MOTOROLA

# MVME165
# VME/VSB Microcomputer VMEmodule
# User's Manual

TECHNICAL PUBLICATIONS M/D DW164
MOTOROLA
Computer Group
2900 S. DIABLO WAY
TEMPE, AZ 85282-9602

# READER COMMENTS

*Write if we're not right (or even if we are)...*

Motorola welcomes your comments on its documentation. We want to know what you think about our manuals and how we can make them better. Please take the time to complete and mail this reply card.

Manual Title ______________________

Part Number ______________ Date ________

Your Name______________________

Your Title ____________ Telephone (___)________

Company ______________________

Address ______________________

______________________

______________________

## General Information:

Do you read this manual in order to:

□ Install the product
□ Use the product
□ Other ______________________

□ Repair the product
□ Reference information

In general, how do you locate information within this manual?

□ Index
□ Table of Contents

□ Page Headings
□ Other

## Completeness:

What topic would you like more information on:

______________________

______________________

______________________

## Presentation:

Is the manual easy to use? □ Yes □ No

Please explain what is good or poor: ______________

______________________

______________________

______________________

Is the information easy to understand? □ Yes □ No

If you checked no, please explain: ______________

______________________

______________________

Is the information easy to find? □ Yes □ No

If you checked no, please explain: ______________

______________________

______________________

## Accuracy:

If you have found technical or typographical errors, please list them here.

| Page Number | Description of Error |
|---|---|
| | |
| | |
| | |

MOTOROLA

# MVME165
# VME/VSB MICROCOMPUTER VMEmodule
# User's Manual
# (MVME165/D2)

## NOTICE

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

## RESTRICTED RIGHTS LEGEND

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

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

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

# PREFACE

This manual provides general information, hardware preparation and installation instructions, operating instructions, and functional description for the MVME165 VME/VSB Microcomputer VMEmodule.

This manual is intended for anyone who wants 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 may need to be familiar with the publications listed in the *Related Documentation* section in Chapter 1 of this manual.

## WARNING

**THIS EQUIPMENT GENERATES, USES, AND CAN RADIATE RADIO FREQUENCY ENERGY AND IF NOT INSTALLED AND USED IN ACCORDANCE WITH THE INSTRUCTIONS MANUAL, MAY CAUSE INTERFERENCE TO RADIO COMMUNICATIONS. IT HAS BEEN TESTED AND FOUND TO COMPLY WITH THE LIMITS FOR A CLASS A COMPUTING DEVICE PURSUANT TO SUBPART J OF PART 15 OF FCC RULES, WHICH ARE DESIGNED TO PROVIDE REASONABLE PROTECTION AGAINST SUCH INTERFERENCE WHEN OPERATED IN A COMMERCIAL ENVIRONMENT. OPERATION OF THIS EQUIPMENT IN A RESIDENTIAL AREA IS LIKELY TO CAUSE INTERFERENCE IN WHICH CASE THE USER, AT HIS OWN EXPENSE, WILL BE REQUIRED TO TAKE WHATEVER MEASURES NECESSARY TO CORRECT THE INTERFERENCE.**

# SAFETY SUMMARY
# SAFETY DEPENDS ON YOU

***The following general safety precautions must be observed during all phases of operation, service, and repair of this equipment. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the equipment. Motorola Inc. assumes no liability for the customer's failure to comply with these requirements. The safety precautions listed below represent warnings of certain dangers of which we are aware. You, as the user of the product, should follow these warnings and all other safety precautions necessary for the safe operation of the equipment in your operating environment.***

## GROUND THE INSTRUMENT.

To minimize shock hazard, the equipment chassis and enclosure must be connected to an electrical ground. The equipment is supplied with a three-conductor ac power cable. The power cable must either be plugged into an approved three-contact electrical outlet or used with a three-contact to two-contact adapter, with the grounding wire (green) firmly connected to an electrical ground (safety ground) at the power outlet. The power jack and mating plug of the power cable meet International Electrotechnical Commission (IEC) safety standards.

## DO NOT OPERATE IN AN EXPLOSIVE ATMOSPHERE.

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

## KEEP AWAY FROM LIVE CIRCUITS.

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

## DO NOT SERVICE OR ADJUST ALONE.

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

## USE CAUTION WHEN EXPOSING OR HANDLING THE CRT.

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

## DO NOT SUBSTITUTE PARTS OR MODIFY EQUIPMENT.

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

## DANGEROUS PROCEDURE WARNINGS.

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

**WARNING**

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

# CONTENTS

# APPENDICES

# INDEX

# FIGURES

# TABLES

# CHAPTER 1
# GENERAL INFORMATION

## Introduction

This user's manual provides general information, preparation for use and installation instructions, operating instructions, and a functional description of the MVME165 series of VME/VSB Microcomputers (referred to as the MVME165 throughout this manual).

## Model Designations

The MVME165 is available in four models, which are listed in Table 1-1.

**Table 1-1.** MVME165 Model Designations

| Product Number | Product Description |
|---|---|
| MVME165-01 | 25MHz, 4Mb Embedded Processor |
| MVME165-02 | 25MHz, 16Mb Embedded Processor |
| MVME165-03 | 33MHz, 4Mb Embedded Processor |
| MVME165-04 | 33MHz, 16Mb Embedded Processor |

## Features

These are some of the major features of the MVME165:

- MC68040 microprocessor with 8Kb cache, MMU (Memory Management Unit), and FPU (Floating Point Unit)
- 4/16Mb of DRAM (burst fill completes in 4-1-1-1 processor clock cycles)
- Cache Monitor providing cache coherency for the local DRAM
- Two 32-pin JEDEC standard sockets for the EPROM
- One 28-pin JEDEC standard socket for the TOD/NVRAM (Time-of-Day Clock/8K x 8 Non-Volatile RAM)
- Two front panel RS-232C serial ports (MC68681)
- One 16-bit programmable timer (MC68681)
- Three 24-bit programmable counter/timers implemented as the Local Resource Controller (LRC)

- VSB bus Master/Slave Interface using the MVSB2400 (VSBchip) which has:
  - VSB system controller/bus arbiter
  - VSB requester with programmable timeout module
  - Block transfer module with byte count register
  - Control and Status Register (CSR)
- VMEbus A32/D32 Master/Slave Interface using the MVME6000 (VMEchip) which has:
  - VMEbus System Controller with 4-Level Arbiter
  - VMEbus A32/A24/A16;D32/D16/D08 Master Interface
  - VMEbus 7-Level Interrupter
  - VMEbus 7-Level Interrupt Handler
  - VMEbus 4-Level Requester
  - Multi-processor Control/Status Registers
- Front panel FAIL, HALT, RUN, and SCON LED (Light Emitting Diode) indicators
- Front panel ABORT and RESET push-button switches
- Thirteen general purpose software control jumpers
- VMEbus double-high, single-wide form factor

## Specifications

General specifications for the MVME165 are listed in Table 1-2.

**Table 1-2.** MVME165 Specifications

| Characteristics | Specifications |
|---|---|
| Microprocessor | MC68040 32-bit microprocessor |
| Clock signal | 25.0-33.0 MHz CPU clock frequency<br>(The specific operating frequency is determined by the oscillator installed.) |
| CACHE | The MC68040 contains an internal 8Kb cache (with 4K instruction and 4K data) which supports one cycle read accesses and external (to the processor) bus snooping of local DRAM memory. |
| DRAM | 4/16Mb of DRAM (Dynamic RAM) are provided using either one megabit or four megabit RAMs. |

**Table 1-2.** MVME165 Specifications (cont'd)

| Characteristics | Specifications |
|---|---|
| MMU | The MC68040 contains an internal Memory Management Unit (MMU) which supports 4K or 8K page sizes. |
| FPU | The MC68040 contains a Floating Point Unit (FPU) which implements a subset of the MC68881/MC68882 FPU. |
| SRAM | One 28-pin JEDEC socket which provides 8K x 8 of NVRAM (Non-Volatile RAM) and a TOD (Time-Of-Day) Clock. |
| ROM | Two 32-pin JEDEC sockets which provide support for 128K x 8, or 256K x 8 EPROMs (165Bug supplied with 256K x 8 devices). |
| I/O | Two RS-232C asynchronous serial ports are provided through the MC68681 DUART (Dual Asynchronous Receiver/Transmitter). |
| Timers | Three 24-bit counter/timers are in the LRC and one 16-bit timer is in the MC68681 DUART. |
| VMEbus | A VMEbus master/slave interface (A32lA24/D32lD16) is implemented with the MVME6000 (VMEchip). |
| VSB | A VSB (bus) master/slave interface (A32/D32) is implemented using the MVSB2400 (VSBchip). |
| Power requirements | +5Vdc, 5.9 Amps maximum (5.4 A typical)<br>+12Vdc, 250 mA maximum<br>-12Vdc, 250 mA maximum |
| Operating temperature (refer to the *Cooling Requirements* section) | 0° to 55° C at point of entry of forced air cooling (approximately 490 LFM) |
| Storage temperature | -40° to 85° C |

**Table 1-2.** MVME165 Specifications (cont'd)

| Characteristics | Specifications |
|---|---|
| Relative humidity | 5% to 95% (non-condensing) |
| Physical characteristics: | VME double-high, single-wide form-factor |
| Height | 6.30 inches (160.0 mm) |
| Width | 9.19 inches (233.4 mm) |
| Thickness | 0.062 inch (1.57 mm) |
| Part projections: | |
| Component side | 0.50 inch (12.7 mm) maximum |
| Solder side | 0.067 inch (1.7 mm) maximum |
| Connectors | The 96-pin P1 backplane connector provides the interface to the VMEbus signals.<br>The 96-pin P2 backplane connector provides the interface to the extended VMEbus signals and VSB signals.<br>The 9-pin J1 front panel connector provides the interface to the Channel A serial port.<br>The 9-pin J2 front panel connector provides the interface to the Channel B serial port. |

## Cooling Requirements

The Motorola MVME165 VMEmodule is specified, designed, and tested to operate reliably with an incoming air temperature range from 0° to 55° C (32° to 131° F) with forced air cooling at a velocity typically achievable by using a 100 CFM axial fan. Temperature qualification is performed in a standard Motorola VMEsystem 1000 chassis. Twenty-five watt load boards are inserted in two card slots, one on each side, adjacent to the board under test, to simulate a high power density system configuration. An assembly of three axial fans, rated at 100 CFM per fan, is placed directly under the VME card cage. The incoming air temperature is measured between the fan assembly and the card cage, where the incoming airstream first encounters the module under test. Test software is executed as the module is subjected to ambient temperature variations. Case temperatures of critical, high power density integrated circuits are monitored to ensure component vendors specifications are not exceeded.

While the exact amount of airflow required for cooling depends on the ambient air temperature and the type, number, and location of boards and other heat sources, adequate cooling can usually be achieved with 10 CFM and 490 LFM flowing over the module. Less airflow is required to cool the module in environments having lower maximum ambients. Under more favorable thermal conditions, it may be possible to operate the module reliably at higher than 55° C with increased airflow. It is important to note that there are several factors, in addition to the rated CFM of the air mover, which determine the actual volume and speed of air flowing over a module.

## FFC Compliance

The MVME165 was tested in an FCC-compliant chassis, and meet the requirements for Class A equipment. FCC compliance was achieved under the following conditions:

1. Shielded cables on all external I/O ports.
2. Cable shields are connected to earth ground via metal shell connectors bonded to a conductive module front panel.
3. Conductive chassis rails connected to earth ground. This provides the path for connecting shields to earth ground.
4. All chassis and MVME165 front panel attachment screws are properly tightened.

For minimum RF emissions, it is essential that the conditions above be implemented; failure to do so could compromise the FCC compliance of the equipment containing the module.

# General Description

The MVME165 microcomputer is a VME/VSB-based CPU engine, utilizing the Motorola MC68040 microprocessor. The MC68040 features 8Kb of internal cache, a floating point unit, and a memory management unit.

The MVME165 includes the MC68681 DUART for serial I/O and software timing, the Local Resource Controller (LRC) for local CSR, counter/timers, and local interrupt source; a full VMEbus 32-bit master/slave interface, a VMEbus interrupt handler, interrupter, system controller and a global register set (MVME6000); a full VSB bus master/slave interface (MVSB2400); two 32-pin JEDEC standard sockets for EPROM, one 28-pin JEDEC socket for Time-of-Day Clock/Non-Volatile RAM (TOD/NVRAM), and a 4/16Mb DRAM.

The optional MVME714 transition module with two DB-25 connectors is available to connect two RS-232C devices to the MVME165. A 2400 baud modem is also available with the MVME714M, which provides one console port and one port for terminal/printer/modem.

## Related Documentation

The following publications are applicable to the MVME165 and may provide additional helpful information. If not shipped with this product, they may be purchased from Motorola, Inc, Computer Group, Technical Literature Center, 1919 West Fairmont Drive, Suite 8, Tempe, Arizona 85282; telephone: 1-800-458-6443; FAX: (602) 438-0240.

| Document Title | Motorola Publication Number |
|---|---|
| SIMVME165 VME/VSB Microcomputer VMEmodule Support Information | SIMVME165 |
| MVME165BUG 165Bug Debugging Package User's Manual | MVME165BUG |
| MVME165 Diagnostic Firmware User's Manual | MVME165DIAG |
| MVME714 and MVME714M Two-Channel Serial I/O Distribution Modules User's Manual | MVME714 |
| MC68040 Enhanced 32-Bit Microprocessor User's Manual | MC68040UM |
| MC68040 Designer's Handbook | MC68040DH |
| MC68681 Dual Asynchronous Receiver/Transmitter (DUART) Data Book | MC68681 |
| MC68881/MC68882 Floating-Point Coprocessor User's Manual | MC68881UM |
| MVME6000 (VMEchip) VMEbus Interface User's Manual | MVME6000UM |
| MVSB2400 VSBchip User's Manual | MVSB2400UM |
| VSB (bus) - The Parallel Sub System Bus of the IEC 821 Bus | MVMESB |

**NOTE:** Although not shown in the above list, each Motorola Computer Group manual publication number is suffixed with characters that represent the revision level of the document, such as /D2 (the second revision of a manual); each supplement bears the same number as the manual but has a suffix such as /A1 (the first supplement to the manual).

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

ANSI/IEEE Std 1014-1987
Versatile Backplane Bus: VMEbus

The Institute of Electrical and Electronics
Engineers, Incorporated
Publication and Sales Department
345 East 47th Street
New York, New York 10017-2633
Telephone: 1-800-678-4333

EIA RS-232C Serial Interface
Specification

Electronic Industries Association
2001 Eye Street N.W.
Washington, D.C. 20006
Telephone (202) 457-4900

## Support Information

The SIMVME165 contains the connector signal information, parts list, and schematic diagram for the MVME165.

The manual may be obtained free of charge from Motorola, Inc, Computer Group, Technical Literature Center, 1919 West Fairmont Drive, Suite 8, Tempe, Arizona 85282; telephone: 1-800-458-6443; FAX: (602) 438-0240.

## 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 a 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.

"Words" are 16-bit entities.

When individual bits are discussed in the text, "set" means 1 and "clear" means 0.

# CHAPTER 2
# HARDWARE PREPARATION AND INSTALLATION

## Introduction

This chapter provides the unpacking, hardware preparation, and installation instructions for the MVME165 VMEmodule.

## Unpacking Instructions

NOTE

If shipping carton is damaged upon receipt, request that the carrier's agent be present during unpacking and inspection of equipment.

Carefully unpack the equipment from the shipping carton. Refer to the packing list and verify that all items are present. Save the shipping carton and packing materials for storing or reshipping of the equipment.

CAUTION

**Avoid touching areas of integrated circuits. Static discharge can damage these components.**

Inspect the equipment for any shipping damage. If no damage exists, then the module can be prepared for operation according to the following sections of this chapter.

## Hardware Preparation

To select the desired configuration and ensure proper operation of the MVME165, certain modifications may be necessary before installation. These modifications are made through jumper block settings as described in the following sections. Therefore, the user should verify the jumper block configurations before installation and alter the jumpers, as required, for the user's particular system operation. The MVME165 has been factory tested and is shipped with factory-installed jumper configurations as illustrated in Figure 2-1.

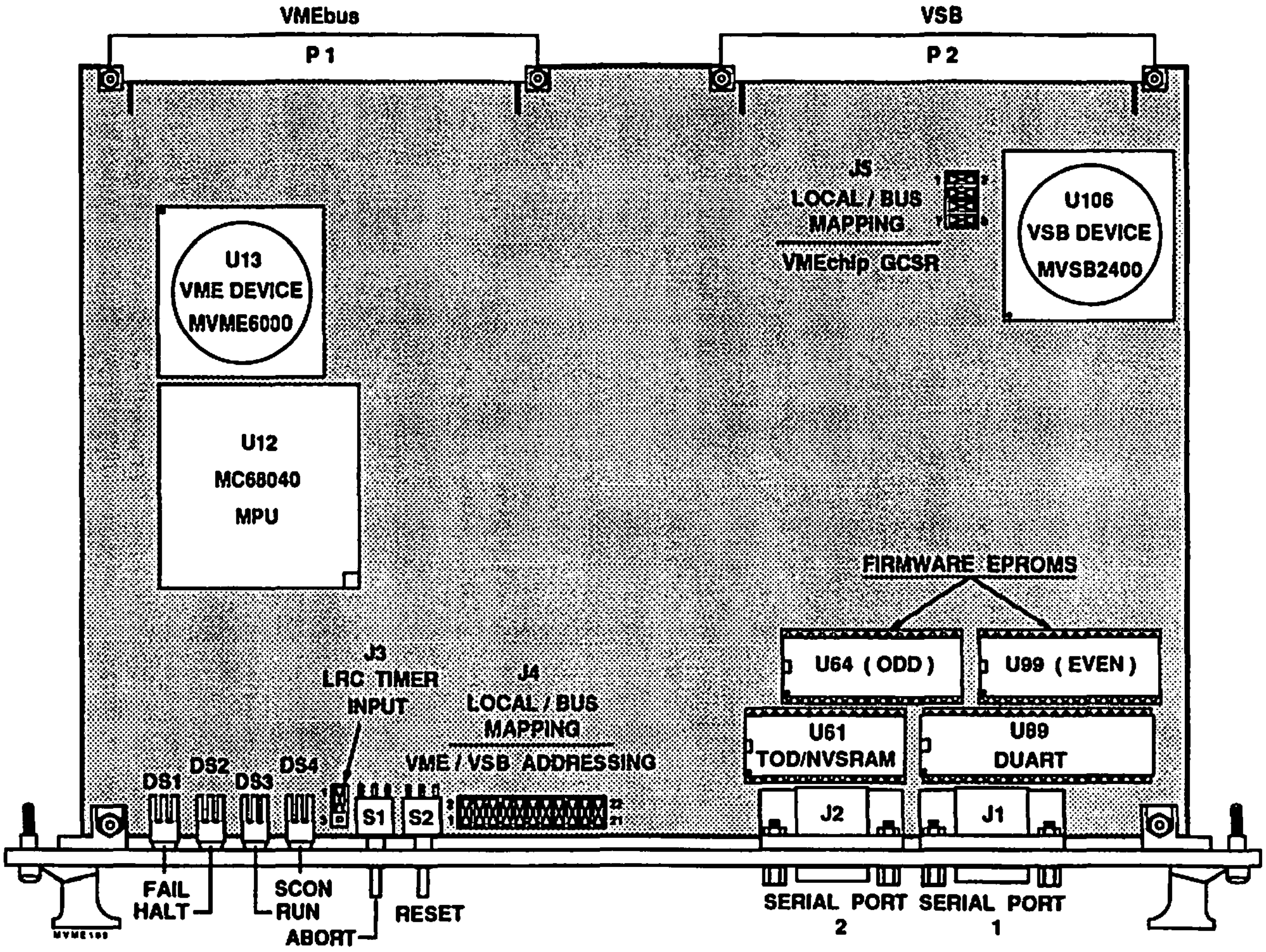


**Figure 2-1.** MVME165 Jumper, Switch, and Connector Location Diagram

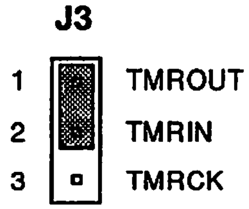
Table 2-1 lists the jumper blocks by designation, function, and factory-installed configuration. A more detailed description of these jumper blocks is provided in the following sections. In addition, four LED indicators (**FAIL, HALT, RUN,** and **SCON**) are located on the front panel. Refer to Chapter 3 for detailed information regarding the front panel and the use of these control indicators.

Table 2-1. Jumper Block Placements

| Jumper | Function | Factory Configuration |
|---|---|---|
| J1 | Serial Port 1 Connect | Front Panel Terminal Connector |
| J2 | Serial Port 2 Connect | Front Panel Host Connector |
| J3 | LRC Timer Input | J3 (1-2) |
| J4 | Local/Bus Mapping | J4 (1-2),(3-4),(5-6),(7-8),(9-10),(11-12),<br>(13-14),(15-16),(17-18),(19-20),(21-22) |
| J5 | Local/Bus Mapping | J5 (1-2),(3-4),(5-6),(7-8) |

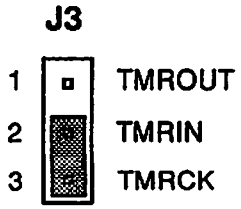
## LRC Timer Input (J3)

Jumper block J3 selects the source for the timer input pin (TMRIN) to the LRC's Timer Function. With a jumper cap across pins 1-2, the timer output pin (TMROUT) is connected to the TMRIN pin which allows separate timers within the LRC to be tied together for longer timer periods (refer to the *Timers* section in Chapter 4). With a jumper cap across pins 3-4, the timer clock (OP3) of the local DUART (MC68681) is connected directly to the TMRIN pin of the LRC for similar longer, but different time periods (refer to the *DUART Port Assignments* section in Chapter 4).



