MVME334B Multiprotocol Communications Controller Module User's Manual

VME334BA/UM1

Notice

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

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

Restricted Rights Legend

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

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

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

Preface

The *MVME334B Multiprotocol Communications Module User's Manual* describes the installation, components, and configurations of the MVME334B. The documents should be used by anyone who wants general as well as technical information about the MVME334B.

Related Documentation

The following publications are applicable to the MVME334B and may provide additional helpful information. If not shipped with this product, they may be purchased by contacting your local Motorola sales office. Non-Motorola documents may be obtained from the sources listed.

Document Title	Motorola Publication Number
MVME334ABug Debugging Package User's Manual	MVME334ABUG
MVME709-1/ -2 Three Channel Transition Module User's Manual	MVME709-1/ -2
MVME709-1/ -2 Three Channel Transition Module Support Information	SIMVME709-1/ -2
MC68605 X.25 Protocol Controller (XPC) User's Manual	MC68605UM
MC68020 32-Bit Microprocessor User's Manual	MC68020UM

NOTE: Although not shown in the above list, each Motorola Computer Group manual publication number is suffixed with characters which represent the revision level of the document, such as / D2 or / UM1 (the second revision of a manual); a supplement bears the same number as the manual but has a suffix such as / A1 or / UM2A1 (the first supplement to the manual).

The following publications are available from the sources indicated.

ANSI/ IEEE Std. 1014-1987 Versatile Backplane Bus: The Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, New York, NY 10017, USA. (VMEbus specification)

SCN68562 Dual Universal Serial Communications Controller (DUSCC) Data Sheet; Signetics Corporation, 811 E. Arques Avenue, P.O Box 3409, Sunnyvale, CA 94088-3409.

8254 Programmable Interval Timer Data Sheet; Intel Corporation, 3065 Bowers Avenue, Santa Clara, CA 95051.

HD63450 DMA Controller Data Sheet; Hitachi America, Ltd., Semiconductor & IC Division, 2000 Sierra Point Parkway, Brisbane, CA 94005-1819.

Manual Terminology

Throughout this manual, a convention has been maintained whereby data and address parameters are preceded by a character which specifies the numeric format as follows:

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

Unless otherwise specified, all address references are in hexadecimal throughout this manual.

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

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

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

Safety Summary
Safety Depends On You

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

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

Ground the Instrument.

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

Do Not Operate in an Explosive Atmosphere.

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

Keep Away From Live Circuits.

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

Do Not Service or Adjust Alone.

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

Use Caution When Exposing or Handling the CRT.

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

Do Not Substitute Parts or Modify Equipment.

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

Dangerous Procedure Warnings.

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

WARNING

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

WARNING

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

CE

European Notice: Board products with the CE marking comply with the EMC Directive (89/ 336/ EEC). Compliance with this directive implies conformity to the following European Norms:

EN55022 (CISPR 22)	Radio Frequency Interference
EN50082-1 (IEC801-2, IEC801-3, IEEC801-4)	Electromagnetic Immunity

The product also fulfills EN60950 (product safety) which is essentially the requirement for the Low Voltage Directive (73/ 23/ EEC).

This board product was tested in a representative system to show compliance with the above mentioned requirements. A proper installation in a CE-marked system will maintain the required EMC/ safety performance.

Printed in the United States of America
December 1995

Contents

Chapter 4 How the MVME334B Works

Figures

Tables

General Information 1

Introduction

This manual provides general information, preparation for use and installation instructions, operating instructions, functional description, and support information for the MVME334B Multiprotocol Communications Controller Module.

Features

The features of the MVME334B include:

- MC68020 12.5 MHz microprocessor
- 4 MB RAM with parity (dual ported)
- Four-channel Direct Memory Access Controller (DMAC)
- Four serial ports with 50 to 38400 baud in asynchronous mode and
  4 megabits/ per second in synchronous mode
- Two X.25 serial ports with 4 megabits/ per second transfer rate
- Two 4 MB One-Time-Programmable (OTP) ROMs containing bootstrap and debug firmware
- 2 KB user-programmable EEPROM
- VMEbus interrupter
- VMEbus slave interface (A32:D16/ 8 or A24:D16/ 8 compatible)
- VMEbus requester
- FAIL LED on front panel
- RESET and ABORT switches on front panel

- Timers for periodic interrupt generation, malfunction monitoring, bus access supervision, and serial data clocking

Specifications

The MVME334B specifications are given in Table 1-1. Cooling requirements and FCC compliance are discussed in the sections following the table.

Table 1-1. MVME334B Specifications

Characteristics		Specifications
Power requirements		+5 Vdc, 4.5 A maximum (4.0 A typical)
Microprocessor		MC68020
Clock signal		12.5 MHz to MPU
Addressing	Total address range (on and off-board)	4 GB
	OTP PROM	8 MB (two 4 MB 44 pin devices, 256K x 32)
	EEPROM	2 KB
	Dynamic RAM	4 MB
I/ O ports		Six serial ports
Timers	Total number	10
	Watchdog	14-stage counter
	Bus	Monitors bus requests and data transfer cycles
	Multifunction	Four 16-bit counter/ timers (one for each serial channel)
	Programmable interval	One 16-bit time slice counter and two Tx/ Rx data clock generators
	DTACK generator	One for inserting two wait states during MPU local accesses

Table 1-1. MVME334B Specifications (Continued)

Characteristics		Specifications
Interrupt handler		Any or all onboard
Interrupt requester		Seven VMEbus interrupts
Bus arbitration		Local bus arbiter
Reset		RESET switch which can be enabled or disabled by software.
Operating temperature		0° to 55° C at point of entry of forced air (approximately 150 LFM)
Storage temperature		-40° to 85° C
Relative humidity		5% to 95% (noncondensing)
Physical characteristics (excluding front panel and mezzanine)	Height	9.187 inches (233.35 mm)
	Depth	6.299 inches (160.0 mm)
	Thickness	0.063 inches (1.6 mm)

Cooling Requirements

Motorola VMEmodules are specified, designed, and tested to operate reliably with an incoming air temperature range from 0° C to 55° C (32° F to 131° F) with forced air cooling.

Temperature qualification is performed in a standard Motorola VMEsystem chassis. Twenty-five watt load boards are inserted in the 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 71 CFM per fan, is placed directly under the MVME 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 150 LFM flowing over the module. Less air flow 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 air flow. 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 of air flowing over a module.

FCC Compliance

This VMEmodule (MVME334B) 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 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. Front panels screws 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 modules.

General Description

The MVME334B is a VMEmodule base board and mezzanine combination that provides all the hardware for a universal intelligent controller for serial data communications on six full duplex channels. Four of the channels are multiprotocol channels controlled by two SCN68562 Dual Channel Universal Communication Controllers (DUSCC), located on the base board. The other two channels are controlled by two MC68605 X.25 Protocol Controllers (XPC), located on the mezzanine. A 4-channel DMA Controller (DMAC) can be employed for data transfers between the DUSCC channels and local memory.

The module has sockets containing up to 8 MB of factory-supplied OTP ROM and 2 KB of EEPROM on the base board, and a complete 32-bit microcomputer consisting of an MC68020 MPU and 4 MB of RAM on the mezzanine. This microcomputer controls the DUSCC and XPC devices and has access to the VMEbus through an A32:D16/ 8 and A24:D16/ 8 master interface, either of which are software selectable. The host system has access to the MVME334B through an A32:D16/ 8 and A24:D16/ 8 slave interface that can be mapped anywhere into the 4 GB VMEbus address space.

The host can send interrupt, reset, and command bytes to the MVME334B and obtain status information from the MVME334B through a VMEbus command/ control area in the dual ported local RAM. VMEbus accesses are supported by bus arbitration and requester and interrupter logic. Dual port logic enables the local shared RAM to be accessed by either the local MPU or XPC devices or the DMA controller via the local bus or by the host via the VMEbus.

The MVME334B can also operate as a stand-alone unit via a minimum VMEbus system controller.

Data can be transferred by the local MPU between local memory and system memory, between local memory and DUSCC devices, between local memory and XPC devices, and between local

memory and DMAC/ DUSCC devices. The four DUSCC channels can be configured for either half or full duplex operation, controlled by the local MPU or DMAC.

Data transfer between local memory and the XPC devices is carried out under DMA control using the DMA controller integrated into each of the XPC devices.

The MVME334B is designed for high performance VMEbus systems, and because of its capabilities of data link control, data pre-/ post-processing, and system memory access, can relieve the host system of serial communications tasks.

The MVME334B is intended for use with the following software architecture:

- The host establishes a structure in system memory or in the MVME334B shared RAM for the transfer of high level commands and messages to and from the MVME334B.
- The MVME334B fetches commands from this structure, executes the commands and returns status messages.
- Both polled and interrupt driven modes of operation may be used for the host/ MVME334B software interface.

Software development for the MVME334B is facilitated with the MVME334ABug debug firmware package. This is a debug/ monitor program, contained in the OTP ROMs, that provides a self-contained programming and operating environment. The program interacts with the user through predefined commands that are entered via a terminal. These commands allow you to: display and modify memory or MVME334B registers, execute a program under various levels of control, and access input/ output resources.

Each of the six serial channels can be configured to conform to either the V.24 or V.35 standards. This is implemented on separate boards:

- MVME709-1 Three Channel Transition Module, which supports two X.25 channels and one multiprotocol channel.

- MVME709-2 Three Channel Transition Module, which supports three multiprotocol channels.

The 8-bit port portion of the local data bus (the transition board bus) is buffered and fed to connector P2 in order to provide additional control and monitor lines that may be required for certain expansions on the MVME709-1 and/ or MVME709-2. The transition modules are connected via ribbon cable to the A and C rows of the MVME334B connector P2.

Preparing and Installing Your MVME334B

2

Introduction

This chapter provides instructions for unpacking, preparing, and installing the MVME334B module.

Unpacking Your Hardware

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

Unpack the equipment from the shipping carton. Refer to the packing list and verify that all items are present. Save the packing material for storing and reshipping of the module.

Configuring Your MVME334B

To select the desired configuration and ensure proper operation of the MVME334B module, you may make certain changes to it before installation. You make these changes by setting jumpers on headers on the board. The location of the headers is illustrated in Figure 2-1.

The module has been factory tested and is shipped with factory-installed jumper configurations that are shown in the following sections with the header descriptions. The module is operational with the factory-installed jumpers; the factory configuration provides the system functions required for a VMEbus system.

