Single Board Computers Programmer's Reference Guide (Part 2 of 2)

VMESBCA2/PG1

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, MVME176, MVME177, and MVME187 Single Board Computers. The information in this manual applies to the single board computers listed in the following table:

MVME166 Models	MVME167 Models	MVME176 Models	MVME177 Models	MVME187 Models
MVME166-011a	MVME167-001a	MVME176-001a	MVME177-001a	MVME187-001a
MVME166-012a	MVME167-002a	MVME176-002a	MVME177-002a	MVME187-002a
MVME166-013a	MVME167-003a	MVME176-003a	MVME177-003a	MVME187-003a
MVME166-014a	MVME167-004a	MVME176-004a	MVME177-004a	MVME187-004a
MVME166-015a	MVME167-031a	MVME176-005a	MVME177-005a	MVME187-023a
MVME166-016a	MVME167-032a	MVME176-006a	MVME177-006a	MVME187-024a
	MVME167-033a		MVME177-011a	MVME187-031a
	MVME167-034a		MVME177-012a	MVME187-032a
	MVME167-035a		MVME177-013a	MVME187-033a
	MVME167-036a		MVME177-014a	MVME187-034a
			MVME177-015a	MVME187-035a
			MVME177-016a	MVME187-036a

The letter “a” in the model number indicates the major revision level.

Notes

This document is bound in two parts. Part 1 (VMESBCA1/PG*x*) contains Chapters 1 through 4. Part 2 (VMESBCA2/PG*x*) contains Chapters 5 through 9.

This manual replaces the *MVME166/167/187 Single Board Computers Programmer’s Reference Guide*, MVME187PG/D3, and its supplement, MVME187PG/D3A1. They are obsolete.

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 *Related Documentation* below.

Related Documentation

The following publications are applicable to the Single Board Computers 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/D
MVME167 Single Board Computer User's Manual	MVME167/D
MVME176 Single Board Computer Installation and Use Manual	VME176A/IH
MVME177 Single Board Computer Installation and Use Manual	VME177A/IH
MVME167Bug Debugging Package User's Manual	MVME167BUG/D
MVME177Bug Diagnostics User's Manual	V177DIAA/UM
Debugging Package for Motorola 68K CISC CPUs User's Manual (Parts 1 and 2)	68KBUG1/D and 68KBUG2/D
MVME187 RISC Single Board Computer User's Manual	MVME187/D
MVME187Bug Debugging Package User's Manual	MVME187BUG/D
Debugging Package for Motorola 88K RISC CPUs User's Manual	88KBUG1/D and 88KBUG2/D
Single Board Computers SCSI Software User's Manual	SBCSCSI/D
MVME712-06/07/09 I/O Distribution Board Set User's Manual	MVME712IO/D
MVME712-10 Transition Module User's Manual	MVME712-10/D
MVME712M Transition Module and P2 Adapter Board User's Manual	MVME712M/D
MVME712-12, MVME712-13, MVME712A, MVME712AM, and MVME712B Transition Modules and LCP2 Adapter Board User's Manual	MVME712A/D

Document Title	Motorola Publication Number
MC88100 RISC Microprocessor User's Manual	MC88100UM
MC88200 Cache/Memory Management Unit (CMMU) User's Manual	MC88200UM
M68040 Microprocessors User's Manual	M68040UM
M68060 Microprocessors User's Manual	M68060UM
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 "/*xx*2" (the second revision of a manual); a supplement bears the same number as a manual but has a suffix such as "/*xx*2A1" (the first supplement to the second edition of the manual).

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.)

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

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-syste Bus of the IEC 821 VMEbus*, IEC 822 VSB, Bureau Central de la Commission Électrotechnique 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, 15 Inverness Way East, Englewood, CO 80112-5704.

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.

DS1643 Nonvolatile Timekeeping RAM, Dallas Semiconductor Data Manual, 4401 South Beltwood Parkway, Dallas, Texas 75244-3292.

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, 1635 Aeroplaza Dr., Colorado Springs, CO 80916.

MK48T08(B)/MK48T18(B) Timekeeper™ *and 8Kx8 Zeropower*™ *RAM* data sheet in *Static RAMs Databook*, order number DBSRAM71; SGS-THOMSON Microelectronics; 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.

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.

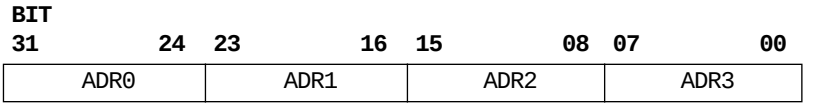
An asterisk (*) following the signal name for signals which are *edge significant* denotes that the actions initiated by that signal occur on high to low transition.

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, MVME176, MVME177, 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, MVME176, MVME177, 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.



The terms *control bit* and *status bit* are used extensively in this document:

control bit	A bit in a register that can be set and cleared under software control.
true	Indicates that a bit is in the state that enables the function it controls
false	Indicates that the bit is in the state that disables the function it controls
status bit	A bit in a register that reflects a specific condition. The status bit can be read by software to determine operational or exception conditions.

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.

WARNING

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

Printed in the United States of America

June 1996

Contents

Chapter 1 Programming Issues

Chapter 2 Hardware Configuration

Chapter 3 Memory Maps

Chapter 4 VMEchip2

Chapter 5 VSBchip2

Chapter 6 PCCchip2

Chapter 7 MEMC040

Chapter 8 MCECC

Figures

Tables

VSBchip2 5

Introduction

This chapter describes the VSB interface chip ASIC (VSBchip2) used only on the MVME166/176 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 Features

This section lists the major features of the VSB interface chip.

- 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.

5

- 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:

- 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 5-1 for a block diagram of the VSBchip2. Detailed descriptions of all registers are provided later in this chapter.

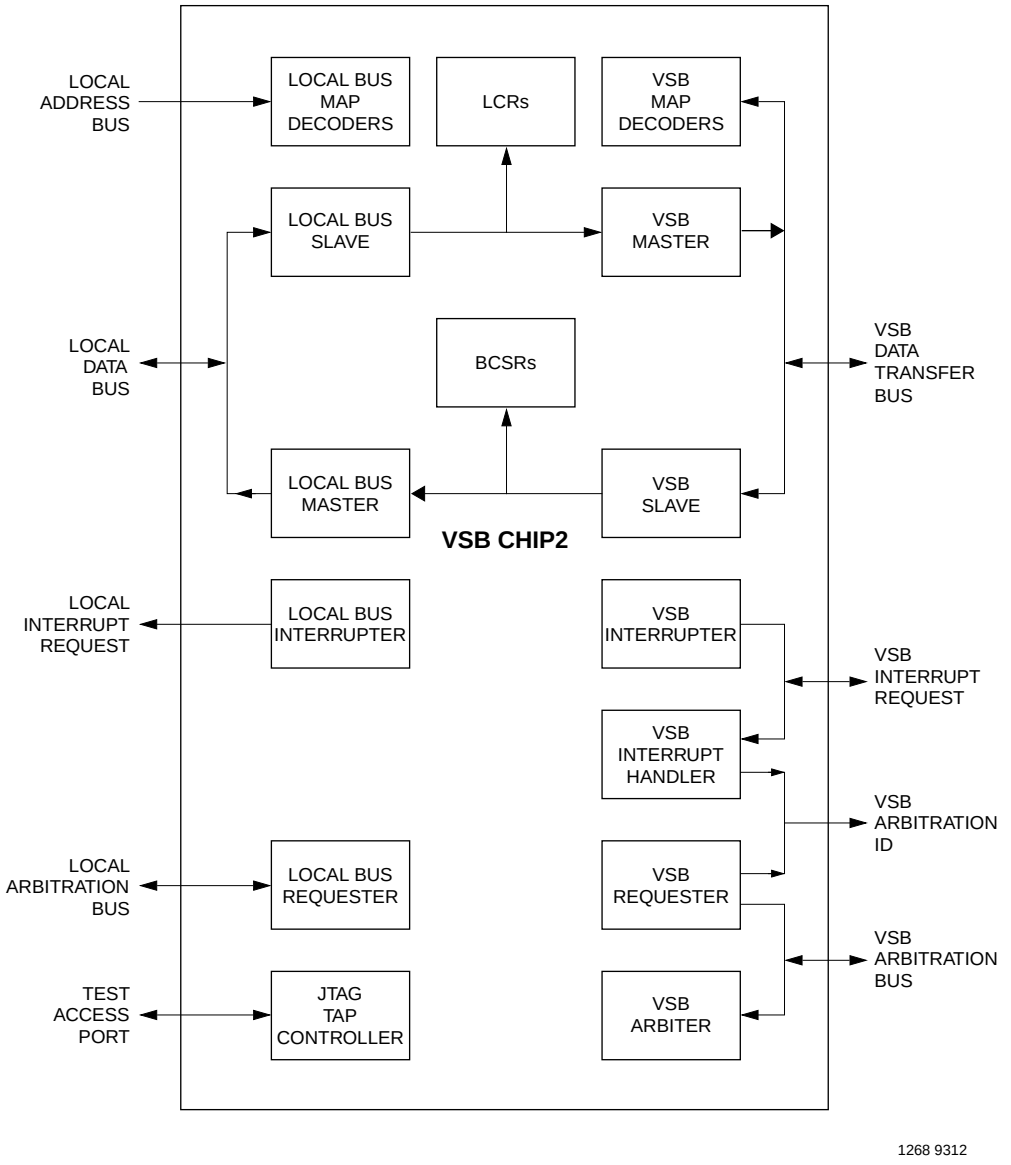


Figure 5-1. VSBchip2 Block Diagram

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 Table 5-1.

Table 5-1. 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 MVME166/176.

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 IDs 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

Note VSB must be implemented using a VSB backplane. A typical VSB backplane presses on the back of the VMEbus backplane behind P2.

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 23 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 5-2 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/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 5-2. 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			
$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) or MVME176BUG (176Bug).

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 166/176Bug).

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	VTXTO	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

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

ADR/SIZ	$FFF41016 (8 bits [0 used] of 32)							
BIT	15	14	13	12	11	10	9	8
NAME	Reserved							
OPER	R							
RESET	0							

ADR/SIZ	$FFF41016 (8 bits of 32)							
BIT	7	6	5	4	3	2	1	0
NAME	Reserved							
OPER	R/W							
RESET	$DF P							

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 30 MHz clock is used, then this register needs to be programmed to $E2 = 226; if a 25 MHz clock is used, then 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 166/176Bug)

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 166/176Bug)

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 166/176Bug)

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 166/176Bug)

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 166/176Bug)

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 166/176Bug)

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 166/176Bug)

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 166/176Bug)

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 166/176Bug)

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 166/176Bug)

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 166/176Bug)

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 166/176Bug)

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 in- crement 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 5-3 shows the memory map of the BCSRs. All registers are accessible through VSB System Address Space.

Table 5-3. 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 MVME166/176.

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	This bit must be 0 on the MVME176/177. Write - Invalidate
1	1	Read - Supply dirty data and mark invalid 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	This bit must be 0 on the MVME176/177. 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.

Introduction

This chapter defines the peripheral channel controller ASIC which is referred to as the PCCchip2 hereafter. The PCCchip2 is designed to interface an MC68040-compatible local bus (Local Bus) to various peripheral devices.

