VMIVME-1183
32-Channel P2 Digital Input Board with Change-of-State Interrupts, Sequence-of-Events and Built-in-Test

Product Manual

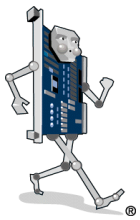
A GE Fanuc Company

12090 South Memorial Parkway
Huntsville, Alabama 35803-3308, USA
(256) 880-0444 ◆ (800) 322-3616 ◆ Fax: (256) 882-0859

500-001183-000 Rev. A

12090 South Memorial Parkway
Huntsville, Alabama 35803-3308, USA
(256) 880-0444 ◆ (800) 322-3616 ◆ Fax: (256) 882-0859

COPYRIGHT AND TRADEMARKS



(I/O man figure)

(IOWorks man figure)

12090 South Memorial Parkway
Huntsville, Alabama 35803-3308, USA
(256) 880-0444 ◆ (800) 322-3616 ◆ Fax: (256) 882-0859

Table of Contents

List of Figures

List of Tables

Overview

Contents

Introduction

Features

The VMIVME-1183 is a 32-channel P2 input board with Change-of-State (COS) interrupt capabilities. The interrupt control logic can be programmed to issue an interrupt upon specific state changes. The user selects the state change to use by programming the Control and Status Register (CSR) and the COS Select register for the desired activity. This board will store up to 512 state changes as long as the COS logic is enabled. This will prevent the board from losing a state change during interrupt servicing. The VMIVME-1183 supports Built-In-Test (BIT). BIT registers are available to test the active components of the input data paths on the board. The board also supports byte, word and longword data transfers during basic input data operations.

The following is a list of some of the features of the VMIVME-1183:

- 32 channels of high voltage digital inputs
- Eight inputs are grouped together and called an input port. Each port can be configured by the user to monitor one of the following input circuits:
  - — Contact closure
  - — Current sinking
  - — Voltage sourcing
- User-configurable input voltage thresholds
- Incoming data is available for direct VMEbus read cycles
- Data can be read using 8-, 16-, or 32-bit data transfers

Features (Continued)

- Each channel can be independently set up by the user to perform the following functions:
  - — No interrupts (data only)
  - — Rising edge only interrupts
  - — Falling edge only interrupts
  - — Any edge interrupts
- Sequence of Events logging with or without optional time stamping (See “Quadrature Counter Feature” on page 64.)
- Differential or Single-Ended inputs for Quadrature Counter, with marker signal input and dasiy-chaining capability
- The test registers can be activated at any time to test the integrity of the active components on the board
- This board complies with the VMEbus Specification Rev. C.1
- The board can be jumpered to respond to short I/O accesses or to standard data I/O accesses

Functional Description

The VMIVME-1183 is a 32-channel digital input board. The input circuitry is user configured by positioning SIP resistors and jumpers. This is explained in detail in Chapter 2 of this manual. The inputs can be set up to monitor either voltage sourcing or current sinking external circuits. Contacts are handled like switching transistors. Once the hardware has been configured and the board is installed in a system, you can start monitoring the external circuits. The data ports are always available for VMEbus data transfers. Simply read the address of the port you want and record the state of the external circuits. This data is available in byte, word or longword accesses.

The board can be programmed to interrupt the system based on a change in the inputs. Change-of-State (COS) monitors the logical state of the inputs. When any bit in an input channel changes its state (for example going from a logic one (1) to a logic zero (0)), and the COS logic is set to trigger on just such an event, then an interrupt will be issued to the host CPU. The COS logic can be told (via the Control and Status Register) to issue interrupts upon certain conditions, such as the falling edge described above, and is bit-specific. The following list states the types of state changes the board will respond to:

- No interrupts (processed as data only)
- Rising edge only
- Falling edge only
- Any edge (or state change)

The VMIVME-1183 stores state changes in on-board FIFO memory. The board can be configured to clear an interrupt request when the FIFO is empty, or an interrupt has been acknowledged (ROAK). This way, if a COS occurs while a previous COS is waiting for service, or during the interrupt service routine, it is saved and will issue its interrupt request as soon as the interrupts are re-enabled.

Additional capabilities are to record Sequence of Events (SOE). In this mode, when a specified change of state occurs on any of the inputs, the previous state of the inputs are stored in the FIFO, followed by the current state of those inputs. Additionally, a counter value from a front panel input quadrature counter (single-ended, or differential) is stored in a separate FIFO to give the SOE an optional time stamp. The quadrature counter feature is available in COS mode as well, See “Quadrature Counter Feature” on page 64.

Test registers are provided to periodically check the integrity of the VMIVME-1183. The registers are mapped into the same address locations as the Input Data registers. By enabling Test Mode in the Control and Status Register 1 (CSRCSR), and then writing data to these registers, you can check the active components of the input data paths on the board and assure yourself of the board's performance. The test register data will not affect the external circuitry. It only overrides the incoming data. When you disable Test Mode, the test data no longer affects the input data.

Reference Material List

For a detailed explanation of the VMEbus and its characteristics, refer to "The VMEbus Specification" available from:

VITA

VMEbus International Trade Association

7825 East Gelding Dr., No. 104

Scottsdale, AZ 85260

(602) 951-8866

FAX: (602) 951-0720

www.vita.com

Physical Description and Specifications: Refer to Product Specification, 800-001183-000 available from:

VMIC

12090 South Memorial Pkwy.

Huntsville, AL 35803-3308, USA

(256) 880-0444

(800) 322-3616

Fax: (256) 882-0859

www.vmic.com

The following Application and Configuration Guides are available from VMIC to assist in the selection, specification, and implementation of systems based upon VMIC's products:

Title	Document No.
Digital Input Board Application Guide	825-000000-000
Change-of-State Application Guide	825-000000-002
Digital I/O (with Built-in-Test) Product Line Description	825-000000-003
Synchro/Resolver (Built-in-Test) Subsystem Configuration Guide	825-000000-004
Analog I/O Products (with Built-in-Test) Configuration Guide	825-000000-005
Connector and I/O Cable Application Guide	825-000000-006

Safety Summary

The following general safety precautions must be observed during all phases of the operation, service, and repair of this product. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture and intended use of this product.

VMIC assumes no liability for the customer's failure to comply with these requirements.

Ground the System

To minimize shock hazard, the chassis and system cabinet must be connected to an electrical ground. A three-conductor AC power cable should be used. 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.

Do Not Operate in an Explosive Atmosphere

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

Keep Away from Live Circuits

Operating personnel must not remove product covers. Component replacement and internal adjustments must be made by qualified maintenance personnel. 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.

Do Not Substitute Parts or Modify System

Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification to the product. Return the product to VMIC for service and repair to ensure that safety features are maintained.

Dangerous Procedure Warnings

Warnings, such as the example below, precede only potentially dangerous procedures throughout this manual. Instructions contained in the warnings must be followed.

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

Safety Symbols Used in This Manual

STOP: This symbol informs the operator the that a practice or procedure should not be performed. Actions could result in injury or death to personnel, or could result in damage to or destruction of part or all of the system.

WARNING: This sign denotes a hazard. It calls attention to a procedure, a practice, a condition, which, if not correctly performed or adhered to, could result in injury or death to personnel.

CAUTION: This sign denotes a hazard. It calls attention to an operating procedure, a practice, or a condition, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the system.

NOTE: Calls attention to a procedure, a practice, a condition or the like, which is essential to highlight.

CHAPTER
1

Theory of Operation

Contents

Overview

The board functions are broken down into seven major blocks as shown in Figure 1-1 below. These blocks are:

- Bus interface
- Register decoder
- Interrupt Processor (IP)
- Change-of-State (COS) logic
- Input circuits
- Built-in-Test (BIT) registers
- Quadrature counter

The bus interface contains the VMEbus interface logic, the board's address decoding logic and the data steering logic. The register decoder selects which of the data registers (BD ID, CSR, IP, BIT or input) are to be used during a data transfer. The IP interface contains the logic to control the IP and interface it with the VMEbus interrupt bus. The COS logic determines if a change-of-state has occurred, issues the interrupt request (when necessary) and controls the FIFOs data flow. The input circuits contain the hardware to configure the topology (the shape or type of circuit), and their trigger thresholds. The CSR contains the control and enables.

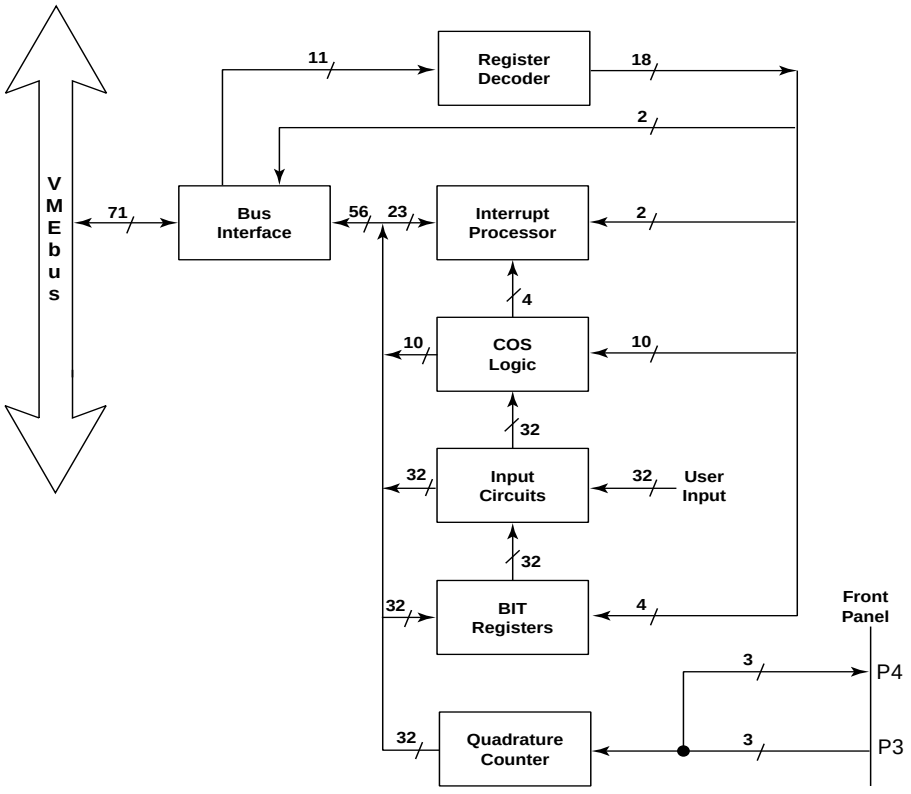


Figure 1-1 VMIVME-1183 Block Diagram

During **DATA ONLY** operations, the board monitors the external circuitry. When the board's address decoder decides the board is being accessed, it clocks the inputs into data registers. In other words, it takes a snapshot of the external world. The data is steered to the correct VMEbus data lines, DTACK* is asserted, and the board re-arms its address decoder for another cycle. These types of transfers are always available to the user. They will have no effect upon the COS or SOE logic.

During **INTERRUPT CYCLES**, the Interrupt Processor (IP) handles the board functions. When the COS logic issues an interrupt request to the IP, it also stores the incoming data in the associated FIFOs. The IP issues an interrupt request to the host CPU and waits for the proper Interrupt Acknowledge cycle. When the IP responds to the acknowledge cycle, it places the programmed vector for the interrupt service routine upon the VMEbus. The interrupt service routine reads the FIFO's data, which clears its interrupt request if Release On FIFO Empty (ROFE) is enabled.

Functional Organization

The VMIVME-1183 is divided into the following functional categories. Refer to Figure 1-1 on page 18. All of these functions are discussed in detail in this section:

- VMEbus interface
- Data transfer cycles
- Interrupt acknowledge cycles
- Register decoder
- IP interface
- Change-of-State logic
- Inputs
- Built-in-Test
- Quadrature Counter

VMEbus Interface

The bus interface logic seen in Figure 1-2 below consists of bus signal buffers and transceivers that meet the VMEbus specification loading requirements. The address decoder (or board-select logic) can respond to standard or short data I/O accesses. The steering logic selects which of the VMEbus data lines are to be connected to the board's Internal Data Bus (IDB) and which direction the data will flow. The Board Identification (BD ID) register, the Control and Status Register (CSR) and the COS Select registers are considered part of the bus interface logic.