LRC Timer Output to
LRC Timer Input Connect

(FACTORY CONFIGURATION)

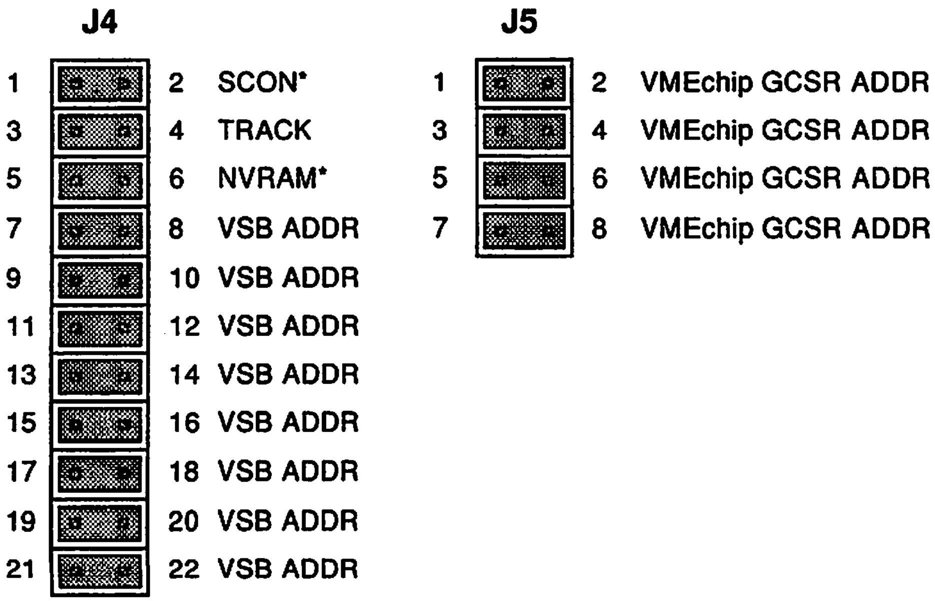


DUART Timer Clock (OP3) to
LRC Timer Input Connect

## Local/Bus Mapping (J4/J5)

Jumper blocks J4 and J5 provide the set-up and decode for the MVME165. These jumpers are readable and the 165Bug reads this set of jumpers at local address $FFFE0004 and then loads this information, or information from the NVRAM or ROM (depending on the TRACK and NVRAM* jumpers), into address FFFE0000 (165CSR), which then provides the selected decode range or particular option. Appendix A (MVME165 - 4Mb, versions -01 and -03) and Appendix B (MVME165 - 16Mb, versions -02 and -04) show how these jumpers are configured and the address range mapped for the two busses and the local DRAM.

The 165Bug default from the factory sets the local, VME and VSB address maps starting at $0. The VMEchip GCSR default 165CSR value is set at $A (see Figure 4-15), which programs the VME slave address of the VMEchip GCSR at $FFFFCAFx (x representing the individual register in the VMEchip, refer to Table 3-7).



Local/Bus Mapping

(FACTORY CONFIGURATION)

## Serial Port Cabling

The following tables provide the recommended cable connections for connecting the MVME165 serial ports to a terminal or modem interface. 165Bug expects the terminal to be attached to Serial Port 1. Serial Port 2 may be used as another terminal interface, or as a connection to a host system for upload/download using S-Record format.

**Table 2-2.** Serial Cabling To A Terminal

| MVME165<br>DB-9 Board Connector | | | RS-232C<br>DB-25 Unit Connector | |
|---|---|---|---|---|
| **Signal** | **Pin** | | **Pin** | **Signal** |
| RXDA | J1-2 | to | 2 | TXD |
| TXDA | J1-3 | to | 3 | RXD |
| CTSA | J1-8 | to | 4 | RTS |
| RTSA | J1-7 | to | 5 | CTS |
| DTRA | J1-4 | to | 8 | DCD |
| GND | J1-5 | to | 7 | SGND |
| DCDA | J1-1 | to | 20 | DTR |
| DSRA | J1-6 | to | 6 | DSR |

**Table 2-3.** Serial Cabling To A Modem

| MVME165<br>DB-9 Board Connector | | | RS-232C<br>DB-25 Unit Connector | |
|---|---|---|---|---|
| **Signal** | **Pin** | | **Pin** | **Signal** |
| TXDB | J2-2 | to | 2 | TXD |
| RXDB | J2-3 | to | 3 | RXD |
| RTSB | J2-7 | to | 4 | RTS |
| CTSB | J2-8 | to | 5 | CTS |
| DSRB | J2-6 | to | 6 | DSR |
| GND | J2-5 | to | 7 | SGND |
| DCDB | J2-1 | to | 8 | DCD |
| DTRB | J2-4 | to | 20 | DTR |

## VME Chassis Installation

The following section discusses installation of the MVME165 into a VME chassis. Ensure that EPROM devices are installed as needed. Factory configuration is with two EPROMs installed for the MVME165Bug debug monitor, in sockets U64 and U99. Ensure that all header jumpers are configured as desired.

a. Turn all equipment power OFF and disconnect power cable from ac power source.

**CAUTION**

**Inserting or removing modules while power is applied could result in damage to module components.**

**WARNING**

**DANGEROUS VOLTAGES, CAPABLE OF CAUSING DEATH, ARE PRESENT IN THIS EQUIPMENT. USE EXTREME CAUTION WHEN HANDLING, TESTING, AND ADJUSTING.**

b. The MVME165 may be installed into any double-high slot on a VMEmodule chassis. Make certain that the intended slot does not have I/O cabling on P2, since that could potentially damage the MVME165 VSB (bus) interface.

c. Using a firm grip on the module, slide the unit into the card slide until the P1 and P2 connectors of the unit align and seat into the backplane sockets. Use a firm, steady pushing motion to install the unit snugly into the backplane.

d. Install any other required VMEmodules in the system.

e. Connect power cable to ac power source and turn equipment power ON.

# CHAPTER 3
# OPERATING INSTRUCTIONS

## Introduction

This chapter provides necessary information to use the MVME165 in a system configuration. This includes operating controls, indicators, and memory map details found on the MVME165 microcomputer.

## Front Panel

The MVME165's front panel is illustrated in Figure 3-1. As shown, one red, one yellow, and two green LED indicators (**FAIL**, **HALT**, **RUN**, and **SCON**, respectively) are located on the top of the front panel. Located below the LED indicators are the two push-button switches S1 and S2 (**ABORT** and **RESET**, respectively). Below the push-button switches is a cutout for accessing the 22-pin jumper header J4. This is followed by the DB-9 connectors for Serial Ports 1 and 2.

### LED Indicators

The MVME165 has four LED indicators: **FAIL**, **HALT**, **RUN**, and **SCON**. They are described below. Table 3-1 provides the MVME165 status for all possible combinations of these LEDs.

The red **FAIL** LED indicator (DS1) is illuminated whenever the software detects an error and writes a "1" to the VMEchip's BRDFAIL bit.

The yellow **HALT** LED indicator (DS2) is illuminated whenever the MC68040 status lines PST3, PST2, PST1, and PST0 indicate a HALT condition (equal 0101).

The green **RUN** LED indicator (DS3) is illuminated whenever there is bus activity on the MC68040 bus, i.e., the TIP* signal is driven. Note that if all the cycles hit in the cache, the **RUN** LED is not ON, even though the MC68040 is running.

The green **SCON** LED indicator (DS4) is illuminated whenever the MVME165 is configured as the system controller; i.e., jumper cap J4(1-2) is installed.

### ABORT Switch (S1)

The **ABORT** switch is a momentary-action switch that causes a local interrupt to the processor when pressed (pressed = 0, not pressed = 1). The button is debounced and fed to an edge-detector. The edge-detector output (the assertion of the ABORT bit in the local CSR register) is used to generate the ABORT interrupt request. The ABORT

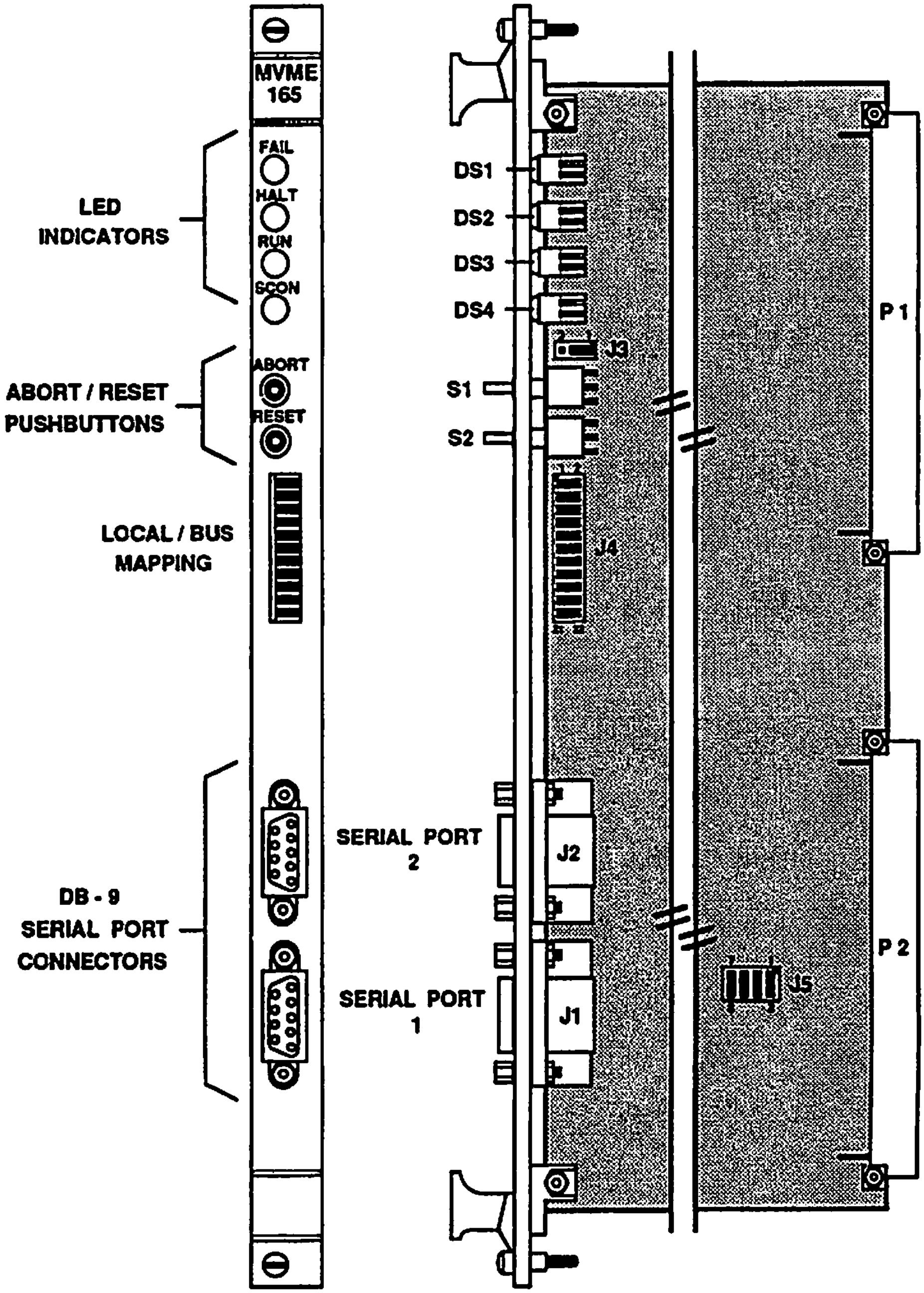


**Figure 3-1.** MVME165 Front Panel

interrupt is used to gain access to the 165Bug debugger firmware located in the MVME165 EPROMs. Refer to the *Interrupt Processing* section in Chapter 4 for more details.

**Table 3-1.** MVME165 Front Panel LED Status

| FAIL DS1<br>Red | HALT DS2<br>Yellow | RUN DS3<br>Green | SCON DS4<br>Green | MVME165 Status |
|---|---|---|---|---|
| OFF | OFF | OFF | OFF | No power is applied to the module, or the MPU is not the current local bus master. |
| OFF | OFF | ON/OFF | ON/OFF | Normal operation. |
| ON/OFF | ON | OFF | ON/OFF | The MC68040 is in a halted state. |
| ON | OFF | ON/OFF | ON/OFF | An error is detected by software and the VMEchip's BRDFAIL bit is ON. |

## RESET Switch (S2)

The **RESET** switch is a momentary-action switch which, when pressed (pressed = 0, not pressed = 1), generates a local reset (PRESET*) that is asserted for 200 milliseconds from the time that the switch is released. If the MVME165 is the system controller, then a VMEbus system reset (SYSRESET*) is also generated; it is asserted for 200 milliseconds from the time that the switch is released.

# Memory Maps

The following tables and text discuss the MVME165 memory maps.

## Main Memory Map

The physical addresses of the devices that respond to the MC68040 MPU are provided in Table 3-2.

Table 3-2. MVME165 Physical Address Memory Map

| Address Range | Devices Accessed | Port Size | Size | Notes |
|---|---|---|---|---|
| $FFFF0000 - $FFFFFFFF | VME Short I/O | A16,D32 | 64Kb | --- |
| $FFFF0008 - $FFFEFFFF | 165CSR (replicated) | --- | --- | --- |
| $FFFE0000 - $FFFE0007 | 165CSR | D32 | 8Kb | --- |
| $FFFD0040 - $FFFDFFFF | DUART (replicated) | --- | --- | --- |
| $FFFD0000 - $FFFD003F | DUART | D8 | 16b | 1 |
| $FFFC8000 - $FFFCFFFF | TOD/NVRAM (replicated) | --- | --- | --- |
| $FFFC7FE0 - $FFFC7FFF | Time-of-Day Clock | D8 | 8b | 1 |
| $FFFC0000 - $FFFC7FDF | NV Static RAM | D8 | 8184b | 1 |
| $FFFB0030 - $FFFBFFFF | VMEchip (replicated) | --- | --- | --- |
| $FFFB0000 - $FFFB002F | VMEchip (CSR) | D8 | 48b | --- |
| $FFFA0008 - $FFFAFFFF | reserved | --- | --- | --- |
| $FFFA0000 - $FFFA0007 | VSBchip CSR (A32) | D32 | 8b | --- |
| $FFF90040 - $FFF9FFFF | LRC (replicated) | --- | --- | --- |
| $FFF90000 - $FFF9003F | LRC-Local CSR | D32 | 64b | --- |
| $FFC00000 - $FFF8FFFF | reserved | --- | --- | --- |
| $FF800000 - $FFBFFFFF | ROM | D32 | 4Mb | 2 |
| $F1000000 - $FF7FFFFF | VSB/VMEbus (A32) | D32 | 232Mb | --- |
| $F0000000 - $F0FFFFFF | VSB/VMEbus (A24) | D32 | 16Mb | --- |
| $01000000 - $EFFFFFFF | VSB/VMEbus (A32) | D32 | 3814Mb | --- |
| $00000000 - $00FFFFFF | VSB/VMEbus (A24,A32) | D32 | 16Mb | --- |

**NOTES:** 1. Least significant byte of each longword.

2. LTO (Local Time Out) on writes.

The address range area from 00000000 to FF7FFFFF is for the VSB, VMEbus resources, and local DRAM. This address area can be allocated differently depending on the state of the bus mode select bit (VSBMD) of the CSR within the LRC. This bit is described in the *Bus Control Register* section in Chapter 4. The address map as it exists with each state of this bit is illustrated in Figures 4-2 and 4-3.

The jumpers at J4 and J5 are provided for user configuration of the local and bus mapping of the local DRAM. These jumpers do not directly determine how the memory will be decoded, but they are interpreted by the 165bug in certain conditions to determine where local memory should be mapped locally as well as from the VMEbus and VSB. These jumpers can be read at $FFFE0004, as illustrated in Figure 4-15.

Mapping is determined by setting bits in the 165CSR register at $FFFE0000 bits D0 through D14. The mapping possibilities for the Local DRAM as viewed from the MC68040 for both versions is provided in Table 3-3. Table 3-4 (MVME165 - 4Mb) and

Table 3-5 (MVME165 - 16Mb) provide the possible mapping configurations of the DRAM as viewed from the VMEbus. VSB mapping options are provided in Table 3-6.

**Table 3-3.** Local DRAM Mapping From The MC68040

| 165CSR | 165CRS | | | Local Access to DRAM from CPU | |
|---|---|---|---|---|---|
| HEX | Data Bits | | | (Starting Address) | |
| Value | D14 | D13 | D12 | MVME165 (4Mb) | MVME165 (16Mb) |
| $0 | 0 | 0 | 0 | $00000000 | $00000000 |
| $1 | 0 | 0 | 1 | $00400000 | $01000000 |
| $2 | 0 | 1 | 0 | $00800000 | $02000000 |
| $3 | 0 | 1 | 1 | $00C00000 | $03000000 |
| $4 | 1 | 0 | 0 | $01000000 | $04000000 |
| $5 | 1 | 0 | 1 | $01400000 | $05000000 |
| $6 | 1 | 1 | 0 | Local DRAM Access Disabled | |
| $7 | 1 | 1 | 1 | Local DRAM Access Disabled | |

**NOTE:** When the VMEbus tracking mode is selected, the debugger ignores the most significant bit of the VME mapping jumper (refer to Appendix A).

**Table 3-4.** MVME165 (4Mb Version) VME Addressing

| 165CSR VMEbus Bits | | | | | Slave Access to DRAM from VME | |
|---|---|---|---|---|---|---|
| HEX | Binary | | | | (Addressing Mode) | |
| D11-D08 | D11 | D10 | D09 | D08 | Extended (A32) | Standard (A24) |
| $0 | 0 | 0 | 0 | 0 | $00000000 - $003FFFFF | $xx000000 - $xx3FFFFF |
| $1 | 0 | 0 | 0 | 1 | $00400000 - $007FFFFF | $xx400000 - $xx7FFFFF |
| $2 | 0 | 0 | 1 | 0 | $00800000 - $00BFFFFF | $xx800000 - $xxBFFFFF |
| $3 | 0 | 0 | 1 | 1 | $00C00000 - $00FFFFFF | $xxC00000 - $xxFFFFFF |
| $4 | 0 | 1 | 0 | 0 | $01000000 - $013FFFFF | No Response |
| $5 | 0 | 1 | 0 | 1 | $01400000 - $017FFFFF | No Response |
| $6 | 0 | 1 | 1 | 0 | $01800000 - $01BFFFFF | No Response |
| $7 | 0 | 1 | 1 | 1 | $01C00000 - $01FFFFFF | No Response |
| $8 | 1 | 0 | 0 | 0 | $02000000 - $023FFFFF | No Response |
| $9 | 1 | 0 | 0 | 1 | $02400000 - $027FFFFF | No Response |
| $A | 1 | 0 | 1 | 0 | $02800000 - $02BFFFFF | No Response |
| $B | 1 | 0 | 1 | 1 | $02C00000 - $02FFFFFF | No Response |
| $C | 1 | 1 | 0 | 0 | $03000000 - $033FFFFF | No Response |
| $D | 1 | 1 | 0 | 1 | $03400000 - $037FFFFF | No Response |
| $E | 1 | 1 | 1 | 0 | $55400000 - $557FFFFF | No Response |
| $F | 1 | 1 | 1 | 1 | $AA800000 - $AABFFFFF | No Response |

**Table 3-5.** MVME165 (16Mb Version) VME Addressing

| 165CSR VMEbus Bits | | | | | Slave Access to DRAM from VME (Addressing Mode) | |
|---|---|---|---|---|---|---|
| HEX | Binary | | | | | |
| D11-D08 | D11 | D10 | D09 | D08 | Extended (A32) | Standard (A24) |
| $0 | 0 | 0 | 0 | 0 | $00000000 - $00FFFFFF | $xx000000 - $xxFFFFFF |
| $1 | 0 | 0 | 0 | 1 | $01000000 - $01FFFFFF | No Response |
| $2 | 0 | 0 | 1 | 0 | $02000000 - $02FFFFFF | No Response |
| $3 | 0 | 0 | 1 | 1 | $03000000 - $03FFFFFF | No Response |
| $4 | 0 | 1 | 0 | 0 | $04000000 - $04FFFFFF | No Response |
| $5 | 0 | 1 | 0 | 1 | $05000000 - $05FFFFFF | No Response |
| $6 | 0 | 1 | 1 | 0 | $06000000 - $06FFFFFF | No Response |
| $7 | 0 | 1 | 1 | 1 | $07000000 - $07FFFFFF | No Response |
| $8 | 1 | 0 | 0 | 0 | $08000000 - $08FFFFFF | No Response |
| $9 | 1 | 0 | 0 | 1 | $09000000 - $09FFFFFF | No Response |
| $A | 1 | 0 | 1 | 0 | $0A000000 - $0AFFFFFF | No Response |
| $B | 1 | 0 | 1 | 1 | $0B000000 - $0BFFFFFF | No Response |
| $C | 1 | 1 | 0 | 0 | $00000000 - $00FFFFFF | (Note 1) |
| $D | 1 | 1 | 0 | 1 | $01000000 - $01FFFFFF | (Note 2) |
| $E | 1 | 1 | 1 | 0 | $550000000 - $55FFFFFF | No Response |
| $F | 1 | 1 | 1 | 1 | $AA000000 - $AAFFFFFF | No Response |

**NOTES:**

1. $xx000000-$xx7FFFFF -- first half of DRAM.
2. $xx800000-$xxFFFFFF -- second half of DRAM.

**Table 3-6.** VSB Addressing

| 165CSR VSB Bits | | | | | Slave Access to DRAM from VSB (Starting Address) | |
|---|---|---|---|---|---|---|
| HEX | Binary | | | | | |
| D11-D08 | D07 | D06 | D05 | D04 | MVME165 (4Mb) | MVME165 (16Mb) |
| $0 | 0 | 0 | 0 | 0 | $00000000 | $00000000 |
| $1 | 0 | 0 | 0 | 1 | $00400000 | $01000000 |
| $2 | 0 | 0 | 1 | 0 | $00800000 | $02000000 |
| $3 | 0 | 0 | 1 | 1 | $00C00000 | $03000000 |
| $4 | 0 | 1 | 0 | 0 | $01000000 | $04000000 |
| $5 | 0 | 1 | 0 | 1 | $01400000 | $05000000 |
| $6 | 0 | 1 | 1 | 0 | $01800000 | $06000000 |
| $7 | 0 | 1 | 1 | 1 | $01C00000 | $07000000 |
| $8 | 1 | 0 | 0 | 0 | $02000000 | $08000000 |
| $9 | 1 | 0 | 0 | 1 | $02400000 | $09000000 |
| $A | 1 | 0 | 1 | 0 | $02800000 | $0A000000 |
| $B | 1 | 0 | 1 | 1 | $02C00000 | $0B000000 |
| $C | 1 | 1 | 0 | 0 | $03000000 | $0C000000 |
| $D | 1 | 1 | 0 | 1 | $03400000 | $0D000000 |
| $E | 1 | 1 | 1 | 0 | $554000000 | $550000000 |
| $F | 1 | 1 | 1 | 1 | $AA800000 | $AA000000 |

## VMEbus Short I/O Memory Map

The VMEchip Global Control and Status Register (GCSR) set appears at odd addresses in the VMEbus short I/O memory map. A map decoder in the VMEchip monitors the address and the address modifier lines and requests the VMEchip global register when they are selected.

The VMEchip GCSR base address is selected using a control register (GCSR base address configuration register) in the MVME165 Local Control and Status Register (165CSR) and the GCSR address register in the VMEchip (LCSR) as shown in the following table.