Summary of Major Features

This section lists the major features of the PCCchip2.

- 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 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 6-1.

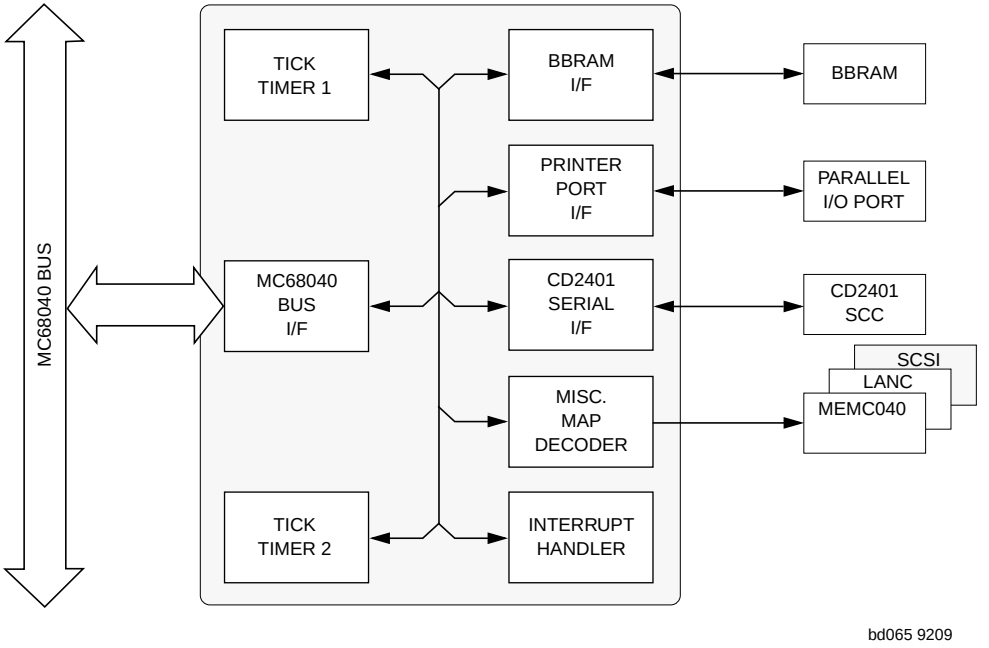


Figure 6-1. PCCchip2 Block Diagram

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 Bus, 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 Local Bus (MC68040-compatible) 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 Local Bus 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 Local Bus (MC68040-bus) signal lines:

- Snoop Control SC1-SC0. (With the value programmed into the LAN Interrupt Control Register.) (Only SC1 is used on the MVME176/177.)
- 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 Single Board Computers 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.

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-compatible Local 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 (MVME187)

When the local MPU is an MC88100, the PCCchip2 provides circuitry to support a seven level, priority, interrupt scheme. This support circuit is enabled and 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 MPU. The priority interrupt scheme is assumed to be provided in the M68000-family MPU 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 Local Bus.

Table 6-1. PCCchip2 Devices Memory Map

Address Range	Selected Device	Comments
$FFF42000-$FFF4203F	PCCchip2 Registers	See Programming Model
$FFF42040-$FFF42FFF	PCCchip2 Registers	Repeated
$FFF43000-$FFF43FFF	MEMC040/MCECC (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
$FFF80000-$FFFBFFFF	Download EPROM (MVME166 only)	External Device
$FFFC0000-$FFFCFFFF	DS1643/MK48T18 (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.

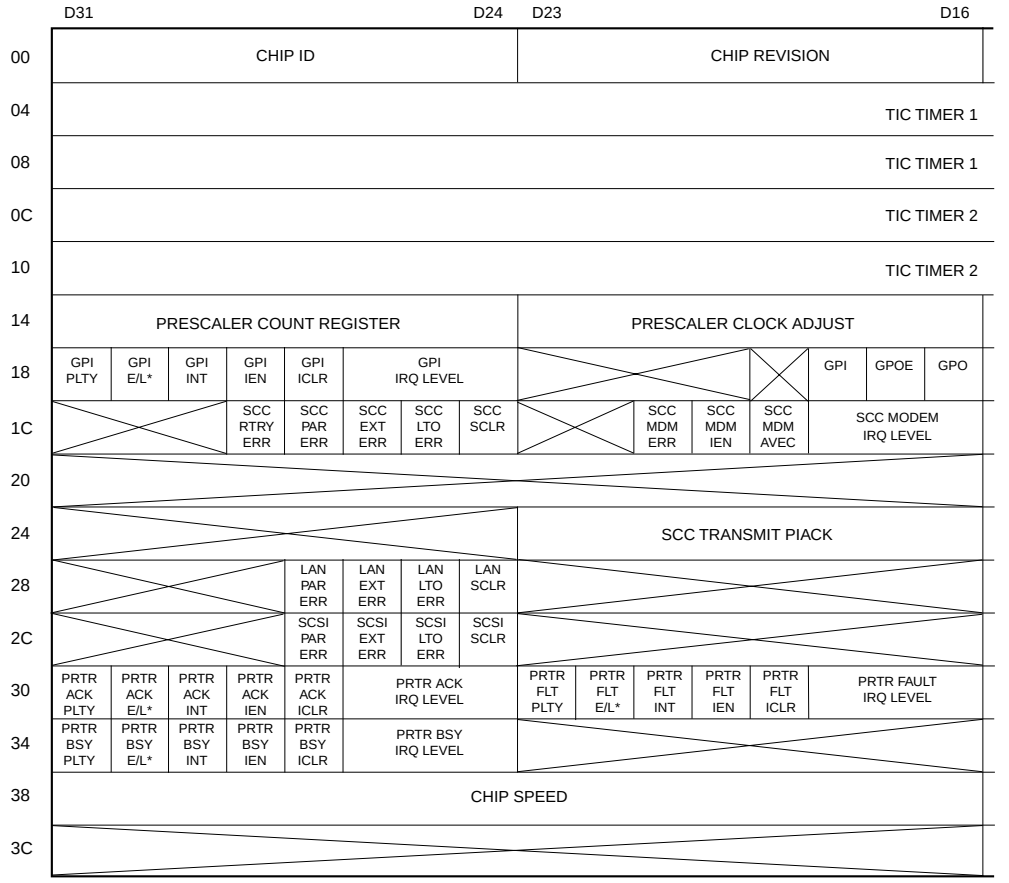
The possible states of the bits after local and power-up reset are as defined below.

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 6-2.

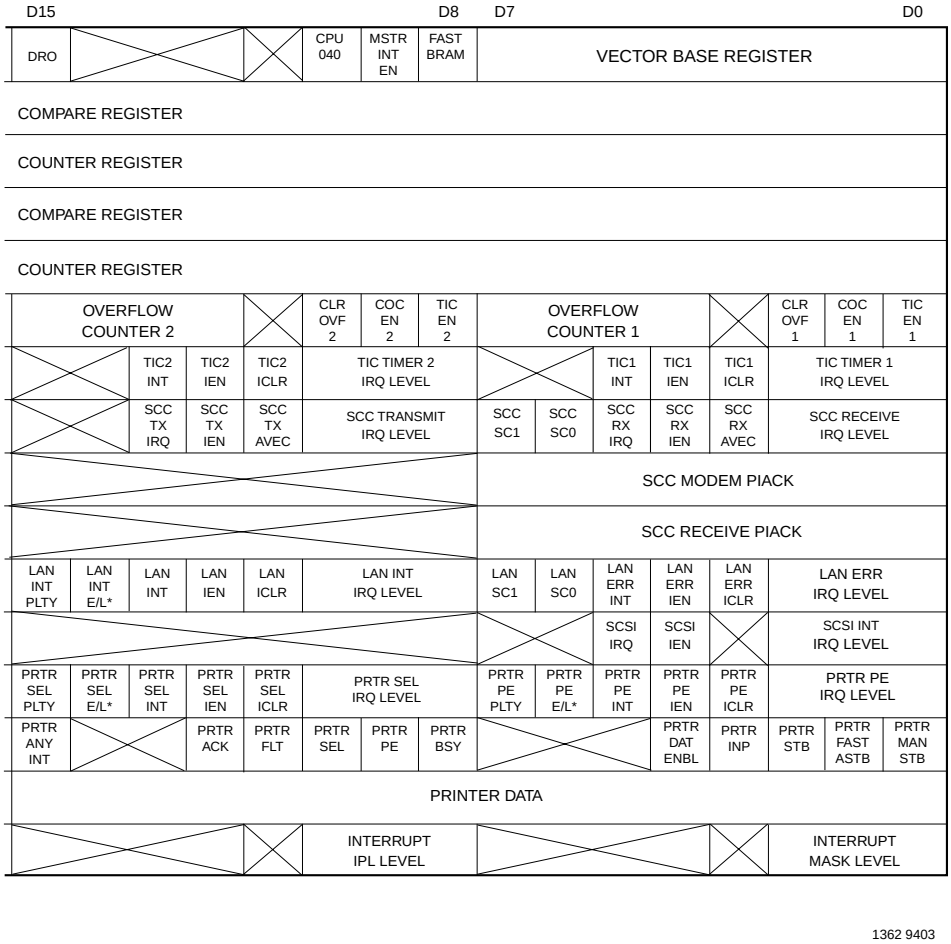
Table 6-2. PCCchip2 Memory Map - Control and Status Registers

PCCchip2 Base Address = $FFF42000
OFFSET:



SCC PROVIDES ITS OWN VECTORS

This sheet continues on facing page. ⟶



1362 9403

← This sheet begins on facing page.

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

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

Note The PCCchip2 runs at half the MPU speed on the MVME176/177. For example, an MVME176/177 with a 50 MHz MPU will run the PCCchip2 at 25 MHz.

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 from the M68000 family. 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).

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

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

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

Note The PCCchip2 runs at half the MPU speed on the MVME176/177. For example, an MVME176/177 with a 50 MHz MPU will run the PCCchip2 at 25 MHz.

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, at 30 MHz it is $E2, 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, the 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, the 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 Single Board Computers, the PCCGPIO1 pin is connected to the remote reset connector 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 and MC68060 user's manuals for details on how it uses the Snoop Control signals.

Notes These bits must be 0 on the MVME187.

On the MVME176/177, which uses only SC1, the SC0 bit must be 0.

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.

Notes These bits must be 0 on the MVME187.

On the MVME176/177, which uses only SC1, the SC0 bit must be 0.

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.

Note The PCCchip2 runs at half the MPU speed on the MVME176/177. For example, an MVME176/177 with a 50 MHz MPU will run the PCCchip2 at 25 MHz.

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 7

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.

Note The MEMC040 is not used with the MVME176/177.

Summary of Features

This section lists the features of the MEMC040 chip.

- 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 7-1.

Performance

The MEMC040 is specifically designed to provide maximum performance for cache line (burst) cycles to and from the MC68040-compatible local bus (Local 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.

Table 7-1 lists the performance specifications for the MEMC040.