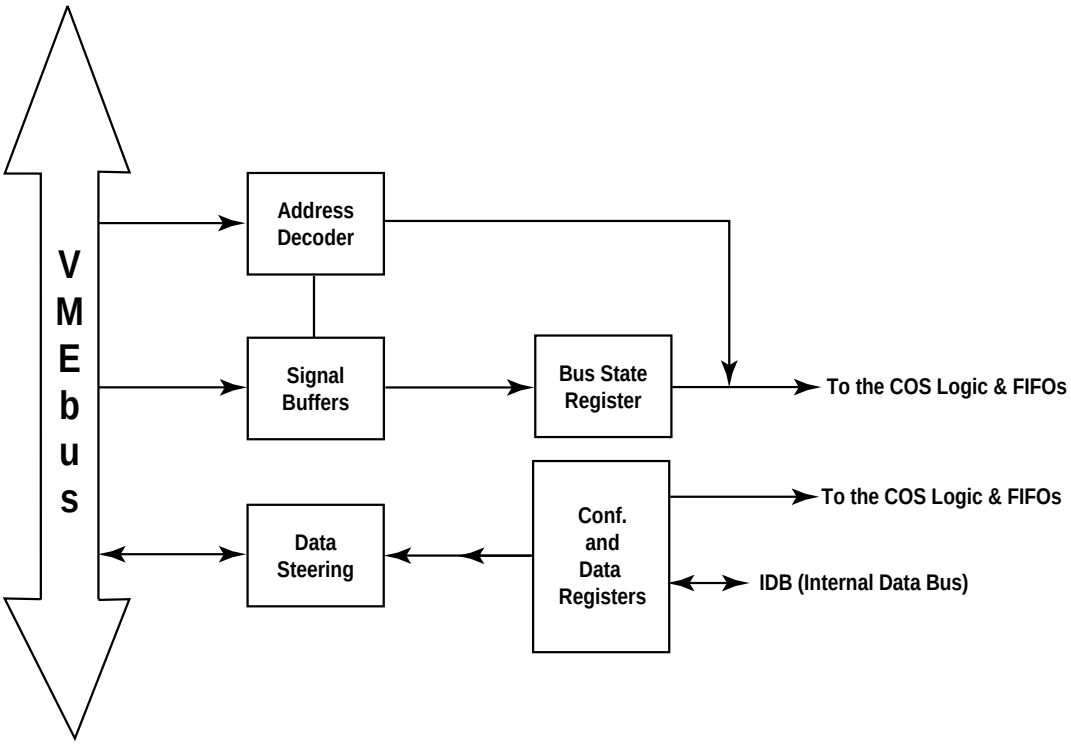


Figure 1-2 VMIVME-1183 Bus Interface

Data Transfer Cycles

The Standard(A24)/Short(A16) jumper establishes the upper address line decoding. If this jumper is installed, the board will only respond to standard A24 data accesses. If the jumper is not installed, the board will only respond to short (A16) I/O accesses. The board's factory default configuration is to respond to short (A16) I/O accesses.

The address decoder compares the 18 address lines (A06 through A23) and the six address modifier lines (AM0 through AM5) with preset conditions. The address lines, the STD(A24)/SHT(A16)CSR line and the AM2 line have jumpers that are used to establish the base address of the board. When a jumper is installed the corresponding address line is compared to a logic zero (0). When the bus address and the address modifier matches the preset address, the board will respond to the bus cycle.

The AM2 line can be jumpered to respond to a specific state (logic HIGH or LOW) or to either logic state. If the jumper is omitted, the board will respond to supervisory data accesses only. If the jumper is in the AM2 position (J53), the board will respond to nonprivileged data accesses only. However, a second jumper (J54) is available for AM2 operations. In this position (AM2 + AM2), the board will respond to either supervisory or nonprivileged accesses.

CAUTION: DO NOT install both jumpers J53 and J54. This will short the VMEbus AM2 signal line to ground and disrupt system operations.

When the decoder determines that the board is being accessed, the VMEbus state is stored in a register and the BD_SEL lines are activated. The board-select line clocks the input registers and records the state of the external circuits. The register decoder uses the bus state information to place the proper data on the IDB, and to energize the proper VMEbus interface data transceivers. The bus interface then issues DTACK* and watches the Data Strobes (DS0* and DS1*). When both Data Strobes are asserted HIGH the decoder resets itself and waits for the next matching address. The cycle then restarts. The BD ID register has a fixed value of $5D00.

CSR is used to control the basic functions of the board. The even addressed byte of CSR controls the board's test mode and the front panel Fail LED.

When test mode is active, the BIT registers are activated. Data written will overwrite the external inputs. This method allows test data to be used to determine the "health" of the board. By comparing the written data with the read data. The BIT registers occupy the same board addresses as the input data ports, making programming BIT functions easier. The Fail LED is under user control and can be turned OFF when the board is working properly. Two bits of the CSR (Bits 13 and 12) are used to monitor the COS memory device (FIFO). Bit 12 (FIFO_Full) and Bit 13 (FIFO_Empty), allow the user to monitor the current condition of the FIFO. If a FIFO is full, any new data will be lost. All bits in the CSR are available during read operations with the exception of Bit 8, 7, 6, 5 and 4 which are reserved. Bit 11 (COS/SOE) of CSR, determines the mode of operation for the board.

The BD ID, CSR, BIT, IP, FIFO and the input registers can be accessed on a byte, word or longword boundary. Please refer to the memory map in Chapter 3 of this manual for the actual relative address locations for these registers. The BD ID is at the base address of the board with the CSR stacked above it, followed by Data/BIT, the COS/SOE FIFO, then the IP control and vector registers. Followed by the longword location where the current value of the SOE quadrature counter can be read. The COS Select 0 through 3 registers are placed next. COS Select registers 0 through 3 controls the monitoring mode (data only, rising/falling/any edge) of each channel. Word and longword accesses are placed upon even address boundaries. This allows you to read the BD ID and the CSR in one longword access. You cannot read the odd byte of the BD ID and the even byte of the CSR in one word transfer.

Interrupt Acknowledge Cycles

During Interrupt Acknowledge (IACK*) cycles, only the lowest three address lines are valid. They carry the interrupt level being acknowledged by the host CPU, bypassing the address decoder and activating the IP. If the IP has an interrupt pending, it places its vector on the bus and issues a DTACK*. If not, the IP passes the IACKIN* signal to the IACKOUT* line.

If the interrupting FIFO is not empty after the interrupt service routine is read and ROAK is selected, then a new interrupt will be generated by the IP as soon as interrupts are enabled. If ROFE is selected, the interrupt will not be released until the FIFO is empty. When more than one longword of data is in the FIFO, a new interrupt occurred before the old one was read by the CPU. If the FIFO gets full, the state changes are too fast for the system to handle.

NOTE: If the COSs come faster than the CPU can service, interrupt starvation will occur, keeping the regular programs from executing. Please keep this in mind when designing your system.

Register Decoder

The register decoder, using the five lowest address lines and the data strobes, selects which of the registers is placed on the IDB. The decoder is used during any board access. It is a simple demultiplexer scheme. The address lines are decoded when the board select lines are asserted. Based upon the address lines and the data strobes, the proper register or registers are activated or clocked. A memory map showing the relative addresses for each register used by this board is listed in Chapter 3 of this manual.

Debounce Clock

The debounce clock (DBNC_CLK) is derived from the VMEbus system clock and is not intended as a synchronizing clock. The COS logic clock signal is the VMEbus clock. The debounce clock is started when a COS condition is detected. This allows for a lower latency, with the glitch-reduction capability of the debounce clock.

Change-of-State Logic

The COS logic bases its action upon the state of its select lines COS_SEL_x_A and COS_SEL_x_B (A and B for short). These lines are controlled by the COS register values. The states are listed in Table 1-1 below:

Table 1-1 COS States

SEL B	SEL A	COS Logic's Action
0	0	COS, passing data only
0	1	COS on rising edges only
1	0	COS on falling edges only
1	1	COS on any edge

The COS logic writes data into a FIFO based on the board configuration and the state stored in the new and old data registers. The COS logic decides when to load data into the FIFO. When a channel (bit) is used as data only (no interrupts), the data changes are not clocked into the FIFO. During the interrupt routine, as a COS is occurring, the logic continues to load the new state into the FIFO through the old data registers. The FIFO stores the new state until the states are read by the host. Until then, the FIFO continues to issue interrupt requests to the IP.

Previous State Data Algorithm

The definition of previous state data is central to a system that detects changes of state. Since all incoming data is compared to this previous data state, what this previous state data is, and when it is stored is critical to the proper operation of the COS state machine.

First of all, several definitions are in order. A "flagged" Change-Of-State (COS) occurs when one or more data input channel's logic state changes in a way that the VMIVME-1183 has been programmed to detect. For example, if Channel 3 changes from a one (1) logic state to a zero (0) logic state, and the COS Select Register has been programmed to detect falling edges for Channel 3, then Channel 3 has caused a "flagged" COS event. A non-flagged COS event would occur if the logic level on Channel 3 changed from a logic zero (0) to a logic one (1), with falling edge detection enabled. Flagged COS data is the value of all of the input channels when one or more "flagged" changes of state have occurred. Non-flagged COS data is data in which one or more channels change state, but are not flagged for saving.

There are two approaches to choosing when input channel data is saved as previous state data:

- Store the value of the inputs only when a flagged COS event occurs.

- Store incoming data at some predetermined rate, or when any COS event, flagged or not, occurs.

On the VMIVME-1183, as incoming data completes the debounce cycle, if no flagged changes of state are detected, all 32 channels of data are clocked into the previous data buffer. The reason for this is to record the non-flagged COS events that will cause the flagged COS events to be detected. If a flagged COS event does occur, the changed state of the inputs is stored after the COS FIFO is written.

Consider the following:

Channel 16 of the COS circuit is programmed to detect rising edge events. The initial condition of Channel 16 is a logic one (1). At some indeterminate point in time, Channel 16's value drops to a logic zero (0). Then, at some future time, Channel 16's value changes back to a logic one (1). If data has been saved to the previous data register only when flagged COS events occur, then the 0 - 1 change of state may be missed since the recording of the previous state of Channel 16 may have been recorded due to a 0 – 1 change on Channel 16 itself. This would cause the previous data value of Channel 16 to be one (1). However, if the input channel data is periodically stored to the previous state register when no flagged COS events are occurring, then the 0 – 1 transition is detected because the zero (0) value of Channel 16 would have been saved.

"Simultaneous" Data Changes, and the Debounce "Stretching" Effect

When conducting Device Verification Testing (DVT), or production testing a digital I/O board, a common test utilized is the "walking ones" test. In this test, all input channels are set to zero (0), and a one (1) is written to each successive channel in the following manner:

State 0	00000000
State 1	00000001
State 2	00000010
State 3	00000100

In this test, two channels are actually changing value after State 1. In State 2, channel 0 has changed from a one (1) to a zero (0), and channel 1 has changed from a zero to a one. In an ideal world, these changes occur simultaneously, and are seen as a single COS event. In actuality, factors can be introduced which will make the arrival of signals on different channels non-simultaneous. Comparators have different times on

rising vs. falling edges. Trace lengths, both inside and outside of FPGAs, can have slightly different lengths. Cables between sensors and I/O boards may not be of precisely matched lengths. If any of the above conditions occur, then data transmitted simultaneously on two or more channels may not arrive simultaneously at the COS detection circuitry.