**Table 3-7.** VMEchip Slave Access to GCSR

| 165CSR GCSR Bits | | | | | VMEchip Slave |
|---|---|---|---|---|---|
| HEX | Binary | | | | Access to GCSR |
| D11-D04 | D03 | D02 | D01 | D00 | (4Mb and 16Mb) |
| $0 | 0 | 0 | 0 | 0 | $FFFF00yz |
| $1 | 0 | 0 | 0 | 1 | $FFFF01yz |
| $2 | 0 | 0 | 1 | 0 | $FFFF02yz |
| $3 | 0 | 0 | 1 | 1 | $FFFF03yz |
| $4 | 0 | 1 | 0 | 0 | $FFFF04yz |
| $5 | 0 | 1 | 0 | 1 | $FFFF05yz |
| $6 | 0 | 1 | 1 | 0 | $FFFF06yz |
| $7 | 0 | 1 | 1 | 1 | $FFFF07yz |
| $8 | 1 | 0 | 0 | 0 | $FFFF08yz |
| $9 | 1 | 0 | 0 | 1 | $FFFF09yz |
| $A | 1 | 0 | 1 | 0 | $FFFFCAyz |
| $B | 1 | 0 | 1 | 1 | $FFFFCByz |
| $C | 1 | 1 | 0 | 0 | $FFFF0Cyz |
| $D | 1 | 1 | 0 | 1 | $FFFF0Dyz |
| $E | 1 | 1 | 1 | 0 | $FFFF0Eyz |
| $F | 1 | 1 | 1 | 1 | Disabled |

**NOTES:**

1. y = VMEchip GCSR base address - 1 of 15 (LCSR of VMEchip).
2. z = Individual register in VMEchip GCSR. (Refer to the VMEchip manual for group and board address description).

# CHAPTER 4
# FUNCTIONAL DESCRIPTION

## Introduction

This chapter provides the overall block diagram level description for the MVME165 module. The general description provides an overview of the module, followed by a detailed description of each section of the module. The simplified block diagram for the MVME165 is illustrated in Figure 4-1.

## General Description

The MVME165 microcomputer is a VME/VSB-based CPU engine, utilizing the Motorola MC68040 microprocessor. The MC68040 microprocessor features 8Kb of internal cache, a floating point unit, and a memory management unit.

The MVME165 includes the MC68681 DUART for serial I/O and software timing, the Local Resource Controller (LRC) for local CSR, counter/timers, local resource decode and control, and interrupt handling; a full VMEbus 32-bit master/slave interface, a VMEbus interrupt handler, interrupter, system controller and a global register set (MVME6000); a full VSB bus master/slave interface (MVSB2400); two 32-pin JEDEC standard sockets for EPROM, one 28-pin JEDEC socket for Time-Of-Day Clock/Non-Volatile RAM (TOD/NVRAM), and a 4/16Mb DRAM array.

## Address Map

The physical address map for the MVME165 is provided in Table 3-2. The top 64Kb of the address map (FFFF0000-FFFFFFFF) is allocated for the VMEbus Short I/O. The 64Kb address range at FFFE0000 through FFFEFFFF (replicated) is allocated to the local CSR (Control Status Register). The 64Kb address range at FFFD0000 through FFFDFFFF (replicated) is allocated to the DUART. The 64Kb address range at FFFC0000 through FFFCFFFF (replicated) is allocated to the local Time-of-Day Clock and Non-Volatile static RAM. The 64Kb address range at FFFB0000 through FFFBFFFF (replicated) is allocated to the VMEchip's CSR. The 64Kb address range at FFFA0000 through FFFAFFFF is allocated to the VSBchip's CSR. The 64Kb address range at FFF90000 through FFF9FFFF (replicated) is allocated to the LRC (Local Resource Controller). The 4Mb address range at FF800000 through FFBFFFFF is allocated to the ROM.

It is up to the software running on the MVME165 to make sure that the MC68040 treats the I/O resources of the board like I/O and not like memory. If not instructed otherwise,

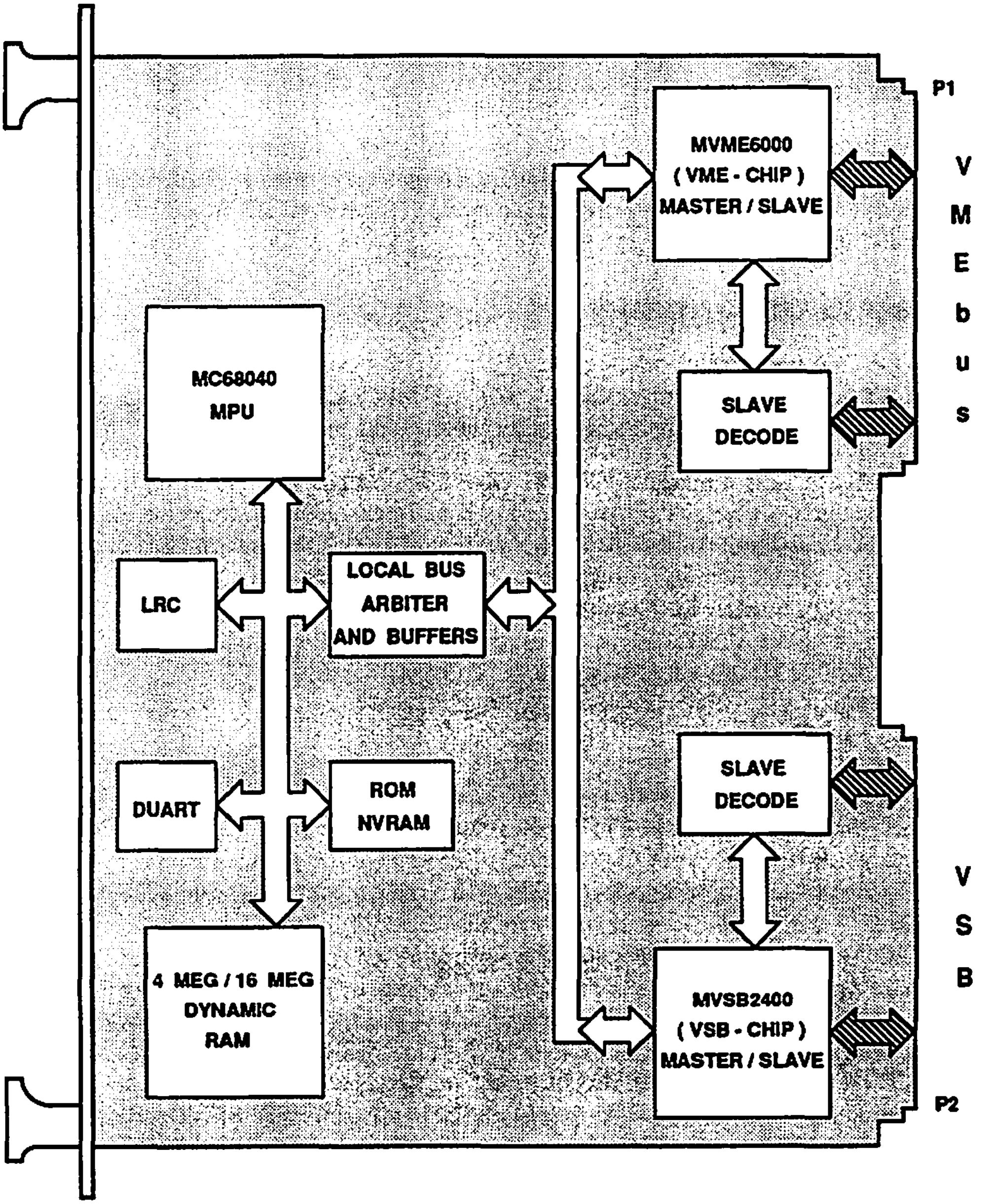


**Figure 4-1.** MVME165 Block Diagram

the MC68040 will use its highest-performance modes and will cache accesses to the I/O devices, resulting in improper operation.

One way to ensure proper I/O operation is to map the MVME165's I/O resources as non-cacheable, serialized using the MC68040's DTT0 or DTT1 registers. For example, the 165Bug debugger initializes the DTT1 register with a value to be treated as non-cacheable, serialized.

The address range area from 00000000 through FF7FFFFF is for the VSB, VMEbus resources, and local DRAM. This address area can be allocated differently depending on the state of the bus mode select bit (VSBMD) and the VSB enable bit (VSBEN) of the CSR within the LRC. These bits are described in the *Bus Control Register* section. The address map as it exists with each state of this bit is illustrated in Figures 4-2, 4-3, and 4-4.

Figure 4-2 represents the system address map for the MVME165 when the board is in the VSB priority mode (VSBMD = 0, VSBEN = 1; note that these two bits are in the LRC CSR). If the VSBchip is enabled and set for bounce mode (these control bits are in the VSBchip), then all accesses that can be reached via the VSB bus will proceed unless no slave responds.

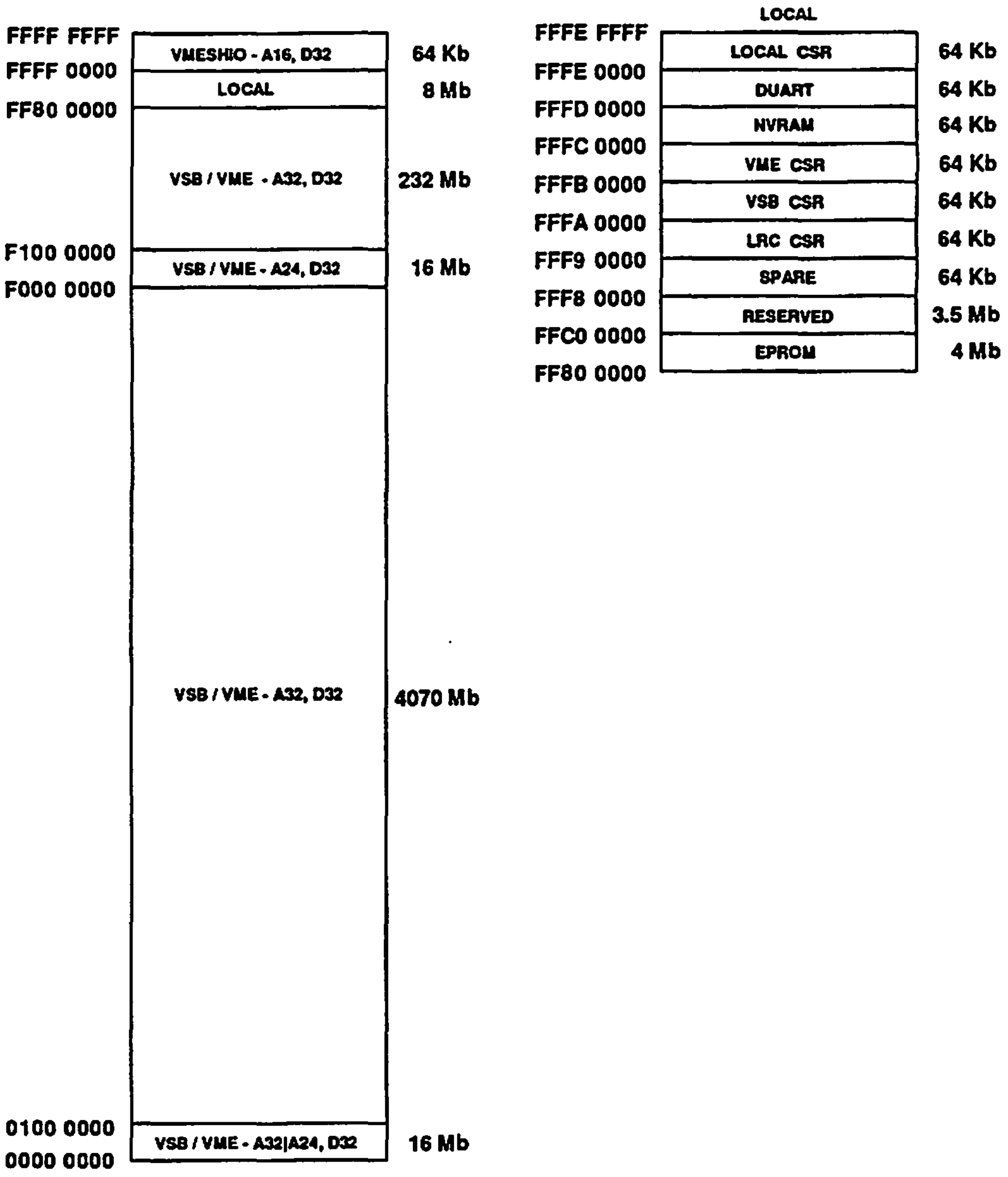
Figure 4-3 shows the MVME165 system address map when the bus select bit is asserted (VSBMD = 1, VSBEN = 1; note that these two bits are in the LRC CSR). In this mode some accesses are assumed to be addressed to the correct bus, and no "bounce" will be performed. When accessing address range 0 to 7FFFFFF or E0000000 through E3FFFFFF, accesses go to the VSB first (if the VSBchip is enabled).

Figure 4-4 shows the MVME165 system address map when the bus select bit is negated (VSBMD = don't care, VSBEN = 0; note that these two bits are in the LRC CSR). In this mode some accesses are assumed to be addressed to the correct bus, and no "bounce" will be performed. When accessing address range 0 to 7FFFFFF, accesses go to the VSB first (if the VSBchip is enabled).

For Figure 4-3, if no slaves responds, then the accesses bounce to the VMEbus. When accessing address range 80000000 through DFFFFFFF or E4000000 through FF7FFFFF, it goes to the VMEbus directly and no bounce is performed. Details of the address map and the resources are given in the following sections. All address references are given as physical addresses at the output side of the MC68040's MMU.

## Transfer Modifiers

The three transfer modifier and the two transfer type signals on the MC68040 specify the type of cycle being run. For normal (TT=0) and MOVE16 access, they specify, Data cache push Access (TM=0), User Data (TM=1), User Program (TM=2), PMMU



| | | |
|---|---|---|
| FFFF FFFF | | |
| | VMESHIO - A16, D32 | 64 Kb |
| FFFF 0000 | | |
| | LOCAL | 8 Mb |
| FF80 0000 | | |
| | VSB / VME - A32, D32 | 232 Mb |
| F100 0000 | | |
| | VSB / VME - A24, D32 | 16 Mb |
| F000 0000 | | |
| | VSB / VME - A32, D32 | 4070 Mb |
| 0100 0000 | | |
| | VSB / VME - A32\|A24, D32 | 16 Mb |
| 0000 0000 | | |

| | LOCAL | |
|---|---|---|
| FFFE FFFF | | |
| | LOCAL CSR | 64 Kb |
| FFFE 0000 | | |
| | DUART | 64 Kb |
| FFFD 0000 | | |
| | NVRAM | 64 Kb |
| FFFC 0000 | | |
| | VME CSR | 64 Kb |
| FFFB 0000 | | |
| | VSB CSR | 64 Kb |
| FFFA 0000 | | |
| | LRC CSR | 64 Kb |
| FFF9 0000 | | |
| | SPARE | 64 Kb |
| FFF8 0000 | | |
| | RESERVED | 3.5 Mb |
| FFC0 0000 | | |
| | EPROM | 4 Mb |
| FF80 0000 | | |

**Figure 4-2.** System Address Map (VSBMD = 0, VSBEN = 1)

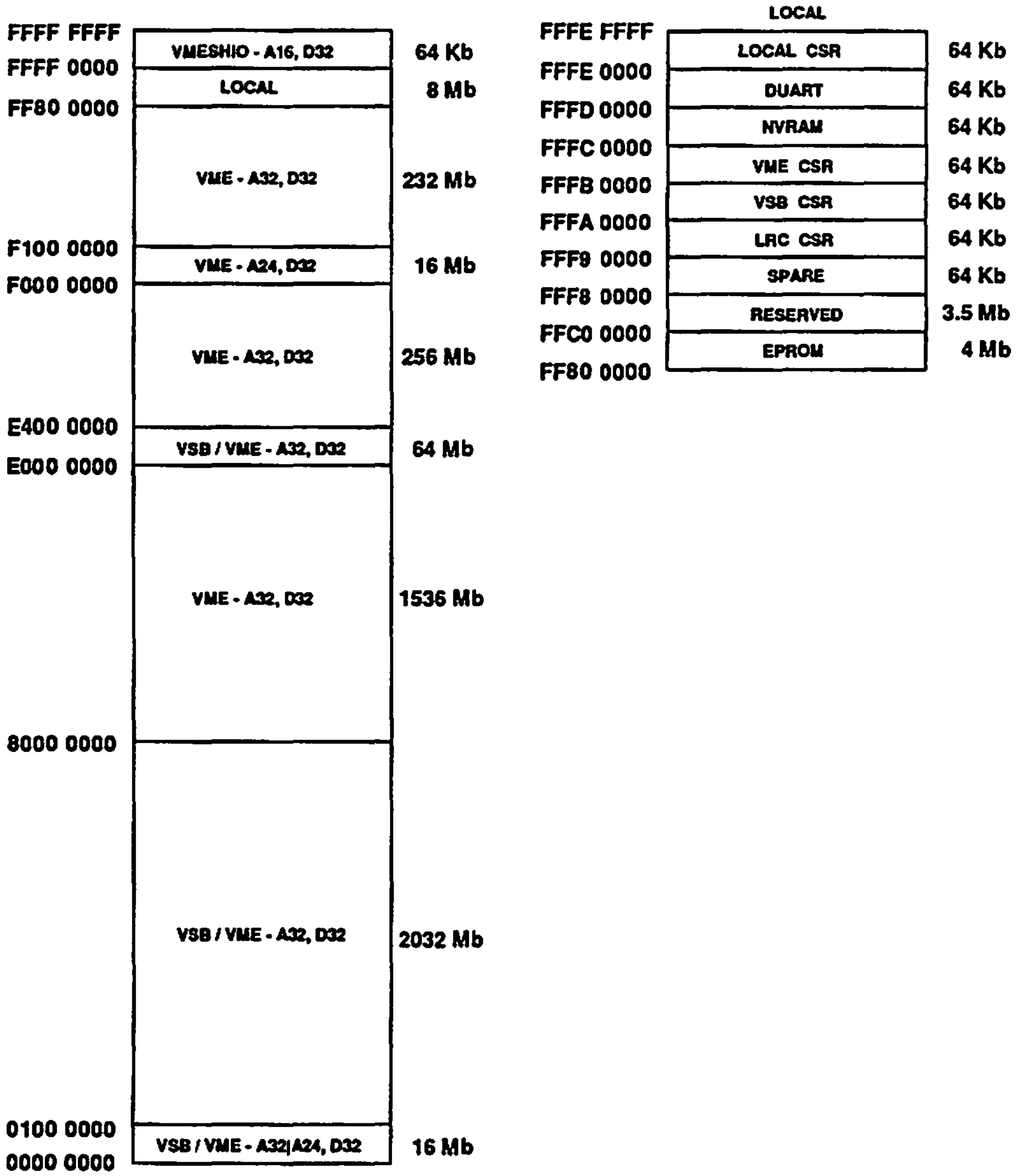


**Figure 4-3.** System Address Map (VSBMD = 1, VSBEN = 1)

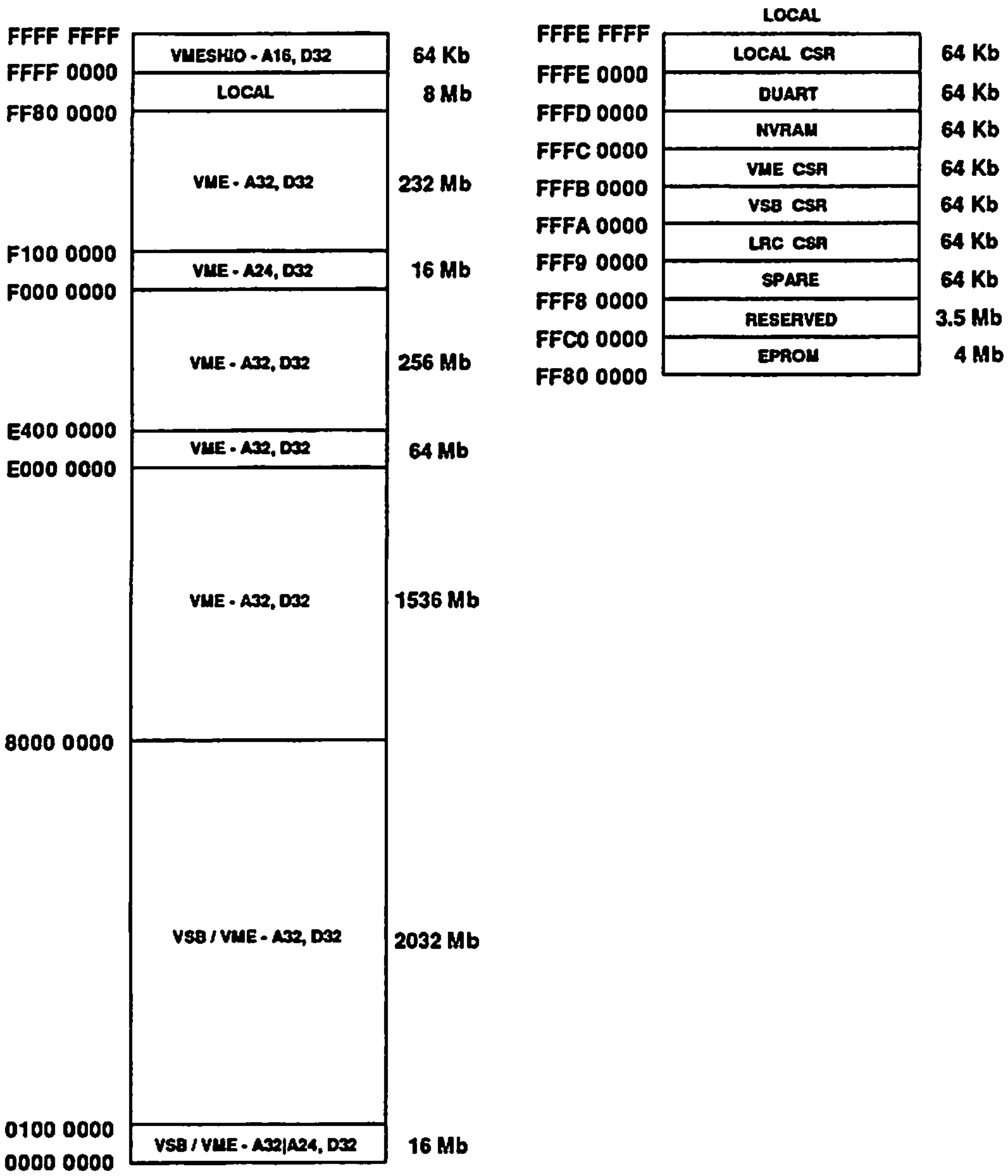


**Figure 4-4.** System Address Map (VSBMD = Don't Care, VSBEN = 0)

Tablewalk Data Access (TM=3), PMMU Tablewalk Code Access (TM=4), Supervisor Data (TM=5), and Supervisor Program (TM=6) cycle types, as well as a Reserved Cycle Type (TM=7). Processor cycles using any of the non-acknowledge access transfer types (TT not equal to 3) will select the resources of the address map shown in Figures 4-2 and 4-3.