Table 7-1. MEMC040 Performance Specifications

Descriptions	Specifications
Reads, FASTREAD = 1	4 clock cycles for random reads 4-1-1-1 clock cycles for line reads (sustained)
Reads, FASTREAD = 0	5 clock cycles for random reads 5-1-1-1 clock cycles for line reads (sustained)
Reads, parity checking on	6 clock cycles for random reads 6-1-1-1 clock cycles for line reads (sustained)

Parity checking ON 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.

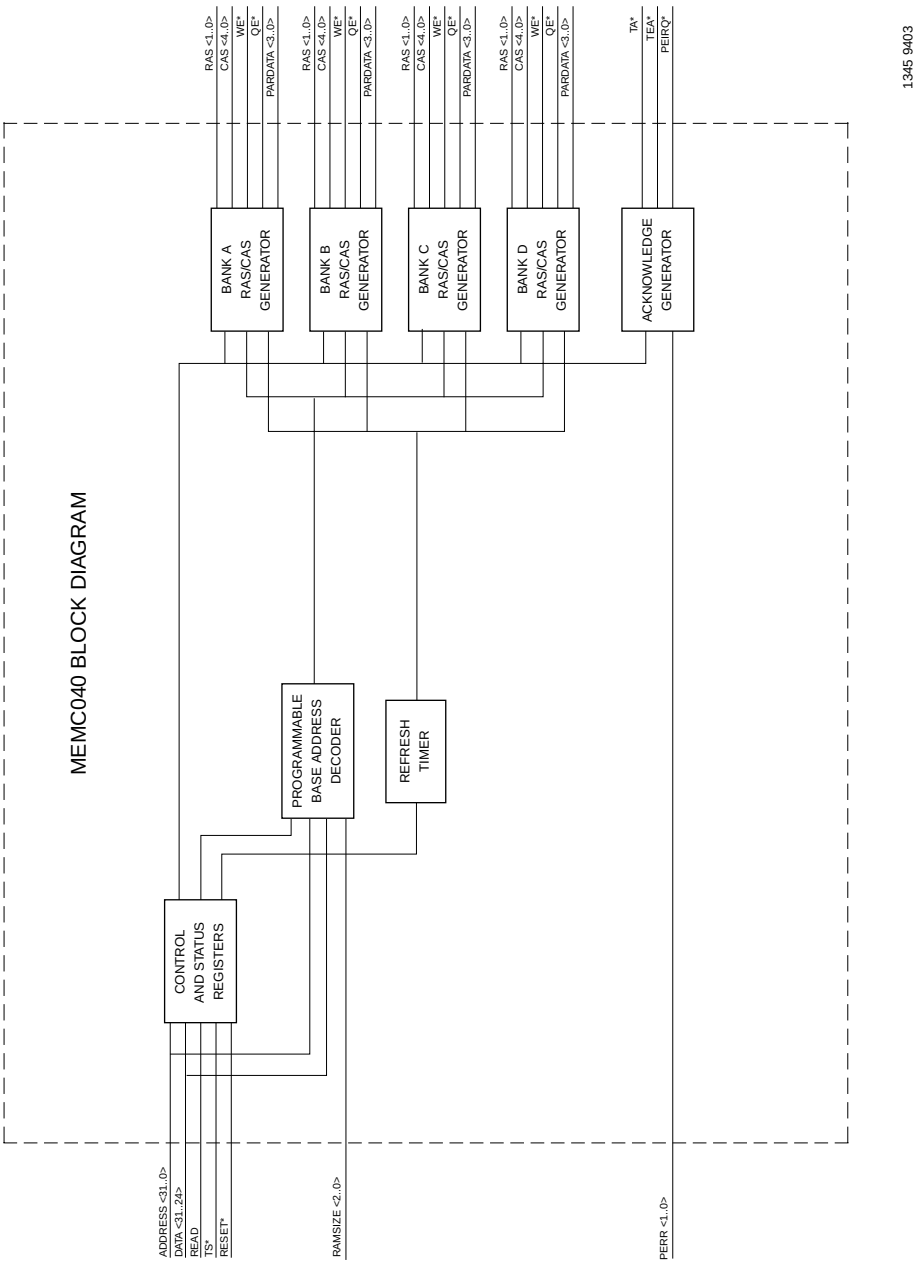


Figure 7-1. Block Diagram for Memory Using MEMC040

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 7-2 shows all MEMC040 internal registers.

Table 7-2. 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

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*.

ADR/SIZ	1st $FFF43008/2nd $FFF43108 (8 bits [6 used])							
BIT	31	30	29	28	27	26	25	24
NAME			FSTRD	EXTPEN	WPB*	MSIZ2	MSIZ1	MSIZ0
OPER			R	R	R	R	R	R
RESET			X	X	X	X	X	X

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 TEA, 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

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 Single Board Computers described in this manual and is used in a set of two, to provide the interface to a 144-bit wide DRAM memory system.

Summary of Features

This section lists the features of the MCECC chip.

- 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. 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 the error. 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-type 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 50ns 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 amounts to two clocks for one random write, and 10 clocks for sustained random writes.

The performance specifications for the MCECC are shown in Table 8-1.

Table 8-1. MCECC Performance 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 8-2, 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 8-2. 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 8-3, 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 8-3. 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

Table 8-3. MCECC Internal Register Memory Map, Part 2 (Continued)

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

Register Offset	Register Name	Register Bit Names							
		D31	D30	D29	D28	D27	D26	D25	D24
$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 on the MEMC040 they reflect input pins, but on the MCECC they reflect register bits that are initialized by the reset serial bit stream.

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 the Single Board Computers).

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 the Single Board Computers).

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 reset 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	SBPD6	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 0.5 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)

This register reflects the current value in the scrub prescaler 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

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)

This register reflects the value that was on bits 31-24 of the local MC68040 address bus at the last logging of an error.

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

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)

This register reflects the value that was on bits 15-8 of the local MC68040 address bus at the last logging of an error.

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

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, software, 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, software, 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, software, 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, software, 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, software, 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, software, 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, software, 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, software, 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 in Error	Bit in Error	Syndrome Code
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

Printer and Serial Port Connections

9

Introduction

This chapter has connection diagrams for the printer port and the four serial ports on the MVME167/177/187, and or the serial ports on the MVME166/176. 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 user's manual for your MVME712 series transition board.

Connection Diagrams

The MVME712*x* transition module connection diagrams are shown in the following figures:

Figure Number	Name
9-13	MVME167/177/187 Serial Port 3 with MVME712A
9-14	MVME167/177/187 Serial Port 4 with MVME712A
9-15	MVME166/176 Serial Ports with MVME712-10 (Sheets 1 through 4)
9-16	MVME166/176 Serial Ports with MVME712-06 (Sheets 1 through 3)