A factor which aggravates this situation is the need for the inputs to the FPGA containing the COS detection circuitry to be synchronized to a common clock. If two incoming signals arrive as little as 2 nanoseconds skewed, the first arriving signal may arrive just before the synchronizing rising clock edge, and the second signal may arrive just after the rising clock edge. If this occurs, then skew between the two signals is now stretched to the period of the synchronizing clock, which on the VMIVME-1183 is 16 MHz. Another factor, which can skew "simultaneous" signals to a greater extent, is the debounce clock. In a situation where two incoming signals are skewed by one or more synchronizing clock cycles, the synchronized incoming signals can encounter another early-late condition (as was in the synchronizing clock) with the debounce circuitry. Since the COS detection and FIFO circuitry runs at the VMEbus clock rate, the first arriving signal will have encountered, been processed by the COS circuit, and stored in the COS FIFO (if it was a flagged COS event) before the second arriving signal has passed the debounce circuitry. If the second signal was also a flagged COS, then two COS events would be stored in the COS FIFO. This "double-hit" COS event can also show what appears to be an invalid state condition. Consider the following example:

Channel 5 is flagged for rising edge events. Channel 4 is flagged for falling edge events. The CSR is set to SOE (Sequence-Of-Events) mode. In testing, a walking ones test is performed. The expected COS FIFO output is:

	Channel 5	Channel 4
Previous State 1	0	1
	01 to 10 change occurs here	
COS State 1	1	0

The actual FIFO output, due to skew introduced by cabling, synchronizing clock skewing, and debounce skewing, is as follows:

	Channel 4	Channel 5
Previous State 1	1	0
	01 to 10 change occurs here, but Channel 4 lags Channel 5 by 3nsec	
COS State 1	1	1

	Channel 4	Channel 5
Previous State 2	1	1
COS State 2	0	1

It appears, from the recovered COS FIFO data, that the previous state, and the current state have been logically “anded” together in COS State 1 and Previous State 2, while the perceived correct COS data does not appear until COS State 2. When viewed on a logic analyzer, the incoming data skew may have been as little as 3 or 4 nsec, but due to synchronizing clock skew stretching and debounce skew stretching, the saved COS sequence appears as two separate events.

If the Channel 4 COS detection circuitry had not been set to detect falling edges, a perceived erroneous result would still occur, but only Previous State 1 and COS State 1 would have occurred. This would still leave the tester with the impression that the values of the previous state (01) and final state (10) had somehow been logically “Anded” together.

A better approach to a walking ones test would be to reset the input channels to all zeros (0s) between walking a one from one channel to the next. This allows for the various skews to settle, and still allows the tester to detect channel cross-talk, which is one of the primary purposes of the walking ones testing.

Inputs

The input circuit topology (or type of circuit) is controlled by the user. Figure 1-3 on page 31 shows the basic circuit. It depicts the circuitry for input channel 30. The user can configure this circuit to monitor the various external inputs by simply adjusting the position of jumpers that control the common pins (pin 1) of the input bias resistors RP2 and RP3. Figure 1-5 on page 33 shows the voltage sourcing input circuit. Figure 1-6 on page 34 depicts the current sinking input topology. For monitoring contacts-to-ground, use the current sinking input circuit of Figure 1-6 on page 34. If you are monitoring contacts-to-power, use the voltage sourcing inputs of Figure 1-5 on page 33.

Once the input topology has been chosen, the trip threshold must be established. This is done by adjusting the voltage at the V_t point. The threshold voltage equation for a threshold resistor of 33 kΩ is $V_t = [(V- 1.25)*(.51)] + 1.25$ V, where V is selected by the user via a jumper field (E11 through E14). This jumper sets V to ground, +1.25V, +5V or whatever the external input voltage Vext is. Table 1-2 on page 35 lists the values for some commonly used values of V.

This biasing arrangement is done on a byte-by-byte basis. E24 and the jumpers shown in the figures control the input channels 31 through 24. This is input byte 0. Byte 1 is controlled in the same manner by E27 and the jumpers shown in Figure 1-1 on page 18 under the column name Byte 1. J12 is in this column. Byte 2 is controlled in the same manner by E9 and the jumpers shown in Figure 1-1 on page 18 under the column name Byte 2. J20 is in this column. Byte 3 is controlled in the same manner by E7 and the jumpers shown in Figure 1-1 on page 18 under the column name Byte 3. J5 is in this column. Table 1-2 on page 35 is valid for these input ports as well.

With this arrangement, each of the input ports can have a different configuration. For example, input Byte 0 can be a voltage sourcing input with a 3.2 V threshold for 5 V signals. Byte 1 could be a current sinking input using a 6.7 V trip level for 12 V inputs. Byte 2 may be set up to handle logic levels. This is a voltage sourcing input using the 1.25 V threshold and 5 V inputs. This circuit is not a true TTL input, but it will work with TTL signals. Finally, Byte 3 can be set to monitor some switches. This is done by using the current sinking topology with a threshold voltage of 6.7 V for 12 V signals. This shows just one of the ways this board can be configured. See Chapter 2 for a detailed discussion on how to configure the board. As you can see, the board can monitor a variety of inputs at the same time just by placing jumpers in the right places on the board.

These jumpers set up the type of input the external circuit will see.
Grounding resistor pack E24 will require an external voltage source input.
Using any of the other jumpers will show the external circuitry a current sink input with the shown voltage level.

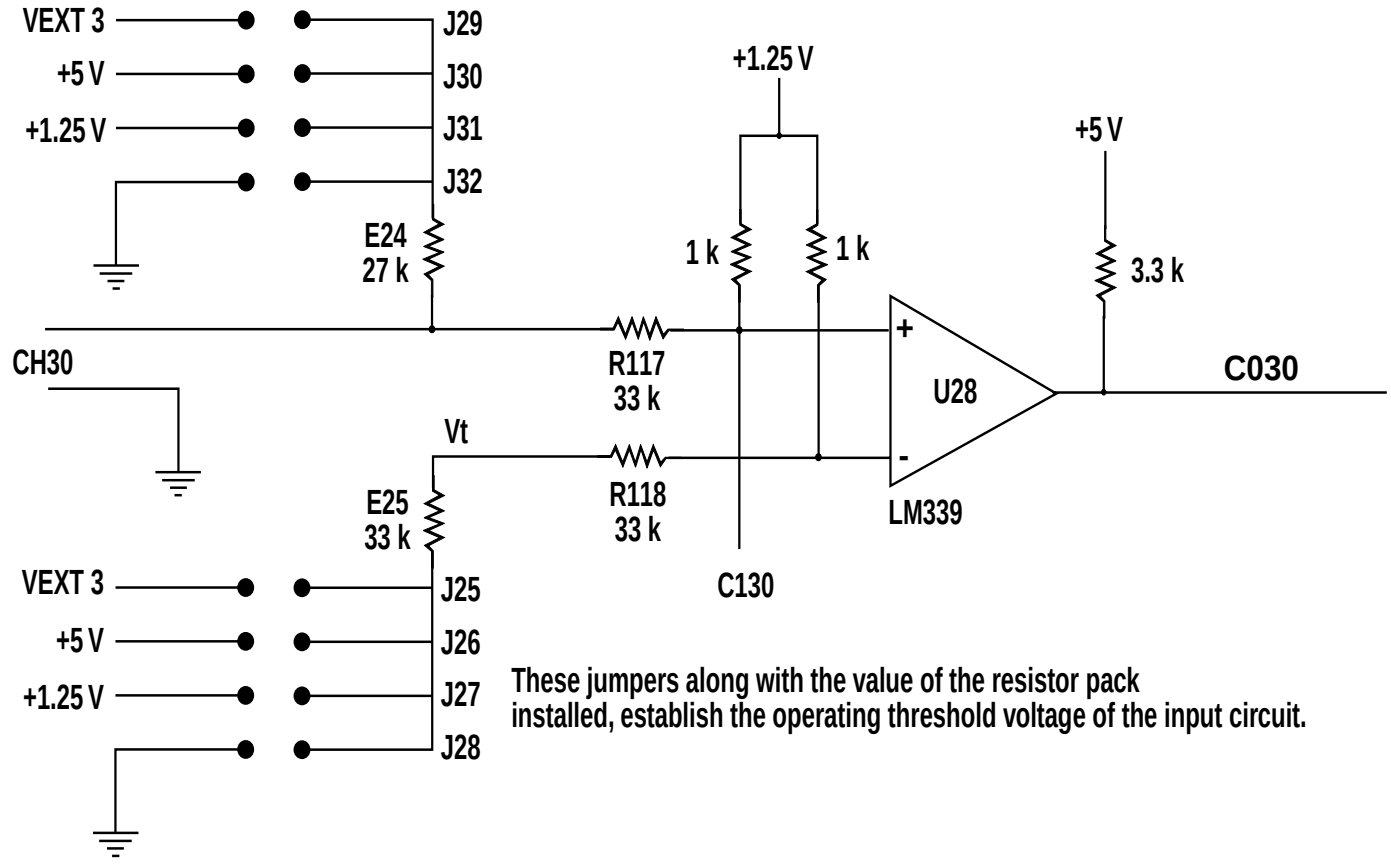
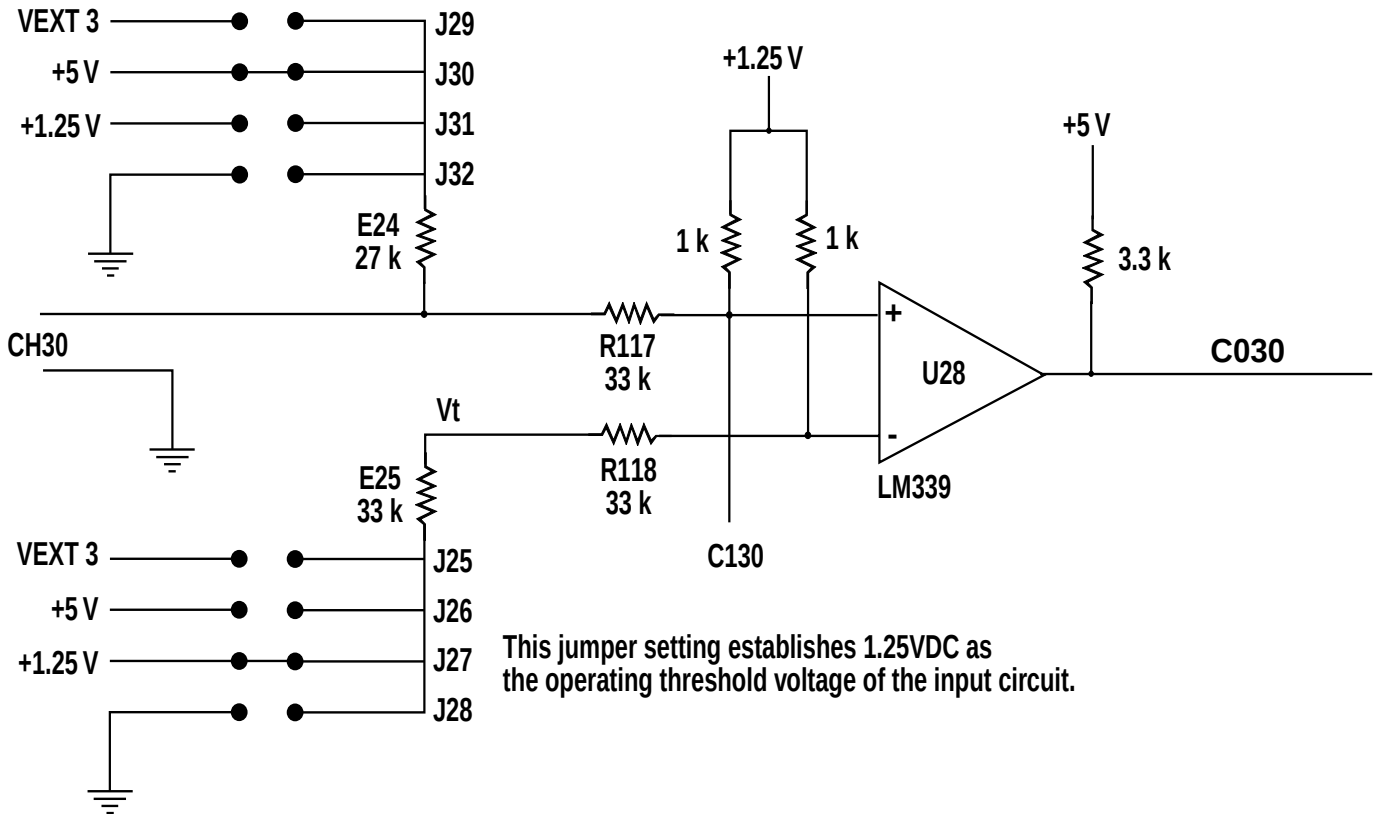


Figure 1-3 Basic Input Circuitry

These jumpers along with the value of the resistor pack installed, establish the operating threshold voltage of the input circuit.

