MVME197PG/D2A1
May 1994

Supplement to

MVME197LE, MVME197DP, and MVME197SP

Single Board Computers

Programmer's Reference Guide

(MVME197PG/D2)

The attached pages are replacements and/or additions to the programmer's guide. They correct minor errors, update some features, and add information on revision 1 of the VMEchip2 ASIC.

Please replace the pages according to the following table and place this page behind the title page of the programmer's guide as a record of this change:

Replace Old	With New	Add New
v/vi,	v/vi,	
xiii/xiv,	xiii/xiv,	
1-1 through 1-4,	1-1 through 1-4,	
1-9 through 1-12,	1-9 through 1-12,	
1-25 through 1-28,	1-25 through 1-28,	
2-5/2-6,	2-5/2-6,	
2-17 through 2-20,	2-17 through 2-20,	
2-25 through 2-30,	2-25 through 2-30,	
2-41 through 2-46,	2-41 through 2-46,	
2-91/2-92,	2-91/2-92,	2-92.a/2-92.b
IN-7/IN-8,	IN-7/IN-8,	
IN-13/IN-14	IN-13/IN-14	

- A vertical bar (|) in the margin of a replacement page indicates a text change or addition.
- The supplement number is shown at the bottom of each replacement page.

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.

Restricted Rights Legend

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

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

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

Related Documentation

The following publications are applicable to the MVME197 module series and may provide additional helpful information. If not shipped with this product, they may be purchased by contacting your Motorola sales office.

Document Title	Motorola Publication Number
MVME197LE Single Board Computer User's Manual	MVME197LE
MVME197LE Single Board Computer Support Information	SIMVME197LE
MVME197DP and MVME197SP Single Board Computer User's Manual	MVME197
MVME197DP and MVME197SP Single Board Computer Support Information	SIMVME197
MVME197BUG 197Bug Debugging Package User's Manual	MVME197BUG
MVME197BUG 197Bug Diagnostic Firmware User's Manual	MVME197DIAG
MVME712M Transition Module and P2 Adapter Board User's Manual	MVME712M
MVME712-12, MVME712-13, MVME712A, MVME712AM, and MVME712B Transition Module and LCP2 Adapter Board User's Manual	MVME712A
MC68040/MC68EC040/MC68CL040 Microprocessor User's Manual	M68040UM/AD
MC88110 Second Generation RISC Microprocessor User's Manual	MC88110UM/AD
MC88410 Secondary Cache Controller User's Manual	MC88410UM/AD

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 the manual but has a suffix such as "/A1" (the first supplement to the manual).**

To further assist your development effort, Motorola has collected user's manuals for each of the peripheral controllers used on the MVME197 module series and other boards from the suppliers. This bundle includes manuals for the following:

68-1X7DS for use with the MVME197 series of Single Board Computers.

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

The following non-Motorola publications may also be of interest and may be obtained from the sources indicated. The VMEbus Specification is contained in ANSI/IEEE Standard 1014-1987.

ANSI/IEEE Std 1014-1987 Versatile Backplane Bus: VMEbus	The Institute of Electrical and Electronics Engineers, Incorporated Publication and Sales Department 345 East 47th Street New York, New York 10017-2633 Telephone: 1-800-678-4333
ANSI Small Computer System Interface-2 (SCSI-2), Draft Document X3.131-198X, Revision 10c	Global Engineering Documents P.O. Box 19539 Irvine, California 92713-9539 Telephone (714) 979-8135

CHAPTER 3 PCCchip2

BOARD LEVEL HARDWARE DESCRIPTION 1

Introduction

This manual provides programming information for the MVME197 series of Single Board Computers. Extensive programming information is provided for the Application-Specific Integrated Circuit (ASIC) devices used on the MVME197 module series. Refer to Table 1-1 for model designations.

This chapter describes the board level hardware features of the MVME197 series of single board computers, hereafter referred to as the MVME197, unless separately specified. The chapter is organized with a board level overview and features listed 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.

All programmable registers in the MVME197 reside in ASIC devices and are covered in device specific chapters. Chapter 2 covers the VMEchip2 (VME2), Chapter 3 covers the PCCchip2 (PCC2), Chapter 4 covers the BusSwitch gate array, Chapter 5 covers the DCAM (DRAM Controller and Address Multiplexer), and Chapter 6 covers the ECDM (Error Correction and Data Multiplexer). 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.

Model Designations

The MVME197 Single Board Computer is avai lable in the following models.

Table 1-0. MVME197 Model Designations

Model Number	Motorola Part Number	Model Description
MVME197LE series		
MVME197-001	01-W3869B03L	Single Processor MC88110, 32MB Onboard ECC DRAM, 50 MHz
MVME197-002	01-W3869B04L	Single Processor MC88110, 64MB Onboard ECC DRAM, 50 MHz

Table 1-1. MVME197 Model Designations (Continued)

Model Number	Motorola Part Number	Model Description
MVME197LE series		
MVME197-003	01-W3869B01M	Single Processor MC88110, 32MB Onboard ECC DRAM, 40 MHz
MVME197-004	01-W3869B02M	Single Processor MC88110, 64MB Onboard ECC DRAM, 40 MHz
MVME197SP series		
MVME197-101	01-W3815B04	Single Processor MC88110, 256KB Cache, 128MB Onboard ECC DRAM, 50 MHz
MVME197DP series		
MVME197-201	01-W3815B03	Dual Processor MC88110, 256KB Cache per Processor, 128MB Onboard ECC DRAM, 50 MHz

Overview

Each MVME197 module is a double-high VMEmodule based on the MC88110 RISC microprocessor.

The **MVME197LE** series have 32/64MB of DRAM, 1MB of flash memory, 8KB of static RAM (with battery backup), a time of day clock (with battery backup), an Ethernet transceiver interface, four serial ports with EIA-232-D interface, six tick timers, a watchdog timer, 128/256KB of BOOT ROM, a SCSI bus interface with DMA (Direct Memory Access), a Centronics printer port, an A16/A24/A32/D8/D16/D32 VMEbus master/slave interface, and a VMEbus system controller.

