Systran BHAS-VMESC4M
VME6U Slave 4-Slot IP Module Carrier

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Limited Availability
Used and in Excellent Condition

Open Web Page

https://www.artisantg.com/44616-1

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

- Critical and expedited services
- In stock / Ready-to-ship
- We buy your excess, underutilized, and idle equipment
- Full-service, independent repair center

VMESC4M

VME6U Slave 4-Slot IP Module Carrier Hardware Reference

Document No. B-T-MR-VMESC4M#-A-0-A2

CURTISS WRIGHT Controls, Inc.
Embedded Computing

FOREWORD

The information in this document has been carefully checked and is believed to be accurate; however, no responsibility is assumed for inaccuracies. Curtiss-Wright Controls, Inc. reserves the right to make changes without notice.

Curtiss-Wright Controls, Inc. makes no warranty of any kind with regard to this printed material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

Systran® is a registered trademark of Curtiss-Wright Controls, Inc. Systran Division.

FibreXpress® is a registered trademark of Curtiss-Wright Controls, Inc..

FibreXtreme® is a registered trademark of Curtiss-Wright Controls, Inc.

LinkXchange® is a registered trademark of Curtiss-Wright Controls, Inc.

SCRAMNet® is a registered trademark of Curtiss-Wright Controls, Inc.

HP™ is a trademark of the Hewlett-Packard Company
ALTERA® and Altera® are registered trademarks of Altera Corp.
Littelfuse® registered trademark of Littelfuse, Inc
Motorola® is a registered trademark of the Motorola Corp.
OS-9® is a registered trademark of Microware Systems Corp.
TRI-STATE® is a registered trademark of National Semiconductor Corp.
VIEWlogic® is a registered trademark of Viewlogic Systems, Inc.
Digi-Key® is a registered trademark of Digi-Key Corp.

Any reference made within this document to equipment from other vendors does not constitute an endorsement of their product(s).

Revised: November 15, 2004

Curtiss-Wright Controls, Embedded Computing
Data Communications
4126 Linden Avenue
Dayton, OH 45432-3068
(937)252-5601

FCC

This product is intended for use in industrial, laboratory or military environments. This product uses and emits electromagnetic radiation, which may interfere with other radio and communication devices. The user may be in violation of FCC regulations if this device is used in other than the intended market environments.

CE

Please note: As a component part of another system, this product has no intrinsic function and is therefore not subject to the European Union CE EMC directive 89/336/EEC.

TABLE OF CONTENTS

APPENDICES

FIGURES

TABLES

1. INTRODUCTION

1.1 Purpose

This is a reference manual for the Curtiss-Wright Controls VME6U Slave 4-Slot IP Module Carrier herein referred to as the VMESC4M board, part number BHAS-VMESC4M.

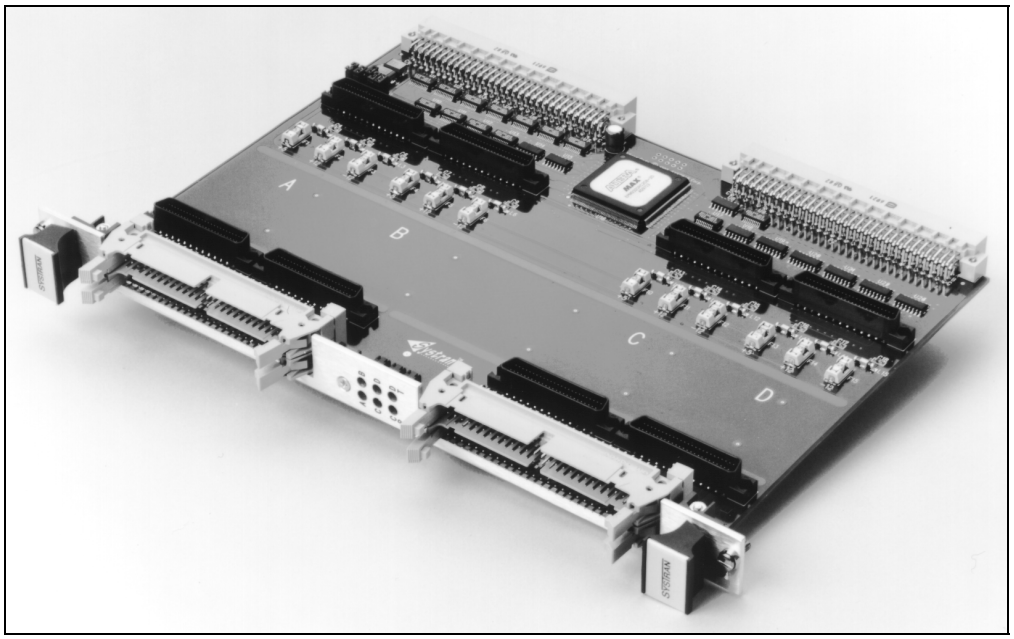


Figure 1-1 VMESC4M Board

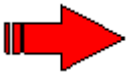
1.2 Scope

This reference manual covers the physical and operational description of the VMESC4M, both from hardware and software perspectives. This reference manual also contains detailed technical information about the VMESC4M's performance characteristics, and a few typical applications. It is assumed that the reader has general understanding of computer processing, software and/or hardware applications experience, and a working knowledge of using IP Modules. Citation of equipment from other vendors within this document does not constitute an endorsement of their product(s).

1.3 Overview

The VMESC4M is a VME6U Slave board that supports four singlewide IP Modules, two singlewide and one doublewide, or two doublewide IP Modules. This carrier is limited to the support of 8 MHz IP Modules. The VMESC4M supports I/O, ID, Interrupt and Memory transfers on all four slots. DMA accesses are not supported. The following sections in this manual describe the board features, overall board layout, data transfer types, and detailed description of the registers. The VMESC4M's VMEbus interface is designed in compliance with the *ANSI/VITA 1–1994*, American National Standard for

VME64, and the IP Module's interface are designed in accordance with *ANSI/VITA 4–1995*, American National Standard for IP Modules.



NOTE: Physical doublewide IP Modules are supported. However, logical doublewide 32-bit transfers are not supported.

The VMESC4M was designed around a very easy to use, intuitive user interface, employing the highest quality (yet, reasonably affordable) components to provide the best performing, highest density VME6U IP Module slave carrier available. The VMESC4M is a super-synchronous design that does not insert IP Module hold cycles. This results in a very high throughput for this VME-to-IP Module bus coupling carrier.

The VMESC4M provides up to 200 I/O points (50 per IP Module) per VME6U slot with the entire area under the IP Modules as a solid ground plane to provide "quiet" operating conditions for sensitive analog IP Modules. The front panel LEDs give the user diagnostic information on accesses to the IP Modules, the VMESC4M local status and control registers, and the VME DTACK. The VMESC4M has a very easy to use and flexible VME interrupting capability where the individual interrupts from each IP Module can be programmed to assert any VME interrupt request signal.

1.3.1 Features

- Up to 200 I/O points in a single, VME6U slot.
- Each of 8 IP Module interrupt requests (2 per IP Module slot) can individually assert any of the 7 VME interrupt levels; equal levels IP Module slot prioritized.
- I/O, ID, INT and memory transfers with no HOLDs. *
- Read-Modify-Write cycles supported to IP Modules and VMESC4M registers.
- Independent field-replaceable fuses, and 'T'-type filters are used for the +5, +12, and -12 V power feeding each IP Module.
- IP Module error signals posted as status.
- IP Modules are individually software resettable.
- Board base address jumper selectable.
- Pure ground plane under IP Modules for "quiet" operation.
- Supports four singlewide, or two singlewide and one doublewide, or two doublewide IP Modules.
- Six front-panel IP Module/Carrier Access and VME DTACK indicators (with pulse stretchers).
- Four 8-bit General Purpose registers.
- Supports 8- and 16-bit transfers to I/O, ID and INT in A16 space (2 KB block using address modifiers 0x29, 0x2D). Will also support memory transfers in A24 space (1 MB for each IP Module using address modifiers 0x39, 0x3D) or A32 space (8 MB for each IP Module using address modifiers 0x09, 0x0D).
- Supports writes to ID space.

* *ANSI/VITA 4–1995* used.

1.4 Related Publications

- Curtiss-Wright Controls, Inc. I/O Products Technical Note #2001 entitled *Programmed Transfer Rate Analysis of the IP Bus Onboard the Motorola MVME162 Controller* (Doc. A-T-ST-IPAC2001)
- *ANSI/VITA 1–1994* published by VMEbus International Trade Association, P.O. Box 19658, Fountain Hills, AZ. 85269 USA. www.vita.com.
- *ANSI/VITA 4–1995* published by VMEbus International Trade Association, P.O. Box 19658, Fountain Hills, AZ. 85269 USA. www.vita.com.

1.5 Quality Assurance

Curtiss-Wright Controls' policy is to provide our customers with the highest quality products and services. In addition to the physical product, the company provides documentation, sales and marketing support, hardware and software technical support, and timely product delivery. Our quality commitment begins with product concept, and continues after receipt of the purchased product.

Curtiss-Wright Controls' Quality System conforms to the ISO 9001 international standard for quality systems. ISO 9001 is the model for quality assurance in design, development, production, installation and servicing. The ISO 9001 standard addresses all 20 clauses of the ISO quality system, and is the most comprehensive of the conformance standards.

Our Quality System addresses the following basic objectives:

- Achieve, maintain and continually improve the quality of our products through established design, test, and production procedures.
- Improve the quality of our operations to meet the needs of our customers, suppliers, and other stakeholders.
- Provide our employees with the tools and overall work environment to fulfill, maintain, and improve product and service quality.
- Ensure our customer and other stakeholders that only the highest quality product or service will be delivered.

The British Standards Institution (BSI), the world's largest and most respected standardization authority, assessed Curtiss-Wright Controls' Quality System. BSI's Quality Assurance division certified we meet or exceed all applicable international standards, and issued Certificate of Registration, number FM 31468, on May 16, 1995. The scope of Curtiss-Wright Controls' registration is: "Design, manufacture and service of high technology hardware and software computer communications products." The registration is maintained under BSI QA's bi-annual quality audit program.

Customer feedback is integral to our quality and reliability program. We encourage customers to contact us with questions, suggestions, or comments regarding any of our products or services. We guarantee professional and quick responses to your questions, comments, or problems.

1.6 Technical Support

Technical documentation is provided with all of our products. This documentation describes the technology, its performance characteristics, and includes some typical applications. It also includes comprehensive support information, designed to answer any technical questions that might arise concerning the use of this product. We also publish and distribute technical briefs and application notes that cover a wide assortment of topics. Although we try to tailor the applications to real scenarios, not all possible circumstances are covered.

Although we have attempted to make this document comprehensive, you may have specific problems or issues this document does not satisfactorily cover. Our goal is to offer a combination of products and services that provide complete, easy-to-use solutions for your application.

If you have any technical or non-technical questions or comments, contact us. Hours of operation are from 8:00 a.m. to 5:00 p.m. Eastern Standard/Daylight Time.

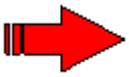
- Phone: (937) 252-5601 or (800) 252-5601
- E-mail: support@systran.com
- Fax: (937) 252-1465
- World Wide Web address: www.cwcembedded.com

1.7 Ordering Process

If you wish to learn more about Curtiss-Wright Controls' products or to place an order, the following contacts are available:

- Phone: **(937) 252-5601**
- E-mail address: **info@systran.com**
- World Wide Web address: **http://www.cwcembedded.com**

2. PRODUCT OVERVIEW



NOTE: The extended implementation of write-transfer capability over the full ID addressing space for this product is over and above that described in the specification.

2.1 Overview

This section describes the VMESC4M in detail. It begins by discussing the VMESC4M in terms of its functionality and features. Next, it covers the VMESC4M theory of operation. Finally, it discusses the VMESC4M schematics.

2.2 Functional Description

2.2.1 VME Memory Mapping Scheme

The VMESC4M responds to VME supervisory and non-privileged data accesses in all three memory sizes: A16, A24 and A32. The memory mapping scheme of the VMESC4M allows I/O and ID addresses for all four IP Modules as well as the carrier's control/status registers to be accessed in A16 mode at fixed addresses relative to a jumper-selectable board base address. These base addresses can be set on 2 K boundaries throughout the VME short address space.

The VMESC4M responds to address modifier codes 0x29 and 0x2D (non-privileged and supervisory, respectively) in an A16 transfer. Memory accesses to all four of the IP Modules can be performed in either standard (A24) address mode or in extended (A32) address mode. The VMESC4M responds to address-modifier codes 0x39 (standard non-privileged) and 0x3D (standard supervisory) in an A24 transfer. If this addressing mode is used, each of the four IP Modules is assigned a 1 MB section of the A24 memory map within an area defined by address bits A22 and A23. These bits are compared to those programmed into the MEMA24BASE Register to determine whether the VMESC4M memory space is being hit in a standard data transfer.

The VMESC4M also responds to address modifier codes 0x09 (extended non-privileged) and 0x0D (extended supervisory) in an A32 transfer. If this addressing mode is used, each of the four IP Modules is assigned an 8 MB section of the A32 memory map within an area defined by address bits 31 down to 25. These bits are compared to those programmed into the MEMA32BASE Register to determine whether the VMESC4M memory space is being hit in an extended data transfer. The MEMA24BASE Register and the MEMA32BASE Register are outlined in table 4-1 in section 4-4.

2.2.2 ID Transfers

IP Module ID transfers are supported as A16/D16/D8 VME transfers. ID transfers are performed by a word or byte read/write operation of the slot IP Module ID address space from the VMEbus. The ID PROM values are located every other byte at odd-byte locations (IP Module byte lane 0). For word-reads the even byte is discarded. The ID PROM area allows read or write accesses to support future IP Modules requiring writes to the ID address space. Read-Modify-Write cycles are supported for ID transfers.

2.2.3 I/O Transfers

IP Module I/O transfers are supported as A16/D16/D8 VME transfers. I/O transfers are performed by a word or byte read or write operation to the slot IP Module I/O address space from the VMEbus. Even and odd single-byte reads are supported. Read-Modify-Write cycles are supported for I/O transfers.

2.2.4 Interrupt Transfers

Interrupts are requested by the IP Module by assertion of N_INTREQ0 or N_INTREQ1. An independent VME interrupter handles each of these IP Module interrupt request lines and asserts one of the seven prioritized VME interrupt request lines. This provides a minimum delay in asserting the interrupt request to the VMEbus master.

NOTE: Signal nomenclature 'N_' prefix represents an active-low IP Module signal and a 'V_' prefix represents an active-low VMEbus signal

Which line is asserted depends on the interrupt level programmed into the carrier's IP Module slot-specific interrupt register. The four interrupt registers, one per IP Module slot, support independent selection of interrupt levels for both interrupt request signals N_INTREQ0 and N_INTREQ1. Upon receipt of the interrupt acknowledge from the VMEbus master, the VMESC4M will conduct a D8 transfer of the interrupt vector from the IP Module to the VMEbus. The interrupt request release will be dependent upon the individual IP Module being used.