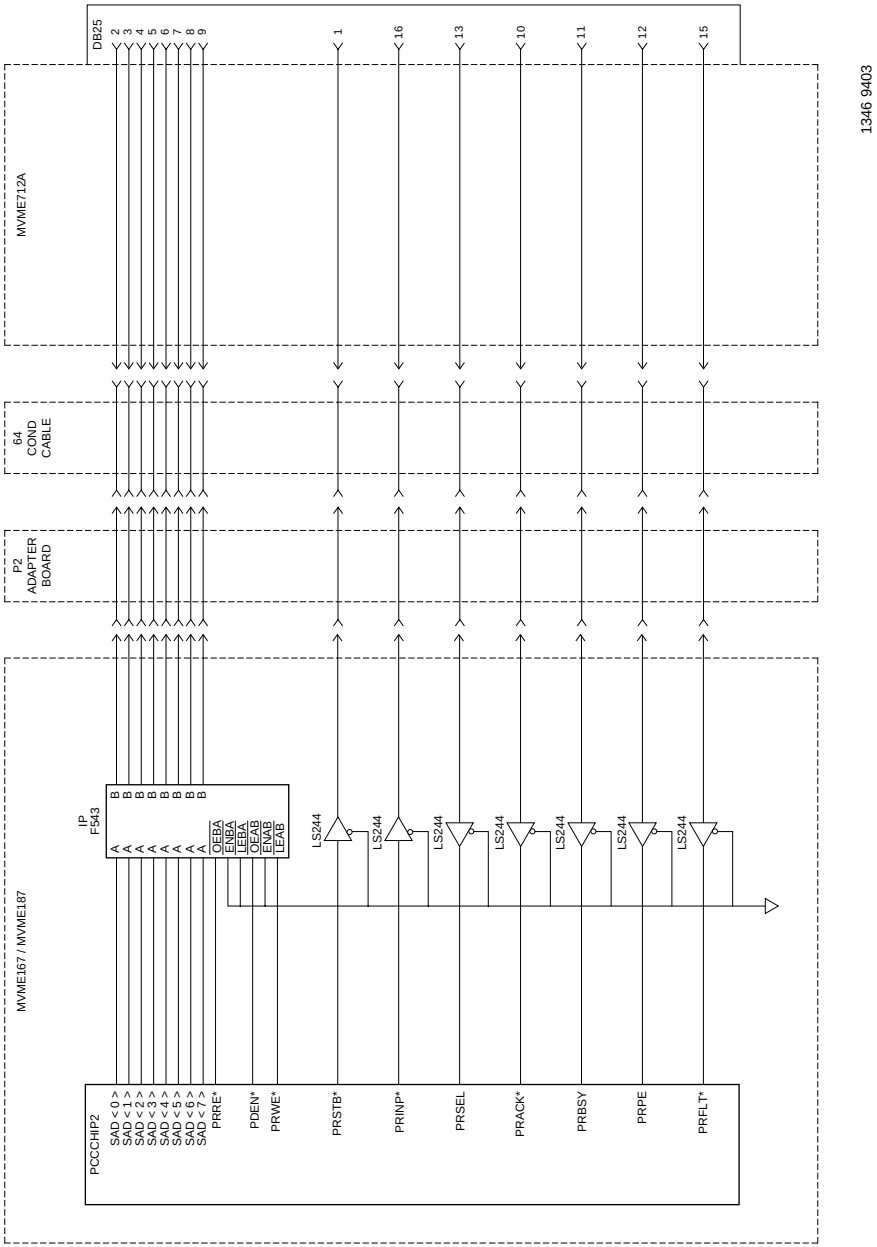


Figure 9-1. MVME167/177/187 Printer Port with MVME712A

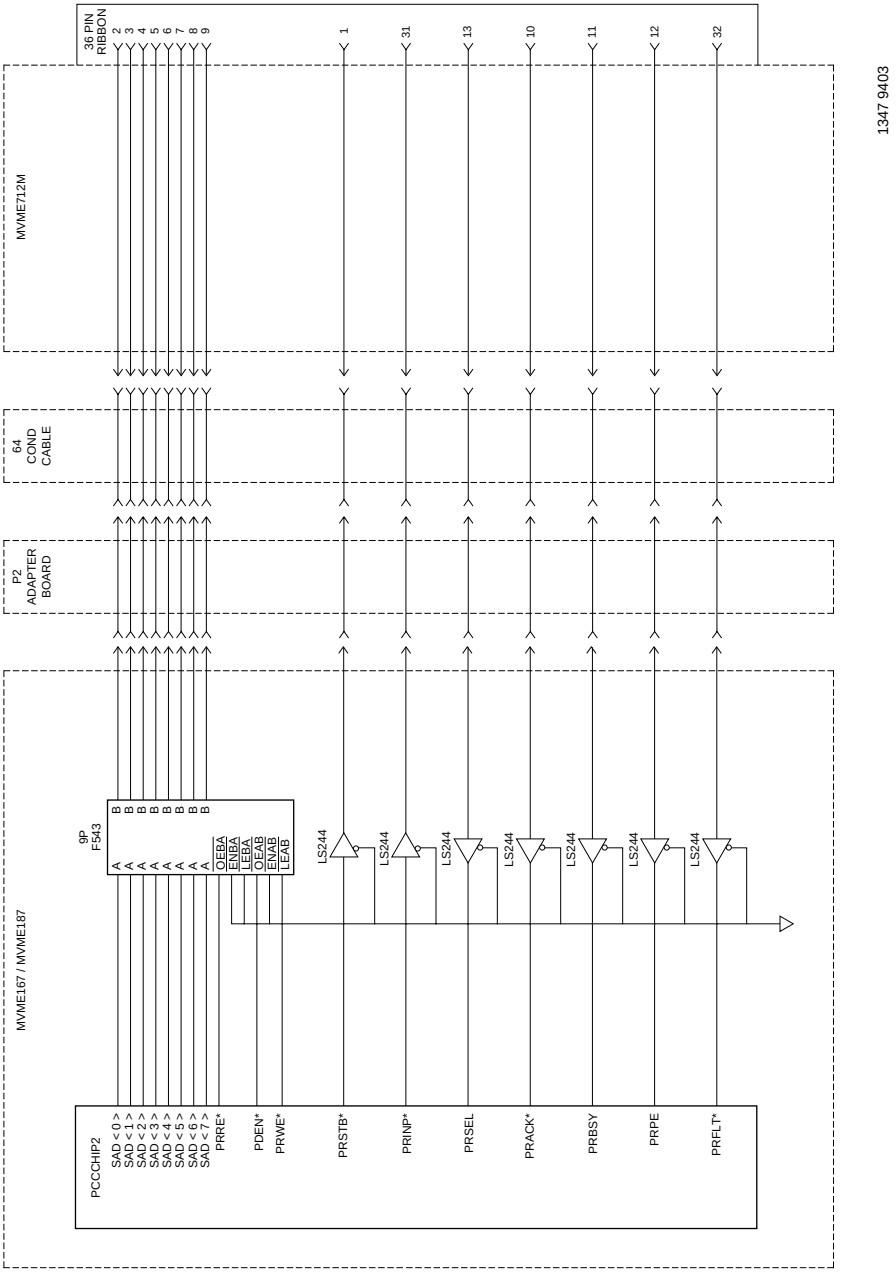


Figure 9-2. MVME167/177/187 Printer Port with MVME712M

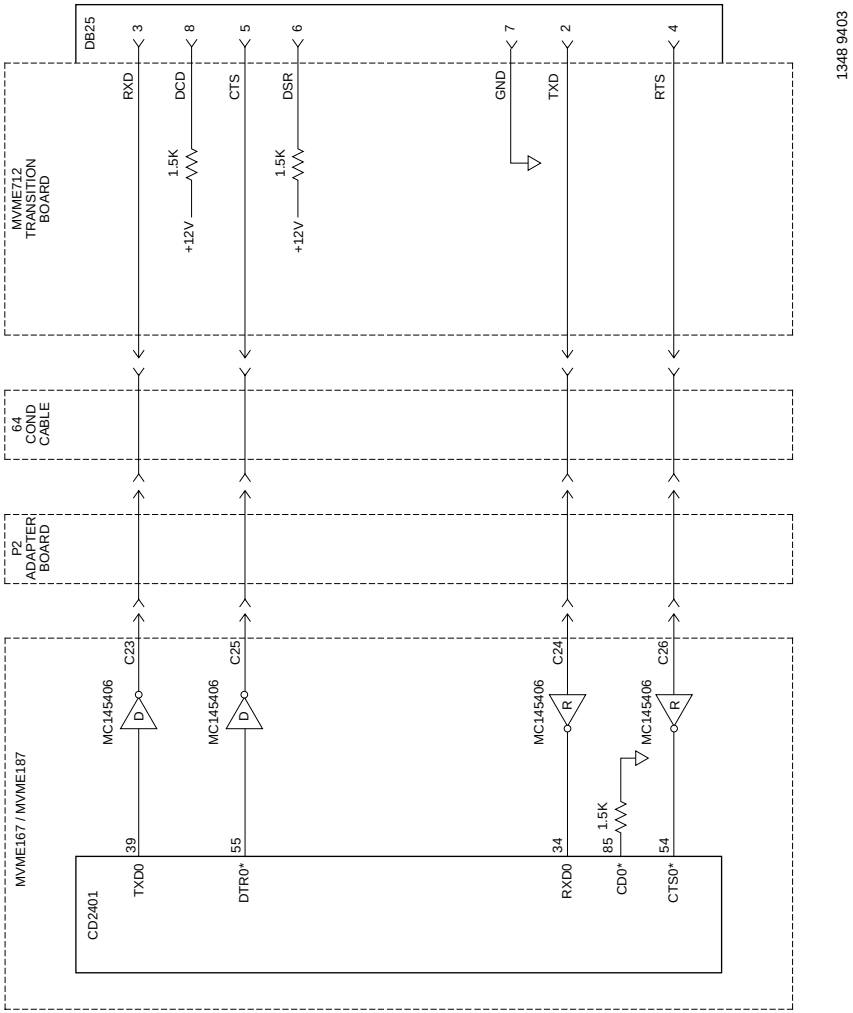


Figure 9-3. MVME167/177/187 Serial Port 1 Configured as DCE

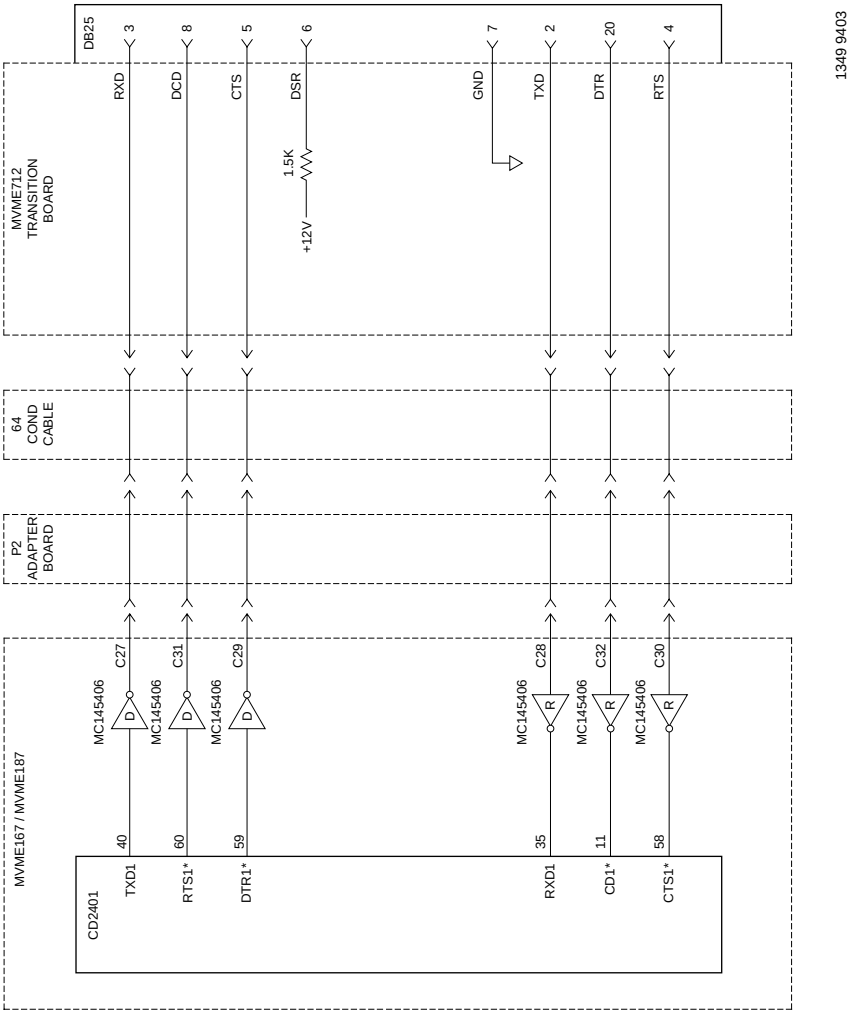


Figure 9-4. MVME167/177/187 Serial Port 2 Configured as DCE