The **MVME197DP/SP** series have 128/256MB of DRAM, 256KB of cache memory for **each** MC88110/MC88410 microprocessor/cache controller combination (note that the MVME197SP version has only one MC88110/MC88410 device combination), 4MB of flash memory, 8KB of static RAM (with battery backup), a time of day clock (with battery backup), an Ethernet transceiver interface, four serial ports with EIA-232-D interface, six tick timers, a watchdog timer, 256KB of BOOT ROM, a SCSI bus interface with DMA (Direct Memory Access), a Centronics printer port, an A16/A24/A32/D8/D16/D32 VMEbus master/slave interface, and a VMEbus system controller.

Input/Output (I/O) signals are routed through the MVME197's backplane connector P2. A P2 Adapter Board or LCP2 Adapter board routes the signals and grounds from connector P2 to an MVME712 series transition module. The MVME197 supports the MVME712M, MVME712A, MVME712AM, and MVME712B transition boards (referred to here as the MVME712X, unless separately specified). The MVME197 also supports the MVME712-12 and MVME712-13 (referred to as the MVME712-XX, unless separately specified). These transition boards provide configuration headers, serial port drivers, and industry standard connectors for the I/O devices.

All MVME197 modules have eight ASIC devices (described in the following order: BusSwitch, DCAM, ECDM, PCC2, and VME2).

Note **For the MVME197 series, the term Local Bus, as used in other MVME1xx Single Board Computer series, is referred to as the Local Peripheral Bus.**

The BusSwitch ASIC provides an interface between the processor bus (MC88110 bus) and the local peripheral bus (MC68040 compatible bus). Refer to the board specific MVME197 block diagram (Figures 1-1, 1-2, and 1-3). It provides bus arbitration for the MC88110 bus and serves as a seven level interrupt handler. It has programmable map decoders for both busses, as well as write post buffers on each, two tick timers, and four 32-bit general purpose registers.

The DCAM (DRAM Controller and Address Multiplexer) ASIC provides the address multiplexers and RAS/CAS/WRITE control for the DRAM as well as data control for the ECDM.

The ECDM (Error Correction and Data Multiplexer) ASIC multiplexes between four data paths on the DRAM array. Since the device handles 16 bits, four such devices are required on the MVME197 to accommodate the 64-bit data bus of the MC88110 microprocessor. Single-bit error correction and double-bit detection is performed in the ECDM.

The PCCchip2 (Peripheral Channel Controller) ASIC provides two tick timers and the interface to the LAN chip, the SCSI chip, the serial port chip, the printer port, and the BBRAM (Battery Backup RAM).

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

Local peripheral bus to VMEbus transfers can be D8, D16, or D32. VMEchip2 DMA transfers to the VMEbus, however, can be 64 bits wide as Block Transfer (BLT).

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

Features

These are some of the major features of the MVME197 module series:

- ❏ Single MC88110 RISC Microprocessor (MVME197LE modules)
- ❏ Single MC88110 RISC Microprocessor with an MC88410 Cache Controller (MVME197SP modules)
- ❏ Dual MC88110 RISC Microprocessors, each processor with one MC88410 Cache Controller (MVME197DP modules)
- ❏ 256 kilobytes of external cache per processor, controlled by the MC88410 Cache Controller (MVME197DP and MVME197SP modules)
- ❏ 32 or 64 megabytes of 64-bit Dynamic Random Access Memory (DRAM) with Error Checking and Correction (ECC) (MVME197LE modules)
- ❏ 128 megabytes of 64-bit Dynamic Random Access Memory (DRAM) with Error Checking and Correction (ECC) (MVME197DP and MVME197SP modules)
- ❏ 4 megabytes of Flash memory
- ❏ Six status LEDs (FAIL, RUN, SCON, LAN, SCSI, and VME)
- ❏ 8 kilobytes of Static Random Access Memory (SRAM) and Time of Day (TOD) clock with battery backup RAM (BBRAM)
- ❏ Two push-button switches (ABORT and RESET)
- ❏ 256 kilobytes of BOOT ROM
- ❏ Six 32-bit tick timers for periodic interrupts
- ❏ Watchdog timer
- ❏ Eight software interrupts

The ABORT switch (S2) can generate an interrupt to CPU0 via the NMI* signal. It is normally used to abort program execution and return to the debugger. This capability is controlled via the ABORT register in the BusSwitch.

The six LEDs on the MVME197 front panel are: FAIL, SCON, RUN, LAN, VME, and SCSI.

The yellow FAIL LED (DS1) is lit when the BRDFAIL signal line is active.

The green SCON LED (DS2) is lit when the VMEchip2 is the VMEbus system controller.

The green RUN LED (DS3) is lit when the MC88110 bus MC* pin is low.

The green LAN LED (DS4) lights when the LAN chip is the local peripheral bus master.

The green VME LED (DS5) lights when the board is using the VMEbus or when the board is accessed by the VMEbus.

The green SCSI LED (DS6) lights when the SCSI chip is the local peripheral bus master.

Data Bus Structure

The data bus structure is arranged to accommodate the various 8-bit, 16-bit, 32-bit, and 64-bit devices that reside on the module. 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.

MC88110 MPU

The MVME197 series of single board computers are based on the MC88000 families of RISC (Reduced Instruction Set Computing) microprocessors. Depending on the specific MVME197 module, the MVME197 series uses the MC88110 RISC microprocessor. Refer to the *Module Designation* section in the beginning of this chapter for MVME197 module/processor variations and to the *MC88110 Second Generation RISC Microprocessor User's Manual* for more detailed information on this device.

MC88410 Cache Controller

(The following text is applicable only to the MVME197DP/SP versions of the MVME197 series).

Depending on the specific MVME197DP/SP module version, each MC88110 microprocessor is connected directly to an MC88410 Secondary Cache

Controller. Each MC88410 controls a 256KB level two cache. Refer to the *MC88410 Secondary Cache Controller User's Manual* and the *MCM62110 Data Sheet* for more information on this device.

