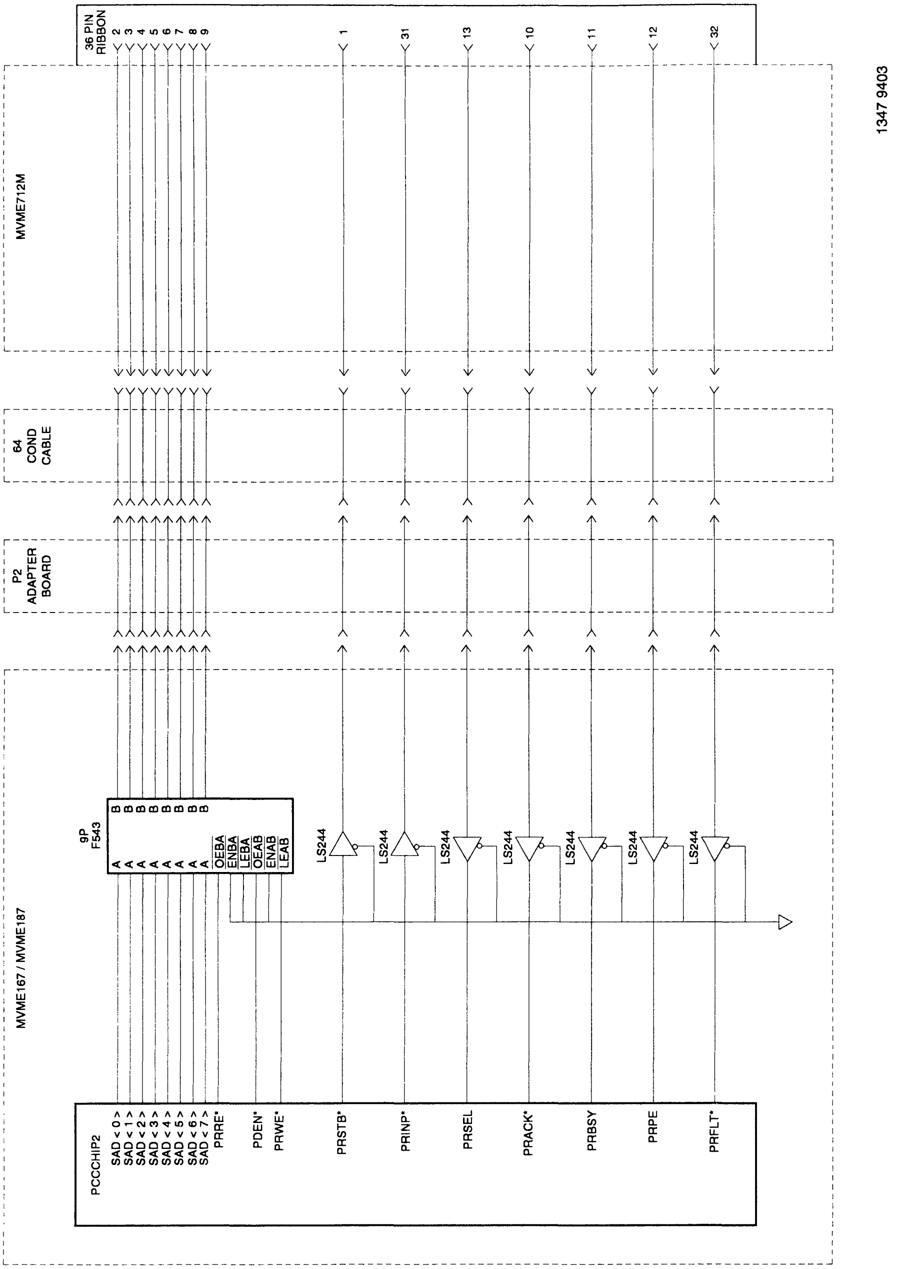


Figure 7-2. MVME167/187 Printer Port with MVME712M

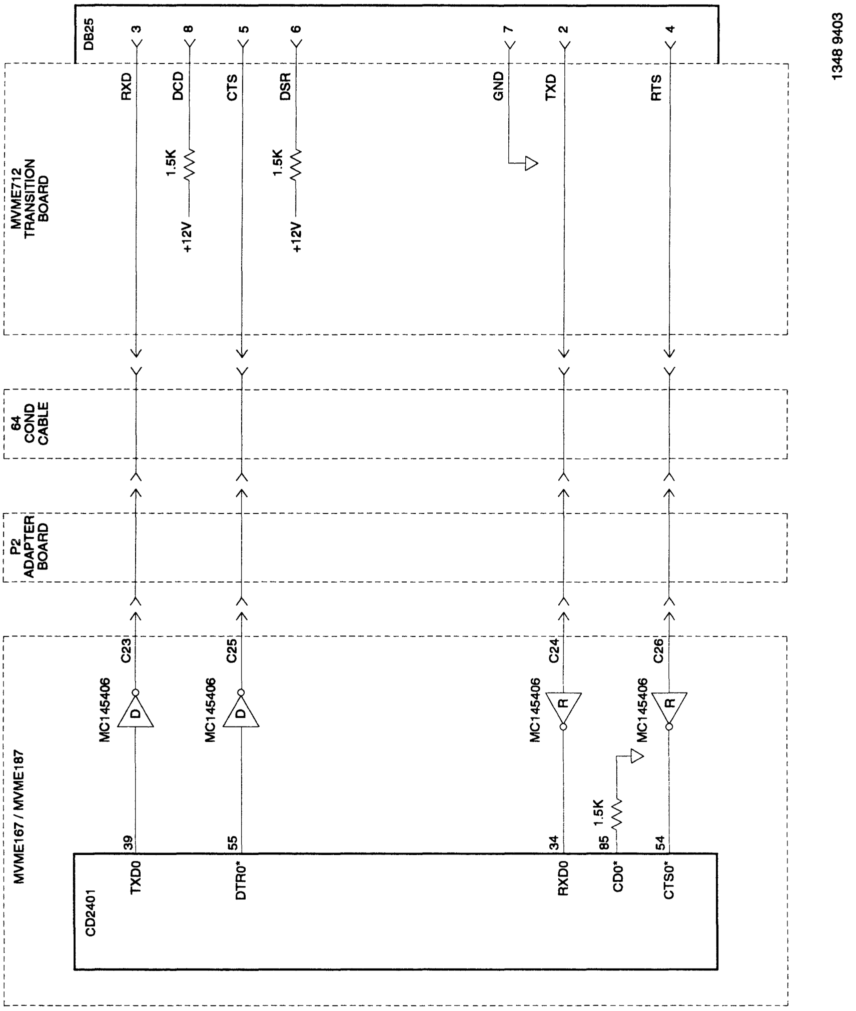


Figure 7-3. MVME167/187 Serial Port 1 Configured as DCE

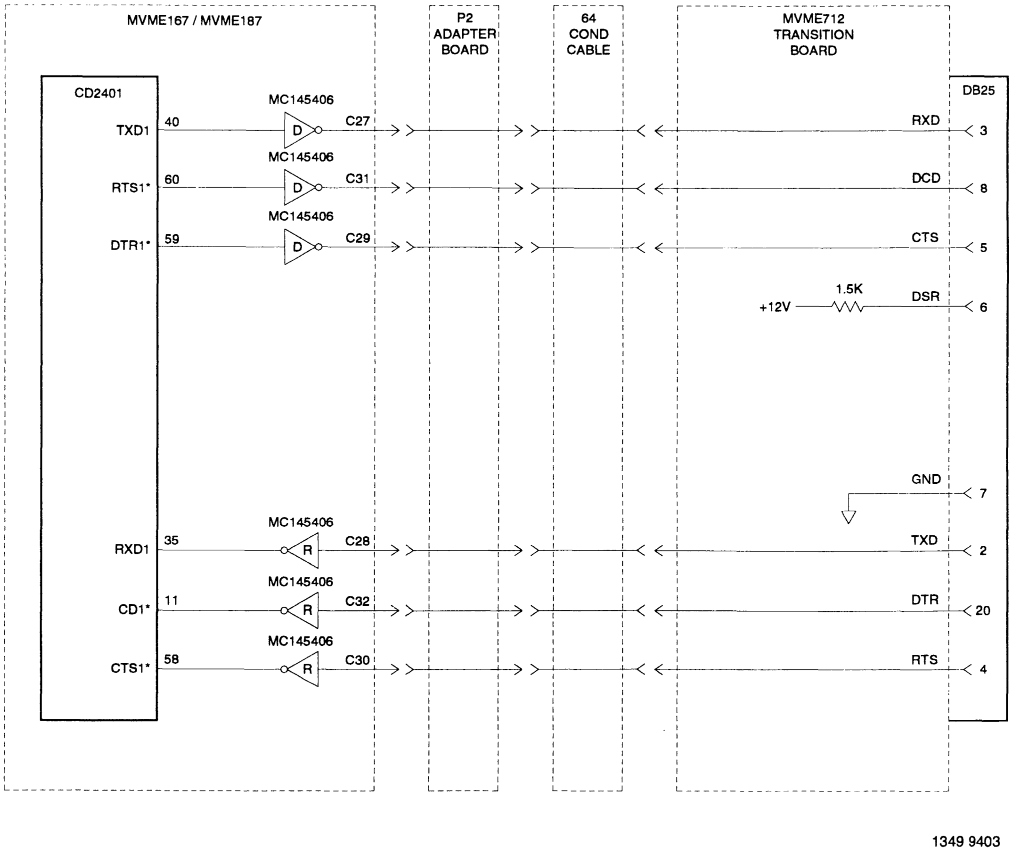


Figure 7-4. MVME167/187 Serial Port 2 Configured as DCE