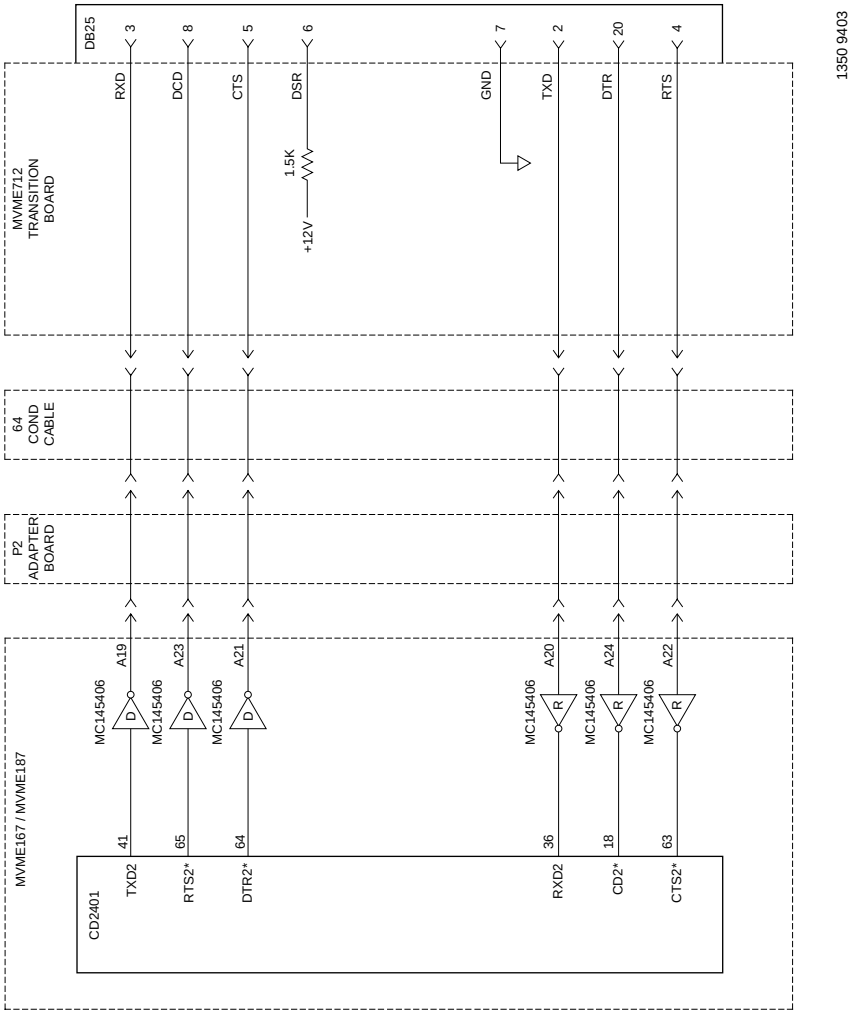


Figure 9-5. MVME167/177/187 Serial Port 3 Configured as DCE

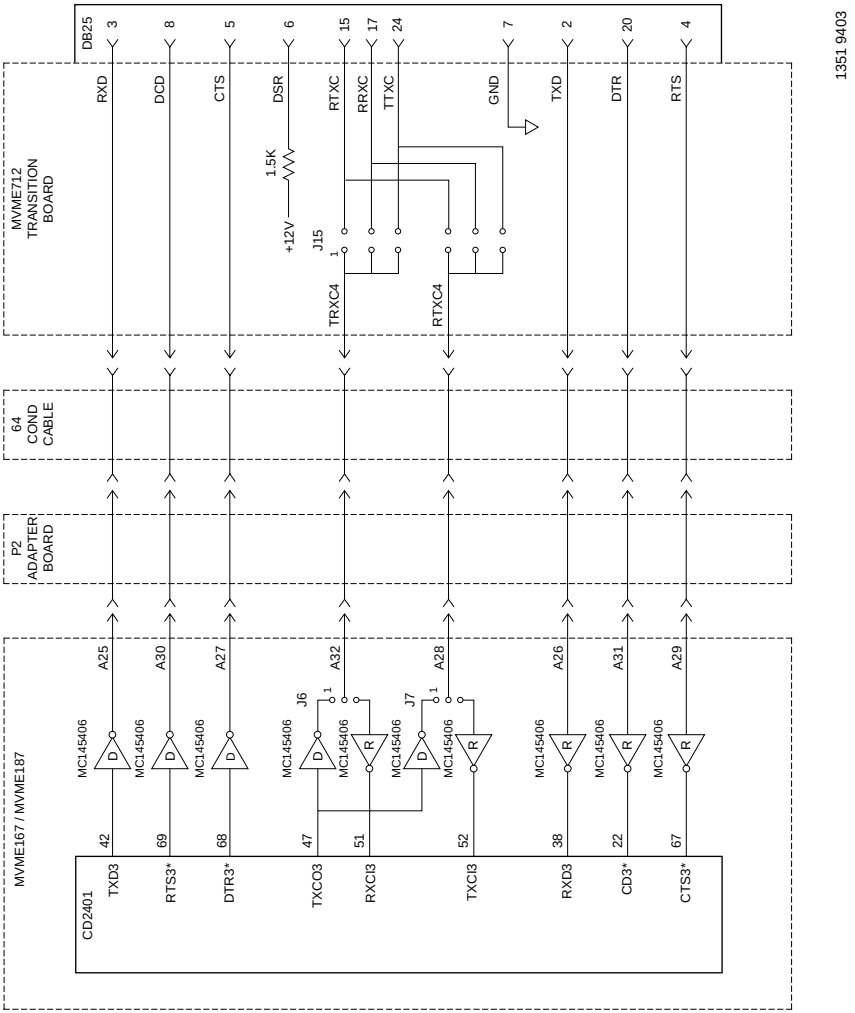


Figure 9-6. MVME167/177/187 Serial Port 4 Configured as DCE

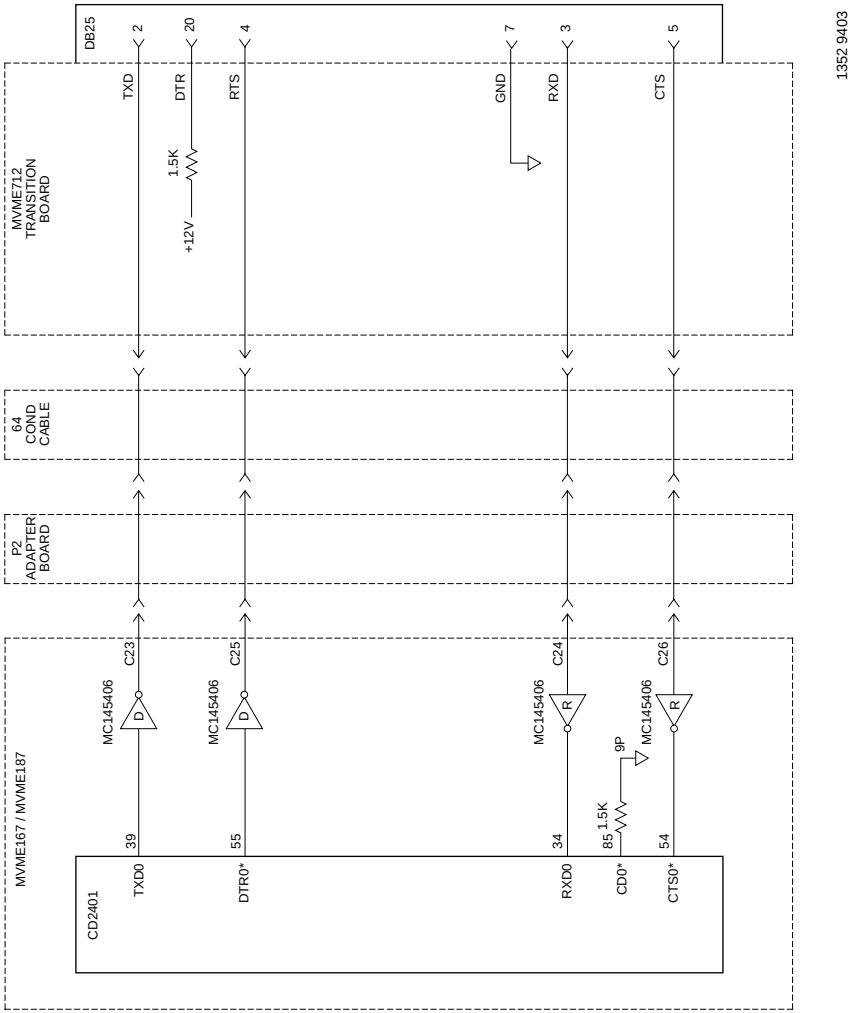


Figure 9-7. MVME167/177/187 Serial Port 1 Configured as DTE

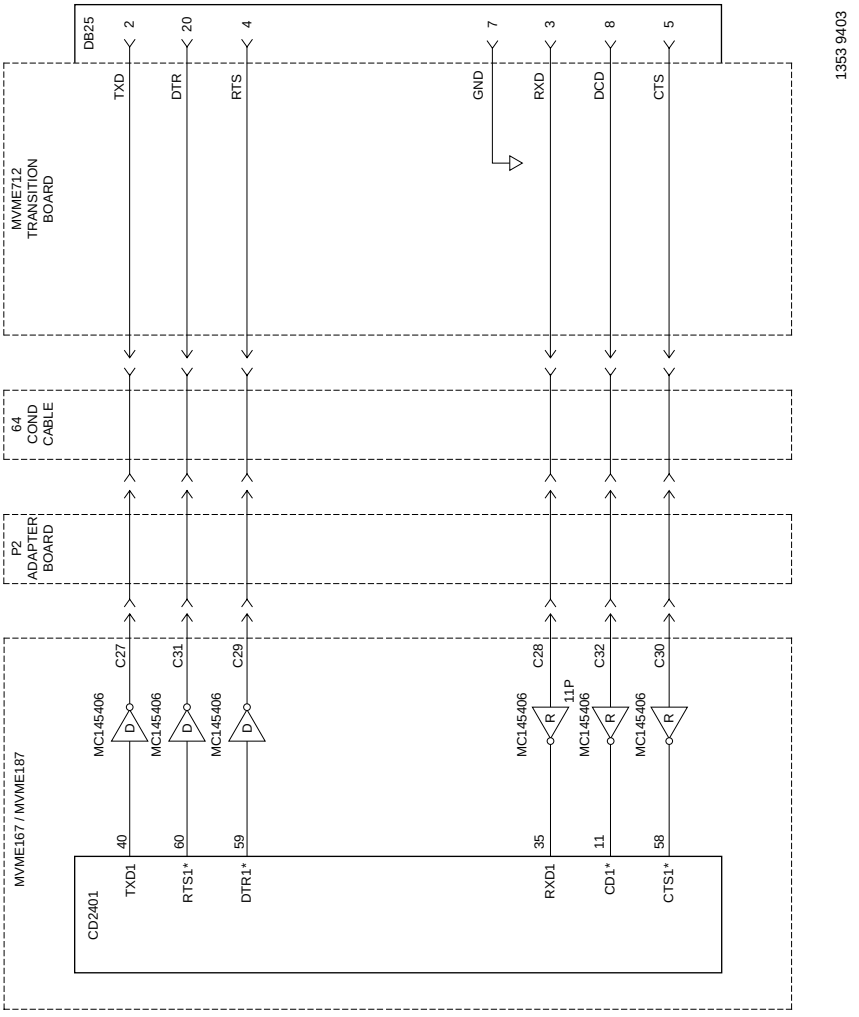


Figure 9-8. MVME167/177/187 Serial Port 2 Configured as DTE