This jumper setting supplies a 5VDC pullup path to the inputs.
When the relay is open the inputs are pulled up to 5VDC and the input turns on.
When the relay is closed, the pullup is eliminated and the input turns off.



This jumper setting establishes 1.25VDC as
the operating threshold voltage of the input circuit.

Figure 1-4 Contact Sensing Example

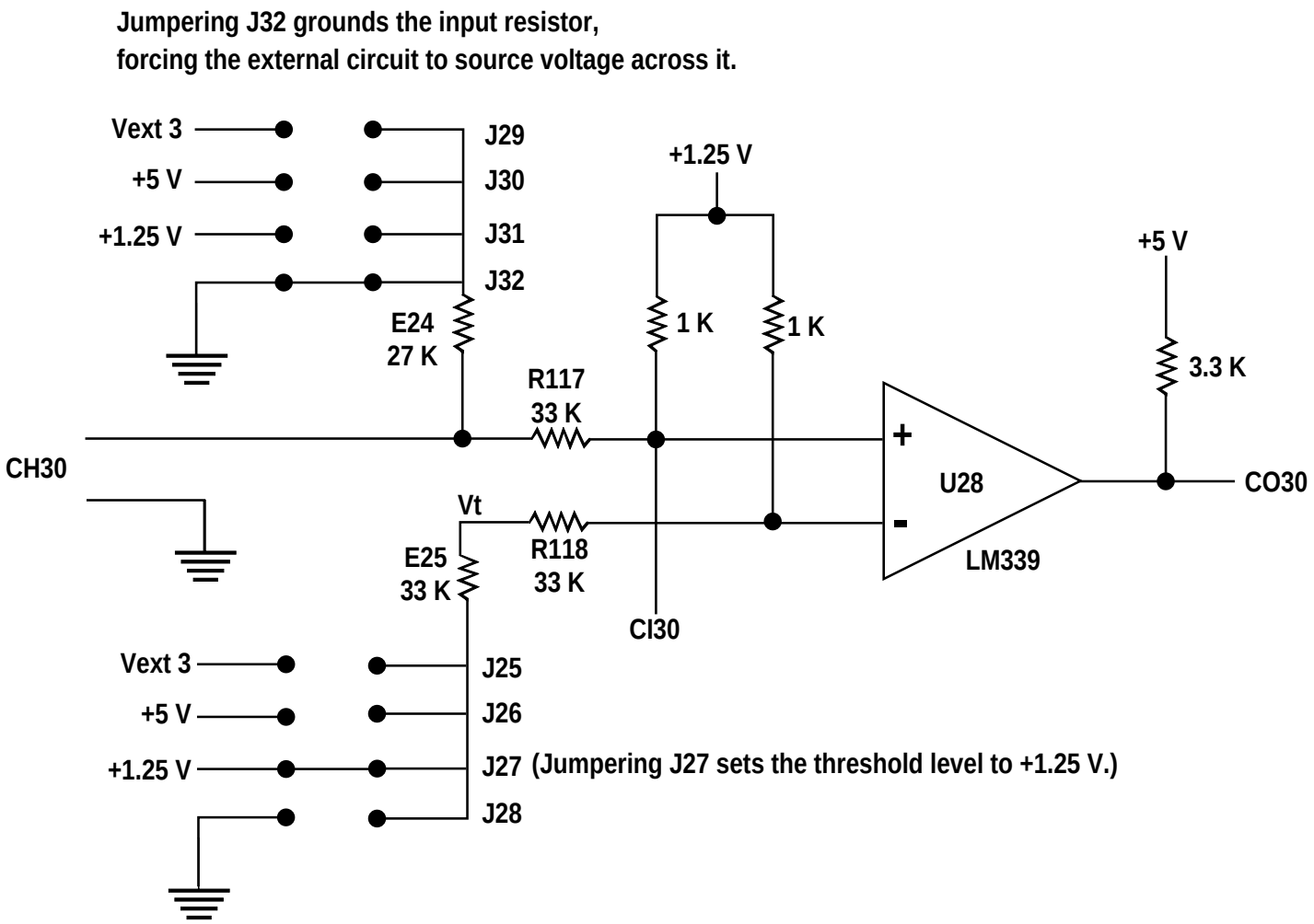


Figure 1-5 Voltage Sourcing Example

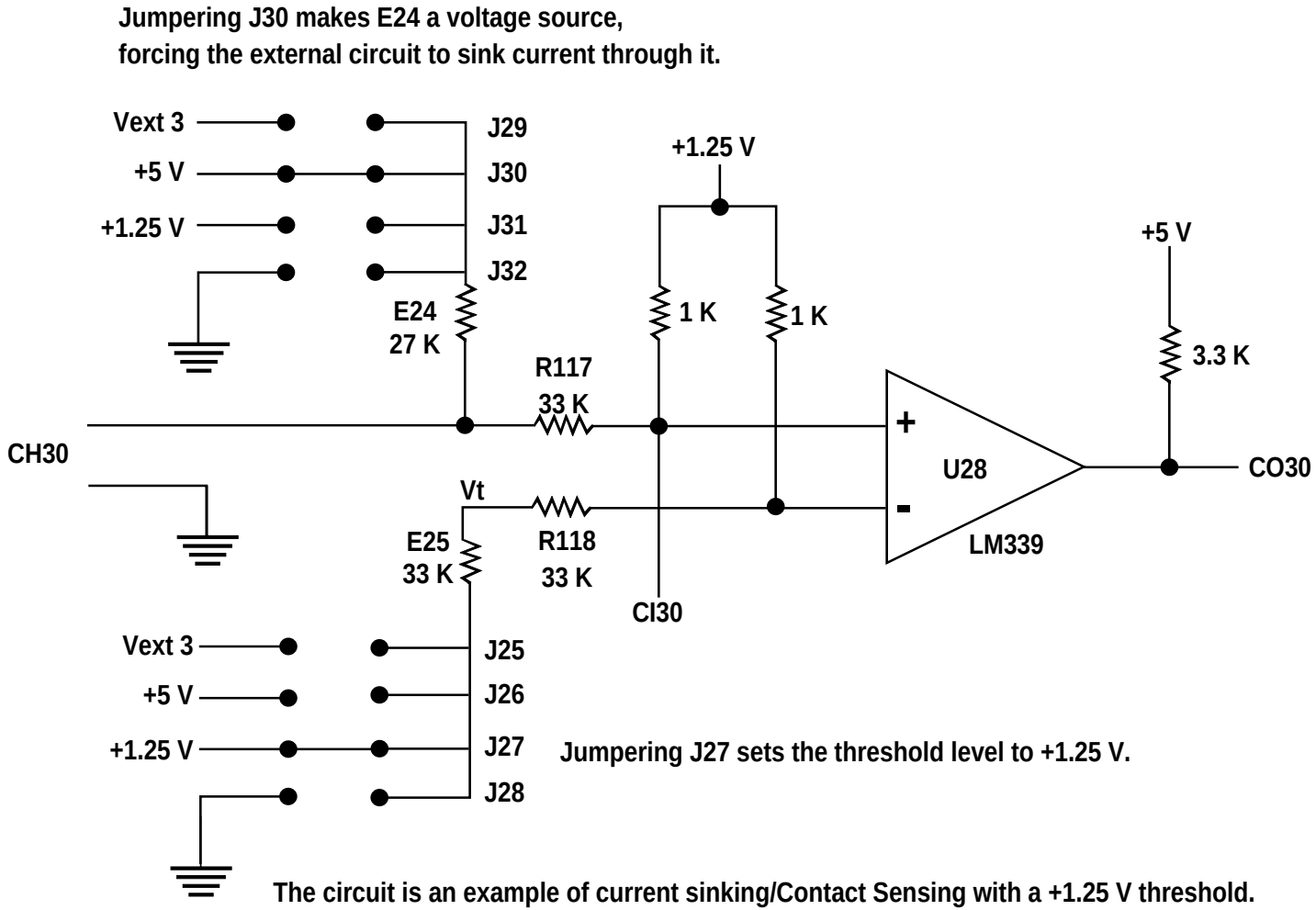


Figure 1-6 Current Sink/Contact Sensing Example

Table 1-2 Threshold Voltages for E25 = 33 kΩ, $V_t = [(V - 1.25)*(.51)] + 1.25$ (Default)

V	Vt
0 V	0.61 V
1.25 V	1.25 V
5 V	3.16 V
12 V	*6.73 V
24 V	*12.85 V
28 V	*14.89 V
48 V	*25.09 V
66 V	*34.27 V

NOTE: *Requires external voltage source for bias and input.

Table 1-3 Input and Threshold Biasing Resistor Pack Position Assignment

Channels	Input Bias	Threshold Bias
0 through 7	E7	E6
8 through 15	E9	E8
16 through 23	E27	E28
24 through 31	E24	E25

Built-In-Test

The VMIVME-1183 is designed with Built-In-Test (BIT) internal logic. Special output registers are provided to permit writing data to the board. This data is written to the same addresses as the input data registers. This way you can check a particular input byte. These registers are enabled via a test mode bit in the CSR (see Chapter 3 for programming information concerning this bit). Once these registers are enabled, the data stored will overwrite the external inputs. The data read can be compared with the written data. In this manner, the "health" of the board can be determined. BIT is always available to the user. There is only one test mode bit in the CSR. When you enable test mode, all of the BIT registers are enabled. You cannot enable the output registers on a byte-by-byte basis.

Since BIT affects the input data, it will also affect the COS logic. If this is not desired, set the COS mode to "No Interrupts" (see the COS selection table). The COS logic simply samples the input data at the rising edge of the debounce clock. Changing input data will be sensed by the COS logic and appropriate action, if any, will be taken. This way, the COS logic and the FIFO data capture registers can be tested. However, if any "real-time" data previously stored in the FIFO must be dealt with in some manner, testing should be done with an empty FIFO. This way the data in the FIFO reflects the BIT data and not some residual external data.

Since BIT also overwrites the external inputs, any toggle changes in the external signals during testing will be lost. For example, if a valid input signal goes from low to high and then back to low, (or vice versa) before testing is done, it will be lost. A "valid input" is an input state that is present at the rising edge of the VMEbus clock.

NOTE: BIT should occur during system resets or before data acquisition begins.

A front panel Fail LED is available to the user. It is illuminated at power-up or after a system reset and can be extinguished under program control upon the successful completion of diagnostic testing. It can be used for any purpose you wish. For example, it can be used to display data activity.

CHAPTER
2

Configuration and Installation

Contents

Introduction

This chapter describes the installation and configuration of the board. Cable configuration, jumper / switch configuration and board layout are illustrated in this chapter.

Unpacking Procedures

CAUTION: Some of the components assembled on VMIC's products may be sensitive to electrostatic discharge and damage may occur on boards that are subjected to a high-energy electrostatic field. When the board is placed on a bench for configuring, etc., it is suggested that conductive material should be inserted under the board to provide a conductive shunt. Unused boards should be stored in the same protective boxes in which they were shipped.

Upon receipt, any precautions found in the shipping container should be observed. All items should be carefully unpacked and thoroughly inspected for damage that might have occurred during shipment. The board(s) should be checked for broken components, damaged printed circuit board(s), heat damage, and other visible contamination. All claims arising from shipping damage should be filed with the carrier and a complete report sent to VMIC together with a request for advice concerning the disposition of the damaged item(s).

Physical Installation

CAUTION: Do not install or remove the boards while power is applied.

Before Applying Power: Checklist

Before installing the board in a VMEbus system, check the following items to ensure that the board is ready for the intended application.