BOOT ROM

The board accommodates a 32-pin PLCC/CLCC ROM/EPROM referred to as BOOT ROM or DROM (Download ROM). It is organized as a 256K x 8 device, but as viewed from the processor it looks like a 32K x 64 memory. This memory is mapped starting at location $FFF80000, but after a local reset it is also mapped at location 0, providing a reset vector and bootstrap code for the processor. The DR0 bit in the General Control Register (GCR) of the PCCchip2 must be cleared to disable the BOOT ROM memory map at 0. In addition, the ROM0 bit in the ROMCR register of the BusSwitch must be cleared.

Flash Memory

4MB of flash memory is available on the board. Flash memory works like EPROM, but can be erased and reprogrammed by software. It is organized as 32 bits wide, but to the processor it looks as 64 bits wide. It is mapped at location $FF800000. Reads can be of any size, including burst transfers, but writes are always 32 bits wide, regardless of the size specified for the transfer. For this reason, software should only use 32-bit write transfers. This memory is controlled by the BusSwitch, and the memory size, access time, and write enable capability can be programmed via the ROM Control Register (ROMCR) in the BusSwitch. The flash memory can be accessed from the processor bus only. It is not accessible from the local peripheral bus or VMEbus.

Onboard DRAM

The onboard DRAM on the MVME197LE (2 banks of 32MB memory, one optionally installed) is sized at 32MB using 1M x 4 devices and configured as 256 bits wide.

The onboard DRAM on the MVME197DP/SP (2 banks of 128MB memory, one optionally installed) is sized at 128MB using 4M x 4 devices and configured as 256 bits wide.

The DRAM is four-way interleaved to efficiently support cache burst cycles. The DRAM is controlled by the DCAM and ECDM, and the map decoders in the DCAM can be programmed through the I^2C bus interface in the ECDM to accommodate different base address(es) and sizes. The onboard DRAM is not disabled by a local peripheral bus reset. Refer to the *DCAM* and *ECDM* chapters for detailed programming information.

Battery Backup RAM and Clock

The MK48T08 RAM and clock chip is used on the MVME197. 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* chapter and to the MK48T08 data sheet for detailed programming information.

VMEbus Interface

The local peripheral bus to VMEbus interface, the VMEbus to local peripheral bus interface, and the local-VMEbus DMA controller functions on the MVME197 are provided by the VMEchip2. The VMEchip2 can also provide the VMEbus system controller functions. Refer to the *VMEchip2* chapter for detailed programming information.

I/O Interfaces

The MVME197 provides onboard I/O for many system applications. The I/O functions include serial ports, a printer port, an Ethernet transceiver interface, and a 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). Serial port 4 also supports synchronous modes of operation.

All four of the serial ports are different functionally because of the limited number of pins on the 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 64k bits per second. It uses RXD, CTS, DCD, RTS, and DTR. It also interfaces to the synchronous clock signal lines. Refer to the *Printer and Serial Port Connections* chapter for drawings of the serial port interface connections.

All four serial ports 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 MVME712X transition board. An external I/O transition board such as the MVME712X should be used to convert the I/O connector pinout to industry-standard connectors.

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* chapter for detailed programming information.

The CD2401 supports DMA operations to local memory. Because the CD2401 does not support a retry operation necessary to break VMEbus lock conditions, the CD2401 DMA controllers should not be programmed to access the VMEbus. The hardware does not restrict the CD2401 to onboard DRAM.

Printer Interface

The MVME197 has a Centronics-compatible printer interface. The printer interface is provided by the PCCchip2. Refer to the *PCCchip2* chapter for detailed programming information. Refer to *Printer and Serial Port Connections* chapter for drawings of the serial 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. Therefore, the 82596CA should not be programmed to access the VMEbus.

Every MVME197 module is assigned an Ethernet Station Address. This address is $08003E2XXXXX, where XXXXX is the unique 5-nibble number assigned to the board (i.e., every MVME197 has a different value for XXXXX).

The Ethernet Station Address is displayed on a label attached to the VMEbus P2 connector. In addition, the eight bytes including the Ethernet address are stored in the configuration area of the BBRAM, with the two lower bytes of those set to 0. That is, 08003E2XXXXX0000 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 four bytes (XXXX0000) can be read. Refer to the BBRAM, TOD Clock memory map description later in this chapter. The MVME197 debugger has the capability to retrieve or set the Ethernet address.

Table 1-0. VMEchip2 Memory Map
(Sheet 1 of 4)

VMEchip2 LCSR Base Address = $FFF40000

OFFSET:

	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
00	VMEbus SLAVE ENDING ADDRESS 1																VMEbus SLAVE STARTING ADDRESS 1															
04	VMEbus SLAVE ENDING ADDRESS 2																VMEbus SLAVE STARTING ADDRESS 2															
08	VMEbus SLAVE ADDRESS TRANSLATION ADDRESS 1																VMEbus SLAVE ADDRESS TRANSLATION SELECT 1															
0C	VMEbus SLAVE ADDRESS TRANSLATION ADDRESS 2																VMEbus SLAVE ADDRESS TRANSLATION SELECT 2															
10					(VB) ADDR 2	(VB) SNP 2		(VB) WP 2	(VB) SUP 2	(VB) USR 2	(VB) A32 2	(VB) A24 2	(VB) D64 2	(VB) BLK 2	(VB) PGM 2	(VB) DAT 2					(VB) ADDR 1	(VB) SNP 1		(VB) WP 1	(VB) SUP 1	(VB) USR 1	(VB) A32 1	(VB) A24 1	(VB) D64 1	(VB) BLK 1	(VB) PGM 1	(VB) DAT 1
14	LOCAL BUS SLAVE ENDING ADDRESS 1																LOCAL BUS SLAVE STARTING ADDRESS 1															
18	LOCAL BUS SLAVE ENDING ADDRESS 2																LOCAL BUS SLAVE STARTING ADDRESS 2															
1C	LOCAL BUS SLAVE ENDING ADDRESS 3																LOCAL BUS SLAVE STARTING ADDRESS 3															
20	LOCAL BUS SLAVE ENDING ADDRESS 4																LOCAL BUS SLAVE STARTING ADDRESS 4															
24	LOCAL BUS SLAVE ADDRESS TRANSLATION ADDRESS 4																LOCAL BUS SLAVE ADDRESS TRANSLATION SELECT 4															
28	(LB) D16 EN	(LB) WP EN	(LB) AM 4						(LB) D16 EN	(LB) WP EN	(LB) AM 3						(LB) D16 EN	(LB) WP EN	(LB) AM 2						(LB) D16 EN	(LB) WP EN	(LB) AM 1					
2C	(VB) GCSR GROUP ADDRESS								(VB) GCSR BOARD ADDRESS				LB EN4	LB EN3	LB EN2	LB EN1	LB I2 EN	LB I2 WP	LB I2 SU	LB I2 PD	LB I1 EN	LB I1 D16	LB I1 WP	LB I1 SU	ROM SIZE (XX)		ROM BANK B SPEED (XX)			ROM BANK A SPEED (XX)		
	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