You may wish to make changes in the jumper arrangements for the following conditions:

Function	Header
VMEbus functions select	J5
Bus time-out select	J6
VMEbus requester priority level select	J7
Module address mode select	J8
Module base address select	J9, J14
ABORT/ RESET switches, status register select	J12
XPC data clock select	J15, J16
Serial port configuration select (J18, J19)	J18, J19
DMAC request configuration select	J20

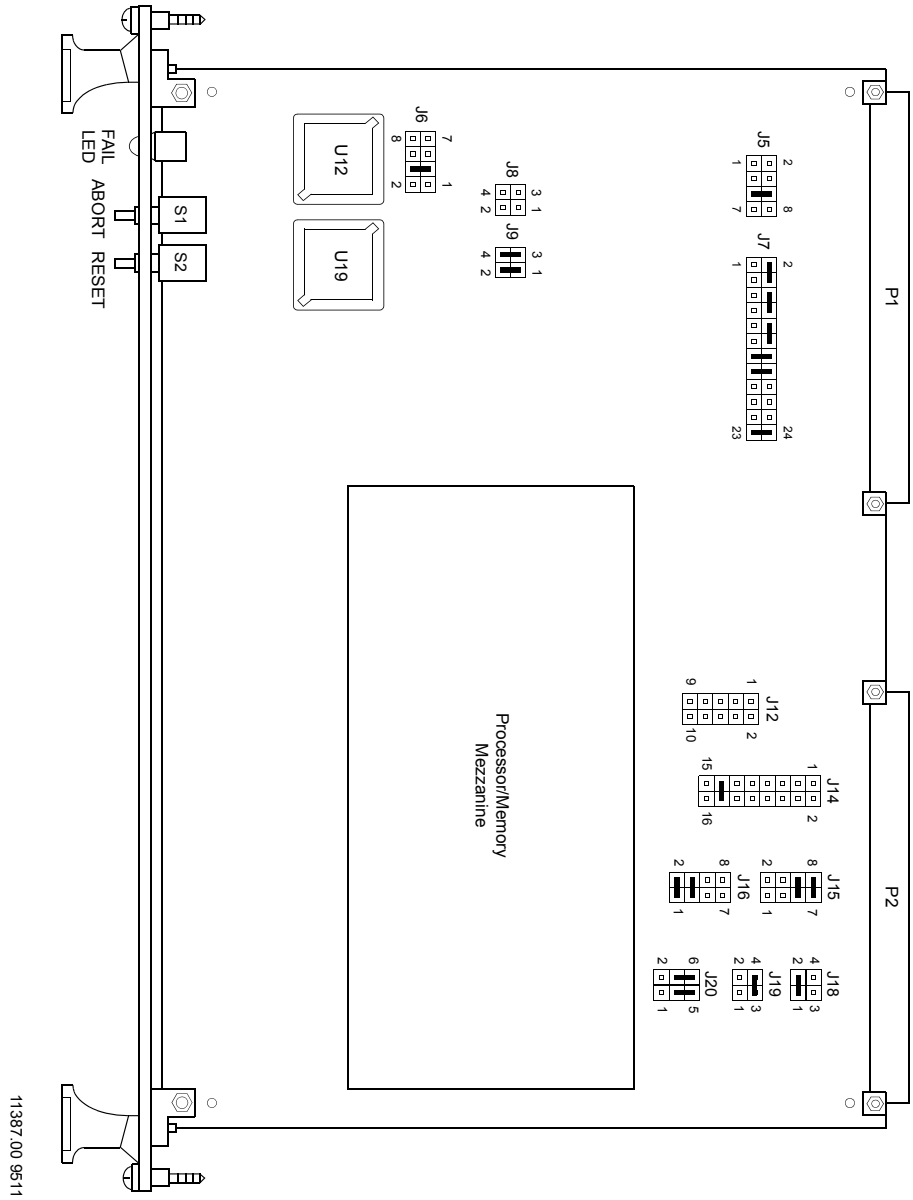
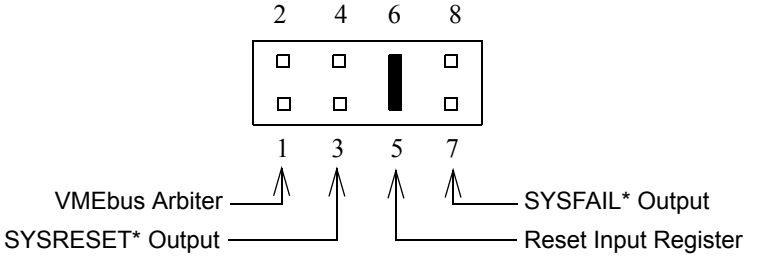


Figure 2-1. MVME334B Header Locations

VMEbus Functions Select Header (J5)

Header J5 is provided for use if the MVME334B is to be employed as the only bus master. A single level arbiter, reset circuitry, SYSRESET driver, and SYSFAIL monitoring are implemented and can be enabled if the MVME334B is assigned to be the system controller. The functions are enabled if the particular jumper is installed. The as-shipped factory configuration is with reset from VMEbus register enabled as shown below.

J5



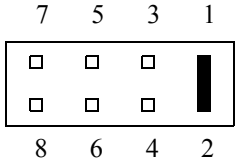
Bus Time-Out Select Header (J6)

Header J6 determines the time-out period of the bus timer which monitors all bus requests and data transfer cycles. The time-out period can be selected to be 57, 114, or 228 µs. A time-out period of infinity can also be selected, thus effectively disabling the bus timer.

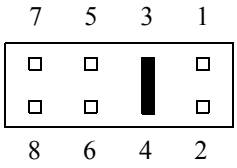
The bus time-out period must be set to a value greater than the longest time period that may elapse between the assertion of a bus request and the reception of a bus grant in the actual system configuration.

The as-shipped factory configuration is 114 µs as shown below:

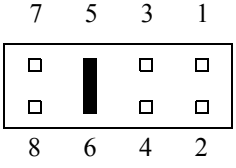
J6



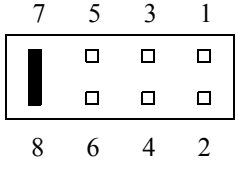
57 µs



114 µs
Factory Configuration



228 µs

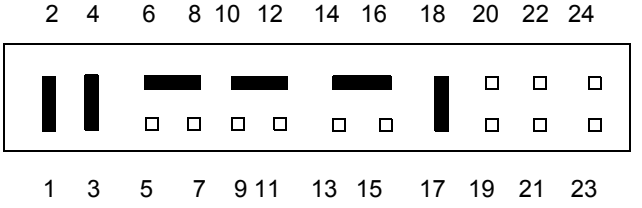


No Time-Out

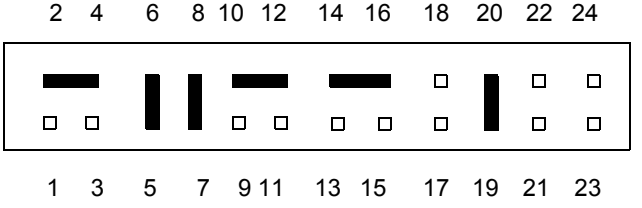
VMEbus Requester Priority Level Select Header (J7)

Header J7 determines the VMEbus requester priority level. The jumpers select the bus request output line and corresponding bus grant daisy chain. The as-shipped factory configuration is level 3 as shown below.

J7

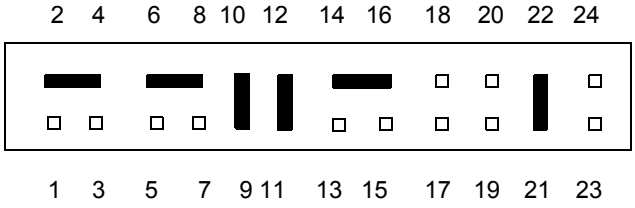


Bus Request Level 0

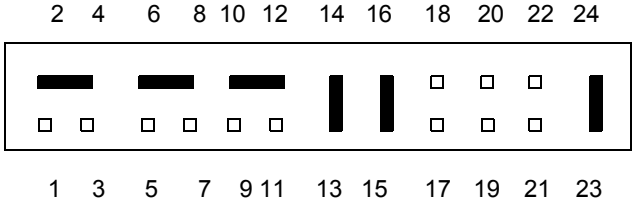


Bus Request Level 1

J7



Bus Request Level 2

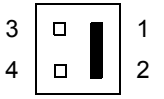


Bus Request Level 3
Factory Configuration

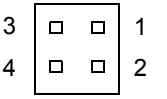
Module Address Mode Select Header (J8)

Header J8 determines whether the module is addressed in standard (A01 through A23) or extended (A01 through A31) mode. If the jumper is installed, the standard addressing is used. If the jumper is removed, extended addressing is used. The as-shipped factory configuration, as shown below, is for extended addressing mode.

J8



Standard Addressing

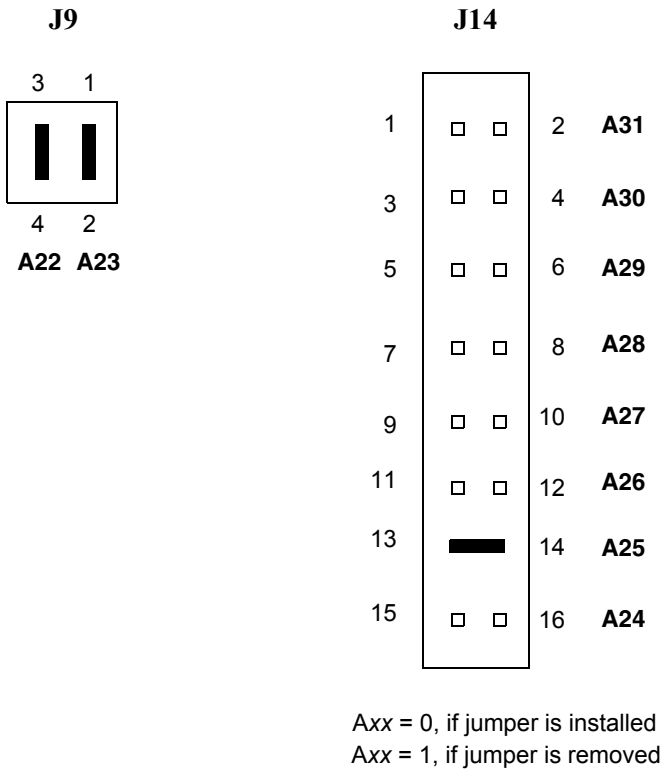


Extended Addressing
Factory Configuration

Module Base Address Select Headers (J9, J14)

Headers J9 and J14 determine the VMEbus base address for the 4 MB onboard RAM (dual ported RAM at local address $FF800000-$FFBFFFFF). Address lines A22 through A31 correspond to these headers and allow base address to be mapped in 4 MB increments anywhere within the 4 GB VMEbus address space.

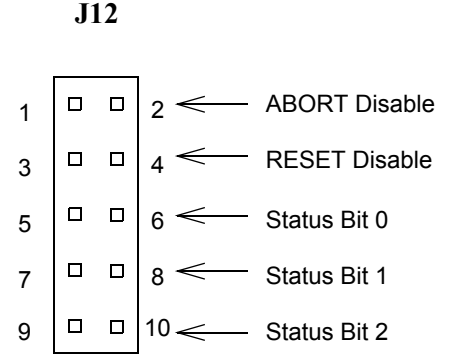
Each jumper corresponds to a specific address line as shown below. Logic high levels on address lines match with removed jumpers and logic low levels match with installed jumpers. The as-shipped factory configuration is with the A22, A23, and A25 address lines set for logic low levels as shown below.



A*xx* = 0, if jumper is installed
A*xx* = 1, if jumper is removed

ABORT/RESET Switches, Status Register Select Header (J12)

Header J12 is a multifunction header for selecting ABORT switch disable, RESET switch disable, and the states of status bits 2-0. As shown below, the as-shipped factory configuration is ABORT switch enabled, RESET switch enabled, and the status bits are set to 1:



Factory Configuration

ABORT and RESET Switches

Pins 1-2 and 3-4 determine whether the RESET and/ or ABORT switch on the front panel are enabled or disabled.

With the jumper installed, the switch is disabled. With the jumper removed, the switch is enabled.

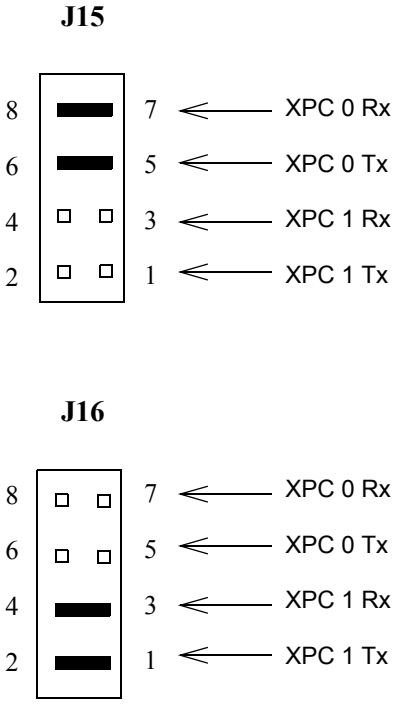
STATUS BITS

Pins 5-6, 7-8, and 9-10 provide three general purpose status bits that can be read by the MPU in the local status register.