For normal and MOVE16 access, the transfer modifiers are similar to the function codes in the MC68020 and MC68030. They are usually generated by the MC68040 automatically and are not directly specified by the user. The user can however, directly specify the function codes for data accesses through the MOVES instruction. The function codes 0, 3, and 4 are translated to a reserved Address Modifier on the VMEbus by the VMEchip that the target device may not respond to. If TM=0, TM=3, or TM=4 are asserted, they are translated into function codes 1, 2, or 5 to the VMEchip, respectively, so that data cache push access or PMMU table search can be performed on the VMEbus. Refer to the *ANSI/IEEE Standard 1014-1987 (VMEbus Specification)* for address modifier usage.

The interrupt acknowledge vectors are accessed automatically by the processor during interrupt acknowledge cycles. The interrupt acknowledge vectors appear normally at address FFFFFFFF with transfer type = 3 (TT=3). The interrupt level appears at the transfer modifier lines. This address normally is not directly specified by the user.

## ROM

The ROM is decoded starting at address FF800000. It is either 256Kb or 512Kb in size depending on whether 128K x 8 or 256K x 8 EPROMs are installed. The ROM is configured as a 32-bit wide (D32) data port for the processor, but the Local Resource Controller (LRC) orchestrates the access.

There are two ROM devices on the MVME165, each eight bits wide. The LRC provides the necessary control signals and storage latches so that two 16-bit accesses are performed to the ROM and then the LRC acknowledges the MC68040, thus providing 32 bits of data from a 16-bit wide port. It is a read-only resource, and any write access to the ROM address space will be terminated with a local bus timeout (LTO). The ROM is automatically selected after any board reset to supply the initial stack pointer and program counter to the processor and will remain selected until an access to the ROM address space is performed.

## Local Memory

### Time-of-Day Clock/NVRAM

There are two blocks of local RAM on the MVME165. One block of memory is configured in a Time-of-Day Clock/NVRAM. This Time-of-Day clock and Non-Volatile static RAM (MK48T08) is decoded at addresses FFFC0000 through FFFC7FFF. The TOD/NVRAM is 8Kb in size, and it is configured as an 8-bit wide data port for the processor. The Time-of-Day clock is located at the top of its address map from FFFC7FE0 through FFFC7FFF, with the Non-Volatile Static RAM being the remaining portion of the decoded space. The TOD/NVRAM can be accessed as a longword with the data in the least significant byte. It can also be accessed as bytes according to Table 4-1. The Time-of-Day clock address map, as implemented on the MVME165, is given in Table 4-1.

**Table 4-1.** Time-Of-Day Clock Address Map

| Address | Register Function | |
|---|---|---|
| FFFC7FFF | Year | (00-99) |
| FFFC7FFB | Month | (01-12) |
| FFFC7FF7 | Date | (01-31) |
| FFFC7FF3 | Day | (01-07) |
| FFFC7FEF | Hour | (00-23) |
| FFFC7FEB | Minutes | (00-59) |
| FFFC7FE7 | Seconds | (00-59) |
| FFFC7FE3 | Control | (01-31) |

The Non-Volatile static RAM portion is similarly addressed. The MK48T08 provides a Time-of-Day clock, oscillator, power fail detection, memory write protection, 8184 bytes of RAM, and a battery in one 28-pin package. The clock provides seconds, minutes, hours, day, date, month, and year in BCD 24-hour format. Corrections for 28, 29 (leap year) and 30 day months are automatically made. No interrupts are generated by the clock. The internal battery has a typical life span of 3 to 5 years when the clock is running and a minimum of 10 years when the clock is stopped. The life span of the battery also depends on the power on duty cycle. Refer to the *MK48T08 8K x 8 Non-Volatile Static RAM Data Sheet* for a complete description.

## DRAM Array

The second block of memory is implemented as a 4/16Mb DRAM array. In addition to the typical control logic necessary for a DRAM array, the data multiplexing for this array is handled by two identical ASICS called the MEMMUX (MEMory MUltipleXer). The MEMMUX is designed such that it can be used in pairs to allow multiplexing of a 128-bit wide memory to the 32-bit wide data path used by the MC68040. The MEMMUX also allows the MC68040 to write four 32-bit longwords to the 128-bit memory. This occurs when the MC68040 flushes one of its data cache lines or when the MC68040 move16 instruction is executed. The MEMMUX generates parity on all writes and can check parity when that feature is enabled. The parity control bits are in the LRC. Figure 4-5 illustrates the functional grouping of the signals in each MEMMUX gate array.

The two transfer size signals on the MC68040 specify the size of cycle being run. They specify, Byte Access (SIZ=1), Word Access (SIZ=2), Longword Access (SIZ=0), and Cache Line Access (SIZ=3).

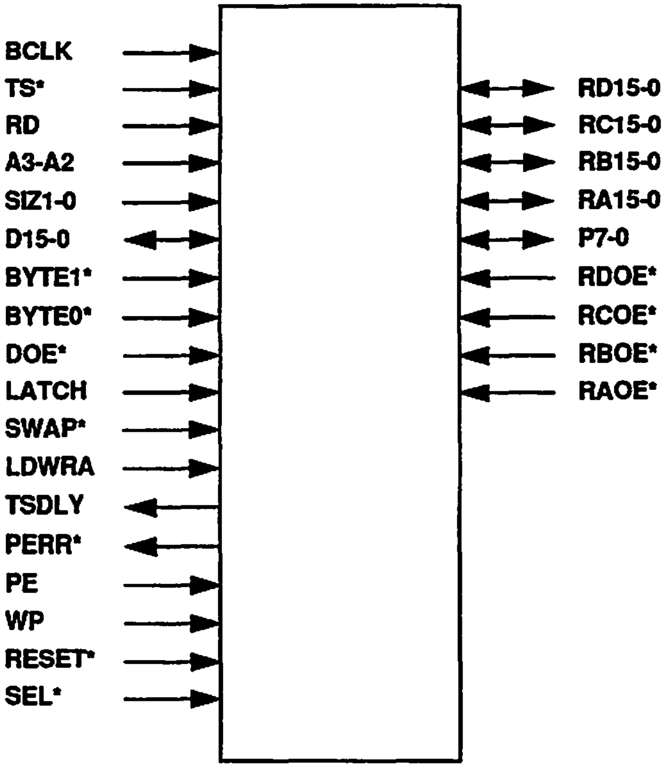


**Figure 4-5.** Memory Multiplexer (MEMMUX)

A byte swap buffer is supplied in the MEMMUX which can be used to swap data on (D7 through D0) to (D15 through D8) during a write or (D15 through D8) to (D7 through D0) on a read (but is not used on the MVME165). The VMEbus, for example, allows 16-bit masters which only drive D15 through D0. This requires 32-bit slaves to move data from these masters from D15 through D0 to memory data bits D31 through D16.

The MEMMUX provides data latches for both read data and write data. Write data is latched during each write cycle. Read data can be latched using the LATCH signal. The write latches are needed in order to capture the four longwords that the MC68040 puts out during its data cache line flush. The read latches are needed to allow data to be held during a read/modify/write cycle as well as during a cache line burst fill operation. The MEMMUX allows writes which involve less than four longwords to be written in a read/ modify/write cycle. To reduce the parts count the MVME165 uses X 4 DRAMS. The 25 MHz MVME165 uses 80 nanosecond DRAMs and a cache line (4 longwords) burst fill completes in 4-1-1-1 processor clock cycles, when parity error checking is disabled. With parity error checking enabled, access time is 5-1-1-1 clock cycles. A burst write to the DRAM occurs in 2-1-1-1 clock cycles, independent of parity selection.

The MEMMUX determines which of the eight memory bytes to fetch or write by using the (A3, A2, SIZ1, SIZ0, BYTE1*, and BYTE0*) signals. A3 and A2 inform the MEMMUX which of the four 16-bit words are involved in the current cycle. BYTE1* and BYTE0* pick which one of the two, or both, bytes are involved. SIZ1 and SIZ0 are used to determine whether this is a byte, word, longword, or cache line transfer. When a cache line transfer is indicated the MEMMUX uses A3 and A2 to determine the starting longword of the cache line. The MEMMUX increments A3 and A2 internally for each of the three subsequent longwords.

The BCLK signal, which drives the MC68040, is also used by the MEMMUXs to synchronize data transfers with the MC68040, which uses the rising edge to drive or accept data.

## Local Resource Controller (LRC)

The LRC is decoded at addresses FFF90000 through FFF9003F. It is replicated throughout the rest of the 64Kb address space (FFF90000-FFFFFFFF) and configured as a 32-bit wide (D32) data port for the processor. The Local Resource Controller provides major decoding for local peripherals, other than the local DRAM array, as well as implement timers and some general purpose functions into a small gate-array package.

Table 4-3 provides the map that the LRC decodes. Figure 4-6 illustrates the functional grouping of the signals in the LRC.

### MC68040 Interface

The LRC receives and drives the necessary signals to provide local control of multiple devices as well as control of the functions of the LRC itself. The address lines of A2-A5 and A16-A31 and data lines D0-D23 are connected to the LRC. Other control lines interfacing to the LRC are TT0-TT1, RD, TS*, TM0-TM2, RESET*, BCLK, IPL0*-IPL2*, TA*, TEA*, and TBI*. Note that the LRC data bus is 24 bits wide and that the SIZ0-SIZ1 signals are not monitored and therefore all accesses should be longwords.

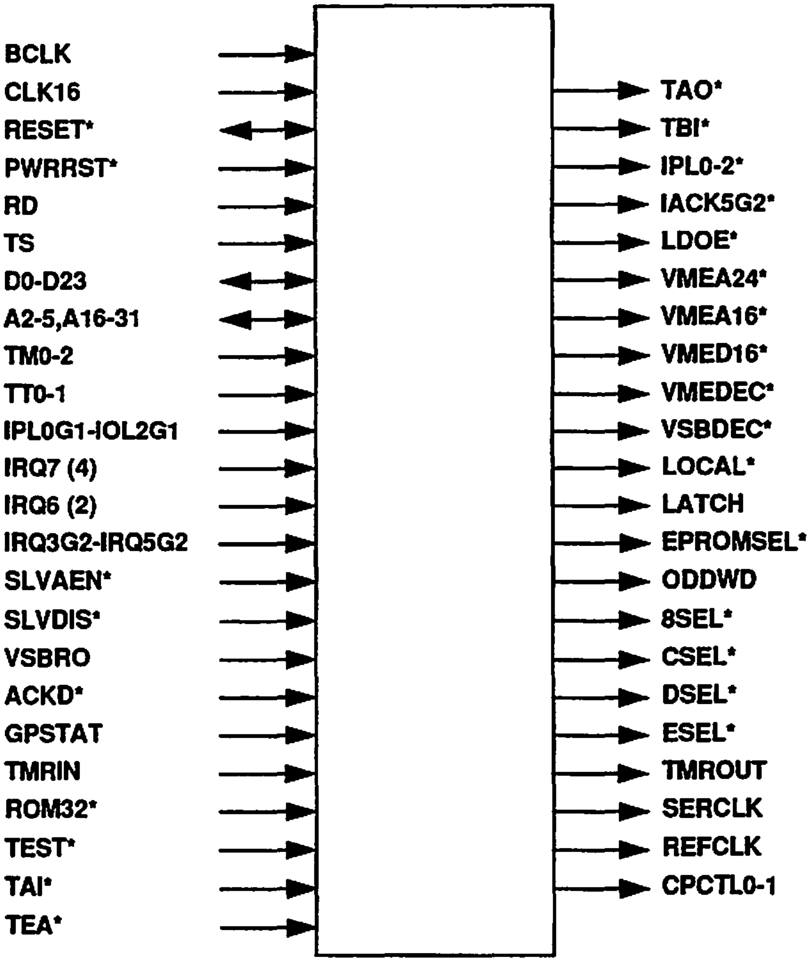


**Figure 4-6.** Local Resource Controller (LRC)

## Interrupt Handler

The LRC provides an interrupt handler for local interrupt control. This handler is programmed via the Interrupt Control Registers 0 ($2C) and 1 ($30) in the LRC CSR and status is accessed via the Status Register ($34). These registers are described in detail in the *Interrupt Control and Status Register* section. Note that the timer interrupt enable control bit is in the Timer Control Registers 0 through 2 ($18, $1C and $20) described in the *Timers* section. The handler interfaces directly with the MC68040 microprocessor. More specific descriptions of all interrupts are detailed in the *Interrupt Processing* section.

The LRC provides a unique interrupt vector for each interrupt source. IRQ5G2 (DRTIRQ), IRQ4G2 (VSBIRQ), and all Group 1 IRQs provide their own interrupt vector during an IACK cycle. The edge sensitive interrupt sources are only cleared by an IACK cycle, but the status is cleared by writing to the Status Register ($34) which is defined in the *LRC Control Status Registers* section. Refer to Table 4-9 for the MVME165 interrupt assignments. The most significant interrupt vector bits (VEC4 through VEC7) are software programmable via the Interrupt Control Register 1 at address $30 of the LRC CSR. VEC0 through VEC3 are assigned by the LRC as shown in Table 4-2.

**Table 4-2.** Interrupt Vector Data Bits (Least Significant)

| IRQ | VEC3 | VEC2 | VEC1 | VEC0 | HEX |
|---|---|---|---|---|---|
| TMR0IRQ * | 0 | 0 | 0 | 0 | 0 |
| TMR1IRQ * | 0 | 0 | 0 | 1 | 1 |
| TMR2IRQ * | 0 | 0 | 1 | 0 | 2 |
| IRQ7G2 (not used) | 0 | 1 | 0 | 0 | 4 |
| TICKTMR (IRQ6G2) ** | 0 | 1 | 0 | 1 | 5 |
| TICKTMR (IRQ3G2) ** | 0 | 1 | 1 | 1 | 7 |
| PARERR | 1 | 0 | 0 | 0 | 8 |
| VSBIRQ (IRQ6G4) *** | 1 | 0 | 0 | 1 | 9 |
| ABORT | 1 | 0 | 1 | 0 | A |
| ACFAIL | 1 | 0 | 1 | 1 | B |

**NOTES:** * = These are LRC timers.

** = TICKTMR. TICKTMR is the OP3 output of the MC68681 (refer to the DUART OP3 description). It is connected to both the IRQ6G2 and IRQ3G2 inputs to the LRC. Only one of these two interrupts should be enabled.

*** = VSBIRQ. VSBIRQ is connected to both IRQ4G2 and ITRQ6G4. Only one of them should be enabled at a time. If IRQ4G2 is enabled, the external device should provide its own vector during IACK cycle. If IRQ6G4 is enabled, the LRC provides the vector.

Care must be taken to initialize the upper nibble of the vector (VEC4 through VEC7), which are located in the Interrupt Control Register 1 ($30) of the CSR. If this register has not been written before an interrupt occurs, then the LRC provides $0F as the vector.

## Local Device Control

Decode is provided within the LRC to allow certain accesses to be routed to the VMEbus and VSB bus via the bus control chips onboard the MVME165. Specifically, VMEA16*, VMEA24*, VMED16* types of transfers as well as general VMEDEC* and VSBDEC* decode lines are provided. Local decode lines supported are addressed at $FFF8XXXX through FFFEXXXX, specifically signals are 8SEL*, CSEL*, DSEL*, and ESEL*. EPROM decode is also provided for the address range from $FF8XXXXX through $FFBXXXXX (4Mb). The signals LATCH and ODDWORD are also provided to allow 32-bit transfers to the MC68040 from 16-bit wide EPROMS, as implemented on the MVME165. See Figure 4-7 for the local memory map. ESEL* is used for access to the MVME165 Local Control and Status Registers. Refer to the *MVME165 Control and Status Registers* section for further details.

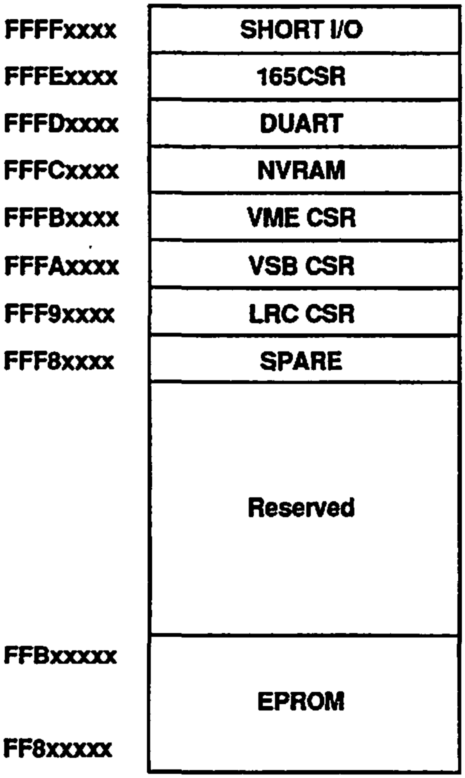


**Figure 4-7.** LRC Memory Map

## Clocks

The BLCK signal is used by the LRC to synchronize data transfers with the MC68040. This signal is tied directly to the MC68040's BCLK. Since BCLK can be one of several frequencies, this clock affects internal LRC timing; specifically the select lines 8SEL*, CSEL*, DSEL*, ESEL*, and EPROMSEL*, as well as when TA* is asserted. A 16 MHZ clock is also necessary for various functions within the LRC, such as the three timers, the serial clock generation and refresh clock generation.

## LRC Control Status Registers

The LRC has sixteen (16) longword addresses at $FFF9XX00 to $FFF9XX3F that perform the functions shown in Table 4-3. All references to CSR locations in the following descriptions will be referred to by their offset from $FFF900xx. Note that all accesses to the LRC's CSR will return $FF in the most significant byte (D24 through D31) because the LRC does not drive those lines. Most of the CSRs are cleared to "0" at reset time. Some status bits in the status register are set or cleared during power-on-reset only (POR).

**Table 4-3.** MVME165 LRC Address Map

| Address | Register | Accessible |
|---|---|---|
| FFF90000 | Timer 0 time constant | R/W |
| FFF90004 | Timer 1 time constant | R/W |
| FFF90008 | Timer 2 time constant | R/W |
| FFF9000C | Timer 0 current count | R |
| FFF90010 | Timer 1 current count | R |
| FFF90014 | Timer 2 current count | R |
| FFF90018 | Timer 0 control | R/W |
| FFF9001C | Timer 1 control | R/W |
| FFF90020 | Timer 2 control | R/W |
| FFF90024 | General control | R/W |
| FFF90028 | Bus control | R/W |
| FFF9002C | Interrupt control 0 | R/W |
| FFF90030 | Interrupt control 1 | R/W |
| FFF90034 | Status register | R,W/C |
| FFF90038 | RAM address | R/W |
| FFF9003C | Bus error status | R/C |

**NOTE:** Registers $0C, $10, and $14 are Read Only while $3C is a read to clear and all other locations can be read and written. These registers are explained in detail in the following sections.

## Timers

The LRC provides three 24-bit programmable timers. These timers are controlled by loading the appropriate Time Constant Register as shown in Table 4-3 and the LRC Timer Control Registers shown in Figure 4-8. The Current Count Registers for timers 0, 1 and 2 are located at offset $0C, $10, and $14, respectively. These timers are incremented at a 1 microsecond interval and they start counting from one. When the timer counts up to the number programmed in the Timer Time Constant Register, the corresponding timeout status bit is set in the Status Register ($34). If the interrupt enable bit is set in the Timer Control Register and global interrupt is enabled, an interrupt is generated. Timer 0 generates a Level 7 interrupt, Timer 1 a Level 6, and Timer 2 a Level 5.

Bits 4 through 7 of the Timer Control Register control timer software counting functions. TEN0 through TEN2 enables the specific timer counting; TLDx loads each timer to the starting count of one; TCYCx controls each timers cycling mode and TxIE controls whether each timer's interrupt is masked.

Bits 0 through 3 of the Timer Control Register 0 control the timer output function. Any one of the three timers can generate a timer output through the timer output pin (TMROUT), if enabled by setting the TOEN bit. The TMROUT pin either toggles or generates a 1 millisecond pulse when the timer times out. This option is controlled by the TOMD bit.

Bits 0 through 3 of the Timer Control Register 1 control the timer input function. Any one of the three timers can receive the timer input pin (TMRIN). The selected timer counts timer input pulses or counts TMRIN's active high duration. This option is controlled by the TISx bits.

Bits 1 through 3 of the Timer Control Register 2 control the watchdog timer function. Any one of the three timers can be programmed to be a watchdog timer. When it times out, it can assert the reset signal pin.

$18 TCTL0 (TIMER CONTROL REGISTER 0, R/W)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| TEN0 | TLD0 | TCYC0 | T0IE | TOS1 | TOS0 | TOEN | TOMD |

$1C TCTL1 (TIMER CONTROL REGISTER 1, R/W)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| TEN1 | TLD1 | TCYC1 | T1IE | TIS1 | TIS0 | TIEN | TIMD |

$20 TCTL2 (TIMER CONTROL REGISTER 2, R/W)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| TEN2 | TLD2 | TCYC2 | T2IE | TWD1 | TWD0 | TWDEN | TTST |

**Figure 4-8.** LRC Timer Control Registers

The Timer Control Bits are described below:

### TOMD Timer Output Mode

0: Square wave output (high for timeout period and low for timeout period).

1: Pulse output (width = 1 microsecond).

At the end of timeout, a 1 microsecond active high pulse is generated.

### TOEN Timer Output Enable

When TOEN is set (= 1), the TMROUT pin toggles as programmed by TOMD. If it is cleared (= 0), the TMROUT pin does not toggle.

### TOS1,0 Timer Output Select 1 and 0

00: Select timer 0 to control TMROUT pin.

01: Select timer 1 to control TMROUT pin.

10: Select timer 2 to control TMROUT pin.

11: Test mode only.

### T0IE-T2IE Timer Interrupt Enable 0, 1, and 2

Allows the appropriate timer to generate an interrupt on timeout. If the TxIE enable bit is cleared, the associated timer interrupt is masked. The timeout status is recorded in the status register whether the timer interrupt is enabled or not.

### TCYC0,1,2 Timer Cycle 0, 1, and 2

Automatically reloads timer at terminal count to start counting from one again. If the TCYCx bit is cleared, the timer continues counting at terminal count.

### TLD0,1,2 Timer Load 0, 1, and 2

0: Do not reload, continue counting (if timer is enabled).

1: Load timer at starting count (one) when this bit goes from 0 to 1. This bit is not self clearing.

### TEN0,1,2 Timer Enable 0, 1, and 2

Allows timer to count. When enabled, the associated timer starts counting; when the bit is cleared, the associated timer stops counting.

### TIMD Timer Input Mode

This timer input mode bit controls how the TMRIN (timer input pin) is used. When the input pin signal is at logic 1 and the Timer Enable bit is on, then the timer enabled counts and when the signal is at logic 0, the timer is stopped.

### TIEN Timer Input Enable

When this bit is set, the selected timer counts TMRIN pulses or use the TMRIN for hardware timer enable functions depending on the Timer Input Mode (TIMD). When TMRIN is high and the software time enable bit is set, the associated timer is enabled for counting.