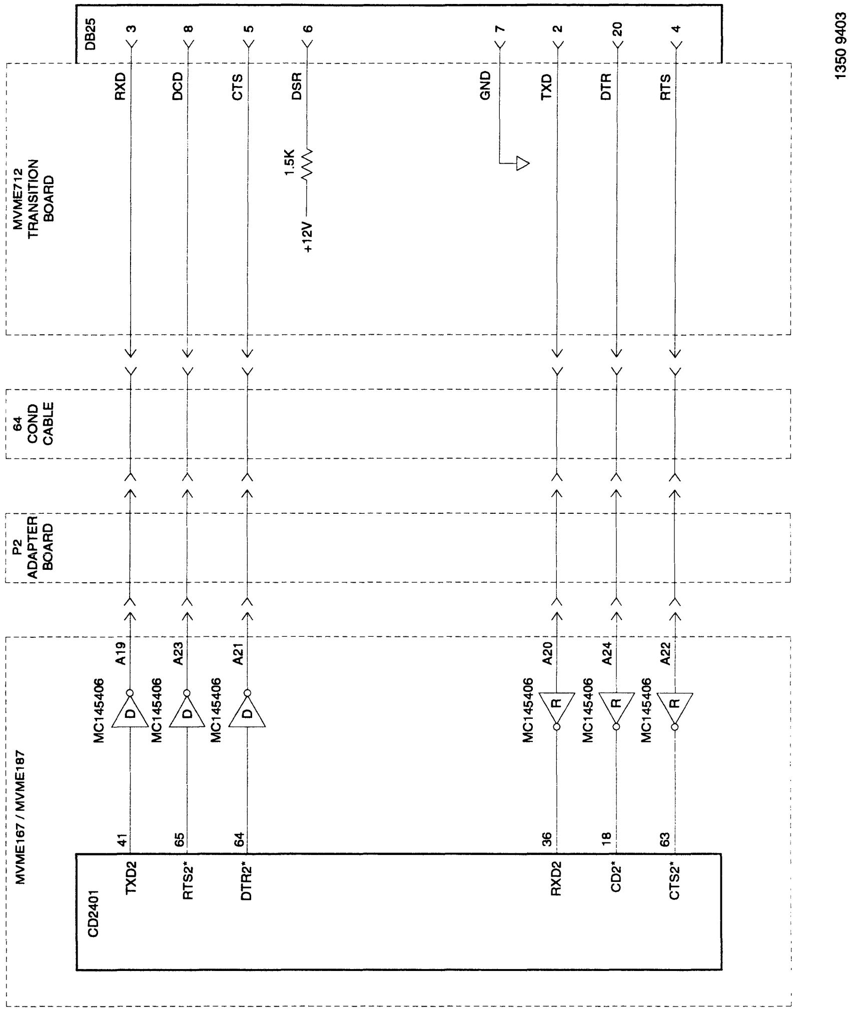


Figure 7-5. MVME167/187 Serial Port 3 Configured as DCE

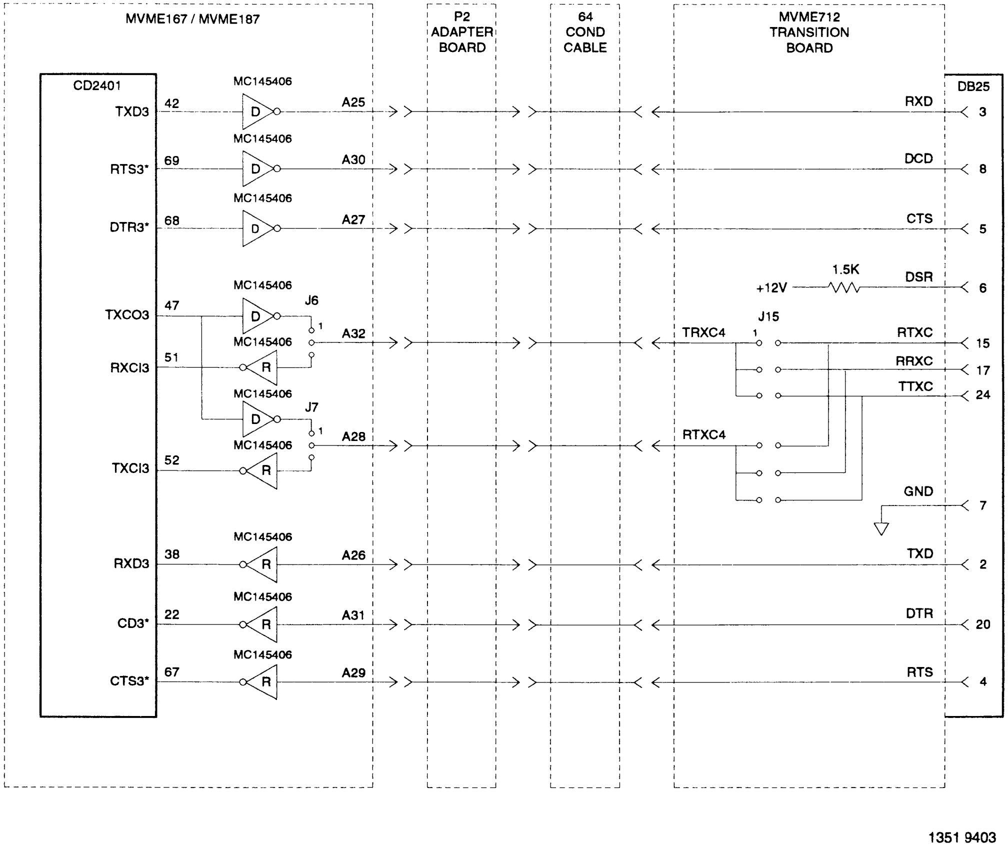


Figure 7-6. MVME167/187 Serial Port 4 Configured as DCE

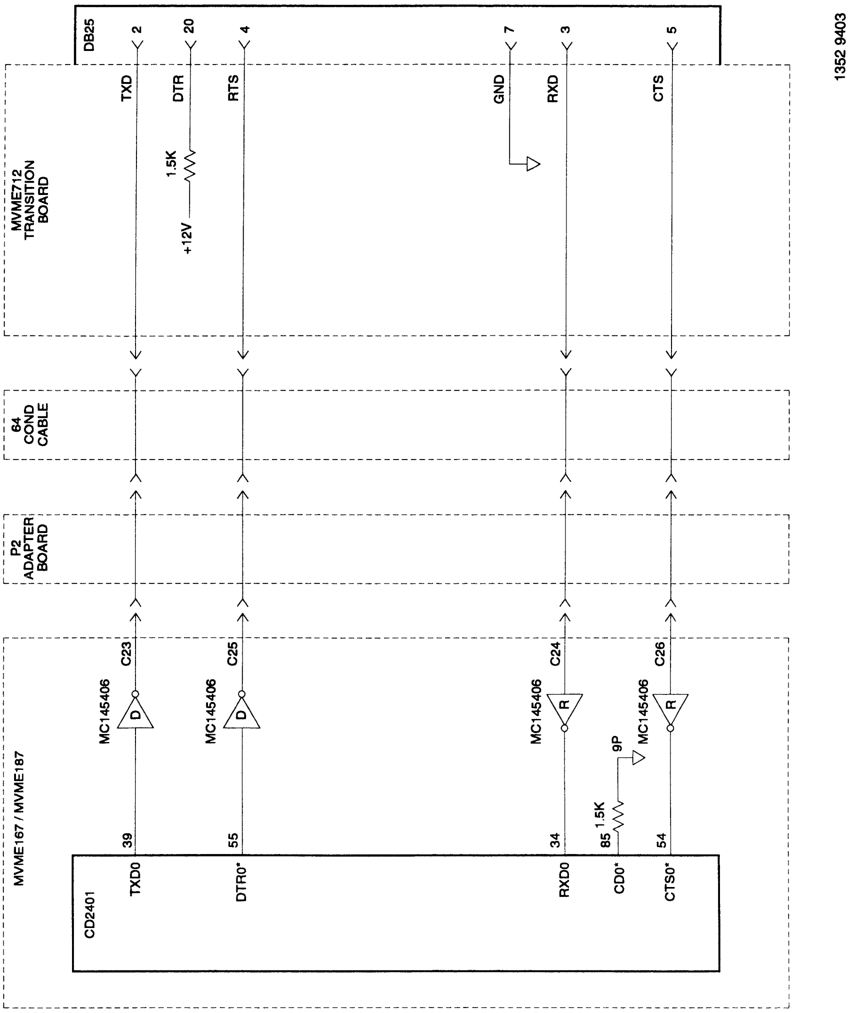