With the jumper installed, the status bit = 0. With the jumper removed, the status bit = 1.

XPC Data Clock Select Header (J15)

Header J15 determines whether the transmit and receive clocks for the two X.25 Protocol Controller (XPC) devices are taken from connector P2 or from the onboard programmable interval timer. The following figure and table show the jumper arrangements for the XPC 0 and XPC 1 serial data clock selections. The as-shipped factory configuration is as shown below. Note that J16 is not used on the MVME334B.



XPC 0 and XPC1 Data Clock Selection
Factory Configuration

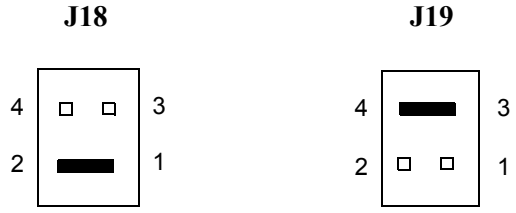
Device	Jumper	Data Clock Selection
XPC 1	1-2 installed	Tx clock from programmable interval timer 2 (DCE configuration - TRXC1 is output)
XPC 1	1-2 removed	Tx clock from connector P2 (DTE configuration - TRXC1 is input)
XPC 1	3-4 installed	Rx clock from programmable interval timer 2 (DCE configuration - RTXC1 is output)
XPC 1	3-4 removed	Rx clock from connector P2 (DTE configuration - RTXC1 is input)
XPC 0	5-6 installed	Tx clock from programmable interval timer 1 (DCE configuration - TRXC0 is output)
XPC 0	5-6 removed	Tx clock from connector P2 (DTE configuration - TRXC0 is input)
XPC 0	7-8 installed	Rx clock from programmable interval timer 1 (DCE configuration - RTXC0 is output)
XPC 0	7-8 removed	Rx clock from connector P2 (DTE configuration - RTXC0 is input)

Note Do not connect external clocks to connector P2 pins which are configured as outputs by related jumper J15 connections.

The J15 jumper settings are also dependent on the transition module connected to P2 (MVME709-1). For further information refer to the transition module user's manual.

Serial Port Configuration Select (J18, J19)

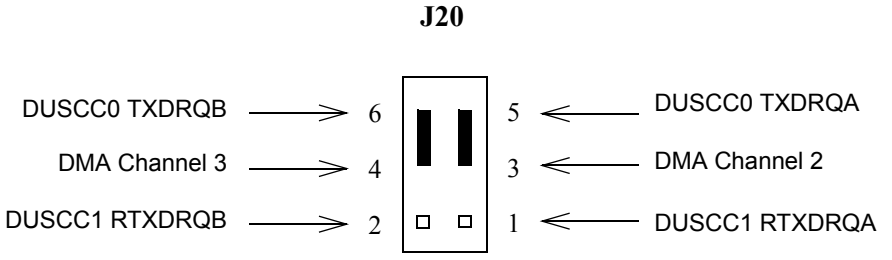
Headers J18 and J19 allow you to select the desired configuration of the serial port. Note that the MVME334B output is always to P2; there is no customer option.



Factory Configuration

DMAC Request Configuration Select Header (J20)

Header J20 is used to configure the connections of Direct Memory Access Controller (DMAC) channels 2 and 3. The DMAC supports either DUSCC0 full duplex or DUSCC0 and DUSCC1 half duplex. The following figure and table show the assignment of signals to pins on header J20. The as-shipped factory configuration is with DUSCC0 full duplex support.



Factory Configuration

Jumper	DMAC Configuration
1-3 installed	DMAC channel 2 connected to DUSCC1 RTXDRQA
3-5 installed	DMAC channel 2 connected to DUSCC0 TXDRQA
2-4 installed	DMAC channel 3 connected to DUSCC1 RTXDRQB
4-6 installed	DMAC channel 3 connected to DUSCC0 TXDRQB

Replacing the OTP ROMs

The MVME334B contains two sockets, labeled XU12 and XU19, that hold two One-Time-Programmable (OTP) ROM devices. These ROMs contain bootstrap and debug firmware. If it becomes necessary to replace these ROMs, follow the procedure below.

Caution

As supplied, the two sockets hold ROM devices that contain bootstrap firmware. If these devices are removed or corrupted, your system will not boot. Replacement ROM devices must perform board initialization and boot capability.

Caution

The ROM components are 44-pin integrated circuit chips; care should be taken not to damage the device pins.

Removing a ROM Device

1. Turn the power off and disconnect the power cable from the power source.
2. Using a PLCC extractor tool appropriate to the 44-pin device, carefully extract the ROM device from the socket.
3. Reconnect the power cable to the power source and turn the power on.

Installing a ROM Device

1. Turn the power off and disconnect the power cable from the power source.
2. Carefully align pin 1 on the ROM device with pin 1 of the socket.

   With a firm action, press the ROM device downward. The Flash device should descend into the socket to be securely installed.
3. Reconnect the power cable to the power source and turn the power on.

Refer to the *ROM and EEPROM Memory* section in Chapter 4 for more details on the ROM memory and compatible ROM devices.

2

Installing the MVME334B

After you have configured the MVME334B's headers as needed for your application, you can install it in the system as follows:

1. Turn all equipment power OFF and disconnect the power cable from the AC power source.

Caution

Connecting modules while power is applied may result in damage to components on the module.

Warning

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

2. Remove the chassis cover as instructed in the equipment user's manual.
3. Remove the filler panel(s) from the appropriate card slot(s) at the front of the chassis. Do not install in card slot 1 unless the module is configured as system controller.
4. Insert the MVME334B into the selected card slot. Be sure the module is seated properly into the connectors on the backplane. Fasten the module in the chassis with the screws provided.
5. Install the appropriate P2 peripheral connector/ cable, which is shipped with the transition module, between the MVME709-1 and/ or MVME709-2 transceiver modules and the connector on the backplane corresponding to the MVME334B P2 connector. Be careful not to connect incompatible signals; e.g., VSB or VMX32) with P2.
6. At the MVME334B slot on the backplane, remove the IACKIN*/ IACKOUT* jumper and the BGIN*/ BGOUT* jumper of the used priority level. Install BGIN*/ BGOUT* jumpers on the unused levels.
7. Carefully replace the cover you previously removed.
8. Turn the equipment power ON.

Using Your MVME334B 3

Introduction

This chapter provides information on the MVME334B memory map and various registers. Included is an aid to software development, outlining points that must be taken into consideration when developing software for the MVME334B.

Controls and Indicators

The MVME334B module has a RESET switch, an ABORT switch, and a FAIL LED indicator, all of which are located on the front panel of the module.

RESET Switch

The RESET switch triggers the reset circuit which generates a board reset signal. The reset circuit causes a full hardware reset of the MPU, the two XPC devices, the two DUSCC devices, the DMA controller, and clears the VMEbus interrupter. The RESET switch can be enabled/ disabled by jumper.

ABORT Switch

An ABORT switch is located on the front panel. The ABORT switch is normally used to abort program execution and return to the debugger.

Whenever the ABORT switch is pressed while running target code, a “snapshot” of the processor state is captured and stored in the target registers. When it is enabled, the ABORT causes a level 7 interrupt to the MC68020. The ABORT switch can be enabled/ disabled by jumper.

FAIL Indicator

The red FAIL LED indicator, located on the front panel, indicates a severe, nonrecoverable malfunction of the MVME334B. The FAIL LED is lit when watchdog time-out occurs. Also, a jumper option asserts SYSFAIL on the VMEbus.

Memory Map

The MVME334B local address area extends from $FF800000 to $FFFFFFFF and is shown in Table 3-1.

The local RAM base address is $FF800000 and extends for 4 MB to $FFBFFFFF. The second byte contains the VMEbus interrupt status/ ID.

The local RAM is shared between the local bus masters and the VMEbus via a dual port controller. The local address area of $FF800000 to $FFBFFFFF may therefore be accessed from the VMEbus at the address area Base Address +($000000-$3FFFFF). Base Address denotes the jumper selectable (J9, J14, and J8) module VMEbus base address, and can be mapped anywhere into the 4 GB VMEbus address space in 4 MB increments.

The local ROM base address is $FFF80000 and extends to 8 MB.

All local devices and registers, including the EEPROM, are resident in the 128 KB area above local ROM, starting at local address $FFFE0000.

Table 3-1. MVME334B Memory Map

Address	Device
$00000000-$FF7FFFFF	VMEbus
$FF800000	Local RAM
$FF800001	Local RAM (VMEbus interrupt vector location)
$FF800000-$FFBFFFEF	Dual ported RAM (general read/ write area) This area is accessible to the host from VMEbus at Base Address + ($000000-$3FFFEF) (See Note 2)
$FFFEE000-$FFFEE0FF	Reserved (access causes BERR*)
$FFBFFFF0-$FFBFFFFF	Dual ported RAM (command/ control area) This area is accessible to the host from VMEbus at Base Address + ($3FFFF0-$3FFFFF) (See Note 2)
$FFF80000-$FFFBFFFF	ROM. Only 256K 32-bit wide words are addressable from the CPU. On the board, A18 is hardwired to 0 and A19 is hardwired to 1.
$FFFE0000	Local status register
$FFFE4000	VMEbus interrupt level register
$FFFE4010	VMEbus interrupt request register
$FFFE8000	Watchdog timer reset
$FFFEC000	XPC 0 reserved
$FFFEC001	XPC 0 command register/ semaphore register
$FFFEC002	XPC 0 reserved
$FFFEC003	XPC 0 interrupt vector register
$FFFEC004-$FFFEC005	XPC 0 data register (high)
$FFFEC006-$FFFEC007	XPC 0 data register (low)
$FFFED000	XPC 1 reserved
$FFFED001	XPC 1 command register/ semaphore register
$FFFED002	XPC 1 reserved

Table 3-1. MVME334B Memory Map (Continued)

Address	Device
$FFFED003	XPC 1 interrupt vector register
$FFFED004-$FFFED005	XPC 1 data register (high)
$FFFED006-$FFFED007	XPC 1 data register (low)
$FFFEF000-$FFFEF0FF	DMAC register table
$FFFF0000	Programmable timer (PIT) counter 0
$FFFF0001	Programmable timer (PIT) counter 1
$FFFF0002	Programmable timer (PIT) counter 2
$FFFF0003	Programmable timer (PIT) control word register
$FFFF1000-$FFFF103F	DUSCC 0 register table
$FFFF2000-$FFFF203F	DUSCC 1 register table DUSCC registers: OMRB bit 6 = 1 ($FFFF202B) and PCRB bit 2 = 0 ($FFFF202E) are reserved for WRITE WRONG PARITY
	Transition board bus
$FFFF4000	Address modifier register
$FFFF8000	Bus time-out clear location
$FFFFC000-$FFFFC7FF	EEPROM

Notes 1. Access to MVME334B local address ranges:

```
$FFC00000-$FFF7FFFF
$FFFC0000-$FFFDFFFF
$FFFEE000-$FFFEEFFF
```

Results in assertion of BERR* to the local MPU.

2. Base Address denotes the jumper selectable (J9, J14) VMEbus board base address, containing address lines A31 - A22.

Register Descriptions

The MVME334B contains a number of registers and memory locations for controlling various onboard functions and for information interchange with the VME host system.

The following sections contain details of all local registers in the order that they occur in the memory map. The descriptions include:

- A short description of the register function.
- The location of the register in the MVME334B address map.
- The type of access (BYTE, WORD, or LWORD; READ or WRITE).
- The size of the register (BYTE, WORD, or LWORD).

VMEbus Command/Control Area

A local reset/ interrupt register and a local command/ status register are mapped into the uppermost 16 bytes of the dual ported local RAM. A write from the VMEbus to even locations with the value \$01*xx* sends an autovector 1 interrupt to the MVME334B, and a write from the VMEbus to even locations with the value \$02*xx* resets the MVME334B. Odd locations can be read or written by either the VMEbus or the local MPU and allow message bytes to be passed, even locations are write only. Any write from the local MPU to the reset/ interrupt register neither resets nor interrupts the local MPU.