0: Count rising edge pulses at input pin.

1: Hardware timer enable.

### TIS1,0 Timer Input Select 1 and 0

00: Select timer 0 to use TMRIN pin.

01: Select timer 1 to use TMRIN pin.

10: Select timer 2 to use TMRIN pin.

11: None

### TTST Timer Test

This control bit is used for the LRC chip test only. For normal timer operation, clear this bit.

### TWDEN Timer Watchdog Enable

When the watchdog function is enabled, the LRC generates a 4 microsecond reset pulse when the selected timer is timed out. This bit controls the watchdog reset function. If a watchdog with interrupt generated on timeout is desired (instead of reset), simply select one timer with the appropriate interrupt level, enable its interrupt feature and use that for the interrupt generated watchdog.

### TWD1,0 Timer Watchdog Select 1 and 0

00: Select timer 0 to generate reset on timeout.

01: Select timer 1 to generate reset on timeout.

10: Select timer 2 to generate reset on timeout.

11: None of the timers generates a watchdog reset.

### General Control Register (GCR)

The LRC provides one of two choices to output a serial clock for an external serial port controller chip. The two frequencies supported are 3.6864 MHz and 2.4576 MHz. The MVME165 uses an MC68681 DUART; therefore, the LRC should be programmed to output the 3.6864 MHz source; i.e., SCFREQ = 0. Control for this feature is programmed in the General Control Register ($24), as illustrated in Figure 4-9.

The LRC includes several other general purpose bits and functions. There are two hardware/software bits (GPCTL0 and GPCTL1), an EPROM acknowledge speed bit, and a RESET output bit. A full description of the GCR follows. Although no control bit is provided, an output pin provides a DRAM Refresh Clock for a 15 microsecond refresh rate. On the MVME165, GPCTL0 is connected to Parity Error Enable and GPCTL1 is connected to Write Wrong Parity.

$24 GCR (GENERAL CONTROL REGISTER, R/W)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| --- | --- | --- | SCFREQ | FAST | GPCTL1 | GPCTL0 | RSTO |

**Figure 4-9.** General Control Register

The General Control Bits are described below:

### RSTO Reset Output (asserted if = 1)

When writing a one to this bit, the LRC asserts reset for 4 microseconds. For the MVME165, the LRC reset is connected so that it only resets local resources. It does not generate SYSRESET even if it is the system controller. (To generate a whole system reset, use the SRESET bit in the VMEchip). This bit is self-cleared to "0" after the reset sequence.

### GPCTL0,1 General Purpose Control Output 0 and 1

Bit 0 - PARITY error enable = 1.

This bit, when set to "1", enables parity error detection. It is cleared on power up or reset, therefore, parity error checking is disabled. The parity enable bit together with the parity interrupt enable bit in the LRC Interrupt Control Register 0 (ICTL0), the PARBERRDIS bit in the MVME165 Local Control the Status Register (FFFE0000) determine whether and how the MVME165 reports a parity error. The MVME165 always generates parity on writes, but only does parity error checking if the parity enable bit is set. If the parity detection is enabled, the parity error status is updated in the LRC Status Register PARERR bit. If the PARBERRDIS bit is cleared, the parity enable bit is set and the MVME165 generates a bus error on parity error detection. If the parity enable bit is set and the parity interrupt enable is set, a level 7 interrupt is generated on parity error detection. If all three bits are set, both bus error and interrupt can be generated.

Bit 1 - Write wrong parity = 1.

This bit, when set to "1", causes a wrong parity to be written into the parity RAMs on write cycles. It is cleared on power up or reset.

### FAST EPROM Speed Select (BCLK frequency dependent)

FAST - EPROM FAST Acknowledge = 1.

25 MHz -- 150 nanoseconds EPROMs.
33 MHz -- 85 nanoseconds EPROMs.

SLOW - EPROM SLOW Acknowledge = 0.

25 MHz -- 250 nanoseconds EPROMs
33 MHz -- 200 nanoseconds EPROMs

This bit, together with the BCLK frequency, determines the timing of the TA* generation for EPROM accesses. Two EPROM speeds are provided. From power on or reset, the slower speed is selected.

### SCFREQ Serial Clock Frequency Select

0: Select 3.6864 MHz (MVME165)

1: Select 2.4576 MHz

## Bus Control Register

The LRC provides several bits for bus control on the VMEbus and the VSB interfaces. These bits, together with the VSBchip and VMEchip CSRs and control signals, control the way the MVME165 accesses the VMEbus and VSB. Figure 4-10 describes these bits.

$28 BCR (BUS CONTROL REGISTER, R/W)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| --- | --- | --- | VA24 | VSBEN | VSBMD | ROEN | ROMD |

**Figure 4-10.** Bus Control Register

The Bus Control Bits are described below:

### ROMD Read Only Mode

When the Read Only option is selected (i.e., ROEN=1), this bit controls the scope of the VSB read only function. When the Read Only option is selected, make sure to program the READONLY* bit in the VSBchip to a "0", and set the ROEN bit in the LRC bus control register.

0: Read only for entire Bus.

1: Controlled by the VSBRO input pin.

On the MVME165, the VSBRO input pin is not connected to any control signal and it is pulled down to a "0".

### ROEN VSB Read Only Enable

This bit should be set consistent with the VSBchip READONLY* bit as described above.

### VSBMD VSB Mode (see VSBEN below)

### VSBEN VSB Enable

The VSBEN and VSBMD bits in the LRC bus control register control the VSB access together with the VSBchip control bits. When the VSBEN bit in the VSBchip is cleared, all the master accesses to the VSB are disabled. If the VSBEN bit in the VSBchip is enabled and A31 = 0, a separate high speed decode path is used on the MVME165. When the VSBEN bit in the VSBchip is set and the VSBEN bit in the LRC is set, the VSBMD bit determines the scope of the VSB. It is advised to keep the VSBEN bit programmed exactly the same in the VSBchip and in the LRC.

VSBEN = 0: VSB is disabled.

| | |
|---|---|
| VSBEN = 1, VSBMD = 0: | entire bus map enabled for VSB access. |
| VSBEN = 1, VSBMD = 1: | VSB is enabled for A31 = 0 and for address range E0000000 through E3FFFFFF. |

**NOTE**

When A31 = 0, a separate high speed decode path is used on the MVME165. Also within the VSBchip there is a control signal also called VSBEN and it will have an effect on offboard accesses. When the VSBEN bit is cleared in the VSBchip, all the accesses to VSB are disabled. When the bit is set and the LRC VSBEN bit is set, the VSBMD bit determines one scope of the VSB. Refer to the *MVSB2400 VSBchip User's Manual* for further details.

### VA24 VMEbus Address 24-Bit Option

The VA24 control bit selects the address mode for VMEbus accesses in the lower 16 megabytes of the address map. Negating VA24 (VA24 = 0) selects the A32 address mode when accessing address range 00XXXXXX in which one of the extended VMEbus address modifiers 0E, 0D, 0A, or 09 will be generated on the bus. These address modifiers specify that the address bits A1 through A31 on the bus contain valid address information. Asserting VA24 (VA24 = 1) selects the A24 address mode in which one of the standard VMEbus address modifiers 3E, 3D, 3A, or 39 will be generated instead. These address modifiers specify that only the address bits A1 through A23 on the bus contain valid address information, and that the VMEbus address bits A24 through A31 should be ignored by the slave when decoding the address. The specific one of four address modifiers used is determined by the function codes of the processor during the bus access. Any VMEbus access above the lower 16 megabyte boundary of the address map will be performed in A32 mode.

The VA24 control bit allows for mix-mode operation in which A24 and A32 VMEmodules can be used in the same system. There is a similar bit in the VMEchip LCSR called MAS24 which can override the VA24 control bit and configure the entire VMEbus for A24 (16 megabytes) operation only. Generally, the use of the VA24 control bit is preferred over the MAS24 bit where A24 operation is desired in an A32-capable system.

### Interrupt Control And Status Register

The LRC provides three registers for interrupt control and status. Note that the interrupt enable bits unmask the associated interrupt. An edge triggered interrupt source stays pending until the interrupt is serviced. Clearing the interrupt status does not take away the pending IRQ. The IACK cycle only clears the interrupt request. Figures 4-11, 4-12, and 4-13 describe these registers.

$2C ICTL0 (INTERRUPT CONTROL REGISTER 0, R/W)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| GIE | IRQ3G2IE | IRQ4G2IE | IRQ5G2IE | IRQ6G2IE | IRQ6G4IE | IRQ7G2IE | IRQ7G4IE |

**Figure 4-11.** Interrupt Control Register 0

The Interrupt Control bits are described below (Enable=1):

### IRQ7G4IE PARERR Interrupt Enable

If this enable bit is set, a level 7 interrupt is generated when a parity error occurs. If this bit is cleared, an interrupt is not generated when a parity error occurs, but the parity error status is still recorded in the LRC Status Register (bit 2).

### IRQ7G2IE IRQ7G2 Interrupt Enable (currently unused)

### IRQ6G4IE VSBIRQ Interrupt Enable

The VSB interrupt is enabled at level 6, group 4. The LRC provides the vector during IACK cycles.

**NOTE**

> Do not enable both bits IRQ6G4IE and IRQ4G2IE (VSBIRQ) at the same time (refer to the *Interrupt Handler* section and Table 4-2).

### IRQ6G2IE TICKTMR Interrupt Enable

The MC68681 tick timer is enabled at level 6, group 2.

**NOTE**

> Do not enable both bits IRQ6G2IE and IRQ3G2IE (TICKTMR) at the same time (refer to the *Interrupt Handler* section and Table 4-2).

### IRQ5G2IE DRTIRQ Interrupt Enable

The MC68881 DUART timer is enabled at level 5, group 2.

### IRQ4G2IE VSBIRQ Interrupt Enable

The VSB interrupt is enabled at level 4, group 2. The LRC does not provide a vector during the IACK cycle. The external device should provide its own vector. Refer to the note on the IRQ6G4IE bit above.

### IRQ3G2IE TICKTMR Interrupt Enable

The MC68681 tick timer is enabled at level 3, group 2. Refer to the note on the IRQ6G2IE bit above.

### GIE Global Interrupt Enable

If this bit is cleared (GIE = 0), all interrupts are disabled.

$30 ICTL1 (INTERRUPT CONTROL REGISTER 1, R/W)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| VEC7 | VEC6 | VEC5 | VEC4 | RSVD | RSVD | IRQ7G6IE | IRQ7G5IE |

**Figure 4-12.** Interrupt Control Register 1

The Interrupt Control bits are described below (Enable=1):

### IRQ7G5IE ABORT Interrupt Enable

Writing a "1" to this bit unmasks the ABORT switch interrupt.

### IRQ7G6IE ACFAIL Interrupt Enable

Writing a "1" to this bit unmasks the ACFAIL interrupt.

### RSVD Reserved

### VEC4-VEC7 Interrupt Vector Bits 4-7

The LRC provides a unique interrupt vector for each interrupt source except IRQ5G2, IRQ4G2, and all group 1 interrupts. The vector's most significant bits (VEC4 through VEC7) are software programmable by writing to D4 through D7 in the ICTL1 register. The least significant bits are assigned by the LRC as described in Table 4-2.

$34 STAT (STATUS REGISTER, R/W)

| D23 | D22 | D21 | D20 | D19 | D18 | D17 | D16 |
|---|---|---|---|---|---|---|---|
| --- | --- | --- | --- | --- | --- | WTS | IRQ7G6S |

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 |
|---|---|---|---|---|---|---|---|
| IRQ7G5S | GPSTATL | GPSTATS | PORS | IRQ3G2 | IRQ4G2 | IRQ5G2 | IRQ6G2 |

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| IRQ7G2 | TMR2 | TMR1 | TMR0 | IRQ6G4 | IRQ7G4 | IRQ7G5 | IRQ7G6 |

**Figure 4-13.** Status Register

This status register provides status information on all the interrupt sources, the SYSFAIL status, the power on reset status, and the watchdog timer status. For the edge sensitive interrupt sources, the interrupt status are latched on the asserting edge of the interrupt request inputs. The interrupt request status is recorded whether the interrupt is masked or not. Writing a "1" to the latched status bit, clears that status bit. It does not change the interrupt request if it is still pending. The unlatched status bits (bits 9, 10, 13, 15, and 16) reflect status information at the time they are read. They can not be cleared by writing to them. The Status Register bits are described below:

### IRQ7G6 ACFAIL Interrupt Status

A "1" in this bit indicates an assertion of the ACFAIL signal is detected. This latched status is cleared when writing a "1" to this bit. Refer to the *ACFAIL Interrupt* section for the ACFAIL* interrupt description.

### IRQ7G5 ABORT Interrupt Status

This active high status bit indicates a push-button ABORT has occurred. This bit is latched at the asserting edge of the ABORT*. A non-latched version of the ABORT status is reflected in bit 15 of this status register. Refer to the *ABORT Interrupt* section for the ABORT interrupt description.

### IRQ7G4 PARERR Interrupt Status

This status bit is set to a "1" when the parity enable bit is set and a parity error is detected. If the parity interrupt is enabled, a level 7 interrupt is also requested to the processor. This status bit is latched on the asserting edge of the parity error interrupt input to the LRC. It is cleared by writing a "1" to this status bit. Refer to the *Parity Error Interrupt (PARERR)* section for more on the parity error interrupt.

### IRQ6G4 VSBIRQ Interrupt Status

This status bit is set when a VSB interrupt request, generated by the IRQ* signal on the VSB interface, is asserted. This latched status bit is cleared by writing a "1" to it. This status bit is set independent of the interrupt mask. The VSBIRQ* signal is also connected to the IRQ4G2 input on the LRC. The IRQ4G2 status in the LRC status register has the non-latched real time status of the IRQ*. Refer to the *VSB Interrupt (VSBIRQ)* section for more on the VSB interrupt.

### TMR0 Timer 0 Timeout Status

This active high status bit is set when a timeout occurs in LRC timer 0. This status bit is latched until a "1" is written to it. If the TMR0 interrupt is enabled, a level 7 interrupt is also requested to the processor. The setting or clearing of the status bit is independent of the enabling or clearing of the interrupt request.

### TMR1 Timer 1 Timeout Status

This active high status bit is set when a timeout occurs in LRC timer 1. This status bit is latched until a "1" is written to it. If the TMR1 interrupt is enabled, a level 6 interrupt is also requested to the processor. This status bit is set whether the interrupt is enabled or not. Clearing the status does not take away the pending interrupt either.

### TMR2 Timer 2 Timeout Status

This active high status bit is set when timeout occurs in LRC timer 2.

### IRQ7G2 IRQ7G2 Interrupt Status (currently unused)

### IRQ6G2 TICKTMR Interrupt Status

This active high status bit indicates a high to low transition on the IRQ6G2* input line was detected. This bit is latched until a "1" is written to clear it. The IRQ6G2* input is driven by the MC68881's OP3 output. The OP3 output also drives the IRQ3G2* input to the LRC. Refer to the *DUART Timer Interrupt (TICKTMR)* section for more information on the DUART timer interrupt.

### IRQ5G2 (unlatched) DRTIRQ Interrupt Status

This non-latched status bit reflects the MC68881 DUART interrupt line (IRQ) status. It is a "1" when DUART IRQ is asserted and a "0" when DUART IRQ is negated. This status bit can not be cleared by software since it reflects the real-time IRQ status.

### IRQ4G2 VSBIRQ Interrupt Status

This non-latched status bit reflects the status of the VSB interrupt request (IRQ*) signal on the VSB interface. It is a "1" when VSB IRQ* is asserted and a "0" when VSB IRQ* is negated. This status bit can not be cleared by software since it is not latched. Refer to the *VSB Interrupt (VSBIRQ)* section for more information on the VSB interrupt.

### IRQ3G2 TICKTMR Interrupt Status

This active high status bit is set when a high to low transition on the IRQ3G2* line was detected. The IRQ3G2* line is driven by the MC68881 OP3 output. The reason for tying OP3 to both interrupt request lines is to allow the timer to generate either a level 6 interrupt or a level 3 interrupt. It is recommended that only one of the two interrupts be enabled at one time. Refer to the *DUART Timer Interrupt (TICKTMR)* section for more information on the DUART timer interrupt.

### PORS Power On Reset Status

This bit is set when a power up condition is detected. Writing a "1" to this bit clears it and rearms it for future power-up detection.

### GPSTATS General Status of SYSFAIL

This is an unlatched SYSFAIL status bit. It is a "1" when the VMEbus SYSFAIL* is asserted and it is a "0" when the VMEbus SYSFAIL* signal is negated.

### GPSTATL Latched Version of SYSFAIL

This is a latched version of the SYSFAIL status. This bit is set on the high to low transition of the VMEbus SYSFAIL* signal. This bit stays latched until a "1" is written to clear it.

### IRQ7G5S (unlatched) ABORT Status

This unlatched status bit is a "1" when the push-button ABORT* signal is asserted. It is a "0" when the ABORT* signal is negated. This status information is independent of the enabling of the interrupt. It can not be cleared since it is not latched. Refer to Bit 1 of the status register description.

### IRQ7G6S (unlatched) ACFAIL Status

This unlatched status bit is a "1" when the ACFAIL* signal is asserting. A latched version of the ACFAIL* status is set in Bit 0 on the assertion of the ACFAIL* signal.

### WTS Watchdog Timeout Status

This status bit indicates a LRC watchdog timeout occurred. This bit is cleared during power-on-reset. It is set when the LRC watchdog times out. The LRC also asserts the RESET signal for 4 microsecond when the watchdog timer times out. When coming out of reset, this bit can be checked to determine if the previous reset is caused by the watchdog timeout. Writing a "1" to this bit clears the status.

**NOTE**

Bit 17, the Watchdog Timeout Status, is not cleared by Reset, but is cleared by Power-On-Reset.

**ALL WRITES TO THE STATUS REGISTER MUST BE PERFORMED AS A LONGWORD ACCESS.** Writing a one (1) clears the specific status bit of this register with the exception of bits 9, 10, 13, 15, and 16 which are instantaneous values and therefore cannot be cleared.

### RAM Address Register

To support MC68040 snoop operation, the LRC can drive the address lines on the processor address bus with the values programmed in the RAM Address Register during slave accesses to the on-board RAM. When the Slave Address Enable (SLVAEN*) input is asserted then the LRC drives the non-masked address lines. This input pin is controlled by the LRCADDEN bit in the MVME165 Local CSR (refer to the *MVME165 Control and Status Registers* section). If Address lines A20 through A26 are programmed to be masked, then the LRC will not drive those lines. Figure 4-14 describes this register. This feature should be used when snoop operation is desired and VMEbus or VSB slave address of on-board RAM does not match the locally decoded address.

## $38 RADR (RAM ADDRESS REGISTER, R/W)

| D19 | D18 D12 | D11 D0 |
|---|---|---|
| RSVD | MSKA26-MSKA20 | RA31-RA20 |

**Figure 4-14.** RAM Address Register

The RAM Address bits are described below:

**RA20-31** **RAM Base Address Bits 20-31**

Example: D0-D11 = $000 indicates a base address at 0.
D0-D11 = $004 indicates a base address at 4M.
(i.e., local memory starts at $00400000)

D0-D11 = $010 indicates a base address at 16M.
(i.e., local memory starts at $01000000)

**MSKA20-26** **Mask Address Bits 20-26 (decode size 1M-64Mb)**

Example (MVME165): 4Mb Version: MSKA20-26 = 0000011
16Mb Version: MSKA20-26 = 0001111

**RSVD** **Reserved**

### Composite Bus Error Status

The LRC provides a local bus error status bit which is composed of both offboard and local bus error conditions. Figure 4-15 describes this bit and its function.

$3C BERR (BUS ERROR REGISTER, R/C)

| | D0 |
|---|---|
| | BERR |

**Figure 4-15.** Bus Error Register

The Bus Error bit is described below:

**BERR** **Bus Error Status**

Any bus error (TEA* asserted but not TA*) sets this bit. Note that for line transfers (four longword burst), bus error at the first longword will always be reported, but bus error at the second, third, and fourth longword may not be reported since both TEA* and TA* could be asserted.

## LRC Reset Operation

INPUT - The LRC has a reset signal which clears most of its flip-flops and/or functions to a known state. There is also a power on reset which is used in the serial clock section to clear it, as well as initialize the Refresh Clock.

OUTPUT - The LRC asserts the Reset output pin for 4 microseconds if it's watchdog function is enabled and the watchdog timer times out. Refer to the *VMEbus* section for more information on timers. The LRC can also assert the Reset signal by writing a "1" to the RSTO bit in the General Control Register ($24). The LRC RESET* pin is connected to the PRESET* signal on the MVME165. When asserted, it sets/resets the control logic and the status register affected by PRESET*. It does not generate SYSRESET even if it is the system controller. It is different than the reset bit in the VMEchip LCSR (refer to the *Reset Operation* section).

# MVME165 Control And Status Registers

The MVME165 has two longword locations which are used to configure various mapping capabilities of the board. Address location FFFE0000 is implemented in a read/write register (D18 and D20 are read only), which is not cleared by Reset, and should be initialized by software at power-up. All definitions in address $FFFE0000 are hard-wired and/or defined. Address location FFFE0004 is implemented in two jumper blocks and therefore, is read only.

All references to these jumpers are as implemented by the 165Bug, but are configured, with the exception of jumper J4, pin positions 1-2 (SCON), which is hard-wired to the VMEchip for the System Controller function. These two longword addresses and their functions are described in Figures 4-16 and 4-17.

**$FFFE0000**

| Address | D31 | D30 | D29 | D28 | D27 | D26 | D25 | D24 |
|---|---|---|---|---|---|---|---|---|
| 0 | X | X | X | X | X | X | X | X |
| | D23 | D22 | D21 | D20 | D19 | D18 | D17 | D16 |
| 1 | X | X | X | RESRVD | (A) | 4MEG* | (B) | (C) |
| | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 |
| 2 | RESRVD | LOCALADDR | | | VSBADDR | | | |
| | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 3 | VMEADDR | | | | GCSRADDR | | | |

(A) = PARBERRDIS (B) = LRCADDEN (C) = SNOOPENBLE

**Figure 4-16.** MVME165 Control and Status Register ($FFFE0000)

The MVME165 Control and Status Register bits at $FFFE000 are described below:

### GCSRADDR (D3-D0) - VMEchip GCSR Base Address (A8-A11)

These bits, together with the GCSR base address configuration register in the VMEchip LCSR, define the VMEchip GCSR base address. Default can be configured by reading corresponding bits in jumper positions J5(1-2), (3-4), (5-6), and (7-8) at $FFFE0004.

### VMEADDR (D7-D4) - VME Base Address (for local DRAM)

These bits define where the MVME165 RAM is located as a VME slave. They define the VME base address. Default can be configured by reading corresponding bits in jumper positions J4(15-16), (17-18), (19-20), and (21-22) at $FFFE0004.