2.2.5 Interrupt Prioritization

Interrupts in the VME system are prioritized in a two-tier fashion; by Interrupt Request Level (IRQ) and relative position. This tier effect is demonstrated in Figure 2-1.

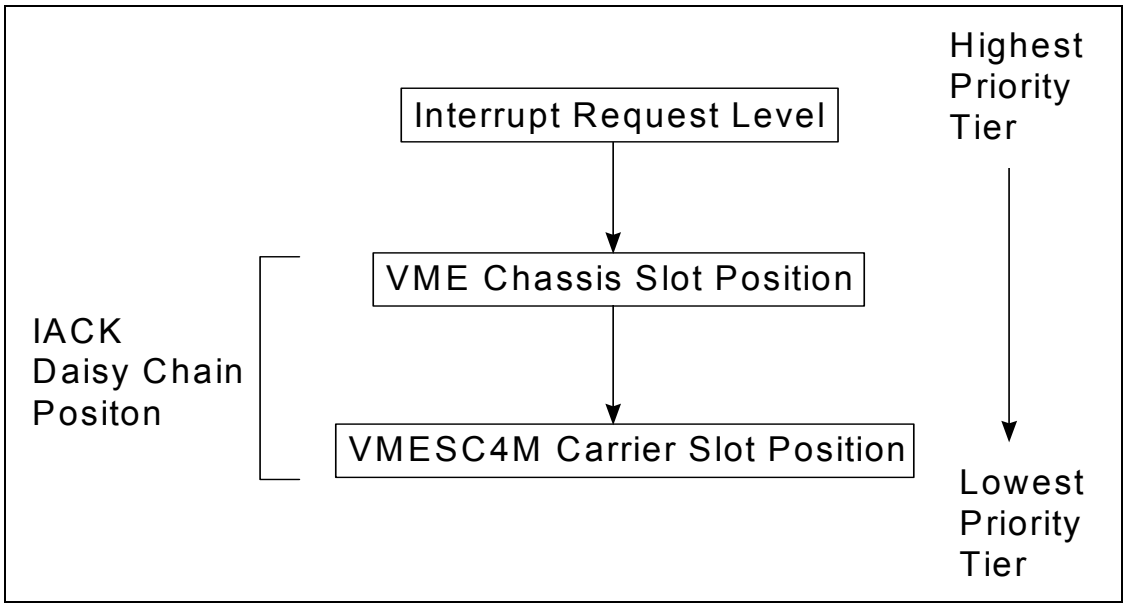


Figure 2-1 Interrupt Priority Tiers

The first tier of prioritization is the interrupt request level, with level 7 being the highest priority and level 1 the lowest as shown in Table 2-1.

Table 2-1 Interrupt Request Level Priorities

Interrupt Request Level	Priority
7	Highest
6	2nd Highest
5	•
4	•
3	•
2	2nd Lowest
1	Lowest

The second tier of prioritization is determined by the relative position in the interrupt daisy chain. The closer the interrupt requester is to the interrupt handler in the daisy chain, the higher its priority for a given request level. The relative position in the daisy chain is determined by the VME chassis slot location (see Figure 2-2) and the IP Module slot location on the VMESC4M carrier (see Figure 2-3).

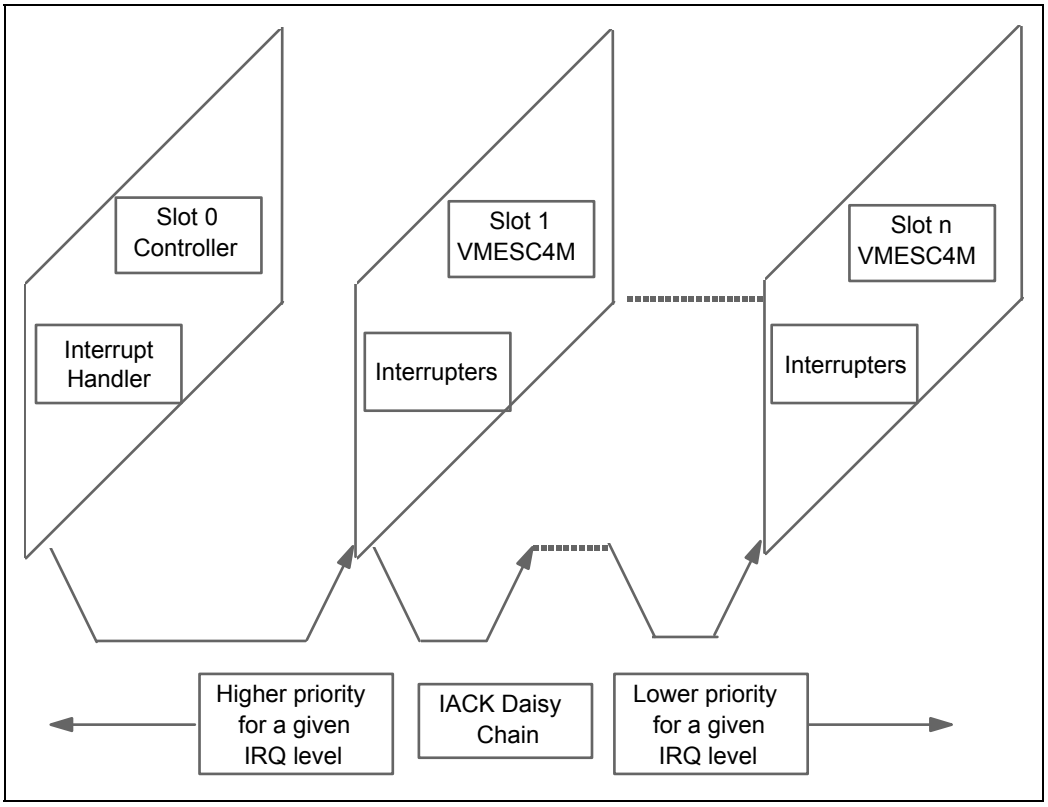


Figure 2-2 VME Chassis Slot Prioritization Through IACK Daisy Chain

The VMESC4M IP Module slot priority order from highest to lowest is A0, A1, B0, B1, C0, C1, D0, D1.

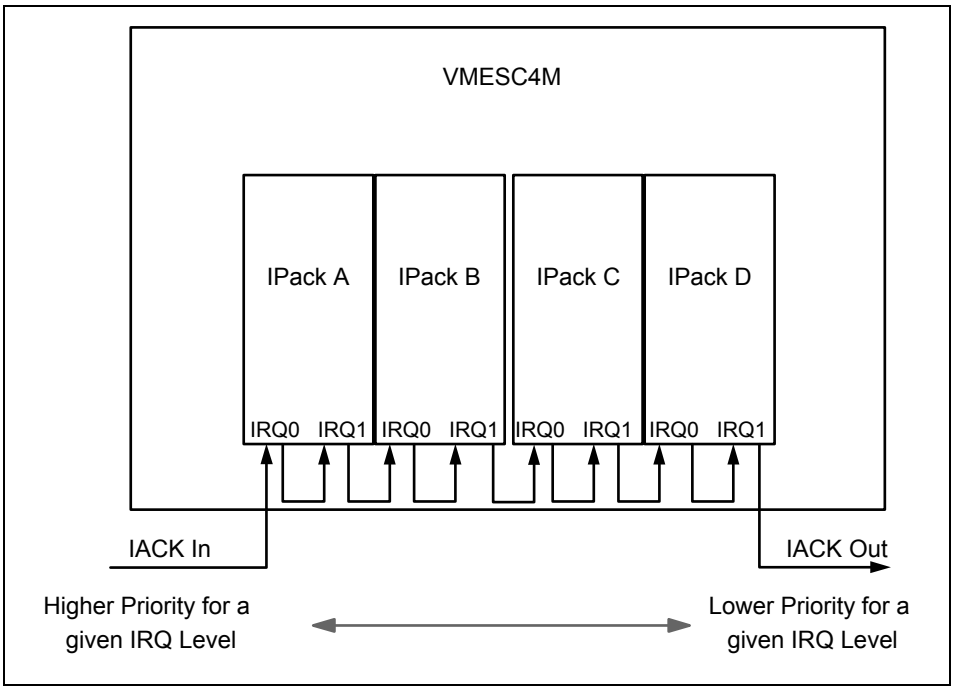


Figure 2-3 IACK Daisy Chain on the Carrier

2.2.6 General Purpose Registers

Four 8-bit general purpose user-defined registers are provided. Read-Modify-Write cycles are supported for general purpose register accesses.

2.2.7 Diagnostic LEDs

Six diagnostic LEDS are located on the front panel of the VMESC4M. Five of the LEDs, labeled A, B, C, D, and CRA, indicate when a read or write access is attempted to one of the four IP Module slots or the carrier registers. The sixth LED, labeled DT, indicates an acknowledgment by the IP Module or the VMESC4M of the attempted access. This scheme provides independent verification of the access and the acknowledgment.

2.2.8 IP Module Reset

Independent or simultaneous resetting of the IP Modules is facilitated through writes to the Reset register. Monitoring of the pulse-stretched reset signals is made possible by reading the Reset register. See Appendix B PROGRAMMING GUIDE for a detailed description of this register.

2.2.9 Error Status

Monitoring of the IP Module ERROR signals is made possible by reading the Error Status register. See Appendix B PROGRAMMING GUIDE for a detailed description of this register.

2.2.10 Strobe Signals

The IP Module STROBE signals can be connected via a 6-pin header (J17) located near the front panel.

2.2.11 Power Supply Filtering

Independent 'T' type filters are used for the +5, +12, and -12 volt power feeding each IP Module. This arrangement provides superior power supply filtering.

2.2.12 Fuses

Independent field replaceable fuses are used for the +5, +12, and -12 volt power feeding each IP Module. Although this arrangement costs a little more, it provides superior short-circuit protection.

NOTE: Replacement fuses can be purchased from DigiKey Phone 1-800 344-4593
1 Amp Fuse Manufacturer's Part Number R451001 (Littelfuse®, Inc.)
2 Amp Fuse Manufacturer's Part Number R451002 (Littelfuse®, Inc.)

2.3 Theory of Operation

2.3.1 Block Diagram Description

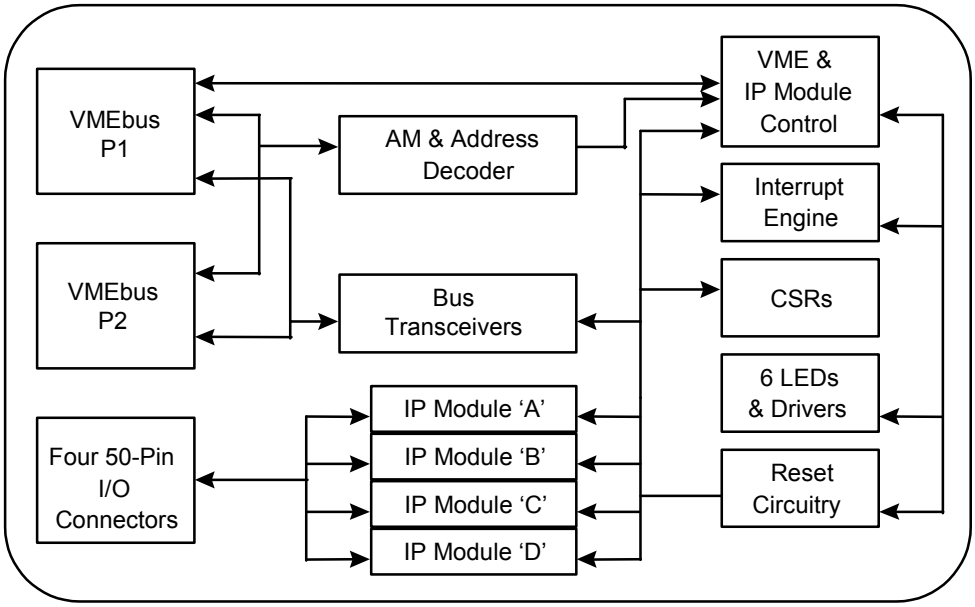


Figure 2-4 VMESC4M Block Diagram

Figure 2-4 is a simplified block diagram of the VMESC4M representing the signal flow between the VMEbus, IP Module Logic Bus and the IP Module I/O connectors. On the left side of the diagram is the VMEbus P1 and P2 connectors through which all transfers between the VME host computer and the VMESC4M's registers or IP Modules are conducted. The VMEbus transfers **do not** insert any IP Module hold cycles into the accesses to the IP Modules.

The block labeled "AM & Address Decoder" handles decoding of the VME address and address modifiers. The block labeled "VME & IP Module Control" handles the transfer of data between the VME and both the VMESC4M CSRs and the IP Modules.

The block labeled “Interrupt Engine” controls the VME Interrupt requests, and prioritization of VME Interrupt Acknowledge cycles. The large block in the center of the diagram represents the four IP Module slots labeled IP Module ‘A’ down to IP Module ‘E’. The IP Module I/O connectors are directly connected to the VMESC4M’s four front panel 50-pin I/O connectors. These connectors consist of two sets of 100-pin “stacked” headers such that I/O cables for IP slots “A” and “B” can be inserted with one directly over the other as can be for IP modules “C” and “D” respectively. The last two blocks at the lower right hand corner of the diagram contain the reset circuitry for each IP Module and drivers for each of the six front panel LEDs.

The P1 connector is located on the upper left of the VMESC4M Assembly Drawing Figure 2-5. When looking at the “top” or component side of the VMESC4M, the IP Module slots are labeled from left to right for slot A through slot D. All of the I/O signals are connected through the four 50-pin connectors located at the bottom of Figure 2-5.

2.3.2 Functional Logic Module Overview

The address decoding, VME and IP bus interface controllers, interrupt handler, and the reset control and LED driver circuitry are all contained within one functional logic module which was created using VHDL modeling techniques. The circuitry for this module, the VME4CTL is programmed into a single Altera 9000 series EPLD and can be sectioned into several functional areas as defined below.

2.3.2.1 Data Transfer Control

VME ADDRESS DECODER

This block detects VME host accesses to local VMESC4M Board Registers and IP Module I/O and ID spaces. An address-detect (HIT) signal corresponding to one of these address spaces is asserted when a matching address and address modifier (0x29 and 0x2D) is detected. This functional area will also compare values of the VME address bus during A24 (standard) or A32 (extended) address cycles to values in the memory base registers to generate internal memory hit signals.

DATA TRANSFER ENGINE

The DTE block handles the transfer of data between the VMEbus and the IP modules. It monitors the HIT signals from the address decoders to determine when a transfer is to take place. The VME control signals and address lines provide the rest of the transfer information. The DTE asserts the appropriate IP Module ID, I/O, or memory select line and executes an IP Module synchronous transfer. The DTE is implemented using combinational logic and four state machines: The State Machines include:

- One for VME Read cycles from the local control and status registers
- One for VME Write cycles to the local control and status registers
- One for IP Module I/O, ID, and memory Read cycles
- One for IP Module I/O, ID, and memory Write cycles

The IP Module read and write state machines complete a transfer in three IP Module clock cycles (six VME SYSCLOCK cycles). Specifically the IP Module READ state machine will place the data on the VMEbus upon detection of the appropriate IP Module acknowledge signal N_ACK, and the WRITE state machine will latch the data from the VMEbus and assert DTACK upon detection of the appropriate IP Module N_ACK signal.