Address lines:	A31-A22, A21-A1
Address:	VMEbus base address + \$3FFFF0-3FFFFF
VMEbus access:	WORD: WRITE \$01*xx* to even locations for MVME334B autovector 1 interrupt
	WORD: WRITE \$02*xx* to even locations for MVME334B reset (where *xx* can be used for passing a message byte)

BYTE: READ
(odd locations contain status byte from MVME334B)

MVME334B access: BYTE: WRITE to odd locations
(status byte to VMEbus)

BYTE: READ from odd locations
(command byte from VMEbus)

Size: WORD

Local Address	Reset/Interrupt Register	Command/Status Byte	VMEbus Address
$FFBFFFF0	Not used RES INT	User defined	BA +$3FFFF0
$FFBFFFF2	Not used RES INT	User defined	BA +$3FFFF2
$FFBFFFF4	Not used RES INT	User defined	BA +$3FFFF4
$FFBFFFF6	Not used RES INT	User defined	BA +$3FFFF6
$FFBFFFF8	Not used RES INT	User defined	BA +$3FFFF8
$FFBFFFFA	Not used RES INT	User defined	BA +$3FFFFA
$FFBFFFFC	Not used RES INT	User defined	BA +$3FFFFC
$FFBFFFFE	Not used RES INT	User defined	BA +$3FFFFE
Bits:	15-10 9 8	7-0	

Note BA (Base Address) refers to address lines A31 - A22.

VMEbus Status/ID Location

The programmable VMEbus status/ ID is written into the second location of the local RAM. This byte is sent to the local MPU when an interrupt acknowledge is received at a level equal to the content of the interrupt level register, thus selecting the desired vector from the host vector table for the interrupt service routine.

Address:	$FF800001
Access:	BYTE, WRITE (status/ ID)/ READ
Size:	BYTE

VMEbus Interrupt Level Register

An interrupt level of 1 to 7 must be written into the VMEbus interrupt level register by the local MPU before a VMEbus interrupt is generated.

Address:	$FFFE4000
Access:	BYTE, WRITE only (level 1-7)
Size:	BYTE

VMEbus Interrupt Request Register

When the MVME334B needs to generate a VMEbus interrupt, the local MPU accesses the VMEbus interrupt request register. It must be ensured that the status/ ID location ($FFE00001) and the interrupt level register ($FFFE4000) are set before a VMEbus interrupt is generated.

Address:	$FFFE4010
Access:	BYTE, WORD or LWORD; READ or WRITE
Size:	BYTE

Local Status Register

The local status register is a 6-bit read only register that reflects the status of 3 system signals and 3 jumper positions.

Address:	$FFFE0000
Access:	BYTE or WORD; READ only
Size:	BYTE

Bit	Description	
5	SYSFAIL	This bit reflects the status of the VMEbus line SYSFAIL*.
4	VMEbus Interrupt Acknowledge	When set, this bit indicates that the MVME334B interrupt request to the VMEbus has been acknowledged. When reset, the bit indicates that an MVME334B interrupt request to the VMEbus is pending.
3	VMEbus Time-out (STOUT	When set, this bit indicates that either a VMEbus request has not been granted in time or that the VMEbus access has not been acknowledged by DTACK. This bit is reset if the bus time-out clear location is accessed.
2-0	Readable Jumper Positions	These bits reflect the settings of pins 5-10 on header J12.

Bus Time-Out Clear Location

Any access by the local MPU to the bus time-out clear location resets the bus time-out bit 6 (STOUT) of the local status register. The STOUT bit is set if the bus timer times out during any bus access of the local MPU or the XPCs.

Address: $FFFF8000

Access: BYTE, WORD or LWORD; WRITE or READ

Size: BYTE

Watchdog Timer Reset Location

The watchdog timer must be regularly reset by the MPU within 168 ms periods in order to prevent it from timing out and generating a failure signal. The watchdog timer is reset by simply accessing the watchdog timer reset location with any read or write operation.

Address: $FFFE8000

Access: BYTE or WORD; READ or WRITE

Size: BYTE

Programmable Timer

The 8254 Programmable Interval Timer (PIT) contains three separate 16-bit timers. Timer 0 is used for generating time slice periods of up to 1.342 seconds and timers 1 and 2 are dedicated as baud rate generators for XPC 0 and XPC 1. The following table shows the programmable timer registers.

Table 3-2. Programmable Timer Registers

Address	Register	Size	Access
$FFFF0000	Counter/ status 0 register	BYTE	READ/ WRITE
$FFFF0001	Counter/ status 1 register	BYTE	READ/ WRITE
$FFFF0002	Counter/ status 2 register	BYTE	READ/ WRITE
$FFFF0003	Control word register	BYTE	WRITE

For periodic time slice generation, it is recommended that the square wave operation mode (mode 3) be used for timer 0. In this mode, after loading the appropriate control word and initializing the count register to the desired value, the count down is started by transferring the contents of the count register to the counting element. The counting element is then decremented by two on each

succeeding clock pulse. When the count reaches zero, the counter output line is toggled and the count register contents are reloaded into the counting element and the process repeats indefinitely.

The time slice counter period for mode 3 is given by the formula:

Time slice period = (*preloaded value* + 1) * 20.48 microseconds

Preload value range: minimum = 2, maximum = 0 (corresponding to $10000)

Do not load $0001 into the counter, because in square wave mode it would generate a constant TTL low output level instead of a periodic square wave.

For non-periodic, software synchronized time slice generation, it is recommended that the interrupt on terminal count operation mode (mode 0) be used for timer 0. In this mode, the counter output line goes low upon loading the count byte and the counting element is decremented by 1 on each succeeding clock pulse. Upon reaching zero, the counter output line goes high until the count registers are reloaded.

The time slice counter period for mode 0 is given by the formula:

Time slice period = (*preloaded value* + 1) * 20.48) microseconds

The rising edge of the counter 0 output signal is latched and generates an autovectored level 2 interrupt request to the local MPU. The latched interrupt request is reset when acknowledged by the MPU.

If timers 1 and 2 are to be used as baud rate generators, square wave mode (mode 3) is recommended. The table below shows counter preload values for mode 3 generation of typical baud and bit rates.

Table 3-3. Mode 3 Baud/Bit Rate Generation

Counter Value		Baud Rate (Bits/second)
Hexadecimal	Decimal	
2	2	3072000
4	4	1536000
8	8	0768000
.	.	.
.	.	.
$A0	160	38400
$140	320	19200
$280	640	9600
$500	1280	4800
$A00	2560	2400
$1400	5120	1200
$2800	10240	600
$5000	20480	300

XPC Registers

The X.25 Protocol Controller (XPC) devices contain a large number of registers, only four of which are directly accessible by the MPU. These registers are mapped into the MVME334B memory map as shown below. XPC0 and XPC1 correspond to serial ports 0 and 1, respectively, on the MVME709-1 module.

Detailed information as to the accessibility and programming of the complete XPC register set can be found in the *M68605 X.25 Protocol Controller (XPC) User's Manual*.

Table 3-4. XPC Directly Accessible Registers

Device	Address	Register	Size	Access
XPC 0	$FFFEC001	Command	BYTE	WRITE
XPC 0	$FFFEC001	Semaphore	BYTE	READ
XPC 0	$FFFEC003	Interrupt Vector	BYTE	READ/ WRITE
XPC 0	$FFFEC004	Data (high)	WORD	WRITE
XPC 0	$FFFEC006	Data (low)	WORD	WRITE
XPC 1	$FFFED001	Command	BYTE	WRITE
XPC 1	$FFFED001	Semaphore	BYTE	READ
XPC 1	$FFFED003	Interrupt Vector	BYTE	READ/ WRITE
XPC 1	$FFFED004	Data (high)	WORD	WRITE
XPC 1	$FFFED006	Data (low)	WORD	WRITE

DUSCC Registers

The Dual Universal Serial Communications Controller (DUSCC) devices contain a large number of registers, all of which can be directly addressed. For accesses to specific registers in these devices, the register addresses are taken as offsets to the respective base address. The DUSCC register address map is shown in Table 3-5. The DUSCC Data Book contains detailed information about the use of these registers.

Base address: DUSCC 0 $FFFF1000
DUSCC 1 $FFFF2000

Two locations in DUSCC 1 are reserved to control the function write wrong parity (refer to the *RAM Memory* section in Chapter 4). These locations are bit 2 in OMR(B) and bit 6 in PCR(B). After reset, these bits are cleared and are therefore in the normal operation

mode. Bit 2 of OMR(B) must be set to 1 to activate the write wrong parity function. It must also be ensured that bit 6 of PCR(B) is set to 0.

DUSCC 1 channel A corresponds to serial port 4 on the MVME709-1. DUSCC 0 channel A, DUSCC 0 channel B and DUSCC 1 channel B correspond to serial ports 2, 3, and 5 on the MVME709-2.

Table 3-5. DUSCC Register Memory Map

Address Bits (1) 6 5 4 3 2 1	Abbrev.	Register Name	Mode	Affected by Reset
C 0 0 0 0 0	CMR1	Channel Mode Register 1	R/ W	Yes - 00
C 0 0 0 0 1	CMR2	Channel Mode Register 2	R/ W	Yes - 00
C 0 0 0 1 0	S1R	SYN1/ Sec Address 1 Register	R/ W	No
C 0 0 0 1 1	S2R	SYN2/ Sec Address 2 Register	R/ W	No
C 0 0 1 0 0	TPR	Transmitter Parameter Register	R/ W	Yes - 00
C 0 0 1 0 1	TTR	Transmitter Timing Register	R/ W	No
C 0 0 1 1 0	RPR	Receiver Parameter Register	R/ W	Yes - 00
C 0 0 1 1 1	RTR	Receiver Timing Register	R/ W	No
C 0 1 0 0 0	CTPRH	Counter/ Timer Preset Register High	R/ W	No
C 0 1 0 0 1	CTPRL	Counter/ Timer Preset Register Low	R/ W	No
C 0 1 0 1 0	CTCR	Counter/ Timer Control Register	R/ W	No
C 0 1 0 1 1	OMR	Output and Miscellaneous Register	R/ W	Yes - 00
C 0 1 1 0 0	CTH	Counter/ Timer High	R	No
C 0 1 1 0 1	CTL	Counter/ Timer Low	R	No
C 0 1 1 1 0	PCR	Pin Configuration Register	R/ W	Yes - 00
C 0 1 1 1 1	CCR	Channel Command Register	R/ W	No
C 1 0 0 x x	TxFIFO	Transmitter FIFO	W	No

Table 3-5. DUSCC Register Memory Map (Continued)

Address Bits (1) 6 5 4 3 2 1	Abbrev.	Register Name	Mode	Affected by Reset
C 1 0 1 x x	RxFIFO	Receiver FIFO	R	No
C 1 1 0 0 0	RSR	Receiver Status Register	R/ W(2)	Yes - 00
C 1 1 0 0 1	TRSR	Tx and Rx Status Register	R/ W(2)	Yes - 00
C 1 1 0 1 0	ICTSR	Input+Counter/ Timer Status Register	R/ W(2)	Yes
D 1 1 0 1 1	GSR1	General Status Register	R/ W(2)	Yes - 00
C 1 1 1 0 0	IER	Interrupt Enable Register	R/ W	Yes - 00
C 1 1 1 0 1		Not Used		
0 1 1 1 1 0	IVR	Interrupt Vector Register Unmodified	R/ W	Yes - 0F
1 1 1 1 1 0	IVRM	Interrupt Vector Register Modified	R	Yes - 0F
0 1 1 1 1 1	ICR	Interrupt Control Register	R/ W	Yes - 00
1 1 1 1 1 1		Not Used		

Notes 1. C = 0 for channel A.
C = 1 for channel B.