LB = Local Bus

(LB) = Local Bus Slave

LV = Local Bus to VMEbus

VB = VMEbus

(VB) = VMEbus Slave

(XX) = Not Used on the MVME197 Series

Table 1-7. VMEchip2 Memory Map (Continued)
(Sheet 2 of 4)

VMEchip2 LCSR Base Address = $FFF40000
OFFSET:

	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
30											WAIT RMW (XX)	ROM0 (XX)	DMAC TB SNP MODE (XX)		SPRAM SPEED (XX)		LV ROBN MODE	LV DHB	LV DWB		LV FAIR	LV RWD	LV REQUEST LEVEL		DMAC HALT	DMAC EN	DMAC TBL	DMAC FAIR	DMAC RELM		DMAC REQUEST LEVEL	
34																	DMAC INTE	DMAC LB SNP MODE			DMAC VME INC	DMAC LB INC	DMAC TVME	DMAC D16	DMAC BLK TR	DMAC BLK TR	DMAC AM	DMAC AM	DMAC AM	DMAC AM	DMAC AM	DMAC AM
38	DMAC LOCAL BUS ADDRESS COUNTER																															
3C	DMAC VMEbus ADDRESS COUNTER																															
40	DMAC BYTE COUNTER																															
44	DMAC TABLE ADDRESS COUNTER																															
48		INTERRUPT 1 SIGNAL		IRQ CLR	IRQ STAT	VMEbus INTERRUPT LEVEL			VMEbus INTERRUPT VECTOR								DMAC INTERRUPT COUNTER				MPU CLR	MPU LB EN	MPU LB PERR	MPU LB OB	MPU LB TO	DMAC LB EN	DMAC LB PERR	DMAC LB OB	DMAC LB TO	DMAC LB ERR	DMAC VME ERR	DMAC DONE
4C								VB TO	DMAC TIME OFF			DMAC TIME ON			VMEbus GLOBAL TIMEOUT		VMEbus ACCESS TIMEOUT	LOCAL BUS TIMEOUT			WATCHDOG TIMEOUT				PRESCALER ADJUST								
50	TICK TIMER 1 COMPARE																															
54	TICK TIMER 1 COUNTER																															
58	TICK TIMER 2 COMPARE																															
5C	TICK TIMER 2 COUNTER																															
60		SCON	SYS FAIL	BRD FAIL STAT	PURS STAT	CLR PURS STAT	BRD FAIL OUT	RST SW EN	SYS RST	WD CLR STAT	WD CLR CNT	WD TO STAT	WD BDF EN	WD SYS LRST	WD RST EN	WD EN	OVERFLOW COUNTER 2				CLR OVF 2	COC 2	CNT EN 2	OVERFLOW COUNTER 1						CLR OVF 1	COC 1	CNT EN 1
64	PRESCALER COUNTER																															
	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

LB = Local Bus
(LB) = Local Bus Slave
LV = Local Bus to VMEbus
VB = VMEbus
(VB) = VMEbus Slave
(XX) = Not Used on the MVME197 Series

Table 1-7. VMEchip2 Memory Map (Continued)
(Sheet 3 of 4)

VMEchip2 LCSR Base Address = $FFF40000

OFFSET:

	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
68	AC FAIL IRQ	AB SW IRQ	SYS FAIL IRQ	MWP ERR IRQ	PE IRQ	IRQ1 EDGE IRQ	TIC TIM2 IRQ	TIC TIM1 IRQ	VME IACK IRQ	DMAC IRQ	GCSR SIG3 IRQ	GCSR SIG2 IRQ	GCSR SIG1 IRQ	GCSR SIG0 IRQ	GCSR LM1 IRQ	GCSR LM0 IRQ	LB SW7 IRQ	LB SW6 IRQ	LB SW5 IRQ	LB SW4 IRQ	LB SW3 IRQ	LB SW2 IRQ	LB SW1 IRQ	LB SW0 IRQ	SPARE	VME IRQ7 IRQ	VME IRQ6 IRQ	VME IRQ5 IRQ	VME IRQ4 IRQ	VME IRQ3 IRQ	VME IRQ2 IRQ	VME IRQ1 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	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
70																	SET IRQ 15	SET IRQ 14	SET IRQ 13	SET IRQ 12	SET IRQ 11	SET IRQ 10	SET IRQ 9	SET IRQ 8								
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	CLR IRQ 15	CLR IRQ 14	CLR IRQ 13	CLR IRQ 12	CLR IRQ 11	CLR IRQ 10	CLR IRQ 9	CLR IRQ 8								
78		ACFAIL IRQ LEVEL				ABORT IRQ LEVEL				SYSFAIL IRQ LEVEL				MASTER WRITE POST ERROR IRQ LEVEL				PARITY ERROR IRQ LEVEL				IRQ1 EDGE-SENSITIVE IRQ LEVEL				TICK TIMER 2 IRQ LEVEL				TICK TIMER 1 IRQ LEVEL		
7C		VMEbus ACKNOWLEDGE IRQ LEVEL				DMAC IRQ LEVEL				GCSR SIG 3 IRQ LEVEL				GCSR SIG 2 IRQ LEVEL				GCSR SIG 1 IRQ LEVEL				GCSR SIG 0 IRQ LEVEL				GCSR LM 1 IRQ LEVEL				GCSR LM 0 IRQ LEVEL		
80		SW7 IRQ LEVEL				SW6 IRQ LEVEL				SW5 IRQ LEVEL				SW4 IRQ LEVEL				SW3 IRQ LEVEL				SW2 IRQ LEVEL				SW1 IRQ LEVEL				SW0 IRQ LEVEL		
84		SPARE IRQ LEVEL				VMEbus IRQ7 IRQ LEVEL				VMEbus IRQ6 IRQ LEVEL				VMEbus IRQ5 IRQ LEVEL				VMEbus IRQ4 IRQ LEVEL				VMEbus IRQ3 IRQ LEVEL				VMEbus IRQ2 IRQ LEVEL				VMEbus IRQ1 IRQ LEVEL		
88	VECTOR BASE REGISTER 0				VECTOR BASE REGISTER 1				MST IRQ EN	SYS FAIL LEVEL	AC FAIL LEVEL	ABORT LEVEL	GENERAL PURPOSE I/O ENABLE				GENERAL PURPOSE I/O OUTPUT				GENERAL PURPOSE I/O INPUT				GENERAL PURPOSE INPUT							
8C																									MP IRQEN (XX)	REVE ROM (XX)	DIS SRAM (XX)	DIS MST	NOEL BBSY	DIS BSYT	EN INT	DIS BGN
	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