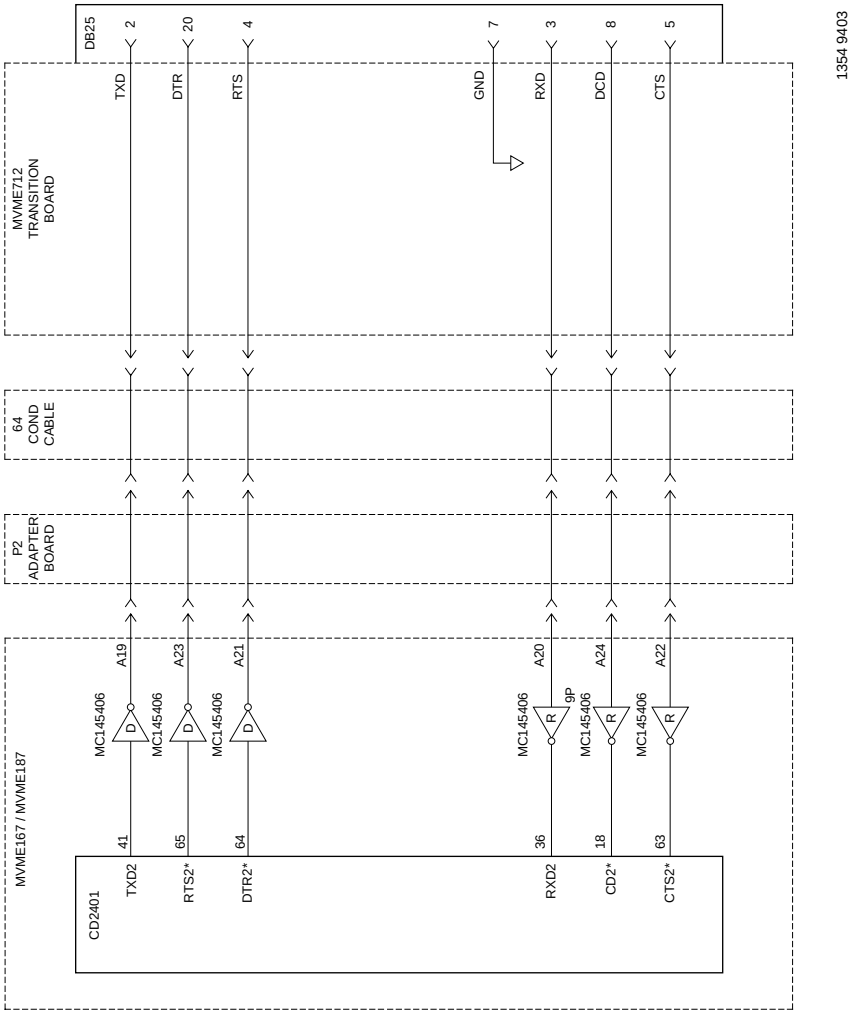


Figure 9-9. MVME167/177/187 Serial Port 3 Configured as DTE

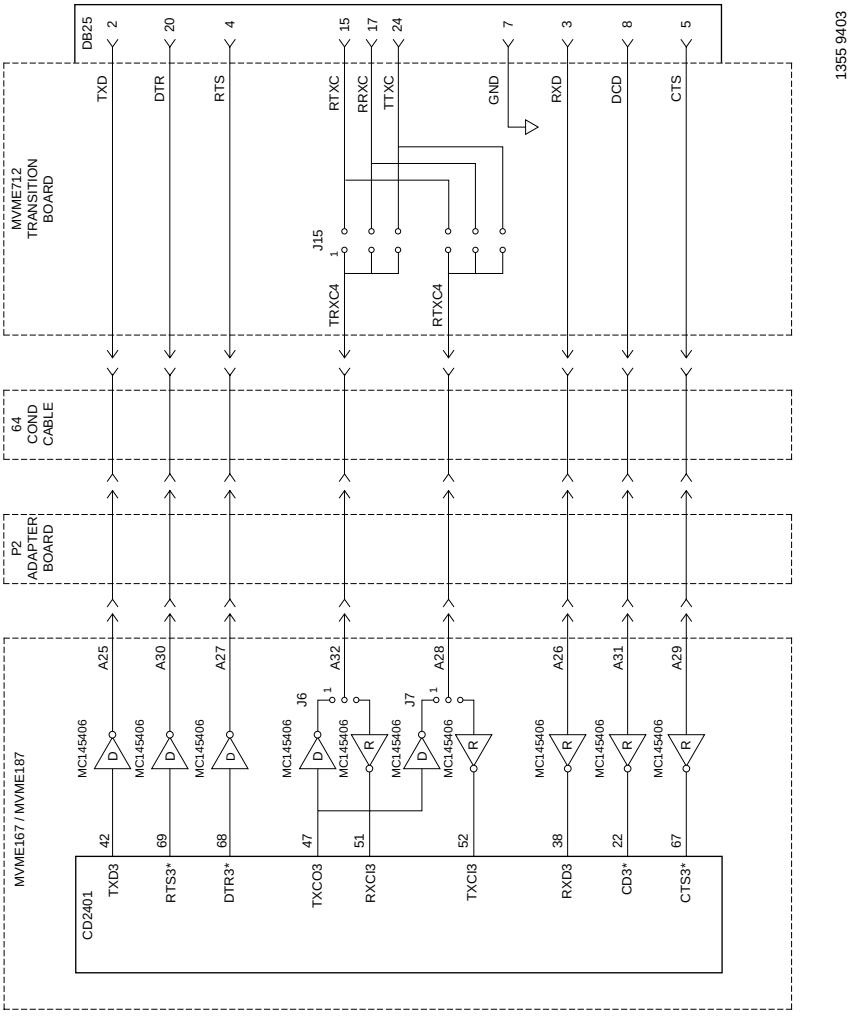


Figure 9-10. MVME167/177/187 Serial Port 4 Configured as DTE

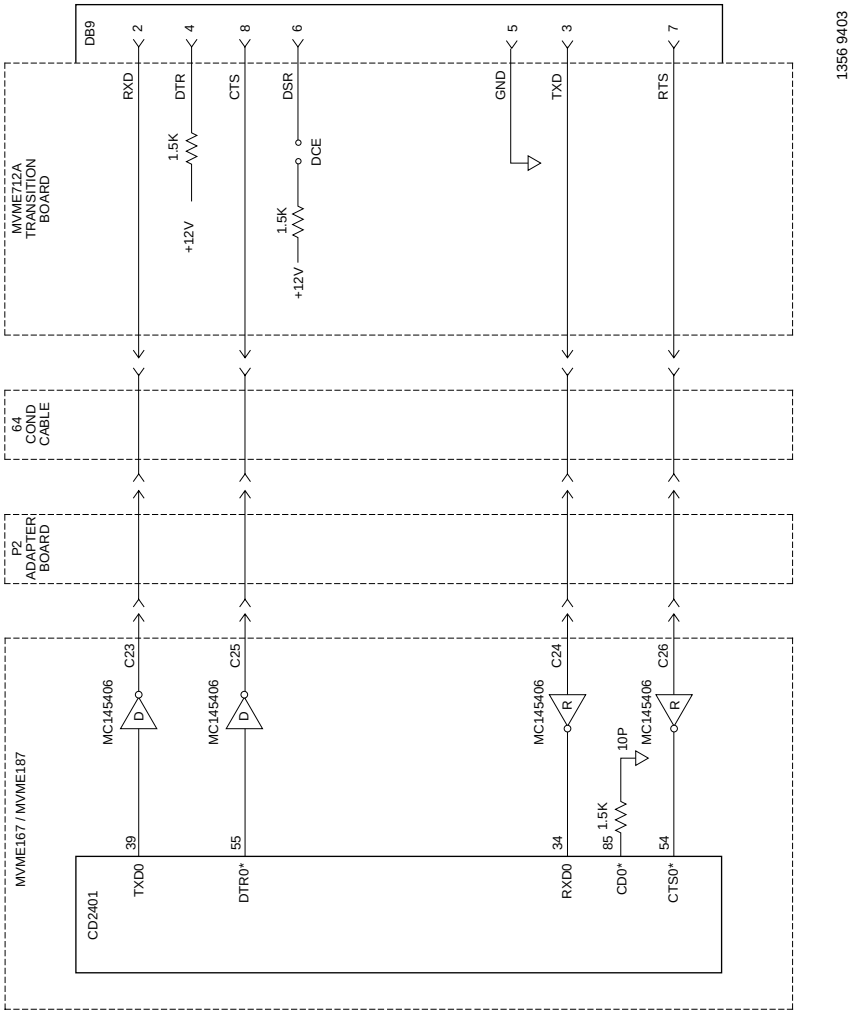


Figure 9-11. MVME167/177/187 Serial Port 1 with MVME712A

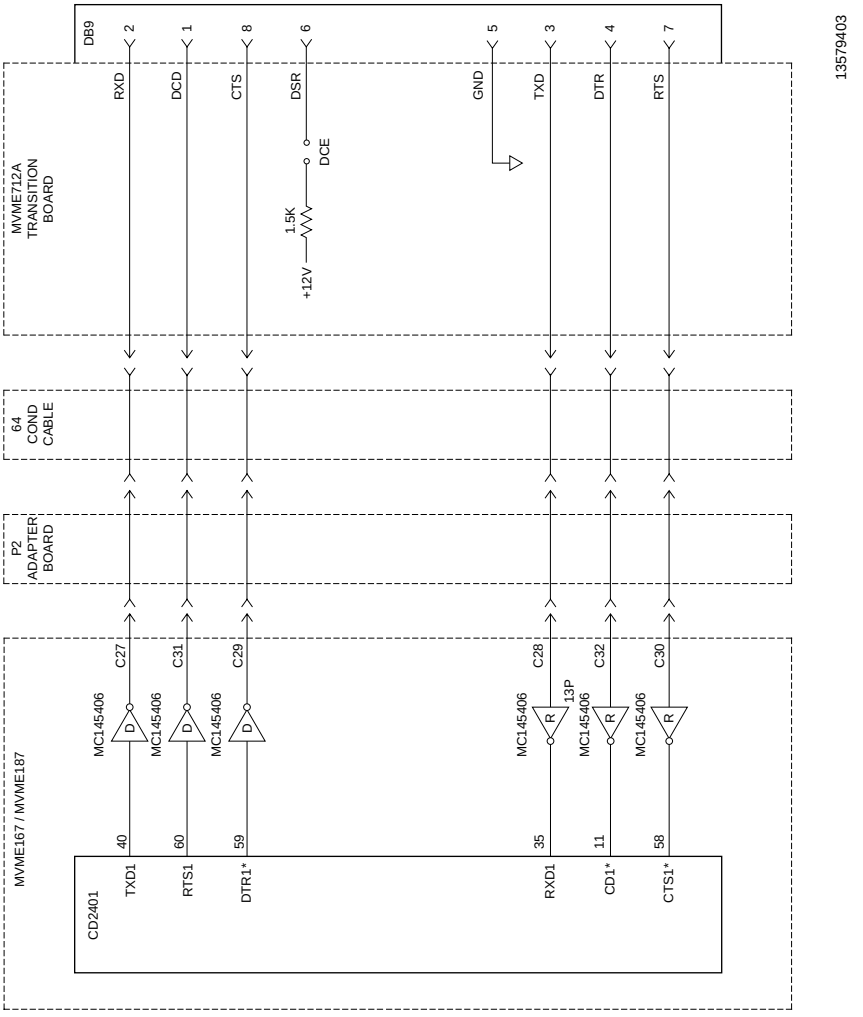


Figure 9-12. MVME167/177/187 Serial Port 2 with MVME712A