D = don't care. Register may be accessed as either channel.

x = don't care. FIFOs are addressable at any of four adjacent addresses to allow them to be addressed as BYTE/ WORD/ LWORD.

2. A write to this register may perform a status resetting operation.

DMAC Registers

Each of the four channels in the DMAC has a separate set of registers. Detailed descriptions of these registers can be found in the HD63450 documentation (refer to the *Related Documentation* section in the *Preface* of this manual). The following table lists the DMAC register address offsets from the DMAC base address, and the register sizes.

Address: $FFFEE000 (base address)

Access: READ/ WRITE (except CER which is READ only)

Table 3-6. DMAC Registers

Register	Size	Address Offset				Comment
		CH0	CH1	CH2	CH3	
General Control Register	BYTE	$FF	$FF	$FF	$FF	
Base Function Codes	BYTE	$39	$79	$B9	$F9	
Device Function Codes	BYTE	$31	$71	$B1	$F1	
Channel Priority Register	BYTE	$2D	$6D	$AD	$ED	
Memory Function Codes	BYTE	$29	$69	$A9	$E9	
Error Interrupt Vector	BYTE	$27	$67	$A7	$E7	
Normal Interrupt Vector	BYTE	$25	$65	$A5	$E5	
Base Address Register	LWORD	$1F	$5F	$9F	$DF	lo byte
		$1E	$5E	$9E	$DE	lo-mid byte
		$1D	$5D	$9D	$DD	hi-mid byte
		$1C	$5C	$9C	$DC	hi byte
Base Transfer Counter	WORD	$1B	$5B	$9B	$DB	lo byte
		$1A	$5A	$9A	$DA	hi byte

Table 3-6. DMAC Registers (Continued)

Register	Size	Address Offset				Comment
		CH0	CH1	CH2	CH3	
Device Address Register	LWORD	$17	$57	$97	$D7	lo byte
		$16	$56	$96	$D6	lo-mid byte
		$15	$55	$95	$D5	hi-mid byte
		$14	$54	$94	$D4	hi byte
Memory Address Register	LWORD	$0F	$4F	$8F	$CF	lo byte
		$0E	$4E	$8E	$CE	lo-mid byte
		$0D	$4D	$8D	$CD	hi-mid byte
		$0C	$4C	$8C	$CC	hi byte
Memory Transfer Counter	WORD	$0B	$4B	$8B	$CB	lo byte
		$0A	$4A	$8A	$CA	hi byte
Channel Control Register	BYTE	$07	$47	$87	$C7	
Sequence Control Register	BYTE	$06	$46	$86	$C6	
Operation Control Register	BYTE	$05	$45	$85	$C5	
Device Control Register	BYTE	$04	$44	$84	$C4	
Channel Error Register	BYTE	$01	$41	$81	$C1	read only
Channel Status Register	BYTE	$00	$40	$80	$C0	

Transition Board Bus

The 8-bit portion of the local bus is buffered and fed to connector P2 in order to provide additional control and monitoring lines that may be required for certain expansions; e.g., an additional peripheral chip on the transition board.

Address: $FFFF3000-$FFFF3003

Access: BYTE; WRITE or READ

Size: BYTE

Address Modifier Register

The address modifier register contains the 6-bit code used for VMEbus accesses via the MVME334B VMEbus master interface.

Address: $FFFF4000
Access: BYTE; WRITE only (address modifier code)
Size: BYTE

Bit	Description
7-6	Not used
5-0	VMEbus address modifier code

Extended Addressing Address Modifier Codes:

AM Code	Function
$09	Non-privileged data access
$0A	Non-privileged program access
$0B	Non-privileged block transfer
$0D	Supervisory data access
$0E	Supervisory program access
$0F	Supervisory block transfer

User-Defined Address Modifier Codes:

AM Code	Function
$10-$1F	Non-privileged data access

Short Addressing Address Modifier Codes:

AM Code	Function
$29	Supervisory Access
$2D	Non-privileged access

Standard Addressing Address Modifier Codes:

AM Code	Function
$39	Non-privileged data access
$3A	Non-privileged program access
$3B	Non-privileged block transfer
$3D	Supervisory data access
$3E	Supervisory program access
$3F	Supervisory block transfer

Software Considerations

When developing software for the MVME334B, there are a number of points that must be taken into account. Some of these are hardware dependent and others are dependent upon the driving software or the intended application. The following sections outline these points and give recommendations where necessary. It is intended as a general guide and should be treated as such.

Hardware Initialization

After a board reset has been carried out, the parity bits in RAM are in an undefined state and must be initialized.

When RAM has been initialized, shadow registers should be set up in RAM for any of the write only registers in the MVME334B that will be altered during operation. Following this, the write only registers themselves can be initialized.

The MVME334B contains the following write only registers that cannot be read by the local MPU:

- VME interrupt level register
- Address modifier register

The watchdog and time slice counters are both in an undefined state after a board reset is carried out. The watchdog timer has a period of 168 ms and must be reset within this period (by an access to $FFFE8000) to prevent it from timing out and generating a system fail signal. The time slice counter (counter 0 of the programmable timer/ counter 8254) must be loaded with a start count value if periodic interrupts are required.

If counters 1 and 2 of the programmable timer/ counter 8254 are to be used for clocking the XPC data transfers, these must be initialized to provide the correct clock rate.

The DUSCC devices must be initialized to bring them into a defined state, dependent upon driving software and application.

The MPU must establish the station table, transmit frame specification table, and receive frame specification table for use by each of the XPC devices. The two XPC devices can then be initialized and the addresses of their respective station table passed to them by the MPU.

Note The XPC devices do not come out of software or hardware reset without the transmit clock applied.

Host/MVME334B Dialog Initialization

The MVME334B is intended for use with the following software architecture:

- The host establishes a structure in system memory or in the MVME334B shared memory for the transfer of high level commands and messages to and from the MVME334B.
- The MVME334B fetches commands from this structure, executes the commands, and returns status messages.
- Both polled and interrupt driven modes of operation may be used for the host/ MVME334B software interface. With interrupt driven dialog, the VMEbus interrupt register must be loaded with the interrupt level and status/ ID assigned to the MVME334B. If they are not to be changed during operation, the interrupt level and status/ ID can be programmed into ROM; otherwise, they must be passed to the MVME334B by the host.

The VME command/ control area in the uppermost 16 bytes of the dual ported portion of local RAM is primarily intended for use during system initialization, although it can also be used during normal operation. The host can send bytes of information as well as a local reset or interrupt to the MVME334B and the MVME334B can return status information to the host via the VME command/ control area.

Information passed from host to MVME334B through the VME command/ control area may include:

- Interrupt level and status/ ID assigned to the MVME334B
- Pointers to structures set up in system memory or in the MVME334B shared memory
- Address modifier codes

Information passed from MVME334B to host through the VME command/ control area may include:

- Power up self-test status
- Initialization status
- Malfunction and error codes

DUSCC Programming

The programming of the DUSCC devices is dependent upon driving software and intended application.

In order to ease the start-up procedure, a basic routine for initialization of DUSCC 0 for asynchronous terminal I/ O is shown below. This routine waits for character input on transition board channel 2 or 3 (both configured as DCE) and echoes back any received character.

```
BEGIN
;
SETUP    MOVE.B  #$0F,$FFFF3001   ;set DTR2*, DTR3*, DTR4*, DTR5*
          MOVE.B   #07,CMR1A         ;no parity, async mode
          MOVE.B   #07,CMR1B
          MOVE.B   #$38,CMR2A        ;normal mode, polled/int
          MOVE.B   #$38,CMR2B
          MOVE.B   #$7F,TPRA         ;1 stop bit,RTS, CTS, 8-bit
Tx char
          MOVE.B   #$7F,TPRB
          MOVE.B   #13,RPRA          ;control RTS, no DCD, 8-bit
Rx char
          MOVE.B   #13,RPRB
          MOVE.B   #$2D,RTRA         ;Rx = BRG clk, 9600 baud
          MOVE.B   #$2D,RTRB
          MOVE.B   #1,OMRA           ;set RTS2*
          MOVE.B   #1,OMRB           ;set RTS3*
          MOVE.B   #$A6, PCRA        ;IDC pin 45; RTS Txclk x 1
          MOVE.B   #$A6, PCRB        ;IDC pin 9; RTS Txclk x 1
          MOVE.B   #$3D, TTRA        ;Tx = BRG clock 9600 baud
          MOVE.B   #$3D, TTRB
          MOVE.B   #0,CCRA           ;reset Tx
          MOVE.B   #0,CCRB
          MOVE.B   #$40,CCRA         ;reset Rx
          MOVE.B   #$40,CCRB
          MOVE.B   #02,CCRA          ;enable Tx
          MOVE.B   #02,CCRB
          MOVE.B   #$42,CCRA         ;enable Rx
          MOVE.B   #$42,CCRB
;
;         Read and echo input character
;
DUSCC0RA; BTST.B   #0, GSR           ;check for RxA ready
          BEQ      DUSCC0RB          ;check DUSCC0B
          MOVE.B   RX_FIFOA, TX_FIFOA;echo character
DUSCC0RB; BTSTB.B  #4, GSR           ;check for RxB ready
          BEQ      DUSCC0RA          ;check DUSCC0A
          MOVE.B   RX_FIFOB, TX_FIFOB;echo character
          BRA      DUSCC0RA
;
END SETUP
```

XPC Programming

Initialization of the XPC devices consists of the MPU passing the station table address and function code, system configuration information, and the XPC interrupt vector to each of the XPC devices. Further programming of the XPC devices is dependent upon driving software and intended applications.

VMEbus Interrupter Programming

Execute the following procedure step by step in order to generate a VMEbus interrupt request from the MVME334B to the host system:

1. Check the local status register for SIRQ acknowledged.
2. Store the desired status/ ID in the status/ ID location.
3. Store the desired interrupt level in the interrupt level register.
4. Generate the VMEbus interrupt by reading or writing to the interrupt request register.

```
VMEbus interrupt acknowledged? ($FFFE0000) = (bit 4 (SIRQ) = 1)
LOOPUNTIL SIRQ ACKNOWLEDGED
                .
                .
($FFFE0001) : = Status/ID
                .
                .
($FFFE4000) : = Interrupt level
                .
                .
Generate VMEbus interrupt by reading or writing memory location
$FFFE4010
```

VMEbus Addressing

Data transfers to and from system memory are carried out by the local MPU directly addressing the area from $00000000 to $FF7FFFFF via the VMEbus master interface. This is supported by programmable address modifier codes.

Reset Vector

After a board reset, for the first two memory cycles, onboard ROM is mapped locally into the address space starting at $00000000 and the MPU fetches the reset vector from the bottom addresses of local ROM. Therefore, the firmware resident in local ROM must provide the initial supervisor stack pointer value at $FFF80000-$FFF80003 and program counter value at $FFF80004-$FFF80007.

How the MVME334B Works 4

Introduction

This chapter provides the functional description of the MVME334B at block level. The functional description provides an overview of the module, followed by a detailed description of each section of the module.

Overview

The MVME334B is a VMEmodule for driving serial transceiver devices either as a stand-alone board or as a versatile interface between the VMEbus and the transceivers. It is shown in functional block form in Figure 4-1.

The MVME334B contains a complete microcomputer built around an MC68020 microprocessor which relieves the system host processor from serial communications controlling tasks. Data preprocessing and postprocessing, insertion and deletion of control information, error control, and flow control are typical tasks that the MVME334B processor can carry out.

The local memory consists of 4 MB of dynamic RAM with byte parity generation and checking, 2 KB of EEPROM, and 8 MB of factory- or user-supplied ROM.

The 4 MB of local RAM is 32/ 16 bits wide and is accessible by the MPU, DMAC, and XPC devices. Read accesses are carried out with zero wait states, with one wait state being inserted for write accesses. The local RAM is controlled by a dual port controller and can also be accessed by a host system via the VMEbus. Accesses from the VMEbus are 16 bits wide.