Figure 7-7. MVME167/187 Serial Port 1 Configured as DTE

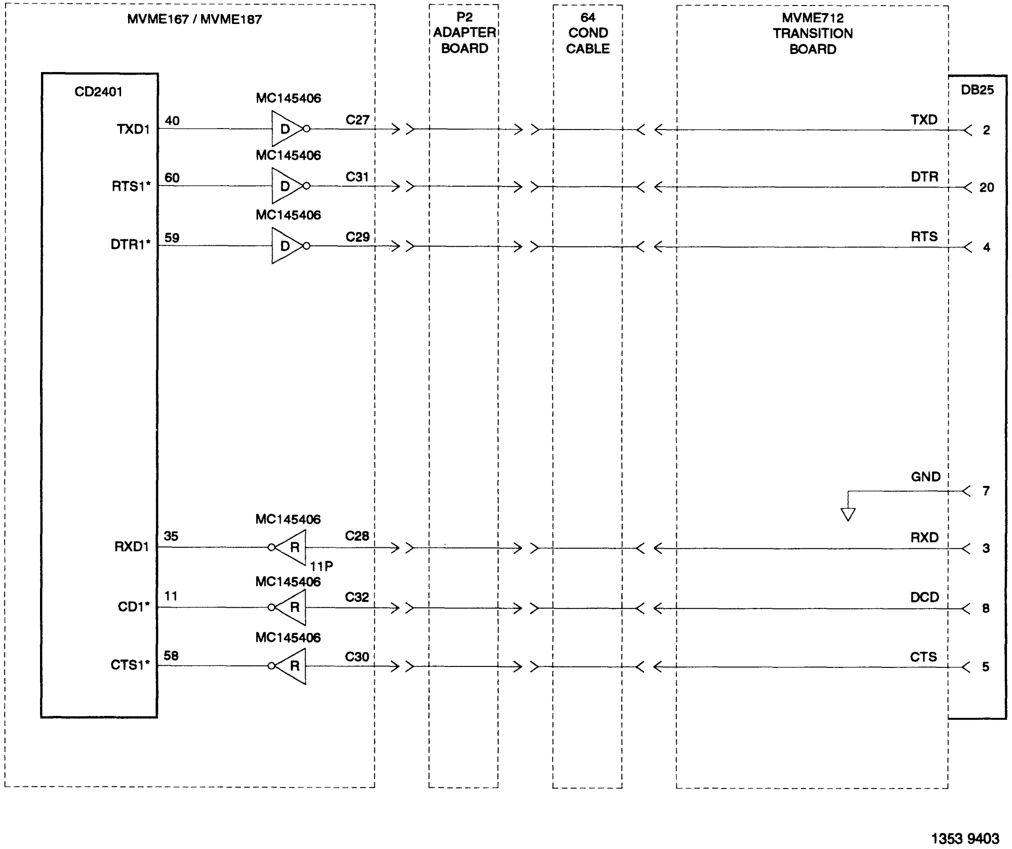


Figure 7-8. MVME167/187 Serial Port 2 Configured as DTE

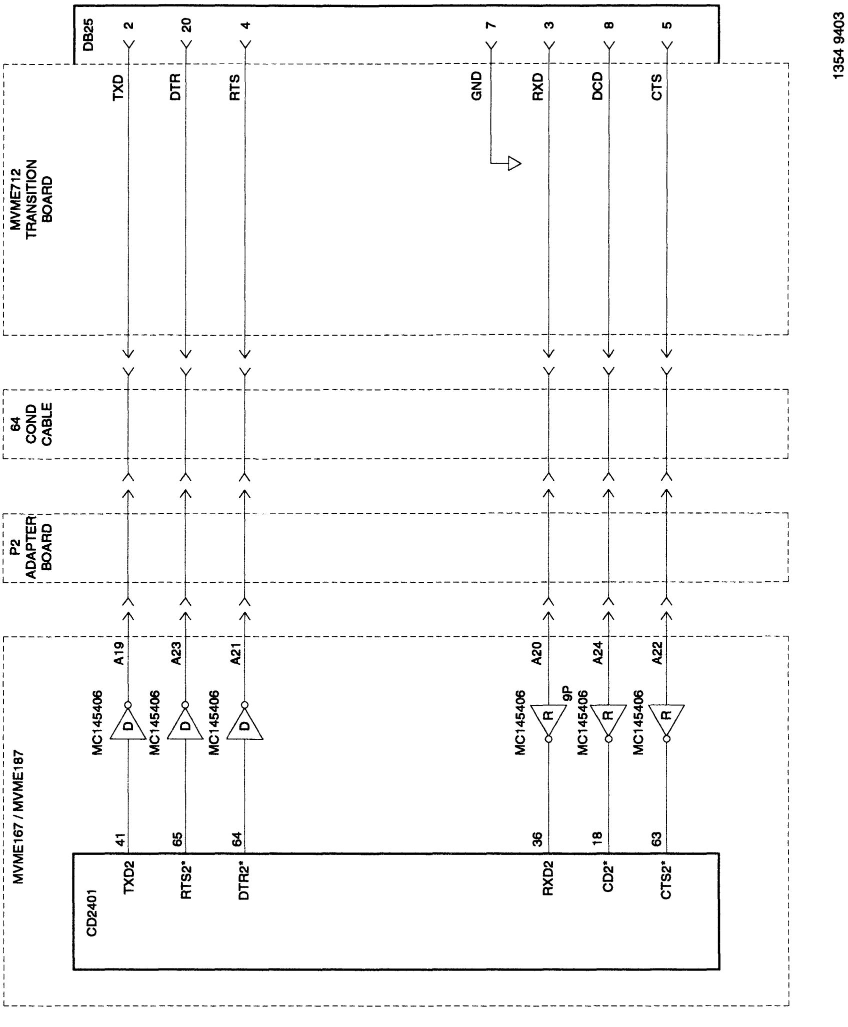


Figure 7-9. MVME167/187 Serial Port 3 Configured as DTE

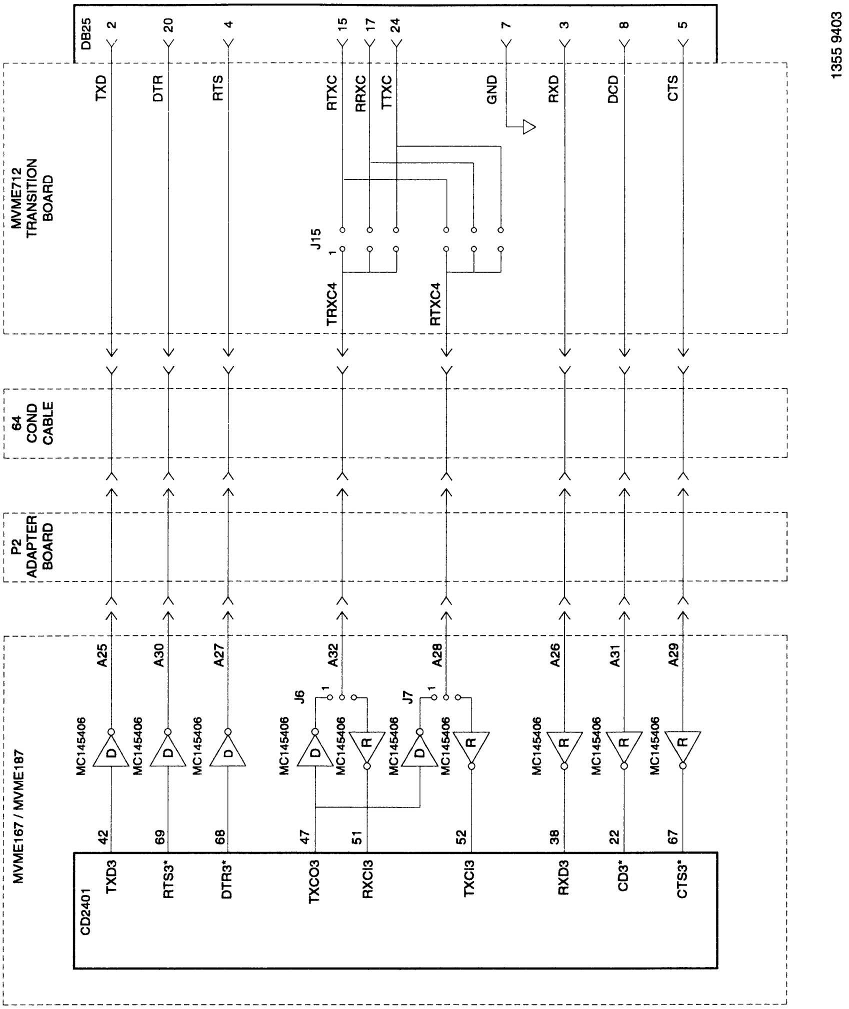