LB = Local Bus

(LB) = Local Bus Slave

LV = Local Bus to VMEbus

VB = VMEbus

(VB) = VMEbus Slave

(XX) = Not Used on the MVME197 Series

Table 1-7. VMEchip2 Memory Map (Continued)
(Sheet 4 of 4)

VMEchip2 GCSR Base Address = $FFF40100

L	V	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	CHIP REVISION								CHIP ID							
4	2	LM3	LM2	LM1	LM0	SIG3	SIG2	SIG1	SIG0	RST	ISF	BF	SCON	SYS FL			
8	4	GENERAL PURPOSE CONTROL AND STATUS REGISTER 0															
C	6	GENERAL PURPOSE CONTROL AND STATUS REGISTER 1															
10	8	GENERAL PURPOSE CONTROL AND STATUS REGISTER 2															
14	A	GENERAL PURPOSE CONTROL AND STATUS REGISTER 3															
18	C	GENERAL PURPOSE CONTROL AND STATUS REGISTER 4															
1C	E	GENERAL PURPOSE CONTROL AND STATUS REGISTER 5															
		D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

NOTES: L = Local Bus Offset.
V = VMEbus Offset.

Local Peripheral Bus to VMEbus Interface

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

Using programmable map decoders with programmable attribute bits, the local peripheral 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 peripheral bus slave includes six local peripheral 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 peripheral 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 peripheral bus to be replaced by bits from the alternate address register. The feature allows the local peripheral 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 peripheral bus addresses $FFFF0000 through $FFFFFFFF as the short I/O A16/D16 or A16/D32 area. The second I/O map decoder 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 peripheral bus address and data and then acknowledges the local peripheral bus master. The local peripheral 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 4-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 master is interrupted. The address of the error is not saved. Normal memory never returns a bus error on a write cycle. However, some 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 peripheral 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 device on the local peripheral bus 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 4-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 peripheral 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 peripheral 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 peripheral bus error signal after 64 µsecs, 1 msec, or 32 msecs.

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 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 peripheral 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 peripheral 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:

1. R - This bit is a read-only status bit.
2. R/W - This bit is readable and writable.
3. W/AC - This bit can be set and it is automatically cleared. This bit can also be read.
4. C - Writing a one to this bit clears this bit or another bit. This bit reads zero.
5. 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 defined below.

1. P - The bit is affected by power-up reset.
2. S - The bit is affected by SYSRESET.
3. L - The bit is affected by local reset.
4. X - The bit is not affected by reset.

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

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

VMEchip2 LCSR Base Address = $FFF40000

OFFSET:

	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
00	VMEbus SLAVE ENDING ADDRESS 1																VMEbus SLAVE STARTING ADDRESS 1															
04	VMEbus SLAVE ENDING ADDRESS 2																VMEbus SLAVE STARTING ADDRESS 2															
08	VMEbus SLAVE ADDRESS TRANSLATION ADDRESS 1																VMEbus SLAVE ADDRESS TRANSLATION SELECT 1															
0C	VMEbus SLAVE ADDRESS TRANSLATION ADDRESS 2																VMEbus SLAVE ADDRESS TRANSLATION SELECT 2															
10					(VB) ADDR 2	(VB) SNP 2		(VB) WP 2	(VB) SUP 2	(VB) USR 2	(VB) A32 2	(VB) A24 2	(VB) D64 2	(VB) BLK 2	(VB) PGM 2	(VB) DAT 2					(VB) ADDR 1	(VB) SNP 1		(VB) WP 1	(VB) SUP 1	(VB) USR 1	(VB) A32 1	(VB) A24 1	(VB) D64 1	(VB) BLK 1	(VB) PGM 1	(VB) DAT 1
14	LOCAL BUS SLAVE ENDING ADDRESS 1																LOCAL BUS SLAVE STARTING ADDRESS 1															
18	LOCAL BUS SLAVE ENDING ADDRESS 2																LOCAL BUS SLAVE STARTING ADDRESS 2															
1C	LOCAL BUS SLAVE ENDING ADDRESS 3																LOCAL BUS SLAVE STARTING ADDRESS 3															
20	LOCAL BUS SLAVE ENDING ADDRESS 4																LOCAL BUS SLAVE STARTING ADDRESS 4															
24	LOCAL BUS SLAVE ADDRESS TRANSLATION ADDRESS 4																LOCAL BUS SLAVE ADDRESS TRANSLATION SELECT 4															
28	(LB) D16 EN	(LB) WP EN	(LB) AM 4						(LB) D16 EN	(LB) WP EN	(LB) AM 3						(LB) D16 EN	(LB) WP EN	(LB) AM 2						(LB) D16 EN	(LB) WP EN	(LB) AM 1					
2C	(VB) GCSR GROUP ADDRESS								(VB) GCSR BOARD ADDRESS				LB EN4	LB EN3	LB EN2	LB EN1	LB I2 EN	LB I2 WP	LB I2 SU	LB I2 PD	LB I1 EN	LB I1 D16	LB I1 WP	LB I1 SU	ROM SIZE (XX)		ROM BANK B SPEED (XX)			ROM BANK A SPEED (XX)		
	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