1. Review *Factory Installed Jumpers* on page 40 and Table 2-1 on page 41 to verify that all the desired jumpers are in place. To change the board address or address modifier response, refer to *Board Address and Address Modifier Selection* on page 40. _____
2. Review *Input Configurations* on page 46 to configure the input circuits for the desired operation. _____
3. Have the I/O cables, with the proper mating connectors, been connected to the input/output connector P2? Refer to *Connector Description* on page 48 for a description of the P2 connector. _____
4. Have the sections pertaining to theory and programming (Chapter 1 and 3) been reviewed and applied to the system requirements? _____
5. De-energize the equipment and insert the board into an appropriate slot of the chassis. While ensuring that the board is properly aligned and oriented in the supporting card guides, slide the board smoothly forward against the mating connector until firmly seated.

Operational Configuration

The VMIVME-1183 Board's base address and I/O access mode are determined by field replaceable, on-board jumpers. This section describes the use of these jumpers. The locations and function of all the jumpers on the VMIVME-1183 are shown in Figure 2-1 on page 44 and Table 2-1 on page 41.

Factory Installed Jumpers

Each VMIVME-1183 is configured at the factory with the specific jumper arrangement shown in Figure 2-1 on page 44. The factory configuration establishes the following functional baseline for the board, and ensures that all essential jumpers are installed.

- Base address is set to 0000 HEX
- Short supervisory and short nonprivileged I/O accesses
- Input biasing of 1.25 V, threshold biasing of 1.25 V
- Current sinking inputs
- A 10 µs debounce time

Board Address and Address Modifier Selection

The jumper field shown in Figure 2-1 on page 44 shows the jumpers and their associated address weights. The values of these bits is determined by the presence or absence of a jumper shunt. When a jumper shunt is installed the address bit's value must be a "zero" for a match. If the jumper shunt is omitted, the value is a "one" to get a match. Therefore, a shunt is installed at every "zero" in the base address to be assigned to this board.

This board is designed to respond to standard (A24) or short (A16) address ranges. Jumper J56 is used to establish the address range for the board. With a jumper shunt omitted from J56 the board will operate in the **standard** address region. In this mode, the upper 8 (A16 to A23) address lines will be ignored. Thus, the jumpers for those address lines can be removed. If J56 jumper shunt is installed, then these lines will be decoded. Therefore, they must be configured as the base address demands.

Jumpers J54 and J55 determine which address modifier the board will respond to during data transfers (see Figure 2-1 on page 44). These jumpers determine supervisory or nonprivileged transfers. Installing a jumper at J54 selects nonprivileged transfers only. Omitting both jumpers will have the board respond only to supervisory accesses. But by installing a jumper at J55, the board will accept either I/O access.

NOTE: DO NOT install both jumpers. If J54 or J55 is installed, the AM2 line wil be grounded and this will disrupt the system's operations.

Table 2-1 Jumper Functions and Factory Configuration

Ref Des	Name/Function	Factory Configuration
J1	External Voltage for the threshold biasing of INPUT BYTE 0	OMITTED
J2	+5 V for the threshold biasing of INPUT BYTE 0	OMITTED
J3	+1.25 V for the thershold biasing of INPUT BYTE 0	INSTALLED
J4	For grounding the threshold biasing network of INPUT BYTE 0	OMITTED
	USE ONLY ONE OF THESE FOUR JUMPER POSITIONS	
J5	External Voltage for the input biasing of INPUT BYTE 0	OMITTED
J6	+5 V for the input biasing of INPUT BYTE 0	INSTALLED
J7	+1.25 V for the input biasing of INPUT BYTE 0	OMITTED
J8	For grounding the input network of INPUT BYTE 0	OMITTED
	USE ONLY ONE OF THESE FOUR JUMPER POSITIONS	
J9	External Voltage for the threshold biasing of INPUT BYTE 1	OMITTED
J10	+5 V for the threshold biasing of INPUT BYTE 1	OMITTED
J11	+1.25 V for the threshold biasing of INPUT BYTE 1	INSTALLED
J12	For grounding the threshold network of INPUT BYTE 1	OMITTED
	USE ONLY ONE OF THESE FOUR JUMPER POSITIONS	
J13	External Voltage for the input biasing of INPUT BYTE 1	OMITTED
J14	+5 V for the input biasing of INPUT BYTE 1	INSTALLED
J15	+1.25 V for the input biasing of INPUT BYTE 1	OMITTED
J16	For grounding the input biasing network of INPUT BYTE 1	OMITTED
	USE ONLY ONE OF THESE FOUR JUMPER POSITIONS	
J17	External Voltage for the threshold biasing of INPUT BYTE 2	OMITTED
J18	+5 V for the threshold biasing of INPUT BYTE 2	OMITTED
J19	+1.25 V for the threshold biasing of INPUT BYTE 2	INSTALLED
J20	For grounding the threshold network of INPUT BYTE 2	OMITTED
	USE ONLY ONE OF THESE FOUR JUMPER POSITIONS	
J21	External Voltage for the input biasing of INPUT BYTE 2	OMITTED
J22	+5 V for the input biasing of INPUT BYTE 2	INSTALLED
J23	+1.25 V for the input biasing of INPUT BYTE 2	OMITTED
J24	For grounding the input biasing network of INPUT BYTE 2	OMITTED
	USE ONLY ONE OF THESE FOUR JUMPER POSITIONS	

Table 2-1 Jumper Functions and Factory Configuration (Continued)

J25	External Voltage for the threshold biasing of INPUT BYTE 3	OMITTED
J26	+5 V for the threshold biasing of INPUT BYTE 3	OMITTED
J27	+1.25 V for the threshold biasing of INPUT BYTE 3	INSTALLED
J28	For grounding the threshold network of INPUT BYTE 3	OMITTED
	USE ONLY ONE OF THESE FOUR JUMPER POSITIONS	
J29	External Voltage for the input biasing of INPUT BYTE 3	OMITTED
J30	+5 V for the input biasing of INPUT BYTE 3	INSTALLED
J31	+1.25 V for the input biasing of INPUT BYTE 3	OMITTED
J32	For grounding the input biasing network of INPUT BYTE 3	OMITTED
	USE ONLY ONE OF THESE FOUR JUMPER POSITIONS	
J33	A23 / decodes this address line as part of the board decoder	INSTALLED
J34	A22 / decodes this address line as part of the board decoder	INSTALLED
J35	A21 / decodes this address line as part of the board decoder	INSTALLED
J36	A20 / decodes this address line as part of the board decoder	INSTALLED
J37	A19 / decodes this address line as part of the board decoder	INSTALLED
J38	A18 / decodes this address line as part of the board decoder	INSTALLED
J39	A17 / decodes this address line as part of the board decoder	INSTALLED
J40	A16 / decodes this address line as part of the board decoder	INSTALLED
J41	A15 / decodes this address line as part of the board decoder	INSTALLED
J42	A14 / decodes this address line as part of the board decoder	INSTALLED
J43	A13/ decodes this address line as part of the board decoder	INSTALLED
J44	A12 / decodes this address line as part of the board decoder	INSTALLED
J45	A11 / decodes this address line as part of the board decoder	INSTALLED
J46	A10 / decodes this address line as part of the board decoder	INSTALLED
J47	A09 / decodes this address line as part of the board decoder	INSTALLED
J48	A08 / decodes this address line as part of the board decoder	INSTALLED
J49	A07/ decodes this address line as part of the board decoder	INSTALLED
J50	A06 / decodes this address line as part of the board decoder	INSTALLED
J51	Not used - Do not install jumper	OMITTED
J52	Not used - Do not install jumper	OMITTED
J53	Not used - Do not install jumper	OMITTED
J54	AM2 / installed for nonprivileged : omit for supervisory	OMITTED
J55	install this jumper for either nonprivileged or supervisory	INSTALLED
	DO NOT INSTALL A JUMPER ON BOTH J54 AND J55.	

This will short the AM2 VMEbus line to ground and disrupt chassis operations.

Table 2-1 Jumper Functions and Factory Configuration (Concluded)

J56	Installed for STANDARD (A24) addressing : omit for SHORT (A16)	INSTALLED
J57	Selects a 10 µs Debounce Clock for the COS logic	INSTALLED
J58	Selects a 1 ms Debounce Clock for the COS logic	OMITTED
J59	Selects a 5 ms Debounce Clock for the COS logic	OMITTED
J60	Selects a 10 ms Debounce Clock for the COS logic	OMITTED
	USE ONLY ONE OF THESE FOUR JUMPER POSITIONS	
E4	Source of External Voltage for INPUT BYTE 3 (position 3-5)	OMITTED
	or ground return for INPUT CHANNEL 7 (position 1-3)	INSTALLED
	Source of External Voltage for INPUT BYTE 2 (position 4-6)	OMITTED
	or ground return for INPUT CHANNEL 15 (position 2-4)	INSTALLED
E8	Source of External Voltage for INPUT BYTE 1 (position 3-5)	OMITTED
	or ground return for INPUT CHANNEL 23 (position 1-3)	INSTALLED
E9	Source of External Voltage for INPUT BYTE 0 (position 4-6)	OMITTED
	or ground return for INPUT CHANNEL 31 (position 2-4)	INSTALLED

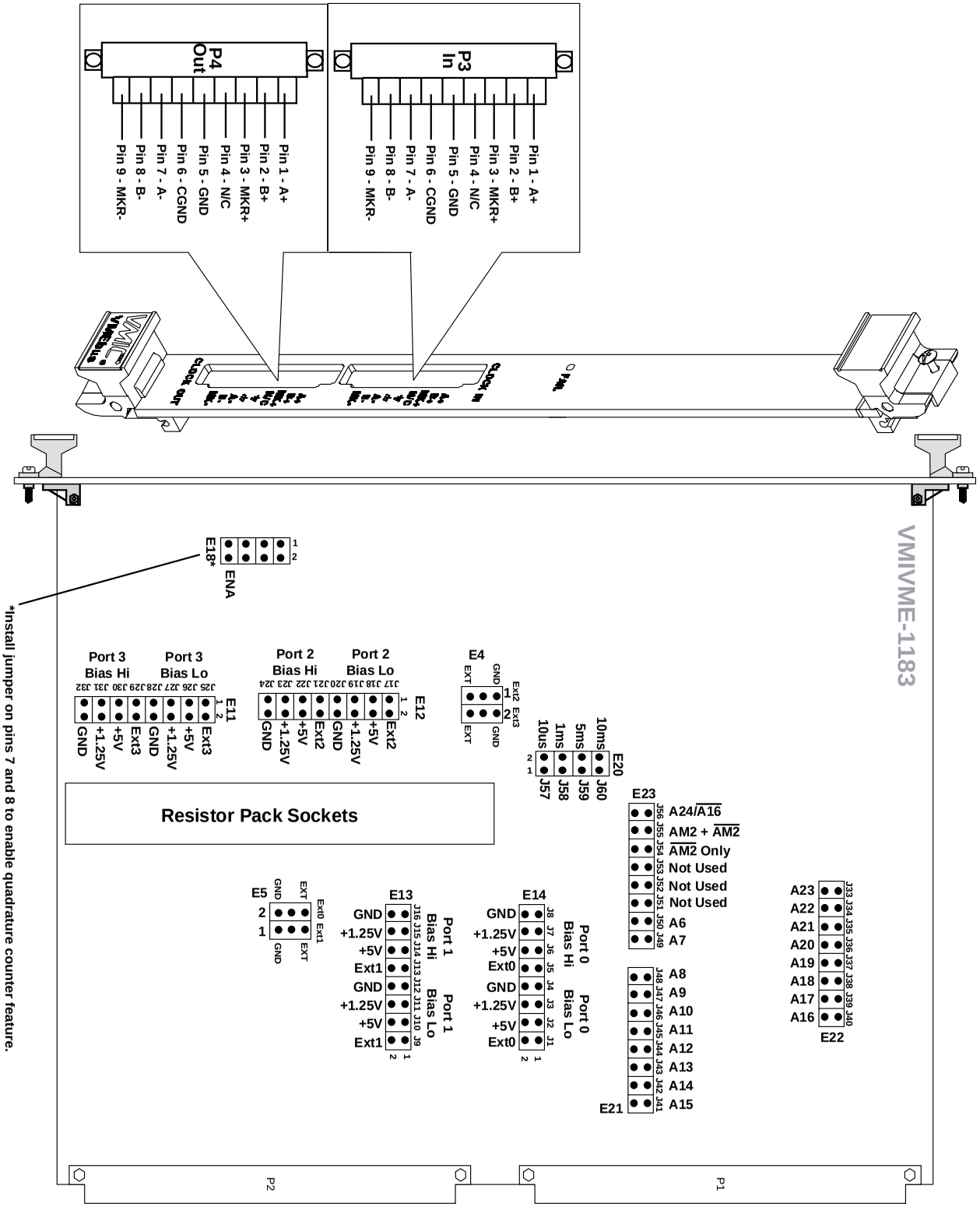


Figure 2-1 Jumper Locations and Factory Configuration

Debounce Timing

Input debouncing is done by an adjustable sample clock. The user chooses this clock by installing a jumper in one of the positions marked J57 through J60. Table 2-1 on page 41 describes these jumpers. Figure 2-1 on page 44 shows the jumpers and their timing values. If the jumpers are omitted for J57 and J60, the debounce timer is 1 µsec. The Change-of-State logic periodically takes a sample of the input level to determine the present state of the input channel. Unlike the input circuitry which continuously monitors the external world, the COS constantly takes a sample of the outside environment at the VMEbus clock rate (16 MHz). The debounce times available have been chosen to provide control for a wide range of inputs. Fast times for electrical noise and slow times for switches and relays.