Figure 7-10. MVME167/187 Serial Port 4 Configured as DTE

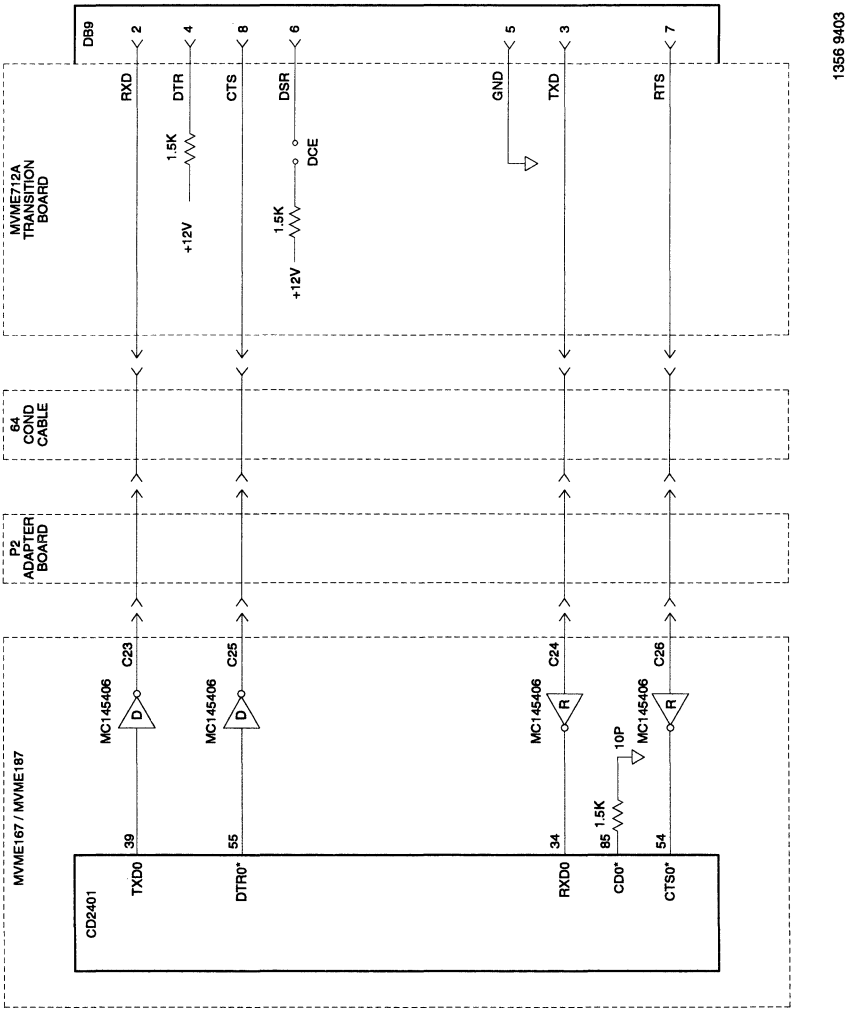


Figure 7-11. MVME167/187 Serial Port 1 with MVME712A

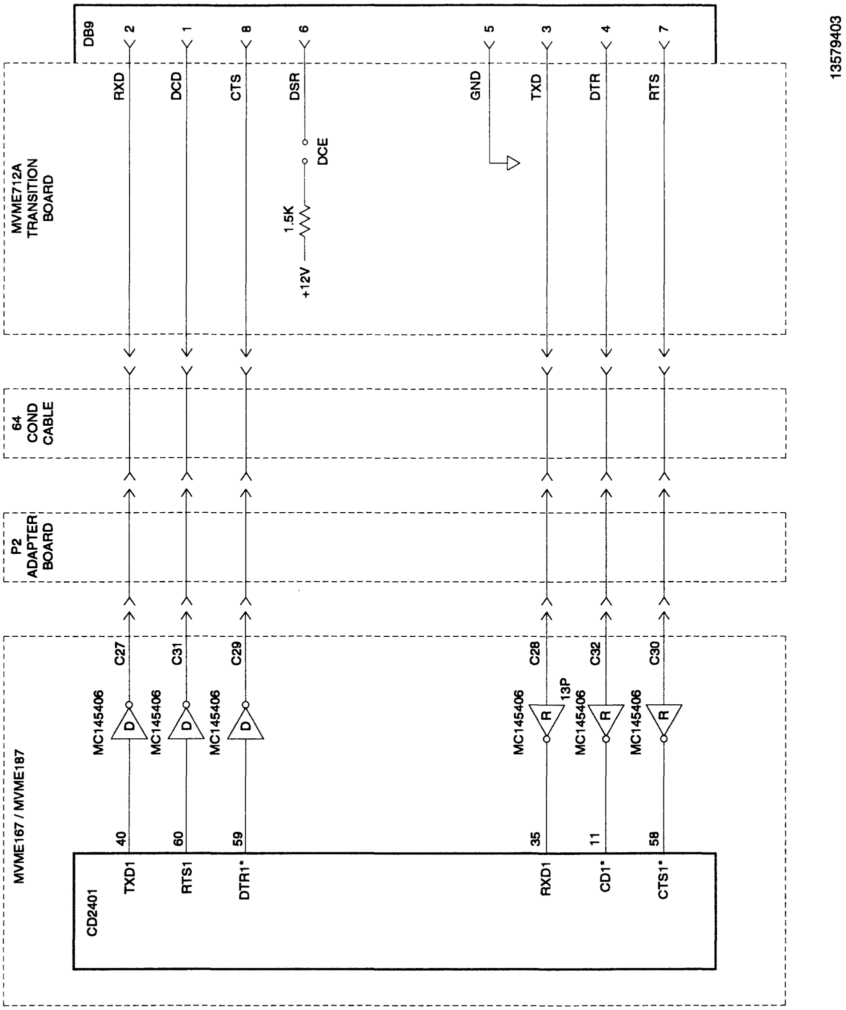


Figure 7-12. MVME167/187 Serial Port 2 with MVME712A