REGISTERS

The VME4CTL module contains registers necessary for data transfers and interrupt cycles. These are the General Purpose Registers, Interrupt Level Registers, Reset Control Register, Error Status Register and the memory base address registers.

2.3.2.2 Interrupt Control

VME INTERRUPT ENGINE

This functional area detects (an) interrupt request(s) from one or more IP Modules, then asserts and selects VME interrupt request level(s). It monitors the VME V_IACKIN and A3-A1 signals and determines if the interrupt being acknowledged is the one it requested. If the level is different than the one requested, the acknowledge signal is passed on via V_IACKOUT daisy chain. If the acknowledge is for this IP Module's interrupt, the DTE asserts an internal interrupt HIT signal indicating that the corresponding IP Module's interrupt vector is to be transferred. If more than one IP Module interrupts at the same level and time, the interrupts are processed in a prioritized fashion. This scheme is from IP Module slot A's N_INTREQ0 as the highest priority, then slot A's N_INTREQ1 as the second highest, down to the slot D's N_INTREQ1 as the lowest. Refer to section 2.2.5 for a detailed description of interrupt prioritization.

This interrupt priority scheme is implemented by the user by programming a specific level in the interrupt level registers. The Interrupt Engine is implemented using combinational logic and a state machine. The state machine places an interrupt vector on the VMEbus just like a VME read cycle. The IP Module Interrupt Acknowledge Cycle places the interrupt vector on the VMEbus in three IP Module clock cycles (six VME SYSCLOCK cycles). Specifically the IP Module interrupt vector state machine will place the vector on the VMEbus upon detection of the appropriate IP Module acknowledge signal N_ACK.

2.3.2.3 Access LED/Reset Control

FRONT PANEL LED AND IP MODULE RESET DRIVERS

The VME4CTL module is also responsible for combinationally driving the front panel LEDs from the one-shot drivers, and for logically "ORing" the VME system reset (V_SYSRESET) with the individual IP Module reset one-shot signals for driving each IP Module Reset signal.

2.4 VMESC4M Schematics, Components and Functions

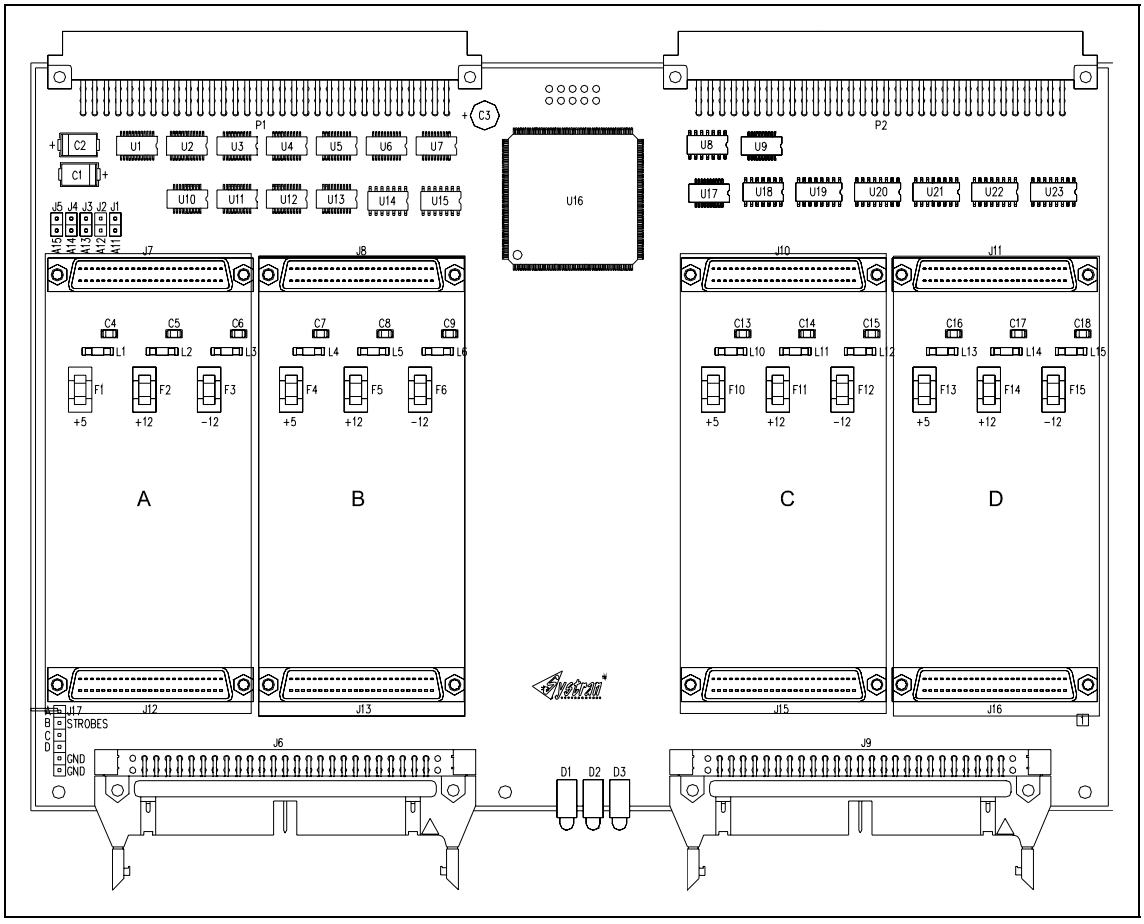


Figure 2-5 VMESC4M Mechanical Assembly

The VMESC4M shown in Figure 2-5 was developed using VHDL and Synthesis, and targeted to an Altera EPLD. The design was schematic-captured and integrated to standard interface devices, connectors and discrete components using VIEWLogic CAE tools. This topdown methodology is evident in the schematics and product design material presented in this document. For those readers not familiar with the topdown methodology used to develop the VMESC4M, the following is a short presentation on the schematics of the VMESC4M.

3. INSTALLATION

3.1 Unpacking the VMESC4M

Table 3-1 lists the contents of the VMESC4M shipping package.

Table 3-1 Contents of VMESC4M Shipping Packages

Qty	Description
1	VMESC4M Printed Circuit Assembly
2	Surface-mount spare fuses (one-1Amp, one-2Amp)
1	VMESC4M User Manual *

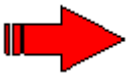
* One documentation CD is shipped with each order. Extra manuals may be printed from pdf files on the CD as needed.

The Printed Circuit Assembly is enclosed in an anti-static box. The box and the manual are packaged together in a larger box. Save the shipping material in case the board needs to be returned.

3.2 Visual Inspection of the VMESC4M

Examine the VMESC4M to determine if any damage occurred during shipping.

3.3 VMESC4M Installation



NOTE: The VMESC4M is an Electrostatic Sensitive Device (ESD). The hardware installation of the VMESC4M must be conducted on a good anti-static workbench to protect the IP Modules and carrier board.

The tools required for the VMESC4M installation are listed in Table 3-2.

Table 3-2 VMESC4M Installation Tools

QTY	DESCRIPTION
1	ESD Static Control Kit/Ground Strap/Etc.
1	Standard Flat Head Screwdriver (Optional)

3.3.1 Installation of IP Modules on the VMESC4M

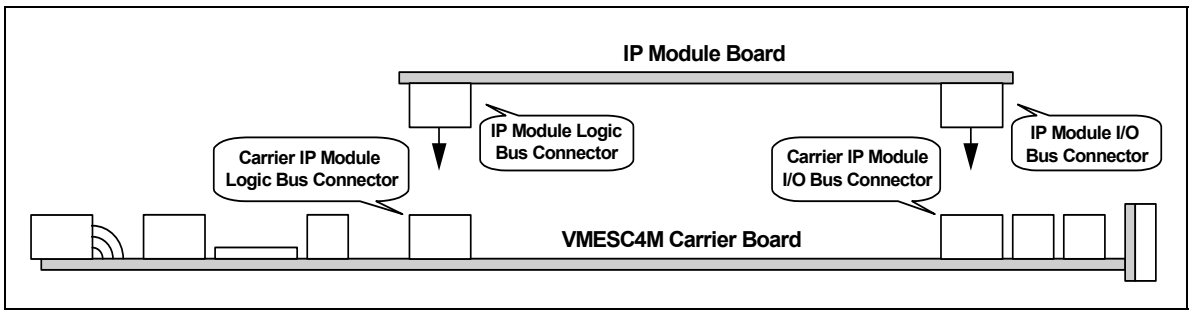


Figure 3-1 Installation of an IP Module on the VMESC4M

Referring to the appropriate figures and table described below, perform the following steps in removing the VMESC4M from it shipping container (or from an existing installation) and installing IP Modules. The asterisk (*) denotes optional items.

1. Turn off all power to the host system.

2a. Remove the VMESC4M from it's shipping container and move it to the ESD controlled area where the installation of the IP Modules can be made.

*2b. Remove the VMESC4M from the VME card cage and move it to the ESD controlled area where the installation of the IP Modules can be made.

3. Set the VMEbus base address on the J5-J1 jumpers (see Figure 3-2 and Figure 3-3).

4. Install the IP Module(s) onto the carrier board by applying adequate and equal pressure to the IP Module(s) at both ends and the VMESC4M board.

*5. Install four M2x5mm flat head machine screws onto the IP Module's connectors.

*6. Install I/O cables necessary onto VMESC4M.

7. Install the VMESC4M into desired slot of a VME card cage.

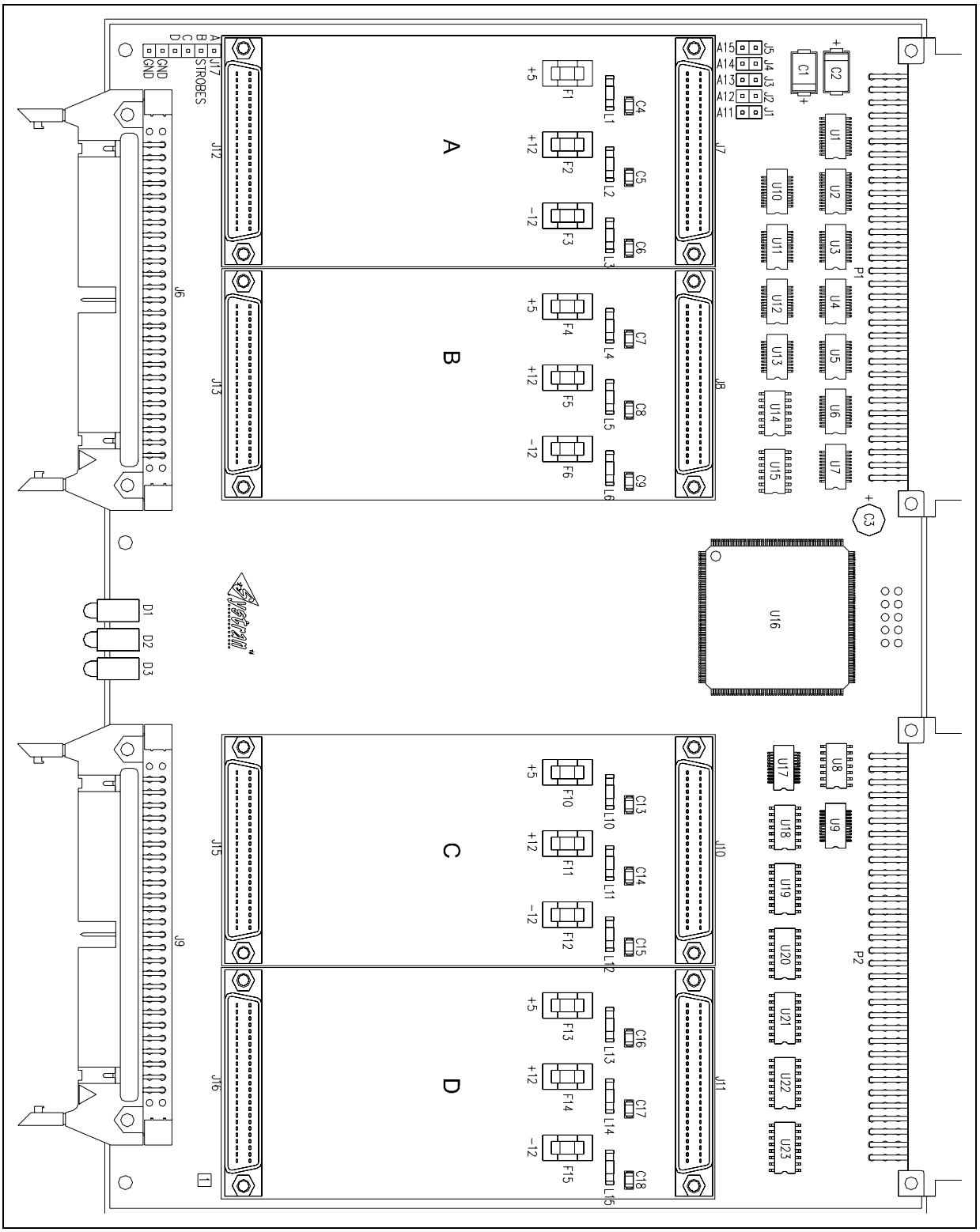


Figure 3-2 VMESC4M Assembly Drawing

CURTISS WRIGHT Controls, Inc. Embedded Computing

3.3.2 VMESC4M VME Base Address

The VMESC4M VME board base address is set with five jumpers J5 - J1, corresponding to VME address bits A15 to A11, respectively. The board's register and I/O space can be placed at one of 32 possible base addresses on 2 KB boundaries in the VME short address space.

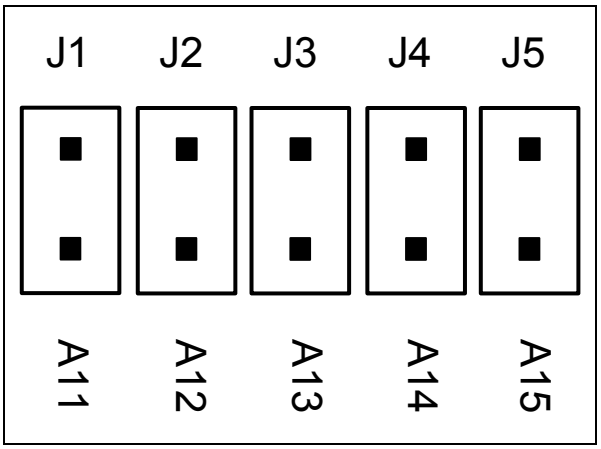


Figure 3-3 VMESC4M Jumpers J5 - J1

Placing a jumper across both posts corresponds to a logic low, or 0, for that address bit. Removing the jumper, or storing it on a single post, corresponds to a logic high, or 1, for that address bit.

NOTE: The VMESC4M is shipped with all jumpers installed and appears in memory at address 0x0000.