LB = Local Bus

(LB) = Local Bus Slave

LV = Local Bus to VMEbus

VB = VMEbus

(VB) = VMEbus Slave

(XX) = Not Used on the MVME197 Series

Table 2-1. VMEchip2 Memory Map - LCSR Summary (Continued)
(Sheet 2 of 3)

VMEchip2 LCSR Base Address = $FFF40000

OFFSET:

	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
30											WAIT RMW (XX)	ROM0 (XX)	DMAC TB SNP MODE (XX)		SPRAM SPEED (XX)		LV ROBN MODE	LV DHB	LV DWB		LV FAIR	LV RWD	LV REQUEST LEVEL		DMAC HALT	DMAC EN	DMAC TBL	DMAC FAIR	DMAC RELM		DMAC REQUEST LEVEL	
34																	DMAC INTE	DMAC LB SNP MODE			DMAC VME INC	DMAC LB INC	DMAC TVME	DMAC D16	DMAC BLK TR	DMAC BLK TR	DMAC AM	DMAC AM	DMAC AM	DMAC AM	DMAC AM	DMAC AM
38	DMAC LOCAL BUS ADDRESS COUNTER																															
3C	DMAC VMEbus ADDRESS COUNTER																															
40	DMAC BYTE COUNTER																															
44	DMAC TABLE ADDRESS COUNTER																															
48		INTERRUPT 1 SIGNAL		IRQ CLR	IRQ STAT	VMEbus INTERRUPT LEVEL			VMEbus INTERRUPT VECTOR								DMAC INTERRUPT COUNTER				MPU CLR	MPU LB EN	MPU LB PERR	MPU LB OB	MPU LB TO	DMAC LB EN	DMAC LB PERR	DMAC LB OB	DMAC LB TO	DMAC LB ERR	DMAC VME ERR	DMAC DONE
4C								VB TO	DMAC TIME OFF			DMAC TIME ON			VMEbus GLOBAL TIMEOUT		VMEbus ACCESS TIMEOUT	LOCAL BUS TIMEOUT			WATCHDOG TIMEOUT				PRESCALER ADJUST								
50	TICK TIMER 1 COMPARE																															
54	TICK TIMER 1 COUNTER																															
58	TICK TIMER 2 COMPARE																															
5C	TICK TIMER 2 COUNTER																															
60		SCON	SYS FAIL	BRD FAIL STAT	PURS STAT	CLR PURS STAT	BRD FAIL OUT	RST SW EN	SYS RST	WD CLR STAT	WD CLR CNT	WD TO STAT	WD BDF EN	WD SYS LRST	WD RST EN	WD EN	OVERFLOW COUNTER 2				CLR OVF 2	COC 2	CNT EN 2	OVERFLOW COUNTER 1						CLR OVF 1	COC 1	CNT EN 1
64	PRESCALER COUNTER																															
	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

LB = Local Bus
(LB) = Local Bus Slave
LV = Local Bus to VMEbus
VB = VMEbus
(VB) = VMEbus Slave
(XX) = Not Used on the MVME197 Series

Table 2-1. VMEchip2 Memory Map - LCSR Summary (Continued)
(Sheet 3 of 3)

VMEchip2 LCSR Base Address = $FFF40000

OFFSET:

	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
68	AC FAIL IRQ	AB SW IRQ	SYS FAIL IRQ	MWP ERR IRQ	PE IRQ	IRQ1 EDGE IRQ	TIC TIM2 IRQ	TIC TIM1 IRQ	VME IACK IRQ	DMAC IRQ	GCSR SIG3 IRQ	GCSR SIG2 IRQ	GCSR SIG1 IRQ	GCSR SIG0 IRQ	GCSR LM1 IRQ	GCSR LM0 IRQ	LB SW7 IRQ	LB SW6 IRQ	LB SW5 IRQ	LB SW4 IRQ	LB SW3 IRQ	LB SW2 IRQ	LB SW1 IRQ	LB SW0 IRQ	SPARE	VME IRQ7 IRQ	VME IRQ6 IRQ	VME IRQ5 IRQ	VME IRQ4 IRQ	VME IRQ3 IRQ	VME IRQ2 IRQ	VME IRQ1 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	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
70																	SET IRQ 15	SET IRQ 14	SET IRQ 13	SET IRQ 12	SET IRQ 11	SET IRQ 10	SET IRQ 9	SET IRQ 8								
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	CLR IRQ 15	CLR IRQ 14	CLR IRQ 13	CLR IRQ 12	CLR IRQ 11	CLR IRQ 10	CLR IRQ 9	CLR IRQ 8								
78		ACFAIL IRQ LEVEL				ABORT IRQ LEVEL				SYSFAIL IRQ LEVEL				MASTER WRITE POST ERROR IRQ LEVEL				PARITY ERROR IRQ LEVEL				IRQ1 EDGE-SENSITIVE IRQ LEVEL				TICK TIMER 2 IRQ LEVEL				TICK TIMER 1 IRQ LEVEL		
7C		VMEbus ACKNOWLEDGE IRQ LEVEL				DMAC IRQ LEVEL				GCSR SIG 3 IRQ LEVEL				GCSR SIG 2 IRQ LEVEL				GCSR SIG 1 IRQ LEVEL				GCSR SIG 0 IRQ LEVEL				GCSR LM 1 IRQ LEVEL				GCSR LM 0 IRQ LEVEL		
80		SW7 IRQ LEVEL				SW6 IRQ LEVEL				SW5 IRQ LEVEL				SW4 IRQ LEVEL				SW3 IRQ LEVEL				SW2 IRQ LEVEL				SW1 IRQ LEVEL				SW0 IRQ LEVEL		
84		SPARE IRQ LEVEL				VMEbus IRQ7 IRQ LEVEL				VMEbus IRQ6 IRQ LEVEL				VMEbus IRQ5 IRQ LEVEL				VMEbus IRQ4 IRQ LEVEL				VMEbus IRQ3 IRQ LEVEL				VMEbus IRQ2 IRQ LEVEL				VMEbus IRQ1 IRQ LEVEL		
88	VECTOR BASE REGISTER 0				VECTOR BASE REGISTER 1				MST IRQ EN	SYS FAIL LEVEL	AC FAIL LEVEL	ABORT LEVEL	GENERAL PURPOSE I/O ENABLE				GENERAL PURPOSE I/O OUTPUT				GENERAL PURPOSE I/O INPUT			GENERAL PURPOSE INPUT								
8C																									MP IRQEN (XX)	REVE ROM (XX)	DIS SRAM (XX)	DIS MST	NOEL BBSY	DIS BSYT	EN INT	DIS BGN
	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