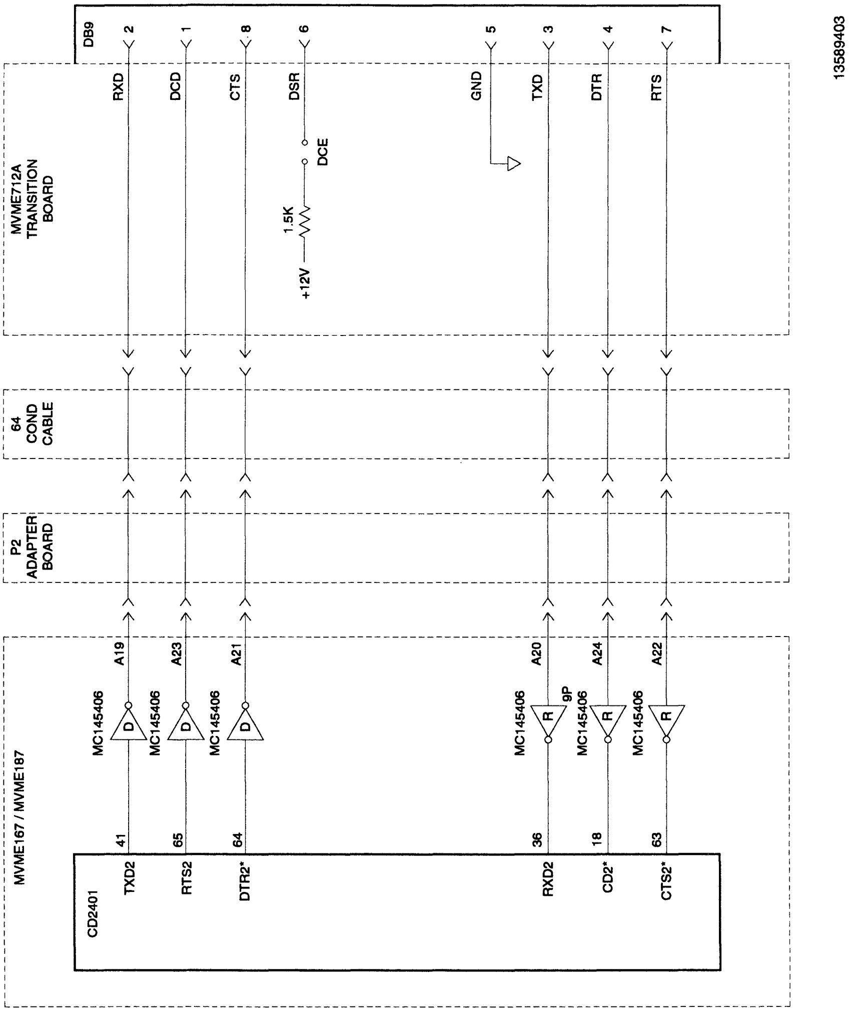


Figure 7-13. MVME167/187 Serial Port 3 with MVME712A

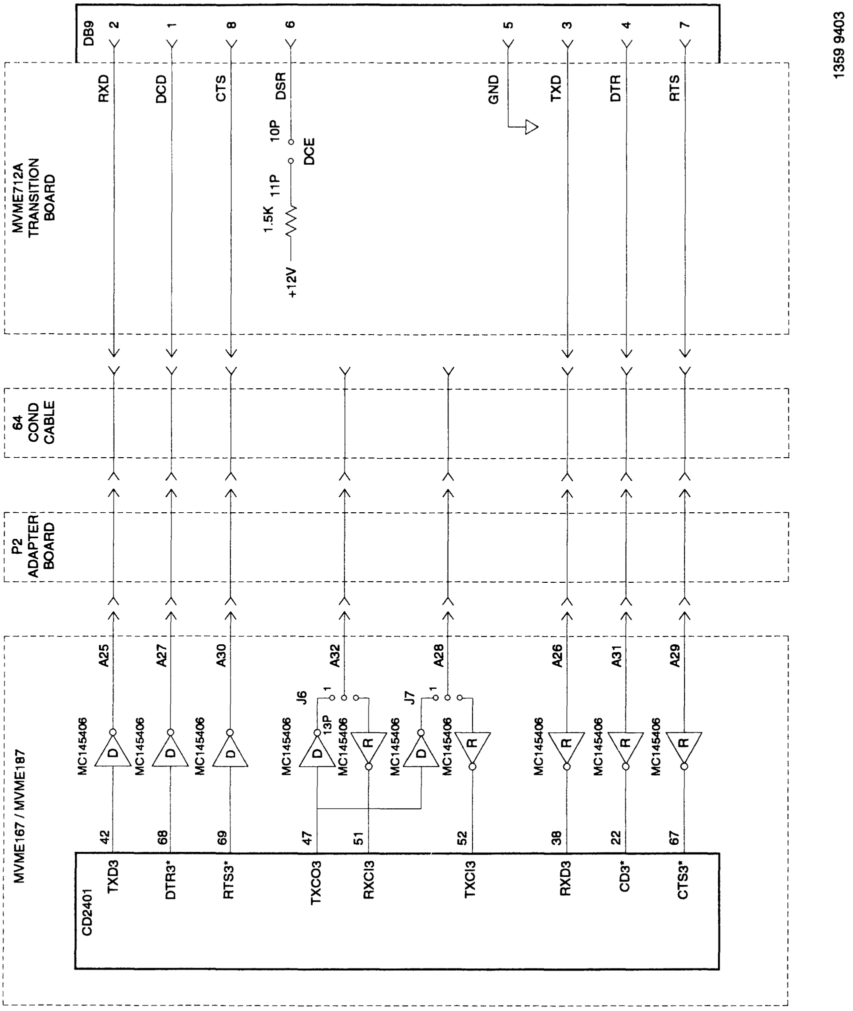


Figure 7-14. MVME167/187 Serial Port 4 with MVME712A

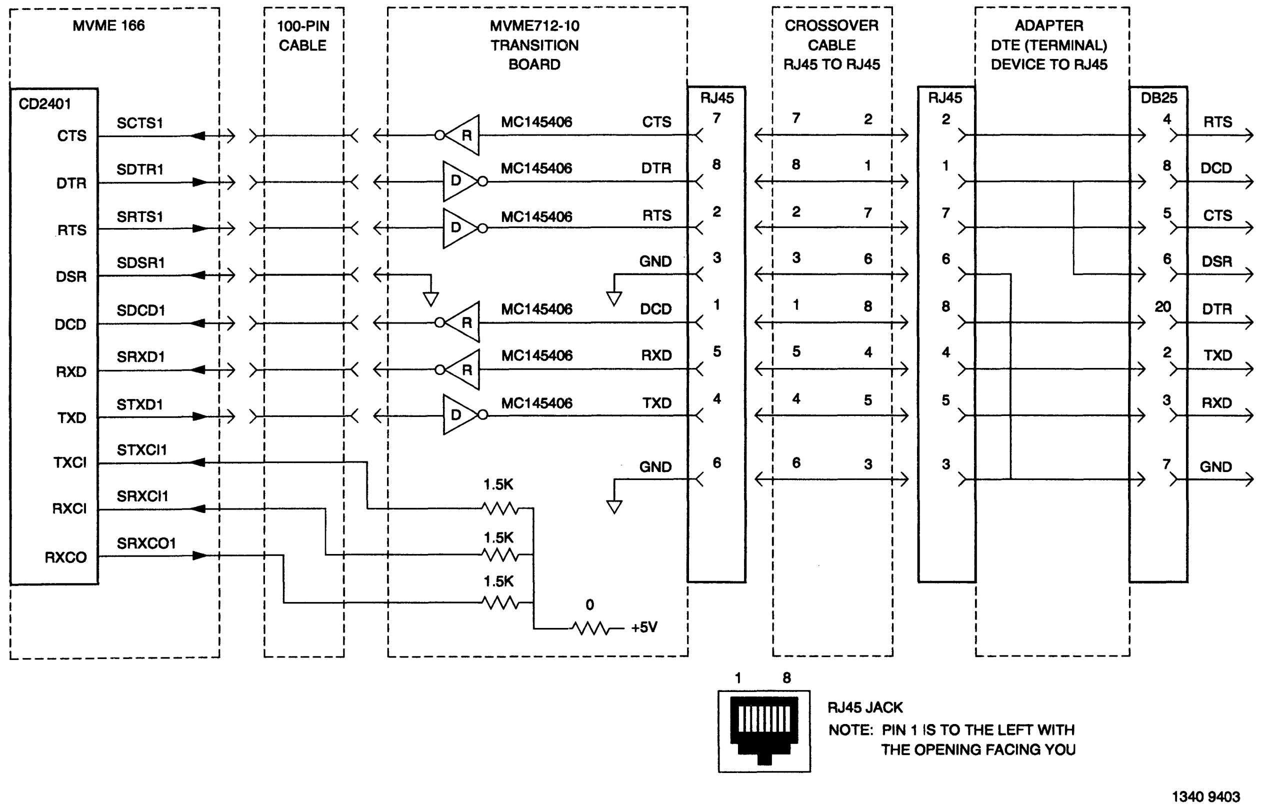
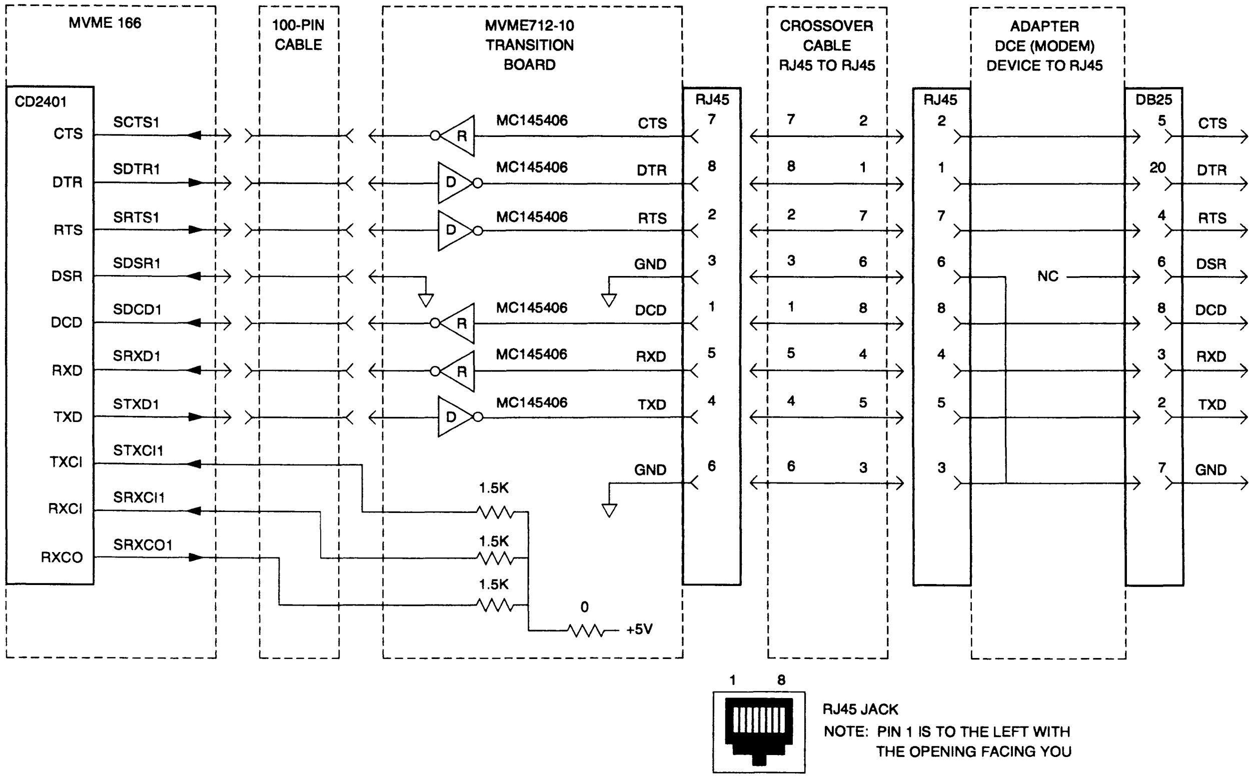