EXAMPLE:

The short I/O space on the MVME162 is located from 0xFFFF0000 to 0xFFFFFFFF, so the VMESC4M could be placed from 0xFFFF0000 to 0xFFFFF800 on 2 K boundaries.

Table 3-3 indicates all possible jumper settings for board base addressing.

Refer to Appendix B PROGRAMMING GUIDE for details of VME accesses to the VMESC4M on-board registers or IP Modules.

NOTE: When using the MVME162 with OS-9, OS-9 maps the VMEChip2 GCSRs (Global Control Status Registers) to the beginning of short I/O space. This causes a bus conflict with boards placed at this address. To avoid a bus conflict:

1) do not put any VME boards at locations 0xFFFF0000 to 0xFFFF0010;
1) move the GCSR location by changing the Group Select and Board Select values in the VMEChip2 LCSR registers (the 3 bytes starting at 0xFFF4002C); or
1) disable the map decoder by writing an 'F' to the Board Select value (0xFFF4002E). Refer to the MVME162 manual for more details.

Also, OS-9 probes location 0xFFFF10C0 for a disc controller on power-up. Locating a VME board at this address can cause unpredictable results..

Table 3-3 VMESC4M Board Base Address Jumper Selection Table

J5(A15)	J4(A14)	J3(A13)	J2(A12)	J1(A11)	VME Short I/O Address
ON	ON	ON	ON	ON	0x0000
ON	ON	ON	ON	OFF	0x0800
ON	ON	ON	OFF	ON	0x1000
ON	ON	ON	OFF	OFF	0x1800
ON	ON	OFF	ON	ON	0x2000
ON	ON	OFF	ON	OFF	0x2800
ON	ON	OFF	OFF	ON	0x3000
ON	ON	OFF	OFF	OFF	0x3800
ON	OFF	ON	ON	ON	0x4000
ON	OFF	ON	ON	OFF	0x4800
ON	OFF	ON	OFF	ON	0x5000
ON	OFF	ON	OFF	OFF	0x5800
ON	OFF	OFF	ON	ON	0x6000
ON	OFF	OFF	ON	OFF	0x6800
ON	OFF	OFF	OFF	ON	0x7000
ON	OFF	OFF	OFF	OFF	0x7800
OFF	ON	ON	ON	ON	0x8000
OFF	ON	ON	ON	OFF	0x8800
OFF	ON	ON	OFF	ON	0x9000
OFF	ON	ON	OFF	OFF	0x9800
OFF	ON	OFF	ON	ON	0xA000
OFF	ON	OFF	ON	OFF	0xA800
OFF	ON	OFF	OFF	ON	0xB000
OFF	OFF	OFF	OFF	OFF	0xB800
OFF	OFF	ON	ON	ON	0xC000
OFF	OFF	ON	ON	OFF	0xC800
OFF	OFF	ON	OFF	ON	0xD000
OFF	OFF	ON	OFF	OFF	0xD800
OFF	OFF	OFF	ON	ON	0xE000
OFF	OFF	OFF	ON	OFF	0xE800
OFF	OFF	OFF	OFF	ON	0xF000
OFF	OFF	OFF	OFF	OFF	0xF800

Where ON = logic ‘0’, OFF = logic ‘1’

3.3.3 Installation of I/O Cables

The VMESC4M provides easy access to all four IP Module I/O connectors. The I/O connectors are located near the front panel where standard 50-pin flat cables can be installed by directly plugging into headers protruding through the front panel and securing the tabs behind the cable ends. This allows for easy system integration.

All of the VMESC4M IP Module I/O connector pin numbers directly correspond to each IP Module I/O pinout (i.e. a one-to-one relationship with pin 50 connected to pin 50 down to pin 1 connected to pin 1).

On the VMESC4M:

- Connector J6 is for interconnect to slots “A” and “B” IP Modules. This connector is located at the upper end of the panel (facing the front) and consists of two 50-pin standard headers stacked side-by-side. The “A” I/O connector is the left side of the double connector while the “B” I/O connector is on the right.
- Connector J9 is for interconnect to slots “C” and “D” IP Modules. This connector is located at the lower end of the panel (facing the front) and consists of two 50-pin standard headers stacked side-by-side. The “C” I/O connector is the left side of the double connector while the “D” I/O connector is on the right.

3.4 VMESC4M IP Module STROBE Connector

IP Module Logic Bus connector has a signal called “N_STROBE” and is user defined by the IP Module Logic Bus Specification. On the VMESC4M each of the strobe signals are provided through J17 with the IP Module slot “A” strobe signal on pin 1 through slot “D” on pin 4 with pin 5 and 6 grounded. Each of the strobe signals on the VMESC4M are pulled-up to +5 V via 10 KΩ resistors. The strobe connector (J17) is located at the upper left of Figure 3-2 VMESC4M Assembly Drawing (which is at the top of the VMESC4M when installed).

Table 3-4 VMESC4M STROBE Connector Pin Assignments

Pin Number	Signal Name
1	N_STROBE_A
2	N_STROBE_B
3	N_STROBE_C
4	N_STROBE_D
5	GROUND
6	GROUND

Table 3-5 shows the pin assignments for the VMEbus P1 connector. The signals on the left side are of the original VME specification signal nomenclature, and the signals on the right are those used by Curtiss-Wright.

Table 3-5 VMEbus P1 Rows A, B and C Pin Assignments

VMEbus P1 Pin #	VMEbus ROWa Signal	Curtiss-Wright ROWa Signal	VMEbus ROWb Signal	Curtiss-Wright ROWb Signal	VMEbus ROWc Signal	Curtiss-Wright ROWc Signal
1	D00	VMED0	BBSY*	N/C	D08	VMED8
2	D01	VMED1	BCLR*	N/C	D09	VMED9
3	D02	VMED2	ACFAIL*	N/C	D10	VMED10
4	D03	VMED3	BG0IN*	V_BG0IN	D11	VMED11
5	D04	VMED4	BG0OUT*	V_BG0OUT	D12	VMED12
6	D05	VMED5	BG1IN*	V_BG1IN	D13	VMED13
7	D06	VMED6	BG1OUT*	V_BG1OUT	D14	VMED14
8	D07	VMED7	BG2IN*	V_BG2IN	D15	VMED15
9	GND	GND	BG2OUT*	V_BG2OUT	GND	GND
10	SYSCLK	SYSCLK_IN	BG3IN*	V_BG3IN	SYSFAIL*	N/C
11	GND	GND	BG3OUT*	V_BG3OUT	BERR*	VME_BERR
12	DS1*	VME_DS1	BR0*	N/C	SYSRESET*	VME_SYSRESET
13	DS0*	VME_DS0	BR1*	N/C	LWORD*	V_LWORD
14	WRITE*	VME_WRITE	BR2*	N/C	AM5	AM5
15	GND	GND	BR3*	N/C	A23	VME_A23
16	DTACK*	V_DTACK	AM0	AM0	A22	VME_A22
17	GND	GND	AM1	AM1	A21	VME_A21
18	AS*	V_AS	AM2	N/C	A20	VME_A20
19	GND	GND	AM3	AM3	A19	VME_A19
20	IACK*	V_IACK	GND	GND	A18	VME_A18
21	IACKIN*	V_IACKIN	SERCLK*	N/C	A17	VME_A17
22	IACKOUT*	V_IACKOUT	SERDAT*	N/C	A16	VME_A16
23	AM4	AM4	GND	GND	A15	VME_A15
24	A07	VME_A7	IRQ7*	V_IRQ7	A14	VME_A14
25	A06	VME_A6	IRQ6*	V_IRQ6	A13	VME_A13
26	A05	VME_A5	IRQ5*	V_IRQ5	A12	VME_A12
27	A04	VME_A4	IRQ4*	V_IRQ4	A11	VME_A11
28	A03	VME_A3	IRQ3*	V_IRQ3	A10	VME_A10
29	A02	VME_A2	IRQ2*	V_IRQ2	A09	VME_A9
30	A01	VME_A1	IRQ1*	V_IRQ1	A08	VME_A8
31	-12V	-12V	+5VSTDBY	N/C	+12V	+12V
32	+5V	VDD	+5V	VDD	+5V	VDD

Table 3-6 shows the pin assignments for the VMEbus P2 connector. Note that no connections are made to rows A and C such that these rows will be filled with an "N/C" designation. As with Table 3-5, the signals on the left side are of the original VME specification signal nomenclature while the ones on the right are those used by Curtiss-Wright.

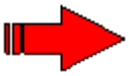
Table 3-6 VMEbus P2 Rows A, B and C Pin Assignments

VMEbus P2 Pin #	VMEbus ROWa Signal	Curtiss-Wright ROWa Signal	VMEbus ROWb Signal	Curtiss-Wright ROWb Signal	VMEbus ROWc Signal	Curtiss-Wright ROWc Signal
1	User Defined	N/C	+5 V	VDD	User Defined	N/C
2	User Defined	N/C	GND	GND	User Defined	N/C
3	User Defined	N/C	Reserved	N/C	User Defined	N/C
4	User Defined	N/C	A24	VME_A24	User Defined	N/C
5	User Defined	N/C	A25	VME_A25	User Defined	N/C
6	User Defined	N/C	A26	VME_A26	User Defined	N/C
7	User Defined	N/C	A27	VME_A27	User Defined	N/C
8	User Defined	N/C	A28	VME_A28	User Defined	N/C
9	User Defined	N/C	A29	VME_A29	User Defined	N/C
10	User Defined	N/C	A30	VME_A30	User Defined	N/C
11	User Defined	N/C	A31	VME_A31	User Defined	N/C
12	User Defined	N/C	GND	GND	User Defined	N/C
13	User Defined	N/C	+5 V	VDD	User Defined	N/C
14	User Defined	N/C	D16	N/C	User Defined	N/C
15	User Defined	N/C	D17	N/C	User Defined	N/C
16	User Defined	N/C	D18	N/C	User Defined	N/C
17	User Defined	N/C	D19	N/C	User Defined	N/C
18	User Defined	N/C	D20	N/C	User Defined	N/C
19	User Defined	N/C	D21	N/C	User Defined	N/C
20	User Defined	N/C	D22	N/C	User Defined	N/C
21	User Defined	N/C	D23	N/C	User Defined	N/C
22	User Defined	N/C	GND	GND	User Defined	N/C
23	User Defined	N/C	D24	N/C	User Defined	N/C
24	User Defined	N/C	D25	N/C	User Defined	N/C
25	User Defined	N/C	D26	N/C	User Defined	N/C
26	User Defined	N/C	D27	N/C	User Defined	N/C
27	User Defined	N/C	D28	N/C	User Defined	N/C
28	User Defined	N/C	D29	N/C	User Defined	N/C
29	User Defined	N/C	D30	N/C	User Defined	N/C
30	User Defined	N/C	D31	N/C	User Defined	N/C
31	User Defined	N/C	GND	GND	User Defined	N/C
32	User Defined	N/C	+5V	VDD	User Defined	N/C

Table 3-7 shows the pin assignments for each IP Module Logic Bus connector. The signals on the left side of the connector are of the original IP Module signal nomenclature, and the signals on the right are those used by Curtiss-Wright. The Upper case "X" in the Curtiss-Wright signal name represents the "A", "B", "C", or "D" mnemonic for each IP Module slot.

Table 3-7 IP Module Logic Bus Pin Assignments

Original IP Module Signals Names	IP Module Logic Bus Pin #	Curtiss-Wright Signal Names	Bussed or Unique	P = Pulled Up Via 10 KΩ Resistor
GND	50	GND	GND	GND
Reserved	49	RESERVED1	U	P
Ack*	48	N_IPXACK	U	P
A6	47	VMEA6	B	
Strobe*	46	N_STROBE_X	U	P
A5	45	VMEA5	B	
Intreq1*	44	N_IPXINTREQ1	U	P
A4	43	VMEA4	B	
Intreq0*	42	N_IPXINTREQ0	U	P
A3	41	VMEA3	B	
Error*	40	N_IPXERROR	U	P
A2	39	VMEA2	B	
DMAEnd*	38	N_DMAEND	U	P
A1	37	IPA1	B	
Reserved	36	RESERVED2	U	P
IOSel*	35	N_IPXIOSEL	U	
DMAck0*	34	N_DMACK0	U	P
IntSel*	33	N_IPXINTSEL	U	
DMAReq1	32	N_DMAREQ1	U	P
MemSel*	31	N_MEMSEL	U	
DMAReq0	30	N_DMAREQ0	U	P
IDSel*	29	N_IPXIDSEL	U	
R/W*	28	V_WRITE	B	
+5V	27	+5VDC	+5 Vdc	+5 Vdc
GND	26	GND	GND	GND
GND	25	GND	GND	GND
+5V	24	+5VDC	+5 Vdc	+5 Vdc
+12V	23	+12VDC	+12 Vdc	+12 Vdc
-12V	22	-12VDC	-12 Vdc	-12 Vdc
BS1*	21	V_BS1	B	
BS0*	20	V_BS0	B	
D15	19	VLD15	B	P
D14	18	VLD14	B	P
D13	17	VLD13	B	P
D12	16	VLD12	B	P
D11	15	VLD11	B	P
D10	14	VLD10	B	P
D9	13	VLD9	B	P
D8	12	VLD8	B	P
D7	11	VLD7	B	P
D6	10	VLD6	B	P
D5	9	VLD5	B	P
D4	8	VLD4	B	P
D3	7	VLD3	B	P
D2	6	VLD2	B	P
D1	5	VLD1	B	P
D0	4	VLD0	B	P
Reset*	3	N_IPXRESET	U	
CLK	2	ICLK_X	B	
GND	1	GND	GND	GND



NOTE: The IP Module data bus, usually defined as IPD[15:0] for all of our IP Modules, is a subset of the information that appears on these signal lines. Since carrier register information also uses these signal lines, the bus was defined as the "VMESC4M Local Data Bus" designated as VLD[15:0].

APPENDIX A
SPECIFICATIONS

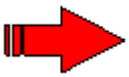
A.1 SPECIFICATIONS

MECHANICAL

- Measurements:
  Length: 9.187 inches (23.33 cm)
  Width: 6.299 inches (15.99 cm)
- Weight: 10.28 ounces (291 grams) (includes front panel)
- Board Thickness: 0.062 inches (0.157 cm), nominally, (6 layers)

ELECTRICAL

- Power (No IP Modules installed): +5 Vdc (+5%) @ 0.43 Amps, +12 Vdc @ 0.0 Amps, - 12 Vdc @ 0.0 Amps



NOTE: The VMEbus ground and IP Modules' ground are not isolated through this board.

ABSOLUTE MAXIMUM SUPPLY RATINGS

- +5 Vdc Supply voltage with respect to ground: -0.5 Vdc minimum and +7.0 Vdc maximum
- +12 Vdc Supply voltage with respect to ground: Dependent on the IP Modules installed.
- -12 Vdc Supply voltage with respect to ground: Dependent on the IP Modules installed.