LB = Local Bus

(LB) = Local Bus Slave

LV = Local Bus to VMEbus

VB = VMEbus

(VB) = VMEbus Slave

(XX) = Not Used on the MVME197 Series

VMEbus Slave Address Translation Select Register 2

REG	VMEbus Slave Address Translation Select Register 2															
ADR/SIZ	$FFF4000C (16 bits of 32)															
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIELD	Slave Address Translation Select 2															
OPER	R/W															
RESET	0 PS															

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

VMEbus Slave Write Post and Snoop Control Register 2

REG	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
FIELD					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 peripheral bus map decoder.

WP2 VMEbus Write Post 2. 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 VMEbus Snoop 2. (This feature is not applicable to any of the MVME197 module series).

ADDER2 VMEbus Adder 2. 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

REG	VMEbus Slave Address Modifier Select Register 2							
ADR/SIZ	$FFF40010 (8 bits of 32)							
BIT	23	22	21	20	19	18	17	16
FIELD	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 peripheral 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 VMEbus Data 2. 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 VMEbus Program 2. 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 VMEbus Block 2. 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 VMEbus Block D64 Access 2. 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 VMEbus A24 Access 2. 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 VMEbus A32 Access 2. 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 VMEbus User 2. 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 responded to VMEbus user access cycles.

SUP VMEbus Supervisory 2. 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

REG	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
FIELD					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 peripheral bus map decoder.

WP1 VMEbus Write Post 1. 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 VMEbus Snoop 1. (This feature is not applicable to any of the MVME197 module series).

ADDER1 VMEbus Adder 1. 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

REG	VMEbus Slave Address Modifier Select Register 1							
ADR/SIZ	$FFF40010 (8 bits of 32)							
BIT	7	6	5	4	3	2	1	0
FIELD	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 peripheral 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 VMEbus Data 1. 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 respond to VMEbus data access cycles.

PGM VMEbus Program 1. 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 VMEbus Block 1. 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 VMEbus Block D64 Access 1. 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 VMEbus A24 Access 1. 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 VMEbus A32 Access 1. 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 VMEbus User 1. 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 responded to VMEbus user access cycles.

SUP VMEbus Supervisory 1. 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 Peripheral Bus to VMEbus Map Decoders

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

The local peripheral bus to VMEbus interface allows onboard local peripheral bus masters access to offboard VMEbus resources. The address of the VMEbus resources as viewed from the local peripheral bus is controlled by the local peripheral bus slave map decoders, which are part of the local peripheral bus to VMEbus interface. Four of the six local peripheral 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 peripheral bus.

Each of the four programmable local peripheral 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 peripheral 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 peripheral 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 peripheral 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 peripheral bus addresses, one with write posting enabled and one with write posting disabled. The address translation registers allow local peripheral 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 peripheral 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 peripheral 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 peripheral 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 peripheral 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 peripheral bus map decoders should not be programmed such that more than one map decoder responds to the same local peripheral 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 peripheral bus address.

Local Peripheral Bus to VMEbus I/O Control Register

REG	Local Peripheral Bus to VMEbus I/O Control Register							
ADR/SIZ	$FFF4002C (8 bits of 32)							
BIT	15	14	13	12	11	10	0	8
FIELD	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 I/O 1 Supervisor. 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 I/O 1 Write Posting. 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 I/O 1 D16 Access. 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 I/O 1 Enable. 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 I/O 2 Program. 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 I/O 2 Supervisor. 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 I/O 2 Write Posting. When this bit is high, write posting is enabled to the local peripheral bus F page. When this bit is low, write posting is disabled to the local peripheral bus F page.

I2EN I/O 2 Enable. 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

REG	ROM Control Register							
ADR/SIZ	$FFF4002C (8 bits of 32)							
BIT	7	6	5	4	3	2	1	0
FIELD	SIZE		BSPD			ASPD		
OPER	R/W		R/W			R/W		
RESET	0 PSL		0 PS			0 PS		

This register is the ROM control register. (This feature is not used on any of the MVME197 module series).

Programming the VMEchip2 DMA Controller

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

The VMEchip2 features a local peripheral bus - VMEbus DMA controller (DMAC). The DMAC has two modes of operation: command chaining, and direct. In the direct mode, the local peripheral 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 peripheral bus interrupter. If the DMAC interrupt is enabled in the local peripheral bus interrupter, the local peripheral bus is interrupted.

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 peripheral bus interrupter. If the DMAC interrupt is enabled in the local peripheral bus interrupter, the local peripheral 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 peripheral bus interrupter. If the DMAC interrupt is enabled in the local peripheral bus interrupter, the local peripheral 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 peripheral bus address counter operation, and the local peripheral bus snoop operation (note that the snoop operation feature is not used on the MVME197 module series). 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 peripheral 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 Peripheral 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.

EPROM Decoder, SRAM and DMAC Control Register

REG	EPROM Decoder, SRAM and DMAC Control Register							
ADR/SIZ	$FFF40030 (8 bits [6 used] of 32)							
BIT	23	22	21	20	19	18	17	16
FIELD			WAIT RMW	ROM0	TBLSC		SRAMS	
OPER			R/W	R/W	R/W		R/W	
RESET			0 PSL	1 PSL	0 PS		0 PS	

Refer to note below. 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 * (bus clocks -1) - 35)

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

(30 * (3-1) -35) = 25 nsecs.