Figure 7-15. MVME166 Serial Ports with MVME712-10 (Sheet 1 of 4)

Configured for DCE



Configured for DTE

Figure 7-15. MVME166 Serial Ports with MVME712-10 (Sheet 2 of 4)

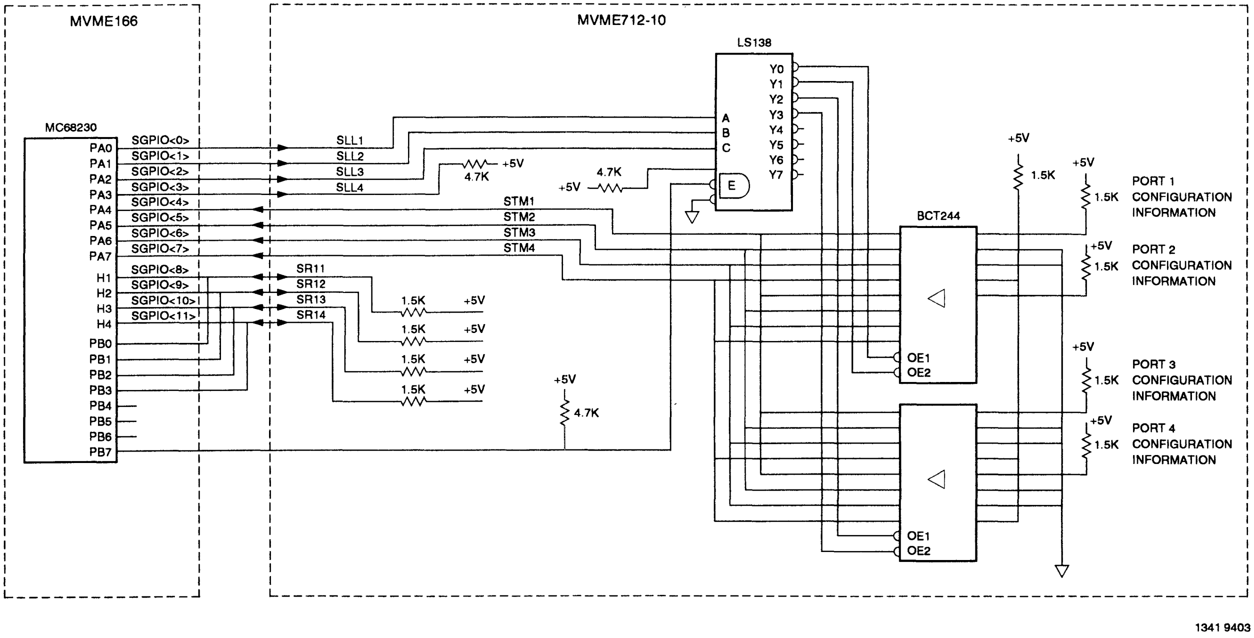


Figure 7-15. MVME166 Serial Ports with MVME712-10 (Sheet 3 of 4)

Configuration Register

Configuration Register Notes

The application software on the MVME166 can read the configuration register to identify the transition board(s) in the system.

On the MVME166, the interrupt level on the MC68230 Parallel Interface Timer (PI/T) is the same as the CD2401 Serial Controller Chip (SCC). The interrupt vector is as programmed in the MC68230. The interrupt priority is (1) first the CD2401, and (2) then the MC68230.

To read the configuration information, the MC68230 (PI/T) must be programmed as follows:

Port A direction must be:
Mode 0, submode 1X
Bits 7 to 4 as inputs
Bits 3 to 0 as outputs

Port B direction must be:
Mode 0, submode 1X
Bit 7 as an output
Bits 6 to 0 as inputs

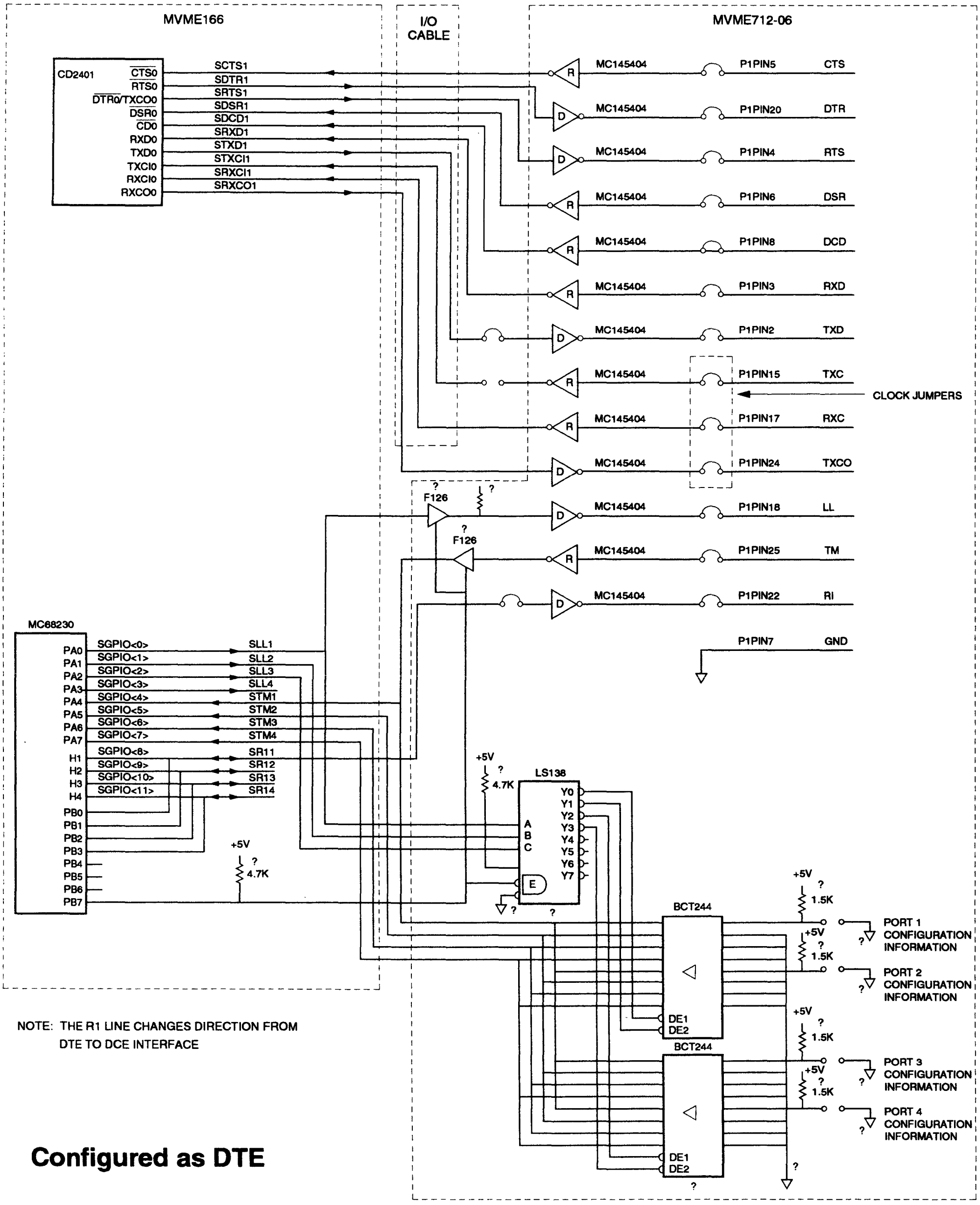
Write port B bits 7 to 0 set configuration mode. Now port A bits 3 to 0 are used to select a port.