RECOMMENDED OPERATING SUPPLY RATINGS

- Supply Voltage with respect to ground: +4.75 Vdc to +5.25 Vdc

ENVIRONMENTAL SPECIFICATIONS:

- Temperature:
  (Operating): -0°C to +70°C
  (Storage): -45°C to +85°C
- Humidity (Noncondensing): 5% to 95%

MEAN TIME BETWEEN FAILURE (MTBF):

514,704 per MIL-HDBK-217F
938,338 Bellcore

The MTBF numbers are based on calculations using MIL-HDBK-217F, Appendix A; and Bellcore 332, Issue 6, for a ground-benign environment.

This page intentionally left blank.

APPENDIX B
PROGRAMMING GUIDE

TABLE OF CONTENTS

TABLES

B.1 Overview

This section of the manual describes the operation of the VMESC4M from the software perspective, detailing the VMESC4M registers, the overall mapping and addressing scheme for the board, and provides programming examples. A more detailed description of the hardware can be found in section 2.0 PRODUCT OVERVIEW, and application concepts are discussed in Appendix D TYPICAL APPLICATIONS.

B.2 Description

The VMESC4M is an easy to use VME6U slave card that holds four singlewide, two singlewide and one doublewide, or two doublewide IP Modules. In addition to providing access to the I/O and ID space of the IP Modules via the VME bus, the VMESC4M has 12 onboard control/status registers.

B.3 Board Base Address

The VMESC4M can be located anywhere in the VME short IO space (A16 space address modifier codes 0x29 and 0x2D) on 2 KB boundaries by setting the board base address jumpers. These jumpers, J1 to J5, correspond to address lines A11 to A15 respectively. A detailed description of setting the board base address is found in section 3.3.2.

B.4 Address Map

The IP Module IO and ID spaces, and the carrier's control and status registers, reside at a fixed relative address from the VMESC4M base address. Table B- 1 shows the complete memory map for the board.

Table B- 1 VMESC4M Address Map

Address Space or Register Name	VME Address (relative to board base address)	Size
IP Module A I/O	Base+0x000	128 bytes
IP Module A ID	Base+0x080	128 bytes
IP Module B I/O	Base+0x100	128 bytes
IP Module B ID	Base+0x180	128 bytes
IP Module C I/O	Base+0x200	128 bytes
IP Module C ID	Base+0x280	128 bytes
IP Module D I/O	Base+0x300	128 bytes
IP Module D ID	Base+0x380	128 bytes
IP Module Reset Register	Base+0x500 - Word Access Base+0x501 - Byte Access	4 bits
Error Status Register	Base+0x502 - Word Access Base+0x503 - Byte Access	4 bits
IP Module A Interrupt Register	Base+0x580 - Word Access Base+0x581 - Byte Access	6 bits
IP Module B Interrupt Register	Base+0x582 - Word Access Base+0x583 - Byte Access	6 bits
IP Module C Interrupt Register	Base+0x584 - Word Access Base+0x585 - Byte Access	6 bits
IP Module D Interrupt Register	Base+0x586 - Word Access Base+0x587 - Byte Access	6 bits
General Purpose Register A	Base+0x600 - Word Access Base+0x601 - Byte Access	8 bits
General Purpose Register B	Base+0x602 - Word Access Base+0x603 - Byte Access	8 bits
General Purpose Register C	Base+0x604 - Word Access Base+0x605 - Byte Access	8 bits
General Purpose Register D	Base+0x606 - Word Access Base+0x607 - Byte Access	8 bits
A24 Memory Base Register	Base+0x680 - Word Access Base+0x681 - Byte Access	2 bits
A32 Memory Base Register	Base+0x682 - Word Access Base+0x683 - Byte Access	7 bits
Memory Enable Register	Base+0x684 - Word Access Base+0x685 - Byte Access	2 bits

B.5 Register Descriptions

The VMESC4M registers can be accessed as byte (D8) or word (D16) values. When accessed as words, the upper byte is not driven by the carrier and its value is all '1's due to VLD[15:8] pull-up resistors. The unused bits in the lower byte are driven as zero's on reads. All unused bits are discarded on writes. The VME addresses for the registers shown below are relative to the board base address and are in the format (WORD/BYTE) for word and byte accesses respectively.

B.5.1 Reset Register (0x500/0x501)

Setting one of the bits RES_D through RES_A to a one, asserts the corresponding IP Module's reset line. The bits can be set individually by writing a '1' to that bit or simultaneously with a single write of 0x0F. When set, a one shot is triggered supplying a minimum 200 ms reset pulse as required by the IP Module specification. When read, these bits return the status of the corresponding IP Module's reset line (a'0' means the IP Module is not being reset, a '1' means it is in a reset condition).

Table B- 2 Reset Register Bit Description

Bit #	B15-B8	B7-B4	B3	B2	B1	B0
Bit Name	Not Used	Not Used	RES_D	RES_C	RES_B	RES_A
R/W	Reads '1's Writes discarded	Read '0's Writes discarded	R/W	R/W	R/W	R/W
Power-up State	N/A	N/A	0	0	0	0

NOTE: RES_D-RES_A: Reset bits for IP Module slots D to A.

B.5.2 Error Status Register (0x502/0x503)

These bits indicate the state of the corresponding IP Module N_xERROR. If the N_xERROR* signal is asserted (active low) for a particular IP Module, then the corresponding ERR_N bit will be set to a 1. If the error signal is not asserted, then the corresponding bit will be cleared.

Table B- 3 Error Status Register Bit Description

Bit #	B15-B8	B7-B4	B3	B2	B1	B0
Bit Name	Not Used	Not Used	ERR_D	ERR_C	ERR_B	ERR_A
R/W	Reads '1's Writes discarded	Reads '0's Writes discarded	R	R	R	R
Power-up State	N/A	N/A	State of IP Module D ERR* Signal	State of IP Module C ERR* Signal	State of IP Module B ERR* Signal	State of IP Module A ERR* Signal

NOTE: ERR_D-ERR_A: Error Status bits for IP Module slots D to A.

B.5.3 Interrupt Level Registers A to D (0x580/0x581 to 0x586/0x587)

The three interrupt level bits, IL2-IL0, determine the interrupt request level that will be asserted on the VME bus when the corresponding IP Module IRQ line is asserted. Valid interrupt levels are 1 to 7. A value of '0', not a valid interrupt request level, will not assert a VME interrupt request. Thus, a value of '0' can be written to these registers to disable interrupts.

The interrupt requests are prioritized first by interrupt request level, then by slot position. Level 7 interrupt requests are the highest priority and level 1 is the lowest. Equal interrupt request levels are serviced in the slot prioritized order A0, A1, B0, ... C1, D0, D1. See section 2.0 PRODUCT OVERVIEW for more details regarding interrupt prioritization.

Table B- 4 Interrupt Level Register Bit Description

Bit #	B15-B8	B7	B6	B5	B4	B3	B2	B1	B0
Bit Name	Not Used	0	IRQ1 IL2	IRQ1 IL1	IRQ1 IL0	0	IRQ0 IL2	IRQ0 IL1	IRQ0 IL0
R/W	Reads '1's Writes discarded	Read '0' Writes discarded	R/W	R/W	R/W	Read '0' Writes discarded	R/W	R/W	R/W
Power-up State		0	0	0	0	0	0	0	0

NOTE: IRQN IL2-IL0: The interrupt level bits for IRQN.

B.5.4 General Purpose Registers A to D (0x600/0x601 to 0x606/0x607)

General purpose registers A to D are completely user definable. These read/write registers can be used for semaphores, shared memory, etc. See Appendix D TYPICAL APPLICATIONS for more application information regarding these registers.

Table B- 5 General Purpose Register Bit Description

Bit #	B15-B8	B7	B6	B5	B4	B3	B2	B1	B0
Bit Name	Not Used	GPR7	GPR6	GPR5	GPR4	GPR3	GPR2	GPR1	GPR0
R/W	Reads '1's Writes discarded	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Power-up State		0	0	0	0	0	0	0	0

NOTE: GPR7-GPR0: The General Purpose Register byte value.

B.5.5 A24 Memory Base Register (0x680/0x681)

When the VMESC4M carrier is configured (via the Memory Enable Register) to use A24 space to access the IP Modules' memory spaces, this register defines the VME base address at which accesses will initiate memory cycles for the corresponding IP Modules. The carrier will occupy 4MB of A24 space; the first 1MB will access IP Module A's first 1MB of memory space, the second 1MB will access IP Module B's first 1MB of memory

space, etc. Accesses to an IP Module's memory space above 1MB are not possible when using A24 address space.

Table B- 6 A24 Memory Base Register Description

Bit #	B15-B8	B7	B6	B5-B0
Bit Name	Not Used	A23	A22	0
R/W	Reads '1's Writes discarded	R/W	R/W	Read '0' Writes discarded
Power-up State		0	0	0

NOTE: A23-A22: The A24 memory base used by the carrier to detect accesses to IP Module memory space.

B.5.6 A32 Memory Base Register (0x682/0x683)

When the VMESC4M carrier is configured (via the Memory Enable Register) to use A32 space to access the IP Modules' memory spaces, this register defines the VME base address at which accesses will initiate memory cycles for the corresponding IP Modules. The carrier will occupy 32 MB of A32 space; the first 8 MB will access IP Module memory space, the second 8 MB region will access IP Module B's 8 MB of memory space, etc.

Table B- 7 A32 Memory Base Register Description

Bit #	B15-B8	B7	B6	B5	B4	B3	B2	B1	B0
Bit Name	Not Used	A31	A30	A29	A28	A27	A26	A25	0
R/W	Reads '1's Writes discarded	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Read '0' Writes discarded
Power-up State		0	0	0	0	0	0	0	0

NOTE: A31-A25: The A32 memory base used by the carrier to detect accesses to IP Module memory space.

B.5.7 Memory Enable Register (0x684/0x685)

Accesses to IP Module memory space are configured using this register. If the MEM_EN bit is cleared, accesses to the memory spaces of the IP Modules are disabled. If the MEM_EN bit is set to a '1', accesses to the memory spaces are enabled. If the MEM_EN bit is set to a '1' and the SPACE_SEL is cleared, the IP Modules' memory spaces can be accessed via VME A24 cycles. If the MEM_EN bit and the SPC_SEL bit are both set to a '1', the IP Modules' memory spaces can be accessed via VME A32 cycles.

Table B- 8 Memory Enable Register Description

Bit #	B15-B8	B7-B2	B1	B0
Bit Name	Not Used	0	SPACE_SEL	MEM_EN
R/W	Reads '1's Writes discarded	Read '0' Writes discarded	R/W	R/W
Power-up State		0	0	0

B.6 VMESC4M Programming Examples

The following examples illustrate how to program the VMESC4M to achieve various operational modes.

B.6.1 Reset Example

This example resets one, then all four IP Modules, and then monitors the reset signals to determine when they are de-asserted. Setting a bit in the reset register causes a one-shot to assert the corresponding IP Module reset signal (the one-shot asserts the reset signal for approximately 240 ms). The reset signal can then be monitored by reading the Reset register to determine when it is de-asserted.

- Write 0x0001 to the Reset register. This resets the IP Module in slot A by asserting the IP Module A reset signal.

LOOP1:

- Read the Reset register.
- If Bit 0 is set then go to LOOP1.
- Write 0x000F to the Reset register. This resets all four IP Module modules by asserting their corresponding reset signals.

LOOP2:

- Read the Reset register.
- If any of bits [3:0] are set then go to LOOP2.

B.6.2 Interrupt Initialization Example

Interrupts in the VME system are prioritized in a two tier fashion:

- The first tier of prioritization is the interrupt request level, with level 7 being the highest priority and level 1 the lowest.
- The second tier of prioritization is determined by the relative position in the interrupt daisy chain.

The closer the interrupt requester is to the interrupt handler in the daisy chain, the higher its priority for a given request level. The relative position in the daisy chain is determined by the VME chassis slot location and the IP Module slot location on the VMESC4M carrier. The VMESC4M IP Module slot priority order from highest to lowest is A0, A1, B0, ... C1, D0, D1. See section 2.0 or Appendix D for more details regarding interrupt

prioritization. The following examples illustrate several interrupt service priority schemes.

EXAMPLE 1

In this example, all interrupters are set to the same interrupt request level.

- Write 0x0011 to Interrupt Level register A
- Write 0x0011 to Interrupt Level register B
- Write 0x0011 to Interrupt Level register C
- Write 0x0011 to Interrupt Level register D

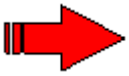
Now all ten interrupt sources are set to interrupt request level 1. If all ten sources assert an interrupt request at the same time, they will be serviced in the following order: A0, A1, B0, B1, C0, C1, D0, D1.

EXAMPLE 2

In this example, the interrupt request levels are set to different values to illustrate that the interrupt-request-level prioritization take precedence over the slot prioritization.

- Write 0x0011 to Interrupt Level register A
- Write 0x0022 to Interrupt Level register B
- Write 0x0032 to Interrupt Level register C
- Write 0x0054 to Interrupt Level register D

For this configuration, if all ten sources assert an interrupt request at the same time, they will be serviced in the following order: D1, D0, C1, B0, B1, C0, A0, A1.



NOTE: It is not normally possible to assert simultaneous interrupt requests, except via special IP Modules like the Curtiss-Wright Controls TESTIP.

B.6.3 Memory Space Example

Certain IP Modules utilize or require memory space for their operation. Examples of such devices include non-volatile storage boards, MIL-STD 1553 IP Modules, or IP Modules which simply require additional address space. The VMESC4M can be configured to allow accesses to the memory space of installed IP Modules.

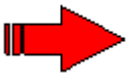
EXAMPLE 1

In this example, the VMESC4M is programmed to use VME A24 base address location 0x800000 for IP Module memory accesses. This is accomplished by writing the appropriate values to the Memory Base register (table 4-6) and the Memory Enable register (table 4-8).

- Write 0x0080 to the A24 Memory Base register
- Write 0x0001 to the Memory Enable register

By writing a 0x0080 to the A24 Memory Base register, the user is selecting a specific VMESC4M carrier memory address. Also, by enabling memory a read or write to a specific VME address is possible. A read or write to 0x800000 through 0xBFFFFF will initiate an access to the appropriate memory location of an IP Module on this carrier. In this example, locations 0x800000 through 0x8FFFFF will be used for IP Module slot 'A', locations 0x900000 through 0x9FFFFF will be used for IP Module slot 'B', for IP Module 'C', locations 0xA00000 through 0xAFFFFF will be used, and for IP Module 'D' locations 0xB00000 through 0xBFFFFF will be used. Access within each one of these 1MB regions will initiate an access to the appropriate memory location of the relevant IP Module. Considering that the VMESC4M is byte-addressed while the IP Modules are word addressed, the memory address used for the IP Module is half of the difference between the address used and the actual address corresponding to the base of the IP Modules memory region.

For instance, a read from 0x808210 will result in a read to IP Module A address 0x4108, as the IP Modules are word addressed. 0x808210 - 0x800000 = 0x8210 0x8210 / 2 =0x4108.



NOTE: Addresses are A24 VME addresses and not necessarily host addresses. The A24 address space may be mapped into a different location for the host CPU. In addition, the full range of this address space may not be supported on your host computer. See the documentation for your VME Single Board Computer for more details.