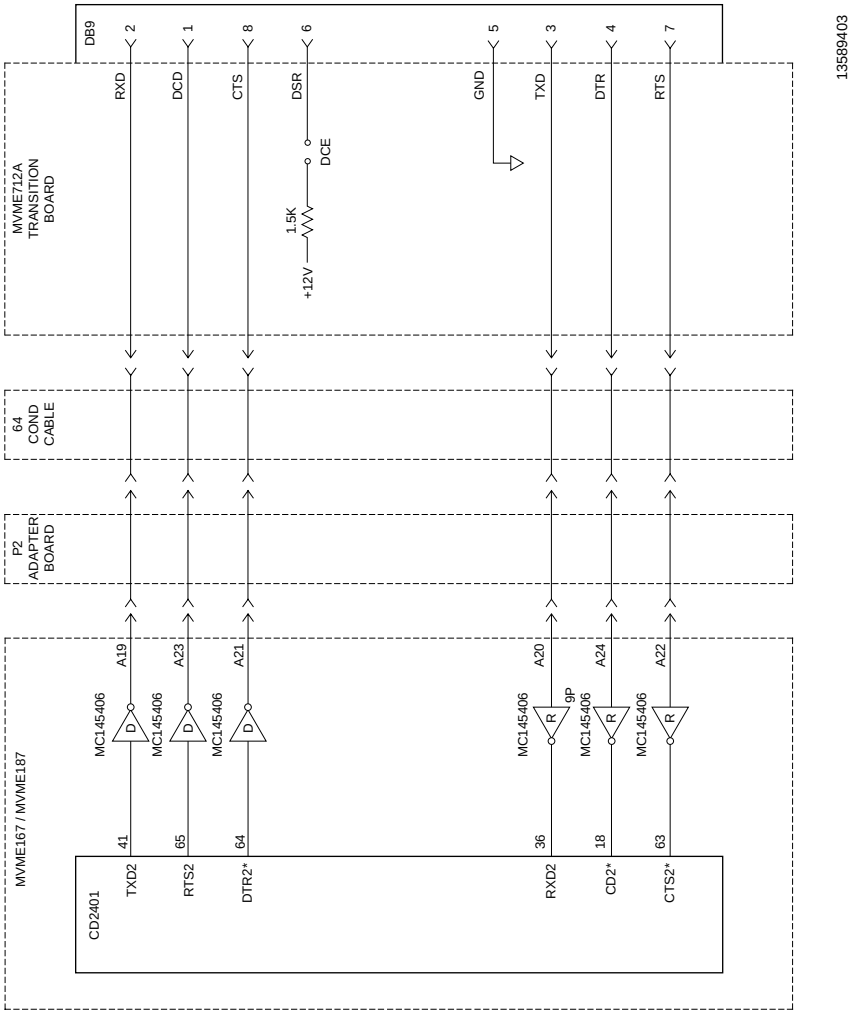


Figure 9-13. MVME167/177/187 Serial Port 3 with MVME712A

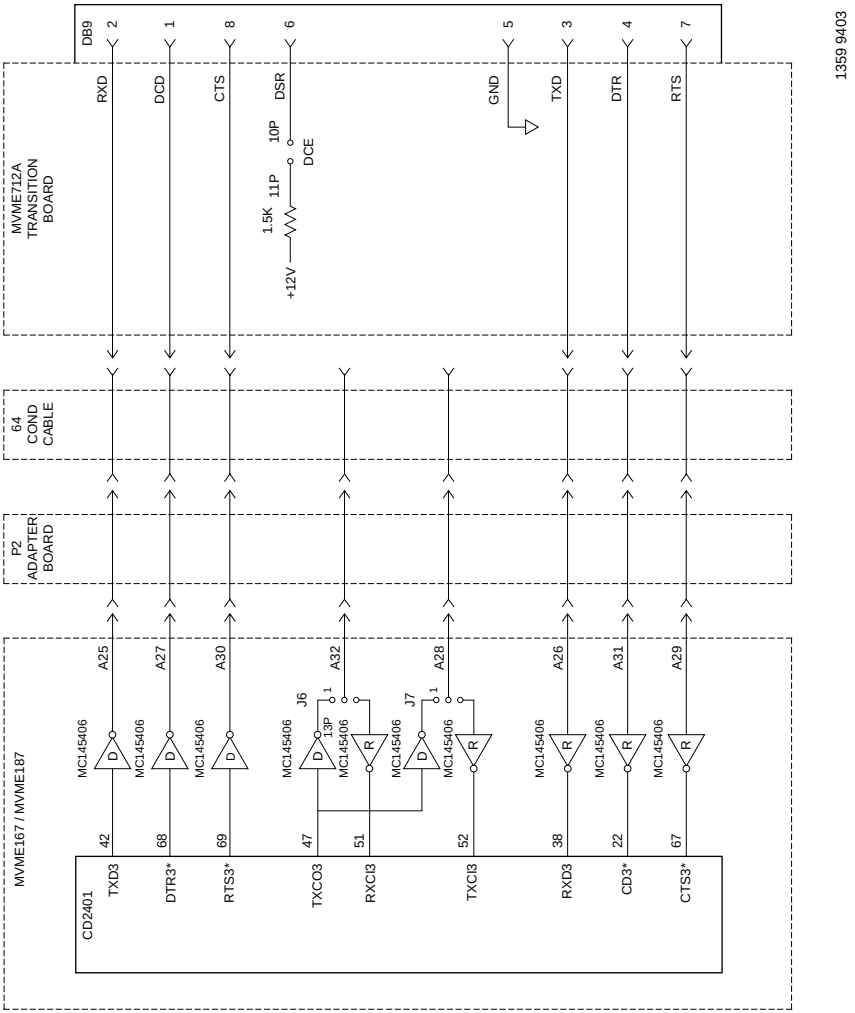


Figure 9-14. MVME167/177/187 Serial Port 4 with MVME712A

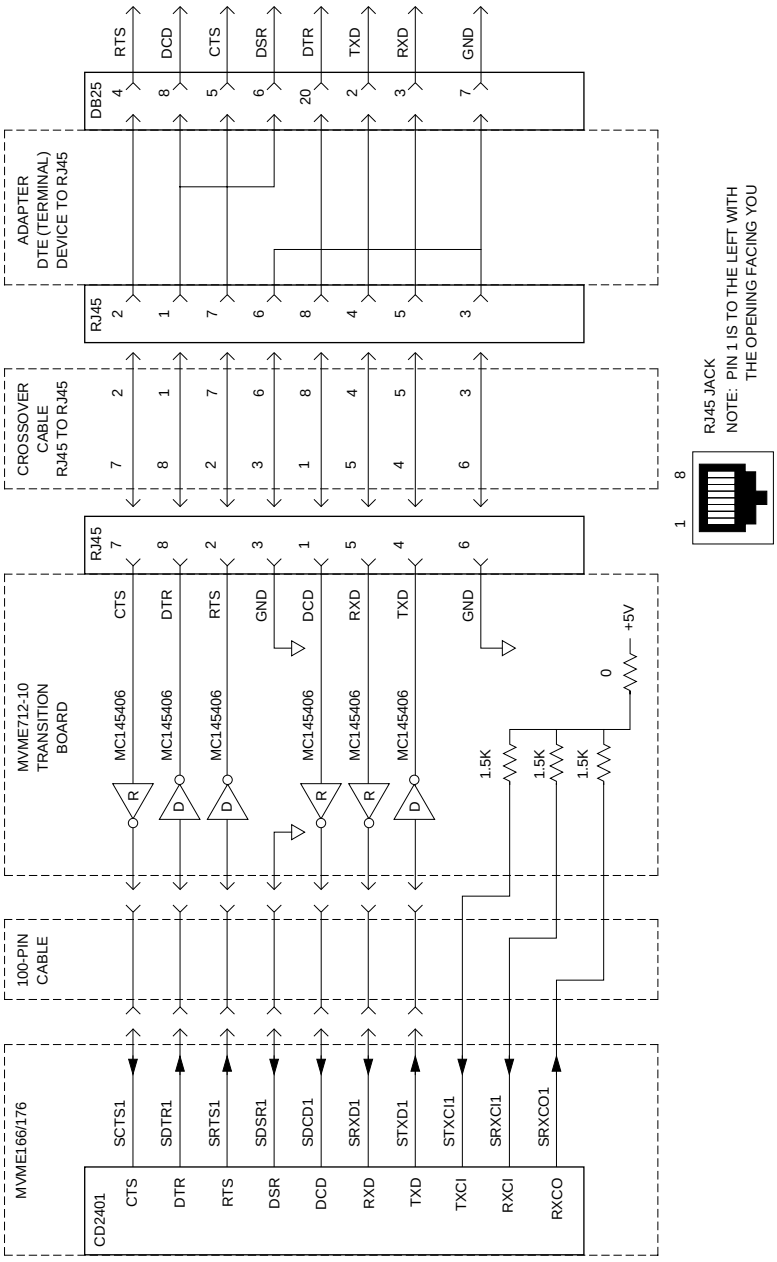


Figure 9-15. MVME166/176 Serial Ports with MVME712-10 (Sheet 1 of 4)

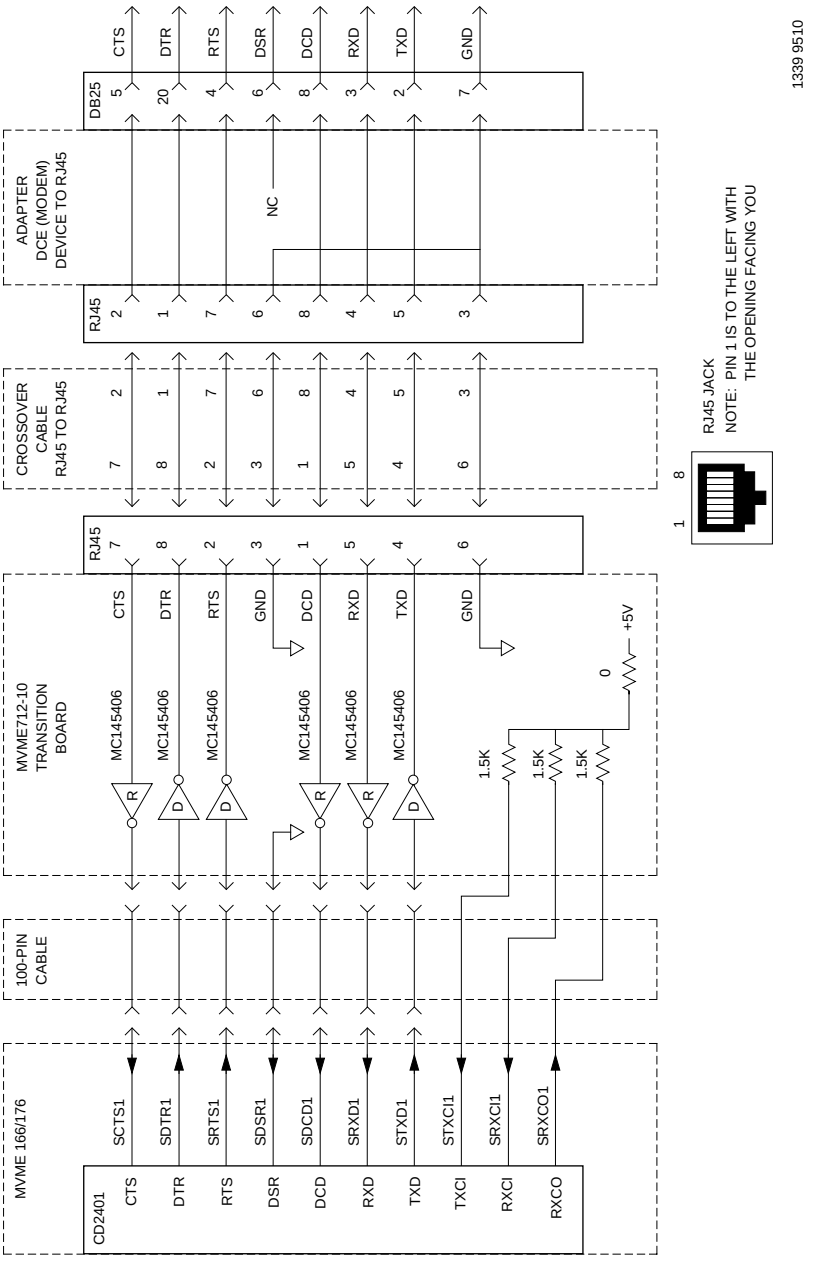


Figure 9-15. MVME166/176 Serial Ports with MVME712-10 (Sheet 2 of 4)