Table 2-2 Jumper E20 (Debounce Timing)

Pin#	Selection	Jumper
J57	10 µs	Installed (Default)
J58	1 ms	Omitted
J59	5 ms	Omitted
J60	10 ms	Omitted

Input Configurations

The input circuitry of the VMIVME-1183 can be configured by the user to monitor a variety of external circuits. The user also has control over the threshold to use. There are jumpers on the board to accomplish this. However, this control is over eight inputs at a time. Each input port can be configured differently, but not each input channel. Chapter 1 discusses the various circuit topologies that can be created with this design.

There are three jumper fields per input port used to control the topology of the input circuitry. Table 2-1 on page 41 lists the jumpers used by each port. Figure 1-5 on page 33 shows a voltage source topology. Figure 1-6 on page 34 shows the current sink configuration. Table 2-3 below lists some of the various thresholds that the circuit will support as well as the threshold equation for them. Use the equation for threshold biases that are not listed here. The voltage sourcing input can be used to monitor a contact switch-to-power. Where as the current sink topology can monitor a contact switch-to-ground.

Table 2-3 Threshold Voltages

Threshold Resistance	Threshold Bias	Threshold Voltage	Threshold Equation
33 kΩ	0.00 V	1.23 V	$V_t=[(V_{BIAS}-1.25)x(.51)]+1.25$ V
	1.25V	1.25 V	
	5.00 V	3.16 V	
	12.00 V	*6.73 V	
	24.00 V	*12.85 V	
	28.00 V	*14.89 V	
	48.00 V	*25.09 V	
	66.00 V	*34.27 V	

NOTE: *Require external bias voltage and inputs.

The controlling jumpers are part of a jumper field (or header). These headers have several positions but only one jumper must be installed. If more than one jumper is installed, field power supplies may get shorted together or to ground. Such an event could cause severe damage to the board as well as the system. Be sure to carefully account for the positions of these jumpers. Make copies of this figure and mark it for the configuration desired, then check it before installing the board and applying power. For your convience, the functions of the various headers are listed in Figure 2-2 on page 47. This assumes the board is oriented as shown in the figure. Each input byte has a column of eight jumpers. The upper four are for the threshold bias, while the lower four are the input bias select lines. Each row of jumpers have the same value: GND (ground), +1.25 V, +5 V, or the external voltage coming from the P2 connector. Be sure to set E4 and E5 (the jumpers used by the bytes selected) correctly when using external voltage.

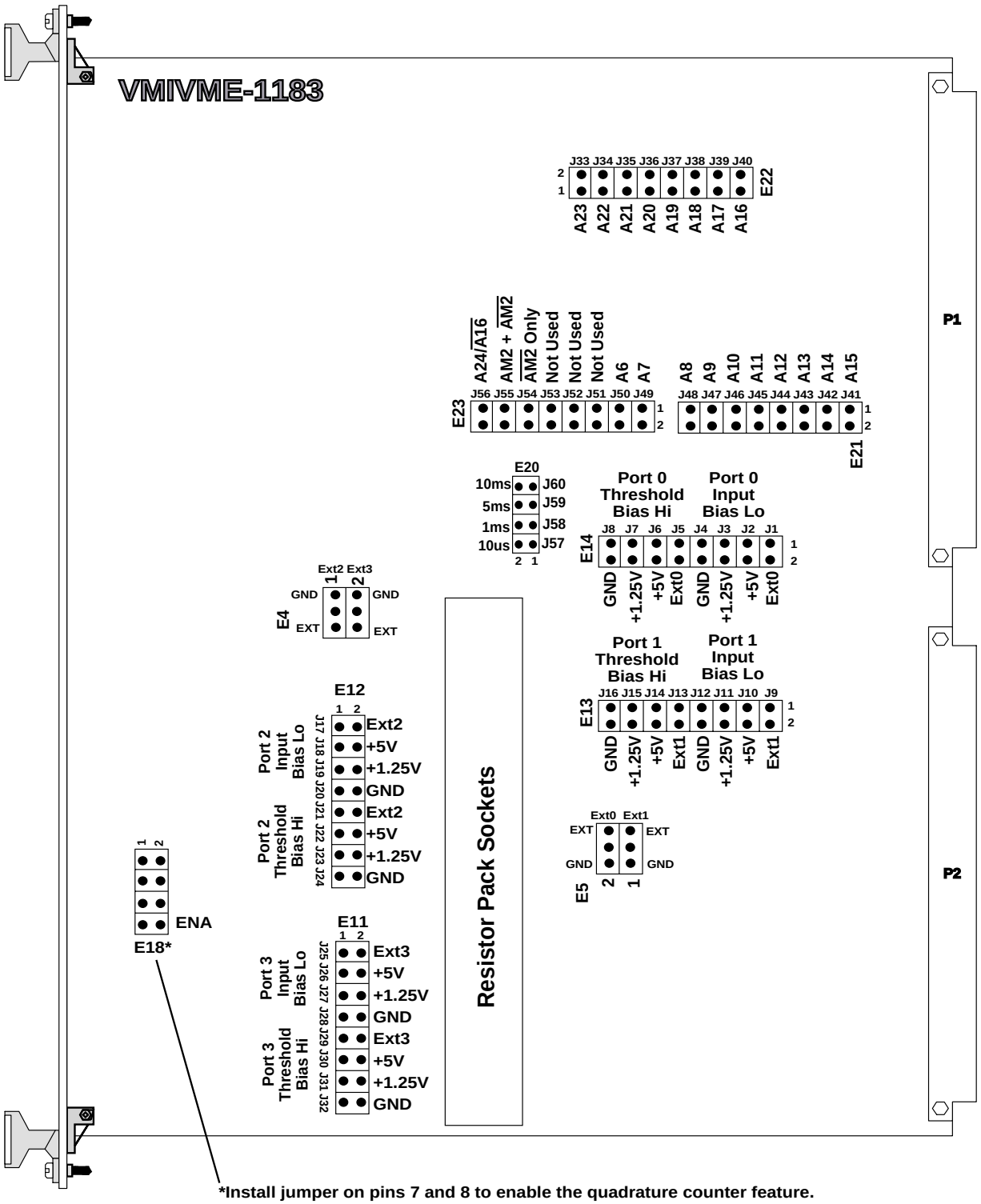


Figure 2-2 Input Configuration Worksheet

Connector Description

Two 96-pin DIN connectors, P1 and P2 (Figure 2-3 on page 49), provide all of the connections to the VMIVME-1183 Board. P1 contains the address, data and control lines, plus all the additional signals necessary to control the VMEbus functions of the board. P2 provides the connections for the upper 16 data lines, additional power and ground pins, and the user inputs.

Orientation of the P1 and P2 connectors are shown in Figure 2-3 on page 49. The P2 signal assignments are listed in Table 2-4 on page 50. The mating connector for P2 (Panduit Model 120-964-435E or equivalent) is designed to be used with a standard 64-wire ribbon-cable with conductor spacing of 0.050 inches. In environments where there is a high degree of electrical noise, a twisted-pair ribbon cable with an overall shield is recommended.

Two nine-channel screw terminal connectors (P3 and P4), provide the respective inputs, and daisy-chain outputs for a 0 - 5V single-ended or differential quadrature counter (with marker pulse). See Table 2-5 on page 51 for connector pinouts.

Barrier Terminal Transition Panels

The VMIVME-1183 can be used with VMIC's BT0X family of BT transitions panels. The VMIACC-BT01, 02, 03, and 04 family of BT transition panels meets ANSI/IEEE SWC TEST.

The BT transition panels are passive and provide an efficient and versatile interfacing between discrete wires to ribbon cables, between external user equipment and VMIC's VMEbus-based interface boards. The BT transition panels eliminate the need for wire lugs and other bulky methods of wiring transitions.

The VMIACC-BT01 differential dual 64-pin transition panel facilitates 64 isolated pair circuits with no common ground (2-wire). The VMIACC-BT02 single-ended dual 64-pin transition panel facilitates 64 non-isolated pair circuits with the C-row tied to common ground E1 (2-wire). The VMIACC-BT03 differential dual 96-pin transition panel facilitates 64 isolated pair circuits with a common ground (3-wire). The VMIACC-BT04 dual 96-pin transition panel breaks out 192 individual circuits to terminal blocks. The BT transition panels are EIA RS-310C standard 19-inch rack mountable. The BT transition panels are compact and convenient to use even where space is a constraint. The panels will accommodate wire size from 22 AWG to 14 AWG stranded wire or to 12 AWG solid wire.