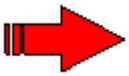
EXAMPLE 2

In this example, the VMESC4M is programmed to use VME A32 base address location 0xFE000000 for IP Module memory accesses. This is accomplished by writing the appropriate values to the Memory Base register (table 4-7) and the Memory Enable register (table 4-8).

- Write 0x00FE to the A32 Memory Base register
- Write 0x0003 to the Memory Enable register

By writing a 0x00FE to the A32 Memory Base register the user is selecting a specific VMESC4M carrier memory address. Also, by enabling memory and space select a read or write to a specific VME address is possible. A read or write to 0xFE000000 through 0xFFFFFFFF will initiate an access to the appropriate memory location of an IP Module on this carrier. In this example, locations 0xFE000000 through 0xFE7FFFFF will be used for IP Module slot 'A', locations 0xFE800000' through 0xFEFFFFFF will be used for IP Module slot 'B', for IP Module 'C', locations 0xFF000000 through 0xFF7FFFFF will be used, and for IP Module 'D' locations 0xFF800000 through 0xFFFFFFFF will be used. Access within each one of these 8MB regions will initiate an access to the appropriate memory location of the relevant IP Module. Considering that the VMESC4M is byte-addressed while the IP Modules are word addressed, the memory address used for the IP Module is half of the difference between the address used and the actual address corresponding to the base of the IP Modules memory region.

For instance, a read from 0xFE008210 will result in a read to IP Module A address 0x4108, as the IP Modules are word addressed. 0xFE008210 - 0xFE000000 = 0x8210; 0x8210 / 2 =0x4108.



NOTE: Addresses are A32 VME addresses and not necessarily host addresses. The A32 address space may be mapped into a different location for the host CPU. In addition, the full range of this address space may not be supported on your host computer. See the documentation for your VME Single Board Computer for more details.

This page intentionally left blank

APPENDIX C
PERFORMANCE

C.1 Overview

The purpose of this section is to provide several sets of empirical data that describe typical performance parameters beyond those provided in the specification. The important feature is that these are typical responses for the configuration cited, and do not supplant the maximum and minimum envelopes presented in sections 1.0 INTRODUCTION and 2.0 PRODUCT OVERVIEW.

C.2 State Timing Diagrams

The VMESC4M test configuration consisted of a typical VME card cage with a MVME-162-23 as the host CPU and four Curtiss-Wright Controls TESTIPS. State waveforms were generated using the HP 1 GHz Timing Master Module and the Curtiss-Wright Controls IP Module Logic Bus Breakout Board (IPLBB) attached to a TESTIP in slot 'A' (refer to Figure C- 1 for the performance test equipment configuration). These state waveforms are shown in Figure C- 2 through Figure C- 12. Several of the signals on the logic analyzer's screen are active low, these signals are represented by a '/' as the first character. The signals are defined and summarized on page C-2.

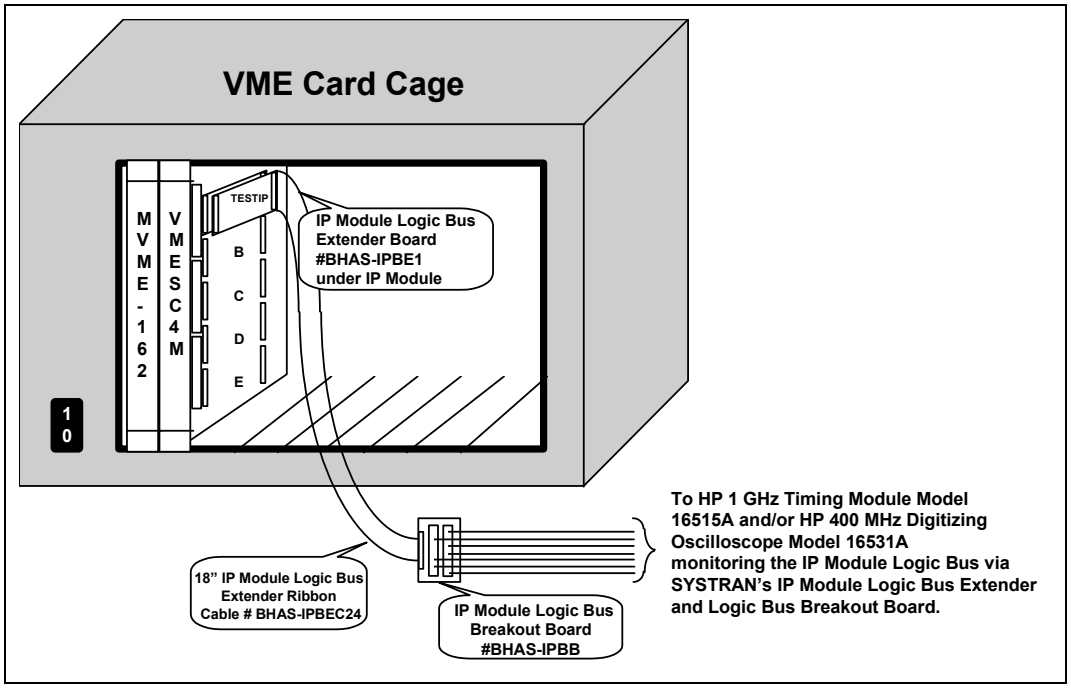


Figure C- 1 Performance Test Equipment Configuration

GLOSSARY OF SIGNALS

/ACK ----------IP Module acknowledgment (active low)
/DTACK ----------VME data transfer acknowledge (active low)
/IAKIN ----------VME interrupt acknowledge in (active low)
/IAKOT ----------VME interrupt acknowledge out (active low)
ICLK----------IP Module 8 Mhz clock
/IDSEL ----------IP Module ID select signal (active low)
/INTRQ----------IP Module interrupt request signal 0 (active low)
/INTSL ----------IP Module interrupt select signal (active low)
IPD1 ----------IP Module data bit 1
/IOSEL ----------IP Module I/O select signal (active low)
/MEMSL ----------IP Module Memory select signal (active low)
SYSCLK ----------VME 16 Mhz system clock
VMEA1 ----------VME address bit 1
VMED0 ----------VME data bit 0
VMED1 ----------VME data bit 1
/VMEAS ----------VME address strobe (active low)
/VMEDS ----------VME data strobe 0 (active low)
/VIACK ----------VME interrupt acknowledge (active low)
/VMEIR ----------VME interrupt request 1(active low)
VMER/W ----------VME write signal (low is write, high is read)

C.2.1 ID Read Cycle

Figure C- 2 is a complete ID read cycle of the TESTIP on the VMESC4M in slot ‘A’. On the left side of the figure the host CPU initiates a VME read cycle by asserting a valid address (VMEA1). A valid address is then followed by an assertion of the VME address and data strobe (/VMEAS and /VMEDS) signals. After the host CPU has begun a valid read to the ID space of IP Module A, within one ICLK period the VMESC4M begins an IP Module read cycle by asserting the /IDSEL signal.

In the very next ICLK period, the TESTIP begins driving the data bus (/IPD1) and asserts the /ACK signal to complete the IP Module read cycle. Toward the right side of the figure, the VMESC4M latches the data from the TESTIP on the rising edge of ICLK (at the end of the /ACK assertion) and drives the VME data bus. About 1 to 4 nanoseconds after the VME data bus is driven, the VMESC4M asserts the VME data acknowledge (/DTACK) signal and waits for the VME host CPU to terminate the VME read cycle by de-asserting the address and data strobe signals. After the VME host CPU terminates the VME read cycle, the VMESC4M releases the data bus and de-asserts the /DTACK signal. The entire VME read cycle is complete in four IP Module (8 MHz) clock cycles (which is 500 ns). The VMESC4M’s worst- and best-case access times are plus or minus one 16 MHz VME SYSCLOCK (± 62 ns) from the VMESC4M’s average 500 ns.

Figure C- 2 ID Read (Snapshot taken at 50 ns/Div)

C.2.2 I/O Read Cycle

Figure C- 3 a complete I/O read cycle of the TESTIP on the VMESC4M in slot ‘A’, which is functionally the same as the ID read cycle. On the left side of the figure, the host CPU initiates a VME read cycle by asserting a valid address (VMEA1). A valid address is then followed by the assertion of the VME address and data strobe (/VMEAS and /VMEDS) signals. After the host CPU has asserted a valid access, within one ICLK period, the VMESC4M begins an IP Module read cycle by asserting the /IOSEL signal.

In the very next ICLK period, the TESTIP begins driving the data bus (/IPD1) and asserts the /ACK signal to complete the IP Module read cycle. Toward the right side of the figure, the VMESC4M latches the data from the TESTIP on the rising edge of ICLK (at the end of the /ACK assertion) and drives the VME data bus. About 1 to 4 nanoseconds after the VME data bus is driven, the VMESC4M asserts the VME data acknowledge (/DTACK) signal and waits for the VME host CPU to terminate the VME read cycle by de-asserting the address and data strobe signals. After the VME host CPU terminates the VME read cycle, the VMESC4M releases the data bus and de-asserts the /DTACK signal. The entire VME read cycle is complete in four IP Module (8 MHz) clock cycles (which is 500 ns). The VMESC4M worst- and best-case access times are plus or minus one 16 MHz VME SYSCLOCK (± 62 ns) from the VMESC4M’s average 500 ns.

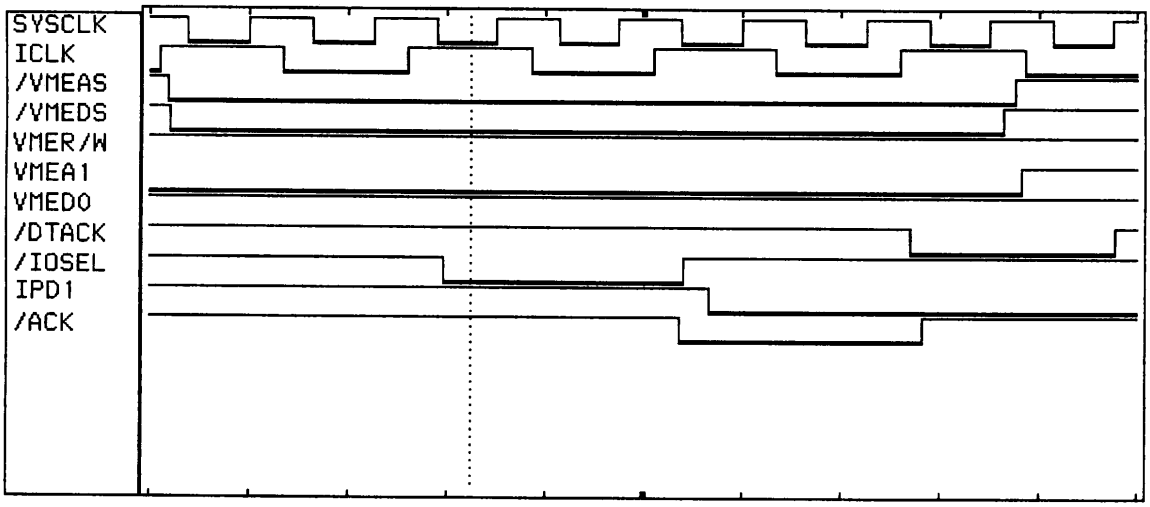


Figure C- 3 I/O Read (Snapshot taken at 50 ns/Div)

C.2.3 I/O Write Cycle

Figure C- 4 is a complete I/O write cycle to the TESTIP on the VMESC4M in slot ‘A’. On the left side of the figure, the host CPU initiates a VME write cycle by asserting a valid address and drives the VME data bus with valid data. This occurs only while the write signal (VMER/W) is low. This is followed by the assertion of the address and data strobes (/VMEAS and /VMEDS respectively). After the host CPU has asserted a valid access, within one ICLK period, the VMESC4M begins an IP Module write cycle by asserting the /IOSEL signal.

In the very next ICLK period, the TESTIP asserts the /ACK signal to complete the IP Module write cycle and latches the data from the VME bus on the rising edge of ICLK (at the end of the /ACK assertion). About 1 to 4 nanoseconds after the TESTIP latched the data, the VMESC4M asserts the VME data acknowledge (/DTACK) signal and waits for the VME host CPU to terminate the VME write cycle by de-asserting the address and data strobe signals. After the host CPU terminates the VME write cycle the VMESC4M de-asserts the /DTACK signal ending the write cycle. In this instance, the entire VME write cycle is completed in approximately four IP Module (8 MHz) clock cycles (which is 500 ns). The VMESC4M worst and best case access times are plus or minus one 16 MHz VME SYSCLOCK (± 62 ns) from the VMESC4M’s average 500 ns. Although not shown, the ID WRITE cycle is functionally the same as the I/O WRITE cycle. The only difference is that the ID_SEL is asserted rather than the IO_SEL line.

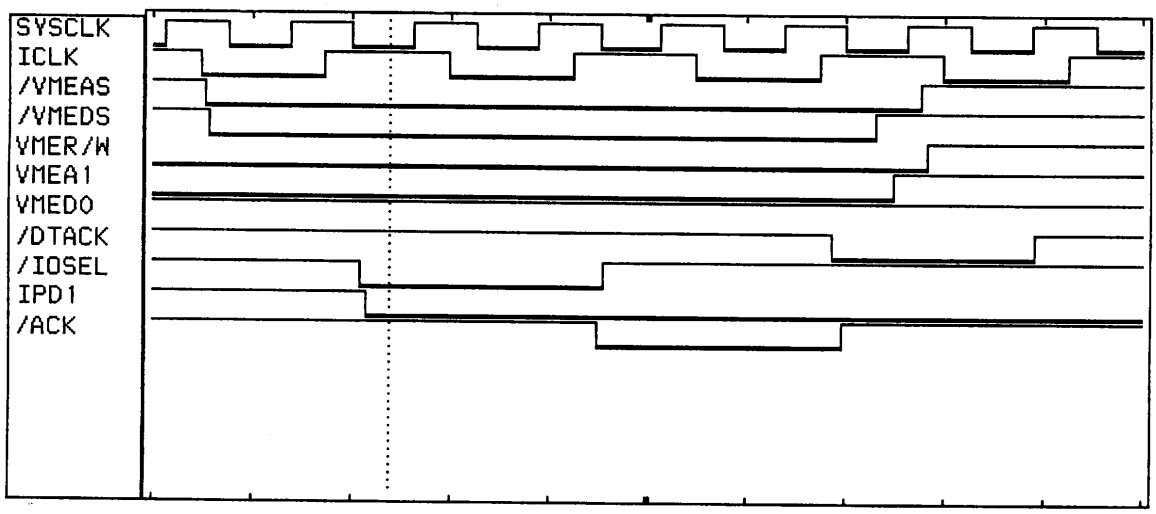


Figure C- 4 I/O Write (Snapshot taken at 50 ns/Div)

C.2.4 Memory Read Cycle

Figure C- 5 is a complete memory read cycle of the TESTIP on the VMESC4M in slot ‘A’. The host CPU initiates a read cycle by driving the read (VMER/W) signal high while also driving a valid address (VMEA1). The host CPU follows a valid address by asserting its address and data strobes (/VMEAS and /VMEDS). Within one ICLK period, the VMESC4M begins an IP Module read cycle by asserting the /MEMSL signal.