### VSBADDR (D11-D8) - VSB Base Address (for local DRAM)

These bits define where the MVME165 RAM is located as a VSB slave. Default can be configured by reading corresponding bits in jumper positions J4(7-8), (9-10), (11-12), and (13-14) at $FFFE0004.

### LOCALADDR (D14-D12) - Local Base Address

These bits define where the MVME165 RAM is mapped locally, as viewed from the MC68040 microprocessor. Default address is at zero. If any other base address is required, software must provide it or retrieve it from the Non-Volatile static RAM. Currently only six locations are possible due to hardware limitations. Base addresses supported for the 4Mb board are as shown below (refer to Appendix B for the 16Mb board):

| Data Bits | | | Local DRAM Base Address |
|---|---|---|---|
| D14 | D13 | D12 | (local memory starts at) |
| 0 | 0 | 0 | $00000000 |
| 0 | 0 | 1 | $00400000 |
| 0 | 1 | 0 | $00800000 |
| 0 | 1 | 1 | $00C00000 |
| 1 | 0 | 0 | $01000000 |
| 1 | 0 | 1 | $01400000 |
| 1 | 1 | 0 | Local DRAM Disabled |
| 1 | 1 | 1 | Local DRAM Disabled |

### RESRVD (D15) - Reserved

### SNOOPENBLE (D16) - SNOOP Enable

This bit is directly connected to the MC68040's SC0 pin. The SC1 pin is driven low on the MVME165. When this bit is set, the MC68040 is instructed to snoop alternate master transfers, i.e.; VMEbus or VSB slave accesses. On a read access, the MC68040 will inhibit local memory, supply dirty data, and leave dirty data. On a write, local memory is inhibited and the MC68040 will sink the byte, word, or longword. Both cases are assuming a hit in the data cache. When the bit is cleared, snooping is inhibited.

### LRCADDEN (D17) - LRC Slave Address Enable

This bit is used as an input to the LRC gate-array. When equal to 1, then it indicates to the LRC that a slave access to the local DRAM from the VMEbus or the VSB requires that the LRC drive A20 through A31 with the contents of the RAM Address Register of the LRC's CSR (refer to the *RAM Address Register* section). This is to support MC68040 snoop operation when the VMEbus and VSB slave address of on-board RAM does not match the local address.

### 4MEG* (D18) - 4 Megabyte*

4 MEG (active low, i.e.; equal to 0) indicates, when read, that this board has 4Mb of DRAM onboard. If equal to 1, then this board has 16Mb of DRAM onboard.

**PARBERRDIS (D19) - Parity Bus Error Disable**

When this bit is equal to 1, it indicates that a parity error in the DRAM array will not generate a bus error. When PARBERRDIS is equal to 0, a bus error will occur when a parity error is detected.

**RESRVD (D20) - Reserved**

**X (D31-D21) - Not Implemented**

**$FFFE0004**

| Address | D31 | D30 | D29 | D28 | D27 | D26 | D25 | D24 |
|---|---|---|---|---|---|---|---|---|
| 4 | X | X | X | X | X | X | X | X |
| | D23 | D22 | D21 | D20 | D19 | D18 | D17 | D16 |
| 5 | X | X | X | X | X | X | (D) | (E) |
| | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 |
| 6 | (F) | (READ AS ZERO) | | | J4(7-8) | (9-10) | (11-12) | (13-14) |
| | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 7 | J4(15-16) | (17-18) | (19-20) | (21-22) | J5(1-2) | (3-4) | (5-6) | (7-8) |

(D) = SCON; J4(1-2)  (E) = TRACK; J4(3-4)  (F) = NVRAM*; J4(5-6)

**Figure 4-17.** MVME165 Control and Status Register ($FFFE0004)

The MVME165 Control and Status Register bits at $FFFE004 are described below:

**NVRAM* (D15) - Non-Volatile RAM**

This bit is used to indicate to the MVME165BUG software to obtain the alternate; i.e., not using the jumpers for base addressing, but use what was stored in NVRAM by the 165Bug's "ENV" utility. If NVRAM* is equal to 1, then use the jumpers for base addressing, and if it is equal to 0, then get the base address from the NVRAM. This bit is at jumper J4(5-6), read at address $FFFE0004 (D15). (Refer to Appendices A and B.)

**TRACK (D16) - TRACK**

When NVRAM* is equal to 1 (i.e.; jumper cap at J4 pins 3 and 4, removed) and TRACK is equal to 1 (i.e.; jumper cap at J4 pins 5 and 6, removed), then the debugger programs the local memory so that it is mapped to track the VMEbus memory as defined in Appendices A and B.

**SCON*** **VMEbus System Controller**

Jumper J4(1-2) is hard-wired to the VMEchip for the System Controller function which means that this MVME165 is the VMEbus System Controller.

Note that all jumpers are shown as pairs; i.e., J4(3-4), meaning that if a jumper cap is installed at those two pins then that bit will be read as a logic 0. If the jumper cap is absent (not installed), then that bit will be read as a logic 1.

## DUART

The DUART (MC68681) is decoded at addresses FFFD0003-FFFD003F. It is 16b in size, and it is configured as an 8-bit wide port for the processor. The registers of the DUART are accessed at the least significant byte of a longword address. The register assignments are summarized in Table 4-4.

**Table 4-4.** MVME165 DUART Address Map

| Address | Read Cycle | Write Cycle |
|---|---|---|
| FFFD0003 | Serial Port A - Mode R. | Serial Port A - Mode R. |
| FFFD0007 | Serial Port A - Status R. | Serial Port A - Clock Select R. |
| FFFD000B | (Do Not Access) | Serial Port A - Command R. |
| FFFD000F | Serial Port A - Receiver R. | Serial Port A -Transmitter R. |
| FFFD0013 | Input Port - Change R. | Auxiliary Control R. |
| FFFD0017 | Interrupt - Status R. | Interrupt - Mask R. |
| FFFD001B | Counter - Current MSB R. | Counter/Timer - MSB Load R. |
| FFFD001F | Counter - Current LSB R. | Counter/Timer - LSB Load R. |
| FFFD0023 | Serial Port B - Mode R. | Serial Port B - Mode R. |
| FFFD0027 | Serial Port B - Status R. | Serial Port B - Clock Select R. |
| FFFD002B | (Do Not Access) | Serial Port B - Command R. |
| FFFD002F | Serial Port B - Receiver R. | Serial Port B -Transmitter R. |
| FFFD0033 | Interrupt - Vector R. | Interrupt - Vector R. |
| FFFD0037 | Input Port - Status Buffer | Output Port - Configuration R. |
| FFFD003B | Counter - Start Command | Output Port - Bit Set Command |
| FFFD003F | Counter - Stop Command | Output Port - Bit Clear Command |

**NOTE:** R denotes Register.

The DUART contains two serial port channels. Channel A operates through serial port connector #1 on the front panel (SP1, lower connector). Channel B operates through serial port connector #2 (SP2, upper connector). The input port and output port on the

DUART are reserved for the serial port handshake signals and the timer interrupt clock. They are described in the following section. The crystal frequency used to operate the DUART is the recommended 3.6864 MHz. The DUART is reset by any power-up or reset condition. Refer to the *MC68681 Dual Asynchronous Receiver/Transmitter (DUART) Data Book* for a detailed description of the register functions and for general operating information.

## DUART Port Assignments

Port assignments for the input and output port on the DUART are listed in Table 4-5.

**Table 4-5.** MVME165 DUART Port Assignments

| Register Bit | I/O | Port Assignment |
|---|---|---|
| OP0 | --> | RTS for channel A |
| OP1 | --> | RTS for channel B |
| OP2 | --> | DTR for channel A |
| OP3 | --> | Timer Interrupt Clock |
| OP4 | --> | Not Used |
| OP5 | --> | DTR for channel B |
| OP6 | --> | Not Used |
| OP7 | --> | Not Used |
| IP0 | <-- | CTS for channel A |
| IP1 | <-- | CTS for channel B |
| IP2 | <-- | DCD for channel A |
| IP3 | <-- | DCD for channel B |
| IP4 | <-- | DCD for channel A |
| IP5 | <-- | DCD for channel B |

**NOTES:**

1. Channel A uses serial port connector 1 (SP1).
2. Channel B uses serial port connector 2 (SP2).
3. IP4 and IP5 do not provide change-of-state detection.
4. Refer to the *Interrupt Processing* section and Tables 2-2 and 2-3.

The port pins are reserved for the serial port handshake signals and the timer interrupt clock. The handshake signals are defined by the RS-232C guidelines for interfacing data terminal equipment. The signal pair RTS/CTS is typically used for hardware flow control between two serial ports. This signal function is automatically managed by the DUART circuitry after appropriately programming the chip. The signal pair DTR/DCD is typically used to indicate the state of presence or readiness between two serial ports. This signal function must be managed by the firmware or software running on the board.

Note that the use of these handshake signals may vary with different computer products. The handshake signal DSR is not used by the MVME165. It is pulled up to +12V for both serial ports. The DUART supports change of state detection for the input port pins assigned to CTS and DCD for both serial ports.

The output port pin OP3 on the DUART is used for the tick-timer clock. A high-to-low signal transition on this pin will generate a timer interrupt to the processor. The timer interrupt request to the MC68040 and the interrupt handling are generated by the LRC on the board. A tick timer can be implemented by programming a continuous, square wave output on OP3. A counter load value of 4800 HEX or 7800 HEX will generate a 100 Hz or 60 Hz timer interrupt when the counter/timer is programmed for the timer mode using the 1x clock source. Refer to the *MC68681 Dual Asynchronous Receiver/Transmitter (DUART) Data Book.*

## VMEbus

The VMEbus (except for VMEbus Short I/O) is selected in the processor address range 00000000-FF7FFFFF. This range corresponds to any access outside the top 8Mb of the address map. The MVME165 operates in master/slave mode on the VMEbus.

The VMEbus specification allows master and slave boards to use either the full 32 bits of the address bus (A32 mode), or only the lower 24 bits of the address bus (A24 mode) to select or decode the cycle. It also allows boards to use the full 32 bits of the data bus, or only the lower 16 bits of the data bus to transfer data. The address and data mode must be specified by the master during the bus access, and it must be compatible with the slave for the cycle to be completed successfully.

**NOTE**

> All references to the VMEbus master reflect either A32/D32 or A24/D32 transfers. The MC68040 microprocessor does not support Dynamic Bus sizing. The VMEchip cannot break the 32-bit transfer into two 16-bit transfers. If the user wishes to perform a D16 or D8 transfer on the VMEbus, the user's software must perform the transfer as a word (D16) or as a byte (D8).

### VMEbus Control/Status Registers

The address and data modes used by the MVME165 when accessing VMEbus are shown in Table 3-2 and Figures 4-1 and 4-2. The A32 or A24 option is selected by the VA24 bit in the BCR of the LRC ($28, D4). Control of the VMEbus is orchestrated by the MC68040 through the VMEchip. The VMEchip has two groups of registers which control it's operation; the Local Control and Status Register (LCSR), and the Global

Control and Status Registers (GCSR). These register sets are shown in Tables 4-6 and 4-7. Refer to Chapters 4 and 5 of the *MVME6000 (VMEchip) VMEbus Interface User's Manual* for a detailed description of each of these registers.

**Table 4-6.** VMEchip Local Control and Status Registers

| Address | Register |
|---|---|
| FFFB0001 | System Controller Configuration Register |
| FFFB0003 | Requester Configuration Register |
| FFFB0005 | Master Configuration Register |
| FFFB0007 | Slave Configuration Register |
| FFFB0009 | Timer Configuration Register |
| FFFB000B | Slave Address Modifier Register |
| FFFB000D | Master Address Modifier Register |
| FFFB000F | Interrupt Handler Mask Register |
| FFFB0011 | Utility Interrupt Mask Register |
| FFFB0013 | Utility Interrupt Vector Register |
| FFFB0015 | Interrupt Request Register |
| FFFB0017 | Status/ID Register |
| FFFB0019 | Bus Error Status Register |
| FFFB001B | GCSR Base Address Register |

**Table 4-7.** VMEchip Global Control and Status Registers

| Address | Global Offset | Register |
|---|---|---|
| FFFB0021 | 01 | Global Register 0 |
| FFFB0023 | 03 | Global Register 1 |
| FFFB0025 | 05 | Board ID Register |
| FFFB0027 | 07 | General Purpose Register 0 |
| FFFB0029 | 09 | General Purpose Register 1 |
| FFFB002B | 0B | General Purpose Register 2 |
| FFFB002D | 0D | General Purpose Register 3 |
| FFFB002F | 0F | General Purpose Register 4 |

**NOTE:** Global offset = Offset to GCSR registers from VMEbus GCSR base address in the VMEshort I/O space (FFFF xxyz), where xx denotes the 165CSR GCSR decode, y denotes the GCSR base address register, and z denotes the global offset.

The GCSR is accessed over VMEbus within the Short I/O space. The address is partially programmable from the local CPU. Address bits A04-A07 are specified in bits 0-3 of the GCSR Base Address Configuration Register and address bits A08-A15 are decoded external to the VMEchip on the MVME165 by the 165CSR value in bits 3-0. Certain address ranges for VMEbus are hard-decoded by the MVME165 for A32 mode or A24 mode only. This enables different address size boards to be used in a system without having to configure the entire VMEbus to suit the smallest mode.

## VMEbus Short I/O

The VMEbus Short I/O (SHIO) space is selected in the processor address range FFFF0000-FFFFFFFF. It is an alternate 64Kb space on VMEbus that uses only the lower 16 bits of the address bus to simplify decoding. This space is typically used for CSRs and other I/O devices on the bus that can occupy small address ranges. During Short I/O accesses, the MVME165 generates one of the two VMEbus short I/O address modifiers 2D or 29. The particular address modifier used is determined by the transfer modifiers of the processor. The Short I/O space is configured as a 32-bit wide (D32).

## VME Subsystem Bus (VSB)

The VME Subsystem Bus is implemented on the MVME165 to support multiprocessor systems by providing a secondary data transfer path. It includes a multiplexed asynchronous data transfer bus which allows the MVME165 to access data stored in slaves. Four types of bus cycles are possible; an address-only cycle, a single-transfer cycle, a block-transfer cycle, and an interrupt-acknowledge cycle. The interface to the VSB bus is implemented with the MVSB2400 (VSBchip). The MVSB2400 provides bus arbitration, a direct 32-bit address/data interface to the bus, a 32-bit block transfer counter, and a VSB transfer watchdog timer. The internal registers associated with the VSBchip are shown in Table 4-8.

**Table 4-8.** VSBchip Address Map

| Address | Register | Size |
|---|---|---|
| FFFA0000 | Control and Status Register | 16 bits |
| FFFA0004 | Block-Transfer Byte-Count Register | 32 bits |
| See Note | Address-Decode Register | 32 bits |

**NOTE:** This register is not accessible on the MVME165.

The control and status register is used to control and configure the operation of the VSBchip as well as allow the MC68040 to monitor status information regarding the VSB. The block-transfer byte-count register allows the CPU to control the number of bytes transferred during a VSB block-transfer cycle. Master block-transfers should not be performed by the MVME165. Slave block-transfers are supported. Circuitry to access the address-decode register is not implemented on the MVME165, therefore; the base address for the VSBchip registers can not be changed from what is shown in Table 4-1. Refer to Chapter 3 of the *MVSB2400 VSBchip User's Manual* for bit definitions and functions of the registers. The master and slave functions of the VSBchip are implemented on the MVME165. The MVME165 is not capable of sourcing an interrupt on the VSB; but, circuitry is provided to handle an interrupt from another source on the bus.

The *VSB Interrupt (VSBIRQ)* section describes in more detail how the MVME165 VSB interrupt handler functions.

## Local Timeout

The Local Timeout is supported by the VMEchip and must be enabled in the Timer-On-Configuration Register of the LCSR set. A Local Timeout (LTO) error is indicated for the onboard address ranges where no resources are selected, and for write operations to the ROM space. The local address of FFF00000 through FFF7FFFF is specifically reserved and any access will cause a local timeout. A local timeout will terminate the processor access with a bus error exception.

## Interrupt Processing

The MVME165 supports 19 interrupt requests to the processor. The interrupt requests are prioritized by the interrupt handler circuitry on the LRC into seven interrupt request levels with six prioritized groups per level. The prioritization is such that any higher level interrupt will take precedence over any lower level interrupt, and any higher group interrupt on a given level will take precedence over any lower group interrupt on the same level. The interrupt sources and their prioritizations are given in Table 4-9. Level 7 is the highest priority and level 1 is the lowest. The LRC provides a unique interrupt vector for all interrupts except Group 1 interrupt and (IRQ5G2 (DRTIRQ) and IRQ4G2 (VSBIRQ)), refer to the *Interrupt Handler* and *Interrupt Control and Status Register* sections for further information.

The interrupt vectors for the group 1 VMEbus interrupts are passed on to the MC68040 by the VMEchip. The VMEchip Utility interrupts are generated for locally detected conditions and the VMEchip provides the vector based on contents of the Utility Interrupt Vector Register and the source of interrupt. The Interrupt Handler Mask Register and the Interrupt Mask Register provide individual masking capability.

In addition, all interrupt requests can be masked with the global interrupt mask bit of the LRC. The VMEchip has multiple groups within Group 1. Refer to the *MVME6000 (VMEchip) VMEbus Interface User's Manual* for further details.

If the MVME165 DEBUGGER is installed, then the LRC and the VMEchip are already programmed for the proper vector, as shown by the byte in parenthesis (xn). Where x is the user programmable portion and n is the hardware defined portion of the vector.

**Table 4-9.** MVME165 Interrupt Assignments

| Highest | | | | | | Lowest |
|---|---|---|---|---|---|---|
| **Level** | **Group 6** | **Group 5** | **Group 4** | **Group 3** | **Group 2** | **Group 1** |
| 7 | ACFAIL | ABORT | PARERR | TMR0IRQ | IRQ7G2(@) | VIRQ7 |
| 6 | ----- | ----- | VSBIRQ | TMR1IRQ | TICKTMR | VIRQ6 |
| 5 | ----- | ----- | ----- | TMR2IRQ | DRTIRQ | VIRQ5 |
| 4 | ----- | ----- | ----- | ----- | VSBIRQ | VIRQ4 |
| 3 | ----- | ----- | ----- | ----- | TICKTMR | VIRQ3 |
| 2 | ----- | ----- | ----- | ----- | ----- | VIRQ2 |
| 1 | ----- | ----- | ----- | ----- | ----- | VIRQ1 |

**NOTES:**

1. All interrupt requests are edge sensitive except IRQ5G2 (DRTIRQ), IRQ4G2 and all Group 1 (VMEchip) interrupts.
2. @ indicates interrupt not used.

## ACFAIL Interrupt

The AC Fail (ACFAIL) interrupt (level 7, group 6) is generated by the ACFAIL* signal on the VMEbus if the ACFAIL interrupt is unmasked in LRC ICTL1. This signal asserts when AC power is lost, and it indicates that a minimum of 2 milliseconds of operating time is left before DC power will be lost to the board. The ACFAIL interrupt is cleared by the IACK cycle. A vector of $xB ($4B) is provided to the MPU by the LRC.

## ABORT Interrupt

The ABORT (level 7, group 5) interrupt is generated by the Abort push-button switch located on the front panel of the MVME165. If unmasked, a level 7 interrupt is asserted to the processor on the asserting edge of the ABORT. The ABORT interrupt is cleared by the IACK cycle. The LRC provides a vector of $xA ($4A) to the local MPU.

## Parity Error Interrupt (PARERR)

The Memory parity error interrupt (level 7, group 4) is connected to the local DRAM's parity error detected signal. The LRC provides a vector of $x8 ($48) to the local MPU.

The parity error status is recorded in the LRC status register. This latched status stays there until a "1" is written to that status bit. The status information is reported whether the interrupt is enabled or not.

## DUART Timer Interrupt (TICKTMR)

The TICK TiMeR interrupt (level 6 and level 3, group 2) is triggered by a high-to-low signal transition on output port signal OP3 of the DUART. The interrupt is latched by the Local Resource Controller. The level 6 vector of $x5 ($45) and the level 3 vector of $x7 ($47) is provided by the LRC. This allows the user to use the TICKTMR to generate a level 6 interrupt or a level 3 interrupt depending on the user's requirement. The timer interrupt may already be asserted after a power-up or reset condition. The on board firmware should keep it masked until ready for use. A tick timer can be implemented by programming a continuous, square wave output on OP3. A counter load value of 4800 HEX or 7800 HEX will generate a 100 Hz or 60 Hz timer interrupt when the counter/timer is programmed for the timer mode using the 1x clock source. The timer interrupt can be masked within the Interrupt Control Register 0 of the LRC. When this off-chip timer interrupt is used, the timer interrupt source on the DUART chip should be masked.

## VSB Interrupt (VSBIRQ)

The VSB Interrupt ReQuest is generated by the IRQ* signal on the VSB interface. It is connected to both IRQ6G4* (level 6, Group 4) and IRQ4G2* (level 4, Group 2) on the MVME165. The IRQ6G4* interrupt is a edge sensitive interrupt, and the LRC supplies the vector during the IACK cycle. The IRQ4G2* interrupt is a level sensitive interrupt and the vector is fetched across the bus from the interrupting device. Only one of these two interrupts should be enabled. The purpose of connecting both IRQ6G4* and IRQ4G2* to the VSBIRQ* line is to allow the user to have an option of supply vector by the LRC or by the external VSB devices.

## LRC Timer Interrupt (TMR0IRQ,TMR1IRQ,TMR2IRQ)

The LRC timer interrupts (levels 7,6, 5; respectively, group 3) are generated by the counter/timers within the Local Resource Controller chip (LRC). Vectors of $x0 ($40), $x1 ($41) and $x2 ($42); respectively, are supplied by the LRC for each of the timers. The 24-bit counters are clocked by the LRC's 16MHz clock divided by 16. The counter signals the occurrence of an event primarily through a time out detection. This sets the time out status bit in the LRC timer status register. It may be checked by the MPU or may be used to generate a TMRxIRQ interrupt. The MVME165BUG sets this up as an interrupt with a vector supplied by the LRC. The time out status bit can be cleared by writing a one to the timer status register in that bit position independent of timer operation. Refer to the *Timers* section for more information regarding the LRC counter/timers.

### DUART Interrupt (DRTIRQ)

The DUART interrupt (level 5, group 2) is generated by the various programmable interrupt sources of the DUART chip. The interrupt sources are associated with the two serial ports, the input port, and the counter/timer. Refer to the *MC68681 Dual Asynchronous Receiver/Transmitter (DUART) Data Book* and to the *DUART Port Assignments* section on DUART Port Assignments. The input port signals are connected to the CTS and DCD signals of the two serial ports. They can be programmed to generate a change-of-state interrupt on the serial port's flow control or connect/disconnect status. The interrupt vector of $xn ($xn) is provided by the DUART, and must be programmed by firmware/software.