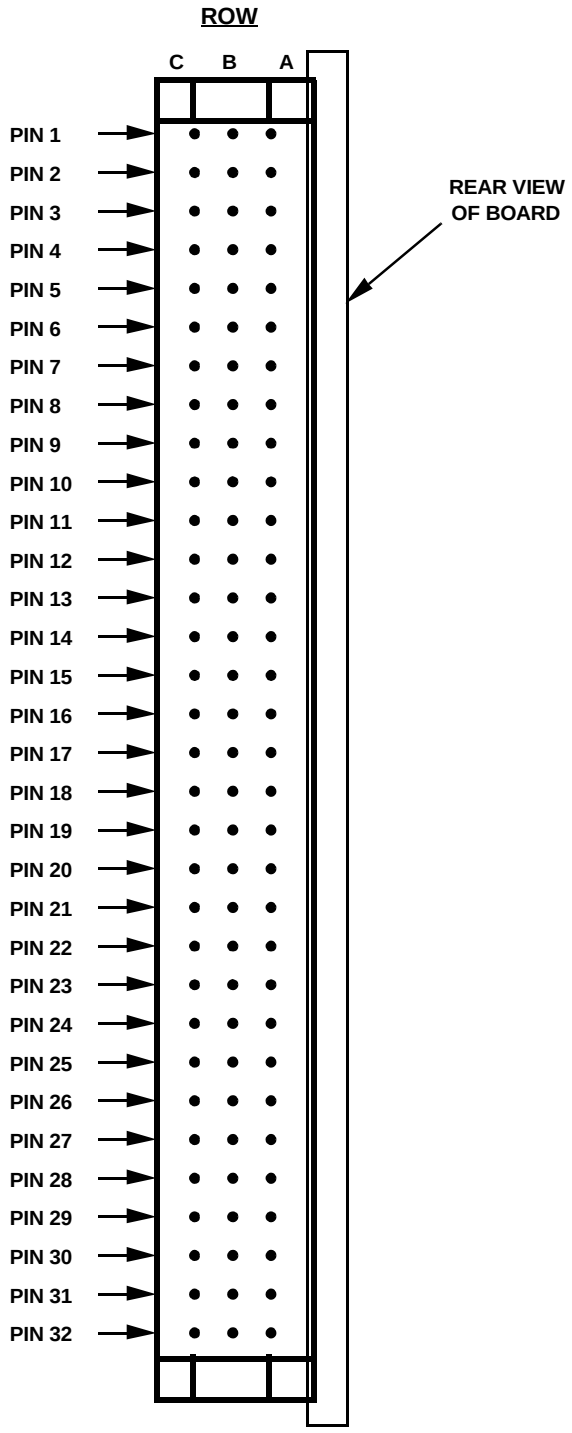


Figure 2-3 P2 Connector Pin Layout

Table 2-4 P2 Pin Assignments

Pin	Function	Pin	Function
A1	CH00 INPUT	C1	CH00 GND
A2	CH01 INPUT	C2	CH01 GND
A3	CH02 INPUT	C3	CH02 GND
A4	CH03 INPUT	C4	CH03 GND
A5	CH04 INPUT	C5	CH04 GND
A6	CH05 INPUT	C6	CH05 GND
A7	CH06 INPUT	C7	CH06 GND
A8	CH07 INPUT	C8	*CH07 GND or Vext for Port 3
A9	CH08 INPUT	C9	CH08 GND
A10	CH09 INPUT	C10	CH09 GND
A11	CH10 INPUT	C11	CH10 GND
A12	CH11 INPUT	C12	CH11 GND
A13	CH12 INPUT	C13	CH12 GND
A14	CH13 INPUT	C14	CH13 GND
A15	CH14 INPUT	C15	CH14 GND
A16	CH15 INPUT	C16	*CH15 GND or Vext for Port 2
A17	CH16 INPUT	C17	CH16 GND
A18	CH17 INPUT	C18	CH17 GND
A19	CH18 INPUT	C19	CH18 GND
A20	CH19 INPUT	C20	CH19 GND
A21	CH20 INPUT	C21	CH20 GND
A22	CH21 INPUT	C22	CH21 GND
A23	CH22 INPUT	C23	CH22 GND
A24	CH23 INPUT	C24	*CH23 GND or Vext for Port 1
A25	CH24 INPUT	C25	CH24 GND
A26	CH25 INPUT	C26	CH25 GND
A27	CH26 INPUT	C27	CH26 GND
A28	CH27 INPUT	C28	CH27 GND
A29	CH28 INPUT	C29	CH28 GND
A30	CH29 INPUT	C30	CH29 GND
A31	CH30 INPUT	C31	CH30 GND
A32	CH31 INPUT	C32	*CH31 GND or Vext for Port 0

NOTE: Selected by jumpers E4 and E5.

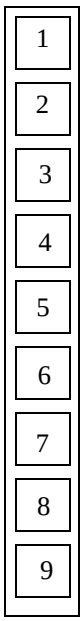


Figure 2-4 P3/P4 Connector

Table 2-5 P3 Connector Pinout

P3 Connector	
Pin#	**Function**
1	Channel A (+) Input
2	Channel B (+) Input
3	Marker Pulse (+) Input
4	N/C
5	Signal Ground
6	Chassis Ground
7	Channel A (-) Input
8	Channel B (-) Input
9	Marker Pulse (-) Input

Table 2-6 P4 Connector Pinout

P4 Connector	
Pin#	**Function**
1	Channel A (+) Output
2	Channel B (+) Output
3	Marker Pulse (+) Output
4	N/C
5	Signal Ground
6	Chassis Ground
7	Channel A (-) Output
8	Channel B (-) Output
9	Marker Pulse (-) Output

CHAPTER

3

Programming

Contents

Introduction

The VMIVME-1183 can reside in short I/O space or standard data space. There are 18 jumpers to establish the base address of the board. Table 3-1 on page 54 lists the registers and their relative (or offset) addresses. The relative address is added to the base address to generate actual address for the board's register. The board logic uses address lines A5 through A1 to decode the registers listed in Table 3-1 on page 54. The rest of the address lines used for the board are set by the address jumpers. In Chapter 2 there is a detailed explanation of how to set up these jumpers.

Table 3-1 on page 54 only lists the offset to the registers. This offset is added to the base address of the board to create the actual address for the register.

Programming the VMIVME-1183 involves setting up the IP and the COS logic. Table 3-1 on page 54 is a memory map showing the relative locations of these registers. When the board is first powered-up or after a system reset, the CSR and the IP's control registers are cleared while the IP's vector registers initialize to $0F.

NOTE: The $ denotes the characters are hexadecimal, while a % implies that the digits are binary.

It is recommended that the IP be programmed first with its Interrupt Enable (IRE) bits cleared. Then set up the COS logic via the COS Select and the Channel Interrupt Enable registers. In the IP Control registers set the IRE bits to start the interrupts. During interrupt acknowledge cycles, the IP will supply the service routine's vector. In the service routine, read the FIFO to clear the interrupt from the IP.

To perform data transfers, read from or write to the Data addresses (Offset $04). The FIFOs and the BD ID registers are read-only. The board will not respond to writes to these registers, and the data will be ignored. During a read of the Data registers, the board will sample the data received at the input (P2) and transfer that data to the VMEbus. Writing to these registers has no effect unless Test Mode is active. All registers except the FIFO register, CTR_FIFO and the QUAD_CTR can be accessed using byte, word or longword transfers. The FIFO register, CTR_FIFO and the QUAD_CTR can only be accessed with longword transfers.

Table 3-1 VMIVME-1183 Address Map

Relative Address	Register Name	Register Function
$00	BD ID	Automatic system setup (fixed @ $5D00)
$02	CSR	Board Control bits and Status flags
$04	Data Register	Input or BIT channels 31 through 0
$08	Data FIFO Register	COS data for channels 31 through 0
$0C	IP CTRL Register	Interrupt processor controls
$0D	IP COS Vector Register	COS/SOE Interrupt Vector
$0E	IP MKR Vector Register	Marker Pulse Interrupt Vector
$10	CTR FIFO Register	Counter FIFO register
$14	Quad_CTR	Quadrature Counter Current Value
$18	COS_SEL Register 0	COS Select Register for channels 31 through 24
$1A	COS_SEL Register 1	COS Select Register for channels 23 through 16
$1C	COS_SEL Register 2	COS Select Register for channels 15 through 8
$1E	COS_SEL Register 3	COS Select Register for channels 7 through 0
$20	FIFO_CNT	Number of unread samples in the COS FIFO
$22	CTR_FIFO_CNT	Number of unread samples in the Quad CTR FIFO
$24	CH_INT_ENA	Channel Interrupt Enable register
$30	FREV	Firmware Revision ID

Board ID (BD ID) Register

Table 3-2 Board ID Register Bit Map

BD ID: Offset $XXXX00							
Bit 31	**Bit 30**	**Bit 29**	**Bit 28**	**Bit 27**	**Bit 26**	**Bit 25**	**Bit 24**
0	1	0	1	1	1	0	1

Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
0	0	0	0	0	0	0	0

Control and Status Register

Table 3-3 Control and Status Register Bit Map

CSR: Offset $XXXX02							
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 09**	**Bit 08**
Fail LED	Disable Test Mode	FIFO Empty	FIFO Full	COS/SOE (0/1)	ROFE/ROAK (0/1)	EN Quad CTR FIFO	Reserved

Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
Reserved				CTR_FIFO_Full	CTR_FIFO_Empty	MRK_Reset	CTR_ENA

Control and Status Register Bit Definitions

Bit 15: **Fail LED** - Writing a one (1) to this bit will turn OFF the LED. Writing a zero (0) to this bit will turn ON the LED.

Bit 14: **Disable Test Mode** - Writing a one (1) to this will disable (stop) Test Mode operations. When this bit is a zero (0) Test Mode is active. This will permit writing data to the input data registers, which can be used to check the COS logic and the data paths on the board.

Bit 13: **FIFO Empty** – This bit is a read-only flag that when set, indicates that the COS/SOE FIFO is empty.

Bit 12: **FIFO Full** – This bit is a read-only flag that when set, indicates that the COS/SOE FIFO is full, and any additional COS/SOE events may be lost.

Bit 11: **COS/SOE (0/1)** - Change-of-State/Sequence-of-Events: Bit 11 is set to zero (0) for the Change-of-State mode, and to a one (1) for the Sequence-of -Events mode.

Bit 10: **ROFE/ROAK(0/1)** - Release on FIFO Empty/Release on Acknowledge.

Bit 9: **EN Quad CTR FIFO** - Bit 9 enables storage of the Quadrature Counter value each time a COS, or SOE event occurs.

Bits 08 through 04: **Reserved** - These bits must be set to zero (0) for correct FIFO operation.

Bit 03: **CTR_FIFO_FULL** = 1 - Quadrature Counter FIFO is full.

Bit 02: **CTR_FIFO_EMPTY** = 1 - Quadrature Counter FIFO is empty.

Bit 01: **MRK_Reset** = 1 - Marker Pulse resets quadrature count.

Bit 00: **CTR_ENA** = 1 - QUAD_CTR ENA: Setting this bit to a one (1) in conjunction with installing a jumper shunt on pins 7 and 8 of jumper E18, enables the quadrature counter function.

Data Register

The Data register is a 32-bit register, reflecting the current value of the channel inputs, or the value of the test latches when the Built-In-Test (BIT) function is enabled. It is readable on 8, 16, and 32-bit boundaries. In BIT mode, it is writable on 8, 16, and 32-bit boundaries.

Table 3-4 Data Register Bit Map

Data Register: Offset $XXXX04							
Bit 31	**Bit 30**	**Bit 29**	**Bit 28**	**Bit 27**	**Bit 26**	**Bit 25**	**Bit 24**
CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24

Data Register: Offset $XXXX05							
Bit 23	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16

Data Register: Offset $XXXX06							
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 09**	**Bit 08**
CH15	CH14	CH13	CH12	CH11	CH10	CH09	CH08

Data Register: Offset $XXXX07							
Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
CH07	CH06	CH05	CH04	CH03	CH02	CH01	CH00

Data FIFO Register

The Data FIFO register is a read-only register, containing the stored COS or SOE data. In the SOE mode of operation, the first read of this register will yield the value of the channel inputs previous to the change-of-state that triggered the storage of data. The second read will yield the final data that triggered the storage of data. The Data FIFO register is readable longword only.

NOTE: Since all reads to this register increments the FIFO counter to the next FIFO location, a read of less than a longword are ignored.

COS = 1, Read to capture the data
SOE = 2, Read to capture the data

Table 3-5 Data FIFO Register Bit Map

FIFO Register: Offset $XXXX08							
Bit 31	**Bit 30**	**Bit 29**	**Bit 28**	**Bit 27**	**Bit 26**	**Bit 25**	**Bit 24**
CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24
Bit 23	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit10**	**Bit 09**	**Bit 08**
CH15	CH14	CH13	CH12	CH11	CH10	CH09	CH08
Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
CH07	CH06	CH05	CH04	CH03	CH02	CH01	CH00

Interrupt Processor Control Register

This Read / Write register is used to configure the operations of the Interrupt Processor.

Table 3-6 Interrupt Processor Control Register Bit Map

Interrupt Processor Control Register: Offset $XXXX0C							
Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
MKR_INT_ENA	**Interrupt Levels**			**COS_INT_ENA**	**Interrupt Levels**		
	0	0	0		0	0	0
	L2	L1	L0		L2	L1	L0

Interrupt Levels

These three bits (L2 through L0) set the interrupt level that the IP will present to the host when a COS, or Marker request is made. The interrupt levels and the field values are:

Bits 7 through 4: Marker Interrupt Enable and Level bits.

Bits 3 through 0: COS Interrupt Enable and Level bits.

L2	L1	L0	IRQ Level
0	0	0	Disabled
0	0	1	IRQ1
0	1	0	IRQ2
0	1	1	IRQ3
1	0	0	IRQ4
1	0	1	IRQ5
1	1	0	IRQ6
1	1	1	IRQ7

The interrupt for the COS / SOE is cleared automatically when either the FIFO is read until empty, or the COS_INT_ENA bit is cleared. The Quadrature Marker Pulse interrupt is cleared when its interrupt is acknowledged, or when the MKR_INT_ENA bit is cleared.