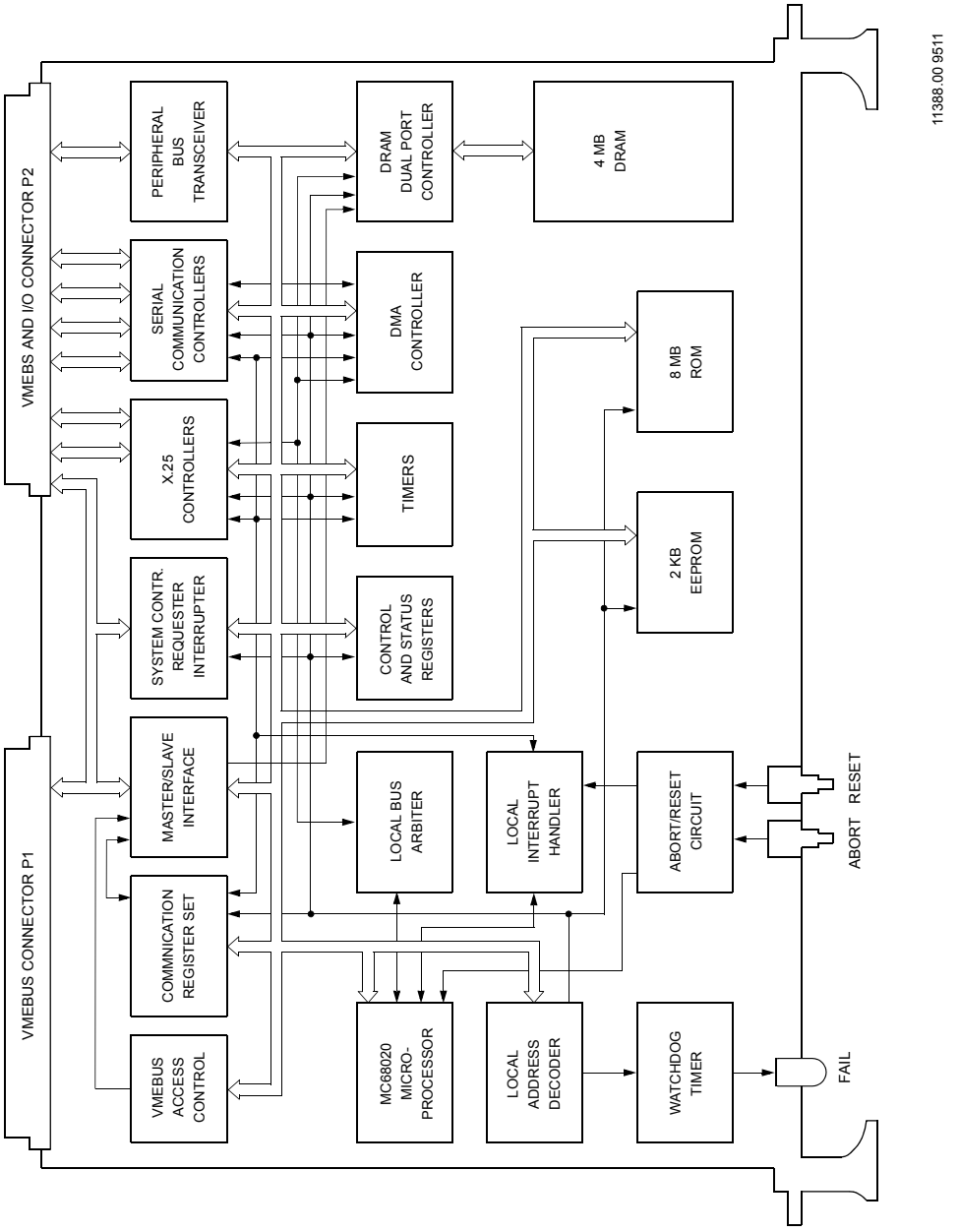


Figure 4-1. MVME334B Block Diagram

The MVME334B is shipped with two 4 MB OTP ROMs installed in EPROM sockets on the base board, supplying 8 MB of read-only memory. These ROMs contain bootstrap firmware and the debug monitor. Refer to Chapter 2 for cautions to observe when replacing these devices. ROM reads are 32 bits wide.

A 2 KB EEPROM is provided for non-volatile storage of parameters that may require modification. Accesses to the EEPROM are byte wide.

Two SCN68562 DUSCC, two MC68605 XPC devices, and a HD63450 DMAC are controlled by the MPU. Each DUSCC is a dual channel, multiprotocol, serial data input/ output device capable of operating asynchronously at up to 38400 baud or synchronously at up to 4 megabits per second. Baud rates, data characteristics, and control functions are software selectable. Each XPC is a serial data input/ output device that fully implements the X.25 recommendation LAPB procedure and can operate safely at up to 10 megabits per second if clocked externally.

The DMAC can be used to control either two DUSCC channels full duplex or four DUSCC channels half duplex.

The local microprocessor accesses VME system memory through an option A32:D16/ 8 or A24:D16/ 8 VMEbus master interface with programmable address modifier codes. The bus arbitration is performed by a Release-When-Done (RWD) bus requester that operates in the "early release of BBSY*" mode and with a selectable priority level. The handshake control logic contains a bus timer which, after a jumper selectable time period, terminates an unsuccessful attempt to transfer data over the VMEbus.

Interrupts to the VME system host processor are generated by a VMEbus interrupter with programmable priority level and status/ ID.

The uppermost 16 bytes of the dual ported local RAM are used for resetting and interrupting the MVME334B and for single-byte transmission/ reception of control/ status bytes between the host and the MVME334B. The VME system host processor can access the dual ported local RAM through an A32:D16/ 8 or A24:D16/ 8 slave

interface whose address modifier decoder is programmed to allow standard and extended data accesses. The dual ported RAM can be mapped anywhere in the 4 GB VMEbus address space in 4 MB increments and used for data interchange with the host.

The MVME334B also contains ten programmable timer/ counters:

- One 16-bit time slice counter that generates periodical interrupts to the MPU. The period is programmable between 40.96 µs and 1.342 seconds for time sliced software concepts.
- Two Tx/ Rx data clock generators for the XPC devices, allowing bit rates of up to 3.072 megabits per second.
- Four 16-bit general purpose timers (two in each DUSCC device) for use as bit rate generators, event counters, etc.
- One watchdog timer with a time-out period of 168 ms that must be periodically reset by the local MPU before this count is reached. If the watchdog timer times out, the FAIL LED on the front panel is turned on and a jumper selectable option also asserts the SYSFAIL* line on the VMEbus.
- One bus timer that monitors all VME and local bus transactions and, if an acknowledge signal is not received within the preset time, generates a BERR* signal and sets a bit in the local status register. The timer period may be jumper selected to be either 57, 114, or 228 µs or infinity.
- One DTACK* generator for inserting two wait states during MPU local accesses to ROM, EEPROM, and onboard registers.

  These timers are described in greater detail in the *Timers* section in this chapter.

  All peripheral signals, at TTL level, are available at connector P2 on the MVME334B. From there they can be fed through a flat ribbon cable to the MVME709-1 or MVME709-2. Three Channel Transition Module. These modules configure the signals to conform to the V.24 or V.35 standards for the serial channels and provide 25-pin sub-D connectors on the front panel for the connection of peripheral devices.

VME System Interface

The following sections describe the VME system operations supported by the MVME334B. This includes the slave interface, command/ control area, master interface, requester and, interrupter.

The VMEbus interface provides the signal path between the MVME334B and the VMEbus backplane. The interface complies with all requirements for the signal driver/ receiver characteristics and bus operation protocols, as specified in the VMEbus specification.

VMEbus Slave Interface

The MVME334B VMEbus slave interface is of the A32:D16/ 8 and A24:D16/ 8 type and allows access from the VMEbus to the local RAM memory. The portion of the local RAM area ($FF800000-$FFBFFFFF) is shared between the local bus masters (MPU and XPC devices) and the VMEbus via a dual port controller and may be accessed from VMEbus addresses Base Address + ($000000-$3FFFFF). The Base Address is jumper selectable (headers J9 and J14) on address lines A22-A31 in 4 MB increments and can be mapped anywhere in the 4 GB VMEbus address space as shown below.

Base Address + ($000000-$3FFFEF)	Dual ported RAM - general read/ write (4 MB minus 16 bytes)
Base Address + ($3FFFF0-$3FFFFF)	Dual ported RAM - command/ control area (16 bytes)

The local RAM area at VMEbus Base Address + ($000000-$3FFFEF) is directly accessible for data interchange between the VMEbus and the MVME334B. The average access time through the slave interface to the shared RAM is approximately 600 ns for read and write accesses.

The 16-byte local RAM area at VMEbus Base Address + ($3FFFF0-$3FFFFF) is employed as a command/ control area and is described in the *VMEbus Command/Control Area* section in Chapter 3.

The addressing capability of the VMEbus slave interface is determined by the configuration of header J8 which allows standard (A24) or extended (A32) addressing. The interface refers to the appropriate address lines for decoding, dependent on the address modifier code.

The slave interface is restricted to data access only. Therefore, the host cannot execute program code in the dual ported RAM. The slave interface data bus is 16 bits wide (D16) and the host MPU must therefore be configured to access the dual ported RAM with either word or byte accesses. The dual ported local RAM is accessed from the VMEbus through the slave interface with address modifier codes $3D and $39 for standard supervisory and standard non-privileged data access; $0D and $09 for extended supervisory and extended non-privileged data access.

VME Command/Control Area

The MVME334B contains a 16-byte area at the upper end of the dual ported portion of local RAM that can be used for the transfer of control and status information between the VME system host processor and the MVME334B.

A write access from the VMEbus to even locations in the command/ control area Base Address + ($3FFFF*x*) with the value $*x*1*xx* results in an interrupt being generated in the MVME334B. A write access from the VMEbus to even locations with the value $*x*2*xx* results in the MVME334B being reset. Accesses from the VMEbus to odd locations in the command/ control area may be read or write, allowing the exchange of command and status information between the host and the MVME334B. It is therefore possible with a single word access with the value $*x*1*xx* to generate an interrupt in the MVME334B and also pass a control byte to the MVME334B.

There is no way to check directly whether the MVME334B has recognized an interrupt placed in the command/ control area. Therefore, to avoid loss of interrupts, you should design your software so that the MVME334B informs the host, through a "ready" message anywhere in the shared RAM, that its interrupt has been processed.

VMEbus Master Interface

The MVME334B contains a VMEbus master interface between the local MPU and the VMEbus. The addressing capabilities of this interface are software programmable via the address modifier decoder register. Short (A16), standard (A24), or extended (A32) addressing is implied by the use of the appropriate address modifier code. The MVME334B VMEbus data width is 16 bit (D16). The full VMEbus address bandwidth from $00000000 to $FF7FFFFF may be accessed, the remaining area from $FF800000 to $FFFFFFFF being occupied by local devices, registers, and RAM.

A minimum VMEbus system controller is implemented on the MVME334B, allowing the module to operate as the sole bus master in a minimum configuration VME system. The functions implemented are power up reset circuitry, SYSRESET driver, SYSFAIL monitoring, and a single level arbiter. These functions are selectable via jumpers and are to be enabled if the MVME334B is assigned to be the system controller.

The master interface is not restricted to data transfer accesses, but it is recommended that program code not be executed from the VMEbus. The code should be downloaded into local memory and executed from there.

The address modifier register is a 6-bit write only register and is used to pass additional address information to the VMEbus. This information can be used in a number of different ways, dependent upon the configuration of the system (this is explained in more detail in the VMEbus specification. Details of the separate bits and location of the address modifier register can be found in the *Address*

Modifier Register section in Chapter 3. It must be ensured that the application software correctly sets the address modifier register prior to any VMEbus operations.