Port A Bits 3 to 0	Port Selected
0	0
1	1
2	2
3	3

Port A bits 7 to 4 are the configuration data returned from the port selected:

Configuration Register Port A Bits 7 to 4	Module Type	Module Implemented on
0	EIA-232 DCE	MVME712-06
1	EIA-232 DTE	MVME712-06
2-8	Reserved	
9	EIA-232 RJ45 DTE	MVME712-10
A-F	Reserved	

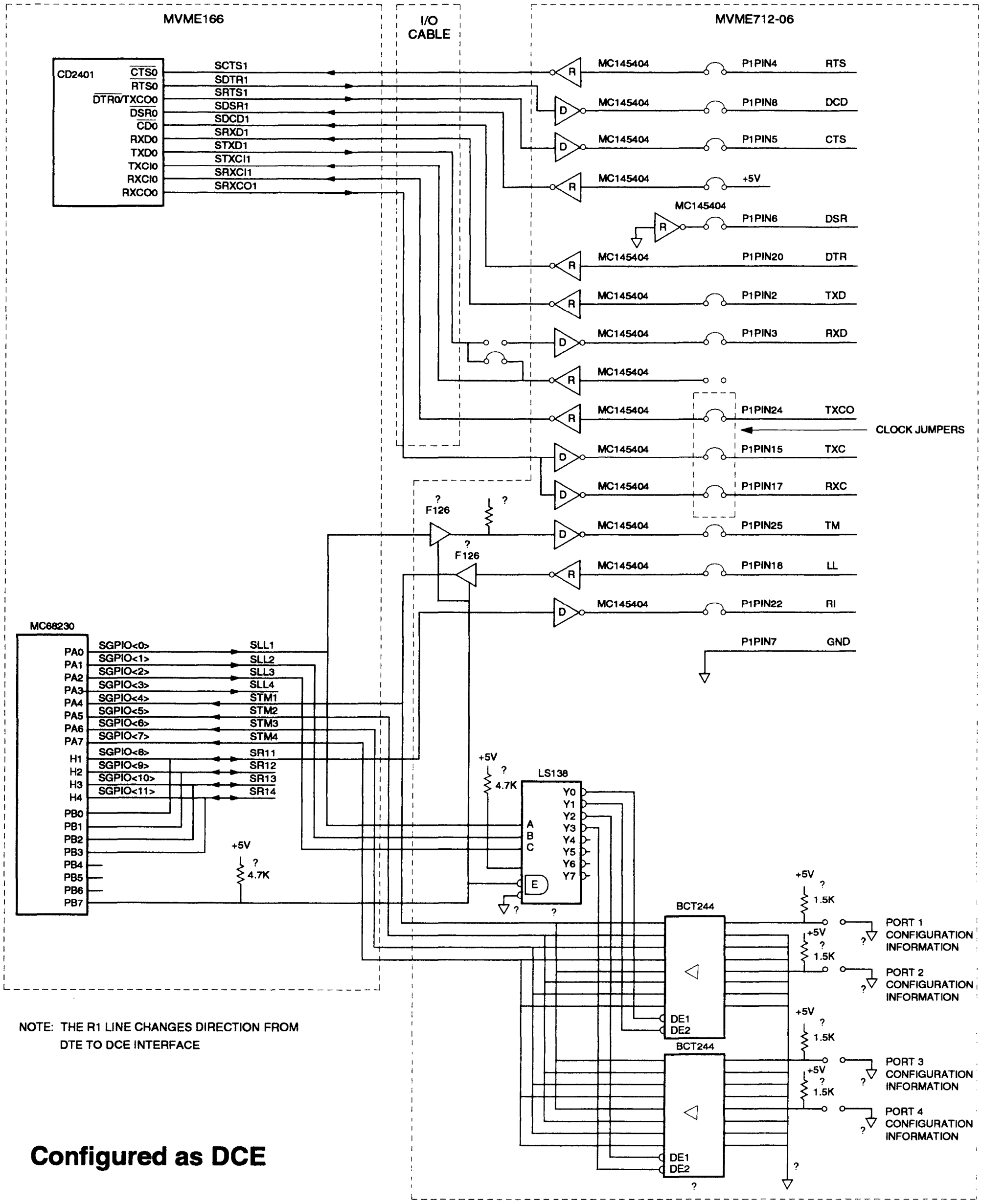
Figure 7-15. MVME166 Serial Ports with MVME712-10 (Sheet 4 of 4)



NOTE: THE R1 LINE CHANGES DIRECTION FROM DTE TO DCE INTERFACE

Configured as DTE

Figure 7-16. MVME166 Serial Ports with MVME712-06 (Sheet 1 of 3)



NOTE: THE R1 LINE CHANGES DIRECTION FROM DTE TO DCE INTERFACE

Configured as DCE

Figure 7-16. MVME166 Serial Ports with MVME712-06 (Sheet 2 of 3)

Configuration Register Notes

The application software on the MVME166 can read the configuration register to identify the transition board(s) in the system.

On the MVME166, the interrupt level on the MC68230 Parallel Interface Timer (PI/T) is the same as the CD2401 Serial Controller Chip (SCC). The interrupt vector is as programmed in the MC68230. The interrupt priority is (1) first the CD2401, and (2) then the MC68230.

To read the configuration information, the MC68230 (PI/T) must be programmed as follows:

Port A direction must be:
Mode 0, submode 1X
Bits 7 to 4 as inputs
Bits 3 to 0 as outputs

Port B direction must be:
Mode 0, submode 1X
Bit 7 as an output
Bits 6 to 0 as inputs

Write port B bits 7 to 0 set configuration mode. Now port A bits 3 to 0 are used to select a port.

Port A Bits 3 to 0	Port Selected
0	0
1	1
2	2
3	3

Port A bits 7 to 4 are the configuration data returned from the port selected:

Configuration Register Port A Bits 7 to 4	Module Type	Module Implemented on
0	EIA-232 DCE	MVME712-06
1	EIA-232 DTE	MVME712-06
2-8	Reserved	
9	EIA-232 RJ45 DTE	MVME712-10
A-F	Reserved	

Figure 7-16. MVME166 Serial Ports with MVME712-06 (Sheet 3 of 3)

USING INTERRUPTS ON THE MVME166/167/187 A

Introduction

This appendix demonstrates how to use interrupts on the MVME166, MVME167, and MVME187.

Read this entire appendix before performing any of these procedures.

The first section gives an example of how to generate and handle a VMEchip2 Tick Timer 1 interrupt on the MVME167. Specific values have been given for the register writes.

The second section of this appendix talks about how interrupts are handled on the MVME187. Interrupts are handled substantially differently on the MVME187 than on the MVME167.

MVME167 VMEchip2 Tick Timer 1 Periodic Interrupt Example

A. Set up Tick Timer 1.

Step	Register	Address	Action and Reference
1.	Prescaler Control Register	$FFF4004C	If not already initialized by the debugger, initialize as follows: Prescaler Register = 256 - **Bclock** (MHz). This gives a 1 MHz clock to the tick timers. **Bclock** is the bus clock rate, such as 25 MHz. 256 - 25 = $E7.
2.	Tick Timer 1 Compare Register	$FFF40050	For periodic interrupts, set the Compare Register value = **Period** (µs). For example, if you want an interrupt every millisecond, set the register value to 1000 ($3E8). Refer to the Tick Timer 1 Compare Register description in Chapter 2.

Step	Register	Address	Action and Reference
3.	Tick Timer 1 Counter Register	$FFF40054	Write a zero to clear.
4.	Tick Timer 1 Control Register	$FFF40060 (8 bits)	Write $07 to this register (set bits 0, 1, and 2). This enables the Tick Timer 1 counter to increment, resets the count to zero on compare, and clears the overflow counter.

B. Set up local bus interrupter:

Step	Register	Address	Action and Reference
5.	Vector Base Register	$FFF40088 (8 of 32 bits)	If not already initialized by the debugger, set interrupt base register 0 by writing to bits 28-31. Refer to the Vector Base Register description and to Table 2-3, the Local Bus Interrupter Summary, in Chapter 2.
6.	Interrupt Level Register 1 (bits 0-7)	$FFF40078 (8 of 32 bits)	Write desired level of Tick Timer 1 interrupt to bits 0-2.
7.	Local Bus Interrupter Enable Register	$FFF4006C (8 of 32 bits)	Set bit 24 (ETIC1) to one to enable Tick Timer 1 interrupts.
8.	I/O Control Register 1	$FFF40088 (8 of 32 bits)	Write a one to bit 23 to enable interrupts from the VMEchip2. A zero masks *all* interrupts from the VMEchip2.

Periodic Tick Timer 1 interrupts now occur, so you need an interrupt handler. Section C gives the details, as follows.

C. How to set up an interrupt handler routine. (Your interrupt handler should include the following features.)

Step	Action and Reference
1.	Be sure the MC68040 vector base register is set up. Set the proper MC68040 exception vector location so the processor vectors to your interrupt handler location. You can determine proper exception vector location to set from the MC68040 vector base register, the VMEchip2 base register, and Table 2-3, the Local Bus Interrupter Summary, in Chapter 2, from which you can determine the actual interrupt vector given on a Tick Timer 1 interrupt. Lower the MC68040 mask so the vector level you programmed is accepted. The *interrupt handler itself* should include the following (steps 2 through 5).
2.	Confirm the Tick Timer 1 interrupt occurred, by reading the status of bit 24 of the Interrupter Status Register at $FFF40068. A high indicates an interrupt present.
3.	Clear Tick Timer 1 interrupt by writing a one to bit 24 of the Interrupt Clear Register at $FFF40074.
4.	Increment a software counter to keep track of the number of interrupts, if desired. Output a character or some other action (such as toggling the FAIL LED) on an appropriate count, such as 1000.
5.	Return from exception.