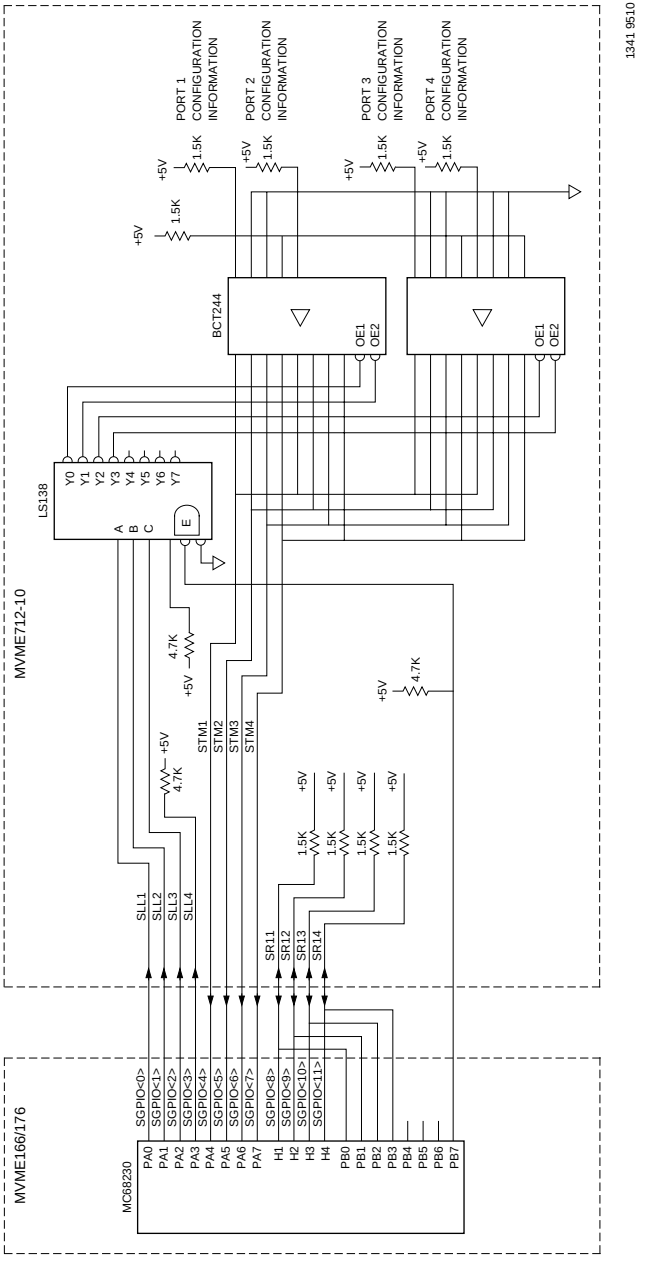


Figure 9-15. MVME166/176 Serial Ports with MVME712-10 (Sheet 3 of 4)

Configuration Register Notes

The application software on the MVME166/176 can read the configuration register to identify the transition board(s) in the system.

On the MVME166/176, the interrupt level on the MC68230 Parallel Interface Timer (PI/T) is the same a 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:	Port B direction must be:
Mode 0, submode 1X	Mode 0, submode 1X
Bits 7 to 4 as inputs	Bit 7 as an output
Bits 3 to 0 as outputs	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 9-15. MVME166/176 Serial Ports with MVME712-10 (Sheet 4 of 4)

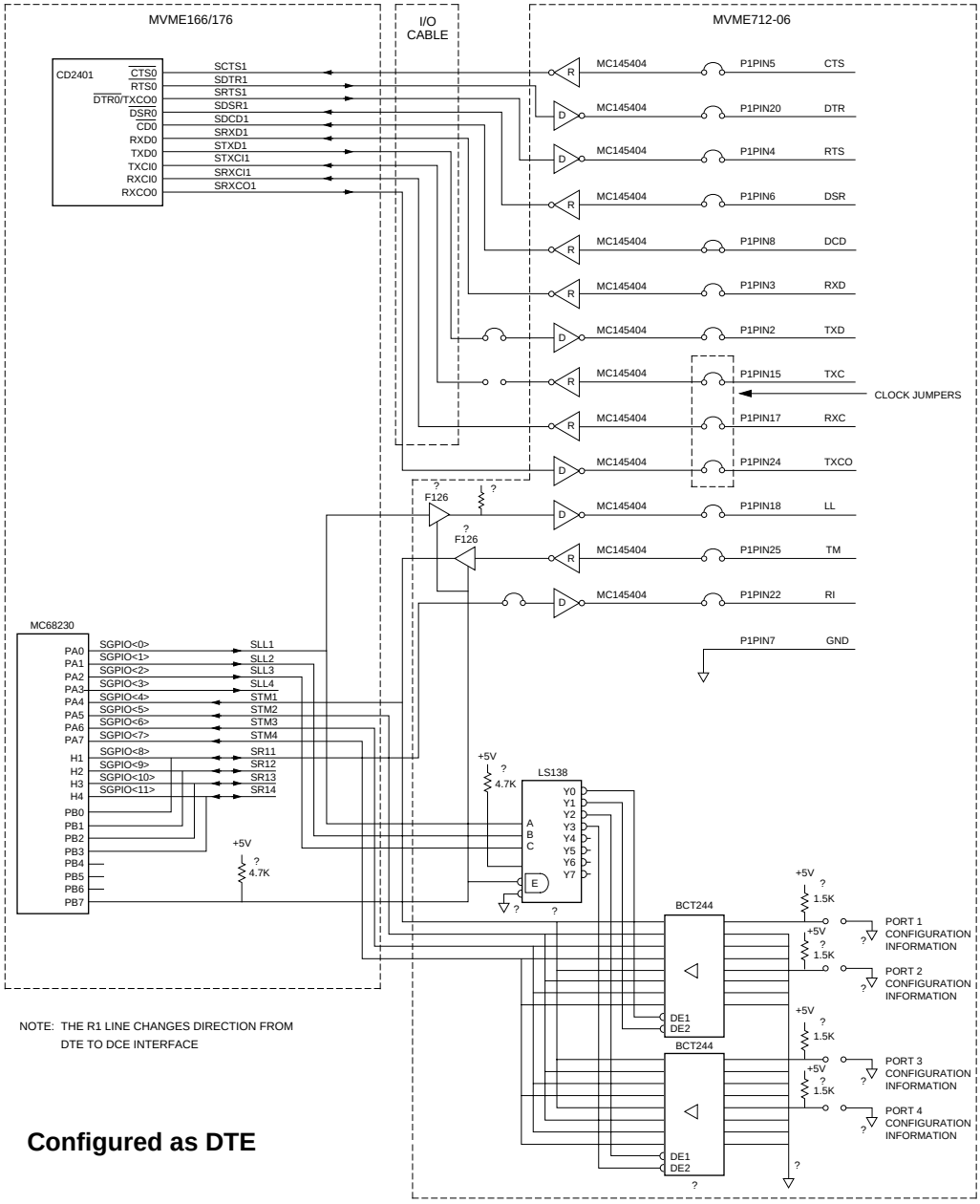


Figure 9-16. MVME166/176 Serial Ports with MVME712-06 (Sheet 1 of 3)

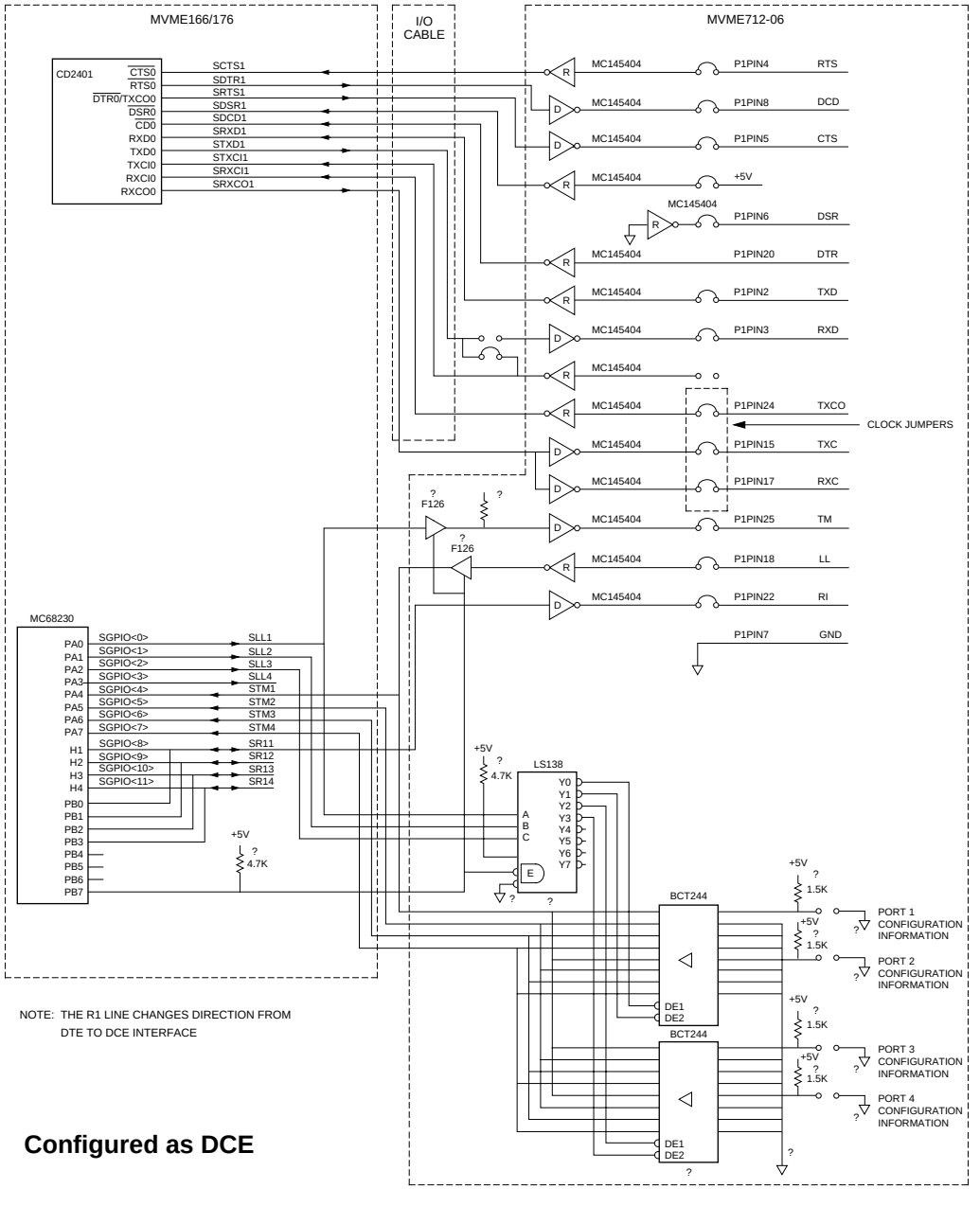


Figure 9-16. MVME166/176 Serial Ports withMVME712-06 (Sheet 2 of 3)

Configuration Register Notes

The application software on the MVME166/176 can read the configuration register to identify the transition board(s) in the system.

On the MVME166/176, the interrupt level on the MC68230 Parallel Interface Timer (PI/T) is the same a 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 9-16. MVME166/176 Serial Ports with MVME712-06 (Sheet 3 of 3)

Index

D

G

H

I

M

INDEX

INDEX

R

INDEX

S

INDEX

W

Z