Programming considerations with respect to VMEbus addressing are given in the *VMEbus Addressing* section in Chapter 3.

VMEbus Requester

System memory accesses are controlled in the MVME334B by the VMEbus requester which operates in the "early release of BBSY*" mode. In this mode BBSY* is negated as soon as the MVME334B has asserted its address strobe signal. The VMEbus requester has a jumper selectable (J9) priority level of from 0 to 3. Refer to the *VMEbus Requester Priority Level Select Header* section in Chapter 2 for details.

When the address decoder logic detects that the MPU has asserted an address that is off-board, the VMEbus requester asserts a bus request at the selected priority level. When a bus grant is received at that level, the bus requester asserts the bus busy signal and enables the VMEbus master interface.

If the VMEbus requester detects a bus grant signal for which it has no request pending, the VMEbus requester asserts its bus grant out signal for downstream daisy chain participants.

VMEbus requests are monitored by time-out circuitry that generates BERR* and sets a bit in the MVME334B local status register if bus grant or DTACK* is not received within the selected time period. The time-out period is selectable via header J6 as described in the *Bus Time-out Select Header* section in Chapter 2.

If the VMEbus request from the MPU coincides with an attempt to access MVME334B local RAM by the current bus master, a dual port lockout condition occurs. To alleviate this problem, the bus requester forces a retry to the local MPU upon occurrence of a lockout. If the bus request by the local MPU contains a read-modify-write cycle, the MPU does not accept a retry and the bus requester asserts BERR* to the local MPU.

VMEbus Interrupter

Interrupt requests from the MVME334B to the host system are generated by a VMEbus interrupter with programmable interrupt level and status/ ID. The local MPU must write the level into the interrupt level register ($FFFE4000) and the status/ ID into the status/ ID location ($FF800001) before any interrupt requests are initiated.

When the local MPU accesses the VMEbus interrupt request register at address $FFFE4010 with any read or write instruction, the VMEbus interrupt request signal is asserted at the selected level. When the interrupt acknowledge at that level is received, the VMEbus interrupter issues a status/ ID request to the dual port controller which fetches the VMEbus status/ ID from local RAM and asserts it to VMEbus lines D00-D07. The interrupter logic locks out an MPU access to the interrupt level register until after the pending request is acknowledged.

If the VMEbus interrupter detects an interrupt acknowledge signal for which it has no interrupt pending, the VMEbus interrupter asserts its IACKOUT signal for downstream daisy chain participants.

The local status register contains a bit that reflects the actual state of the interrupt request output. The local MPU can poll this bit to determine whether the interrupt has been acknowledged by the host. After acknowledgment, the bit is automatically cleared by the interrupter.

The *VMEbus Status/ID Location* section contains details of the VMEbus status/ ID location. Details of the VMEbus interrupt request register are given in the *VMEbus Interrupt Request Register* section. The *VMEbus Interrupt Level Register* section contains details of the VMEbus interrupt level register. These three sections are in Chapter 3.

Local Memory

The MVME334B contains 4 MB of dynamic RAM with byte parity, two ROM sockets for 8 MB of ROM, and a 2 KB EEPROM. Accesses to the local RAM are controlled by a dual port controller, allowing direct access by the local MPU or XPC devices or access via the slave interface through the VMEbus.

RAM Memory

The MVME334B RAM memory array consists of one 4 MB device. Local MPU read accesses are carried out with zero wait states and one wait state is inserted for MPU write accesses. Accesses by the XPC devices are carried out with one wait state. The local RAM base address is mapped to $FF800000 in the local memory map.

The RAM is organized 32 bits wide and allows byte, word, or longword accesses by any device, as well as unaligned transfers by the local MPU.

Byte parity generation and checking during RAM accesses are carried out automatically, with parity errors generating an autovectored level 7 interrupt to the MPU. VMEbus and XPC access parity errors assert the BERR* line. A “write wrong parity” function is available for parity logic test purposes. A general purpose output port of DUSCC 2 is used to assert this function (refer to the *DUSCC Registers* section in Chapter 3).

The dynamic RAM is refreshed automatically by refresh logic, this being independent of RAM accesses or any other board activity. A refresh cycle, using CAS before RAS refresh mode and the refresh counters in the RAM devices, is carried out every 10 µs.

ROM and EEPROM Memory

Two 44-pin ROM sockets are provided on the MVME334B for local firmware. The sockets contain two factory-installed 4 MB, 256K x 32 OTP ROMs, which are programmed with bootstrap firmware and

the MVME334ABug debug firmware. Refer to Chapter 2 for cautions that apply when replacing these ROMs with user-supplied ROMs.

The ROM area is mapped into one contiguous block starting at address $FFF80000. This is organized as longword wide memory with only aligned accesses being allowed.

After reset, the ROMs are mapped to starting location 0 during the first two accesses by the MPU to fetch the stack pointer and program counter.

A 2K x 8 bit EEPROM device is provided to enable system parameters to be stored when power is removed from the module. Its base address is mapped to $FFFFC000 in the local memory map. After a write access to this device, which must be bytewise, further accesses to the device are not allowed for 10 ms.

A DTACK generator is provided and allows installation of ROM, PROM, or EPROM devices with maximum 300 ns access times.

Timers

The MVME334B contains a number of timer/ counters for periodic interrupt generation, MVME334B malfunction monitoring, bus access supervision, and serial data clocking.

Programmable Interval Timer

The 8254 Programmable Interval Timer (PIT) on the MVME334B contains three separate programmable 16-bit timers.

Timer 0 is dedicated as a programmable time slice counter for time periods of up to 1.342 seconds. This time slice counter can be used to generate periodic interrupts to the MPU for time sliced software concepts. The counter contains a 16-bit preload register and continuously counts down from the loaded value to 0 at which point it generates an autovectored level 2 interrupt request to the MPU. The interrupt requests are latched and reset when

acknowledged by the MPU. The time slice counter is clocked from the 12.5 MHz MPU clock divided by 256, resulting in a clock frequency of 48.828 KHz. This gives a programmable time slice period of between 40.96 µs (if the counter is loaded with $0002 (refer to note below), and 1.342 seconds (if it is loaded with $0000 (corresponding to $10000), with programmable increments of 20.48 µs.

Caution

Do not load $0001 into the counter because in square wave mode that would generate a constant TTL low output instead of a periodic square wave.

If the preload value is changed while the time slice counter is running, the current period is not affected, and the new value is used with the next period.

Refer to the *Programmable Timer* section in Chapter 3 for time slice period calculation.

Timers 1 and 2 are dedicated as Tx/ Rx data clock generators for XPC devices 0 and 1, respectively. Both timers are clocked with 6.144 MHz allowing a maximum bit rate of 3.072 megabits per second.

Multifunction Counter/Timer

Each of the two DUSCC devices contains two 16-bit counter/ timers, one for each serial channel. These counter/ timers can be programmed to function as bit rate generators, event counters, interval timers, and may also be used for bit length measurements or to count transmitted or received characters.

The SCN68562 DUSCC data book contains detailed information about the programming and use of these counter/ timers.

Watchdog Timer

The watchdog timer is a free running counter that generates a system fail signal upon timing out to indicate a severe, nonrecoverable malfunction of the MVME334B. Upon timing out, it lights the FAIL LED on the front panel and a jumper option (J5) for the watchdog timer also asserts the SYSFAIL* line on the VMEbus.

The watchdog timer is a 14 stage counter which is clocked at a frequency of 48.828 KHz, resulting in a full count time after reset of 168 milliseconds. In order to prevent the counter timing out and generating the above signals, it must be regularly reset by the MVME334B software before it times out. This is achieved by simply addressing the watchdog timer reset location ($FFFE8000). No data transfer is necessary.

Bus Timer

The bus timer monitors all bus requests and data transfer cycles and if the addressed device does not respond with a data transfer acknowledge signal within the preset time, it generates a BERR* signal and sets a bit in the local status register. A time-out period of either 57, 114, or 228 μs, or infinity may be selected by header J6.

The time-out period must be set to a value greater than the longest time period that may elapse between the assertion of a bus request and the reception of a bus grant in the actual system configuration.

DTACK Generator

The DTACK generator is a wait state generator for MPU accesses to ROM and all onboard registers. Two wait states are inserted in order to accommodate ROM/ EEPROM devices with 300 ns access times.

Local Status Register

The MVME334B contains a local status register (at local address $FFFE0000) that enables the local MPU to monitor various MVME334B functions. The local status register is a 6-bit read only register with the following functions:

Bit	Description
5	Reflects the status of the VMEbus SYSFAIL* line.
4	VMEbus interrupt acknowledge bit (SIRQ) and indicates that the interrupt request to the VMEbus has been acknowledged.
3	Time-out status bit (STOUT) and is set if an addressed device does nor respond before the time-out specified by header J6.
2-0	Connected directly to header J12 and reflect whether the corresponding jumper is installed or removed. They are intended for general purpose use and could, for example, be used to encode specific tart up conditions.

Local Bus Arbiter

The local MPU is the permanent master of the local bus until one of the potential bus masters issues a bus request. When either an XPC device, the DMA controller, or the dual port controller requests local bus mastership, the local bus arbiter issues a bus request to the local MPU. When a bus grant is received from the local MPU, the local bus arbiter grants bus mastership to the requesting device. If more than one potential master requests bus mastership, the local bus arbiter arbitrates according to the priority scheme shown below.

Priority	Bus Master
Highest	XPC 0, XPC 1, DMAC
Lowest	VMEbus (dual port controller)

Bus requests are serviced immediately after the local MPU finishes its current bus cycle.

The bus arbiter operates in a “fair” mode; i.e., it does not allow the higher priority devices to monopolize the local bus. If lower priority requests are also pending, the arbiter queues these requests and grants them between successive higher priority requests.

Local Interrupts

The local interrupt handler controls all interrupts within the MVME334B. In addition to interrupt requests caused by the host writing $01*xx* to the VMEbus command/ control area (Base Address + $3FFFF0), it handles interrupt requests generated by the DUSCC devices, the programmable time slice counter, the XPC devices, the RAM parity logic, or the software ABORT switch. Each of these interrupt request sources is allocated a unique interrupt level and status/ ID.

After receiving an interrupt request, the local interrupt handler places a code on the three interrupt request input lines to the MPU, which is dependent upon the source of the request. The MPU then acknowledges the interrupt and executes the appropriate service routine.

The XPC and DUSCC devices supply programmable status/ IDs, whereas all other sources initiate autovectored interrupt processing.

The different interrupt request sources together with their associated priority levels and status/ IDs used are listed in Table 4-1.

When the local RAM parity logic detects a RAM read error, an autovectored level 7 interrupt is generated in addition to the data transfer acknowledge.

The interrupt level 7 is used for the software ABORT switch and RAM parity error signals because these have the highest priority in the MVME334B and the level 7 interrupt is non-maskable.

The interrupt request lines of the two DUSCC devices are connected in a wired-OR configuration with the level 6 interrupt line. The interrupt enable input/ output pins of the devices are daisy chained, with DUSCC 0 having the highest, and DUSCC 1 the lowest priority. Each DUSCC has eight sources of interrupts which are themselves prioritized in a daisy chain. The order of priority is: channel A receiver, transmitter, status, and external or C/ T status; channel B receiver, transmitter, status, and external or C/ T status. Each DUSCC device can be programmed with a "base" status/ ID which is modified internally by the DUSCC to provide a unique status/ ID corresponding to the specific interrupt source.

Each of the two XPC devices generates a unique interrupt level, XPC 0 is interrupt level 4 and XPC 1 is interrupt level 3. Each XPC device also has four possible internal interrupt sources: transmitter and receiver, receiver, transmitter, and bus error or address error while accessing local memory. Each XPC device can be programmed with a "base" status/ ID which is modified internally by the XPC to provide a unique status/ ID corresponding to the specific interrupt source.

The time slice counter can be programmed to generate periodic, autovectored level 2 interrupts.