In a memory read cycle, the IP Module data bus is used to drive the upper IP Module address lines (A22:A7). This “extended” address is valid during the select state of the IP Module transfer cycle. In the very next ICLK period, the TESTIP begins driving the data bus (/IPD1) and acknowledges the read and completes the IP Module read cycle by asserting the /ACK signal. On the right side of the figure, the VMESC4M latches the valid data from the TESTIP on the rising edge of ICLK (toward the end of the /ACK signal’s assertion) and begins to drive the VME data bus. About 1 to 4 ns after the VME data bus is driven, the VMESC4M asserts the VME data acknowledge (/DTACK) signal and waits for the VME host CPU to terminate the VME read cycle by de-asserting the address and data strobe signals. After the VME host CPU terminates the VME read cycle, the VMESC4M releases the data bus and de-asserts the /DTACK signal. The entire VME read cycle is complete in four IP Module (8 MHz) clock cycles (which is 500 ns). The VMESC4M’s worst and best case access times are plus or minus one 16 MHz VME SYSCLOCK (± 62 ns) from the VMESC4M’s average 500 ns.

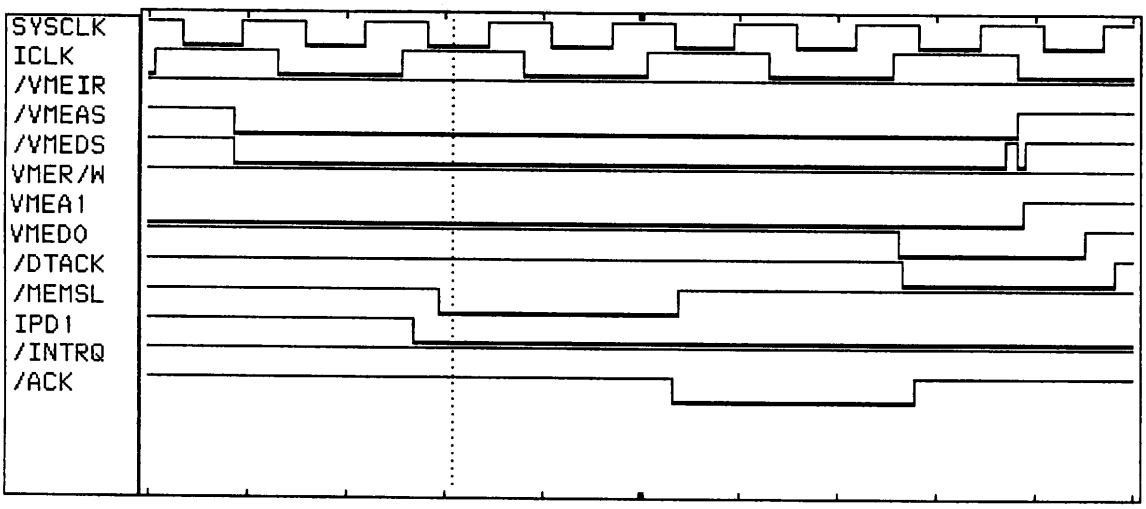


Figure C- 5 Memory Read (Snapshot taken at 50 ns/Div)

C.2.5 Memory Write Cycle

Figure C- 6 is a complete memory write cycle to the TESTIP on the VMESC4M in slot ‘A’ . On the left side of the figure, the host CPU initiates a VME write cycle by driving the ‘VMER/W’ signal low, while also asserting a valid address and data. The valid address and data is followed by the active low assertion of the VME address and data strobes (/VMEAS and /VMEDS) signals. After the host CPU has begun a valid access, within one ICLK period, the VMESC4M begins an IP Module write cycle by asserting the /MEMSL signal.

In the very next ICLK period, the TESTIP asserts the /ACK signal to complete the IP module write cycle and latches the data from the VME bus. About 1 to 4 nanoseconds after the TESTIP latched the data, the VMESC4M asserts the VME data acknowledge (/DTACK) signal and waits for the VME host CPU to terminate the VME write cycle by de-asserting the address and data strobe signals. Finally, upon the de-assertion of the address and data strobe the VME host CPU terminates the VME write cycle and the VMESC4M de-asserts the /DTACK signal. In this instance the entire VME write cycle is completed in approximately four IP Module (8 MHz) clock cycles (which is 500 ns). The VMESC4M worst and best case access times are plus or minus one 16 MHz VME SYSCLOCK (± 62 ns) from the VMESC4M’s average 500 ns.

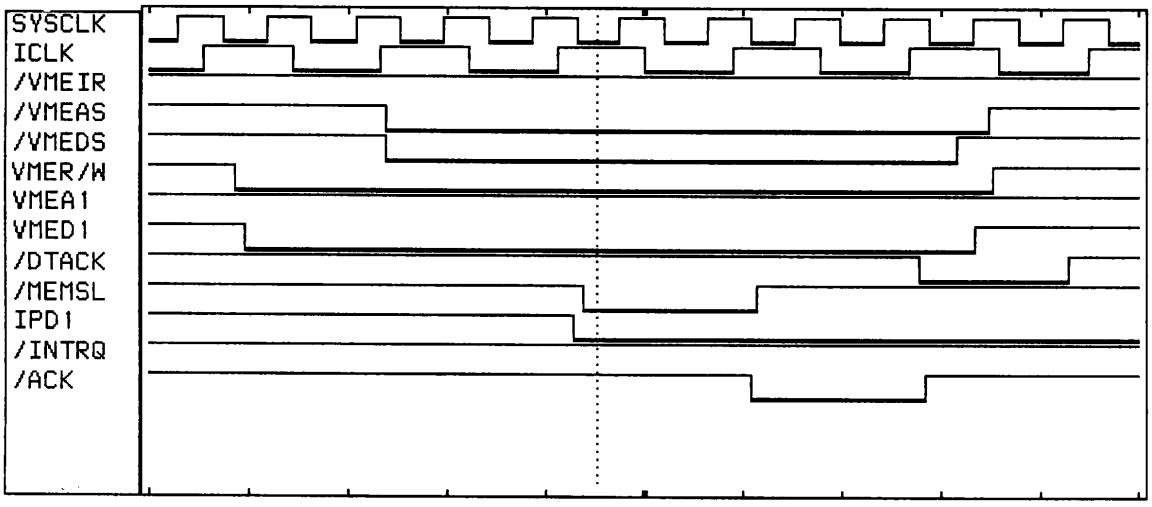


Figure C- 6 Memory Write (Snapshot taken at 70 ns/Div)

C.2.6 I/O Read-Modify-Write Cycle

Figure C- 1 is a complete I/O read-modify-write cycle to the TESTIP on the VMESC4M in slot 'A'. On the left side of the figure, the host CPU initiated a VME read cycle by asserting a valid address and then asserting the VME address and data strobe (/VMEAS and /VMEDS) signals. The read cycle completes when the /DTACK is toggled. In the center of the figure, the VME host CPU de-asserts the data strobe signal while keeping the address strobe asserted. A few clock cycles later, the host CPU begins a VME write cycle by asserting a valid address and data line along with the VME write (VMER/W) signal and then re-asserts the VME data strobe (/VMEDS) signal. On the right side of figure, the write cycle terminates normally when the address and data strobes are de-asserted, completing the VME read-modify-write cycle. Notice that the complete read-modify-write cycle was executed by the host CPU in approximately 1.8 µs.

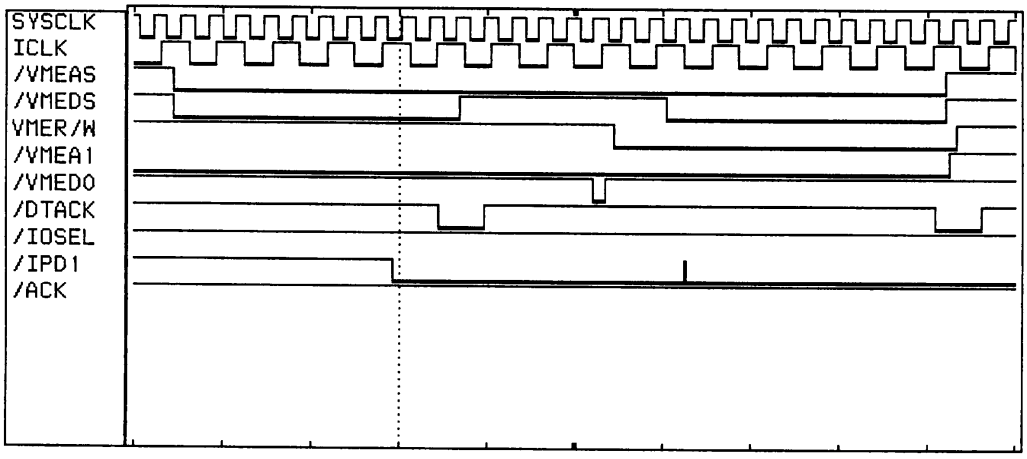


Figure C- 7 Read-Modify-Write (Snapshot taken at 200 ns/Div)

C.2.7 I/O Write Cycle With IP Module Wait States

Figure C- 8 is a complete I/O write cycle to the TESTIP on the VMESC4M in slot ‘A’. This test is similar to Figure C- 4 except that IP Module wait states are inserted. On the left side of the figure, the host CPU initiates a VME write cycle by asserting a valid address and drives the VME data bus with valid data. This occurs only while the address and data strobe (/VMEAS and /VMEDS) and the VMER/W signal are asserted low. Next, the VMESC4M begins an IP Module write cycle by asserting the /IOSEL signal. In the center of the figure, the TESTIP has inserted five IP Module wait states and then asserts the /ACK signal to complete the IP Module write cycle. Then, on the right side of the figure, the write cycle terminates normally as in Figure C- 4 completing the VME write cycle with wait states. Note that the VMESC4M supports ID, IO, memory, and Interrupt IP module cycles with inserted wait states.

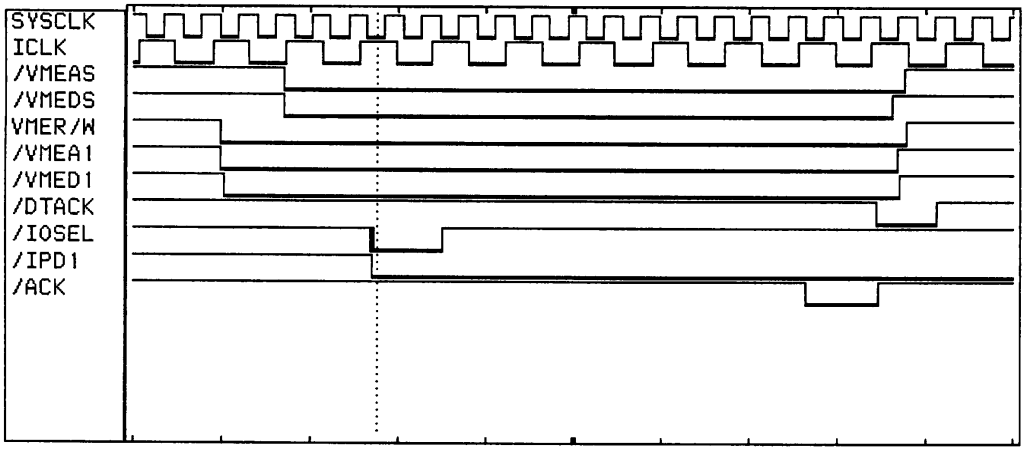


Figure C- 8 I/O Write With Wait States (Snapshot taken at 150 ns/Div)

C.2.8 Interrupt Request

Figure C- 9 is a waveform view of a complete interrupt request from the TESTIP and the following interrupt service routine (ISR) from the host CPU. For this test, the VMESC4M was configured to drive the VME interrupt level one request (/VMEIR) when the TESTIP asserted the IP Module interrupt request signal zero (/INTRQ). First, the TESTIP asserts an interrupt request and in response to this action, the VMESC4M asserts the VME interrupt request to the host CPU. Then about 1.5 µs later, the host CPU begins an interrupt acknowledge cycle. This interrupt acknowledge cycle fetches the interrupt vector from the TESTIP and services the interrupt. In the center of the figure, the host CPU is reading a status register on the TESTIP. The host CPU then performs a write to the TESTIP to clear the interrupt request. In response to the de-assertion of the IP Module interrupt request, the VMESC4M de-asserts the VME interrupt request. Figure C- 10 and 5-11 show more detail timing data of the VMESC4M interrupt cycles.

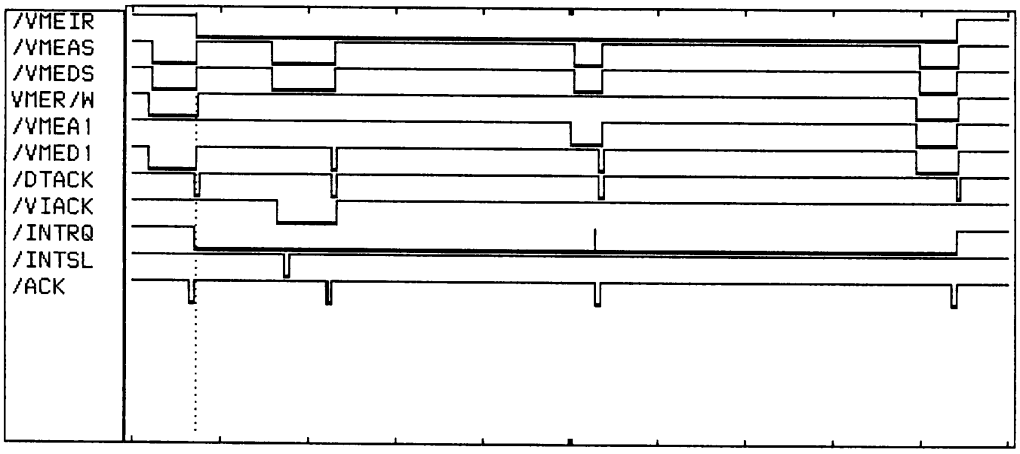


Figure C- 9 Complete Interrupt Request and Service Routine (Snapshot taken at 2.00 µs/Div)

Figure C- 10 is a zoomed-in view of the TESTIP driving its interrupt request (/INTRQ) signal and in turn the VMESC4M driving the VME interrupt request signal (/VMEIR) in about 20 ns.