Interrupt Processor COS Vector Register (Offset: $XXXX0D)

This is a read-write register, programmed by the user with the interrupt vector value desired for the COS interrupt bits 7 through 0.

Interrupt Processor Marker Vector Register (Offset: $XXXX0E)

This is a read-write register, programmed by the user with the interrupt vector value desired for the Marker interrupts bits 7 through 0.

Counter FIFO Register (Offset: $XXXX10)

This read-only 32-bit data register contains the value of the Quadrature Counter at the time that the data capture was triggered. It should be read once with each pair of reads to the Data register when used in SOE mode so that time stamp / data alignment will be maintained. This will result in the complete SOE data structure as shown:

- QUAD_Counter_Value
- PREV_Data
- COS_Data

In COS mode, this counter can also be used to provide a time / position stamp to stored data:

- QUAD_Counter_Value
- COS_Data

Setting Bit 9 in the CSR to one (1) enables this register.

Quadrature Counter Register (Offset: $XXXX14)

This register contains the current number of Quadrature pulses received from the front panel input (P3). The Quadrature Counter register is a read / write register, so that if the user wishes to begin the count with a predetermined value, that value may be directly written to the register with a Longword write. Each Quadrature pulse (series) results in the current count being increased by four (4) counts.

COS Select Register 0

This register's bits are used to select the COS trigger condition for bits 31 through 24.

Table 3-7 COS Select Register 0 Bit Map

COS Select Register 0 $XXXX18							
Bit 31	**Bit 30**	**Bit 29**	**Bit 28**	**Bit 27**	**Bit 26**	**Bit 25**	**Bit 24**
31B	31A	30B	30A	29B	29A	28B	28A

Bit 23	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
27B	27A	26B	26A	25B	25A	24B	24A

COS Select Register 1

This register's bits are used to select the COS trigger condition for bits 23 through 16.

Table 3-8 COS Select Register 1 Bit Map

COS Select Register 0 $XXXX1A							
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 09**	**Bit 08**
23B	23A	22B	22A	21B	21A	20B	20A

Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
19B	19A	18B	18A	17B	17A	16B	16A

COS Select Register 2

This register's bits are used to select the COS trigger condition for bits 15 through 8.

Table 3-9 COS Select Register 2 Bit Map

COS Select Register 0 $XXXX1C							
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 09**	**Bit 08**
15B	15A	14B	14A	13B	13A	12B	12A

Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
11B	11A	10B	10A	9B	9A	8B	8A

COS Select Register 3

This register's bits are used to select the COS trigger condition for bits 7 through 0.

Table 3-10 COS Select Register 3 Bit Map

COS Select Register 0 $XXXX1E							
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 09**	**Bit 08**
7B	7A	6B	6A	5B	5A	4B	4A

Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
3B	3A	2B	2A	1B	1A	0B	0A

COS SEL B/A

Channel x

These bits work in pairs for the stated channel. They define the COS trigger condition for the associated port. Their states and functions are:

B/A	Function
00	No interrupts (data only)
01	Rising edge interrupts only
10	Falling edge interrupts only
11	Any edge interrupt

Example:

Setting Bits 15 and 14 of address $XXXX1E will cause a COS trigger on any edge occurrence for channel 7.

FIFO Count Register (FIFO_CNT)

This 16-bit register contains the count of the number of samples in the COS FIFO.

NOTE: if SOE is enabled, the FIFO count will be 2x the number of SOE events since the FIFO contains both the previous state, and the changed state for each event.

Table 3-11 FIFO Count Register Bit Map

FIFO_CNT: Offset $XXXX20							
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 09**	**Bit 08**

Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**

Counter FIFO Count Register (CTR_FIFO_CNT)

This 16-bit register contains the count of the number of samples in the quadrature counter FIFO.

Table 3-12 Counter FIFO Count Register Bit Map

CTR_FIFO_CNT: Offset $XXXX22							
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 09**	**Bit 08**

Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**

Channel Interrupt Enable Register (CH_INT_ENA)

This 32-bit register allows the user to select, on a channel-by-channel basis, which COS/SOE events will trigger an interrupt. If the bit corresponding to a particular channel is set to a one (1), and that channel has a valid COS/SOE event, then the data is stored in the appropriate FIFO's, and a COS/SOE interrupt is set. If the correspoding bit is set to zero (0), then the event is stored in the appropriate FIFO's, but no interrupt is set.

The COS/SOE interrupt for a particular channel corresponds to the channel number, i.e. setting Bit 23 will enable the COS/SOE interrupt for events occurring on input channel 23.

Table 3-13 Channel Interrupt Enable Register Bit Map

CH_INT_ENA Register: Offset $XXXX24							
Bit 31	**Bit 30**	**Bit 29**	**Bit 28**	**Bit 27**	**Bit 26**	**Bit 25**	**Bit 24**
CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24
Bit 23	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit10**	**Bit 09**	**Bit 08**
CH15	CH14	CH13	CH12	CH11	CH10	CH09	CH08
Bit 07	**Bit 06**	**Bit 05**	**Bit 04**	**Bit 03**	**Bit 02**	**Bit 01**	**Bit 00**
CH07	CH06	CH05	CH04	CH03	CH02	CH01	CH00

Firmware Revision Register (FREV)

The Firmware Revision register is a 32-bit, read-only register located at offset $30. It is accessible as bytes, words or longword. For example, the current firmware revision is 1.2.17. This is read from the register as 00 01 02 17.

Quadrature Counter Feature

The VMIVME-1183 has the capability of accepting a Quadrature Counter (either single-ended or differential) input at the P3 connector on the front panel. The input range of this input is 0 to 48 V. There is also a marker input, which the VMIVME-1183 can treat as a counter reset signal, and a possible interrupt source. The marker input can be wired for either logic level high or low marker pulses. For logic high markers, the (+) differential lead should connect to the marker pulse (-) input, and the (-) differential lead should connect to the marker pulse (+) input. For single-ended logic high markers, connect the marker input to the Marker pulse (-) input. For logic low marker inputs, the connection scheme is opposite that of the logic high marker connections. The jumper shunt on E18 (pins 7-8) along with bit 0 of the CSR is used to enable the Quadrature Counter feature.

The VMIVME-1183 is also capable of passing the Quadrature counter signals to the output connector (P4). The output signals will match the input signals at P3 (single-ended or differential). The pinouts for P3 and P4 connectors are shown in Table 3-14 below.

Table 3-14 P3/P4 Connector Pinout

P3		P4	
Pin#	**Function**	**Pin#**	**Function**
1	Channel A (+) Input	1	Channel A (+) Output
2	Channel B (+) Input	2	Channel B (+) Output
3	Marker Pulse (+) Input	3	Marker Pulse (+) Output
4	N/C	4	N/C
5	Signal Ground	5	Signal Ground
6	Chassis Ground	6	Chassis Ground
7	Channel A (-) Input	7	Channel A (-) Output
8	Channel B (-) Input	8	Channel B (-) Output
9	Marker Pulse (-) Input	9	Marker Pulse (-) Output

Test Mode

Test Mode simply enables the outputs of the on-board data registers. These registers determine if the data being read from the board is valid. The output registers use the same addresses as their input data counterparts. This makes testing rather straightforward. Write to the address. Read from the address. Compare the data written to the data read, and determine if the board is working. This should be performed when Change-of-State operations are **not** active.

Test Mode determines if the data paths are working as well as the Change-of-State and interrupt logic. Test data written to these output registers overrides the incoming signals that go to the input registers. This data can be used in conjunction with interrupt levels and COS functions to yield a thorough evaluation of the board's functions. This kind of testing takes some time and should be done once at board initialization time before beginning data acquisition. This way you won't miss any COSs during testing.

If you are using Test Mode operations during COS, be careful of the FIFO data. In order to test the board the input data must be changed, generating a COS during Test Mode. This data will be placed in the FIFO, and can be read. Thus, Test Mode can be used to determine the "health" of all the electronic components on the board. However, the test data will be stored **behind** any externally generated COS data already stored in the FIFO. This means that the FIFOs should be emptied before Test Mode operations are started.

Test Mode, along with COS Selection and interrupt enables in the IP, gives the user many choices in how to test this board. For example, in the paragraphs above, it was assumed that the IP's interrupt enable bits were turned off and some form of COS logic is enabled, such as rising edge only. Now writing proper data to the board will generate a COS. This data will get stored in the FIFO. The input data and the FIFO can be read and compared to see if the board is working properly. This is just one of many different setups that can be used to test this board.

Example Setup:

Enter test mode.

Reset all four (4) COS Select registers to zero (0).

Run test until complete.

Exit test mode.

Rearm all four (4) COS Select registers

Initialization

Due to the nature of computers using the VMEbus, the following examples are just descriptive outlines of what you need to do in your program. We are also assuming that you understand the VMEbus priority interrupt system.

The following is an example setup of the VMIVME-1183. Since each input channel is individually programmable regarding the type of transition that will trigger an interrupt, caution should be used in configuration of the board.

We will program four inputs for each of the COS / SOE operating modes. We will configure Channel 31 for data transfers only. It will NOT be doing any interrupts. Channel 28 will cause an interrupt to be issued on rising edge transitions. This is when a data bit goes from a logic zero (0) to a logic one (1). Channel 26 will do falling edge interrupts. This is the opposite of Channel 28. Channel 25 will cause an interrupt to be issued on any transition (rising or falling).

The board will operate in SOE (Sequence Of Events) mode, with a Quadrature clock (using the Marker Pulse to reset the counter) input through the front panel connectors (P3, P4).

First of all, an interrupt level, and interrupt vector value must be assigned to the board. We will assign interrupt level 4 to COS / SOE interrupts, and a interrupt vector value of $83. Since we will be using this board in the SOE mode, with a Quadrature clock (with marker pulse) input to the front of the VMIVME-1183 card. We will set the marker pulse interrupt at level 2, with an interrupt vector of $45. The board will be set in ROFE (Release On FIFO Empty) mode.

The board in this example is addressed in A16 mode, and is jumper configured at a base address of $8000.

The required writes to the VMIVME-1183 board are shown in the following Table :

Table 3-15 Example Setup of the VMIVME-1183

Address	Register	Value	Task
$XXXX8002	CSR	0xCA03	Set up CSR
$XXXX8018	COS_SEL0	0x012C	Set monitoring conditions for Channels 31, 28, 26 and 25
$XXXX800C	IP_CTRL_REG		Set level for marker and COS interrupts
$XXXX800D	IP_COS_VECT	0x83	Set COS/SOE interrupt vector
$XXXX800E	IP_MKR_VECT	0x45	Set Quadrature Marker Pulse Interrupt Vector
$XXXX8024	CH_INT_ENA	0x16000000	Enable COS Interrupts

CSR may or may not have Test Mode active when these data transfers are being done. If test mode is active, the BIT registers will overwrite the incoming data. So you should turn OFF the test mode, assuming the board has passed your diagnostic programs. The Fail LED can be turned off at this time, too. A write of $CA03 to this location will turn off the LED, disable Test Mode, and leave the board operating in SOE mode with the Quadrature counter FIFO enabled.

The board is now ready to start looking for interrupts. Although data can be processed at any time, the interrupt enable bits should only be set when you are ready to handle interrupts. At this time the COS/SOE interrupt, and Marker interrupt enable bits will have to be set. One way is to write the entire byte of data again, only this time with the enable bits set. For this example, the value to write would be $AC. Now the board will issue interrupts as the COS logic and Marker Pulse dictate.

Maintenance

Maintenance

This section provides information relative to the care and maintenance of VMIC's products. If the product malfunctions, verify the following:

- System power
- Software
- System configuration
- Electrical connections
- Jumper or configuration options
- Boards are fully inserted into their proper connector location
- Connector pins are clean and free from contamination
- No components of adjacent boards are disturbed when inserting or removing the board from the chassis
- Quality of cables and I/O connections

If products must be returned, contact VMIC for a Return Material Authorization (RMA) Number. **This RMA Number must be obtained prior to any return**.

Contact VMIC Customer Care at 1-800-240-7782, or
E-mail: customer.service@vmic.com

Maintenance Prints

User level repairs are not recommended. The drawings and tables in this manual are for reference purposes only.