MVME187 Interrupt Handling

The M88000 architecture currently does not support prioritized interrupts like the MC68000 family. A single interrupt request is connected to the MC88100, and all device interrupt requests are summed into this signal. Software prioritization is required to implement interrupt priority encoding. Logic on the MVME187 assists the 88K software by encoding the interrupt priority in hardware.

A summary of the interrupt handling protocol for the MVME187 follows. Step numbers refer to circled items on Figure A-1.

Step **Action and Reference**

1. A device requests interrupt service from the MPU. The hardware logic on the MVME187 PCCchip2 encodes this request on the INT signal connected to the MC88100. INT is asserted if the following is true:
   a. PCCchip2 Master Interrupt Enable is on, and
   b. External (Device) Interrupt Request Level is greater than the Interrupt Mask Level Register and greater than 0.

2. The MC88100 saves its current context in order to service the interrupt request. The M88000 architecture requires that nested exceptions be handled totally in software. The MVME187 logic provides an external prioritizer for the hardware interrupts to assist in nested interrupt architecture.

   Software applications (Operating Systems, etc.) are encouraged to take advantage of the interrupt prioritizer. An example of how to use the prioritizer is given in the following steps.

   After the context of the processor is saved, the MC88100 is able to mask out interrupts with priorities lower than the current request under service. This masking operation is optional but recommended.

3. The MC88100 then (through the Data CMMU) reads the Interrupt Priority Level Register on the PCCchip2 to determine the priority level of the interrupt request. At this time, the priority level is determined and the corresponding interrupt mask may be written to the PCCchip2, which masks any interrupts of equal or lower priority. This masking allows the application software (or Operating System) to unfreeze shadowing and quickly re-enable exception handling. Without the Interrupt Mask Level Register and its associated interrupt prioritization, interrupt prioritization would have to be handled totally through software.

4. The PCCchip2 Interrupt Priority Level Register contains a 3-bit value corresponding to the level of the interrupt requesting service. The MC88100 determines (through a table lookup or some other clever means) the address that emulates an IACK cycle on the MC68040 bus (local bus) corresponding to this priority. The MC88100/Data CMMU drives this address on the bus (e.g., through a LOAD instruction) which generates an MC68040-like IACK cycle on the local bus.

5. The IACK cycle causes a "vector" to be passed to the Data CMMU/MC88100 that can be used to index a table of interrupt service routine pointers. The initialization of the MVME187 module determines the interrupt priority and the device interrupt vectors, relieving considerable software overhead to determine interrupt priority.

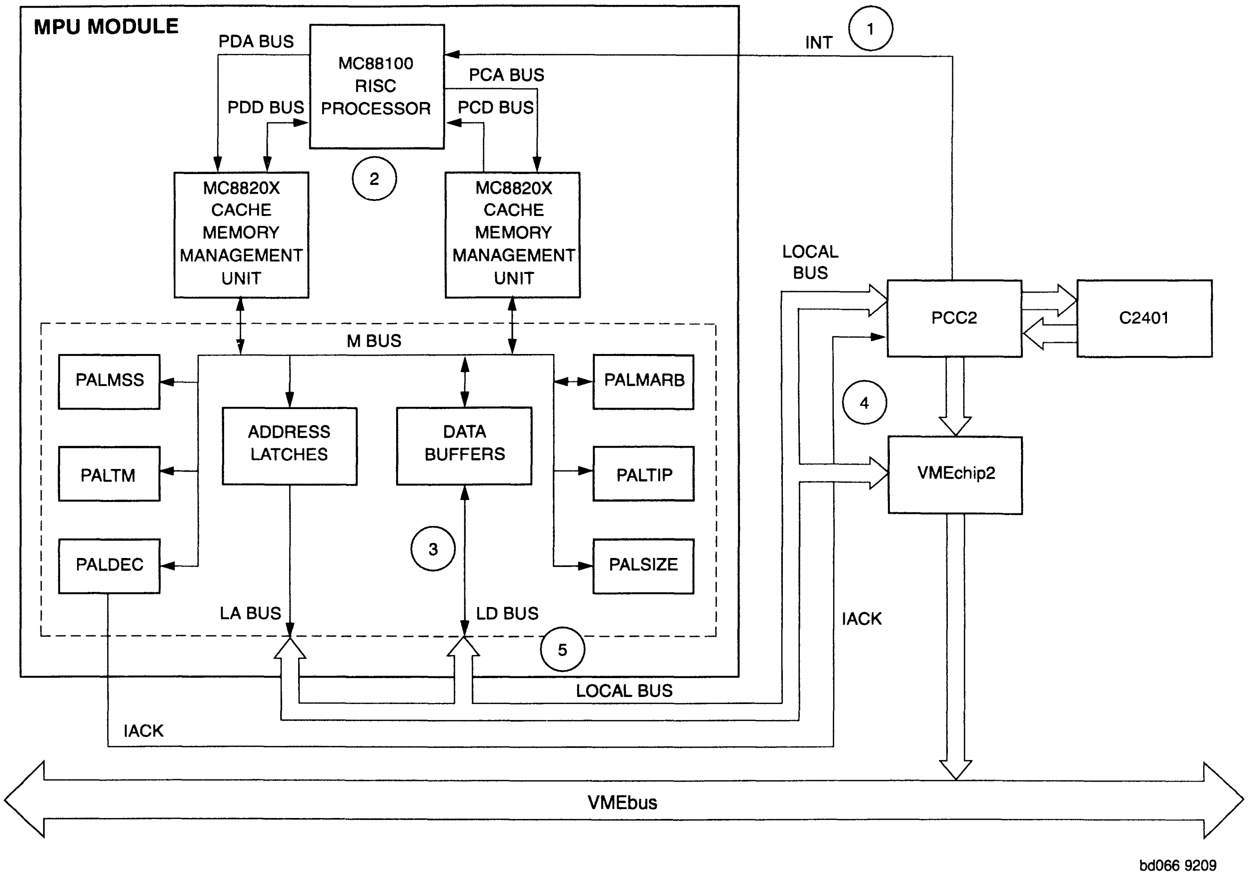


Figure A-1. MVME187 Interrupt Handling Protocol

Index

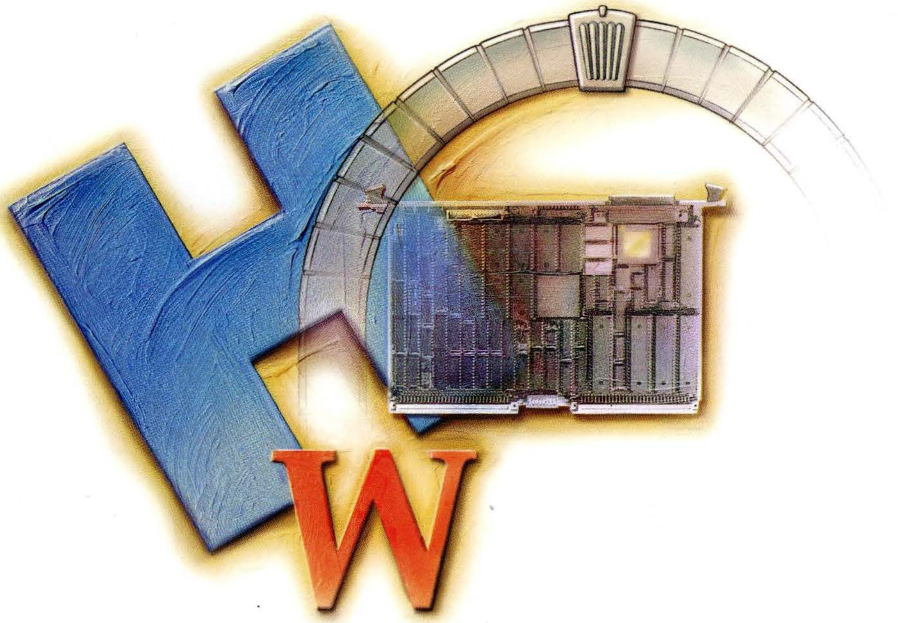
When using this index, keep in mind that a page number indicates only where referenced material begins. It may extend to the page or pages following the page referenced.

INDEX

INDEX

INDEX

INDEX



MOTOROLA

Computer Group

2900 South Diablo Way
Tempe, Arizona 85282
P.O. Box 2953
Phoenix, Arizona 85062-2953

To order additional Computer Group literature, contact your local sales office.

To comment on Motorola hardware, software, or system products, contact:
Motorola Field Service Operations
Customer Support Center
1-800-551-1016

(P) MVME187PG/D3

70568 PRINTED IN USA 5-94 IRONWOOD LITHO 500