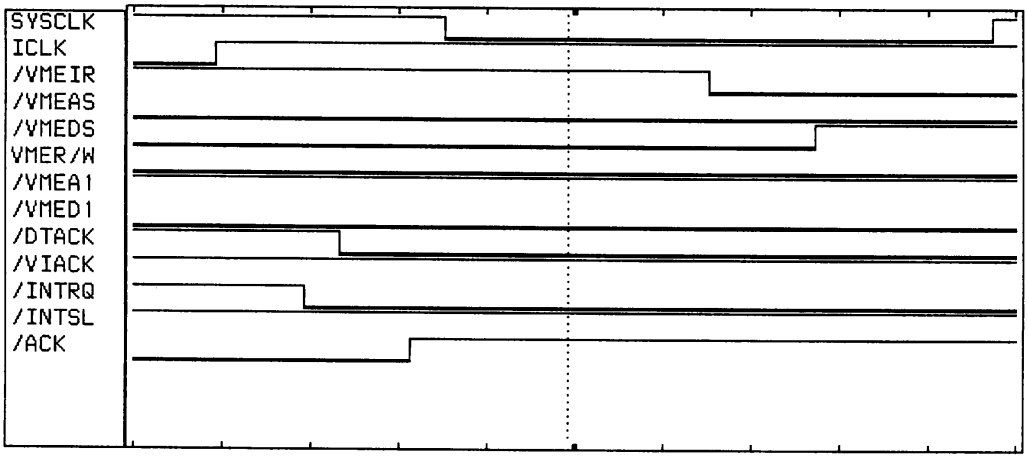


Figure C- 10 IP Module INTRQ0 Driving VME IRQ1 (Snapshot taken at 5 ns/Div)

Figure C- 11 is a zoomed in view of the TESTIP placing an interrupt vector onto the VME data bus. This interrupt vector cycle functions the same as the ID and IO read cycles. On the left side of the figure, the host CPU initiates an interrupt acknowledge cycle by asserting a valid address. The valid address (presented on VMEA3:1) determines which of the interrupt requests is to be serviced. After the address line is valid, the VME address strobe, VME data strobe, and interrupt acknowledge input signals (/VMEAS, /VMEDS, and VIACK) are asserted. Next, in the center of the figure, the interrupt select (/INTSL) signal is asserted. This denotes the beginning of an IP module interrupt cycle. This can only happen after the VMESC4M's interrupt engine determines if the current interrupt acknowledge cycle is for that particular IP.

The VMESC4M always begins an IP Module select cycle within one ICLK cycle of detecting a valid VME access. In the very next ICLK period, the TESTIP begins driving the data bus and asserts the /ACK signal to complete the IP Module interrupt acknowledge cycle. On the right side of the figure, the VMESC4M latches the data from the TESTIP on the rising edge of ICLK (at the end of the terminate cycle) and drives the VME data bus. About 1 to 4 ns after the VME data bus is driven, the VMESC4M asserts the VME data acknowledge (/DTACK) signal and waits for the VME host CPU to terminate the VME interrupt acknowledge cycle by de-asserting the address strobes, data strobes, and IACKIN signals. After the VME host CPU terminates the VME interrupt acknowledge cycle, the VMESC4M releases the data bus and de-asserts the /DTACK signal. The entire VME interrupt acknowledge cycle is complete in about four IP Module (8 MHz) clock cycles (which is 500 ns). Again the VMESC4M worst and best case access times are plus or minus one 16 MHz VME SYSCLOCK (± 62 ns) from the VMESC4M's average 500 ns.

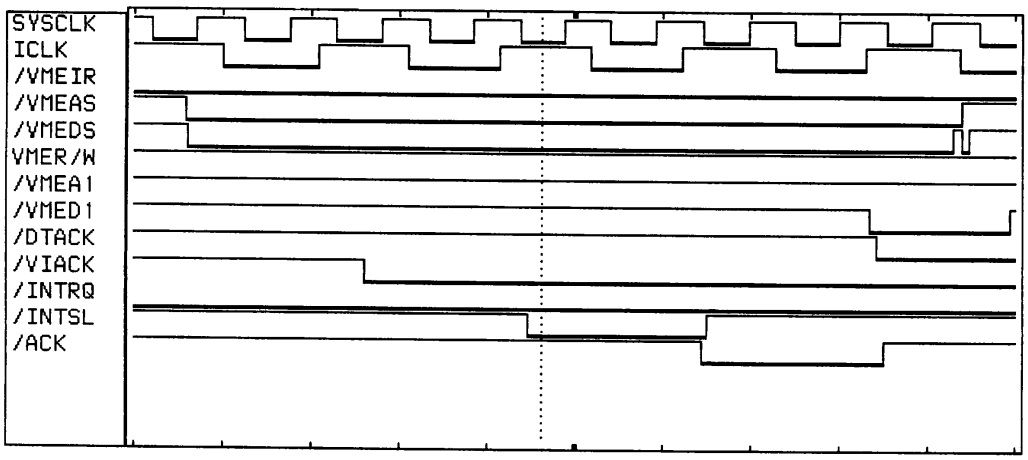


Figure C- 11 IP Module Interrupt Vector Cycle (Snapshot taken at 60 ns/Div)

Figure C- 12 is a zoomed in view of the VMESC4M driving the VME IACKOUT (connected to /IAKIN on the next VME board) signal in about 36 ns. In this figure, the VMESC4M interrupt engine determined that this interrupt acknowledge cycle was not for any of its IP Modules and very quickly passed on the /IAKOT signal in the VME daisy-chain.

NOTE: Figure C- 12 is very important information for users concerned about system response time with the VME IACKOUT daisy-chain signal being passed along by the VMESC4M in about 36 ns. This means that the VMESC4M will not cause a VME system time-out or bus error by driving the IACKOUT (/IAKOT) signal as fast as possible. If a VME card cage had one host CPU (VME bus master) and twenty VMESC4M IP Module carriers installed, the VMESC4M in the slot furthest from the host CPU with an IP Module in slot 'D' will see the IACKIN (/IAKIN) signal in about .72 µs.

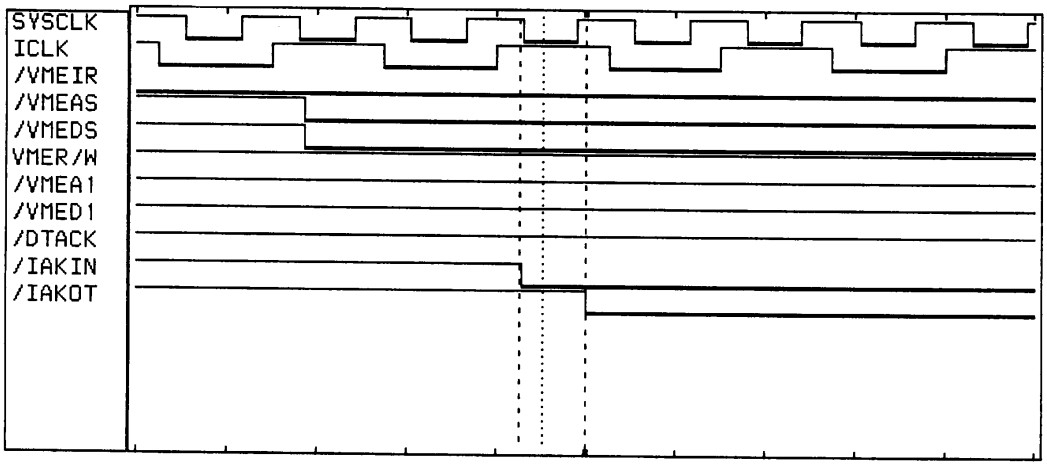


Figure C- 12 VME Signals IACKIN to IACKOUT (Snapshot taken at 50 ns/Div)

Time between /IACKIN to /IACKOUT is 36 ns.

CURTISS WRIGHT Controls, Inc. Embedded Computing

APPENDIX D
TYPICAL APPLICATIONS

D.1 Applications

Curtiss-Wright Controls, Inc. extends an open invitation to all users to freely submit their applications that might, or do, use the VMESC4M Slave Carrier to solve a problem. This section of the manual will be revised periodically to include new application ideas for all users to consider. Help advance the level of technology by participating with the Curtiss-Wright Controls team, while simultaneously publishing your ideas.

Submission constitutes permission to publish without additional consent or compensation, and Curtiss-Wright Controls, Inc. reserves the right to modify submissions to provide for more generic appeal, when necessary.

D.2 Sharing IP Modules Between Multiple VME Masters

It is sometimes desirable for multiple VME bus masters to share IP Module resources. In this case, a semaphore is typically used to lock out access to the resource when it is in use. This example illustrates using the General Purpose registers on the VMESC4M for this purpose.

Figure D- 1 depicts a system that is comprised of two VME masters, and one or more VMESC4M carriers populated with I/O modules, in a process monitor and control application. In this hypothetical system, one master performs the process control functions while the second master monitors all of the I/O data values and reports the information to a remote computer. The second master is presumably used because one master cannot handle both the process control and communication functions.

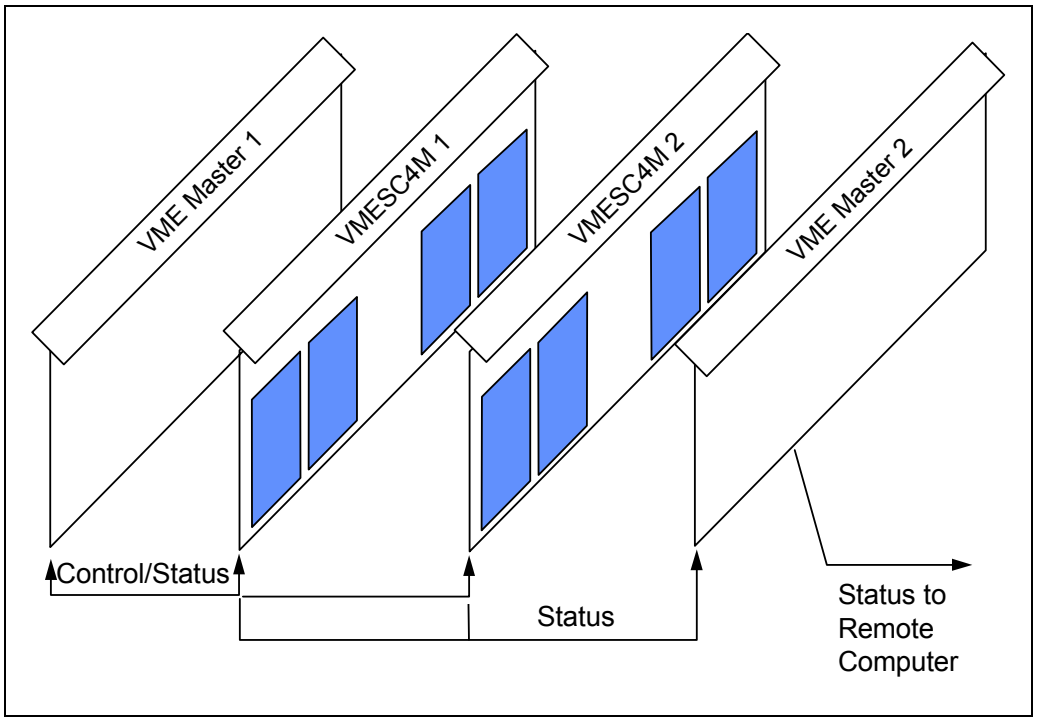


Figure D- 1 A System With Two VME Masters

The data being reported to the remote computer from any one IP Module must be from the same frame. To insure this requirement is met, a separate semaphore is used for each IP Module. The exchange of data between the VME masters and the IP Modules would then go something like this, where "x" is the last IP Module on the last carrier:

D.2.1 VME Master 1

- Reserve IP Module A semaphore (in General Purpose register A). If not available then wait until it is available.
- Set the new output values for IP Module A.
- Release IP Module A semaphore.
- Reserve IP Module x semaphore (in corresponding General Purpose register). If not available then wait until it is available.
- Set the new output values for IP Module x.
- Release IP Module x semaphore.

D.2.2 VME Master 2

- Reserve IP Module A semaphore (in General Purpose register A). If not available then wait until it is available.
- Read the data from IP Module A.
- Release IP Module A semaphore.
- Send the IP Module A data to the remote computer.
- Reserve IP Module x semaphore (in corresponding General Purpose register). If not available then wait until it is available.
- Read the data from IP Module x.
- Release IP Module x semaphore.
- Send the IP Module x data to the remote computer.

This is just one example of using the General Purpose registers. They are completely user definable and therefore can be used for any purpose.

D.3 Configuring Interrupts for Multiple Priority Schemes

Some systems must run more than one application for a given hardware configuration. The two applications may require two distinct interrupt priority schemes. This example illustrates how to configure the interrupt level registers on the VMESC4M to achieve two different priority schemes.

The following assumptions are used for this example:

1. The two applications each use four IP Modules, A through D.
2. Each IP Module utilizes interrupts IRQ0 and IRQ1.
3. Application one requires that the interrupts be serviced in the following order:
   C0, C1, B0, B1, A0, A1, D1, D0.
4. Application two requires that the interrupts be serviced in the following order:
   D1, D0, C0, C1, A0, A1, B0, B1.
5. Interrupt level 7 should not be used.

Recall that the IACK daisy chain slot priority is:
A0, A1, B0, B1, C0, C1, D0, D1
which represents the servicing order for equal level interrupt requests. Therefore, the solution is to set interrupt request levels to override the slot priority, where necessary, to achieve the desired servicing order.

To achieve the desired servicing order for application one, the interrupt levels could be set as follows:

- Write 0x0033 to Interrupt Level register A
- Write 0x0044 to Interrupt Level register B
- Write 0x0055 to Interrupt Level register C
- Write 0x0022 to Interrupt Level register D

To achieve the desired servicing order for application two, the interrupt levels could be set as follows:

- Write 0x0022 to Interrupt Level register A
- Write 0x0022 to Interrupt Level register B
- Write 0x0033 to Interrupt Level register C
- Write 0x0054 to Interrupt Level register D

In both cases, there is more than one combination of Interrupt Level register settings to produce the desired results.

This page intentionally left blank

GLOSSARY

[x:y] ---------- Nomenclature designating a bit-range, where “x” is the left-most bit and “y” is the right-most bit. (e.g. Data bus [7:0] refers to the Least Significant eight bits).

byte-lane ---------- 8 bits of a data bus on octal boundaries.

CSR ---------- Control and Status Register.

doublewide ---------- An IP Module that is twice the size of the singlewide board.

EPLD ---------- Erasable Programmable Logic Device.

ID PROM ---------- The circuitry that presents the proper data pattens to the low 8 bits of the IPDbus, with upper-byte zero fills, during the ID (read) transfers.

IP Module ---------- Business-card size mezzanine-type subsystems designed with a common digital interface known as the IP bus. These field-installable plug-and-play modules are automatically recognized by system software. An open industry standard defines the mechanical and electrical interface to the carrier board.

IP Module logic bus ---------- A synchronous, 4 MTransfers/sec, 16-bit wide bus that includes I/O, memory, ID PROM, and interrupts. The address bus is 6 bits wide (22 bits wide in memory mode).

IPDbus ---------- IP Module Data Bus.

ISR ---------- Interrupt Service Routine

MTBF ---------- Mean Time Between Failures.

ns, µs, ms ---------- Nanoseconds, microseconds, and milliseconds respectively.

singlewide ---------- An IP Module printed circuit board (3.9" by 1.8"). Each module has two 50-pin connectors.

VHDL ---------- Very high speed integrated circuit Hardware Description Language.

This page intentionally left blank

INDEX

This page intentionally left blank