Note **The MVME197 module series do not implement a SRAM and do not use the EPROM decoder. Instead, the MVME197 module series use the BusSwitch to generate Flash EPROM control.**

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

SRAMS	Bus Clocks	Maximum SRAM Access Time at 25 MHz
0	6	165 nanosecond
1	5	125 nanosecond
2	4	85 nanosecond
3	4	45 nanosecond

TBLSC DMAC Table Snoop Mode. These bits control the snoop signal lines on the local peripheral bus when the DMAC is table walking. (This feature is not used on any of the MVME197 module series).

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 ROM 0. When this bit is set to 1, the EPROM decoder responds at $00000000 to $0003FFFF 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. (This feature is not applicable to any of the MVME197 module series).**

WAIT RMW Wait RMW Cycle. (This feature is not used on any of the MVME197 module series).

Local Peripheral Bus to VMEbus Requester Control Register

REG	Local Peripheral Bus to VMEbus Requester Control Register							
ADR/SIZ	$FFF40030 (8 bits [7 used] of 32)							
BIT	15	14	13	12	11	10	0	8
FIELD	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 peripheral bus to VMEbus interface.

LVREQL VMEbus Request Level. 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 re-requested at the old level. The requester always requests the VMEbus at level 3 the first time following a SYSRESET.

0 The request level is 0.

1 The request level is 1.

2 The request level is 2.

3 The request level is 3.

LVRWD Release-When-Done Mode. 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 Fair Mode. 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. When the VMEbus has been acquired, the DHB bit is set.

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

GPIOO0 General Purpose I/O Output 0. 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 General Purpose I/O Output 1. 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 General Purpose I/O Output 2. 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 General Purpose I/O Output 3. 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

1. **The GPIO0 pin on the MVME197 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.**
2. **The GPIO1 pin on the MVME197 is used to control the STS LED* signal, which is at the remote RAL connector. 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.**
3. **The GPIO2 pin on the MVME197 is used to control bus error handling by the 82596CA interface logic. Refer to the *82596CA LAN Controller Interface* section in the *PCCchip2* chapter.**
4. **The GPIO1 and GPIO3 pins on the MVME197 are connected to the remote reset connector J1 pins 16 and 18, respectively.**
5. **On the MVME197, the GPIO3 pin may be used as an input or output.**

I/O Control Register 3

REG	I/O Control Register 3							
ADR/SIZ	$FFF40088 (8 bits of 32)							
BIT	7	6	5	4	3	2	1	0
FIELD	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-GPIO7). On the MVME197, the GPI pins are connected to the general purpose switch S3. (Refer to the *Board Level Hardware Description* chapter for details on selected switch settings).

GPI0 General Purpose Input 0. When this bit is low, pin GPI0 is low. When this bit is high, pin GPI0 is high.

GPI1 General Purpose Input 1. When this bit is low, pin GPI1 is low. When this bit is high, pin GPI1 is high.

GPI2 General Purpose Input 2. When this bit is low, pin GPI2 is low. When this bit is high, pin GPI2 is high.

GPI3 General Purpose Input 3. When this bit is low, pin GPI3 is low. When this bit is high, pin GPI3 is high.

GPI4 General Purpose Input 4. When this bit is low, pin GPI4 is low. When this bit is high, pin GPI4 is high.

GPI5 General Purpose Input 5. When this bit is low, pin GPI5 is low. When this bit is high, pin GPI5 is high.

GPI6 General Purpose Input 6. When this bit is low, pin GPI6 is low. When this bit is high, pin GPI6 is high.

GPI7 General Purpose Input 7. When this bit is low, pin GPI7 is low. When this bit is high, pin GPI7 is high.

Miscellaneous Control Register

REG	Miscellaneous Control Register							
ADR/SIZ	$FFF4008C (8 bits of 32)							
BIT	7	6	5	4	3	2	1	0
FIELD	MPIRQ EN	REVE ROM	DIS SRAM	DIS MST	NOEL BBSY	DIS BSYT	EN INT	DIS BGN
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 Disable VMEbus BGIN Filters. 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 Enable Local Peripheral Bus Interrupt Filters. When this bit is high, the local peripheral bus interrupt filters are enabled. When this bit is low, the local peripheral bus filters are disabled. This bit should not be set.

DISBSYT Disable VMEbus BBSY. When this bit is low, the minimum VMEbus BBSY* time when the local peripheral bus master has been retried off the local peripheral bus is 32 local peripheral bus clocks. When this bit is high, the minimum VMEbus BBSY* time when the local peripheral bus master has been retried off the local peripheral bus is 3 local peripheral bus clocks.

When a local peripheral bus master attempts to access the VMEbus and a VMEbus master attempts to access the local peripheral bus, a deadlock is created. The VMEchip2 detects this condition and requests the local peripheral bus master to give up the local peripheral bus and retry the cycle. This allows the VMEbus master to complete the cycle to the local peripheral 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 nsec) and then releases the VMEbus. If the local master does not return from the retry within this 90 nsec 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 peripheral bus clocks, which allows the local peripheral 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.

NOEL BBSY No Early BBSY Release. 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 Disable VMEchip2 Local Peripheral Bus Master. When this bit is high, the VME LED on the MVME197 is lit when the local peripheral bus reset is asserted or the VMEchip2 is driving local peripheral bus busy. When this bit is low, the VME LED on the MVME197 is lit when the local peripheral bus reset is asserted, the VMEchip2 is driving local peripheral bus busy, or the VMEchip2 is driving the VMEbus address strobe.

DIS SRAM Disable SRAM. (This feature is not used by any of the MVME197 module series).

REVE ROM Reverse EPROM Map Decoder. (This feature is not used by any of the MVME197 module series).

MPIRQ EN Multiple Processor Interrupt Enable. (This feature is not used by any of the MVME197 module series).

M

N

O

P

INDEX