An autovectored level 1 interrupt is generated when the host writes the value $01*xx* into the command/ control area at VMEbus Base Address + ($3FFFFF).

Table 4-1. Local Interrupt Levels and Vectors

Interrupt Source	IRQ Level	Chain Priority	Exception Vector
RAM parity error	7	-	IRQ7 autovector
Software ABORT switch	7	-	IRQ7 autovector

Table 4-1. Local Interrupt Levels and Vectors (Continued)

Interrupt Source		IRQ Level	Chain Priority	Exception Vector
DUSCC 0	Channel 2 Rx ready	6	16	Programmable base vector
	Channel 2 Tx ready	6	15	Programmable base vector
	Channel 2 ext, C/ T status	6	14	Programmable base vector
	Channel 3 Rx/ Tx status	6	13	Programmable base vector
	Channel 3 Rx ready	6	12	Programmable base vector
	Channel 3 Tx ready	6	11	Programmable base vector
	Channel 3 Rx/ Tx status	6	10	Programmable base vector
	Channel 3 ext, C/ T status	6	9	Programmable base vector
DUSCC 1	Channel 4 Rx ready	6	8	Programmable base vector
	Channel 4 Tx ready	6	7	Programmable base vector
	Channel 4 Rx/ Tx status	6	6	Programmable base vector
	Channel 4 ext/ T status	6	5	Programmable base vector
	Channel 5 Rx ready	6	4	Programmable Base vector
	Channel 5 Tx ready	6	3	Programmable Base vector
	Channel 5 Rx/ Tx status	6	2	Programmable Base vector
	Channel 5 ext, C/ T status	6	1	Programmable Base vector

Table 4-1. Local Interrupt Levels and Vectors (Continued)

Interrupt Source		IRQ Level	Chain Priority	Exception Vector
DMAC Interrupt		5	-	
XPC 0	Channel 0 BERR, AERR	4	-	Programmable Base vector
	Channel 0 Rx	4	-	Programmable Base vector
	Channel 0 Tx	4	-	Programmable Base vector
	Channel 0 Tx and Rx	4	-	Programmable Base vector
XPC 1	Channel 1 BERR, AERR	3	-	Programmable Base vector
	Channel 1 Rx	3	-	Programmable Base vector
	Channel 1 Tx	3	-	Programmable Base vector
	Channel 1 Tx and Rx	3	-	Programmable Base vector
Time slice counter		2	-	IRQ2 autovector
VMEbus Interrupt		1	-	IRQ1 autovector

Bus Errors

Bus errors are generated in each of the following cases: when the bus timer times out, when an attempt is made to access an undefined address, when a VMEbus request retry is not accepted by the MPU, when a BERR* signal is received from the VMEbus, or when a local bus requester does not acknowledge its bus grant.

Power-Up and Reset

The MVME334B contains a reset circuit which generates a board reset signal when power is applied to the module. The reset circuit is also triggered by the RESET switch on the front panel, and by the VMEbus writing the value $02*xx* into VMEbus Base Address + ($3FFFF0) (Base Address denoting the MVME334B VMEbus board base address).

The reset circuit asserts the board reset signal for approximately 300 milliseconds. This causes a full hardware reset of the MPU, the two XPC, and the two DUSCC devices and clears the VMEbus interrupt. After the reset period, the MPU fetches the initial supervisor stack pointer and program counter from the local ROM which is initially mapped to location 0 for these first two accesses.

The flip-flop indicating bus time-out is in an undefined state after power-up. Therefore, in its self-initialization routine, the MVME334B should access the bus time-out clear location.

SCN68562 DUSCC Devices

The MVME334B contains two SCN68562 Dual Universal Serial Communications Controller (DUSCC) devices for serial data transfer. Each device provides two independent full duplex serial channels and is capable of transferring data at up to 4 megabits per second.

The DUSCC devices are clocked at 14.7456 MHz and 16 common bit rates from 50 to 38400 baud are available from the internal bit rate generator separately.

Each serial channel consists of a transmitter, a receiver, a 16-bit multifunction counter/ timer, a digital phase locked loop (DPLL), a parity/ CRC generator and checker, and associated control circuits. The two channels in each DUSCC device share a common bit rate generator.

Each serial channel can be independently programmed to provide either synchronous or asynchronous serial communications with programmable parameters and encoding.

Asynchronous Operation

With asynchronous data transfer, 16 fixed baud rates from 50 to 38400 are supported, or user-defined bit rates may be derived from the internal programmable counter/ timer devices. Character lengths of 5 to 8 bits with odd or even parity, no parity, or force parity and up to 2 stop bits in 1/ 16 bit increments can be programmed. Data encoding/ decoding may be NRZ, NRZI, FM0, FM1, or Manchester.

Synchronous Operation

Both byte and bit oriented synchronous data transfer are supported, and a number of different synchronization modes can be programmed.

Byte oriented synchronous data transfer can be used in BISYNC, DDCMP, or X.21 modes, with or without LRC or CRC generation and checking.

Bit oriented synchronous data transfer can be used with HDLC/ ADCCP, SDLC, SDLC loop, X.25, or X.75 link level protocols, with CRC generation and checking. Occurrence of an underrun generates an interrupt to the MPU.

For both byte and bit oriented synchronous data transfer, character lengths may be from 5 to 8 bits, parity may be odd, even, none, or forced and data coding may be NRX, NRZI, FM0, FM1, or Manchester.

HD63450 DMAC

The HD63450 4-channel Direct Memory Access Controller (DMAC) allows fast data transfers between any of the 4 DUSCC channels and local RAM as well as block transfers within local RAM. The following sections describe the various DMA configuration options for the MVME334B.

For detailed programming instructions refer to the SCN68562 DUSCC and HD63450 DMAC documents listed in the *Related Documentation* section in the preface to this manual.

Data Transfers Between DUSCC and Memory

The DMA controller supports these communication channels:

2 DUSCC 0, CH A

3 DUSCC 0, CH B)

4 DUSCC 1, CH A

5 DUSCC 1, CH B

The four DMAC channels can be configured by positioning jumpers on header J20 to control either:

DUSCC channels 2 and 3 full duplex

Or:

DUSCC channel 2 full duplex and
DUSCC channels 3 and 5 half duplex

Or:

DUSCC channels 2 and 4 half duplex and
DUSCC channel 3 full duplex

Or:

DUSCC channels 2, 3, 4, and 5 half duplex

The following table shows channel assignment and header J20 jumper positions for the DUSCC/ DMAC configurations.

Table 4-2. DUSCC/DMAC Configurations

Configuration	DUSCC/DMAC Channel Connection			J10	DUSCC Pin Programming
Channel 2 full duplex	DUSCC 0	RTXDRQA	DMA CH 0		For receive DMA request
		TXDRQA	DMA CH 2	3 - 5	
Channel 3 full duplex	DUSCC 0	RTXDRQB	DMA CH 1		For receive DMA request
		TXDRQB	DMA CH 3	4 - 6	
Channel 2 half duplex and	DUSCC 0	RTXDRQA	DMA CH 0		For receive or transmit DMA request
Channel 4 half duplex	DUSCC 1	RTXDRQA	DMA CH 2	1 - 3	For receive or transmit DMA request
Channel 3 half duplex and	DUSCC 0	RTXDRQB	DMA CH 1		For receive or transmit DMA request
Channel 5 half duplex	DUSCC 1	RTXDRQB	DMA CH 3	2 - 4	For receive or transmit DMA request

For data transfers between DUSCC devices and memory, the DMAC operates in the explicit dual address mode, the device port (DUSCC) size is byte, and the operand size is byte.

Program the DMAC as follows:

Device Control Register

XRM = Burst mode

DTYP = MC68000-compatible device, explicit address

DPS = 8-bit port

Operation Control Register:

SIZE = Byte

CHAIN = Chain operation enabled or disabled

REQG = REQ* line initiates an operand transfer

Sequence Control Register

MAC = Counts

DAC = Does not count

Function Code Registers

DFCR = Any value other than 7

MFCR = Any value other than 7

MC68605 XPC Devices

Two MC68605 X.25 Protocol Controller (XPC) devices are installed on the MVME334B and provide the capability of high speed serial communications using the X.25 LAPB protocol. Each device provides one full duplex channel and when configured as DCE, is clocked from a dedicated onboard bit-rate generator providing data transfer rates of up to 3.072 megabits per second. When the channel is configured as DTE, it can be clocked from an external oscillator with a maximum frequency of 10 MHz.

The XPC fully implements the CCITT X.25 recommendation LAPB data link access procedure and relieves the MPU from having to manage the communications link by providing sequencing using HDLC framing, error control, retransmission using CRC, and flow control.

Primary communication between the local MPU and the XPC is via three tables in shared memory structures. These are the station table, the transmit frame specification table, and the receive frame specification table.

- The station table contains operating information to the XPC, the pointers for the transmit and receive frame tables and status or error information from the XPC.
- The transmit frame specification table contains the queued specifications of the frames to be transmitted.
- The receive frame specification table contains the specifications of the buffers for frames to be received.

These structures are set up by the MPU during initialization and the pointer to the station table is passed to the XPC. Because the XPC is a full local bus master, data transfer between local memory and the XPC can be carried out using the shared memory structures, by the onchip DMA controller, and the two 22-byte FIFO buffers integrated into the XPC device. DMA transfer to the VMEbus is, however, not possible.

Peripheral Port Signals

The input/ output signals of all six serial ports are routed to rows A and C of connector P2. All signals are TTL compatible. The outputs are capable of sinking 2.0 mA at 4.0 V and sourcing 0.25 mA at 2.4 V. The inputs are high impedance with a leakage current of 10 µA maximum.

The peripheral port signals fall into one of the following categories:

Data signals	Contain the information that is to be interchanged between DTE and DCE.
Control signals	Used by DTE and DCE to ensure that no data is sent until both pieces of equipment are ready for the data interchange.
Timing signals	Used by DTE and DCE for synchronizing the data interchange. They are only required if the data communication is carried out in synchronous mode.
Transition board bus signals	Additional signals provided by the MVME334B for the control of devices connected to the peripheral port. The signals provided are: ❑ The 8-bit port portion of the local bus (P2: PD0-PD7). ❑ Two address lines A00, A01 (P2: PA0* and PA1*). ❑ A select line (P2: PS*). ❑ A reset line (P2: RESET*). ❑ A read/ write line (P2: RD*).

These signals are buffered and the data transfer handshake is controlled by the DTACK generator. The minimum duration for the select signal is 160 ns.

The MVME334B signal nomenclature is that of a DTE device, and the signal descriptions include standard functions and connections with DUSCC devices, XPC devices, and registers on the MVME334B.

MVME709-1/-2 Three Channel Transition Modules

The MVME709-1 supports three serial channels as serial ports 0, 1 and 4, and the MVME709-2 supports three serial channels as serial ports 2, 3, and 5.

The MVME709-1 and MVME709-2 Three Channel Transition Modules provide the transmitter and receiver drivers for converting the TTL level peripheral input/ output signals of the MVME334B module to the V.24 or the V.35 standard for the serial channels.

Each of the serial channels on the MVME709-1 and the MVME709-2 can be configured separately for either the V.24 or the V.35 standard. This is achieved by inserting either the V.24 or the V.35 transmitter and receiver devices into the onboard I.C. sockets and changing the positions of jumpers.

The transition module is connected to the MVME334B via the bypacked DIN 41612 C64 connector and a 64-conductor flat ribbon cable. Three standard, 25-pin sub-D connectors are mounted on the front panel of the transition board for the attachment of serial peripherals.

Furthermore, each serial port connector can be configured independently as DCE for connecting terminals, printers, etc., or as DTE for connecting modems, computers, etc.

For a more detailed description of the transition modules, refer to the *MVME709-1/-2 Three Channel Transition Module User's Manual*.

Index

E

F

G

H

I

J

INDEX

L

M

N

O

P

INDEX

R

S

T