**NOTE**

> The on-chip counter/timer interrupt should remain masked when the off-chip timer interrupt is used. If the on-chip counter/timer interrupt is used instead, the output port pin OP3 should be programmed as a static output to avoid triggering the off-chip timer interrupt (refer to the *DUART Timer Interrupt (TICKTMR)* section).

### VMEbus/Utility Interrupts

The seven VMEbus interrupts and seven Utility interrupts (levels 1 through 7, group 1) are passed to the processor at the same interrupt request levels as they are on the bus, via the VMEchip. Each interrupt has a corresponding mask bit in the Interrupt Handler Mask Register and the Utility Interrupt Mask Register of the VMEchip (refer to the *MVME6000 (VMEchip) VMEbus Interface User's Manual* for more details). During the interrupt acknowledge cycle for the 7 VMEbus interrupts, the interrupt vector is fetched across the bus from the interrupting device. The utility interrupts are internally generated, with the VMEchip providing the correct interrupt vector during Utility Interrupt acknowledge cycles; $x1 through $x7 ($61 through $67) for levels 1 through 7, respectively.

## RMC Operation

RMC operations are sequences of read and write cycles by the processor that should complete without interruption from other bus masters. The processor will signal an RMC operation during the TAS, CAS, and CAS2 instructions, and during all MMU table walk operations. To ensure uninterrupted operation of these operations over VMEbus, the bus requester on the MVME165 (which is in the VMEchip) will not release the bus to other boards in the system during the RMC sequences. For VSB cycles the VSBchip maintains bus ownership for the onboard processor.

## Reset Operation

The MVME165 has five sources of reset. They are as follows:

SYSRESET*: is the master reset signal on VMEbus and when asserted, resets all onboard devices including the processor and VMEchip. SYSRESET* will also reset all devices in the VMEbus system.

Power-On-Reset: is generated by an integrated circuit and resets all onboard devices if asserted. Also, if the MVME165 is configured as system controller, then SYSRESET* is asserted on the VMEbus.

Front Panel Reset: is switch mounted on the front panel of the MVME165, and when pushed, resets all onboard devices and, if system controller, asserts SYSRESET*. Push-button reset on the MVME165 is debounced and the VMEchip ensures that when PRESET* is driven to the CPU and other local devices, it will be for a minimum duration of 200 milliseconds.

Software Reset: is also possible, not by executing a reset instruction from the MC68040, but by programming the VMEchip to assert a Reset. The reset bit in the LCSR (bit D1 at FFFB0001), when set, will assert SYSRESET* on the VMEbus (if that board is the system controller) which will reset the MVME165 and all devices on the bus. The Reset and Hold bit in the GCSR (bit D7 at GCSR register offset $01), while set, will hold the local devices on the MVME165 in reset. The Reset bit in the LRC, when set, will assert reset for four microseconds. It will not generate SYSRESET*, even if the MVME165 is the system controller.

Watchdog Timer: can be programmed in the Local Resource Controller (LRC) by using one of the 24-bit timers. When the watchdog timer times out, it generates a reset at the Reset pin for four microseconds. Also refer to *LRC Reset Operation* section for LRC related reset functions.

Any of the resets described will initialize the various bits of the 165CSR as well as registers within the VMEchip, the VSBchip, the DUART, and the LRC.

# APPENDIX A
# 4Mb DEBUGGER OPERATION PER JUMPERS

**Table A-1.** MODE 0 AND 1: USE NVRAM FOR ALL PARAMETERS

| Jumper Block J4 (see notes) | | | | | | Debugger | Local | VMEbus | VSB and |
|---|---|---|---|---|---|---|---|---|---|
| NVRAM* | TRACK | VMEbus | | | | Memory | Memory | Memory | VMEchip-GCSR |
| (5-6) | (3-4) | (15-16) | (17-18) | (19-20) | (21-22) | Start | Mapping | Mapping | Mapping |
| IN | X | X | X | X | X | NVRAM or ROM search/default parameters. | From NVRAM or ROM image MVME165 CSR. | From NVRAM or ROM image MVME165 CSR. | From NVRAM or ROM image MVME165 CSR. |

Notes: IN = Jumper cap installed. OUT = Jumper cap removed. x = Don't care.

**Table A-2.** MODE 2: USE LOCAL DRAM AT $0 (DO NOT USE NVRAM)

| Jumper Block J4 (see notes) | | | | | | Debugger | Local | VMEbus | VSB and |
|---|---|---|---|---|---|---|---|---|---|
| NVRAM* | TRACK | VMEbus | | | | Memory | Memory | Memory | VMEchip-GCSR |
| (5-6) | (3-4) | (15-16) | (17-18) | (19-20) | (21-22) | Start | Mapping | Mapping | Mapping |
| OUT | IN | IN | IN | IN | IN | $0 | $0 | $00000000 | per jumpers |
| OUT | IN | IN | IN | IN | OUT | $0 | $0 | $00400000 | per jumpers |
| OUT | IN | IN | IN | OUT | IN | $0 | $0 | $00800000 | per jumpers |
| OUT | IN | IN | IN | OUT | OUT | $0 | $0 | $00C00000 | per jumpers |
| OUT | IN | IN | OUT | IN | IN | $0 | $0 | $01000000 | per jumpers |
| OUT | IN | IN | OUT | IN | OUT | $0 | $0 | $01400000 | per jumpers |
| OUT | IN | IN | OUT | OUT | IN | $0 | $0 | $01800000 | per jumpers |
| OUT | IN | IN | OUT | OUT | OUT | $0 | $0 | $01C00000 | per jumpers |
| OUT | IN | OUT | IN | IN | IN | $0 | $0 | $02000000 | per jumpers |
| OUT | IN | OUT | IN | IN | OUT | $0 | $0 | $02400000 | per jumpers |
| OUT | IN | OUT | IN | OUT | IN | $0 | $0 | $02800000 | per jumpers |
| OUT | IN | OUT | IN | OUT | OUT | $0 | $0 | $02C00000 | per jumpers |
| OUT | IN | OUT | OUT | IN | IN | $0 | $0 | $03000000 | per jumpers |
| OUT | IN | OUT | OUT | IN | OUT | $0 | $0 | $03400000 | per jumpers |
| OUT | IN | OUT | OUT | OUT | IN | $0 | $0 | $55400000 | per jumpers |
| OUT | IN | OUT | OUT | OUT | OUT | $0 | $0 | $AA800000 | per jumpers |

Notes: IN = Jumper cap installed. OUT = Jumper cap removed. x = Don't care.

**Table A-3.** MODE 3: USE LOCAL DRAM, TRACK VMEbus

| Jumper Block J4 (see notes) | | | | | | Debugger | Local | VMEbus | VSB and |
|---|---|---|---|---|---|---|---|---|---|
| NVRAM* | TRACK | VMEbus | | | | Memory | Memory | Memory | VMEchip-GCSR |
| (5-6) | (3-4) | (15-16) | (17-18) | (19-20) | (21-22) | Start | Mapping | Mapping | Mapping |
| OUT | OUT | IN | IN | IN | IN | $00000000 | $00000000 | $00000000 | per jumpers |
| OUT | OUT | IN | IN | IN | OUT | $00400000 | $00400000 | $00400000 | per jumpers |
| OUT | OUT | IN | IN | OUT | IN | $00800000 | $00800000 | $00800000 | per jumpers |
| OUT | OUT | IN | IN | OUT | OUT | $00C00000 | $00C00000 | $00C00000 | per jumpers |
| OUT | OUT | IN | OUT | IN | IN | $01000000 | $01000000 | $01000000 | per jumpers |
| OUT | OUT | IN | OUT | IN | OUT | $01400000 | $01400000 | $01400000 | per jumpers |
| OUT | OUT | IN | OUT | OUT | IN | $0 (off-board) | disabled | $01800000 | per jumpers |
| OUT | OUT | IN | OUT | OUT | OUT | $0 (off-board) | disabled | $01C00000 | per jumpers |
| OUT | OUT | OUT | IN | IN | IN | $00000000 | $00000000 | $02000000 | per jumpers |
| OUT | OUT | OUT | IN | IN | OUT | $00400000 | $00400000 | $02400000 | per jumpers |
| OUT | OUT | OUT | IN | OUT | IN | $00800000 | $00800000 | $02800000 | per jumpers |
| OUT | OUT | OUT | IN | OUT | OUT | $00C00000 | $00C00000 | $02C00000 | per jumpers |
| OUT | OUT | OUT | OUT | IN | IN | $01000000 | $01000000 | $03000000 | per jumpers |
| OUT | OUT | OUT | OUT | IN | OUT | $01400000 | $01400000 | $03400000 | per jumpers |
| OUT | OUT | OUT | OUT | OUT | IN | $0 (off-board) | disabled | $55400000 | per jumpers |
| OUT | OUT | OUT | OUT | OUT | OUT | $0 (off-board) | disabled | $AA800000 | per jumpers |

Notes: IN = Jumper cap installed. OUT = Jumper cap removed. x = Don't care.

# APPENDIX B

## 16Mb DEBUGGER OPERATION PER JUMPERS

**Table B-1.** MODE 0 AND 1: USE NVRAM FOR ALL PARAMETERS

| Jumper Block J4 (see notes) | | | | | | Debugger | Local | VMEbus | VSB and |
|---|---|---|---|---|---|---|---|---|---|
| NVRAM* | TRACK | VMEbus | | | | Memory | Memory | Memory | VMEchip-GCSR |
| (5-6) | (3-4) | (15-16) | (17-18) | (19-20) | (21-22) | Start | Mapping | Mapping | Mapping |
| IN | X | X | X | X | X | NVRAM or ROM search/default parameters. | From NVRAM or ROM image MVME165 CSR. | From NVRAM or ROM image MVME165 CSR. | From NVRAM or ROM image MVME165 CSR. |

Notes: IN = Jumper cap installed. OUT = Jumper cap removed. x = Don't care.

**Table B-2.** MODE 2: USE LOCAL DRAM AT $0

| Jumper Block J4 (see notes) | | | | | | Debugger | Local | VMEbus | VSB and |
|---|---|---|---|---|---|---|---|---|---|
| NVRAM* | TRACK | VMEbus | | | | Memory | Memory | Memory | VMEchip-GCSR |
| (5-6) | (3-4) | (15-16) | (17-18) | (19-20) | (21-22) | Start | Mapping | Mapping | Mapping |
| OUT | IN | IN | IN | IN | IN | $0 | $0 | $00000000 | per jumpers |
| OUT | IN | IN | IN | IN | OUT | $0 | $0 | $01000000 | per jumpers |
| OUT | IN | IN | IN | OUT | IN | $0 | $0 | $02000000 | per jumpers |
| OUT | IN | IN | IN | OUT | OUT | $0 | $0 | $03C00000 | per jumpers |
| OUT | IN | IN | OUT | IN | IN | $0 | $0 | $04000000 | per jumpers |
| OUT | IN | IN | OUT | IN | OUT | $0 | $0 | $05000000 | per jumpers |
| OUT | IN | IN | OUT | OUT | IN | $0 | $0 | $06000000 | per jumpers |
| OUT | IN | IN | OUT | OUT | OUT | $0 | $0 | $07000000 | per jumpers |
| OUT | IN | OUT | IN | IN | IN | $0 | $0 | $08000000 | per jumpers |
| OUT | IN | OUT | IN | IN | OUT | $0 | $0 | $09000000 | per jumpers |
| OUT | IN | OUT | IN | OUT | IN | $0 | $0 | $0A000000 | per jumpers |
| OUT | IN | OUT | IN | OUT | OUT | $0 | $0 | $0B000000 | per jumpers |
| OUT | IN | OUT | OUT | IN | IN | $0 | $0 | $00000000 @ | per jumpers |
| OUT | IN | OUT | OUT | IN | OUT | $0 | $0 | $01000000 @ | per jumpers |
| OUT | IN | OUT | OUT | OUT | IN | $0 | $0 | $55000000 | per jumpers |
| OUT | IN | OUT | OUT | OUT | OUT | $0 | $0 | $AA000000 | per jumpers |

Notes: IN = Jumper cap installed. OUT = Jumper cap removed. x = Don't care.
@ = These two maps are special decoding maps for decoding of A24 address space.

## Table B-3. MODE 3: USE LOCAL DRAM, TRACK VMEbus

| Jumper Block J4 (see notes) | | | | | | Debugger | Local | VMEbus | VSB and |
|---|---|---|---|---|---|---|---|---|---|
| NVRAM* | TRACK | VMEbus | | | | Memory | Memory | Memory | VMEchip-GCSR |
| (5-6) | (3-4) | (15-16) | (17-18) | (19-20) | (21-22) | Start | Mapping | Mapping | Mapping |
| OUT | OUT | IN | IN | IN | IN | $00000000 | $00000000 | $00000000 | per jumpers |
| OUT | OUT | IN | IN | IN | OUT | $01000000 | $01000000 | $01000000 | per jumpers |
| OUT | OUT | IN | IN | OUT | IN | $02000000 | $02000000 | $02000000 | per jumpers |
| OUT | OUT | IN | IN | OUT | OUT | $03000000 | $03000000 | $03000000 | per jumpers |
| OUT | OUT | IN | OUT | IN | IN | $04000000 | $04000000 | $04000000 | per jumpers |
| OUT | OUT | IN | OUT | IN | OUT | $05000000 | $05000000 | $05000000 | per jumpers |
| OUT | OUT | IN | OUT | OUT | IN | $0 (off-board) | disabled | $06000000 | per jumpers |
| OUT | OUT | IN | OUT | OUT | OUT | $0 (off-board) | disabled | $07000000 | per jumpers |
| OUT | OUT | OUT | IN | IN | IN | $00000000 | $00000000 | $08000000 | per jumpers |
| OUT | OUT | OUT | IN | IN | OUT | $01000000 | $01000000 | $09000000 | per jumpers |
| OUT | OUT | OUT | IN | OUT | IN | $02000000 | $02000000 | $0A000000 | per jumpers |
| OUT | OUT | OUT | IN | OUT | OUT | $03000000 | $03000000 | $0B000000 | per jumpers |
| OUT | OUT | OUT | OUT | IN | IN | $04000000 | $04000000 | $00000000 | per jumpers |
| OUT | OUT | OUT | OUT | IN | OUT | $05000000 | $05000000 | $01000000 | per jumpers |
| OUT | OUT | OUT | OUT | OUT | IN | $0 (off-board) | disabled | $55000000 | per jumpers |
| OUT | OUT | OUT | OUT | OUT | OUT | $0 (off-board) | disabled | $AA000000 | per jumpers |

Notes: IN = Jumper cap installed. OUT = Jumper cap removed. x = Don't care.

# APPENDIX C
# VMEchip LOCAL CONTROL & STATUS REGISTER SET

The CSR bits for the local control and status register (LCSR) of the VMEchip are illustrated in Figure C-1. The register names, bit names, and access modes for the LCSR bits are provided in the accompanying table.

| | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| $01 | --- | --- | --- | --- | ROBIN | BDFAIL | SRESET | SCON |
| $03 | DWB | DHB | RONR | RWD | RNEVER | --- | RQLEV1 | RQLEV0 |
| $05 | --- | --- | MASWP | CFILL | MASUAT | MASA16 | MASA24 | MASD16 |
| $07 | SLVEN | --- | SLVWP | --- | --- | --- | --- | SLVD16 |
| $09 | --- | ARBTO | VBTO1 | VBTO0 | ACTO1 | ACTO0 | LBTO1 | LBTO0 |
| $0B | SUPER | USER | EXTED | STND | SHORT | BLOCK | PROGRM | DATA |
| $0D | AMSEL | --- | AM5 | AM4 | AM3 | AM2 | AM1 | AM0 |
| $0F | IEN7 | IEN6 | EN5 | IEN4 | IEN3 | IEN2 | IEN1 | --- |
| $11 | WPEREN | SFIEN | SIGHEN | LM1EN | IACKEN | LM0EN | SIGLEN | --- |
| $13 | UVB7 | UVB6 | UVB5 | UVB4 | UVB3 | UID2 | UID1 | UID0 |
| $15 | --- | --- | --- | --- | --- | IL2 | IL1 | IL0 |
| $17 | D07 | D06 | D05 | D04 | D03 | D02 | D01 | D00 |
| $19 | --- | --- | --- | --- | RMCERR | VBERR | ACTO | LBTO |
| $1B | --- | --- | --- | --- | GCSRA7 | GCSRA6 | GCSRA5 | GCSRA4 |

**Figure C-1.** VMEchip LCSR Bit Assignments

**Table C-1.** VMEchip LCSR Register Summary

| LCSR Register/Bit | Bit Name | Access |
|---|---|---|
| Offset $01 - System Controller Configuration Register | | |
| ROBIN | Round Robin Select | Read/Write |
| BRDFAIL | Board Fail | Read/Write |
| SRESET | System Reset | Read/Write |
| SCON | System Controller On | Read only |
| Offset $03 - Requester Configuration Register | | |
| DWB | Device Wants Bus | Read/Write |
| DHB | Device Has Bus | Read only |
| RONR | Release On No Request | Read/Write |
| RWD | Release When Done | Read/Write |
| RNEVER | Release Never | Read/Write |
| RQLEV1 | Request Level 1 | Read/Write |
| RQLEV0 | Request Level 0 | Read/Write |
| Offset $05 - Master Configuration Register | | |
| MASWP | Master Write-Posting | Read/Write |
| CFILL | Cache Fill | Read/Write |
| MASUAT | Master Unaligned Transfers | Read/Write |
| MASA16 | Master A16 | Read/Write |
| MASA24 | Master A24 | Read/Write |
| MASD16 | Master D16 | Read/Write |
| Offset $07 - Slave Configuration Register | | |
| SLVEN | Slave Enable | Read/Write |
| SLVWP | Slave Write Posting | Read/Write |
| SLVD16 | Slave D16 | Read/Write |
| Offset $09 - Timer Configuration Register | | |
| ARBTO | Arbitration Timer | Read/Write |
| VBTO1 | VMEbus Time Out 1 | Read/Write |
| VBTO0 | VMEbus Time Out 0 | Read/Write |
| ACTO1 | Access Time Out 1 | Read/Write |
| ACTO0 | Access Time Out 0 | Read/Write |
| LBTO1 | Local Bus Time Out 1 | Read/Write |
| LBTO0 | Local Bus Time Out 0 | Read/Write |

**Table C-1.** VMEchip LCSR Register Summary (cont'd)

| LCSR Register/Bit | Bit Name | Access |
|---|---|---|
| Offset $0B - Slave Address Modifier Register | | |
| SUPER | Supervisor Address Space | Read/Write |
| USER | User Address Space | Read/Write |
| EXTED | Extended (32-bit) Addressing Range | Read/Write |
| STND | Standard (24-bit) Addressing Range | Read/Write |
| SHORT | Short (16-bit) Addressing Range | Read/Write |
| BLOCK | Block Transfer | Read/Write |
| PROGRAM | Program Space | Read/Write |
| DATA | Data Space | Read/Write |
| Offset $0D - Master Address Modifier Register | | |
| AMSEL | Address Modifier Select | Read/Write |
| AM5 | Address Modifier Code 5 | Read/Write |
| AM4 | Address Modifier Code 4 | Read/Write |
| AM3 | Address Modifier Code 3 | Read/Write |
| AM2 | Address Modifier Code 2 | Read/Write |
| AM1 | Address Modifier Code 1 | Read/Write |
| AM0 | Address Modifier Code 0 | Read/Write |
| Offset $0F - Interrupt Handler Mask Register | | |
| IEN7 | Interrupt Enable 7 | Read/Write |
| IEN6 | Interrupt Enable 6 | Read/Write |
| IEN5 | Interrupt Enable 5 | Read/Write |
| IEN4 | Interrupt Enable 4 | Read/Write |
| IEN3 | Interrupt Enable 3 | Read/Write |
| IEN2 | Interrupt Enable 2 | Read/Write |
| IEN1 | Interrupt Enable 1 | Read/Write |
| Offset $11 - Utility Interrupt Mask Register | | |
| WPEREN | Write Posting Bus Error Enable | Read/Write |
| SFIEN | SYSFAIL Interrupt Enable | Read/Write |
| SIGHEN | Signal High Priority Interrupt Enable | Read/Write |
| LM1EN | Location Monitor 1 Interrupt Enable | Read/Write |
| IACKEN | IACK Interrupt Enable | Read/Write |
| LM0EN | Location Monitor 0 Interrupt Enable | Read/Write |
| SIGLEN | Signal Low Priority Interrupt Enable | Read/Write |

**Table C-1.** VMEchip LCSR Register Summary (cont'd)

| LCSR Register/Bit | Bit Name | Access |
|---|---|---|
| Offset $13 - Utility Interrupt Vector Register | | |
| UVB7 | Utility Vector 7 | Read/Write |
| UVB6 | Utility Vector 6 | Read/Write |
| UVB5 | Utility Vector 5 | Read/Write |
| UVB4 | Utility Vector 4 | Read/Write |
| UVB3 | Utility Vector 3 | Read/Write |
| UID2 | Utility Vector ID 2 | Read only |
| UID1 | Utility Vector ID 1 | Read only |
| UID0 | Utility Vector ID 0 | Read only |
| Offset $15 - Interrupt Request Register | | |
| IL2 | Interrupt Level Select 2 | Read/Write |
| IL1 | Interrupt Level Select 1 | Read/Write |
| IL0 | Interrupt Level Select 0 | Read/Write |
| Offset $17 - VMEbus Status/ID Register | | |
| D07 | (VMEbus IACK vector) Data 07 | Read/Write |
| D06 | (VMEbus IACK vector) Data 06 | Read/Write |
| D05 | (VMEbus IACK vector) Data 05 | Read/Write |
| D04 | (VMEbus IACK vector) Data 04 | Read/Write |
| D03 | (VMEbus IACK vector) Data 03 | Read/Write |
| D02 | (VMEbus IACK vector) Data 02 | Read/Write |
| D01 | (VMEbus IACK vector) Data 01 | Read/Write |
| D00 | (VMEbus IACK vector) Data 00 | Read/Write |
| Offset $19 - Bus Error Status Register | | |
| RMCERR | RMC Lock Error | Read only |
| VBERR | VMEbus Bus Error | Read only |
| ACTO | Access Time Out | Read only |
| LBTO | Local Bus Time Out | Read only |
| Offset $1B - GCSR Base Address Register | | |
| GCSRA7 | GCSR Base Address Decode 7 | Read/Write |
| GCSRA6 | GCSR Base Address Decode 6 | Read/Write |
| GCSRA5 | GCSR Base Address Decode 5 | Read/Write |
| GCSRA4 | GCSR Base Address Decode 4 | Read/Write |

# APPENDIX D

# VMEchip GLOBAL CONTROL & STATUS REGISTER SET

The CSR bit assignments for the global control and status register (GCSR) of the VMEchip are illustrated in Figure D-1. The register names, bit names, and access modes for the GCSR bits are provided in the accompanying table.

| | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| $01 | LM3 | LM2 | LM1 | LM0 | CHPID3 | CHPID2 | CHPID1 | CHPID0 |
| $03 | R&H | SCON | ISF | BDFAIL | --- | --- | SIGHP | SIGLP |
| $05 | BRDID7 | BRDID6 | BRDID5 | BRDID4 | BRDID3 | BRDID2 | BRDID1 | BRDID0 |
| $07 | GENERAL PURPOSE CONTROL AND STATUS REGISTER 0 | | | | | | | |
| $09 | GENERAL PURPOSE CONTROL AND STATUS REGISTER 1 | | | | | | | |
| $0B | GENERAL PURPOSE CONTROL AND STATUS REGISTER 2 | | | | | | | |
| $0D | GENERAL PURPOSE CONTROL AND STATUS REGISTER 3 | | | | | | | |
| $0F | GENERAL PURPOSE CONTROL AND STATUS REGISTER 4 | | | | | | | |

**Figure D-1.** VMEchip GCSR Bit Assignments

**Table D-1.** VMEchip GCSR Register Summary

| GCSR Register/Bit | Bit Name | Local Access | Global Access |
|---|---|---|---|
| Offset $01 - Global Register 0 | | | |
| LM3 | Location Monitor 3 | Read/Set | Read/Set |
| LM2 | Location Monitor 2 | Read/Set | Read/Set |
| LM1 | Location Monitor 1 | Read/Set | Read/Set |
| LM0 | Location Monitor 0 | Read/Set | Read/Set |
| CHIPID3 | Chip ID 3 | Read Only | Read Only |
| CHIPID2 | Chip ID 2 | Read Only | Read Only |
| CHIPID1 | Chip ID 1 | Read Only | Read Only |
| CHIPID0 | Chip ID 0 | Read Only | Read Only |
| Offset $03 - Global Register 1 | | | |
| R&H | Reset and Hold | Read/Write | Read/Write |
| SCON | System Controller On | Read Only | Read Only |
| ISF | Inhibit SYSFAIL | Read/Write | Read/Write |
| BRDFAIL | Board Fail | Read Only | Read Only |
| SIGHP | Signal High Priority | Read/Clear | Read/Set |
| SIGLP | Signal Low Priority | Read/Clear | Read/Set |
| Offset $05 - Board Identification Register | | | |
| BRDID7 | Board Identification 7 | Read/Write | Read Only |
| BRDID6 | Board Identification 6 | Read/Write | Read Only |
| BRDID5 | Board Identification 5 | Read/Write | Read Only |
| BRDID4 | Board Identification 4 | Read/Write | Read Only |
| BRDID3 | Board Identification 3 | Read/Write | Read Only |
| BRDID2 | Board Identification 2 | Read/Write | Read Only |
| BRDID1 | Board Identification 1 | Read/Write | Read Only |
| BRDID0 | Board Identification 0 | Read/Write | Read Only |
| Offset $07 - General Purpose CSR 0 | | Read/Write | Read/Write |
| Offset $09 - General Purpose CSR 1 | | Read/Write | Read/Write |
| Offset $0B - General Purpose CSR 2 | | Read/Write | Read/Write |
| Offset $0D - General Purpose CSR 3 | | Read/Write | Read/Write |
| Offset $0F - General Purpose CSR 4 | | Read/Write | Read/Write |

**Table C-1.** VMEchip LCSR Register Summary (cont'd)

| LCSR Register/Bit | Bit Name | Access |
|---|---|---|
| Offset $0B - Slave Address Modifier Register | | |
| SUPER | Supervisor Address Space | Read/Write |
| USER | User Address Space | Read/Write |
| EXTED | Extended (32-bit) Addressing Range | Read/Write |
| STND | Standard (24-bit) Addressing Range | Read/Write |
| SHORT | Short (16-bit) Addressing Range | Read/Write |
| BLOCK | Block Transfer | Read/Write |
| PROGRAM | Program Space | Read/Write |
| DATA | Data Space | Read/Write |
| Offset $0D - Master Address Modifier Register | | |
| AMSEL | Address Modifier Select | Read/Write |
| AM5 | Address Modifier Code 5 | Read/Write |
| AM4 | Address Modifier Code 4 | Read/Write |
| AM3 | Address Modifier Code 3 | Read/Write |
| AM2 | Address Modifier Code 2 | Read/Write |
| AM1 | Address Modifier Code 1 | Read/Write |
| AM0 | Address Modifier Code 0 | Read/Write |
| Offset $0F - Interrupt Handler Mask Register | | |
| IEN7 | Interrupt Enable 7 | Read/Write |
| IEN6 | Interrupt Enable 6 | Read/Write |
| IEN5 | Interrupt Enable 5 | Read/Write |
| IEN4 | Interrupt Enable 4 | Read/Write |
| IEN3 | Interrupt Enable 3 | Read/Write |
| IEN2 | Interrupt Enable 2 | Read/Write |
| IEN1 | Interrupt Enable 1 | Read/Write |
| Offset $11 - Utility Interrupt Mask Register | | |
| WPEREN | Write Posting Bus Error Enable | Read/Write |
| SFIEN | SYSFAIL Interrupt Enable | Read/Write |
| SIGHEN | Signal High Priority Interrupt Enable | Read/Write |
| LM1EN | Location Monitor 1 Interrupt Enable | Read/Write |
| IACKEN | IACK Interrupt Enable | Read/Write |
| LM0EN | Location Monitor 0 Interrupt Enable | Read/Write |
| SIGLEN | Signal Low Priority Interrupt Enable | Read/Write |

**Table C-1.** VMEchip LCSR Register Summary (cont'd)

| LCSR Register/Bit | Bit Name | Access |
|---|---|---|
| Offset $13 - Utility Interrupt Vector Register | | |
| UVB7 | Utility Vector 7 | Read/Write |
| UVB6 | Utility Vector 6 | Read/Write |
| UVB5 | Utility Vector 5 | Read/Write |
| UVB4 | Utility Vector 4 | Read/Write |
| UVB3 | Utility Vector 3 | Read/Write |
| UID2 | Utility Vector ID 2 | Read only |
| UID1 | Utility Vector ID 1 | Read only |
| UID0 | Utility Vector ID 0 | Read only |
| Offset $15 - Interrupt Request Register | | |
| IL2 | Interrupt Level Select 2 | Read/Write |
| IL1 | Interrupt Level Select 1 | Read/Write |
| IL0 | Interrupt Level Select 0 | Read/Write |
| Offset $17 - VMEbus Status/ID Register | | |
| D07 | (VMEbus IACK vector) Data 07 | Read/Write |
| D06 | (VMEbus IACK vector) Data 06 | Read/Write |
| D05 | (VMEbus IACK vector) Data 05 | Read/Write |
| D04 | (VMEbus IACK vector) Data 04 | Read/Write |
| D03 | (VMEbus IACK vector) Data 03 | Read/Write |
| D02 | (VMEbus IACK vector) Data 02 | Read/Write |
| D01 | (VMEbus IACK vector) Data 01 | Read/Write |
| D00 | (VMEbus IACK vector) Data 00 | Read/Write |
| Offset $19 - Bus Error Status Register | | |
| RMCERR | RMC Lock Error | Read only |
| VBERR | VMEbus Bus Error | Read only |
| ACTO | Access Time Out | Read only |
| LBTO | Local Bus Time Out | Read only |
| Offset $1B - GCSR Base Address Register | | |
| GCSRA7 | GCSR Base Address Decode 7 | Read/Write |
| GCSRA6 | GCSR Base Address Decode 6 | Read/Write |
| GCSRA5 | GCSR Base Address Decode 5 | Read/Write |
| GCSRA4 | GCSR Base Address Decode 4 | Read/Write |

# APPENDIX E
# MVME165 INITIALIZATION SCHEME

The following describes the initialization of the MVME165 processor board by the debugger. The general order is (1) cpu registers, (2) 165 - csr, (3) LRC, (4) VSBchip, (5) VMEchip, and (6) DUART.

1. **CPU Registers** - The first thing to be initialized is the MC68040's transparent translation registers. The debugger does not use the MMU proper of the 040, but relies on transparent translation. It is necessary to set up one of the two data transparent translation registers to assure that data-space accesses to addresses corresponding to the MVME165 I/O devices will be serialized and non-cacheable. The debugger initialization of the transparent translation registers is as follows.

| MC68040 Register | Initialization Value |
|---|---|
| ITT0 | $00000000 |
| DTT0 | $00000000 |
| ITT1 | $00000000 |
| DTT1 | $E01FC040 |

The MC68040's stack pointer should also be set up very early in the Initialization. Remember that the initial SP value must correspond to a valid memory location and the stack pointer will decrement as the stack grows.

2. **MVME165 Local CSR** - It is important to set this up early in the initialization because it controls where the local DRAM array will appear on all three busses (i.e., zero on all three busses by writing the Local CSR as follows.

| Register | Address | Initialization Value |
|---|---|---|
| 165 CSR 1 | $FFFE0000 | $0000000A |

3. **Local Resource Controller (LCR)** - The following table illustrates how the LRC gate-array is set up. Those registers that are not on this initialization table are not used by the debugger, or come out of reset with an acceptable value and will be initialized by the user's application.

Table E-1. LCR Registers

| LCR Register | Address | Init Value | Comments |
|---|---|---|---|
| LRC Tmr Ctl 0 | $FFF90018 | $XX000000 | Timer 0 and its interrupt disabled. |
| LRC Tmr Ctl 1 | $FFF9001C | $XX000000 | Timer 1 and its interrupt disabled. |
| LRC Tmr Ctl 2 | $FFF90020 | $XX000000 | Timer 2 and its interrupt disabled. Watchdog timer reset disabled. |
| LRC Gen Ctl | $FFF90024 | $XXxxxx00 | Bit 4 (D4) must be clear to obtain the correct serial clock frequency for DUART. Reset not asserted (D0), Parity-err BERRs/ints disabled (D1), Write wrong parity disabled (D2), slow (250/200 nsec) EPROM Ack (D3). |
| LRC Bus Ctl | $FFF90028 | $XXxxxx00 | VSB R/O mode affects entire Bus (D0), VSB R/O mode disabled (D1), VSB disabled (D3), VME 24-bit address option disabled (D4). |
| LRC Int Ctl 0 | $FFF9002C | $XXxxxx80 | Global Int Enable asserted (D7) to allow ABORT, ACFAIL and VME Ints. Groups 2 and 4 ints disabled (D0-D6). |
| LRC Int Ctl 1 | $FFF90030 | $XXxxxx43 | IRQ7G5 (ABORT;D0) and IRQ7G6 (ACFAIL;D1) interrupts enabled. |
| LRC Status | $FFF90034 | $XXx1AFFF | Clear all status. |
| LRC RAM Addr Register | $FFF90038 | $XXx03000<br>$XXx0F000 | Local Addr at zero (D0-D11). A20 and A21 address lines masked (not driven by LRC) (D12-D18). |

NOTES:

1. All address and data references are shown as a Hex ($) value. XX in the data field indicates that the LRC does not drive or receive and an x in the data field indicates that the LRC does not implement part or all of that byte.

2. If the board being initialized is a 4Mb MVME165 (version -01 or -03); then use the initialization value $XXx03000. If the board being initialized is a 16Mb MVME165 (version -02 or -04); then use the initialization value $XXx0F000.

4. **VSBchip** - The VSB gate array is set up by writing to its CSR as a single longword. This set up initially enables the VSB and selects bounce mode for VSB accesses.

| Register | Address | Initialization Value |
|---|---|---|
| VSBchip CSR | $FFFA0000 | $FCBEFCBE |

5. **VMEchip** - The VMEchip is set up as shown in the table below. Group 1 interrupts from the VMEbus must be enabled while the debugger is running in order to be able to boot UNIX. Registers that are not shown will only need to be set up if the user's application so demands.

**Table E-2.** VMEchip Registers

| Register | Address | Init Value | Comments |
|---|---|---|---|
| SCCR | $FFFB0001 | $00 | Priority arbitration mode is selected. BRDFAIL is negated. |
| RCT | $FFFB0003 | $03 | VMEbus request level 3. |
| MCR | $FFFB0005 | $00 | Use PBERR and PHALT to break deadlocks. |
| SCR | $FFFB0007 | $80 | Slave accesses are enabled. VMEchip write posting is disabled. |
| TCR | $FFFB0009 | $6A | Local, VME and VME acc timeouts are enabled. |
| SAMR | $FFFB000B | $F3 | Does not respond to "short" or "block" AMs. |
| IHMR | $FFFB000F | $FF | All VMEbus interrupts are enabled. |
| UIMR | $FFFB0011 | $00 | All utility interrupts are masked. |
| UIVR | $FFFB0013 | $60 | Utility init vectors are $61-$67. |
| IRR | $FFFB0015 | $00 | All IRQ lines are negated. |
| SIDR | $FFFB0017 | $0F | "Un-initialized interrupt" vector. |
| GBAR | $FFFB001B | $0F | GCSR disabled (will not appear). |
| BIDR | $FFFB0025 | $00 | Board ID = 0 (global csr register). |

6. **DUART** - The MC68681 DUART may be set up by a series of byte-writes as shown in the sequence presented in the table below. Note that multiple accesses to the same register is necessary in some cases for proper operation.

**Table E-3.** DUART Registers

| Register | Address | Data | Comments |
|---|---|---|---|
| Set-up Channel A | | | |
| CRA | $FFFD000B | $10 | Reset mode register pointer |
| CRA | $FFFD000B | $20 | Reset receiver |
| CRA | $FFFD000B | $30 | Reset transmitter |
| CRA | $FFFD000B | $40 | Reset error status |
| CRA | $FFFD000B | $50 | Reset break change interrupt |
| MR1A | $FFFD0003 | $13 | No parity, 8 bits per character |
| MR2A | $FFFD0003 | $07 | No RTS/CTS flow-control, 1 stop bit |
| CSRA | $FFFD0007 | $BB | Port A: 9600 baud |
| Set-up Channel B | | | |
| CRB | $FFFD002B | $10 | Reset mode register pointer |
| CRB | $FFFD002B | $20 | Reset receiver |
| CRB | $FFFD002B | $30 | Reset transmitter |
| CRB | $FFFD002B | $40 | Reset error status |
| CRB | $FFFD002B | $50 | Reset break change interrupt |
| MR1B | $FFFD0023 | $13 | No parity, 8 bits per character |
| MR2B | $FFFD0023 | $07 | No RTS/CTS flow-control, 1 stop bit |
| CSRB | $FFFD0027 | $BB | Port B: 9600 baud |
| Set-up Global Registers | | | |
| ACR | $FFFD0013 | $E0 | Timer mode, clocked externally |
| IMR | $FFFD0017 | $00 | Mask all interrupts (polled environment |
| OPCR | $FFFD0037 | $04 | OP3 = output of timer |
| IVR | $FFFD0033 | $4C | Serial-port interrupt vector = $4C |
| Enable Receivers/Transmitters | | | |
| CRA | $FFFD000B | $05 | Port A: enable receiver and transmitter |
| CRB | $FFFD002B | $05 | Port B: enable receiver and transmitter |

**NOTE:** The above set-up routine assumes no hardware flow control (i.e., RTS/CTS. To use these handshaking lines, make the following substitution: MR1A = MR1B = $93, MR2A = MR2B = $37.

The Interrupt Vector Register (IVR) of the DUART must be set up to provide an appropriate vector for the serial port interrupt from the MC68681 (IRQ5G2; "DRTIRQ"). This interrupt is not handled through the LRC, but may be masked in the LRC.

The DUART's counter/timer interrupt is handled entirely through the LRC in order to obtain a unique vector for this event. The LRC puts out the vector corresponding to IRQ6G2 (TICKTMR).

# APPENDIX F
# CONFIGURING A VME SUBSYSTEM BUS

The MVME165 module gains much of its power and flexibility by use of the VME Subsystem Bus (VSB). VSB is implemented in a gate array device and performs almost all VSB requirements.

VSB on the MVME165 is a master and slave, multiple master interface. Multiple MVME165s can be used on a single subsystem and share VSB slave resources. VSB can be implemented with a 64-pin ribbon cable. In this configuration, only two MVME165s may share VSB with up to four VSB slave devices. It is important to note that when a ribbon cable is used, at least one of the slave devices must connect VSB BGIN* to VSB BG* (P2 A31 to P2 C32).

VSB backplanes are available in varied configurations, typically up to six slots. These backplanes plug into P2 connectors on the VME P2 backplane. VSB backplanes are needed for applications requiring more than two masters. The backplane will require jumpering for configuring VSB BGIN* to BG* for each slot where a VSB master such as the MVME165 is installed.

Socketed signal terminators (R10, R11 and R12) are provided on the MVME165 module. If a ribbon cable is used, these terminators do not require reconfiguration and should remain installed in the module.

If a VSB backplane is used that provides proper VSB signal termination, then R10, R11 and R12 must be removed. Also, if more than two MVME165s are installed in a single subsystem, then the terminators should only be installed in the modules that reside at each end of the subsystem bus. Proper VSB signal termination is essential for reliable VSB operation. Refer to the Motorola publication MVMESB (*VSB - Parallel Sub System Bus of the IEC 821 Bus*) for more detailed information on VSB configuration and signal terminators.

# APPENDIX G
# VSB CONTROL AND STATUS REGISTER

The VSB Control and Status Register (CSR) allows software to control the operation of the VSBchip and to determine the cause of a bus error. The following figure shows the organization of the CSR. The CSR is accessed at an offset of $00 from the base address of the registers as defined by the contents of the address decode register. After the chip undergoes a reset sequence, and as long as the address decode register has not been changed, the CSR's address is $FFFA0000. Bits 0-10 of the CSR provide control bits, while bits 11-1 provide status bits. Upon reset, the Control register, i.e., bits 00-10 of the CSR, are initialized to the binary value carried on the local data lines D00-D10. This feature allows the VSBchip to be configured to operate on a board that does not include a microprocessor. The CSR can be read and written to by software, allowing for dynamic reconfiguration of the operation mode of the chip.

| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 |
|---|---|---|---|---|---|---|---|
| ERR* | ASACK1* | ASACK0* | TIMEOUT* | WRERROR* | FAIR* | ENTO0* | ENTO1* |

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| --- | MASTEREN* | READONLY* | BOUNCE | TEN* | BLOCKEN* | --- | SCON* |

**FIGURE G-1.** VSB Control and Status Register

### Bit 00 - SCON* - System Controller On

The SCON* control bit is used to enable the VSBchip to operate as the system controller. Clearing this bit to 0 enables the chip's arbiter.

### Bit 02 - BLOCKEN* - Enable Block-Transfer Mode

In conjunction with the block-transfer byte count register, this bit allows software to initiate a block-transfer cycle. Writing a non zero value into the block-transfer byte-count register and then clearing BLOCKEN* to 0 will cause the master to operate in the block-transfer mode. The master will remain in the block-transfer mode until the byte-count register counts down to zero or ERR* was asserted in the course of the cycle which ever is encountered first.

### Bit 03 - TEN* - Enable MC68010 Mode

The TEN* control bit allows software to configure the VSBchip to interface an MC68010 microprocessor to the VSB. THis is accomplished by clearing the TEN* bit to 0. When in this mode, the master combines the upper eight bits of the block-transfer byte-count register with the 24 bits of address driven by the MC68010 to get the 32-bit address required by the VSB. In addition, during the data transfer phase of VSB cycles, the master swaps and replicates data bytes so as to comply with the data positioning requirements of both the VSB and the MC68010. When the TEN* control bit is set, the MC68010 mode is disabled, and the master is configured to operate with a MC68020/ MC68030 microprocessor.

### Bit 04 - BOUNCE - Enable Address Bounce

The BOUNCE control bit allows software to increase the concurrency of the address decode and the address broadcast phase. Setting the BOUNCE bit to 1 enables the master to operate in the address-bounce mode. Clearing the bit to zero disables the address-bounce mode.

### Bit 05 - READONLY* - Enable Read-Only Mode

The READONLY* bit allows software to configure the chip to only use the VSB for the purpose of reading data. Clearing the bit to 0 enables the read-only mode, while setting it to one disables it.

### Bit 06 - MASTEREN* - Enable Master

The VSBchip allows onboard logic to disable the chip's VSB master by setting the MASTEREN* control bit to 1. This feature might be used when the chip is used to only provide a slave interface to the VSB. In this case, onboard logic will set the bit to one by driving D06 to high when RESET* is asserted. Clearing the MASTEREN* bit to 0 enables the master. However, it should be noted that the master will indeed be enabled only when both MASTEREN* is cleared and the SLAVE* input signal is negated.

### Bit 08 - ENTO1* - Enable Timeout 1
### Bit 09 - ENTO0* - Enable Timeout 0

The transfer timer allows the subsystem to recover from a lockup condition when, due to a failure, a VSB slave does not properly respond to either address broadcast or the data transfer phase of VSB cycle. If a time-out occurs, the chip asserts the ERR* output signal, and the TIMEOUT* bit in the status register is cleared. The time-out period of the transfer timer is encoded as shown in the following table.

**TABLE G-1.** Programming the Transfer Timer

| ENTO1 | ENTO0 | Time-Out Period |
|---|---|---|
| 0 | 0 | 64 microseconds |
| 0 | 1 | 126 microseconds |
| 1 | 0 | 256 microseconds |
| 1 | 1 | Timer disabled |

### Bit 10 - FAIR* - Enable Fair Request Mode

The FAIR* bit is used to configure the requester to operate in the FAIR mode. Clearing the FAIR* bit to 0 will cause the requester to requesting the bus until the bus request line is negated. This allows bus requests from boards that are installed down the daisy-chain to be granted.

**NOTE**

> Fairness in accessing the bus will be achieved only if all the boards in the subsystem are configured to operate in the FAIR mode.

Bits 11-15 of the CSR reflect the status when the most recent error condition was detected, i.e., ERR* was asserted. Reading the register causes all of its bits to be set to 1.

### Bit 11 - WRERR* - Write Error Status

When cleared, this status bit indicates that onboard logic has attempted a write cycle to the VSB when the chip was configured to operate in the read-only mode.

### Bit 12 - TIMEOUT* - Bus Time-Out Status

When cleared, this status bit indicates that the VSB transfer timer has timed out.

### Bit 13 - ASACK0* - Address and Size Status 0
### Bit 14 - ASACK1* - Address and Size Status 1

These two status bits reflect the state of the VSB ASACK0* and ASACK1* signal lines at the last time that ERR* was detected low.

### Bit 12 - ERR* - VSB Bus Error Status

When cleared, this status bit indicates that the VSB ERR* signal was asserted in the course of a cycle.

# INDEX

When using this index, keep in mind that a page number indicates only where referenced material begins; it may extend to the following page or pages.

INDEX

INDEX

INDEX

MOTOROLA

Computer Group
2900 South Diablo Way
Tempe, Arizona 85282
P.O. Box 2953
Phoenix, Arizona 85062-2953

To order additional Computer Group literature, contact your local sales office.

To comment on Motorola hardware, software or system products, contact:

Motorola Field Service Operations
Customer Support Center
1-800-551-1016

Just in Time

PRINTED IN USA IRONWOOD ♻